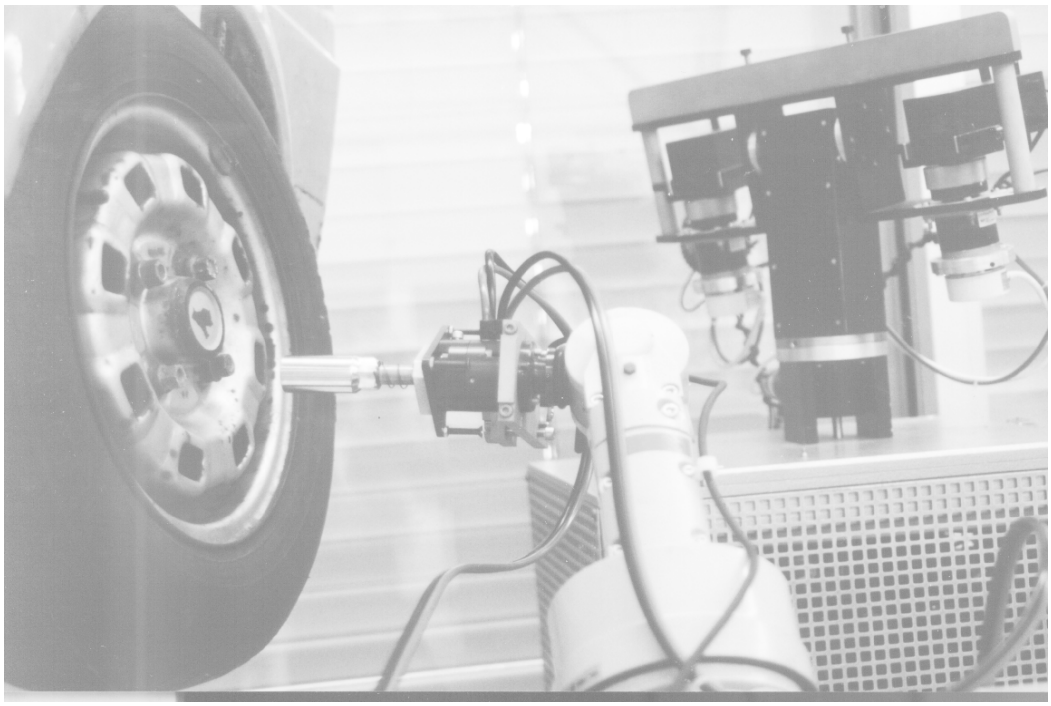
Aktive Szenenauswertung und Objekterkennung



Ulrich Büker
Universität Paderborn
Fachbereich Elektrotechnik und Informationstechnik

Vorwort

Die vorliegende Arbeit ist während meiner Zeit als Oberingenieur im Fachbereich Elektrotechnik und Informationstechnik der Universität Paderborn entstanden. In der Arbeitsgruppe von Herrn Prof. Dr. Hartmann fand ich durch die Mitarbeit in ambitionierten Forschungsprojekten die notwendigen hohen Anforderungen und gleichzeitig aber auch den nötigen Freiraum, um zu den hier beschriebenen Ergebnissen zu gelangen. Für die erfahrene Förderung möchte ich an dieser Stelle Herrn Prof. Dr. Hartmann herzlichst danken. Mein Dank gilt aber auch den Koreferenten dieser Habilitationsschrift, den Herren Prof. Dr. Sagerer und Prof. Dr. Kleine-Büning, für die von ihnen übernommene Aufgabe und ihre Anregungen.

Im besonderen Maße möchte ich auch meinen zahlreichen Kollegen, den Diplomanden und studentischen Hilfskräften danken, die durch ihre Diskussionsbeiträge und Implementierungsarbeiten an den praktischen Arbeiten mitgewirkt haben. Namentlich erwähnen möchte ich hierbei insbesondere Herrn Dr. Drüe, Herrn Hempel und Herrn Kalkreuter, sowie die Mitarbeiter des DEMON-Projektes Herrn Dr. Götze, Herrn Stemmer und Herrn Dr. Trapp.

Meiner Familie möchte ich für ihre Geduld und ihre Unterstützung danken, die sie speziell während der Endphase des Schreibens der Arbeit aufgebracht hat.

Paderborn, 2001

Ulrich Büker

Aktive Szenenauswertung und Objekterkennung

Inhaltsverzeichnis

1 Einleitung

Sehende Roboter - Fiktion oder Realität?

Roboter lernen das Sehen?! Erwartet uns mit dem Beginn eines neuen Jahrtausends eine erneute technische und industrielle Revolution? Vieles deutet darauf hin. Mit ungeheurer Intensität wird seit vielen Jahren von Wissenschaftlern unterschiedlicher Disziplinen an der Untersuchung des Sehens gearbeitet. Verschiedenste Systemarchitekturen wurden vorgeschlagen, um auch technischen Systemen das Sehen zu ermöglichen, und es wurden bereits einige bemerkenswerte Ergebnisse erzielt. So werden heute handgeschriebene Adressen auf Briefen automatisch gelesen und die Briefe entsprechend sortiert. Schecks werden gescannt und per Computer ausgewertet. Auch in der industriellen Produktion sind mittlerweile in vielen Bereichen Bildverarbeitungssysteme zu finden. Diese werden zur Zeit noch zum größten Teil zur Qualitätskontrolle und -sicherung eingesetzt. Aber auch im Bereich der Inspektion von Montageprozessen werden in einigen Anlagen bereits bildgebende Sensoren verwendet und ausgewertet. Der Verband Deutscher Maschinen- und Anlagebau e.V. (VDMA) beschreibt in einer aktuellen Erhebung, dass Bildverarbeitungssysteme

in der Industrie an einem Umsatzvolumen von knapp 30 Milliarden DM beteiligt sind.

Doch trotz aller Anstrengungen und aller Erfolge, die auf dem Gebiet der Bildverarbeitung bereits erzielt wurden, sind alle existierenden Systeme immer noch sehr weit davon entfernt, ihr Vorbild, das biologische Sehsystem und dessen Leistungen, zu erreichen. Auch das in dieser Arbeit beschriebene System erhebt nicht den Anspruch, dieses hochgesteckte Ziel zu erreichen. Es wird jedoch ein technisches System vorgestellt, das vielversprechende neue Möglichkeiten im Bereich des Roboter-Sehens, des *Robot Vision*, aufzeigen soll. Dabei wurden verschiedene Anleihen beim biologischen Vorbild genommen.

So stand bei den Architekturentscheidungen die Beobachtung Pate, dass der Mensch in der Lage ist, in seiner Umgebung Objekte bis zu einem gewissen Komplexitätsgrad auf einen Blick, d.h. *ganzheitlich*, *holistisch*, zu erkennen. Auch benötigt er kein aufwendiges Training, um sich ein neues, ihm zuvor unbekanntes, Objekt einzuprägen. Auf der anderen Seite erfolgt aber auch ein gezieltes „Umherschauen“ und „Sich Orientieren“ in einem komplexen Szenario. Desweiteren ist das menschliche Sehen kein passiver Vorgang. Vielmehr handelt es sich um einen bewussten, aktiven Prozess, bei dem - wenn es notwendig ist - gezielt neue Standpunkte und Blickwinkel eingenommen werden, um sich in einer unbekannten Umgebung zurecht zu finden. Unter Einbeziehung dieser vier Grundsätze menschlichen Sehens konnte in den vergangenen Jahren in Zusammenarbeit mit verschiedenen Kollegen ein Bilderkennungssystem aufgebaut werden, das

1. auf einer unteren Ebene eine holistische, biologisch motivierte Auswertung von Bildern durchführt;
2. auf einer übergeordneten Ebene modellgestützt komplexe Szenen analysiert und dabei
3. gezielt aktive Mechanismen einbezieht, so dass mittels einer beweglichen Kamera verschiedene für den Erkennungsprozess wichtige Bilder einer Szene aufgenommen werden und dabei
4. ein einfaches Lernen sowohl auf der unteren subsymbolischen als auch auf der höheren symbolischen Erkennungsebene unterstützt.

Sehende Roboter für die Automation

Da für die Implementierung des aktiven Sehens zum Zweck der Objekterkennung ein Roboter eingesetzt wird, ist es naheliegend, auch Aufgaben aus der Robotik als ein Anwendungsgebiet dieses Erkennungssystems zu betrachten. Somit wird der Roboter, der zunächst lediglich der Bewegung der Kamera diente, zum *sehenden* und somit flexibel auf seine Umwelt reagierenden Werkzeug. Hierdurch erschließen sich völlig neue Möglichkeiten für die Automatisierung. So können beispielsweise bei Montagevorgängen die Bauteile visuell inspiziert und vermessen werden. Hier kann nunmehr auf aufwendige Positioniersysteme verzichtet werden, da die Roboter nicht mehr auf feste Positionen programmiert werden müssen, sondern sich ihre Objekte selbst suchen und die Arbeitsprozesse auf die erkannten Verhältnisse anpassen.

Darüber hinaus lassen sich auch völlig neue Aufgabengebiete erschließen. Ein zukunftsträchtiges Beispiel hierfür ist die automatische Demontage von Objekten zu Recyclingzwecken. Im Gegensatz zur Montage kann bei der Demontage von Objekten zur Rückgewinnung von Rohstoffen grundsätzlich nicht von konstanten Umgebungsbedingungen ausgegangen werden, da die zu demontierenden Objekte beschädigt oder verändert sein können. Bei der in dieser Arbeit als Anwendungsbeispiel aufgezeigten Raddemontage an Altautos ist die Problemstellung unter anderem charakterisiert durch Variation der Lage (Abstand zum Demontageroboter, Radeinschlag, etc.) und des Aussehens der Räder (Felgendurchmesser, Anzahl der Muttern, Verschmutzung,...) sowie durch einen komplex strukturierten Hintergrund. Aus diesem Grund wird eine tolerante Erkennungsstrategie benötigt, die in der Lage ist, die dreidimensionale Struktur der Szene zu erfassen, um Greifvorgänge ohne Kollisionen realisieren zu können.

Die Struktur dieser Arbeit

Die vorliegende Arbeit beabsichtigt, einen Beitrag zur Entwicklung flexibel einsetzbarer, sehender Roboter zu leisten. Hierzu wird ein aktives Objekterkennungssystem entwickelt und vorgestellt. Die Implementierung des Systems wird detailliert beschrieben und es wird die

Leistungsfähigkeit des vorgeschlagenen Ansatzes anhand zweier Anwendungen aus dem Bereich der Robotik gezeigt.

Nach einer ausführlichen Darstellung der verschiedenen Paradigmen im Bereich der Bilderkennung (Kap. 2 und 3) werden in Kapitel 4 die dem System zugrundeliegenden Bildverarbeitungsverfahren vorgestellt, wobei auch systembedingte Probleme aufgezeigt werden. Unter Berücksichtigung der Diskussionsergebnisse aus den vorangegangenen Kapiteln wird dann ein hybrider Ansatz vorgeschlagen (Kap. 5). Dieser hybride Systemansatz ist gekennzeichnet durch die Verwendung eines holistischen Erkennungsansatzes, der eingebettet ist in eine dekompositorische Objektmodellierung mit semantischen Netzwerken auf der Basis von Objektansichten und Objektteilansichten. In Kapitel 6 wird die für die Modellierung verwendete Wissensbeschreibungssprache vorgestellt. Kapitel 7 beschreibt den Instanziierungsprozess, der versucht, die beste Zuordnung von Szenenelementen zu den Elementen der Modellierung zu finden. In Kapitel 8 wird erläutert, wie sich ein solcher Prozess als parallele Implementierung für SMP-Architekturen gestalten lässt, wie sie heute als Multiprozessor-Systeme sowohl im Workstation- als auch im PC-Bereich vertreten sind. In Kapitel 9 wird dann dieses hybride Objekterkennungssystem homogen zu einem aktiven Sehsystem erweitert. Dieses ist in der Lage, dynamisch die für die Szenenauswertung benötigten Bilder aufzunehmen. Kapitel 10 beschreibt die graphische Benutzerschnittstelle, die zur Darstellung des Wissens, der Abläufe bei der Szenenanalyse, zur Erklärung der Analyseergebnisse und zur Fehlersuche in Wissensbasen genutzt wird. Die Anbindung an ein Modul der virtuellen Realität dient darüber hinaus zur Simulation des aktiven Erkennungsprozesses in der Entwicklungs- und Testphase von Wissensbasen. In Kapitel 11 wird die Lernkomponente des Systems vorgestellt, die es ermöglicht, durch einmalige Präsentation der Objekte einer Domäne die für die Erkennung wichtigen Ansichten durch automatisches Bewegen der Kamera um die Objekte zu extrahieren und hieraus eine strukturelle Beschreibung dieser Objekte zu erzeugen. Die Leistungsfähigkeit des Systems wird in Kapitel 12 durch zwei verschiedene Anwendungen aus dem Bereich der Robotik aufgezeigt. Hierbei wird zunächst gezeigt, wie durch Einsatz eines Stereokamerakopfes in Zusammenhang mit Modellwissen eine Lokalisierung und Erkennung von Autorädern und ihrer Befestigungsschrauben möglich ist. Durch den Einsatz aktiver Mechanismen kann dabei die

Vermessung der Schrauben mit einer für den Demontagevorgang hinreichend großen Genauigkeit durchgeführt werden. In der zweiten Anwendung wird dann aufgezeigt, wie die zusätzlichen Freiheitsgrade einer an der Hand eines Industrieroboters befestigten Kamera genutzt werden, um komplexe dreidimensionale Objekte aktiv von verschiedenen Seiten zu betrachten und für die Erkennung charakteristische Details zu suchen. Abschließend werden in Kapitel 13 einige Schlussbetrachtungen und ein Ausblick auf zukünftige Arbeiten gegeben.

2 Die visuelle Wahrnehmung als aktiver Prozess

Mitte der 80er Jahre begründeten Aloimonos und Bajcsy mit ihren Arbeiten das Paradigma des *aktiven Sehens* [1], [5]. Sie konnten zeigen, dass durch aktives Sehen die Regularisierung bei der Rekonstruktion der sensorisch abgebildeten Welt erleichtert oder sogar überflüssig wird. Im diesem Kapitel soll zunächst einmal geklärt werden, was in diesem Zusammenhang unter dem Begriff des *aktiven Sehens* verstanden wird. Bevor dann eine kurze Übersicht über die verschiedenen Forschungsaktivitäten gegeben wird.

2.1 Aktives Sehen

Da der Begriff des *aktiven Sehens* in verschiedener Weise gebraucht und verstanden werden kann, soll an dieser Stelle zunächst geklärt werden, in welcher Art er im weiteren Verlauf des Textes verstanden werden soll. Es werden typischerweise zwei Interpretationen des *aktiven Sehens* unterschieden. Die eine betrachtet die Aktivität als Bestandteil des Sensors, der z.B. elektromagnetische Wellen aussendet und der dann das reflektierte Signal empfängt und auswertet, wie dies z.B. bei einem Sonar oder einem Radar der Fall ist. Bei dem ausgesendeten Signal kann es sich aber auch um strukturiertes Licht handeln, welches von den Objekten einer Szene reflektiert wird und mit einer herkömmlichen, passiven Kamera empfangen wird; eine durchaus gängige Vorgehensweise bei der optischen 3D Vermessung von Objekten. In jedem Falle wird aber bei dieser ersten Interpretation aktiv ein Signal ausgesendet, dieses wieder empfangen und ausgewertet.

Im Gegensatz dazu wird seit den richtungsweisenden Beiträgen von Aloimonos und Bajcsy am Ende der 80er Jahre unter *aktivem Sehen* der aktive Einsatz von passiven Sensoren verstanden. Dabei geht es darum, einen passiven Sensor in einer aktiven Art und Weise einzusetzen, bei der zielorientiert, entsprechend einer Strategie zur Lösung der gestellten Aufgabe, der Zustand des Sensors verändert wird. Dies lässt sich allgemein auf aktives Wahrnehmen beziehen und bedeutet im hier betrachteten Fall des *aktiven Sehens* die Veränderung des Kamerazustandes. Hierzu gehören die Veränderung von Position und Blickrichtung der Kamera im Raum, also die Veränderung der extrinsischen Parameter, sowie die Einflussnahme auf die Bildaufnahme durch Veränderung der intrinsischen Parameter Fokus, Blende und Brennweite der Kamera.

In diesem Beitrag wird sich die aktive Manipulation auf die Position und Blickrichtung der Kamera beziehen, d.h. es werden sechs Freiheitsgrade (drei translatorische und drei rotatorische) Freiheitsgrade genutzt. In begrenztem Umfang wird auch die Veränderung der Brennweite berücksichtigt, während Fokus und Blende nicht aktiv verändert werden, sondern je nach technischen Voraussetzungen der eingesetzten Kameras konstant gelassen werden oder aber der kameraeigenen Auto-

matik überlassen werden, das heißt, es werden Autofokus und automatische Belichtungssteuerung verwendet.

Aloimonos hat in seiner Arbeit herausgestellt, dass durch die Integration von Aktivität in ein visuell wahrnehmendes System, viele Probleme umgangen oder einfacher gelöst werden können als im Fall rein passiver Systeme, in denen ein einzelnes Bild ausgewertet wird. So können Probleme, die für einen passiven Beobachter schlecht gestellt (ill-posed), nichtlinear oder unstabil sind, für einen aktiven Beobachter gut gestellt (well-posed), linear und stabil werden. Er geht dabei vor allem auf die Probleme des *shape from shading*, *shape from contour*, *shape from texture*, *structure from motion* und des *optischen Flusses* ein [1]. Er stellt dabei ganz deutlich in den Vordergrund, dass natürlich auch der Mensch als Vorbild aller technischen, visuell wahrnehmenden Systeme ein aktiver Beobachter ist, der seine Position in der Umgebung, seine Blickrichtung auf ein Objekt oder in ein Szenario hinein verändern kann und dies auch tut, wenn seine Zielsetzung auf diese Weise besser oder einfacher erfüllt werden kann.

> „Human perception, however, is not passive, it is active. Perceptual activity is exploratory and searching. When humans see and understand, they actively look.“ ([1], Seite 552)

Bajcsy betrachtet dem gegenüber das aktive Sehen weniger als ein neues wissenschaftliches Paradigma. Sie stellt vielmehr die Modellierung von Wahrnehmungsstrategien in den Vordergrund. Dabei sieht sie ein solches System als ein rückgekoppeltes System, indem in Abhängigkeit der Sensordaten eine Veränderung des Sensorzustandes herbeigeführt wird, der zu neuen Sensordaten führt. Aufgrund der Komplexität der Verarbeitungsschritte bei der Auswertung der Sensordaten, die bis zur Erkennung und Vermessung komplexer dreidimensionaler Objekte in natürlichen Szenen reichen kann, und aufgrund der Abhängigkeit von a priori Wissen in numerischer und symbolischer Form sieht

sie den Prozess des aktiven Sehens jedoch nicht als rein regelungstechnisches Problem [5].

> „Perceptual activity is exploratory, probing, searching; percepts do not simply fall onto sensors as rain falls onto ground. We do not just see, we look. and in the course, our pupils adjust to the level of illumination, our eyes bring the world into sharp focus, our eyes converge or diverge, we move our heads or change our position to get a better view of something, and sometimes we even put on spectacles.“ ([5], Seite 996)

Das das aktive Sehen und dabei speziell die aktive Objekterkennung in komplex strukturierten Szenarien nicht als rein regelungstechnisches Problem im klassischen Sinne betrachtet werden kann, wird auch im Laufe dieser Arbeit deutlicht werden. Es wird bei der aktiven Objekterkennung in einem hohen Maße auf Wissensmodellierung und -verarbeitung zurückgegriffen. Bajcsy bedient sich daher auch des Begriffes der intelligenten Regelungstechnik (*intelligent control theory*), die Entscheidungsfindung, Begründung und Kontrolle beinhaltet. Dabei sind aber wiederum klassische Regelkreise Bestandteil eines aktiven Sehsystems, z.B. wenn es darum geht, die Motoren zur Bewegung der Kamera zu steuern. Diese klassische Regelung wird jedoch nicht Bestandteil dieser Arbeit sein. Wenn bei der Vorstellung der eigenen Arbeiten auf diesem Gebiet davon geschrieben wird, dass die Kamera einen neuen Blickwinkel einnimmt, so wird davon ausgegangen, dass die existierende Motorsteuerung dies in ausreichender Genauigkeit sicherstellt. Dabei soll hier schon erwähnt werden, dass an diese Genauigkeit nur sehr niedrige Anforderungen gestellt werden.

Anstelle des Begriffes *aktives Sehen* bevorzugt Ballard den Begriff des *animate vision*, des *belebten Sehens*, um die eingangs erwähnte Abgrenzung zu der aktiven Sensorik im Sinne des aktiven Aussendens von Signalen zu treffen [8]. Er sieht dabei das Hauptziel des *animate vision* in einem geeigneten Zusammenspiel zwischen Sehen und intelligentem Verhalten. Damit betrachtet er das aktive Sehen als einen zielgerichteten intelligenten Prozess in Form eines *Wahrnehmungs-Handlungs-Zykluses*. Der Deutlichkeit halber möchte ich jedoch im weiteren bei dem Begriff des *aktives Sehens* bleiben. Ballard legt dabei in seinem grundlegenden Artikel zu diesem Thema den Schwerpunkt auf die Blicks-

teuerung und ihre Bedeutung im menschlichen Sehsystem, das mit Sakkaden d.h. Blicksprüngen arbeitet, um die Fovea so zu positionieren, dass die zur Lösung einer gestellten Aufgabe wichtigen Bild- oder Szenenbereiche in ihr zu liegen kommen [8].

> „The goal of animate vision is the use of vision in bahaviors associated with intelligence, and as such it has its root in theories of robot behaviors. ... Animate vision als has its roots in the study of vision of the lower animals.“ ([8], Seite 58)

Ballard stellt dabei sieben Vorteile von Sehsystemen mit Blicksteuerung heraus:

1. *Aktive Sehsysteme können physikalisch suchen*. Die Kamera kann somit näher an ein Objekt herangebracht werden oder die Blickrichtungen können geändert werden.
2. *Aktive Sehsysteme können bekannte Kamerabewegungen durchführen*, um so im Sinne von Aloimonos durch Einführung zusätzlicher Nebenbedingungen die Problemlösung zu erleichtern oder gar erst zu ermöglichen.
3. *Aktive Sehsysteme können außerhalb der Kameras positionierte (d.h. exozentrische) Koordinatensysteme verwenden*. Durch die Möglichkeit des Fixierens von Gegenständen in einer Szene, ist es möglich, externe Koordinatensysteme an Punkten in der Szene außerhalb des Beobachters zu installieren. Positionsberechnungen können dann mit geringerer Genauigkeit relativ zu diesen Koordinatensystemen durchgeführt werden.
4. *Aktives Sehen kann qualitative Algorithmen verwenden*. Die visuo-motorische Kontrolle von Verhaltensweisen kann relativ zum Fixationspunkt durchgeführt werden.
5. *Eine Blicksteuerung kann eine Szene in interessante Regionen segmentieren*, in dem z.B. mittels einer Tiefenkarte Regionen, die in derselben Entfernung wie der Fixationspunkt liegen, segmentiert werden, ohne dass diese zuvor kategorisiert oder sogar erkannt werden.

6. *Aktive Sehsysteme können den Umweltkontext auswerten* und dabei ebenfalls mit Hilfe der Blicksteuerung Objektpositionen bestimmen.
7. *Aktives Sehen vereinfacht das Lernen*, da die Blicksteuerung und die hohe Auflösung der Fovea mit der damit verbundenen Isolation visueller Merkmale Indizierungsverfahren effizient unterstützen.

Ballard legt dabei besonderen Wert auf die Wichtigkeit exozentrischer Koordinatensysteme, mit der er von den Vorstellungen Marrs [112] abweicht, dass das Sehsystem in einem beobachterzentrierten Koordinatensystem arbeitet. Ballard sieht das Koordinatensystem als ein weltbasiertes aber beobachterorientiertes System an, das im Fixationspunkt des Sehsystems verankert ist und in der Orientierung mit der Blickrichtung abgestimmt ist. Er spricht in diesem Zusammenhang von einem *Fixationsrahmen*, in dem ein einfacher Zugriff auf Informationen über kleine Regionen nahe des Fixationspunktes möglich ist, was seiner Meinung nach besonders im Bereich der *frühen Bildverarbeitung*, des *early vision,* von besonderer Bedeutung ist.

> „One of the most central aspects of animate vision is the use of an exocentric coordinate system termed the frame of fixation. This frame provides direct access to information from a small region near the fixation point. Of particular importance is the information associdated with early vision.“ ([8], Seite 63)

Ballard geht dabei jedoch nicht näher darauf ein, wie sich bei ständig wechselndem Blickpunkt ein beobachterorientiertes Koordinatensystem verhält und ein Update aller relevanten Informationen erfolgen soll.

Die von allen Autoren hervorgehobene Bedeutung der motorischen Veränderung des Kamerazustandes ist auch in dieser Arbeit unter *aktivem Sehen* gemeint. Verbunden mit dieser motorischen Veränderung der Kameraparameter sind verschiedene Teilgebiete oder Anwendungsgebiete des aktiven Sehens. Diese sollen im folgenden Kapitel kurz umrissen werden, um auch auf diese deskriptive Art den Begriff noch näher festzulegen und um vor allem das Forschungsgebiet noch näher zu umreissen.

2.2 Forschungs- und Anwendungsgebiete

In einer ganzen Reihe von Arbeiten wurden in den 90er Jahren unterschiedliche Aspekte aus diesem Themenkreis des aktiven Sehens untersucht. Hierzu gehören insbesondere:

- der Aufbau und die Kalibrierung von Kamerasystemen
- die Rekonstruktion von Tiefeninformation aus Stereobildern
- die Aufmerksamkeits- und Blicksteuerung
- die Bewegungserkennung, -prediktion und Objektverfolgung
- die visuell geführte Navigation und Raumexploration
- die aktive Objekterkennung
- die Auge-Hand-Koordination und deren Integration in Robotersysteme

Im folgenden sollen diese Arbeitsgebiete kurz erläutert werden.

Aufbau und Kalibrierung von Kamerasystemen

Von grundlegender Bedeutung für praktische Arbeiten im Bereich des aktiven Sehens ist das Vorhandensein von Kamerasystemen, die die notwendige motorische Veränderung der Kameraparameter erlauben, um somit einen aktiven Sehvorgang zu unterstützen bzw. erst zu ermöglichen. Aus diesem Grunde haben sich einige Arbeitsgruppen zunächst einmal mit dem Aufbau geeigneter Kamerasysteme beschäftigt. Erwähnt werden sollen hierbei speziell die Kamerasysteme, die in den Arbeitsgruppen an der Universität Stockholm und der University of Pennsylvania entwickelt wurden und inzwischen auch kommerziell vermarktet werden [110], [144]. Daneben existieren auch eine Reihe weiterer universitärer Systeme, die zum Teil unter anderen Gesichtspunkten wie z.B. Größe, Kosten, Flexibilität und Modularität entwikkelt wurden wie z.B. die Systeme der Universitäten Aalborg, Rochester oder Paderborn [23], [46], [174]. Einen umfassenden Überblick über die existierenden Systeme und deren Einsatz in verschiedenen wissenschaftlichen Einrichtungen bietet [33].

Die Kalibrierung eines Kamerasystems ist neben dem mechanischen Aufbau Hauptvoraussetzung für die Umsetzung aktiver Mechanismen, denn nur durch eine Kalibrierung ist eine einfache und schnelle motorische Steuerung möglich. Bestimmt werden hierbei sowohl interne als auch externe Kameraparameter, wie etwa die Brennweite, der Hauptpunkt, die Parameter zur Beschreibung der Linsendistorsion oder auch die Position der motorischen Drehachsen in Bezug zum Kamerahauptpunkt [176]. Die Kamerakalibrierung ist zudem auch elementare Voraussetzung für die Rückgewinnung der Tiefeninformation.

Rekonstruktion von Tiefeninformation

Eine der wichtigsten Verfahrensklassen zur Rekonstruktion der Tiefeninformation aus Bilddaten beruht auf der Stereoskopie. Üblicherweise werden hierbei zwei - selten auch drei - Kameras verwendet, mit denen die auszuwertende Szene aus verschiedenen Positionen aufgenommen wird. In den Bildern müssen dann korrespondierende Bildpunkte ermittelt werden, um hieraus Disparitäten und anschließend eine Tiefenkarte zu bestimmen. Zur Lösung des Korrespondenzproblems, die i.a. nicht eindeutig ist, kann man zwischen drei Verfahrensklassen unterscheiden:

- merkmalsbasierte Verfahren,
- Korrelationsverfahren,
- phasenbasierte Verfahren.

In Verfahren der ersten Klasse werden aus den Bildern Merkmale extrahiert und einander zugeordnet [67], [75]. In der zweiten Klasse werden direkt die Grau- oder Farbwerte einer lokalen Umgebung eines jeden Bildpunktes des einen Bildes mit denen des anderen Bildes korreliert [59]. Für die phasenbasierten Ansätze wird der Verschiebungssatz der Fouriertransformation genutzt. Auch hierbei wird wieder in der lokalen Umgebung eines Bildpunktes gearbeitet [58]. Ein auf der Phasendifferenzmethode basierender Ansatz wird auch in den in Kapitel 12 beschriebenen Beispielanwendungen zur Disparitätsmessung verwendet [173].

Blicksteuerung

Ballard hat wie bereits eingangs erwähnt in seiner Arbeit zum aktiven Sehen die besondere Bedeutung einer Blicksteuerung hervorgehoben. Eine Vielzahl von Arbeiten haben sich seitdem mit der Aufmerksamkeits- und Blicksteuerung beschäftigt. In dieses Arbeitsgebiet fließen sowohl Aspekte der Neuropsychologie, der Physiologie als auch der Informationstheorie ein. Technische Realisierungen solcher Aufmerksamkeitssteuerungen finden sich z.B. in den Arbeiten von Brown, Fukushima, Gross, Tsotsos und Ullmann [62], [71], [153], [178], [180].

Navigation

Autonome Fahrzeuge, die eigenständig navigieren und ihren Arbeitsraum explorieren finden sich aufgrund der großen Komplexität solcher Systeme nur in wenigen großen Arbeitsgruppen wieder. Die führenden Arbeiten auf diesem Teilgebiet kommen dabei sicherlich aus den Arbeitsgruppen Dickmanns, Kanade und v. Seelen mit ihren Arbeiten am führerlosen Fahren im Straßenverkehr [51], [149], [183].

Hand-Auge-Koordination

Für die sichtbasierte Steuerung von Handhabungsrobotern stellt sich das Regelungsproblem, aufgrund der Bildinformation den Endeffektor des Roboters an ein Zielobjekt heranzuführen, um dieses zu greifen oder anderweitig zu manipulieren. Hierbei werden zwei Kategorien von Systemen unterschieden:

- ❒ bildbasierte Lageregelung
- ❒ positionsbasierte Lageregelung.

Bei der bildbasierten Lageregelung wird die Abweichung einer Menge von Bildmerkmalen von ihren a priori festgelegten Zielpositionen im Bild minimiert [188]. Bei positionsbasierten Ansätzen wird hingegen mit geometrischen Objektmodellen und den Abbildungseigenschaften der Kamera aus den Bilddaten die räumliche Position und Orientierung des Zielobjektes bestimmt und ein resultierender Lagefeh-

ler minimiert [76]. Zu diesen positionsbasierten Ansätzen sind z.B. die Arbeiten aus der Arbeitsgruppe Hirzinger zu zählen [194].

Aktive Objekterkennung

Die aktive Objekterkennung ist im Umfeld des aktiven Sehens das sicherlich am wenigsten untersuchte Themengebiet. Da in dem folgenden Kapitel noch eingehend auf die gängigen Techniken zur Objekterkennung und zur aktiven Objekterkennung eingegangen wird, soll hier nur ein kurzer Abriss über diesen Bereich des aktiven Sehens gegeben werden.

Die Idee, eine Objekterkennung auf der Basis mehrerer, unter Umständen auch aus verschiedenen Blickrichtungen aufgenommener Bilder durchzuführen, ist im Prinzip bereits so alt, wie die Formulierung des Paradigmas des Aktiven Sehens. Rosenfeld hat dies bereits recht allgemein in seine Definition des Rechnersehens einfließen lassen:

> „The general goal of computer vision is to derive a description of a scene by analyzing one or more images of a scene.“ (aus [157], Seite 1).

Trotzdem sind seitdem nur sehr wenige aktive Objekterkennungssysteme entstanden. Ein Grund liegt sicherlich darin, dass zunächst einmal die Voraussetzungen hierfür geschaffen werden mussten. Eine aktive Objekterkennung war erst dann möglich, als die entsprechenden steuerbaren Kamerasysteme zur Verfügung standen, als das Gebiet der Kalibrierung untersucht war, als Fovealisierungsstrategien entwickelt waren und letztlich auch als genügend Wissen zum Thema der Modellierung und Erkennung dreidimensionaler Objekte zur Verfügung stand.

Grundlegende Idee der aktiven Objekterkennung ist dabei die aktive Einflussnahme auf die Bildaufnahme und deren Auswertung im Sinne des eingangs dieses Kapitels erläuterten von Aloimonos und Bajcsy formulierten Paradigmas des aktiven Sehens. Daraus ergibt sich unmittelbar das Heranziehen mehrerer Bilder einer Szene für die Auswertung. Desweiteren besteht die Notwendigkeit einer Kopplung der mo-

torischen Steuerung zur Realisierung des aktiven Prozesses in den eigentlichen Erkennungsprozess.

Im folgenden Kapitel soll aufgrund der zentralen Rolle der aktiven Objekterkennung in dieser Arbeit zunächst einmal aufgearbeitet werden, welche Ansätze und Paradigmen zur Objekterkennung im allgemeinen und zur aktiven Objekterkennung im speziellen derzeit Verwendung finden.

3 Objekterkennung: ein Vergleich bestehender Ansätze

In diesem Kapitel soll nun ein ausführlicher Abriss über die verschiedenen Paradigmen der Objekterkennung und über einige konkrete Umsetzungen in Erkennungssysteme gegeben werden. Dabei ist festzustellen, dass im wesentlichen zwei dieser Paradigmen die Arbeiten im Bereich der Objekterkennung prägen: zum einen werden strukturbasierte Beschreibungen von Objekten verwendet, zum anderen werden Objekte ansichtenbasiert beschrieben. Diese zwei Vorgehensweisen, die in den nachfolgenden Abschnitten näher erläutert werden, stellen jedoch nur sehr abstrakte Rahmen dar, die einige in der konkreten Umsetzung noch zu lösende Probleme offen lassen.

3.1 Paradigmen der Objekterkennung

Bei dem hier gegebenen Überblick soll eine Beschränkung auf die Erkennung von dreidimensionalen Objekten in Farb- oder Grauwertbildern erfolgen, also eine Beschränkung des Sensors auf herkömmliche Videokameras. Es soll hier nicht auf andere Sensoren wie zum Beispiel Sonar, Laser o.a. eingegangen werden, die direkt Tiefeninformation liefern.

Untersucht man die bestehenden Ansätze auf Gemeinsamkeiten, so kann man erkennen, dass im Prinzip zunächst einmal zwei Fragestellungen unterschieden werden. Die eine ist die Frage nach der Objektrepräsentation, die andere darauf aufbauend und in enger Beziehung stehend die Frage nach der eigentlichen Erkennung der dem System bekannten Objektmodelle im präsentierten Bild. Einen Überblick über Forschungsergebnisse zu diesen Fragestellungen geben die Übersichtsartikel von Besl und Jain [10] sowie von Chin und Dyer [45]. Eine Zusammenfassung der wichtigsten Verfahren und Ansätze gibt auch Rosenfeld in seinem Übersichtsartikel [157], in dem er die besonderen Probleme der Erkennung dreidimensionaler Objekte gegenüber der Auswertung zwei- oder quasi zweidimensionaler Szenarien wie folgt charakterisiert. Während im zweidimensionalen Fall drei Freiheitsgrade (x-, y-Verschiebung und eine Verdrehung in der Bildebene) berücksichtigt werden müssen, sind dies im dreidimensionalen Fall sechs Freiheitsgrade (drei translatorische und drei rotatorische Freiheitsgrade). Dies hat zur Folge, dass zum einen auch bei gleichmäßiger Ausleuchtung einer Szene selbst uniform reflektierende Oberflächen keinen konstanten Grauwert besitzen. Zum anderen kann eine gleichmäßige Ausleuchtung auch gar nicht garantiert werden, da zum Beispiel Schattenwurf unvermeidbar ist. Darüber hinaus ist die Formanalyse deutlich schwieriger, da durch die zusätzlichen rotatorischen Freiheitsgrade ein dreidimensionales Objekt viele verschiedene zweidimensionale Projektionen in der Bildebene erzeugen kann. Zusätzlich ist von einem gegebenen Blickpunkt aus nur eine Seite des Objektes sichtbar und selbst diese Ansicht kann noch durch andere Objekte in der Szene teilweise verdeckt sein, so dass also eine Erkennung immer auf der Basis unvollständiger Daten durchgeführt werden muss.

Einen aktuellen Stand spiegelt der von Hebert, Ponce, Boult und Gross herausgegebene NSF-ARPA Workshop-Bericht zum Thema Objektrepräsentation dar, der eine Reihe von Beiträgen der wichtigsten aktuellen Forschungsarbeiten zu diesem Thema beinhaltet [80].

Aus dem auf diesem Workshop vorgestellten Arbeiten, die einen recht guten Überblick über die aktuellen Forschungsarbeiten zum Thema Objekterkennung darstellen, wird deutlich, dass auch nach zehn Jahren Forschung zu dieser Fragestellung noch keine allgemeingültigen Antworten gegeben werden können. In vielen Fällen sind die Arbeiten noch nicht über ein Labor-Versuchsstadium hinausgekommen. Dies wird zum Teil auch darauf zurückgeführt, dass in zu geringem Maße Verfahren zur Bewertung der Güte eines Ansatzes zur Verfügung stehen und dass es - von außen betrachtet - in vielen Fällen nur sehr schwierig ist, zu beurteilen, wo die Probleme und Grenzen eines Verfahrens zu finden sind.

> „It is hard to assess quantitatively what represents good performance, but there is a reasonable widespread sense in the community that „good“ systems have large model bases, can operate on „complex“ pictures and offer insights that can be used to generate better approaches.“ ([80], S.15)

Konsens ist aber, dass in einem Objekterkennungssystem drei Aspekte berücksichtigt werden müssen: die Objektrepräsentation oder -modellierung, eine Kontrollstrategie zur Auswertung des Bildmaterials und das eigentliche Matchingverfahren, d.h. der Zuordnung der Bildinformation zu den Objektmodellen.

Objektzentrierte Repräsentation

Im wesentlichen werden bei der Objektmodellierung dreidimensionaler Objekte zwei Strategien unterschieden: die *objektzentrierte* und die *beobachterzentrierte* Repräsentation. Die Namensgebung macht bereits deutlich, dass im ersten Fall bei der Modellierung vom Objekt ausgegangen wird, welches dreidimensional ist. Es wird daher auch eine dreidimensionale Modellierung durchgeführt, bei der parametrisierte Oberflächen- oder Volumenelemente verwendet werden. Hier

existiert ein enger Bezug zur CAD-Konstruktion von Gegenständen, die ja ebenfalls explizite dreidimensionale Daten verwenden. Dies können einfache 3D Primitive wie Kegel, Quader oder Zylinder sein, aus denen komplexere Objekte zusammengefügt werden. Es können im Raum rotierte Flächen sein oder auch komplexe im Raum verlaufende Polygonzüge, mit denen ein Körper beschrieben wird. Im Gegensatz zur Konstruktion, steht ein Bilderkennungssystem aber vor der Frage, wie ein bereits existierendes Objekt zerlegt werden kann, welche Primitive hierfür verwendet werden sollen, bzw. wie eine geeignete Parametrisierung hierfür aussieht und nicht zuletzt, wie denn ein dreidimensionaler Körper in den zweidimensionalen Bilddaten erkannt werden kann. Darüber hinaus ist ungeklärt, wie natürliche Objekte durch solche Primitve modelliert werden können, so dass in heutigen objektzentrierten Systemen üblicherweise recht einfach strukturierte Objekte untersucht werden. Dies kann aber häufig auch in komplex strukturierten Umgebungen geschehen.

Hebert fasst die Probleme der objektzentrierten Repräsentation in folgenden Punkten zusammen ([80], S.16):

- Es ist noch nicht gelöst, wie eine Dekomposition des Objektes in Volumenprimitive erfolgen muss und wie diese Volumenprimitive im Bild gefunden werden können.
- Es ist nicht bekannt, ob eine kanonische oder stabile Dekomposition nur aufgrund von Bildinformation gefunden werden kann.
- Es ist unbekannt, welche Relationen zwischen den Beschreibungselementen aus einem Bild extrahiert werden können und wie diese Relationen genutzt werden können, um den Erkennungsprozess effektiv zu unterstützen.

Beobachterzentrierte Repräsentation

Einen gänzlich anderen Ansatz zur Objektrepräsentation verfolgen hingegen die *beobachterzentrierten* oder auch *ansichtenbasierten* Verfahren. Hier wird, wie der Name schon andeutet, vom Beobachter ausgegangen. Es wird also untersucht und modelliert, wie ein Objekt dem Beobachter erscheint, wie also die zweidimensionale Abbildung eines

Objektes auf der Retina des Auges bzw. im Kamerabild aussieht. Ein dreidimensionales Objekt wird dabei durch eine Menge von zweidimensionalen Ansichten beschrieben, die üblicherweise durch ein Abtasten der Ansichtensphäre um ein Objekt in diskreten Abtastschritten erzeugt werden. Die Aufgabe besteht nun darin, die Elemente dieser Ansichtenmenge mit einer unbekannten Ansicht zu vergleichen. Es ist somit nicht nötig, explizit die dreidimensionale Struktur des Objektes zu repräsentieren oder aus dem Bild zu extrahieren. Hiermit gehen aber andere typische Probleme einher. Da keine explizite Strukturinformation eines Objektes gegeben ist, kann eine Objekt-Hintergrundtrennung lediglich rein datengetrieben erfolgen, ohne unterstützende Einschränkungen aus dem Objektmodell verwenden zu können. Aus diesem Grund können - gerade anders als im Falle der objektzentrierten Repräsentation - häufig recht komplexe Objekte beschrieben werden, dies aber nur in wohlstrukturierten Szenarien ohne übermäßiges Rauschen oder Verdeckungen der Objekte. Hebert fasst die offenen Fragestellungen wie folgt zusammen ([80], S.17):

- Es existiert zur Zeit noch keine befriedigende Antwort auf die Frage, wie die Ansichten eines Objektes ausgewählt werden sollen, wieviele Ansichten eines Objektes für eine Modellierung und zuverlässige Erkennung benötigt werden und wie eine große Menge von Ansichten effektiv verwaltet und bei der Erkennung auch durchsucht werden kann.
- Es ist unbekannt, inwiefern eine rein beobachterzentrierte Repräsentation für eine Abstraktion eines Objektbegriffes genutzt werden kann. Es ist ebenfalls noch unklar, wie eine Erkennung nur teilweise sichtbarer Objekte erfolgen kann.
- Es ist bislang noch nicht ausreichend untersucht, in welchem Ausmaß sich Erkennungshypothesen gegenseitig beeinflussen und wie sie interagieren.
- Es ist noch unbekannt, wie eine ansichtenbasierte Repräsentation Kontextwissen für eine Erkennung zur Verfügung stellen kann.

Im weiteren Verlauf der Arbeit, speziell in den Kapiteln 4 und 6, werden einige Lösungsvorschläge zu den eben aufgeführten Fragestellungen bezüglich der objektzentrierten und der beobachterzentrierten Objektrepräsentation und -erkennung gegeben. Zunächst aber sollen

nach diesem Überblick über die zwei wichtigsten Paradigmen der Objekterkennung nun im folgenden Abschnitt noch einige Details erläutert und auch einige konkrete Implementierungen vorgestellt werden.

Dabei ist festzustellen, dass sich aktuell die Fragestellung nach Objekterkennung quasi in allen Fällen auf die Erkennung von dreidimensionalen Objekten bezieht. Desweiteren unterscheiden sich die Arbeiten zum Teil sehr stark von einander, auch wenn sie alle in die oben bereits kurz benannten zwei Paradigmen hineinfallen. Zunächst sollen davon nun die strukturbasierten Ansätze erläutert werden.

3.2 Strukturbasierte, objektzentrierte Ansätze

Strukturbasierte Ansätze zeichnen sich dadurch aus, dass sie das zu behandelnde Objekt in den Mittelpunkt der Beschreibung stellen. Dazu wird ein Objekt explizit in seiner dreidimensionalen Struktur beschrieben. Bekannte Vertreter dieser Ansätze sind z.B. Binford, Marr und Biedermann [11], [12], [14], [112]. Eine solche explizite dreidimensionale Beschreibung kann durch eine Menge von 3D Punkten auf der Oberfläche des Objektes geschehen [87], es können Polyeder sein [85], [109], es können algebraische Oberflächenbeschreibungen verwendet werden [95], [96] oder es kann eine strukturelle Beschreibung durch Zerlegen des Objektes in Volumenprimitive verwendet werden. Solche Volumenprimitive werden auch als *geons* bezeichnet. Dies können z.B. Quader oder Zylinder sein, es können aber allgemeiner auch sogenannte generalisierte Zylinder sein [134]. Auf der Basis eines solchen 3D-Objektmodells kann dann eine Objekterkennung erfolgen, unabhängig in welcher räumlichen Lage das Objekt dem Erkennungssystem präsentiert wird. Dies ist zunächst einmal eine sehr allgemeiner Rahmen einer Objektmodellierung. Innerhalb dieses Rahmens stellen sich bei der Umsetzung verschiedene Detailfragen, wie z.B. die Frage nach dem genauen Typus der Primitive, ihrer Parametrisierung, der Exaktheit der Modelle und daraus resultierend die Anzahl der für die Modellierung notwendigen Primitive.

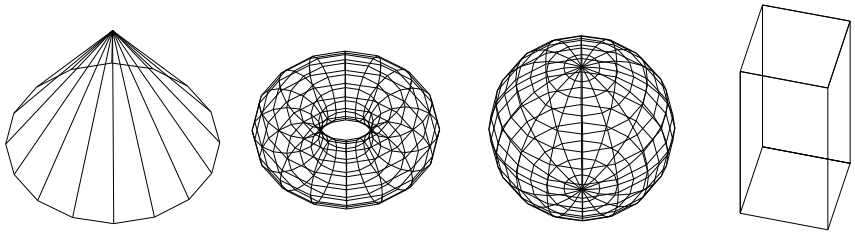


Abb. 3.1: Einige einfache Volumenprimitive für die objektzentrierte Modellierung

Die RBC-Theorie

Biederman beschreibt in seinen Arbeiten nicht nur eine Erkennungstheorie, die er RBC-Theorie nennt, *recognition by components*, sondern auch eine Reihe von psychophysikalischen Untersuchungen zum Objekterkennungsproblem. Mit diesen untermauert er seine Theorie, dass sich ein Objekt durch eine Menge von Volumenprimitiven wie Quader, Zylinder etc. beschreiben lässt. Diese Primitive nennt er *geometric icons* oder kurz *geons.* Er geht davon aus, dass nur relativ wenige *geons* für die Objekterkennung benötigt werden ($n \leq 36$) und dass sich jedes Objekt durch sehr wenige ($n < 10$) in unterschiedlicher Größe und Anordnung beschreiben lässt [11]. Er lehnt sich hierbei an die Erkenntnisse der Sprachverarbeitung an, in der bekannt ist, dass sich aus etwa 55 Phonemen praktisch alle Worte aller Sprachen der Erde zusammensetzen lassen. Für die Bilderkennung verwendet Biederman nun anstelle einer Menge von Phonemen eine Menge von Volumenprimitiven, die *geons*, die in geeigneter Art und Weise zu den verschiedensten komplexen Objekten zusammengesetzt werden können. Für eine Objektbeschreibung sind also die verwendeten *geons* und im besonderen auch die zwischen ihnen geltenden geometrischen Relationen von Bedeutung.

Die Erkennung der *geons* führt er auf die Erkennung von fünf Eigenschaften von Kanten in einem zweidimensionalen Bild zurück: die Krümmung einer Kante, die Kolinearität von Kanten, Symmetrie, die Parallelität zweier oder mehrerer Kanten und die gemeinsame Terminierung von Kanten in einem Punkt. Diese Eigenschaften betrachtet

Biederman als nicht-zufällige Eigenschaften von Kanten, die somit Rückschlüsse auf das Vorhandensein eines Objektes erlauben. Da die Detektion der zuvor genannten Eigenschaften blickpunktunabhängig ist, kann eine Objekterkennung ebenfalls unabhängig vom Blickpunkt und sogar aus völlig neuen Blickrichtungen erfolgen.

Biederman unterstützt seine RBC-Theorie durch eine Reihe psychophysikalischer Experimente. Bei diesen wurden den Versuchspersonen zum einen Objektbilder (Strichzeichnungen) vorgelegt, in denen einzelne Komponenten der Objekte fehlten. Im Sinne seiner Theorie fehlten hier also einzelne *geons*, aus denen ein Objekt zusammengesetzt ist. In anderen Bildern waren die Konturen gestört. Dabei unterscheidet Biederman zwischen Störungen, die eine Rekonstruktion der Objektkomponenten unter Anwendung der oben genannten nicht-zufälligen Kanteneigenschaften erlauben, und solchen bei denen dies nicht möglich ist, da hierbei Eckpunkte der Konturen verfälscht wurden und das Überbrücken von Lücken zu falschen Konturverläufen führt. Er spricht daher von *recoverable* und *nonrecoverable degradation*. Einige der von Biederman verwendeten Bilder sind in Abbildung 3.2 zusammengefasst.

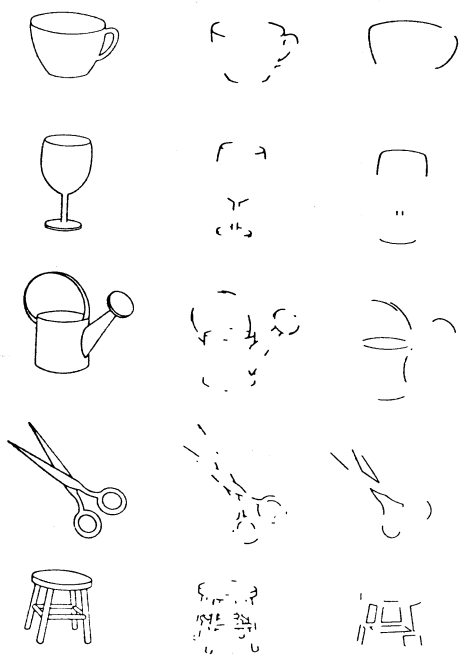


Abb. 3.2: Beispiele der von Biederman verwendeten Bilder (aus [11], S. 57). Die mittlere Spalte zeigt die trotz Fehler rekonstruierbaren Bilder, die rechte Spalte die nicht rekonstruierbaren Bilder.

Biedermans Untersuchungen ergaben, dass eine Objekterkennung praktisch unmöglich ist, wenn die Störungen derart angelegt sind, dass eine Rekonstruktion der Komponenten eines Objektes nicht möglich ist. Wenn jedoch die Komponenten detektiert werden können, so steigt die Erkennungsrate bei ausreichend langer Möglichkeit zur Betrachtung der Bilder signifikant an. Komplexe Objekte hingegen können mit hoher Genauigkeit in relativ kurzer Zeit erkannt werden, wenn einige wenige Komponenten, Biederman spricht von drei oder vier, ohne Störungen dargestellt sind und die übrigen Komponenten vollständig fehlen. Dies bestätigt seine Theorie, dass eine Objekterkennung und daher auch die Objektmodellierung auf der Basis der Komponenten, die für ihn Volumenprimitive sind, erfolgt.

Von besonderer Bedeutung ist für Biederman die Eigenschaft einer solchen strukturbasierten Modellierung, Objekte, die im Bild partiell verdeckt sind, schnell und sicher erkennen zu können. Diese Eigenschaft ist, wie im weiteren Verlauf dieser Arbeit noch deutlich zu erkennen sein wird, einer der wesentlichsten Vorteile strukturbasierter Modelle gegenüber ansichtenbasierter Repräsentationen.

Generalisierte Zylinder

Entscheidet man sich nun für eine strukturbasierte Repräsentation der Objekte, so stellen sich einige Detailfragen, die noch genauer analysiert werden müssen. Zunächst einmal muss geklärt werden, welche Primitive für die Modellierung verwendet werden sollen. Kriegman favorisiert in [96] algebraische Oberflächenbeschreibungen (*algebraic surfaces*) und begründet dies mit der Ausdrucksstärke dieser Repräsentation und ihrer Adäquatheit für das Erkennungsproblem auch bei Auftreten von gekrümmten Oberflächen.

Relativ einfache Beschreibungselemente wie Polyeder oder *quadric surfaces* oder auch komplexere Elemente wie Superquadriken oder Hyperquadriken ([4], [70], [97], [189]) haben zwar in vielen Erkennungssystemen Einsatz gefunden, ihnen fehlt jedoch die Ausdrucksfähigkeit auch komplexe Objekte mit gekrümmten Oberflächen zu beschreiben. Erhöht man nun die Anzahl der Parameter für die Beschreibung, um eine genauere Anpassung des Modells an das zu modellierende Objekt zu erhalten, so ist dies zwar mit der gewünschten

Genauigkeit erreichbar, führt jedoch häufig zu unstabilen Parameterkonfigurationen und kann daher nicht sinnvoll für die Objekterkennung eingesetzt werden. Kriegman argumentiert weiterhin für algebraische Oberflächenbeschreibungen, da diese auch in der Lage sind, viele Formen der generalisierten Zylinder [14], wie z.B. die SHGCs (*straight homogenous generalized cylinders* [167]) zu modellieren.

Die generalisierten Zylinder (GCs) waren erstmalig von Binford definiert worden [14]. Er gilt daher auch als einer der Mitbegründer der objektzentrierten Modellierung. Er verwendet diese GCs in seinen aktuellen Arbeiten, wie z.B. dem *Successor* System [13] und argumentiert dabei sehr entschieden für die Verwendung von Volumenprimitiven anstelle von Oberflächenbeschreibungen. Seiner Ansicht nach stellen Oberflächenelemente kein intuitives Hilfsmittel für die Beschreibungen von Objekten dar. Da ein Objekt in den seltensten Fällen durch ein einzelnes Beschreibungselement modelliert werden kann, benötigt man einen Mechanismus, der eine Zerlegung des Objektes in seine Bestandteile ermöglicht, so dass diese direkt auf die Beschreibungselemente abgebildet werden können. Oberflächen leisten dies im Gegensatz zu Volumenelementen nicht, obwohl ein Volumenelement durch seine Oberfläche beschrieben werden kann. Jedoch beinhaltet eine Oberflächenbeschreibung keine Information über das Innere des Objektes, was in einer Volumenbeschreibung enthalten ist. Letztere trägt also einen höheren Informationsgehalt. Die von ihm vorgeschlagenen generalisierten Zylinder beinhalten außerdem implizite Information über die Relationen zwischen den Oberflächen, z.B. in Form des Durchmessers. Er erläutert seine Überlegungen anhand bekannter Probleme aus der 2D-Erkennung, indem er eine flächenbasierte Modellierung mit einer Umrissmodellierung auf der Basis der umrandenden Kantenstrukturen vergleicht. Er verdeutlicht dabei, dass eine Teilstruktur eines Objektes im Falle einer Umrissmodellierung durch sehr weit von einander entfernte Kantenelemente, Eckenelemente oder dergleichen beschrieben werden (Abb. 3.3).

Diese Überlegungen überträgt Binford auf die Unterscheidung einer volumen- und oberflächenbezogenen Modellierung. Es stellt sich nun die Frage nach geeigneten Kriterien für eine Objektzerlegung. Binford definiert hierzu Teile über ihre Funktion, die er an die Möglichkeiten zur Bewegung, an Elemente der Kinematik knüpft. Somit kommt er

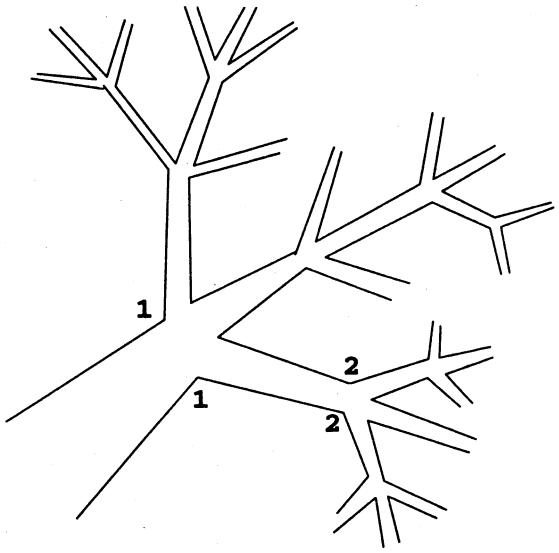


Abb. 3.3: Umrisselemente kennzeichnen die Teile eines Objektes nur unzureichend (aus [13], S. 211)

automatisch zu einer Funktionsfestlegung und somit zu einer Definition des „Teil"-Begriffes anhand einer Formveränderung, die er auf die Kontinuität eines Volumenelementes bzw. eine Veränderung der Hauptachsenrichtung zurückführt.

> „The power of GC representation is that it recognizes and exploits coninuity, i.e. smoothness, thus defining natural parts that lead to an effective part-whole description. Complex objects are decomposed into structures of parts. Complex parts are represented by complex cross sections with simple sweep funktions. (aus [13], Seite 212).

Die Zerlegung eines Objektes in seine Bestandteile ist für Binford die Hauptproblemstellung, die bei der Objektmodellierung zu lösen ist. Dass dies bei der Verwendung seiner generalisierten Zylinder intuitiv geschieht, macht sie für ihn so wichtig für den Aufbau eines Erkennungssystems. An zweiter Stelle erst steht dann für ihn die Frage nach der spezifischen Verwendung und der genauen Zuordnung eines speziellen Typs zu einem Teilobjekt.

Im seinem *Successor*-System werden Objekte durch die Verbindung mehrerer GCs geformt. Dabei müssen für die Erkennung vier Probleme gelöst werden:

1. es muss eine den Erkennungsprozess unterstützende Segmentierung des Bildes gefunden werden;
2. die Objekte müssen vom Hintergrund separiert werden;
3. aus den Bilddaten müssen 3D Objektbeschreibungen erzeugt werden;
4. aus den 3D Objektbeschreibungen müssen Objekthypothesen generiert werden.

Dies wird durch Abbildung 3.4 veranschaulicht. Für die Segmentierung verwendet Binford hierbei einen speziellen Kantendetektions- und Linking-Mechanismus, um in komplex strukturierten Szenen homogene Oberflächen zu finden. Durch eine räumliche Auswertung gekrümmter Kantenstrukturen werden Teile von generalisierten Zylindern detektiert und zur Objekt-Hintergrund Trennung herangezogen. Dabei ist die erzielte Trennung von Objekt und Hintergrund abhängig von der gewählten Definition der Objekte, Objektklassen und der Teile eines Objektes. Die Interpretation der Szene wird dann anschließend durch die Auswertung eines Baysschen Netzes erzielt. Dabei ist jedem Objektteil genau ein Knoten im Baysschen Netz zugeordnet, um eine automatische Umsetzung von Objektmodell in die Netzstruktur zu gewährleisten.

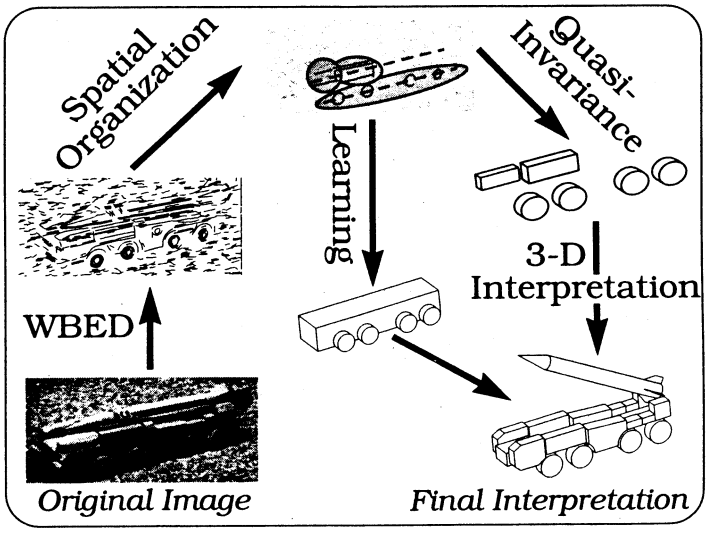


Abb. 3.4: Das *Successor*-System (aus [13], S.215)

Binford macht in seiner Arbeit keine Aussagen darüber, wieviele GCs benötigt werden, um eine gewünschte Modellierungsgenauigkeit

zu erreichen. Auch existieren keine Aussagen darüber, wie gut eine Generalisierung von Objekten und die hiermit verbundene Erzeugung von Objektklassen erfolgen kann. Er sieht die Vorteile einer objektzentrierten Modellierung im expliziten Aufbau von Beziehungen zwischen Objekten, ihren Bestandteilen und einem Klassenbegriff. Diese Möglichkeit spricht er beobachterzentrierten Ansätzen ab und verweist darauf, dass keinerlei Beziehungen zwischen den Ansichten eines Objektes existieren. Das dem nicht so ist, wird im Laufe dieser Arbeit noch aufgezeigt werden (Kap. 6).

3.3 Ansichtenbasierte, beobachterzentrierte Ansätze

Da verschiedene Probleme der objektzentrierten Erkennung, wie z.B. die Problematik des Auffindens eines dreidimensionalen Volumenprimitives in einem zweidimensionalen Bild, bislang noch nicht hinreichend gut gelöst werden konnte und da eine Reihe von psychophysikalischen Untersuchungen ergeben haben, dass die Erkennungsleistung blickpunktabhängig ist, haben in jüngerer Zeit die ansichtenbasierten Ansätze ein größeres Gewicht erlangt. Bei diesen wird darauf verzichtet, ein explizites Objektmodell aufzubauen. Information über ein Objekt wird vielmehr durch eine Menge charakteristischer 2D-Ansichten des Objektes gespeichert. Auf diese Weise entsteht eine beobachterzentrierte Objektrepräsentation. Vorteil dieses Ansatzes ist, dass die gelernten 2D-Ansichten direkt mit der präsentierten 2D-Objektansicht verglichen werden können. Wichtig ist aber, dass eine geeignete Menge von charakteristischen Ansichten ausgewählt wird. Da unser Ansatz ebenfalls in diese Klasse fällt, sollen hier einige wichtige Arbeiten genauer vorgestellt werden.

Psychophysikalische Untersuchungen

In den vergangenen Jahren konnte in einer ganzen Reihe von psychophysikalischen Untersuchungen gezeigt werden, dass die Erken-

nung eines zuvor gelernten Objektes abhängig ist vom Blickpunkt, von dem das Objekt betrachtet wird, bzw. von der damit korrespondierenden Objektansicht. So konnte z.B. Tarr zeigen, dass die Erkennungszeiten von der Winkeldifferenz zwischen gelernter und präsentierter Objektansicht abhängen [170]. Dies geht einher mit den Erkenntnissen Shepards, die besagen, dass ein inkrementell verlaufender Normierungsprozess stattfindet, in dem präsentierte und gelernte Ansichten in einander überführt werden [168]. Somit ergibt sich also ein unmittelbarer Zusammenhang zwischen Winkeldifferenz und Erkennungsleistung bzw. Erkennungszeiten.

Edelman ergänzt diese Aussagen noch um die Ergebnisse weiterer Versuchsreihen, in denen er den Versuchspersonen in einer Trainingsphase künstliche Objekte präsentiert. Dabei werden alle Ansichten mit der gleichen Häufigkeit angeboten. Anschließend werden dann einzelne der zuvor präsentierten Ansichten zur Wiedererkennung präsentiert. Es zeigt sich, dass dabei die Erkennungsgeschwindigkeit nicht konstant ist, obwohl alle Ansichten gleich häufig präsentiert worden sind. Es formen sich hingegen einzelne besonders repräsentative Ansichten heraus, die besonders schnell wiedererkannt werden. Eine gleichförmigere Verteilung der gemessenen Erkennungszeiten ergibt sich jedoch, wenn weitere Trainingssequenzen präsentiert werden. Dies wird darauf zurückgeführt, dass die zeitabhängige Normierungsphase entfällt, wenn ein Objekt dem Betrachter hinreichend gut bekannt ist und er quasi alle auftretenden Ansichten des Objektes gelernt hat. Dies bedeutet, dass also die gelernten charakteristischen Ansichten eines Objektes auf der Ansichtensphäre dicht verteilt sind. Dies erklärt auch, warum die Erkennung alltäglicher Objekte für den Menschen scheinbar mühelos und unabhängig vom Blickwinkel geschieht. Ähnliche Ergebnisse werden auch in [25], [55] und [171] beschrieben.

Neben diesen psychophysischen Erkennungsuntersuchungen und -erkenntnissen gibt es zusätzlich noch neurobiologische Untersuchungen, die ebenfalls einen ansichtenbasierten Ansatz unterstützen. So konnten Zellableitungsexperimente durchgeführt werden, die nachwiesen, dass einzelne Zellgruppen z.B. für die Erkennung von Gesichtern zuständig sind [147].

repräsentative Ansicht untypische Ansicht

Abb. 3.5: Die Erkennungsleistung und -geschwindigkeit ist abhängig vom Blickwinkel des Betrachters (in Anlehnung an [171])

Aus technischer Sicht stellen sich nun - wie schon bei der objektzentrierten Repräsentation - Fragen nach der konkreten Realisierbarkeit. Dabei geht es speziell um die Fragen:

- ❒ Wie wird eine einzelne Ansicht beschrieben und erkannt?
- ❒ Welche Zusammenhänge sollen zwischen den verschiedenen Ansichten eines Objektes modelliert werden?

Diese zwei Fragen betreffend unterscheiden sich die verschiedenen Erkennungssysteme sehr deutlich voneinander. So werden z.B. die einzelnen Ansichten eines Objektes zum Teil mit neuronalen Netzen untersucht bzw. erkannt, aber auch mit Hilfe struktureller 2D-Modelle beschrieben. Die Übergänge von einer Ansicht zu einer benachbarten werden teils implizit in neuronalen Strukturen verarbeitet, teils aber auch explizit in Graphen repräsentiert.

Aspektgraphen

Man spricht dabei von sogenannten *Aspektgraphen*, wenn alle Ansichten eines Objektes repräsentiert werden, die sich in der Topologie der sichtbaren Oberflächenstrukturen unterscheiden. Dabei werden dann die Ansichten oder Aspekte miteinander in Relation gesetzt, die durch Variation des Blickpunktes ineinander überführt werden können. Solche Aspektgraphen wurden in ihrer ursprünglichen Form von Koen-

derink und van Doorn eingeführt und in verschiedenen Arbeiten weiterentwickelt und eingesetzt [91].

Shapiro verwendet in diesem Zusammenhang z.B den Begriff der *view classes* eines Objektes. In einer Klasse werden dabei alle Ansichten eines Objektes eingeordnet, in denen die selben oder zumindest ähnliche Merkmale sichtbar sind. Jede Klasse wird dann als ein Graph repräsentiert. Als Merkmale, die in den Knoten des Graphen repräsentiert werden, dienen dabei Segmente von Linien oder Bildkanten, deren Topologie durch drei verschiedene Typen von Graphkanten charakterisiert wird. Einander im Objekt gegenüberliegende Segmente werden durch *opposite*-Kanten verbunden. Segmente, die sich überlappen, d.h. sie haben einzelne gerade Linien- oder Bildkantenstücke gemeinsam, werden durch eine *strong adjacency*-Kante verknüpft, während Segmente, die nah beieinanderliegende Endpunkte besitzen durch *weak adjacency*-Kanten in Relation gesetzt werden [162]. Abbildung 3.6 zeigt einige Ansichten eines Objektes und die resultierenden Graphen.

Gremban und Ikeuchi beschreiben in [68] die Anordnung der verschiedenen Ansichten oder Aspekte eines Objektes in einem *Klassifikationsbaum* (*classification tree*). In einem solchen Baum beschreibt jeder Knoten eine Menge möglicher Ansichten und eine Operation zum Testen auf das Vorhandensein dieser Ansichten. Die Kante zwischen zwei Knoten *a* und *b* repräsentiert einen Test auf der Menge der durch *a* repräsentierten Ansichten. Der Knoten *b* repräsentiert dann nur noch die Ansichten aus *a*, die diesem Test positiv entsprechen. In den Blättern des Baumes ist dann nur noch eine einzelne Ansicht repräsentiert - das Ergebnis des Klassifikationsprozesses. Das Suchen nach einer möglichst optimalen, d.h. kostengünstigen Klassifikationsstrategie, in der festgelegt ist, wann welche Kante weiterverfolgt wird, beschreiben Gremban und Ikeuchi als Suchprozess in einem *Strategiebaum* (*strategy tree*). In einem *Strategiebaum* repräsentiert jeder Pfad von der Wurzel zu einem Blatt einen anderen *Klassifikationsbaum*. In ihrem Beitrag weisen Gremban und Ikeuchi bereits auf das Problem hin, dass in einigen Fällen am Ende eines Klassifikationsprozesses nicht ein einzelner Objektaspekt als Resultat erreicht wird, sondern, dass in einigen Blättern auch mehrere Aspekte repräsentiert werden können. Dies ist kein Fehler in den Beschreibungsbäumen, sondern stellt dar, dass die verwendeten Merkmale nicht ausreichen, einige sehr ähnliche Aspekte von

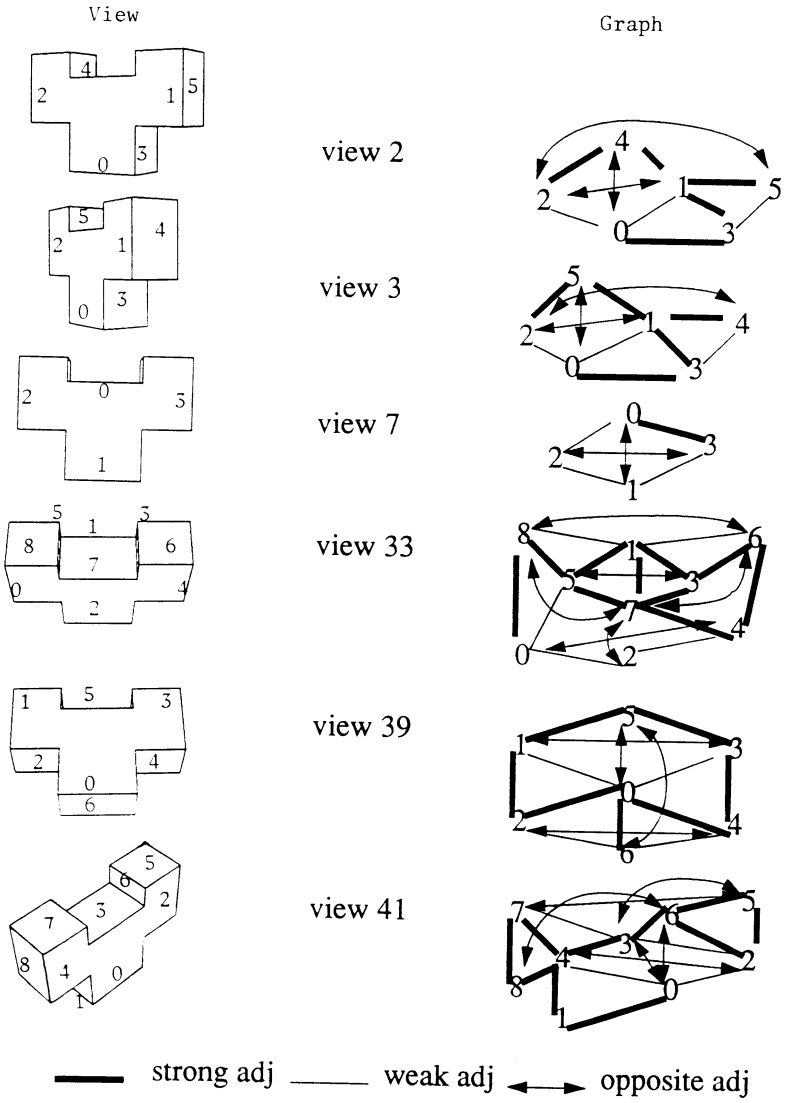


Abb. 3.6: Ansichten eines Objektes und die resultierenden Graphen zur Beschreibung der jeweiligen Ansicht (aus [162], S. 405).

einander zu unterscheiden. Auf diese Problematik wird in Kapitel 9 dieser Arbeit noch gezielt eingegangen. Abbildung 3.7 zeigt am Beispiel eines CAD Objektmodells die verschiedenen Ansichten dieses Objektes sowie einen *Klassifikationsbaum*.

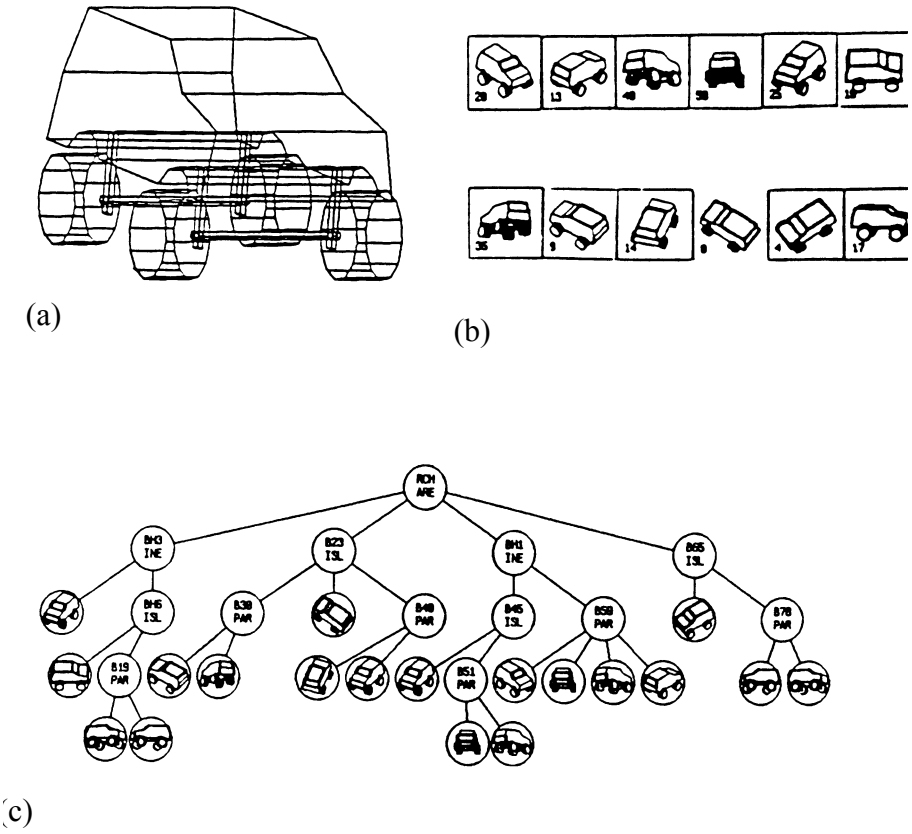


Abb. 3.7: (a) CAD Objektmodell, (b) Objektansichten und (c) Klassifikationsbaum (aus [68], S. 244)

Aspekthierarchien

Zu den Modifikationen des originalen Aspekt-Ansatzes gehört auch die Vorgehensweise von Dickinson, Pentland und Rosenfeld [49]. Sie verwenden zusätzlich für die Objektmodellierung eine Aspekthierarchie, in der die verschiedenen Ansichten auf unterschiedlichen Abstraktionsniveaus beschrieben werden. Da sie hierbei auch Volumenprimitive für die Modellierung einsetzen, kommt es dabei zu einer Vermischung von struktur- und ansichtenbasierter Repräsentation.

Auf der obersten Ebene ihrer Aspekthierarchie verwenden sie wie üblich die Aspekte einer 3D Struktur. Diese setzen sich aus den einzelnen Oberflächenelementen der Volumenprimitive zusammen. Diese Oberflächenelemente werden auch als *Faces* bezeichnet. Wenn ein einzelner Aspekt eindeutig identifiziert werden kann, so ergibt sich hieraus

die Möglichkeit des Rückschlusses auf das zugrundeliegende Volumenprimitiv oder aber zumindest ein Rückschluss auf eine Menge von Primitiven. Zu diesem Zweck werden die Primitive mit den sie repräsentierenden Aspekten durch eine spezielle Verzeigerung verknüpft.

Um dem Problem der Verdeckung einzelner Elemente eines Aspekts entgegenzutreten, werden auf der zweiten Ebene der Aspekt-Hierarchie die verschiedenen *Faces* repräsentiert und mit den Aspekten verknüpft, zu denen sie einen Beitrag leisten. Auf diese Weise kann bei Verdeckungen, die eine Erkennung eines Aspektes verhindern, auf diese zweite Ebene herabgestiegen werden und es kann die Formation der restlichen sichtbaren *Faces* analysiert werden.

Um auch bei der Erkennung einzelner *Faces,* die anhand ihres Konturverlaufes identifiziert werden, auf Verdeckungen desselben reagieren zu können, werden dann auf der dritten Ebene diese Konturverläufe durch einzelne Konturgruppen wie gekrümmten Kanten, parallelen Kantengruppen, etc. repräsentiert.

Diese drei Ebenen der Aspektmodellierung werden in Abbildung 3.8 illustriert.

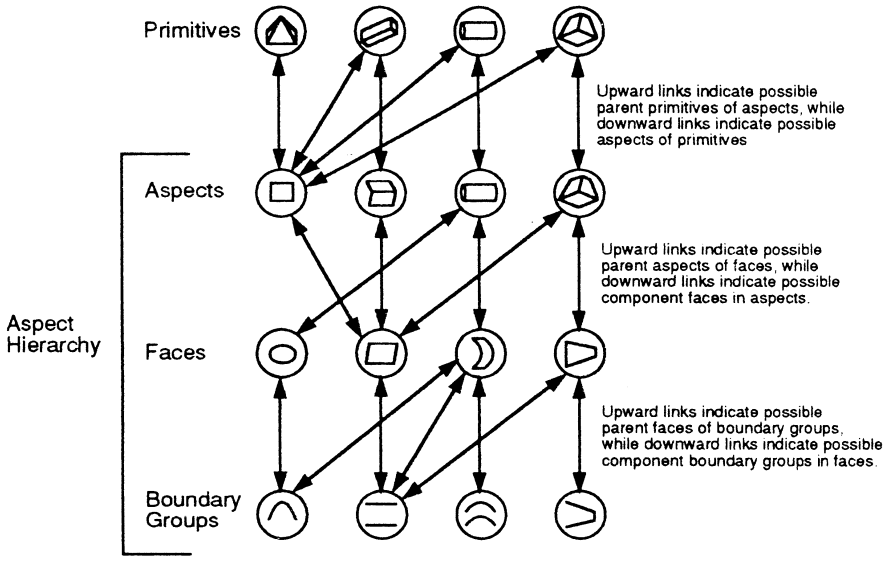


Abb. 3.8: Aspekthierarchie zur Modellierung von Volumenprimitiven (aus [49], S. 135)

Da natürlich üblicherweise einzelne Kantengruppen der untersten Modellierungsebene Bestandteil mehrerer *Faces* sind und gleichzeitig

einzelne Faces Bestandteil mehrerer Aspekte sind, die wiederum zugleich Ansichten mehrerer Volumenprimitive darstellen können, verwenden Dickinson, Pentland und Rosenfeld in ihrer Arbeit eine Matrixrepräsentation zur Festlegung der bedingten Wahrscheinlichkeiten der zuvor aufgeführten Bestandteilrelationen.

Durch die verwendete objektzentrierte Dekomposition von komplexen 3D-Objekten in Volumenprimitive in Verbindung mit einer beobachterzentrierten Modellierung dieser Volumenprimitive mittels einer Aspekthierarchie führen Dickinson, Pentland und Rosenfeld in ihrer Arbeit zwei eigentlich konträre Vorgehensweisen zusammen. Dabei sehen sie in der gewählten Dekomposition, die auf die von Biederman definierten *geons* zurückgreift, eine intuitiv verständliche Modellwahl. Die Volumenprimitive erlauben effektive Indexing-Verfahren beim Erstellen von Objekthypothesen, da sie deutlich diskriminativer sind als einfache Kantenstrukturen. Auf der anderen Seite ermöglicht die aspektbasierte Modellierung der Volumenprimitive eine verbesserte Extraktion derselben aus dem Bildmaterial. Trotzdem bleibt jedoch die Problematik bestehen, die Aspekte eines Primitives anhand einfachster Bildstrukturen, den Kantengruppen, zu erkennen. Der hierzu notwendige Gruppierungsaufwand ist auch nach Aussage von Dickinson enorm. Um hier zu einer Lösung zu gelangen, setzen die Autoren eine Heuristik auf der Basis einer statistischen Analyse der bedingten Wahrscheinlichkeiten zwischen den verschiedenen Elementen der Aspekthierarchie ein.

Eigenräume

Einen völlig anderen Weg der beobachterzentrierten Objekterkennung beschreiten Murase und Nayar in ihren Arbeiten [131], [132], [133]. Sie verzichten gänzlich auf eine Bildsegmentation in Form einer Regionen- oder Kantensegmentation, sondern setzen direkt auf den Grau- oder Farbwerten der Bildpixel auf. Sie charakterisieren ihren Ansatz daher auch als *appearance matching*, da hierbei nicht nur die Form eines Objektes ausschlaggebend ist, sondern z.B. auch seine Reflektionseigenschaften. Um nun ein Objekt trotz variierender Beleuchtung und Objektposition erkennen zu können, ohne dabei eine große Bildmenge für jedes Objekt auswerten zu müssen, bestimmen sie den *Ei-*

genraum der möglichen Erscheinungen eines Objektes. Dazu werden mit Hilfe der Karhunen-Loeve-Transformation die Eigenvektoren einer Bildmenge bestimmt [61]. Diese Eigenvektoren bilden eine orthogonale Basis zur Repräsentation einzelner Bilder der Bildmenge. Wenn es für eine konkrete Anwendung wie die Objekterkennung nun ausreicht, eine grobe Beschreibung der Objekte zu repräsentieren, so benötigt man hierzu im allgemeinen nur sehr wenige Eigenvektoren. Hiermit können dann die prinzipiellen, signifikanten Erscheinungsmerkmale festgehalten werden.

Für die Objekterkennung ist eine Repräsentation mit Hilfe von Eigenvektoren eine sehr komfortable Möglichkeit, die Ähnlichkeit von Bildern festzustellen, ohne direkt im hochdimensionalen Bildraum zu arbeiten. Werden nämlich zwei Bilder auf den Eigenraum projiziert, so stellt der Abstand zwischen den entstehenden Projektionspunkten ein Maß für die Ähnlichkeit der zugrunde liegenden Bilder dar. In der Bilderkennung wird die Karhunen-Loeve-Transformation daher in einer Reihe von Arbeiten verwendet. Hierzu gehört zum Beispiel die Handschrifterkennung [130], [165], [179] oder auch die Erkennung von Gesichtern [125]. Bei diesen Applikationen handelt es sich jedoch um reine Mustererkennungsaufgaben im zweidimensionalen Bereich.

Murase und Nayar haben für die Modellierung und Erkennung komplexer dreidimensionaler Objekte einige Erweiterungen hinzugefügt. Es ist ihr Ziel, eine kompakte Repräsentation der Objekte in Abhängigkeit von den Parametern der Objektlage und der Beleuchtung zu finden. Dabei setzen sie auf der Idee des Eigenraumes auf und benennen ihren eigenen Ansatz als *parametrisierten Eigenraum*. Dazu nehmen sie eine Menge von Bildern eines oder mehrerer Objekte bei variierender Objektlage und Beleuchtung, die zunächst größen- und helligkeitsnormiert werden. Somit sind Objektlage und Beleuchtungsrichtung die freien Parameter der Erscheinung der Objekte in den verschiedenen Bildern. Für die resultierende Bildmenge wird nun der zugehörige Eigenraum berechnet, in dem die zu den zehn größten Eigenwerten gehörenden Eigenvektoren verwendet werden. Im nächsten Schritt werden dann alle Objektbilder auf den Eigenraum projiziert. Somit wird jedes der Objektbilder durch einen einzelnen Punkt repräsentiert, der auf einer Mannigfaltigkeit liegt, die durch Objektlage und Beleuchtung parametrisiert wird.

Hierbei verwenden Murase und Nayar zwei Mannigfaltigkeiten in zwei verschiedenen Eigenräumen. Die erste wird durch alle Bilder aller zu lernenden Objekte gebildet und wird als *universal eigenspace* bezeichnet, die zweite wird durch alle Bilder eines einzelnen Objektes gebildet und wird als *object eigenspace* bezeichnet. Somit arbeiten sie mit einem Eigenraum für alle Objekte, der dazu dient, die Objekte von einander zu unterscheiden, da er die wesentlichen Eigenschaften des Aussehens aller Objekte repräsentiert. Im *object eigenspace*, der zu jedem Objekt existiert, kann dann eine genauere Analyse der freien Parameter Objektlage und Beleuchtung durchgeführt werden. Sowohl für die Erkennung als auch für die Bestimmung der freien Parameter wird daher nach der entsprechenden Vorverarbeitung eine Projektion des präsentierten Bildes auf den jeweiligen Eigenraum durchgeführt. Zu dem entstehenden Projektionspunkt wird nun derjenige Punkt auf der Mannigfaltigkeit mit dem geringsten Abstand gesucht. Da bei dieser Vorgehensweise lediglich zehn Eigenvektoren und alle Projektionspunkte in einem zehndimensionalen Raum der zu lernenden Bilder gespeichert werden müssen, nicht jedoch die zu lernenden Bilder, erzielt man eine enorme Kompressionsrate. Desweiteren reduziert sich die Korrelation zur Ähnlichkeitsberechnung auf eine Abstandsbestimmung zu den Punkten auf der Mannigfaltigkeit, so dass zusätzlich auch eine enorme Laufzeitreduzierung erzielt wird. Abbildung 3.9 zeigt die Projektionspunkte der Objektbilder in einem dreidimensionalen Objekt-Eigenraum. Da in diesem Beispiel zur Vereinfachung nur ein freier Rotationsparameter verwendet wurde, reduziert sich die Mannigfaltigkeit zu einer Kurve anstelle einer Oberfläche.

Murase und Nayar erzielen bei einem Testsatz von zwanzig Objekten sehr gute Erkennungsergebnisse mit einer fehlerfreien Erkennung und einer Objektlageschätzung mit einem Schätzfehler kleiner 1.6 Grad. Hierbei wurde nur ein Rotationswinkel bestimmt, da die Objekte in einer festen Lage auf einem Drehteller dem System präsentiert wurden und somit nur ein freier Lageparameter bestimmt werden musste.

Trotz dieser sehr guten Ergebnisse bleibt ein wesentlicher Nachteil der gewählten Vorgehensweise. Damit ein zufällig variierender Hintergrund keinen Einfluss auf die Objekterkennung nimmt, ist es notwendig, ihn aus dem Bildmaterial im Lernprozess und auch im Erkennungsvorgang zu eliminieren. Somit hängt die Qualität der Erkennung

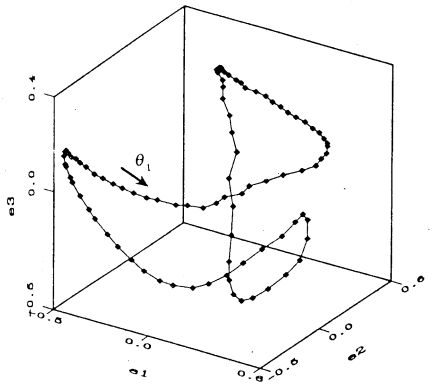


Abb. 3.9: Manigfaltigkeit und Projektionspunkte der Objektbilder in einem dreidimensionalen Objekt-Eigenraum (aus [131], S. 11)

unmittelbar von der Güte der Segmentation ab, bei der Objekt und Hintergrund getrennt werden müssen. Da bei diesem Verfahren auf eine Dekomposition des Objektes vollständig verzichtet wird und im Prinzip eine Korrelation zwischen präsentiertem und gelernten Bildern durchgeführt wird, dürfen darüber hinaus auch keine Verdeckungen der Objekte auftreten. In einem solchen Fall kann keine ausreichende Ähnlichkeit zu einem der gelernten Bilder festgestellt werden.

Elastische Netze

Ein weiterer Ansatz zur beobachterzentrierten Objektrepräsentation wird von v.d. Malsburg verfolgt [92], [93], [181], [182]. Hierbei geht es zunächst einmal um die Erkennung einzelner Ansichten eines Objektes, das nur geringen Blickwinkelveränderungen und perspektivischen Verzerrungen unterliegen darf. v.d. Malsburg geht dabei ebenfalls von einer ansichtenbasierten Objektrepräsentation aus und verwendet dazu in seinem *Dynamic Link Matching* System (DLM) ein Netzwerk lokaler Bildmerkmale. In diesem Netzwerk bilden dynamische Links einen Zusammenhalt zwischen den lokalen Merkmalen und repräsentieren dabei die relativen Positionen innerhalb des Objektmodells. Das Wiedererkennen eines Objektes könnte nun zunächst einmal durch ein Graph-Matching geschehen, bei dem der Modellgraph als Teilgraph im Bild gesucht wird. v.d. Malsburg geht mit seinem System jedoch einen Schritt weiter und hat daher in seinem System dynamische Kanten ein-

geführt, die miteinander kooperieren können. Daher beschreiben sie, obwohl sie selbst nur eine lokale Eigenschaft darstellen, lokale Einschränkungen für mögliche globale Transformationen der Objekte.

In einem neuronal beschriebenen Prozess besitzen dabei benachbarte Links, die zueinander passende Einschränkungen beschreiben und miteinander kooperieren, eine inhibitorische Wirkung auf andere Links. Somit bildet sich in einem parallel verlaufenden Auswahlprozess entsprechend den lokalen Einschränkungen im Modell eine Anpassung des Modellnetzes an das vorliegende Bildmaterial heraus, die eine invariante Erkennung gegenüber globalen Transformationen wie Skalierung, Translation oder auch Punkt- und Achsenspiegelungen erlaubt.

Eine grundlegende Idee des DLM-Systems ist die Topographie-Einschränkung, bei der davon ausgegangen wird, dass egal welche Transformation benötigt wird, um ein lokales Merkmal im Modell mit seinem Gegenstück im Bild zu matchen, dieselbe Transformation mit großer Wahrscheinlichkeit auch in der Nachbarschaft dieses Merkmals verwendet werden muss. Betrachtet man nun zwei Muster (gelernt und präsentiert), die miteinander verglichen werden sollen, als zwei 2D-Lagen X und Y eines neuronalen Systems, so ist es die Aufgabe des Systems, in einem Selbstorganisationsprozess eine Menge korrespondierender Zellen $a \in X$ und $b \in Y$ zu finden. Jede dieser Zellen codiert dabei ein lokales Merkmal f und eine Aktivität x, die an andere Zellen übertragen werden kann. Die Korrespondenz zwischen den Zellen a und b wird durch die Stärke des dynamischen Links J_{ba} beschrieben. Da mehrere Verbindungen von Zellen aus X zur Zelle b führen können, kann $\sum_a J_{ba}$ bei geeigneter Normierung als Wahrscheinlichkeit für das Auftreten des entsprechenden lokalen Merkmals an der Position der Zelle b betrachtet werden.

Der Selbstorganisationsprozess wird durch Lösen der Differentialgleichungen

$$\dot{x}_a = -\alpha x_a + (k \bullet X)_a + I_a^{(x)}, \quad X_a = \sigma(x_a) \tag{3.1}$$

$$\dot{y}_b = -\alpha y_b + (k \bullet Y)_b + I_b^{(y)}, \quad Y_b = \sigma(y_b) \tag{3.2}$$

beschrieben, wobei σ() eine sigmoide Ausgabefunktion definiert. *k* beschreibt einen statischen Interaktionskern zwischen den beiden Layern. Als Eingabe I_a für den Layer *X* dient ein sich langsam veränderndes Rauschen, während I_b sich ergibt als $I^{(y)}{}_b = \varepsilon_1 \sum_a J_{ba} T_{ba} X_a$ mit einem Verbindungsfaktor ε_1 und einem Ähnlichkeitsmaß T_{ba} für die lokalen Merkmale, die den Zellen *a* und *b* zugeordnet sind. Eine detaillierte Beschreibung dieses Selbstorganisationsprozesses und eine schnelle Implementierung hierfür ist in [93] zu finden.

Als Entscheidungskriterium, ob zwischen den beiden verarbeiteten Mustern eine globale Transformation existiert oder nicht, dient ein Korrelationsmaß zwischen Paaren von Zellen *a* und *b*.

$$C_{ba} = \frac{\langle Y_b X_a \rangle - \langle Y_b \rangle \langle X_a \rangle}{\Delta Y_b \Delta X_a} \quad \text{mit} \quad \Delta A = \sqrt{\langle A^2 \rangle - \langle A \rangle^2} \tag{3.3}$$

Dabei beschreibt der Operator ⟨ ⟩ eine Mittelwertbildung über die Iterationsschritte im Selbstorganisationsprozess. Die Anzahl aller Paare (*a, b*) mit einem Korrelationswert $C_{ba} > 0.9$ wird nun mit einem vorgegebenen Schwellwert verglichen und dient somit als Erkennungsmaß in einem unüberwachten Erkennungssystem.

Die ART-Architektur

Eine ganze Familie von neuronalen Erkennungssystemen ist aus der *adaptive resonance theory* (kurz *ART*) von Stephen Grossberg entstanden [72], [73]. Ausgangspunkt seiner Überlegungen war das Problem, dass beim Lernen neuer Muster mit Hilfe eines neuronalen Netzes die Gefahr besteht, dass alte bereits gelernte Muster wieder verlernt werden. Diese Problematik, die auch als das *Stabilitäts-Plastizitäts Dilemma* bezeichnet wird, versucht Grossberg dadurch zu beheben, dass ein Informationsfluss zwischen den neuronalen Schichten seiner Architektur nicht nur in einer Richtung, von der Eingabe-Schicht F_1 zur Ausgabe- oder Kategorisierungsschicht F_2 verläuft. Grossberg führt stattdessen einen zweiten Informationsfluss ein, der von der Ausgabeschicht wieder auf die Eingabeschicht zurückführt. Dieser wird genutzt, um zu überprüfen, ob die durch die Aktivität der Schicht F_2 beschriebene Kategorie wirklich zum Eingabemuster an der Schicht F_1 passt. Zu diesem Zweck, haben die Neurone der Schicht F_2 gelernt, durch welche Ein-

gangssignale sie bislang zur Aktivität angeregt wurden. Bei erneuter Aktivität eines Satzes dieser Neurone, assoziieren sie dadurch eine Erwartung, welches Eingangssignal an der Schicht F_1 vorliegen muss. Dieses wird nun auf die Eingangsschicht zurückprojiziert. Grossberg spricht in diesem Zusammenhang von der *feedback expectation*. Unterscheiden sich diese Erwartung und das vorliegende Eingangssignal, so kommt es zu einer kurzfristigen Unterdrückung des Eingangssignals und ein Lernen der fehlerhaften Kategorisierung des Eingangssignals wird verhindert. Dieses kurzfristige Auslöschen des Eingangssignals führt nun wiederum zu einer Rückmeldung an die Schicht F_2 die somit über die zuvor fehlerhafte Kategorisierung informiert wird. Dabei werden die bei der fehlerhaften Zuordnung aktiv gewesenen F_2-Neurone inhibiert, so dass nun andere alternative Kategorien die Möglichkeit zur Aktivität bekommen. Hierdurch können nun weitere Hypothesen getestet werden oder auch neue Kategorien gebildet werden. Abbildung 3.10 verdeutlicht die Rückkopplung aus der Ausgabe- auf die Eingangsschicht, die über ein Orientierungssubsystem *A* geführt wird.

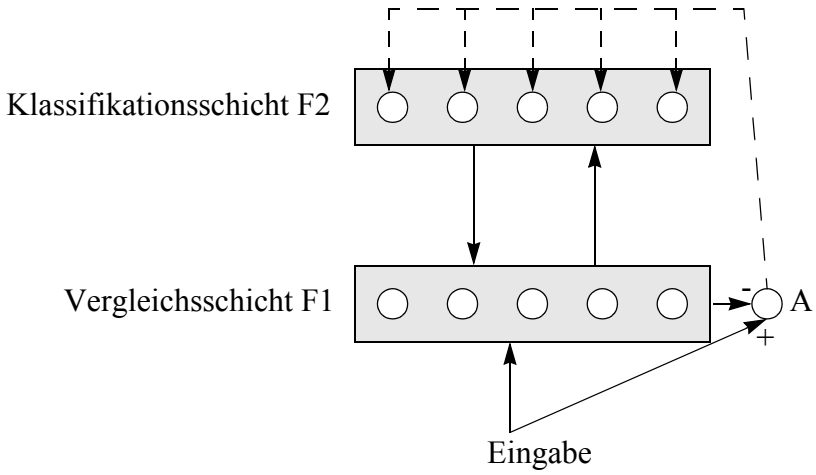


Abb. 3.10: Prinzipieller Aufbau der *ART*-Architektur zur Vermeidung des unkontrollierten Verlernens zuvor antrainierter Objektklassen.

Durch diese Vorgehensweise der Rückprojektion einer Erwartung auf das anliegende Signal, verhindert Grossberg ein fehlerhaftes Überschreiben zuvor gelernter Kategorien.

Eine Vielzahl von Variationen der *ART*-Architektur sind inzwischen entstanden. Aufbauend auf *ART1*, einer Implementierung, die bi-

närwertige Eingangssignale verarbeiten konnte, wurde *ART2* entwikkelt, um auch analoge Eingangsvektoren behandeln zu können [36], [37]. *FuzzyART* verwendet fuzzyfizierte Logikoperatoren [38], während *ART2-A* eine schnell lernende Implementierung der *ART*-Architektur mit fast identischen Leistungsmerkmalen ist [39]. *ARTMAP* verbindet zwei *ART1* Netze [40]. Während dem ersten Netz das Eingangssignal zugeführt wird, wird dem zweiten Netz während einer Trainingsphase der gewünschte Ausgabevektor angelegt. Da *ART* unüberwacht lernt, bildet das erste Netz eine Kategorie für das Eingangssignal. Dieses wird dann mit der Aktivität des zweiten Netzes, dem gewünschten Ausgabewert, assoziiert. Auf diese Weise werden die Vorteile des unüberwachten Lernens mit denen des überwachten Lernens gekoppelt. Ersetzt man eines der *ART* Module durch ein *FuzzyART* Modul so erhält man eine *FuzzyARTMAP* [41].

Hybride Systeme

Innerhalb der Klasse der zuvor vorgestellten ansichtenbasierten Erkennungssysteme können zwei wesentliche Gruppen unterschieden werden:

- dekompositorische Ansätze, wie das System von Dickinson, Pentland und Rosenfeld. Hier werden die zu erkennenden Objekte in verschiedene Strukturprimitive zerlegt, die dann aus dem Bildmaterial extrahiert werden müssen.
- holistische Ansätze, wie die Verfahren von Murase und Nayar oder auch die Arbeiten von v.d. Malsburg. Hierbei wird das Objekt als Ganzes betrachtet und es wird die Ähnlichkeit zum präsentierten Bild bestimmt.

Da beide Vorgehensweisen gewisse Vor- und Nachteile aufweisen, bietet es sich an, beide Paradigmen miteinander zu koppeln, um die Vorteile miteinander zu verbinden und die Nachteile aufzuheben. Hieraus ergeben sich dann hybride Systeme, zu denen auch das in dieser Arbeit vorgestellte Erkennungssystem zu zählen ist. Den Ansatz der hybriden Erkennung verfolgen auch Sagerer und Ritter mit der Kopplung von semantischen Netzen zur dekompositorischen Beschreibung von Objekten und neuronalen Netzen zur ganzheitlichen Erkennung von Objektansichten [98], [100], [126].

Für die holistische Erkennung wird dabei eine Kombination mehrerer neuronaler Netzwerke verwendet, die als VPL-Klassifikator bezeichnet wird. Dieser besteht aus einer **V**ektorquantisierungsstufe, einer lokalen Principle Component Analysis (**P**CA) und einem Netz vom Typ der Local Linear Map (**L**LM) [81]. Dabei dient die Vektorquantisierungseinheit dazu, nach einem *winner takes all* (WTA) Prinzip den für das angebotene Bildmaterial geeigneten Verarbeitungspfad auszuwählen. Dabei steht für jeden Verarbeitungspfad eine eigene PCA und ein eigenes LLM zur Verfügung (siehe auch Abb. 3.11).

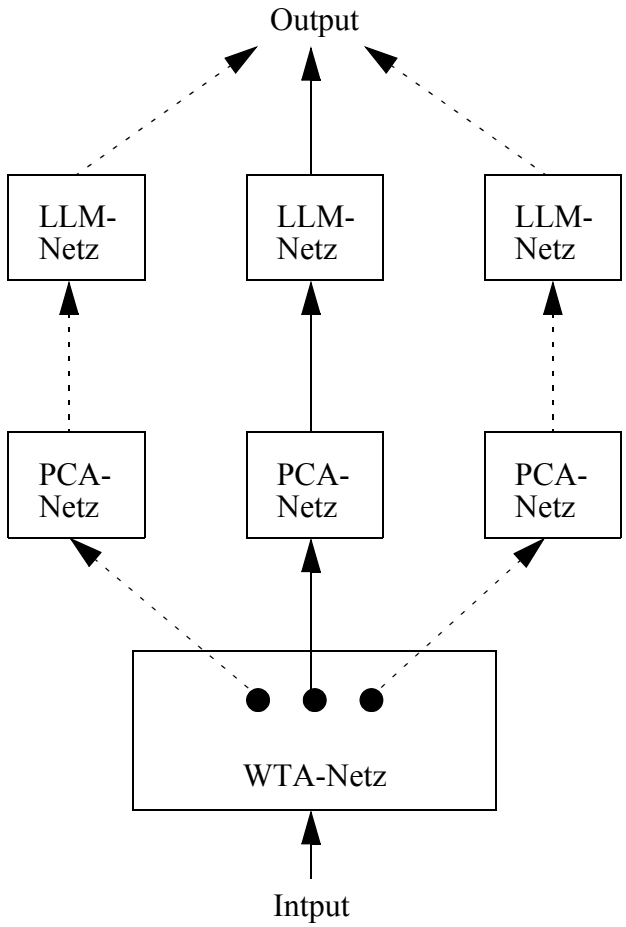


Abb. 3.11: Holistische Erkennung mit dem VPL-Klassifikator (nach [81].

Dieser holistische Erkenner wird nun in einem semantischen Netz verwendet, um die dort modellierten Objektansichten zu erkennen. Als Grundlage für dieses semantische Netz dient das System ERNEST (Erlangener Erkennungssystem) [136]. Dabei erfolgt in dem semantischen Netz eine Dekomposition der Objektansichten in einfache geometri-

sche Grundelemente. Ausgehend von den Erkennungshypothesen des VLP-Klassifizierers auf Objektebene, wird dann das semantische Netz herangezogen, um die Hypothesen zu verifizieren.

Für die Auswertung von Bildfolgen besitzt das System, darüber hinaus ein Modul zur schritthaltenden Verfolgung. Hierbei wird das Erkennungsergebnis zum Zeitpunkt t verwendet, um eine Initialisierung des semantischen Netzes für den Zeipunkt t+1 zu erhalten. Lediglich die Regionen des Bildes, die dem semantischen Netzwerk nicht direkt zugeordnet werden können, bedürfen dabei einer Erkennung durch den holistischen Erkenner [101].

3.4 Physikalische Modellierungsansätze

Die Erweiterung herkömmlicher modellbasierter Erkennungsansätze zur Modellierung formveränderlicher Objekte mit Hilfe physikalischer Eigenschaften führt zur Klasse der *physicsbased* Ansätze. Hierbei wird ein Objekt als ein System betrachtet, auf welches verschiedenste Kräfte wirken. Diese Kräfte resultieren aus der Interaktion des Objektes mit seiner Umwelt und zeigen sich zum Beispiel in Form von Bewegungen. Weitere physikalische Eigenschaften eines Objektes sind zum Beispiel seine Oberflächenbeschaffenheit, die sich im Bild direkt als Reflexionen niederschlagen kann. Wenn diese physikalischen Eigenschaften bekannt sind und im Modell festgehalten werden, so besteht die Möglichkeit, zum einen das Aussehen eines Objektes in einer bekannten Umgebung vorherzusagen und zu simulieren. Zum anderen kann dies aber auch genutzt werden, um eine Objekterkennung durchzuführen und die äußeren Parameter zu bestimmen.

In diesen Bereich der Objekterkennung fallen eine Vielzahl von Arbeiten, die sich mit der Modellierung der verschiedenen Randbedingungen und Parameter beschäftigen. So gehören hierzu zum Beispiel Modelle zur Beschreibung von Reflexionen, die über das klassische Lambertsche Gesetz hinausgehen (siehe z.B. [60]). Die Modellierung natürlicher Objekte, die ihr Aussehen durch Krafteinwirkungen von außen oder durch innere Prozesse wie die Muskeltätigkeit verändern, ist

ein anderer sehr wichtiger Teilbereich, speziell für die Objekterkennung. Hierdurch können Objekte beschrieben werden, die entweder im Laufe der Zeit ihre äußere Form langsam, z.B. durch Wachstumsprozesse oder Alterung verändern. Aber auch schnelle Formveränderungen z.B. durch Muskelbewegungen können beschrieben werden, um auf diese Weise Menschen, Tiere oder Gesichter zu erkennen [172]. Desweiteren gehört hierzu die Modellierung von Interaktion verschiedener Objekte, die zu einer Szenenauswertung auf einer sehr hohen Verständnisebene führen kann.

Metaxas stellt in [120] ein Framework vor, welches deformierbare Objektmodelle verwendet, um globale und lokale Verformungen eines Objektes erfassen zu können. Damit auch komplexere Objekte mit nur wenigen globalen Formparametern beschrieben werden können, benutzt Metaxas einen Verformungsprozess, den er *shape blending* nennt. Hierbei wird mittels einer Verformungsfunktion α der Übergang von einem Volumenprimitv auf eine andere Form beschrieben. Dabei erfolgt in einem Fittingprozess eine schrittweise Annäherung an die gewünschte Form. In [120] wird in diesem Zusammenhang vorgestellt, wie sich aus einer Punktwolke, die einem Tiefenbild entnommen wurde, aus dem initialen Volumenprimitiv Kugel nach mehreren Iterationsschritten der Verformung eine Volumenbeschreibung einer Tasse herausbildet. In dieser Volumenbeschreibung hat sich nicht nur der Tassengrundkörper als Zylinder entwickelt, sondern es konnte auch eine Beschreibung des Tassengriffes erzeugt werden. Zudem wird eine symbolische Beschreibung der Tasse erzeugt, die in einem Grafen die geometrischen Beziehungen zwischen dem Grundkörper Zylinder, dem Tassengriff und dem Loch im Griff beschreibt.

Metaxas beschreibt weitergehend, wie bei durch die Verwendung von Funktionen anstelle von Konstanten für die globalen Verformungsparameter auch zeitabhängige Verformungen eines Objektes modelliert werden können. Diese Möglichkeit verwendet er, um in biomedizinischen Anwendungen Objektbeschreibungen durchführen zu können. Er stellt dies am Beispiel eines Modells für die linke Herzkammer und seiner Bewegungen vor. Durch die Analyse von Bildsequenzen können darüber hinaus auch zusammengesetzte Objektstrukturen, die sich unabhängig voneinander bewegen können, automatisch erzeugt werden. Am Beispiel einer sich bewegenden Person zeigt er, wie die Positionen

einzelner Gelenke detektiert werden und wie sich die verformbaren Objektmodelle hieran anpassen. Dazu erfolgt an der detektierten Gelenkposition eine Zerteilung des initialen Volumenprimitives in zwei sich leicht überlappende Primitive. Dies wird in Abbildung 3.12 verdeutlicht.

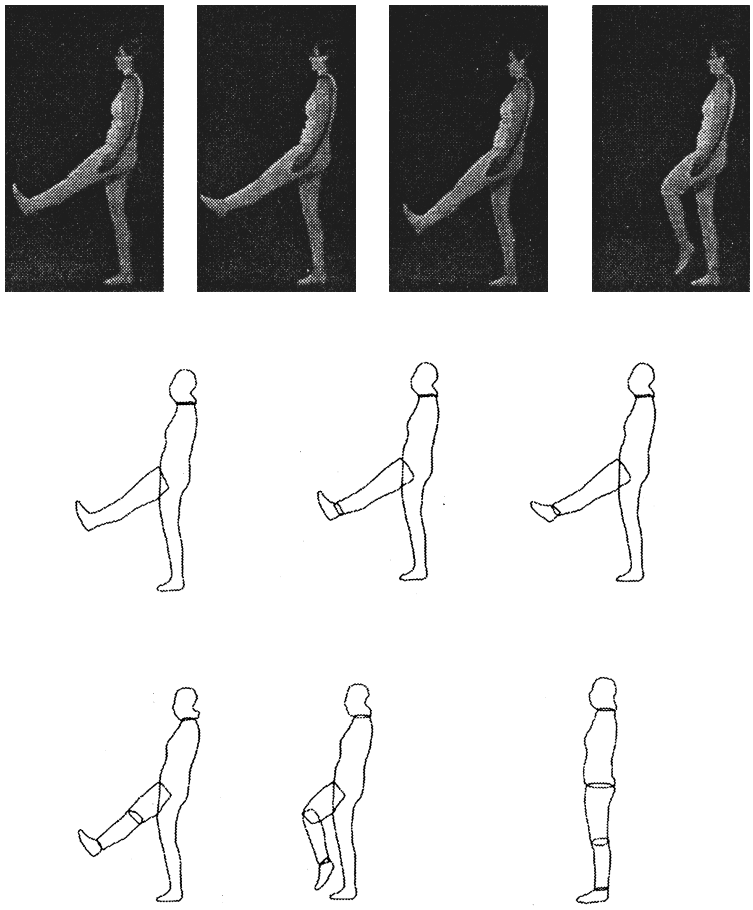


Abb. 3.12: Automatische Erzeugung eines Körpermodells (aus [120], Seite 243)

Eine Erkennung der a priori unbekannten Objekte in einer Szene erfolgt jedoch noch nicht. Metaxas beschränkt sich zur Zeit in seinen Arbeiten noch auf den Aufbau eines Objektmodells und die Detektion und Verfolgung der Veränderungen des modellierten Objektes. Seine Verfahren dienen noch nicht zur Unterscheidung verschiedener Objektmodelle. Dies wird von ihm als Zielsetzung diskutiert. Aufgrund der Komplexität der bestehenden Verfahren sehen Hebert u.a. den Einsatz

solcher Verfahren jedoch eher in zeitunkritischeren Bereichen wie z.B. Rendering oder virtuelle Realität, nicht jedoch in der Objekterkennung.

> „By contrast, it is difficult to construct abstractions of accurate but complex physical models for use in high-level vision tasks. [...]
>
> Although recognition is the natural application for most object representation research, it may not be so for physics-based models“ (aus [80], Seite 10)

3.5 Erkennung durch Bildfolgenauswertung

Die oben angeführten Ansätze, die zunächst einmal beschreiben, wie ein einzelnes Bild eines Objektes ausgewertet werden kann, können durch die Auswertung von Bildfolgen ergänzt werden. Hierbei kann unterschieden werden, ob

1. die Erkennung aufgrund einer Folge von einzelnen Bildauswertungen und der anschließenden Verschmelzung der einzelnen Ergebnisse erfolgt (Bradski u. Grossberg [21])
2. die Erkennung durch Auswertung der Übergänge der einzelnen Erkennungsergebnisse, von einem Bild zum nächsten (Aspektübergänge) erfolgt, wie z.B. in den Arbeiten von Mertsching [113] oder von Seibert und Waxman [163] vorgeschlagen.
3. die einzelne Bildinformation nur implizit ausgewertet wird, indem direkt die Veränderungen in der Bildfolge zum Beispiel in Form des optischen Flusses ausgewertet werden, wie z.B. von Nagel vorgeschlagen oder auch von Milanova und Büker [123] untersucht.

Einige Arbeiten zu dieser Thematik sollen nun exemplarisch vorgestellt werden.

VIEWNET

Im VIEWNET System von Bradski und Grossberg erfolgt auf recht einfache Art eine Auswertung von Bildsequenzen zur Steigerung der Erkennungsrate, die durch eine Einzelbilderkennung erzielt wird [21]. Dabei führen sie zunächst für jedes Bild der Sequenz eine Einzelbilderkennung durch, bei der im ersten Schritt durch ein einfaches Schwellwertverfahren eine Objekt-Hintergrundtrennung durchgeführt wird. Im nächsten Schritt wird dann eine Reihe von Kantendetektoren unterschiedlicher Größe und Orientierung verwendet, um eine Kantenbeschreibung des Objektes zu extrahieren. Dazu wird an jeder Position die Summe der Filterantworten gebildet. Das resultierende Kantenbild wird dann anschließend logarithmisch-polar transformiert, wobei als Zentralpunkt für die Transformation der Schwerpunkt des Kantenbildes verwendet wird. durch einfache Normierungsoperationen erhält man nun eine positions-, größen- und orientierungsinvariante Repräsentation der Objekte, die im weiteren dann noch durch Glättung und Unterabtastung in einer Pyramidenstruktur schrittweise komprimiert und gröber kodiert wird.

Zur Klassifikation der so erzeugten Merkmalsvektoren verwenden Bradski und Grossberg mit dem *FuzzyARTMAP* ein neuronales Netz aus der Reihe der *ART*-Netze, das in einer überwachten Trainingsphase ansichtenbasiert verschiedene charakteristische Ansichten der Objekte lernt. Dabei werden die charakteristischen Ansichten durch die Gewichtsvektoren einer Zwischenschicht des Netzwerkes repräsentiert. In der Ausgabeschicht steht für jedes der gelernten Objekte ein Neuron zur Verfügung, so dass das maximal erregte Neuron in der Ausgabeschicht das erkannte Objekt repräsentiert.

Diese wenigen hier kurz skizzierten Verarbeitungsschritte reichen bereits aus, zu relativ guten Erkennungsraten bei der Erkennung von Flugzeugsilhouetten zu gelangen [21]. Um die Erkennungsrate noch weiter zu steigern, werden aber darüber hinaus auch Bildsequenzen verarbeitet. Hierbei werden die Klassifikationsergebnisse eines jeden Einzelbildes miteinander kombiniert. Hierzu verwenden Bradski und Grossberg ein Abstimmungsverfahren mit einem einfachen Mehrheitsentscheid. Diese Vorgehensweise stellt die einfachste Form der Bildsequenzauswertung dar, da hierbei keinerlei strukturelle Zusammenhänge

zwischen den Bildern der Sequenz ausgewertet werden, sondern völlig losgelöst von einer zeitlichen Reihenfolge eine Menge von Klassifikationsergebnissen betrachtet und miteinander verrechnet wird.

Die im folgenden beschriebenen Ansätze gehen hierüber hinaus und betrachten auch die Reihenfolge der Bilder und der dazu gehörigen Auswertungen.

Aspektübergänge

Seibert und Waxman verwenden in ihrem System eine ansichtenbasierte Repräsentation der gelernten Objekte [163]. Dazu wird zunächst mittels eines einfachen Schwellwertverfahrens eine Objekt-Hintergrundtrennung durchgeführt. Als Merkmale des Objektes werden dann Eckenpositionen im Konturverlauf bestimmt. Um eine translations-, skalierungs- und orientierungsinvariante Repräsentation zu erhalten, wird in einem nächsten Verarbeitungsschritt eine logarithmisch-polare Transformation auf dem resultierenden Eckenbild durchgeführt. Diese wird auf dem Schwerpunkt der detektierten Eckenkoordinaten aufgesetzt, um eine translationsinvariante Darstellung zu erhalten. Nach einer Zentrierung der Eckenkonstellation in der logarithmisch-polaren Bildmatrix steht nun eine skalierungs- und orientierungsinvariante Repräsentation zur Verfügung. Die noch ausstehende Aufgabe besteht nun darin, die wesentlichen charakteristischen Objektansichten sowie die Übergänge zwischen diesen Ansichten zu ermitteln und zu lernen. Dazu verwenden Seibert und Waxman ein *ART*-Netzwerk [39], welches in einem unüberwachten Verfahren die Merkmalsvektoren der charakteristischen Objektansichten lernt. Für die Repräsentation der Übergänge wird eine n x n-Matrix verwendet, in der für Paare von Ansichten gespeichert wird, mit welcher Wahrscheinlichkeit sie in einer Bildsequenz aufeinanderfolgen. Hierbei ist natürlich für jedes Objekt eine eigene Übergangsmatrix notwendig. Die charakteristischen Objektansichten werden in einer Aspekt-Graph ähnlichen Form miteinander in Beziehung gesetzt. Da in diesem System die einzelnen Ansichten nur durch sehr wenige grob gerasterte Merkmale repräsentiert werden, sind sehr viele der Ansichten mehrdeutig und können nicht direkt einem spezifischen Objekt zugeordnet werden. Erst durch die Auswertung von Bildsequenzen und durch die Auswertung der objektspezifi-

schen Ansichtenübergänge lässt sich diese Mehrdeutigkeit aufheben [163], [187].

STORE-Netzwerke

Bradski, Carpenter und Grossberg stellen in [20] eine neuronale Architektur vor, die in der Lage ist, die zeitlichen Zusammenhänge von Mustern einer Bildsequenz auszuwerten. Dabei werden die Muster in ein sogenanntes *item-and-order coding* transformiert. Hier wird nicht nur der Übergang zwischen verschiedenen Aspekten codiert, sondern gleichzeitig auch der zeitliche Zusammenhang festgehalten.

Massad, Mertsching und Schmalz [113] verwenden ein solches *STORE* Netzwerk, um in ihrem ansichtenbasierten 3D-Objekterkennungssystem Mehrdeutigkeiten und auch Fehlklassifikationen durch die Auswertung von Bildsequenzen aufzulösen.

In ihrem Ansatz verwenden sie Gaborjets, mit denen lokale Kantenstrukturen in 12 Orientierungen und 2 Auflösungsebenen detektiert werden und als ein Satz von 24-dimensionalen Merkmalen in einer Merkmalskarte abgelegt werden. In dieser wird eine Auflösung von 64 x 64 Bildpunkten verwendet. Für die Klassifikation dieser Merkmalskarten wird das von v. d. Malsburg vorgeschlagene *Dynamic Link Matching* Verfahren eingesetzt (siehe auch Seite 47).

Zur Lösung des Klassifikationsproblems untersuchen Massad, Mertsching und Schmalz die Regularität der sich ergebenden Netze für jedes der gelernten Muster. Während die reguläre Anordnung der Netzwerkknoten ein starker Hinweis für die Ähnlichkeit zweier Muster ist, deutet ein irreguläres Netzwerk auf großes Unterschiede zwischen zwei Mustern.

Diese Klassifikationsergebnisse werden als Eingabe für ein *STORE* Netzwerk verwendet, das das zeitliche Auftreten verschiedener Objektansichten aufzeichnet und kodiert. In einem letzten Schritt werden dann zwei Netzwerke der *ART*-Architektur verwendet, um zunächst das Ergebnis des *STORE*-Netzwerks auf eine Ansichten-Sequenz und um dann anschließend diese auf eine Objektklasse abzubilden.

Bildfolgenauswertung mit Assoziativspeichern

Einen anderen Weg verfolgt ein eigener Ansatz, der in Zusammenarbeit mit Milanova entwickelt wurde [123], [124]. Zwar werden auch dort Bildsequenzen eines Objektes ausgewertet. Es erfolgt jedoch keine Erkennung aufgrund von Einzelbildern. Es wird statt dessen die Änderung zwischen den Bildern der Sequenz gelernt und für eine Objekterkennung verwendet. Die Änderung der Bildinformation bei einer bekannten Bewegung des Objektes enthält implizit Information über die dreidimensionale Struktur des Objektes. Dies machen sich auch die zahlreichen *shape from motion* - Ansätze zunutze, die diese Strukturinformation wieder explizit zugänglich machen.

In dieser Arbeit wird dabei auf den Ideen von Little und Boyd aufgesetzt, die ein System zur Personenidentifikation aufgrund der typischen Gehweise einer Person entwickelt haben [105]. Analog zu der Vorgehensweise von Little und Boyd wird mit Hilfe des Verfahrens von Bülthoff, Little und Poggio [24] der optische Fluss zwischen den Bildern der Sequenz berechnet. Dabei wird davon ausgegangen, dass die Bewegung des Objektes in der Sequenz von konstanter Art ist. In ihren Experimenten befinden sich die Objekte auf einem Drehteller und es werden in 5° Schritten Bilder des Objektes aufgenommen. Auf die Berechnung des optischen Flusses folgt dann eine Merkmalsextraktion, bei der ein Satz von 13 Merkmalen berechnet wird, der den Charakter des optischen Flusses der verschiedenen Objekte wiedergibt. Hierzu erfolgt eine Segmentierung des optischen Flusses. Für eine bewegte Region werden dann Momente niedriger Ordnung bestimmt. Auf diese Weise entsteht eine Zeitreihe der 13 Merkmale, die dann einem Assoziativspeicher für die Wiedererkennung zugeführt wird.

Der verwendete Assoziativspeicher basiert auf dem Paradigma der *Cellulären Neuronalen Netzwerke* und wurde zuvor in einem Syntheseschritt auf die zu lernenden Merkmalsreihen adaptiert [47], [106]. Abb. 3.13 zeigt den Aufbau einer Zelle eines CNN.

Obwohl nur sehr einfache Merkmale verwendet werden, die eigentlich für eine Unterscheidung der Objekte nicht ausreichen, wird dennoch eine robuste Erkennung erzielt. Diese begründet sich allein darin,

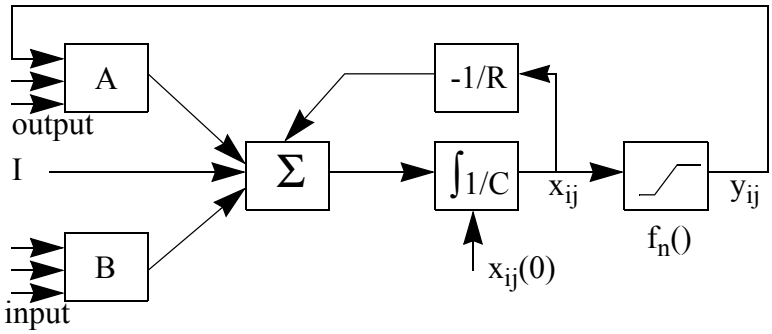


Abb. 3.13: Blockdiagramm einer Zelle des Cellulären Neuronalen Netzwerkes

dass eine entsprechend lange Zeitreihe ausgewertet wird und somit auch diese einfachen Merkmale genügend Information beinhalten.

Abbildung 3.14 zeigt einige der verwendeten Testobjekte und die als Grauwertbild visualisierten Merkmalsvektoren. Dabei beschreibt eine Spalte jeweils ein Merkmal.

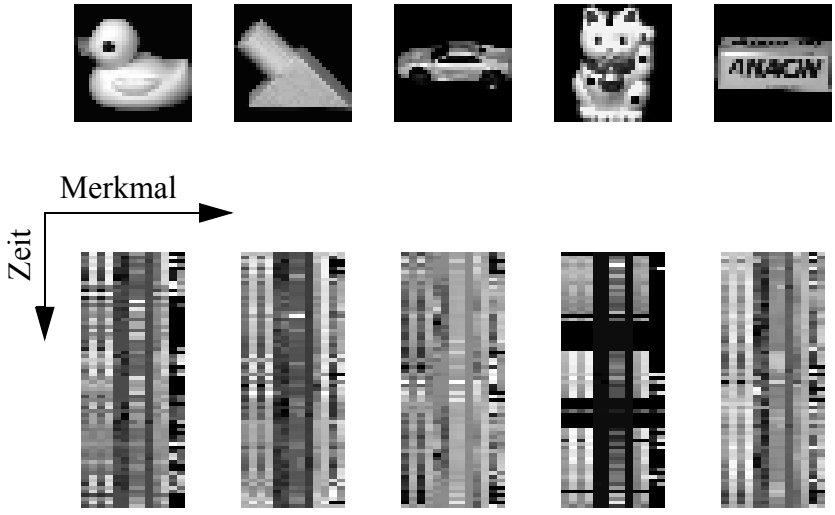


Abb. 3.14: Einige Testobjekte und die dazugehörigen Merkmalsvektoren

3.6 Aktive Objekterkennungssysteme

Wie bereits in Kapitel 2 erwähnt existieren nur relativ wenige Arbeiten zum Thema der aktiven Objekterkennung, worunter im Zusammenhang dieser Arbeit verstanden wird, dass aufgrund von Sensordaten aktiv die Parameter der Kamera verstellt werden, um somit zu neuen Bildern, zu neuen Sensordaten zu gelangen. Somit können aktiv die für die Erkennung benötigten Informationen aus einer Szene extrahiert werden, um die Erkennung robust und eindeutig zu machen. Dabei stellen die Kameraposition und der Blickwinkel der Kamera auf die Szene, also die sechs Freiheitsgrade der Kameralage, die wichtigsten Parameter dar. Wesentliche Aufgabengebiete sind daher die Blicksteuerung und die Frage, wohin in einer gegebenen Szene als nächstes geschaut werden soll, sowie die Frage, wie die einzelnen Erkennungsergebnisse miteinander zu einem einheitlichen Klassifikationsergebnis verknüpft werden können und wie die Integration in ein funktionsfähiges System gestaltet werden kann.

Maver und Bajcsy

Maver und Bajcsy stellen in [114] eine Blicksteuerung für ein aktives binokulares Kamerasystem vor, das mit einem Laser gekoppelt ist, dessen Lichtstrahl in kontrollierter Weise in die Szene gerichtet und durch diese hindurch bewegt wird. Durch die Auswertung der Position dieses Lichtstrahls in beiden Kamerabildern kommt man bei diesem Verfahren der aktiven Triangulation zu sehr gut aufgelösten Tiefenbildern.

Aufgrund von Verdeckungen kann aber natürlich keine vollständige Tiefenkarte bestimmt werden. Aus diesem Grunde werden aus verschiedenen Blickrichtungen Bilder aufgenommen, um die vorhandenen Mehrdeutigkeiten aufzulösen. Die Aufgabe der Blicksteuerung liegt nun in der Bestimmung einer Folge von Blickpunkten, von denen aus Tiefenkarten generiert werden können, die zu einer vollständigen Rekonstruktion der 3D Szene führen. Dabei starten sie von einem beliebigen Blickpunkt und entscheiden dann sukzessive auf der Basis der unvollständigen Information der bereits bestimmten Tiefenkarten,

welcher Blickpunkt als nächstes angesteuert werden sollte. Die vollständige Rekonstruktion ergibt sich dann als Vereinigung der Tiefenkarten aller ausgewerteten Blickrichtungen.

Wilkes und Tsotsos

Wilkes und Tsotsos versuchen für die Objekterkennung eine genormte Blickrichtung auf ein beliebig positioniertes Objekt zu finden [190]. Dazu definieren sie eine Standardansicht für ihre Objekte. Dabei werden zwei auffällige Liniensegmente des Objektes ausgewählt und die Standardansicht so bestimmt, dass diese zwei Liniensegmente eine maximale Länge im Bild aufweisen. Bei der Erkennung versuchen Wilkes und Tsotsos nun zunächst eines dieser Liniensegmente L_0 zu finden. In mehreren Schritten wird dann die Blickrichtung der an der Hand eines 5-DoF Roboterarms montierten Kamera in kleinen Schritten verändert. Dabei wird das Liniensegment L_0 in den Bildern verfolgt, und die Kamerapositionen werden durch Translationsbewegungen jeweils so korrigiert, dass L_0 zentriert und in einer vorgegebenen Ausdehnung im Bild erscheint. Dieses schrittweise Verändern der Blickrichtung dient dazu, festzustellen, von welcher Position aus betrachtet, das Liniensegment L_0 seine größte Ausdehnung zeigt.

Im Anschluss hieran wird ein zweites Liniensegment L_1 hinzugezogen, welches nicht parallel zu L_0 aber in dessen Nähe liegt, um sicherzustellen, dass es sich um ein Liniensegment des gleichen Objektes handelt. Nun wird wieder schrittweise die Blickrichtung variiert, wobei die Bewegungen senkrecht zu L_0 erfolgen, um jetzt das Abbild von L_1 zu maximieren. Die zu dieser Kameraposition gehörende Objektansicht wird dann als Standardansicht betrachtet und für die Objekterkennung auf der Basis eines Indexingverfahrens verwendet.

Rimey und Brown

Eine auf Bayes-Netzen basierende Vorgehensweise zur Bestimmung von Kamerabewegungen schlagen Rimey und Brown vor [154]. Dazu modellieren sie in ihrem System *TEA-1* sowohl Objekte als auch Vorgehensweisen zur Lösung gestellter Aufgaben in *Bayes-Netzen* [42], [84]. In solchen *Bayes-Netzen* beschreiben die Knoten die für die

jeweilige Domäne wichtigen Zufallsvariablen. Eine Kante $e = (x, y)$ des Netzes zwischen den Knoten x und y, die die Zufallsvariablen X und Y darstellen, beschreibt, welchen Wert Y besitzt, wenn X einen gegebenen Wert besitzt. Diese Wertepaare werden dazu in Tabellen abgelegt. Die Glaubwürdigkeit eines Wertes der Zufallsvariablen X ist definiert als $BEL(X) = P(X \mid E)$, also in Abhängigkeit aller im Netz vorhandener Evidenz. Wird nun mit Hilfe geeigneter Bildverarbeitungsoperationen eine im Netz als Zufallsvariable beschriebene Bildstruktur extrahiert, so kann dieser Variablen direkt eine Evidenz zugewiesen werden, die dann im Netz propagiert wird. Es existieren verschiedene Algorithmen, die eine solche Neuberechnung aller Glaubwürdigkeitswerte im Netz vornehmen [146].

Dabei dient zunächst ein *Part-of Netz* zur Modellierung von Teilstrukturbeziehungen sowohl auf einer Objektebene (welche Teilstrukturen gehören zu welchem Objekt?) als auch auf Szenenebene (welche Objekte sind in der Szene enthalten?). Ein *Expected Area Netz* dient darüber hinaus dazu, geometrische Relationen zwischen den Objekten sowie zwischen den Teilstrukturen und den Objekten zu modellieren. Dabei besitzt das *Expected Area Netz* den gleichen Aufbau wie das *Part-of Netz*. Das aufgabenspezifische Wissen ist in einem *Task Netz* abgelegt, wobei zu jeder modellierten Aufgabe ein eigenes Netz gehört. In diesen Netzen sind z.B. die für eine Szeneauswertung und Objekterkennung notwendigen Bildverarbeitungsoperationen beschrieben. Die in den Netzen propagierten Evidenzen führen dann zur Auswahl der nächsten Aktion. Während jedoch die Evidenzbestimmungen im *Part-of* und *Task Netz* unabhängig voneinander erfolgen, werden die durch den Erkennungsvorgang sich in diesen Netzen verändernden Evidenzwerte auf die *Task Netze* übertragen, um so die nächste Aktion auswählen zu können. Das Ausführen einer Aktion kann dann wiederum die Zufallsvariablen und deren Glaubwürdigkeiten in den beiden übrigen Netzen verändern.

Für die Auswahl des nächsten Blickpunktes und der dazu gehörigen Kamerabewegung dient das *Expected Area Netz*, in dem die geometrischen Relationen zwischen den modellierten Objektstrukturen beschrieben sind. Dabei wird in *TEA-1* von einem festen Kameraursprung und der Möglichkeit zu einer Schwenk-Neigebewegung der Kamera (pan- und tilt-Bewegung) ausgegangen. Somit ist der Ort eines Objek-

tes in der Szene definiert durch die zwei Kamerawinkel $\theta = (\varphi_{pan}, \varphi_{tilt})$, die zu einer Zentrierung des Objektes im Bild führen. Diese zwei Winkel werden nun in den Knoten des *Expected Area Netzes* repräsentiert, um die geometrischen Relationen zu beschreiben. Dazu erfolgt eine Diskretisierung des Blickfeldes, so dass in jedem Knoten eine diskrete zweidimensionale Variable repräsentiert wird. Die Glaubwürdigkeitsfunktion $BEL(\theta)$ ist dann eine Funktion dieser zwei Kamerawinkel.

Jeder Kante zwischen zwei Knoten x und y wird nun eine bedingte Wahrscheinlichkeit $P(\theta_y \mid \theta_x)$ zugewiesen. Dabei ergibt sich also für jeden Wert von θ_x eine Tabelle von Wahrscheinlichkeitswerten für alle möglichen Werte von θ_y. Somit erhält man bei einer Diskretisierung des Blickfeldes in ein n x n-Gitter für jede Kante im Netz eine Tabelle der bedingten Wahrscheinlichkeiten einer Größe von n^4 Einträgen. Selbst bei einer sehr groben Diskretisierung der Kamerawinkel in 32 Stufen ergibt sich so für jede Kante eine Tabelle mit mehr als einer Million Einträgen. Da diese nicht mehr sinnvoll handhabbar sind, stellen Rimey und Brown eine Vorgehensweise vor, diese Tabellen zu verkleinern.

Im wesentlichen fließen hierzu zwei Ideen ein. Erstens stellen sie fest, dass die Wahrscheinlichkeitsverteilung $P(\theta_y \mid \theta_x)$ relativ zur Position von x konstant ist. Somit ist es also überflüssig, eine vollständige Tabelle für alle Werte θ_x zu verwalten. Die Größe der Tabelle reduziert sich somit um zwei Dimensionen. Zweitens halten sie fest, dass die Auflösung der Tabelle nicht der Diskretisierung des Blickfeldes entsprechen muss. Sie wählen für ihre Wahrscheinlichkeitstabelle daher eine halbierte Auflösung und nennen diese Tabelle *Relationenkarte*. Bei einer 32 x 32 Auflösung des Blickfeldes ergibt sich somit eine 16 x 16 Relationenkarte anstelle einer Wahrscheinlichkeitstabelle von $32^4 = 1048576$ Einträge. Die Abbildung 3.15 stellt zu zwei Tabellen die entsprechenden Relationenkarten gegenüber.

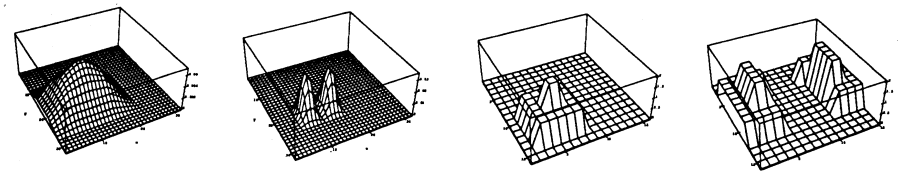


Abb. 3.15: In zwei Tabellen für die bedingten Wahrscheinlichkeiten bei gegebenem θ_x werden die schlechter aufgelösten Relationenkarten gegenübergestellt, die durch einen Schiebe- und Skalierungsmechanismus auf den jeweilign Wert θ_x angepasst werden (aus [154], S. 545).

Dickinson, Christensen, Tsotsos, Olofsson

In Bezug auf die Objektmodellierung waren schon die Arbeiten von Dickinson, Christensen, Tsotsos und Olofsson vorgestellt worden (Seite 42 und [49], [50]). Sie verwenden eine modifizierte Form der Aspektgraphen, die sie in ein aktives Objekterkennungssystem einbinden. Dabei ergänzen sie die Kanten des Aspektgraphen um die bedingten Wahrscheinlichkeiten, die beschreiben, wie verlässlich die Erkennung eines Aspekts auf das Vorhandensein des gesuchten Objektes schließen lässt. Somit interpretieren sie den Aspektgraphen als ein *Bayes-Netz*. Bei der Blickpunktauswahl wird nach dem initialen Erkennen eines Aspektes versucht, den Aspekt des Objektes finden, der im *Bayes-Netz* die Erkennungswahrscheinlichkeit für das vermutete Objekt maximiert. Nachdem dieser Aspekt im Netz bestimmt wurde, werden die in den Kanten des Aspektgraphen gespeicherten Übergänge ausgewertet, um einen Verfahrweg für die Kamera zu bestimmen, der vom aktuellen Blickpunkt zum gesuchten Aspekt führt. Dort wird eine erneute Bildauswertung durchgeführt, um die Hypothese zu verifizieren. Ist dies noch nicht möglich, weil zum Beispiel im neu ausgewerteten Bild die gesuchte Ansicht des Objektes verdeckt ist, so wird die Auswertung eines weiteren Aspekts hinzugezogen, von dem wiederum eine Maximierung der Erkennungswahrscheinlichkeit erwartet wird. Abbildung 3.16 verdeutlicht diese Vorgehensweise der Blickpunktgenerierung.

Während der Bewegung zum neuen Blickpunkt wird ein Trackingverfahren eingesetzt, welches die Bildinhalte mit den erwarteten

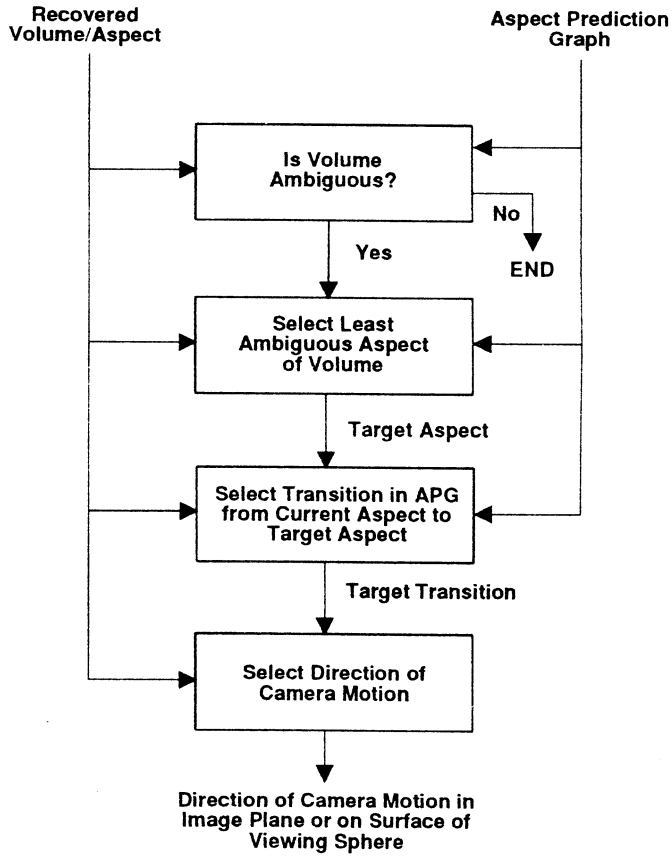


Abb. 3.16: Blickpunktauswertung auf der Basis von Aspektgraphen (aus [50], Seite 250)

Aspekten auf diesem Weg vergleicht. Auf diese Weise wird also ein zusätzliche Informationsquelle für den Erkennungsprozess integriert. Es werden ganz gezielt die Bewegung und die aufgenommenen Bilder mit den Wegen im Aspektgraphen in Verbindung gebracht.

In dem beschriebenen System wird jedoch nicht deutlich, inwieweit das System auf komplexe Szenen übertragen werden kann. Dickinson stellt in erster Linie Untersuchungen an Blockwelten vor.

3.7 Zusammenfassender Vergleich der Objekterkennungsparadigmen

Die vorangegangene Übersicht über die prinzipiellen Erkennungsansätze und einige Systeme hat sicherlich schon deutlich gemacht, dass es noch keinen Konsens zur Festlegung einer „guten“ Objektrepräsentation gibt. Vielmehr werden die verschiedenen Paradigmen von unterschiedlichen Verfechtern der jeweiligen Strategie weiterverfolgt, die auf diesen Paradigmen basierenden Systeme weiterentwickelt und an die jeweilige Problemstellung angepasst.

Wenn es auch sehr schwierig ist, die verschiedenen Ansätze zu bewerten, so besteht aber generelle Übereinstimmung bei den Forderungen an ein „gutes“ Erkennungssystem:

- gute Adaptierbarkeit an verschiedene Aufgabenstellungen und Anwendungsgebiete;
- die Fähigkeit zur Verarbeitung großer Objektmodellbasen;
- die Erkennung der Objekte in komplexen (real world) Bildern;
- die Möglichkeit, die Reaktionen des Systems zu analysieren und zu verstehen, um eine effektive Weiterentwicklung zu gewährleisten.

Von den bestehenden Paradigmen werden diese Anforderungen bislang nur in eingeschränktem Maß erfüllt. So standen ja bereits am Anfang dieses Kapitels verschiedene Fragen und Probleme zu den zwei Ansätzen der objektzentrierten- und der beobachterzentrierten Repräsentation (Seite 28), die hier noch einmal kurz zusammengefasst werden sollen.

Offene Fragen und Probleme der objektzentrierten Repräsentation

Was ist eine Teilstruktur eines Objektes?

Wie können Objekte in Teilstrukturen dekomponiert werden?

Wie kann aus der Bildinformation eine stabile Dekomposition gewonnen werden?

Welche Beziehungen zwischen Teilstrukturen können aus der Bildinformation extrahiert werden?

Offene Fragen und Probleme der beobachterzentrierten Repräsentation

Wie kann eine Abstraktion und Generalisierung von Objektansichten erfolgen?

Wie können ansichtenbasiert partiell verdeckte Objekte erkannt werden?

Wie kann eine Objektsegmentation vermieden werden?

Wie kann im Falle großer Objektdatenbasen die Menge der Objektansichten sinnvoll strukturiert werden?

In den nachfolgenden Kapiteln wird nun ein Erkennungssystem beschrieben, das die Vorteile der beiden Repräsentationsparadigmen miteinander verbindet. Dabei wird auch auf die vorstehend zusammengefassten Fragen eingegangen werden. Der Lösungsvorschlag beruht auf einer Integration beider Paradigmen in einer hybriden Repräsentation, die primär ansichtenbasiert Objekte als Ganzes aber auch in charakteristischen Teilansichten beschreibt. Die für eine solche Modellierung notwendigen Ansichten werden jedoch nicht als eine unstrukturierte Ansichtenmenge betrachtet. Vielmehr werden sie in Graphen organisiert und mit Hilfe einer Dekompositionshierarchie, wie sie aus der objektzentrierten Modellierung bekannt ist, miteinander verknüpft.

Über diesen Integrationsaspekt hinaus ermöglicht die entwickelte Modellierungssprache zusätzlich zur rein deklarativen Objektmodellierung auch die Modellierung von Erkennungsstrategien und die Einbindung aktiver Erkennungsmechanismen.

4 Holistische vs. dekompositorische Erkennung

4.1 Typische Merkmale dekompositorischer Objekterkennung und daraus resultierende Probleme

Während der vergangenen zwei Jahrzehnte ist eine Vielzahl von Methoden aus dem Bereich der *Künstlichen Intelligenz (KI)* in der Bildverarbeitung und dabei speziell im Bereich der Bild- oder Objekterkennung eingesetzt worden. Die prinzipielle Idee dabei ist es, Objektmodelle aufzubauen und diese mit den Bilddaten zu vergleichen. Wenn in dieser Arbeit von Methoden der *Künstlichen Intelligenz* oder von *wissensbasierten Verfahren* gesprochen wird, so ist damit ge-

meint, dass das Wissen, also die Objektmodelle, explizit repräsentiert sind und getrennt von der Verarbeitung des Wissens betrachtet werden. Somit wird also auf die klassische Bedeutung dieser Begriffe zurückgegriffen, die eine deutliche Trennung von Wissen und Verarbeitung sieht (Abb. 4.1).

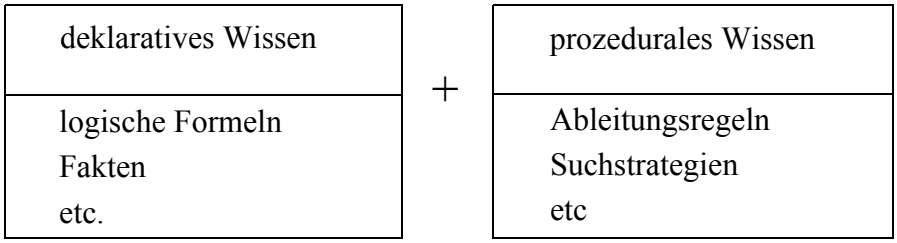


Abb. 4.1: Explizite Trennung von Wissen und Verarbeitung

In der klassischen wissensbasierten Bilderkennung werden dabei die Objektmodelle typischerweise in elementare Bildprimitive wie gerade oder gekrümmte Linien und Kanten, Ecken und Regionen homogenen Grau- oder Farbwertes zerlegt. Man spricht daher auch von einer Dekomposition der Objekte, die über mehrere Hierarchieebenen bis zu den gerade erwähnten Bildprimitiven auf der untersten Ebene der Dekompositionshierarchie führt. Für die Objekterkennung werden dann Bildverarbeitungsmethoden benötigt, die in der Lage sind, die Bildprimitive sicher aus den Bilddaten zu extrahieren. Desweiteren wird ein Kontrollalgorithmus benötigt, der eine möglichst gute Zuweisung von Bildprimitiven zu den Elementen der Objektmodellierung durchführt. Dieses Zuweisungsproblem kann als Suchproblem oder auch als Optimierungsproblem betrachtet werden, bei dem es darum geht, die beste Zuweisung M vom Merkmalsraum zum Modellraum zu finden. Eine grundlegende Einleitung hierzu findet sich in [137], [160].

Eine häufig verwendete Form der Objektrepräsentation sind die *semantischen Netzwerke*. Diese können abstrakt zunächst einmal als ein Graph $G = (V, E)$ mit einer Knotenmenge V und einer Kantenmenge E betrachtet werden. Die Knoten eines semantischen Netzes werden auch als *Konzepte* bezeichnet. In diesen Konzepten werden die Eigenschaften eines Objektes oder auch seiner Teile innerhalb der Dekompositionshierarchie beschrieben, während die Kanten die Relationen zwischen den Konzepten darstellen. In der Objekterkennung haben sich dabei zwei Standardrelationen herauskristallisiert:

- die Teil- / Teil-von- und
- die Spezialisierungs- / Generalisierungs-Relationen.

Diese ermöglichen zum einen den Aufbau einer Dekompositionshierarchie und ermöglichen zum anderen das Zusammenfügen verschiedener Ausprägungen eines Objektes zu Klassenbegriffen. Die Abbildung 4.2 zeigt in einem exemplarischen Netzwerk die Dekomposition mit Hilfe der Teil-Relation.

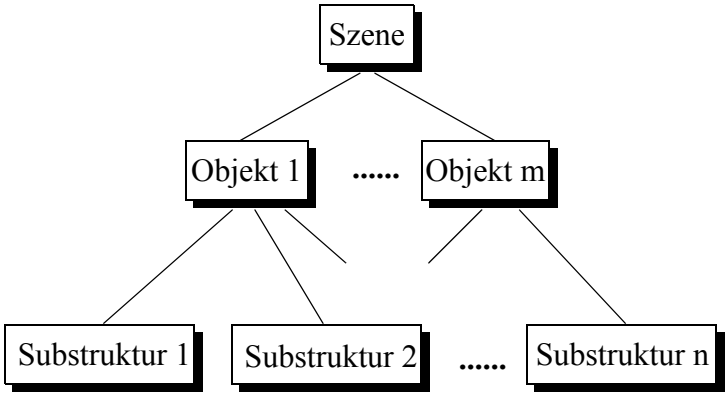


Abb. 4.2: Ein exemplarisches semantisches Netzwerk

Eine solche explizite Objektbeschreibung mit semantischen Netzwerken vereint verschiedene Vorteile. Zum einen ist sie sehr übersichtlich und intuitiv verständlich, zum anderen unterstützt sie inhärent durch die Dekomposition des Objektes in seine Bestandteile die Erkennung von teilweise verdeckten Objekten. Solange nur eine hinreichend große Anzahl an Teilstrukturen erkannt wird, ist auch bei Nichterkennen einzelner Bildprimitive, sei es, weil sie im Bild nicht sichtbar sind oder weil sie durch Störungen im Bild nicht erkannt wurden, das Auffinden eines guten Mappings M und damit die Erkennung des Gesamtobjektes noch möglich.

Es bleiben jedoch einige ganz elementare Probleme dieser wissensbasierten Ansätze bestehen. So muss man zunächst einmal feststellen, dass in den meisten Fällen die extrahierten Bildprimitive nicht dem Objektmodell entsprechen. Durch Bildstörungen kommt es z.B. zum Aufreissen von Kanten oder aber Kanten laufen in den erwarteten Ecken eines Objektes nicht richtig zusammen, etc. In vielen Fällen versucht man, mit Techniken des Gruppierens der Bildprimitive dieses Problem zu umgehen. Dabei beruft man sich auf die Gesetze der *Gestalttheorie*

und fasst Bildprimitive nach vorgegebenen Regeln zusammen [86], [107], [192].

Ein weiteres schwerwiegendes Problem ist die kombinatorische Explosion des Suchraums bei komplexem Bildmaterial und umfangreichen Objektmodellen, so dass ein Auffinden des besten Mappings M wenn überhaupt nur mit sehr großem Aufwand durchführbar ist. Dies kann bereits an einem einfachen Beispiel verdeutlicht werden. Die in Abbildung 4.3 gezeigte Ansicht eines Würfels sei durch neun gerade Linien und die zwischen ihnen geltenden topologischen Relationen (parallel zu, senkrecht zu, links von, rechts von, etc.) modelliert. Bei

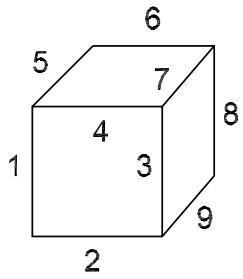


Abb. 4.3: Das Problem der kombinatorischen Explosion; erläutert an einem einfachen Beispielobjekt

dem Versuch der Zuordnung von Bildstrukturen zu den Elementen der Modellierung ergeben sich bereits in diesem sehr einfachen Beispiel 9^9 Möglichkeiten. Und dass, obwohl von optimalen Bedingungen ausgegangen wurde, bei denen vorausgesetzt war, dass alle modellierten Bildprimitive optimal ohne Störungen aus dem Bild extrahiert werden konnten und keine weiteren Strukturen im Bild (z.B. durch Hintergrund, andere Objekte oder durch Rauschen) vorkommen.

Somit sind die wichtigsten Vor- und Nachteile dekompositorischer Objekterkennung bereits herausgearbeitet:

+ die intuitive und übersichtliche Objektmodellierung

+ die inhärente Unterstützung der Erkennung teilverdeckter Objekte

- die Abhängigkeit von der robusten Extraktion der Bildprimitive

- die kombinatorische Explosion des Suchraums in komplexen Anwendungen

Diesem rein wissensbasierten Ansatz soll nun die Vorgehensweise der *holistischen* Objekterkennung, die an zwei Beispielen verdeutlicht wird, gegenübergestellt werden.

4.2 Holistische Erkennungssysteme

An einemBeispiel soll in diesem Abschnitt nun herausgearbeitet werden, was mit dem Begriff *holistische Erkennung* gemeint ist und welche prinzipiellen Vorgehensweisen hiermit verbunden sind. Umfassend wurde dieses Erkennungssystem in [79] beschrieben.

4.2.1 Objektrepräsentation

Auf der Suche nach einer Objektrepräsentation, die auf der einen Seite tolerant gegen geringfügige Variationen z.B. der Position, Größe oder des Aussehens eines Objektes ist, auf der anderen Seite aber eine genügend große Trennschärfe zwischen den Objektklassen sicherstellt, sollen zunächst einmal die Eigenschaften flächenbasierter und kantenbasierter Repräsentationen herausgearbeitet werden. Ohne dabei einen vollständigen Überblick zum Thema Merkmalsextraktion geben zu wollen, wird gezeigt werden, dass flächenbasierte Repräsentationen eine große Toleranz gegenüber den oben genannten Variationen besitzen. Sie besitzen jedoch eine schlechte Trennungsschärfe. Kantenbasierte Repräsentationen zeigen demgegenüber gerade gegenläufige Eigenschaften. In Anlehnung an die Verarbeitungsschritte in visuellen Sehsystemen und die dabei verwendeten Repräsentationen im visuellen Kortex wird im weiteren die Objektrepräsentation auch als Aktivitätsmuster von Modellneuronen beschrieben, mit denen Mechanismen im visuellen Kortex simuliert werden.

Betrachten wir zunächst einmal die flächenbasierte Repräsentation eines Objektes. Dazu werden in einem segmentierten Bild $\boldsymbol{I}$ mit N Abtastpunkten alle Pixel $x_n = 1$ gesetzt, wenn sie zum ausgewählten Segment gehören und $x_n = 0$, wenn sie außerhalb des Segmentes liegen. Damit ergibt sich als ein einfacher flächenbasierter Merkmalsvektor $\boldsymbol{x}$

$$\boldsymbol{x} = (x_1, x_2, \ldots x_n, \ldots x_N) \text{ mit } x_n = \begin{cases} 1 & \text{wenn } x_n \in \text{ Segment} \\ 0 & \text{sonst} \end{cases} \qquad (4.1)$$

Es ist offensichtlich, dass eine solche Repräsentation sehr tolerant ist gegenüber den zuvor genannten Variationen. Abbildung 4.4 verdeutlicht dies. Man sieht, dass ein gelerntes Quadrat mit Merkmalsvektor $\boldsymbol{x_L}$, das leicht verschoben erneut präsentiert wird, nur einen in wenigen Komponenten veränderten Merkmalsvektor $\boldsymbol{x_P}$ aufweist.

Somit wird sich auch ein Ähnlichkeitsmaß nur wenig verändern. Wird als Ähnlichkeitsmaß $d(\boldsymbol{x_L}, \boldsymbol{x_P})$ zum Beispiel das innere Produkt der beiden Vektoren verwendet, so wird dieses durch den Überlapp der beiden Regionen charakterisiert. Über eine einfache Schwellwertbildung kann nun also entschieden werden, ob zwei Objekte zur gleichen Klasse gehören oder nicht. Dabei ist es kein Problem, einen Schwellwert d_0 zu finden, mit dem kleine von grossen Objekten unterschieden werden können oder sehr ausgedehnte Strukturen von kompakten. Wenn dabei jedoch eine große Toleranz angestrebt wird, so können dann, wie in Abbildung 4.4d dargestellt, unterschiedliche Objekte ähnlicher Größe und Ausdehnung nicht mehr voneinander unterschieden werden.

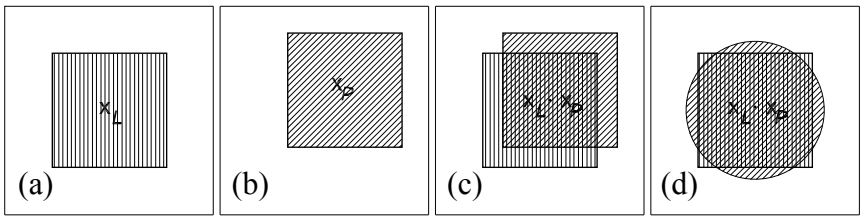


Abb. 4.4: Flächenbasierte Repräsentationen sind tolerant gegen Variationen von Position und Größe (c). Mit zunehmender Toleranz geht jedoch Trennschärfe verloren (d).

Eine Alternative hierzu stellen kantenbasierte Repräsentationen dar. Hierbei wird an jeder Bildposition n ein Satz von Kantendetektoren für L verschiedene Orientierungen verwendet. Somit ergibt sich für jeden Bildpunkt ein Vektor

$$\boldsymbol{k_n} = [k_{n1}, k_{n2}, \ldots, k_{nl}, \ldots, k_L] \text{ mit } l \in \{1,2,\ldots,L\} \tag{4.2}$$

In diesem Vektor beschreibt eine Komponente $k_{nl} = 1$ das Vorhandensein eines entsprechend orientierten Kantenelementes im Bild, während $k_{nl} = 0$ dessen Abwesenheit beschreibt. Für die gesamte Bildbeschreibung ergibt sich daraus ein Vektor

$$\boldsymbol{k} = [\boldsymbol{k_1}, \boldsymbol{k_2}, \ldots, \boldsymbol{k_n}, \ldots, \boldsymbol{k_N}] \text{ mit } n \in \{1, 2, \ldots, N\} \tag{4.3}$$

mit Komponenten $\boldsymbol{k_n}$ für jeden Bildpunkt. Dieser Vektor mit $N \cdot L$ Elementen beschreibt die Konturstrukturen des Bildes und wird als konturbasierte Repräsentation bezeichnet. Aus Abbildung 4.5 wird offensichtlich, dass eine solche konturbasierte Repräsentation gerade die entgegengesetzten Eigenschaften einer flächenbasierten Repräsentation aufweist. Sie zeigt eine große Trennschärfe auch bei Objekten ähnlicher Größe und Ausdehnung, ist aber sehr sensitiv gegenüber leichten Variationen eines Objektes. Das Ähnlichkeitsmaß wird zum Beispiel sogar zu Null bei einer Verschiebung um ein Pixel in diagonaler Richtung (siehe Abb. 4.5c). Somit kann auch nicht durch einfaches Senken des Schwellwertes d_0 eine größere Toleranz erreicht werden.

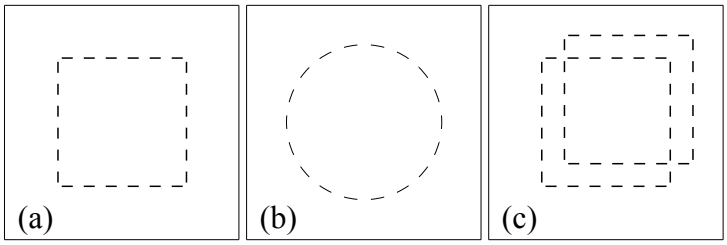


Abb. 4.5: Konturbasierte Repräsentationen besitzen eine hohe Trennschärfe (a, b). Sie zeigen jedoch keinerlei Toleranz gegen Variationen (c).

Im folgenden Abschnitt wird jedoch eine Konturrepräsentation vorgestellt, die unter Beibehalten der Trennschärfe eine gewisse Toleranz gegenüber leichten Variationen aufweist.

4.2.2 Tolerante Konturrepräsentationen

Die Idee einer toleranten Konturrepräsentation wurde abgeleitet aus den Erkenntnissen über biologische Sehsysteme. Neurone des visuellen Kortex, die über orientierte rezeptive Felder verfügen, können als Kantendetektoren k_{nl} betrachtet werden, die entsprechend der zuvor eingeführten Notation an der Position n im visuellen Feld auf Kantenstrukturen einer Orientierung l reagieren. So kann also die kantenbasierte Repräsentation $\boldsymbol{k}$ als ein binarisiertes Aktivitätsmuster der Neurone des visuellen Kortex betrachtet werden. In diesem Vektor sind also all die Komponenten zu 1 gesetzt die für Neurone stehen, deren Aktivität eine bestimmte Mindestschwelle überschreitet, während alle anderen Komponenten zu 0 gesetzt sind und somit fehlende Aktivität anzeigen.

Im visuellen Kortex unterscheidet man dabei sogenannte simple und komplexe Zellen. Diese unterscheiden sich dadurch, dass die Einzugsbereiche oder rezeptiven Felder benachbarter Zellen sich unterschiedlich stark überlappen, wobei es wichtig ist zu erwähnen, dass benachbarte Zellen auch im Gesichtsfeld benachbarte rezeptive Felder besitzen. Somit wird also eine kontinuierliche Bildstruktur durch die Aktivität einer Menge von Kortexzellen charakterisiert, die in ihrer topologischen Anordnung der Topologie der Bildstruktur entsprechen. Eine Linie erzeugt somit ein kettenförmiges Aktivitätsmuster der simplen Zellen des visuellen Kortex (Abb. 4.6).

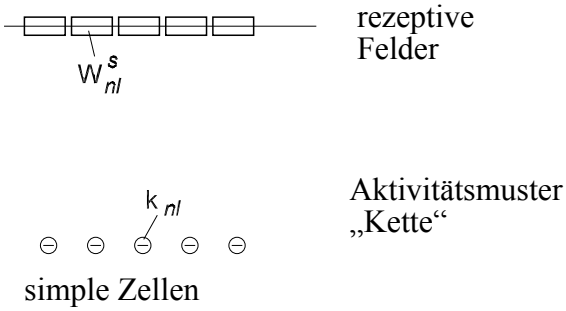


Abb. 4.6: Simple Zellen reagieren mit einem kettenartigen Aktivitätsmuster.

Zusätzlich existiert eine kortikale Repräsentation durch sogenannte komplexe Zellen c_{nl}, die ebenfalls auf Konturelemente der Orientierung l an der Position n im Gesichtsfeld reagieren. Während die rezeptiven Felder W^s_{nl} der simplen Zellen schmal sind und nur einen geringen Überlapp aufweisen, sind die rezeptiven Felder W^c_{nl} der komplexen Zellen groß und stark überlappend. Dadurch wird ein Bildelement durch mehrere rezeptive Felder benachbarter komplexer Zellen erfasst und es entsteht ein wolkenartiges Aktivitätsmuster im visuellen Kortex (Abb. 4.7).

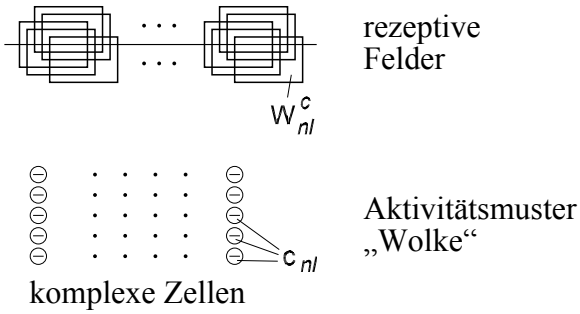


Abb. 4.7: Komplexe Zellen antworten mit einem wolkenartigen Aktivitätsmuster.

In Anlehnung auf die in Abschnitt 4.2.1 gewählte Nomenklatur kann also das Aktivitätsmuster der komplexen Zellen als Vektor

$$\begin{aligned} &\boldsymbol{c} = [\boldsymbol{c_1},\boldsymbol{c_2},\ldots,\boldsymbol{c_n},\ldots,\boldsymbol{c_N}] \quad \text{mit} \\ &\boldsymbol{c_n} = [c_{n1},c_{n2},\ldots,c_{nl},\ldots,c_{nL}] \\ &\text{für } l \in \{1, 2 \ldots L\} \text{ und } n \in \{1, 2 \ldots N\} \end{aligned} \tag{4.4}$$

beschrieben werden.

Von besonderer Bedeutung ist nun, dass dieses wolkenartige Aktivitätsmuster tolerant ist gegenüber Lage-, Größen- und Formvariationen. Abbildung 4.8 verdeutlicht dies am Beispiel eines gelernten Quadrates. Auch bei einer leichten diagonalen Verschiebung des präsentierten Musters $\boldsymbol{c_P}$ gegenüber dem gelernten Muster $\boldsymbol{c_L}$ ergibt sich ein ausreichend gutes Maß an Übereinstimmung. Man kann also wie im flächenbasierten Fall ein Ähnlichkeitsmaß $d(\boldsymbol{c_L}, \boldsymbol{c_P})$ definieren und einen Schwellwert d_0 festlegen, so dass sich eine tolerante Erken-

nung bei $d(\boldsymbol{c_L}, \boldsymbol{c_P}) > d_0$ gewährleistet ist. Gleichzeitig bleibt aber ein gutes Maß an Trennschärfe erhalten, da die einzelnen Komponenten in $\boldsymbol{c}$ nur dann zum Match beitragen, wenn sie an der gleichen Position mit der gleichen Orientierung aktiv sind. Somit unterscheidet sich diese tolerante Repräsentation von einem reinen Ausweiten und Verbreitern einer Kantenstruktur.

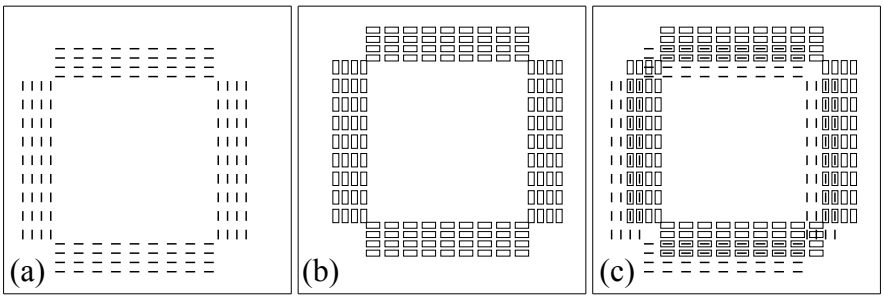


Abb. 4.8: Tolerante Konturrepräsentation eines gelernten (a) und leicht verschoben präsentierten Quadrates (b). Durch die wolkenartige Repräsentationsform wird trotz der Variation eine gute Übereinstimmung festgestellt (c).

Die tolerante Konturrepräsentation kann nun noch um eine Eckenrepräsentation ergänzt werden, die sich dadurch ergibt, dass an den Bildpunkten, an denen zwei Konturelemente mit stark voneinander abweichender Orientierung aneinander stoßen, ein Eckenmerkmal gesetzt wird. Somit ergibt sich analog zur Konturrepräsentation eine tolerante Eckenrepräsentation

$$\mathbf{e} = [e_1, \ldots, e_N] \tag{4.5}$$

Die drei Repräsentationen $\boldsymbol{x}$, $\boldsymbol{c}$, $\boldsymbol{e}$ werden dann in einem Vektor $\boldsymbol{t}$ als Objektrepräsentation zusammengefasst.

In [79] wird über diese Einführung toleranter Konturrepräsentationen hinausgehend auch deren technische Realisierung besprochen. Hier an dieser Stelle sei nur vermerkt, dass orientierungsselektive Gabor-Filter geeignet sind, eine solche Repräsentation aufzubauen (siehe hierzu Abschnitt 4.2.6).

Es sei jedoch noch darauf hingewiesen, dass eine tolerante Konturrepräsentation nicht nur Variationen in Position, Größe und Form ausgleicht, sondern auch leichte räumliche Verkippungen eines Objektes

toleriert. Dies ist von besonderer Bedeutung, wenn ein 3D-Objekt ansichtenbasiert repräsentiert werden soll. Die Toleranz der oben beschriebenen Repräsentation ermöglicht dabei eine recht grobe Diskretisierung der Ansichtensphäre bei der Ermittlung der notwendigen zu speichernden Ansichten des Objektes. Dies wird in Abschnitt 4.2.5 näher erläutert.

4.2.3 Normalisierung der Repräsentationen

Damit Objekte unabhängig von ihrer Entfernung und Position im Bild erkannt werden können, erfolgt im SENROB-System eine Schätzung der entsprechenden Parameter: der Entfernung r, der Orientierung φ und der Position (x, y). Mit Hilfe dieser Parameter kann dann eine Normalisierung erfolgen. Zur Erzielung der Translationsinvarianz wird dazu nach einem Segmentationsschritt der Flächenschwerpunkt des zu untersuchenden Objektes bestimmt und dieser als Fovealisierungspunkt genutzt, d.h. die Kamera wird so verfahren, dass das zu untersuchende Objekt in der Bildmitte liegt.

Auch einer Rotation wird explizit begegnet, indem die Abweichung der Orientierung des Objektes von seiner Vorzugsorientierung bestimmt wird. Dazu wird ein Häufigkeitshistogramm über alle lokalen Kantenorientierungen gebildet. Über das Maximum des Histogramms wird die Orientierung des Objektes festgelegt. Bei symmetrischen Objekten, bei denen sich mehrere gleich hohe Maxima ergeben, wird für jede Vorzugsorientierung eine Repräsentation gelernt.

Anstelle einer oft üblichen Größeninvarianz erbringt das SENROB-System eine Distanzinvarianz, die es erlaubt, gleichartige Objekte unterschiedlicher Größe (z.B. Auto und Spielzeugauto) voneinander zu unterscheiden, gleichzeitig aber ein Objekt, welches aus verschiedenen Entfernungen betrachtet wird und somit in unterschiedlichen Größen auf der Retina abgebildet wird, wiederzuerkennen. Im Falle der Distanzinvarianzleistung muß daher in Abhängigkeit von der geschätzten Entfernung zwischen Kamera und Objekt eine Transformation der Repräsentation erfolgen, so daß unabhängig von der Entfernung das Objekt durch eine konstante Pixelanzahl abgebildet wird.

Da die notwendigen Parameter nicht exakt bekannt sind sondern lediglich geschätzt werden, ist die Toleranzeigenschaft der Kantenrepräsentation von besonderer Bedeutung. Wie zuvor gezeigt, erlaubt diese eine gute Übereinstimmung von präsentierter und gelernter Repräsentation auch bei geringfügigen Schätzfehlern.

4.2.4 Lernen und Erkennen von Objektrepräsentanten

Wenn von Lernen von Objektklassen gesprochen wird, dann ist hiermit oftmals verbunden, dass Objekte in langen Trainingssequenzen immer und immer wieder präsentiert werden, bis sich eine Parameterkonfiguration des Klassifikators herausgebildet hat, die eine ausreichend gut generalisierende Klassifikation ermöglicht. Es wurde aber schon im vorherigen Abschnitt auf die Toleranzeigenschaft einer wolkenartigen Kantenrepräsentation hingewiesen. Diese ermöglicht ein Lernen ohne aufwendige Trainingszyklen, da die tolerante Repräsentation bereits implizit generalisiert. Somit wird also ein Klassenrepräsentant durch einfaches Abspeichern der toleranten Objektrepräsentation erzeugt. Erkennen und Lernen erfolgen einzig über die Auswertung eines Ähnlichkeitsmaßes $d()$, das nun noch ein wenig näher erläutert werden soll.

Betrachten wir der Anschaulichkeit wegen zwei flächenbasierte Repräsentationen $\boldsymbol{x_P}$ und $\boldsymbol{x_L}$. Es sei $p = |\boldsymbol{x_P}|$ und $l = |\boldsymbol{x_L}|$ die Anzahl der aktiven Komponenten in den Vektoren und es sei m die Anzahl der übereinstimmenden aktiven Komponenten. Dann veranschaulicht das Venn-Diagramm in Abbildung 4.9 das folgende Ähnlichkeitsmaß

$$\begin{aligned} d(\boldsymbol{x_L}, \boldsymbol{x_P}) &= m - [(p-m) + (l-m)]/p \\ &= [\boldsymbol{x_L} \cdot \boldsymbol{x_P} - (\boldsymbol{x_L} - \boldsymbol{x_P})^2 \ ]/|\boldsymbol{x_P}| \end{aligned} \tag{4.6}$$

bei dem die Übereinstimmung m („inneres Produkt“) zwischen dem gelernten und dem präsentierten Segment belohnt wird, während die Nichtübereinstimmung („Hamming-Distanz“) bestraft wird. Um ein normiertes Ähnlichkeitsmaß zu erhalten, erfolgt noch eine Division durch die Anzahl der präsentierten aktiven Komponenten. Dieses Ähnlichkeitsmaß lässt sich direkt auf alle binären Merkmalsvektoren über-

tragen, also auch auf die zuvor beschriebene tolerante Repräsentation $\boldsymbol{t} = (\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{e})^T$.

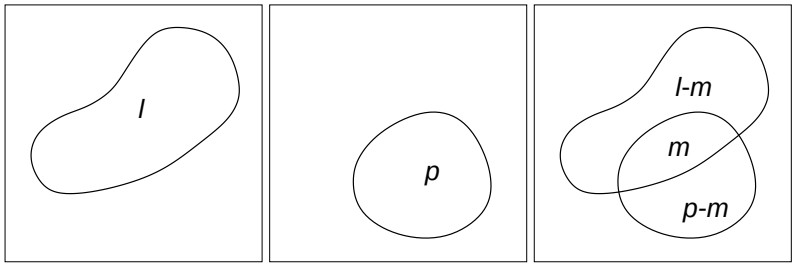


Abb. 4.9: Ein Venn-Diagramm zur Veranschaulichung des Ähnlichkeitsmaßes.

4.2.5 Ansichtenbasierte Repräsentation komplexer 3D-Objekte

Es war eingangs schon erläutert worden, dass die tolerante Repräsentation auch leichte Verkippungen eines Objektes im Raum toleriert, da sich bei geringfügigen Rotationen eines Objektes senkrecht zur Bildebene, das Bild des Objektes nur wenig ändert. Dies ist Voraussetzung für eine problemlose Erweiterung auf eine ansichtenbasierte 3D-Objekterkennung. Dazu wird ein Objekt durch eine Menge von toleranten Repräsentationen beschrieben, die durch Variation der Blickwinkel erzeugt wurden. Die Toleranz der Repräsentation erlaubt nun eine grobe Diskretisierung der Ansichtensphäre und das Speichern nur weniger Repräsentationen, um dennoch das Objekt unter beliebigen Blickwinkeln wiederzuerkennen. Es stellt sich nun die Frage, wieviele Ansichten und welche hierzu notwendig sind. Die Lösung hierzu ist ausführlich von Dunker in [52], [53] beschrieben und soll hier kurz skizziert werden.

Gegenüber der bislang beschriebenen Erkennung normalisierter Repräsentationen müssen für die 3D-Erkennung zwei zusätzliche rotatorische Freiheitsgrade berücksichtigt werden. Diese werden durch die Parameter ϑ und φ wie in Abbildung 4.10 beschrieben. Daher müssen also verschiedene Ansichten (ϑ, φ) in der Ansichtensphäre betrachtet werden.

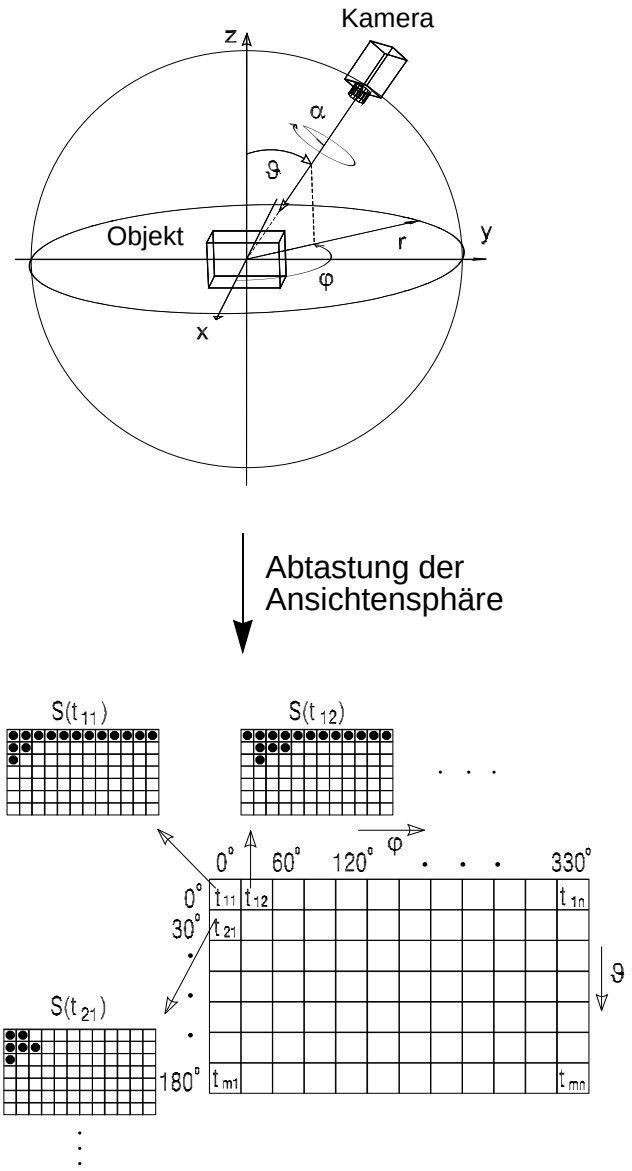


Abb. 4.10: Ein Objekt wird unter verschiedenen Winkeln (ϑ, φ) der Ansichtensphäre gelernt. Die verschiedenen Ansichten decken unterschiedlich große Regionen der Ansichtensphäre ab.

In einem ersten Schritt wird dazu die Ansichtensphäre in konstanten Schrittweiten diskretisiert. Danach werden alle Ansichten $\boldsymbol{t}_{11}$, ..., $\boldsymbol{t}_{mn}$ aufgenommen und ihre Überdeckungsmengen $S(\boldsymbol{t}_{11})$, ..., $S(\boldsymbol{t}_{mn})$ werden berechnet. Diese Überdeckungsmengen $S(\boldsymbol{t}_{ij})$ sind alle die Ansichten, die eine ausreichend große Ähnlichkeit zu einer Ansicht $\boldsymbol{t}_{ij}$ aufweisen, also: $S(\boldsymbol{t}_{ij}) = \{\boldsymbol{t}_{kl} \,|\, d(\boldsymbol{t}_{ij}, \boldsymbol{t}_{kl}) > d_0\}$

Da aufgrund der Toleranzeigenschaft der Repräsentation eine Ansicht $\boldsymbol{t}_{ij}$ mehr als nur eine Ansicht abdeckt, ist es nicht nötig, alle Ansichten für den Wiedererkennungsprozess abzuspeichern. Es reicht vielmehr aus, nur eine geeignete, kleine Teilmenge zu speichern. Da die Suche nach der kleinsten Teilmenge, die die gesamte Ansichtensphäre abdeckt, ein NP-vollständiges Problem ist, wurden verschiedene Heuristiken untersucht, mit deren Hilfe eine möglichst kleine Menge gefunden werden kann. In [54] wird zusätzlich auch noch ein genetischer Algorithmus vorgestellt, der sehr gute Lösungen liefert. Aber bereits eine einfache Heuristik liefert bei kurzen Laufzeiten eine gute Lösung. Dazu wird aus der Menge aller Ansichten diejenige Ansicht $\boldsymbol{t}_{ij}$ ausgewählt, deren Überdeckungsmenge $S(\boldsymbol{t}_{ij})$ am größten ist. Deren Elemente werden in eine zunächst noch leere Ansichtengrundmenge eingetragen. Im nächsten Schritt wird nun eine weitere Ansicht $\boldsymbol{t}_{kl}$ ausgewählt, deren Überdeckungsmenge $S(\boldsymbol{t}_{kl})$ die meisten der noch fehlenden Elemente der Ansichtengrundmenge erfasst. Auch die Elemente aus $S(\boldsymbol{t}_{kl})$ werden dann zur Grundmenge hinzugefügt. Diese Vorgehensweise wird solange wiederholt, bis in der Grundmenge alle Ansichten der Ansichtensphäre enthalten sind. Die ausgewählten Elemente erfassen dann alle Ansichten eines Objektes, so dass mit dieser Menge an Prototypen eine Erkennung in beliebiger Lage des Objektes möglich ist. Je nach Art des Objektes werden typischerweise zwischen 20 und 60 Prototypen benötigt.

Mit dieser einfachen Heuristik, die mit quadratischem Aufwand eine gute Problemlösung liefert, konnte Dunker in seinen Arbeiten sehr gute Ergebnisse nachweisen [53]. So werden auf einem Datensatz von 3600 Flugzeug-Silhouetten der Typen F-16, F-18, HK-1 (Abb. 4.11), der von Seibert vom MIT Lincoln Laboratory zur Verfügung gestellt wurde, Erkennungsraten von 93.3% (F-16), 95.4% (F-18) und 99.4% (HK-1) erzielt. Dabei wurden zwischen 32 und 67 Prototypen für die einzelnen Flugzeugtypen verwendet.

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

Abb. 4.11: Tolerante Repräsentation einer Flugzeug-Silhouette (a) und Beispiele aus dem Bild-Datensatz (b-d)

4.2.6 Gabor-basierte Konturrepräsentationen

Bei der Erzeugung der zuvor vorgestellten toleranten Kantenrepräsentation wurden wie in [79] beschrieben, verschiedene Verfahren untersucht. Hier soll nun kurz erläutert werden, wie Gabor-Filter zu diesem Zweck verwendet werden können.

Dazu wird ein Satz von Gabor-Filtern definiert, der - wie von Trapp in [176] vorgeschlagen - auf $L = 12$ verschiedene Orientierungen und $M = 3$ verschiedene Auflösungsebenen optimiert wurde. Ein Gabor-Filter ist daher auf eine vorgegebene Orientierung und eine vorgegebene Orts-Frequenzauflösung abgestimmt. Eine detaillierte Beschreibung der Filter befindet sich im Anhang 1. Für die weitere Betrachtung soll nur noch einmal herausgehoben werden, dass die Gaborfilter orientierungsselektiv auf Kantenstrukturen reagieren und ein gewisses lokales Einzugsgebiet besitzen, innerhalb derer eine entsprechend orientierte Bildkante verlaufen muss, um von dem Filter detektiert zu werden.

Analog der formalen Beschreibung einer toleranten Konturrepräsentation, wie in Abschnitt 4.2.2 vorgestellt, lässt sich für jede der M Auflösungsebenen aus den Amplitudenwerten g_{nl} der 12 unterschiedlich orientierten Gaborfilter an jeder Pixelposition n ein Vektor $\boldsymbol{g}_n = [g_{n1}, ...g_{nl}, ... g_{n12}]$ formen. Zusammengefasst ergeben diese einen das Bild beschreibenden Vektor $\boldsymbol{g} = [\boldsymbol{g}_1, ...\boldsymbol{g}_n, ... \boldsymbol{g}_N]$. Durch das große Einzugsgebiet des Gaborfilters wird jedes Konturelement auch in seiner Umgebung detektiert und somit ergibt sich eine tolerante Konturrepräsentation. Im Gegensatz zur anfänglichen Notation ist $\boldsymbol{g}$ jedoch nicht binärwertig. Durch eine einfache Schwellwertbildung mit einer festen

oder auch einer adaptiven Schwelle s kann jedoch auch dieses erreicht werden.

$$c_{nl} = \begin{cases} 1 & \text{wenn} \quad g_{nl} \geq s \\ 0 & \text{sonst} \end{cases} \tag{4.7}$$

Bei dieser Vorgehensweise hat es sich als vorteilhaft erwiesen, für das Lernen anstelle einer wolkenartigen, toleranten Repräsentation eine kettenartige Repräsentation, die eher dem Aktivitätsmuster der simplen Zellen entspricht, zu wählen (vgl. Abb. 4.6 und 4.7). Hierdurch wird beim Erkennungsversuch eine wolkenartige mit einer kettenartigen Repräsentation verglichen. Dies hat den Vorteil, dass sich der Match-Wert über eine gewissen Toleranzbereich hin konstant in der Nähe des Maximums befindet und dass sich außerhalb des Toleranzbereiches sehr kleine Match-Werte einstellen. Der konstant hohe Wert innerhalb des Toleranzbereiches erlaubt eine bessere Trennschärfe unter Beibehaltung der Toleranzeigenschaften [79].

Um aus den Gaborfilter-Antworten eine kettenartige Repräsentation zu gewinnen, wird eine Skelettierung der Filterantworten durchgeführt. Dazu wird ein Fenster ${W^{skel}}_{nl}$ mit der Länge einer Gaborperiode und einer Breite von je einem Pixel Abstand zum Pixel n senkrecht zur Orientierung des Filters g_{nl} an dieser Pixelposition verwendet. Innerhalb dieses Fensters wird nun die Filterantwort aller Filter der Orientierung l ausgewertet. Ist die Filterantwort g_{nl} am zentralen Pixel n maximal, so wird in der schmalen Repräsentation eine eins gesetzt.

$$k_{nl} = \begin{cases} 1 & \text{wenn} \quad g_{nl} > g_{\nu l} \quad \text{für} \quad \nu \in {W^{skel}}_{nl} \\ 0 & \text{sonst} \end{cases} \tag{4.8}$$

Im folgenden Abschnitt soll nun noch näher präzisiert werden, wie die Klassifizierung stattfindet, wenn keine ausreichend gute Segmentation des Objektes vom Hintergrund möglich ist und daher keine Normalisierung der Bildrepräsentation durchgeführt werden kann.

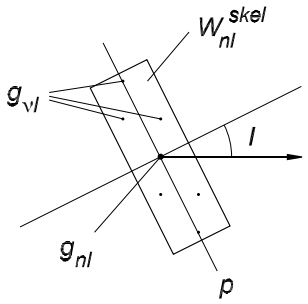


Abb. 4.12: Skelettierung der Gabor Repräsentation

4.2.7 Extrafoveale Erkennung

In den vorhergehenden Abschnitten war erläutert worden, warum sich bei einer Schätzung von Position (x, y), Orientierung ϕ und Entfernung r des Objektes die tolerante Kantenrepräsentation sehr gut für ein Matching-Verfahren mit gelernten kettenartigen Kantenrepräsentationen eignet. In vielen Fällen erlaubt die Arbeitsumgebung eine gute Schätzung dieser Parameter, so z.B. bei Werkstücken, die einem Roboter auf einem homogenen Untergrund zugeführt werden. In Szenarien, in denen eine solche Schätzung jedoch nicht möglich ist, muss die Erkennungsstrategie entsprechend erweitert werden.

Eine bekannte Vorgehensweise, die sich sehr gut für spärlich codierte Repräsentationen wie die hier vorgestellte eignet, ist die sogenannte General-Hough-Transformation [6]. Dabei werden in einem vierdimensionalen Akkumulator, in dem jede Dimension für einen der unbekannten Parameter (x, y, ϕ, r) steht, Hinweise auf die gültige Parameter-Konstellation eingetragen. Ein ausgeprägtes Maximum in diesem Akkumulator deutet dann auf die Existenz des gesuchten Objektes unter den detektierten Parametern hin. Üblicherweise ist jedoch dieses Maximum aufgrund von Bildstörungen oder leichter Variation des Blickpunktes nicht so ausgeprägt, wie erhofft. Aus diesem Grunde muss dann eine Nachbehandlung des Akkumulators erfolgen [17], [142].

Im Gegensatz dazu zeigt die General-Hough-Transformation bei dem Vergleich von toleranten mit schmalen Kantenrepräsentationen deutlich ausgeprägtere Maxima, so dass in unserem Fall auf die Nachbehandlung verzichtet werden kann. Darüber hinaus hat es sich sogar als ausreichend erwiesen, für jede Position (x, y) nur die besten Parameter-Einstellungen für die Orientierung ϕ und die Entfernung r zu speichern. Dadurch reduziert sich der Akkumulator auf ein zweidimensionales Array.

Bei Vorliegen einer ungenauen Schätzung von Position und Entfernung wird die General-Hough-Transformation mit eingeschränkten Suchbereichen $[x - \delta_x, x + \delta_x]$, $[y - \delta_y, y + \delta_y]$ und $[r - \delta_r, r + \delta_r]$ durchgeführt. Die Größe der zu betrachtenden Intervalle hängt dabei von der Genauigkeit der Schätzwerte ab.

Verzichtet man in diesem Fall auf die Bewertung der Nichtübereinstimmung von Kantenelementen und überprüft statt dessen, in wieweit ein gelerntes Muster im präsentierten Bild enthalten ist, so ergibt sich mit der zuletzt gewählten Nomenklatur folgendes Ähnlichkeitsmaß:

$$d(\boldsymbol{c_P}, \boldsymbol{k_L}) = \boldsymbol{c_P} \cdot \boldsymbol{k_L} / l \quad \text{mit } l = |\boldsymbol{k_L}| \tag{4.9}$$

Diese Vorgehensweise zur Objekterkennung und der Bestimmung der Lageparameter (x, y, ϕ, r) des erkannten Objektes hat sich als sehr effektiv und effizient berechenbar erwiesen. So konnten hiermit in schlecht segmentierbaren Szenen für eine Demontage-Applikation das zu demontierende Objekt und die Befestigungsschrauben erkannt werden. Es handelt sich dabei um die Demontage von Rädern an Altautos, auf die in Kapitel 12 noch näher eingegangen wird. Für eine exakte Vermessung der Objekte in Weltkoordinaten wurde dabei eine kalibrierte Stereokamera eingesetzt.

4.2.8 Probleme ganzheitlicher, globaler Abstandsmaße

In den voranstehenden Abschnitten wurde deutlich herausgestellt, wie mit toleranten Repräsentationen ein leistungsstarkes Objekterkennungssystem aufgebaut werden kann. Dabei war die gewählte tolerante Kantenrepräsentation ausschlaggebend, um verschiedene bekannte Probleme von Matchingverfahren zu lösen. So wird hierdurch die To-

leranz bezüglich Schätzfehler der Normalisierungsparameter erzielt, ohne jedoch die Trennschärfe des Klassifikators zu stören. Bei Verwendung der General-Hough-Transformation bilden sich trotz Bildstörungen ausgeprägte Maxima für die korrekten Lageparameter heraus, so dass auf aufwendige Nachbearbeitungen verzichtet werden kann.

Dennoch bleiben zwei generelle Probleme von Objekterkennungssystemen offen. Bei dem vorgestellten holistischen Ansatz, wird ein gelernter Prototyp des gesuchten Objektes als Ganzes mit dem Bildmaterial verglichen und die Ähnlichkeit durch ein global auf dem ganzen Bild berechnetes Ähnlichkeitsmaß beschrieben. Wie bei allen holistischen Ansätzen, ergeben sich Probleme, wenn das Objekt im Bild teilweise verdeckt ist. In diesem Fall wird das Ähnlichkeitsmaß aufgrund partiell fehlender Objektstrukturen unter die für die Erkennung notwendige Schwelle gedrückt. Auch ein Absenken des Schwellwertes kann dieses Problem nicht lösen, da sich hierdurch im allgemeinen die Trennschärfe zu sehr verschlechtert.

Ein ähnliches Problem ergibt sich, wenn Objekte unterschieden werden sollen, die sich nur in einigen wenigen Details voneinander unterscheiden. Auch hierbei kann das Ähnlichkeitsmaß nicht ausdrücken, ob die Unterschiede in genau diesen Details vorliegen. Unter Umständen sind sogar diese Details auflösungsbedingt im Bild gar nicht gut genug sichtbar, um eine Unterscheidung der Objekte hierauf zu stützen. In diesem Fall kann nur eine aktive Erkennungsstrategie, bei der gezielt diese Objektbereiche fixiert und mit längerer Brennweite betrachtet werden, eine robuste Unterscheidung der Objekte leisten.

Ähnliches gilt auch, wenn dreidimensionale Objekte unterschieden werden sollen, die sich in einigen ihrer Ansichten gleichen. Auch in diesem Fall ist eine Unterscheidung nur möglich, wenn aktiv alternative Blickpunkte eingenommen werden.

Die dargestellten Vor- und Nachteile einer ganzheitlichen Erkennung mit Hilfe toleranter Konturrepräsentationen werden im folgenden noch einmal kurz tabellarisch zusammengefasst. In den weiteren Kapiteln wird dann die Integration der ganzheitlichen Erkennung in ein wissensbasiertes Erkennungssystem vorgestellt, mit dem die aufgezeigten Probleme auch durch die Ergänzung um eine Blicksteuerung in einer aktiven Erkennungsstrategie gelöst werden.

Tabelle 1: Vor- und Nachteile der vorgestellten holistischen Erkennung

+	Toleranz bezüglich unvermeidbarer Schätzungenauigkeiten
+	Ausgeprägte Maxima im Akkumulator der Hough-Transformation
+	einfacher Matching-Prozess
+	gute verallgemeinernde Eigenschaften bei gleichzeitigem Erhalt der Trennschärfe
+	unüberwachtes Lernen ohne Trainingssequenzen durch Speichern von Prototypen
-	schlechte Erkennungsleistung bei partiell verdeckten Objekten
-	ungenügende Berücksichtigung kleiner Detailunterschiede der Objekte
-	mangelhafte Unterscheidung von dreidimensionalen Objekten mit zum Teil gleichen Ansichten

5 Hybride Erkennung als Synthese

Nachdem im vorangegangenen Kapitel die Eigenschaften einer holistischen Erkennung besprochen wurden, soll nun erläutert werden, wie die aufgezeigten Probleme durch eine Kopplung der holistischen, neuronal motivierten Erkennung an ein wissensbasiertes System durchgeführt werden kann. Wie in Abschnitt 4.1 aufgezeigt, ermöglicht speziell die wissensbasierte, dekompositorische Objektmodellierung die Erkennung teilverdeckter Objekte, einem der Problemgebiete holistischer Erkenner. Es wurde daher der Ansatz gewählt, auf geeignete Weise eine dekompositorische Objektmodellierung mit einer ganzheitlichen Erkennung zusammenzuführen.

Bevor die gewählte Architektur im Detail besprochen wird, soll jedoch zunächst einmal ein Überblick über hybride Systeme gegeben werden, in denen wissensbasierte und neuronale Ansätze integriert sind.

5.1 Kopplung wissensbasierter Systeme und künstlicher neuronaler Netze - ein thematischer Überblick

Die gegensätzlichen Eigenschaften wissensbasierter Systeme und neuronaler Netze - auf der einen Seite die logische, kognitive und technische Natur wissensbasierter Systeme, auf der anderen Seite die assoziative, selbst-organisierende und biologische Natur neuronaler Netzwerke - haben in jüngster Zeit viele Arbeiten in Bezug auf eine Zusammenführung dieser zwei Paradigmen entstehen lassen. Dabei lassen sich vier Bereiche der Kopplung von wissensbasierten Systemen (WBS) mit künstlichen neuronalen Netzwerken (KNN) - vereinfachend auch als neuronale Netze bezeichnet - feststellen.

1. Künstliche neuronale Netze lernen Wissensbasen
2. Wissensbasierte Systeme steuern die Anwendung von neuronalen Netzen
3. Einsatz von neuronalen Netzen und wissensbasierten Systemen in unterschiedlichen Verarbeitungsschichten
4. Symbolische Verarbeitung durch künstliche neuronale Netze

Diese vier Bereiche sollen im folgenden näher erläutert werden.

Das Lernen von Wissensbasen durch neuronale Netzwerke lässt sich in zwei Teilbereiche unterteilen. Zum einen werden neuronale Netze zum Lernen von Regeln, zum anderen zum Lernen von Grammatiken eingesetzt. Bei beiden werden also zwei Wissensrepräsentationsmechanismen der KI durch neuronale Netze gelernt und ersetzt. Die Vorgehensweise ist in beiden Fällen ähnlich:

1. bekannte Regeln/Grammtiken werden in neuronale Netze übersetzt;
2. das erzeugte neuronale Netz wird durch weitere Beispiele verbessert/erweitert;
3. die im neuronalen Netz enthaltenen, nun modifizierten Regeln/ Grammatiken werden wieder extrahiert.

Die Teilprobleme, die hierbei bearbeitet werden müssen, sind:

- ❒ Übersetzen der Regeln/Grammatiken in künstliche neuronale Netzwerke;
- ❒ Hinzufügen neuer Regeln;
- ❒ Lösen von Regelkonflikten;
- ❒ Vermeidung der Verfälschung korrekter Regeln im Lernprozess;
- ❒ Extraktion modifizierter Regeln.

In den meisten Fällen werden hierzu rekursive neuronale Netzwerke eingesetzt. Untersuchungen und Modelle zum Lernen von Grammatiken finden sich zum Beispiel in den Arbeiten von [44], [66], [141], [161], während einzelne Teilprobleme der Umsetzung von regelbasierten Systemen in neuronale Netzwerke z.B. in [89], [116], [140], [143] behandelt werden. Die in der Literatur vorgestellten Verfahren sind relativ allgemein gehalten und werden typischerweise nicht an Problemen der Bildverarbeitung getestet.

Die skizzierte Vorgehensweise bietet sich immer dann an, wenn das Wissen prinzipiell gut durch Regeln formuliert werden kann, diese jedoch nicht exakt oder nur unvollständig angegeben werden können. Der Einsatz neuronaler Netze unterstützt in diesem Fall den Aufbau der Wissenbasis. Aus Sicht der neuronalen Netze bedeutet dies, dass die bereits vordefinierten Regeln der Wissensbasis den Lernprozess erleichtern und verkürzen.

Im zweiten Teilgebiet der Kopplung von wissensbasierten Systemen und neuronalen Netzwerken werden KI-Methoden genutzt, um die Anwendung der neuronalen Netze zu steuern. Verwendet werden von der KI-Seite hierbei in erster Linie regelbasierte Systeme oder semantische Netzwerke.

- ❒ In regelbasierten Systemen werden neuronale Netze sowohl benutzt, um Prämissen zu bestimmen, als auch um die Konklusion zu beschreiben. Neuronale Netze erkennen hierbei unsichere Daten oder führen unsichere Aktionen aus, wie Meng dies am Beispiel eines bildgestützten autonomen Roboters ausführt [118].

- In semantischen Netzen übernehmen die neuronalen Netze den funktionalen Teil der Attributberechnung. Diese Vorgehensweise wird auch von Kummert und Sagerer gewählt, die aufbauend auf der KI-Systemschale "ERNEST" eine Anbindung an verschiedene neuronale Netze untersuchen (siehe auch Kap. 3.3 “Hybride Systeme” auf Seite 51 sowie [98], [100], [126]).
- Giefing u. Mallot wählen eine etwas andere Kopplung: Sie modellieren Objekte in semantischen Netzen, indem sie Sakkaden zwischen Teilstrukturen des Objektes beschreiben und diese mit den entdeckten Teilstrukturen und ihren Fovealisierungspunkten vergleichen. Hierbei überwiegt jedoch sehr stark der Anteil des neuronal motivierten Teilsystems [65].

Das dritte und vierte Teilgebiet - die Anwendung in unterschiedlichen Verarbeitungsschichten und die symbolische Verarbeitung durch künstliche neuronale Netzwerke - wird nur von relativ wenigen Arbeitsgruppen untersucht. Im dritten Teilgebiet wird dabei von der These ausgegangen, dass bestimmte Probleme sich besser mit neuronalen Netzen, andere besser mit Methoden der KI bearbeiten lassen. Hieraus ergeben sich in Anwendungen üblicherweise Verarbeitungsschichten, in denen die beiden Verfahren bedarfsspezifisch zum Einsatz kommen [90]. Dieser Ansatz kann sich zum Teil mit dem vorherigen vermischen, wenn eine Verarbeitungsschicht die Steuerung der darunter liegenden übernimmt.

Eine Besonderheit stellt das vierte Teilgebiet dar, das von Carpenter und Grossberg vertreten wird. Die diesem Ansatz zugrundeliegende Idee ist die, dass im biologischen System auch die symbolische Verarbeitung durch (biologische) neuronale Netzwerke durchgeführt wird. Carpenter und Grossberg stellen mit *ARTMAP* eine KNN-Systemfamilie vor, welche selbstorganisierend intelligente symbolische Verarbeitung durchführt (vgl. auch Kapitel 3.3). Obwohl diese interne Verarbeitung als ein regelbasiertes System angesehen werden kann, wird es nicht in die 1. Kategorie eingeordnet, da es nicht um das Umsetzen von Regeln in neuronale Netze geht. Vielmehr liegen die Regeln inhärent im Netzwerk vor und wurden durch die Selbstorganisation erzeugt. Carpenter und Grossberg fassen dies auch unter dem vergleichenden Stichwort *Natural Intelligence - Artificial Intelligence* zusammen [36], [37].

5.2 Architekturprinzipien hybrider Systeme

Nachdem ein Überblick über inhaltliche Aspekte hybrider Systeme gegeben wurde, sollen nun noch einige Architekturkonzepte besprochen werden. Zunächst einmal kann zwischen sequentiellen und parallelen Konfigurationen unterschieden werden (siehe auch Abb. 5.1).

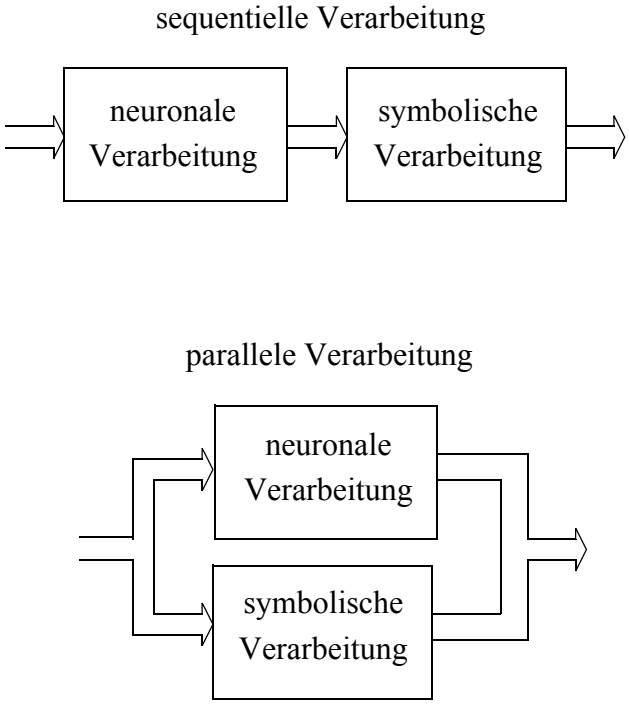


Abb. 5.1: Hybride Konfigurationen (in Anlehnung an [115], S. 81)

❒ **Sequentielle Konfigurationen**

Bei den sequentiellen Konfigurationen erfolgt eine serielle Verarbeitung der Daten, die von einem Modul zum nächsten weitergereicht werden. So wirkt das eine Modul oft als Preprozessor für das darauffolgende, um die Daten in ein für dieses Modul geeignetes Format zu bringen. Dabei kann zum Beispiel ein neuronales Netzwerk Signaldaten so vorverarbeiten, dass sie anschließend für eine symbolische Weiterverarbeitung geeignet sind. In der Bildverarbeitung eignet sich diese Vorgehensweise, wenn dabei die Bildinformation vorstrukturiert wird und gegebenenfalls sogar ein erster Klassifikations-

schritt durchgeführt wird, so dass ein erster Übergang vom Signal zum Symbol geschaffen wurde.

- **Parallele Konfigurationen**
  In einer parallelen Konfiguration arbeiten das subsymbolische und das symbolische Modul gleichzeitig auf den Eingangsdaten. Somit arbeiten sie quasi konkurrierend. In einem Vereinigungsschritt können dann die Ergebnisse der beiden Module miteinander verglichen und auch kombiniert werden. Ein besonders interessanter Aspekt ist hierbei die Rückführung der Ergebnisse vom Ausgang des einen Moduls zum Eingang des anderen Moduls, um von dem jeweils anderen Modul zu lernen.

Bei beiden Konfigurationsarten können natürlich auch mehrere Module miteinander gekoppelt werden. Hierbei können sowohl mehrere - auch verschiedene - neuronale Netze als auch symbolische Verarbeitungseinheiten auftreten. Bei den Kopplungen der einzelnen Module werden dann wiederum verschiedene Kopplungsarten unterschieden. McGarry [115] unterscheidet dabei zwischen drei Kopplungsebenen unterschiedlicher Stärke (siehe auch Abb. 5.2).

- **Passiv gekoppelte Systeme**
  Bei dieser einfachsten Form der Integration kommunizieren mehrere im wesentlichen autonom arbeitende Komponenten typischerweise über Dateien miteinander. Dabei werden lediglich die Ergebnisse in einer Datei abgelegt, die dann von einem anderen Modul gelesen werden kann. Häufig erfolgt dabei der Datenfluss nur in eine Richtung, so dass auf aufwendige Synchronisationsmechanismen verzichtet werden kann. Typische Anwendung ist zum Beispiel die Signalverarbeitung durch ein neuronales Netz, dessen Aktivität am Ausgangslayer dann als Vektor in einer Datei abgelegt wird.

- **Aktiv gekoppelte Systeme**
  Eine engere Kopplung, bei der auch ein bidirektionaler Datenaustausch erfolgen kann, wird häufig über gemeinsame Speicherbereiche realisiert und als *aktive Kopplung* bezeichnet. Hierbei kann dann eine Rückmeldung zwischen den Modulen erfolgen. Dies erfordert natürlich verbesserte Synchronisationsmechanismen zwischen den Modulen, erlaubt dann aber ein schnelleres Reagieren auf Änderungen des Systemzustandes.

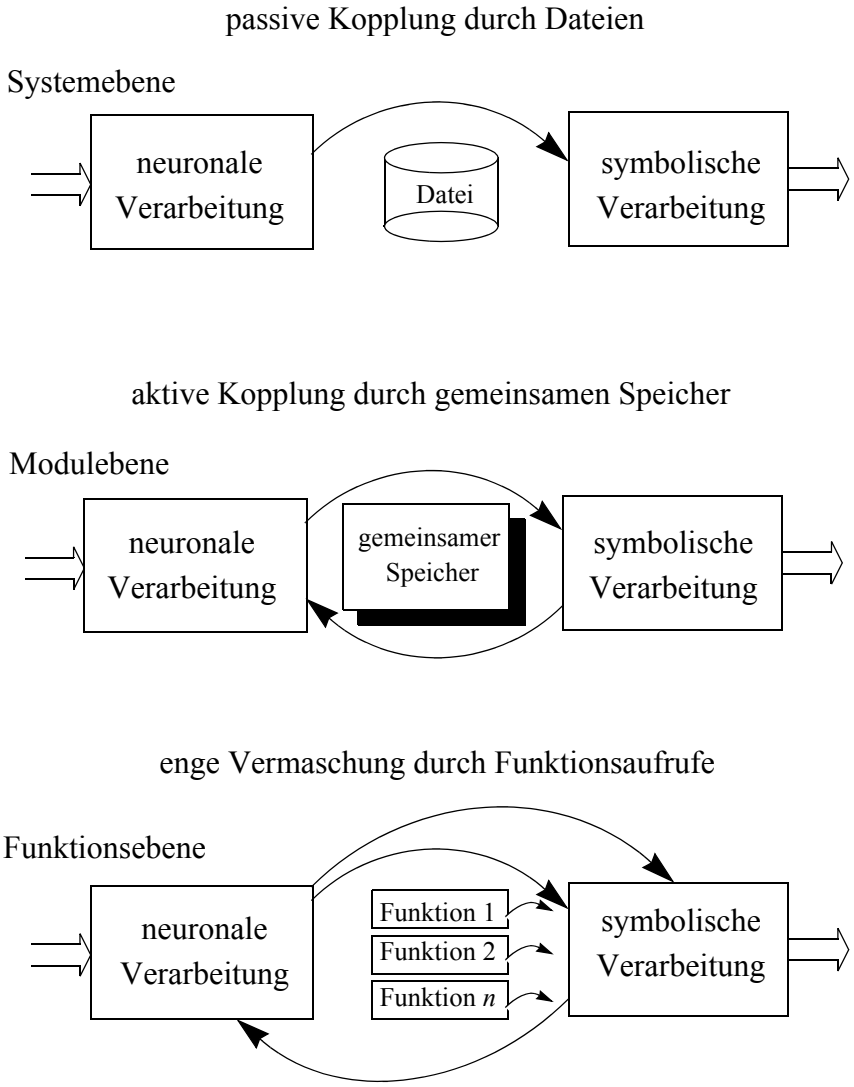


Abb. 5.2: Kopplungsmechanismen in hybriden Systemen (in Anlehnung an [115], S. 82)

Dabei kann zum Beispiel die symbolische Verarbeitungseinheit bei Fehlschlagen der Verarbeitung Rückmeldung an das neuronale Netz geben, um ein Umtrainieren zu veranlassen.

❒ **Eng vermaschte Systeme**

Diese besonders enge Form der Kopplung zwischen den Modulen sieht direkten Zugriff auf einzelne Moduleinheiten vor. Dazu wird ein besonders gut abgestimmtes Kommunikationsprotokoll benötigt, das es erlaubt, dass die verschiedenen Module ihre Arbeitsweise gegenseitig beeinflussen. Dabei können dann zum Beispiel interne Parameter des einen Moduls

durch ein anderes verändert werden. So kann auf diese Weise ein neuronales Netzwerk Elemente der Wissensbasis einer symbolischen Verarbeitungseinheit manipulieren. Oder die symbolische Einheit greift direkt in die Gewichtsmatrix des neuronalen Netzes ein.

Die hier vorgestellte Einteilung der Kommunikationsmechanismen in hybriden Systemen entspricht im wesentlichen auch der Einteilung von Medsker, der dabei von *lose*, *eng* und *vollständig gekoppelten Systemen* spricht [117].

Das in dieser Arbeit im weiteren Verlauf vorgestellte hybride System PAWIAN (**Pa**ralleles **wi**ssensbasiertes Bild**an**alysesystem) fällt unter Verwendung dieses Einteilungsschemas in die Klasse der *aktiv gekoppelten, parallel arbeitenden Systeme*. Dabei obliegt einem wissensbasierten Modul die Ansteuerung verschiedener subsymbolisch arbeitender Module und die Auswertung ihrer Ausgaben. Im wesentlichen sind damit die in Kapitel 4.2 vorgestellten Konturdetektoren und Klassifikatoren gemeint.

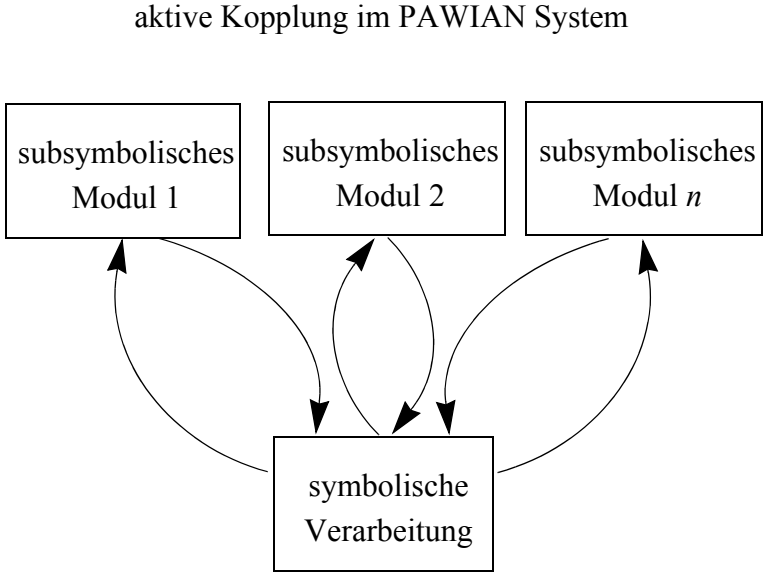


Abb. 5.3: Hybrider Aufbau des PAWIAN Systems

Deutlich wird dabei die Rückkopplung der Module und der Einsatz verschiedener subsymbolischer Module. Dies wird im folgenden Abschnitt noch näher erläutert.

5.3 Die Systemarchitektur des PAWIAN-Systems

Anhand eines Blockschaltbildes in Abbildung 5.4 soll die Systemarchitektur des aktiven hybriden Erkennungssytems beschrieben werden.

Zunächst einmal fallen deutlich die Module des holistisch arbeitenden Erkennungssystems auf, die aus der Bildvorverarbeitung mit der Erzeugung der toleranten Repräsentation, der Normalisierung der Repräsentation bezüglich der geschätzten Lageparameter und der Klassifizierung bestehen. In vielen Fällen kann hiermit die Erkennung eines gesuchten Objektes bereits abgedeckt werden. In schlecht segmentierbaren Fällen oder bei partiell verdeckten Objekten ist jedoch eine robuste Erkennung rein holistisch nicht möglich. In diesem Fall greift ein KI-Modul ein, in dem explizite Objektmodelle abgelegt sind. Es werden nun einzelne Bildbereiche analysiert, oder die Kamera wird aktiv auf bestimmte Objektregionen gerichtet, um Teilstrukturen des gesuchten Objektes zu identifizieren.

Dazu sind die verwendeten Objektmodelle in Form von semantischen Netzen dekompositorisch aufgebaut. Das heißt, es erfolgt im Modell eine Zerlegung des Objektes in wichtige Teilstrukturen. Im Gegensatz zur klassischen dekompositorischen Vorgehensweise, erfolgt diese Zerlegung jedoch nicht in kleinste Bildprimitive sondern in wichtige Teilansichten des Objektes. Jede dieser Teilansichten kann dann wieder mit Hilfe des holistischen Erkenners analysiert und im Bild detektiert werden. Abbildung 5.5 zeigt exemplarisch die Zerlegung eines Objektes in solche Teilansichten.

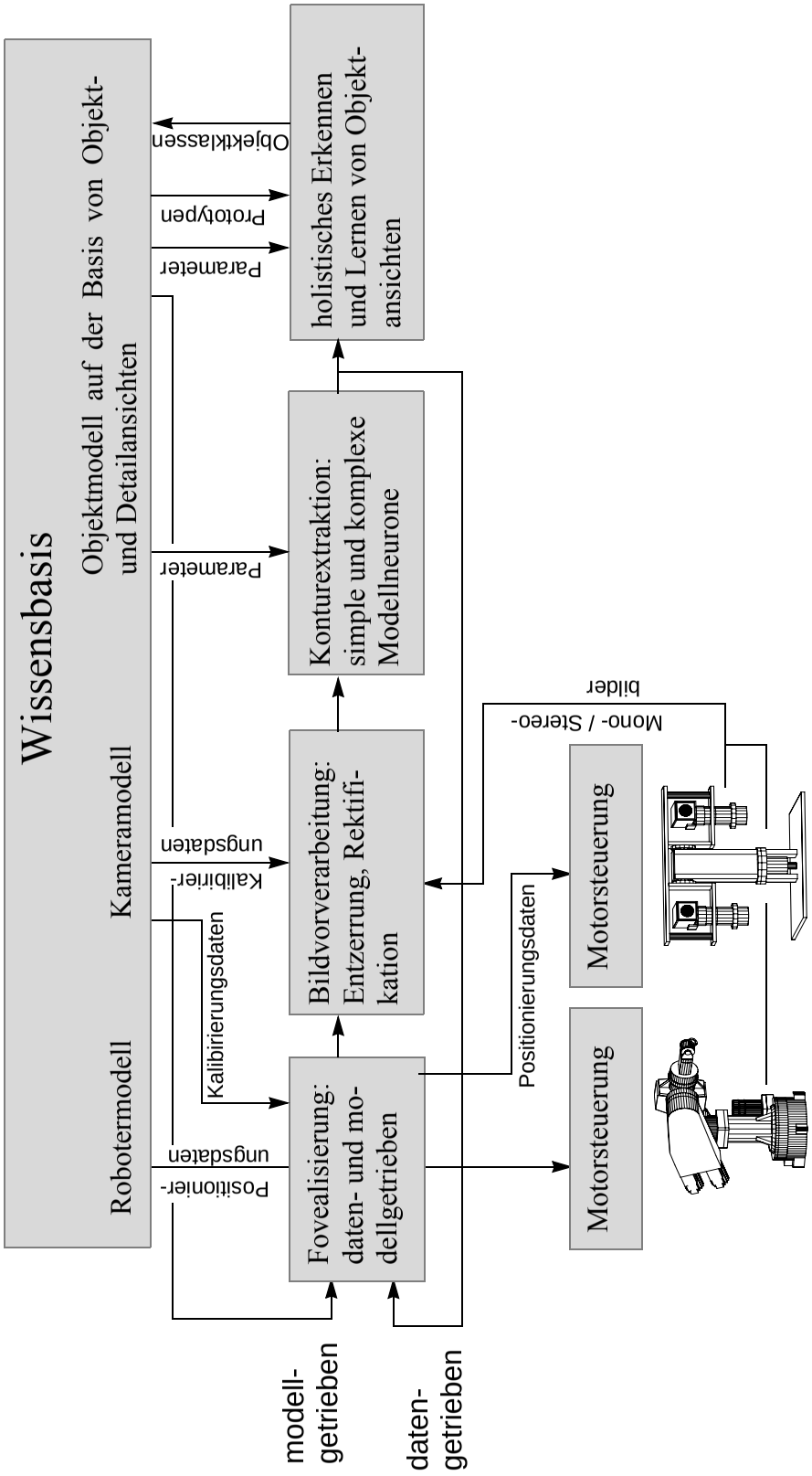


Abb. 5.4: Blockdiagramm des aktiven, hybriden Bilderkennungs-systems

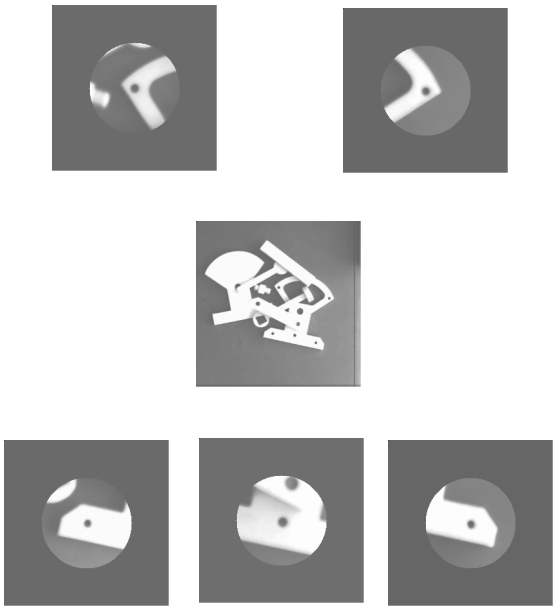


Abb. 5.5: Erkennung eines Objektes anhand einiger Teilansichten

Diese Vorgehensweise hat einige entscheidende Vorteile gegenüber der klassischen Objektmodellierung. So gilt:

- ❒ Teilansichten sind objektspezifisch. Die Erkennung einer Teilansicht gibt Rückschluss auf mögliche Objekte in der Szene, da üblicherweise eine Teilansicht nur Bestandteil weniger verschiedener Objekte sein kann.
- ❒ Teilansichten besitzen Lageinformation. Es kann somit nicht nur auf die Art des Objektes geschlossen werden, sondern auch auf seine Lage in der Szene, wenn im Objektmodell die hierzu notwendige Information abgelegt ist.
- ❒ Durch die Lageinformation besteht direkter Zugriff auf weitere interessante Teilansichten des vermuteten Objektes. Diese können dann gezielt fixiert und ausgewertet werden.
- ❒ Durch diesen direkten Zugriff auf weitere Teilansichten wird die sonst typische kombinatorische Explosion des Suchraums vermieden.

Dieses Prinzip der hybriden Modellierung dreidimensionaler Objekte wurde erstmalig in [27] vorgestellt und in [30], [31] weiterentwikkelt. In den folgenden Kapiteln soll nun noch eingehender beschrieben werden, wie die verwendete Modellierungssprache und der dazugehörende Instanziierungsprozess gestaltet sind.

6 Hybride Objekt-modellierung

6.1 Objektmodellierung mit semantischen Netzwerken

In dem hier vorgestellten System wird eine explizite Objektmodellierung mit *semantischen Netzwerken* verwendet. Wesentlicher Vorteil dieser Modellierungsvariante gegenüber anderen Wissensbeschreibungsverfahren ist die übersichtliche Strukturierung des modellierten Wissens und die gute Visualisierbarkeit der Modelle. Somit besitzt dieser Modellierungsansatz eine hervorragende Ergonomie, Wartbarkeit und Benutzerfreundlichkeit. Aus diesem Grunde sind semantische Netze auch in verschiedenen Erkennungssystemen eingesetzt worden. Dies betrifft sowohl Bild- als auch Spracherkennungssysteme [119], [137], [159], [160]. Einen Überblick über andere Wissensbeschreibungssprachen geben [9], [64], [139], [191].

Der Formalismus der semantischen Netze wurde von Quillian zur Beschreibung der Semantik der englischen Sprache entwickelt [150]. Er versucht dabei, ein menschliches *semantisches* Gedächtnis nachzubilden, in dem zu einem Wort und seiner Bedeutung mit Hilfe *assoziativer* Kanten Verbindungen zu anderen Worten aufgebaut sind. Von seiner Struktur her ist ein semantisches Netzwerk ein markierter gerichteter Graph $G=(V, E)$ mit einer Knotenmenge V und einer Kantenmenge E. Die Knoten des Netzes repräsentieren dabei Informationen über Begriffe der modellierten Domäne, Kanten stellen Beziehungen zwischen den Begriffen dar. Da eine solche Struktur eines semantischen Netzwerkes nicht nur dazu dienen soll, Wissen abzuspeichern, sondern auch inhaltsbezogene, das heißt assoziative Zusammenhänge beschreiben soll, werden sie häufig auch als *assoziative* Netzwerke bezeichnet.

Verschiedene Autoren haben sich seit Quillians Entwurf mit der Weiterentwicklung dieses Ansatzes beschäftigt. Dabei geht es vielfach darum, einen enger gefassten Formalismus zu entwerfen. So ging es Woods beispielsweise darum, eine genauere Trennung zwischen dem in Kanten und dem in Knoten gespeicherten Wissen herauszuarbeiten [193]. Zudem fordert er die Unterscheidung zwischen *intensionalen* und *extensionalen Knoten* eines Netzwerkes. Dabei ist unter der *Intension* eines Begriffes die abstrakte Definition seiner Bedeutung gemeint, während die *Extension* die Menge aller konkreten Sachverhalte umfasst, die dieser Intension genügen ([158], S.42). Brachman führt auf Woods Forderungen aufbauend die Begriffe *Konzept* und *Instanz* ein [18]. Ein Konzept gibt die Intension eines Begriffes an, eine Instanz ein Element aus der extensionalen Menge, welches auch als Ausprägung bezeichnet wird. Weiterführend setzt Brachmann fünf Betrachtungsebenen eines semantischen Netzwerkes ein [19]. Er unterscheidet hierzu zwischen:

- der Implementierungsebene,
- der logischen Ebene,
- der epistemologischen Ebene,
- der konzeptuellen Ebene,
- der linguistischen Ebene.

Diese Betrachtungsebenen lassen sich auch auf die Wissensrepräsentation für ein Bildanalysesystem übertragen, wobei Sagerer hierzu die linguistische Ebene zu einer problemabhängigen intensionalen Beschreibungsebene verallgemeinert ([158], S.47). Die sehr gute Eignung semantischer Netze für die Bilderkennung wurde dann auch von ihm sehr eindrucksvoll mit dem ERNEST-System aufgezeigt. Dieses System wird mittlerweile für eine Vielzahl von Applikationen eingesetzt. Daher setzt auch das hier beschriebene System auf verschiedenen Ideen des ERNEST-Systems auf, wenn es darum geht, Objekte dekompositorisch in einem semantischen Netzwerk zu beschreiben.

Während auf der *Implementierungsebene* die Frage nach den verwendeten Datenstrukturen im Vordergrund steht, betrachtet die *logische* Ebene die formal-logische Bedeutung der auf der *epistemologischen* Ebene festgelegten Beschreibungselemente. Zu diesen zählen z.B. die zur Verfügung stehenden Knotentypen, die Kantentypen, eventuell mögliche Vererbungsmechanismen. Auf der *konzeptuellen* Ebene werden dann die für die Modellierung wichtigen Begriffe und die Beziehungen zwischen ihnen beschrieben, während sich die *intensionale* Ebene mit der zu erwartenden Ausprägung der Begriffe beschäftigt. So muss z.B. sicherlich eine Stadtszene unterschiedlich modelliert werden, wenn sie einmal in Luftbildaufnahmen vorliegt und ein anderesmal aus einem fahrenden PKW betrachtet wird.

Im weiteren sollen einige häufig wiederkehrende Gemeinsamkeiten aufgezeigt werden. Obwohl im allgemeinen mit Hilfe der Kanten beliebige Beziehungen zwischen den Knoten, bzw. den durch sie repräsentierten Begriffen, beschrieben werden können, sind in fast allen semantischen Netzwerken drei verschiedene Kantentypen zu finden (siehe auch [177]):

- *Generalisierung* (*is-a*): Die Generalisierungskante ermöglicht die Gruppierung von Begriffen, die einer gemeinsamen Klasse angehören. Jeder Begriff selbst kann wiederum auch eine Oberklasse für andere Begriffe bilden. Diese Vorgehensweise ermöglicht die implizite Vererbung von Informationen und auch von Beziehungen von Oberklassen auf von ihnen abhängige Unterklassen. Die Umkehrung dieser Kante wird als *Spezialisierung* bezeichnet.

- *Teil-von* (*part-of*): Mit Hilfe der Teil-von-Kanten wird eine weitere Hierarchie definiert, in der Begriffe zu komplexeren Begriffen zusammengefügt werden. Auch hierbei existiert ein Umkehrung der Relation, die *Teilkante* (*has-a*).
- *Individualisierung* (*instance-of*): Dieser Kantentyp dient dazu, ein Individuum mit seinem generischen Typ, seiner Objektklasse, in Verbindung zu setzen. Ein solches Individuum wird dann auch als Ausprägung oder als Instanz der Klasse bezeichnet.

Offensichtlich findet eine Beschränkung auf einige wichtige Kantentypen statt, um effiziente Algorithmen zur Verarbeitung des modellierten Wissens entwickeln zu können. Von besonderer Bedeutung für eine dekompositorische Modellierung der Objekte sind dabei die ersten beiden der oben aufgeführten Kantentypen, die Spezialisierung- und die Teil-von-Kante. Die Individualisierungs-Kante hingegen wird nicht bei der Modellierung verwendet, sondern entsteht erst bei der Bearbeitung des Netzes, in der den Netzknoten Bildstrukturen zugewiesen werden und so Instanzen entstehen. Diese Instanzen werden dann über eine Individualisierungs-Kante mit dem entsprechenden Konzept verknüpft.

6.2 Formale Beschreibung semantischer Netze

Formal kann ein semantisches Netzwerk also als ein gerichteter Graph $G = (V, E)$ mit einer Knotenmenge V und einer Kantemenge E betrachtet werden. Dabei stellt jeder Knoten aus $V = \{v_i \mid i = 1,...,n\}$ ein komplexes Tupel v_i = (Name, Attribute, Bewertung, Ziel) dar, wie es später noch genauer beschrieben wird. Dabei wird in der Knotenmenge V zwischen einer Menge von Konzeptknoten K und Instanzknoten I unterschieden und es gilt somit $V = K \cup I$. Die Kantenmenge $E = \{e_{ij} \mid e_{ij} = (v_i, v_j), i,j = 1,...,n\} \subseteq V \times V$ beschreibt, welche Knoten in Beziehung zueinander stehen. Zusätzlich zu den oben genannten drei Standardrelationen werden im hier beschriebenen System auch noch *Aspektkanten* (*aspect of*) verwendet, um im semantischen Netzwerk verschiedene charakteristische Ansichten eines Objektes zu modellieren. Somit ergibt sich $E = T \cup S \cup A \cup J$ mit

- Teilkanten $T \subseteq K \times K \cup I \times I$,
- Spezialisierungskanten $S \subseteq K \times K \cup I \times I$,
- Aspektkanten $A \subseteq K \times K \cup I \times I$ und
- Instanzkanten $J \subseteq K \times I$.

Darüber hinaus existieren - wie im vorigen Abschnitt bereits angedeutet - auch die entsprechenden Umkehrungen als Kanten im semantischen Netzwerk. Da diese jedoch nur zur Vereinfachung der Algorithmen dienen, werden sie in der folgenden Beschreibung und Diskussion des Systems nicht weiter betrachtet.

Abbildung 6.1 zeigt ein Netzwerk zur Modellierung von Fahrzeugen mit Hilfe der beiden ersten Standardrelationen.

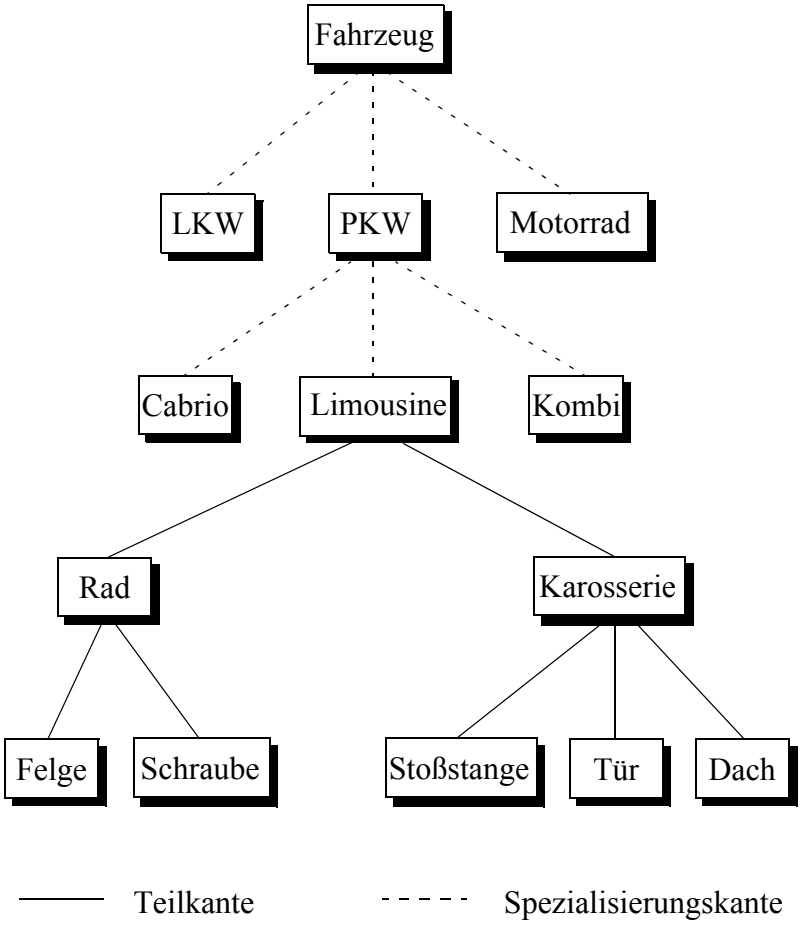


Abb. 6.1: Ausschnitt aus der Modellierung von Fahrzeugen mit Hilfe von Teil- und Spezialisierungskanten

Eine typische Vorgehensweise bei der Bearbeitung eines semantischen Netzes wird in [119] vorgestellt. Dabei wird in einer *Top-Down-*

Expansion zunächst in der Dekompositionshierarchie bis auf die untersten Ebenen herabgestiegen. Die dort modellierten Bildprimitive werden im Bild gesucht, so dass eine Instanziierung dieser Konzepte stattfinden kann. In einem anschließenden *Bottom-Up-Instanziierungs*-Verfahren werden dann die einzelnen Bestandteile zu höher wertigen Beschreibungselemente zusammengesetzt und es erfolgt eine Instanziierung in den oberen Dekompositionsebenen bis hin zum obersten Konzept der Modellierung.

Da üblicherweise mehrere Bildprimitive zur Instanziierung eines Konzeptes herangezogen werden können, entstehen alternativ auszuwertende Instanziierungsmöglichkeiten. Suchverfahren dienen dabei dann zum Auffinden der besten Instanziierung [119], [160] (siehe auch Kap. 7.1).

Da im hier vorgestellten System jedoch von dieser klassischen Modellierung mit Hilfe einfachster Bildprimitive abgewichen wird und stattdessen vielmehr die in Kap. 4.2 vorgestellten holistischen Erkenner eingesetzt werden sollen, wurden gegenüber dem PANTER-System [119] eine Reihe von Änderungen vorgenommen, die in einer ersten Entwicklungsstufe für die Erkennung flacher $2\frac{1}{2}$D-Szenen bereits in [28] vorgestellt wurden. Im folgenden soll nun der aktuelle Stand der Entwicklung für die Auswertung echter 3D-Szenen vorgestellt werden.

6.3 Konzepte und Instanzen

Ziel der expliziten Modellierung ist es, eine Objektbeschreibung auf der Basis von Objektansichten und -teilansichten zu erstellen. Daher ist es notwendig, dass mit den Konzepten beschrieben wird,

- welche Ansichten zur Erkennung herangezogen werden sollen,
- wie die einzelnen Ansichten verarbeitet werden,
- welche topologischen Relationen zwischen den einzelnen Ansichten bestehen.

Es wird daher für jede Ansicht aber auch für alle anderen wichtigen Begriffe der Domäne (z.B. übergeordnete Klassenbeschreibungen) ein

eigenes Konzept im semantischen Netz erstellt. In den Attributen der Konzepte wird dann beschrieben, welche Eigenschaften die einzelnen Ansichten besitzen sollen. Dazu wird für die Attributbeschreibungen eine prozedurale Schnittstelle zur Verfügung gestellt, in der Bildverarbeitungsoperationen eingetragen werden können.

Um nun sowohl die ganzheitliche Erkennung einer Ansicht mit Hilfe der in Kap. 4.2 beschriebenen Verfahren zu ermöglichen, als auch eine dekompositorische Erkennung zu realisieren, werden zwei Attributblöcke verwendet. Im ersten Attributblock wird die holistische Erkennungsstrategie modelliert, während der zweite Block zur Aufnahme topologischer Relationen für die dekompositorische Erkennung dient. Da diese beiden Blöcke zu verschiedenen Zeitpunkten der Bearbeitung des Netzes ausgewertet werden (siehe Kap. 7), werden die Attribute des ersten Blocks als *Pre-Attribute*, die des zweiten Blocks - da sie der herkömmlichen Verwendung in semantischen Netzen für die Bilderkennung entsprechen - als *Attribute* bezeichnet. Hierbei wird bewusst darauf verzichtet, die Attributblöcke anhand ihrer derzeitigen Verwendung z.B. als holistische oder dekompositorische Attribute zu bezeichnen, da die Wissensbeschreibungssprache zunächst einmal unabhängig von der konkreten Anwendung definiert wird.

Neben den Eigenschaften eines Konzeptes ist es aber natürlich auch wichtig, die Standardrelationen zwischen den Konzepten zu beschreiben. Dazu dienen eigenen Slots in den Konzeptframes. Aus Gründen einer verbesserten Übersichtlichkeit der Modellierung wird dabei immer auch die Umkehrung der Relationen explizit mit beschrieben. So ist bei Betrachtung eine Konzepts *A* nicht nur ersichtlich, welche Teile oder Spezialisierungen zu *A* gehören, sondern auch an welche übergeordneten Konzepte das betrachtete Konzept *A* gebunden ist. Darüber hinaus können die Standardrelationen auch attributiert werden. Dies bedeutet, dass im Falle von Mehrfachbestandteilen (Beispiel: ein Auto hat *vier* Räder), die Konzepte nur einmal definiert werden müssen und dann die Relation mit dem entsprechenden Zahlwert attributiert wird.

Zur Steuerung des Suchprozesses und für die Entscheidung, ob ein Konzept erfolgreich instanziiert werden kann, besteht außerdem noch die Möglichkeit, eine Bewertungsfunktion und ein Instanziierungsziel zu modellieren. Auf die Bedeutungen dieser Einträge wird im späteren Abschnitten noch näher eingegangen.

Es ergibt sich also für die Beschreibung eines Konzeptes folgender Aufbau, bei dem mit * gekennzeichnete Elemente mehrfach auftreten können (Abb. 6.2 und 6.3):

Konzept
- Standardrelationen
- Pre-Attribute*
- Attribute*
- Bewertung*
- Ziel

Abb. 6.2: :Konzeptbeschreibung

Standardrelationen
- Generalisierung*
- Spezialisierung*
- Teil-von*
- Teil*
- Aspekt-von
- Aspekt

Abb. 6.3: Standardrelationen

Es sei an dieser Stelle noch vermerkt, dass für die Modellierung eines Objektes natürlich nur Konzepte verwendet werden. Instanzknoten werden erst bei der Bearbeitung des Netzes dynamisch erzeugt. Sie entsprechen dann in ihrer Struktur den Konzepten, wobei aber verschiedene Slots, wie z.B. die Bewertung oder die Ergebnisslots der Bildverarbeitungsoperationen, konkrete Werte enthalten. Daher sind natürlich in einer Konzeptbeschreibung keine Slots zur Modellierung der Instanzrelation vorgesehen.

Der vollständigkeithalber sei ausserdem darauf hingewiesen, dass die Kanten im Netzwerk nicht als eigenständige Datenstruktur organisiert sind. Wie der vorstehend gegebenen Beschreibung entnommen werden kann, ist vielmehr jeder Konzeptbeschreibung eine Adjazenzliste angehängt.

6.4 Die Standardrelationen

Die eingangs erwähnten Standardrelationen, die für die Objektmodellierung verwendet werden, führen offensichtlich zu einer hierarchischen Anordnung der Konzepte und Instanzen im semantischen Netz. Der allgemein übliche Gebrauch der Relationsbezeichnungen *Teil*, *Spezialisierung* und *Aspekt* impliziert bereits eine Ordnung der beteiligten Knoten, die sich dann in einer hierarchischen Anordnung der Knoten mit sehr allgemein gehaltenen Begriffen auf der obersten Ebene der Modellierung und sehr detaillierten Beschreibungselementen auf der untersten Ebene des Modells widerspiegelt.

Da auf dieser Eigenschaft der Modellierung auch der Instanziierungsmechanismus aufsetzt, soll dies hier auch noch einmal formal festgehalten werden.

Es sei $G=(V, E)$ ein semantisches Netz mit den zuvor definierten Relationenmengen T, S, A und J. Für diese formale Betrachtung wird also wiederum auf die Umkehrrelationen verzichtet. Dann gilt:

1. $\forall i = 1,\ldots, n: e_{ii} \notin E$

   d.h.: G ist irreflexiv. Kein Konzept K darf somit zu sich selbst in Relation stehen.

2. $$\begin{aligned} &\forall e_{ij} \in T: e_{ij} \notin S \cup A \cup J \\ \wedge\ &\forall e_{ij} \in S: e_{ij} \notin T \cup A \cup J \\ \wedge\ &\forall e_{ij} \in A: e_{ij} \notin T \cup S \cup J \\ \wedge\ &\forall e_{ij} \in J: e_{ij} \notin T \cup S \cup A \end{aligned}$$

   d.h. wenn zwischen zwei Konzepten A und B eine Relation existiert, so darf zwischen diesen beiden Konzepten keine Relation einer anderen Art bestehen. Es kann also nicht gleichzeitig B ein Teil und eine Spezialisierung von A sein.

3. $\forall e_{ij} \in E: e_{ji} \notin E$

   d.h.: G ist asymmetrisch. Dies bedeutet z.B., dass A nicht Spezialisierung von B sein darf, wenn B bereits Spezialisierung von A ist, wenn B Teil von A ist oder aber wenn B ein Aspekt von A ist.

4. $\forall v_i, v_j \in V: \exists(e_{in_1}, e_{n_1n_2}, \ldots, e_{n_kj}) \in E \Rightarrow$
   $\neg\exists(e_{jm_1}, e_{m_1m_2}, \ldots, e_{m_li}) \in E$

   d.h.: G ist gerichtet zyklenfrei. Hierdurch wird die Bedingung der Asymmetrie noch weiter verschärft, indem verlangt wird, dass auch über längere Wege zwischen zwei Knoten kein Weg in umgekehrter Richtung existieren darf[1].

5. $\exists v_i \in V: \neg\exists e_{ji} \in E$, v_i eindeutig

   d.h.: es existiert genau ein Start- oder Wurzelknoten im Netzwerk. Somit wird ein expliziter Wurzelknoten ausgewiesen, an dem der Instanziierungsprozess des Netzwerkes beginnt.

6. Es sei v_i der Wurzelknoten des Netzwerkes, dann gilt:

   $\forall v_j \in V: \exists(e_{in_1}, e_{n_1n_2}, \ldots, e_{n_kj}) \in E$

   d.h.: vom Wurzelknoten aus existiert zu jedem Knoten v_j ein Weg in E. Diese Bedingung stellt sicher, dass ausgehend vom eindeutigen Wurzelknoten, jedes Konzept im Instanziierungsprozess erreicht werden kann und bedeutet gleichzeitig, dass G zusammenhängend ist.

Desweiteren stellt der Instanziierungsmechanismus sicher, dass gilt:

7. $\forall I_i \in I: \exists k \in K: (k, I_i) \in E$, k eindeutig ,

   d.h.: die Instanzen sind eindeutig einem Konzept zugeordnet

6.5 Die Attributbeschreibungen

Da die Wissensbeschreibungssprache nicht völlig losgelöst von dem dazugehörigen, sie bearbeitenden Kontrollalgorithmus betrachtet werden kann, sei hier ein kurzer Vorgriff auf das folgende Kapitel gestattet. Ziel der wissensbasierten Einheit ist die Erkennung von Strukturen bereits auf Objektebene. Daher ist ein Attributbeschreibungsblock erforderlich, der die hierzu notwendigen Attribute und ihre

1. Ein solcher Weg darf also nur über die nicht in E enthaltenen Umkehrrelationen *Generalisierung*, *Teil-von* und *Aspekt-von* existieren. Dies ist dann trivialerweise ein Weg, der über genau die gleichen Knoten wieder zurückführt.

Operationen aufnehmen kann. Gleichzeitig wird auf Objektebene aber auch ein Attributblock benötigt, der nach erfolgreicher Berechnung der Substrukturen bearbeitet wird und alle notwendigen Attribute beschreibt, die für die Überprüfung der Zugehörigkeit der gefundenen Substrukturen zur übergeordneten Struktur verantwortlich sind. Aus diesem Grund werden auch in der Wissensbeschreibungssprache zwei voneinander getrennte Attributblöcke vorgesehen. Diese Blöcke unterscheiden sich jedoch in ihrem syntaktischen und semantischen Aufbau nicht voneinander. Sie werden lediglich zu unterschiedlichen Zeitpunkten im Instanziierungsprozess bearbeitet und daher namentlich in *Pre-Attribute* und *Attribute* unterschieden. Hierbei dient die Kennung *Pre* dazu, zu verdeutlichen, dass diese Attribute bereits vor der Bearbeitung der Substrukturen ausgewertet werden. Die Beschreibung einer Attributdefinition findet sich in Abbildung 6.4.

(Pre-)Attribut
- Typ
- Wertebereich
- Anzahl
- Operation
  - Operand
  - Operationsgebiet
  - Parameter
  - formales Ergebnis

Abb. 6.4: (Pre-)Attributbeschreibung

Den einzelnen Slots einer Attributbeschreibung kommt dabei folgende Bedeutung zu:

Typ:	Ein Attributwert kann vom Typ String, Boolean, Integer oder Real sein.
Wertebereich:	Entsprechend dem Typ kann der gültige Wertebereich z.B. auf Intervalle oder Listen von Strings weiter eingeschränkt werden.
Anzahl:	Bei mehrdeutigen Operationen besteht die Möglichkeit, dem Attribut eine festgelegte Anzahl von Operationsergebnissen zuzuweisen. Dieser Eintrag ist optional.

Operation:	Über einen symbolischen Bezeichner wird die gewünschte Operation selektiert.
Operand:	Ein oder mehrere zuvor berechnete Operationsergebnisse können als Operanden dienen.
Operationsgebiet:	Die Anwendung der Operation kann auf einen Ausschnitt des Bildes in Form eines Fensters um eine zuvor bearbeitete Struktur herum beschränkt werden. Auf diese Weise kann wissensbasiert der Suchaufwand verringert werden. Dieser Eintrag ist optional.
Parameter:	Zusätzlich zu dem Operanden und dem Operationsgebiet können der Operation noch weitere String-Parameter übergeben werden. Dies sind z.B. applikationsspezifische Schwellwerte, etc. Daher ist auch dieser eintrag optional.
formales Ergebnis:	Dem Operationsergebnis wird ein symbolischer Bezeichner zugewiesen, über den auf das Ergebnis in der weiteren Modellierung zugegriffen werden kann.

Zur Erläuterung sei an dieser Stelle ein typisches Pre-Attribut aufgeführt.

Pre-Attribut	FrontScheibe
Typ	BOOLEAN
Wertebereich	{TRUE}
Operation	SEGMENTATION
Operand	<InitialBild>
Operationsgebiet	<PositionAuto>, r=15
Parameter	Size>20
formales Ergebnis	<FrontScheibeAuto>

Abb. 6.5: Beispiel eines typischen Pre-Attributes

In diesem Beispiel wird beschrieben, dass die Frontscheibe eines Autos durch einen Segmentationsprozess aus einem Initialbild extrahiert werden soll. Als Randbedingungen wird dabei vorgegeben, dass

die Größe eines in Frage kommenden Segmentes mindestens 20 mm im Durchmesser betragen soll. Dieser Wert wird als Schwellwert an die Operation weitergereicht. Zusätzlich wird noch festgelegt, dass bei Vorhandensein einer Hypothese über die Position des Autos lediglich in einem Kreis von 15 mm Radius um den Schwerpunkt eine Untersuchung des Bildes stattfinden muss. Dadurch muss nicht das vollständige Bild in den Segmentationsprozess einbezogen werden. Werden geeignete Segmente gefunden, so stehen diese für eine weitere Verarbeitung unter dem symbolischen Bezeichner *FrontScheibeAuto* zur Verfügung.

Von besonderer Bedeutung für die Verarbeitung der Modellierung ist, dass die in der Attributbeschreibung verwendeten *Operanden* an geeigneter Stelle im Netzwerk als *formale Ergebnisse* definiert sein müssen. Wenn ein einzelner der verwendeten Operanden nicht im Netzwerk gefunden werden kann, so stehen für den Start der Operation nicht alle notwendigen Daten zur Verfügung. Aus diesem Grunde kann dann die Operation nicht aufgerufen werden und somit kein Ergebnis berechnet werden. In Abschnitt 6.8 wird im Detail beschrieben, an welchen Stellen im Netzwerk die Operandenbezeichner gesucht werden.

Eine weitere wichtige Rolle bei der Modellierung kommt dem *Operationsgebiet* zu. Es dient dazu, Suchbereiche im Bild oder in der Szene einzuschränken. So wird bei der Auswahl der Operanden überprüft, ob sie sich innerhalb des Operationsgebietes befinden. Auf diese Weise kann bereits frühzeitig, wenn eine erste Objektstruktur gefunden wurde, der Suchbereich für weitere Strukturen auf eine räumliche Nachbarschaft zum gefundenen Teil eingeschränkt werden.

Eine Einschränkung des Operationsgebietes erlaubt also eine frühzeitige Aussortierung uninteressanter Objekte oder Strukturen. Da jedoch nicht immer davon ausgegangen werden kann, dass eine prägnante Teilstruktur eines Objektes, die ein Operationsgebiet definieren soll, auch tatsächlich sichtbar ist und erkannt werden kann, gilt der Eintrag des Operationsgebietes als optional. Kann also das Operationsgebiet nicht bestimmt werden, so wird das gesamte Bild analysiert. Dies bedeutet, dass durch die Modellierung mit Operationsgebieten eine erhebliche Einschränkung des Suchraums erfolgen kann, ohne dass dies eine zu starke Einschränkung für die Erkennung teilverdeckter Objekte zur Folge hat.

6.6 Die Bewertungsslots

Mit Hilfe der Bewertungsslots kann während der Wissensakquisition festgelegt werden, auf welche Weise die Bewertung der Instanzen eines Konzeptes berechnet werden soll. Prinzipiell besteht hier die Möglichkeit, die Bewertung einzelner Attribute <*Attributname*> oder die Bewertung zuvor berechneter Subkonzepte [*Konzeptname*] zu übernehmen. Weiterhin können Bewertungen auch gewichtet einfließen oder mit den Bewertungen anderer Strukturen verrechnet werden. Die einzelnen Bewertungsslots liefern somit Beiträge für die Güte oder Sicherheit des Erkennungsvorganges. Diese Beiträge werden anschließend wie in Kapitel 7.1 ausführlich beschrieben mit Hilfe der Dempster-Shafer-Evidenztheorie akkumuliert.

Für die Modellierung innerhalb eines Bewertungsslots sind arithmetische Ausdrücke zugelassen, die neben den bekannten arithmetischen Operatoren '+', '-', '*', '/' Zahlen und Verweise auf dem Konzept zugehörige Attribute und Pre-Attribute sowie auf Subkonzepte verwenden. Zur Unterscheidung zwischen Attribut- und Konzeptnamen werden Attribut- (und analog dazu Pre-Attributnamen) in spitzen Klammern <*Attributname*>, Konzeptnamen hingegen in eckigen Klammern [*Konzeptname*] geschrieben. Folgende Regeln beschreiben die Syntax eines Bewertungsslots:

bewertungsslot := BEWERTUNG: term

term := wert | (term) | term operator term

wert := zahl | [*Konzeptname*] | <*Attributname*>

zahl := $\{0,1,2,..,9\}^+$

operator := + | - | * | /

Die Modellierung arithmetischer Ausdrücke erlaubt es dabei z.B. eine gewichtete Mittelwertbildung über verschiedene Attributbewertungen durchzuführen. Typischerweise liefern aber verschiedene Attribute oder Konzepte unabhängig voneinander Hinweise auf die Existenz des gesuchten Objektes. In diesem Fall werden diese in mehreren Bewertungsslots eingetragen. Hierbei erlauben es die arithmetischen Ausdrücke, diese Hinweise ihrer Bedeutung entsprechend zu gewichten.

Somit fließen zum Beispiel sehr charakteristische Teilstrukturen eines Objektes stärker in die Bewertung ein als weniger aussagekräftige Teile. Die von den formerkennenden Operationen (Klassifikator) gelieferten Erkennungsmaße können auf diese Weise gewichtet werden. Zur Bestimmung der Konzeptbewertung werden sie dann mit Hilfe der Dempster-Shafer-Evidenztheorie akkumuliert.

Dies wird durch das Beipiel in Abbildung 6.6 verdeutlicht. Die ganzheitliche Erkennung des Ferrari geht hierbei mit der vollen Bewertung des entsprechenden Attributes in die Bewertung ein, während die Erkennung des Vorderrades und des Hinterrades nur mit jeweils der halben Bewertung der entsprechenden Konzepte hinzugenommen wird.

```
Bewertung       [Vorderrad]/2
Bewertung       [Hinterrad]/2
Bewertung       <KlassifikationFerrari>
```

Abb. 6.6: Verkürztes Beipiel für die Bewertungsslots der Modellierung einer Seitenansicht eines Ferrari mit den Teilstrukturen *Vorderrad* und *Hinterrad*

6.7 Das Zielslot

Das Zielslot hat direkten Einfluss auf die Steuerung der Bearbeitung des Konzeptes und dient dazu, ein Zielattribut oder eine logische Verknüpfung mehrerer Attribute festzulegen. Die Erfüllung dieses Slots, d.h. die erfolgreiche Berechnung der hierin eingetragenen Attribute wird nach der Berechnung der Pre-Attribute überprüft, um in Abhängigkeit hiervon zu entscheiden, ob ein weiteres Herabsteigen in der Modellierungshierarchie erfolgen soll oder nicht. Dies ermöglicht es dem Knowledge Engineer, die Bearbeitung der Attribute zu erzwingen, auch wenn die Pre-Attribute bereits zu einer Erkennung des Objektes geführt haben. Dies ist z.B. dann notwendig, wenn eine detaillierte Vermessung des Objektes für Montagezwecke erfolgen soll und diese aufgrund zu großer Toleranzen nicht am Gesamtobjekt durchgeführt werden kann.

Im Zielslot werden dazu boolsche Ausdrücke mit den logischen Operatoren '&', '|' sowie Vergleiche zwischen zwei Attributen oder einem Attribut und einer konstanten Zeichenkette erwartet. Formal lässt sich dies wie folgt beschreiben:

zielslot := ZIEL: ausdruck

ausdruck := wert operator wert | (ausdruck) |
ausdruck operator ausdruck

wert := <*Attributname*> | zeichenkette

zeichenkette := {0,...9, a,...,z, A,...,Z}$^+$

operator := '&' | '|' | '='

Die Belegung des Zielslots ist optional und nur dann notwendig, wenn eine Modifikation der üblichen Instanziierungsstrategie gewünscht ist. Wird z.B. im Zielslot ein boolscher Ausdruck eingetragen, der stets den Wert *falsch* liefert, so wird hiermit ein Herabsteigen in der Modellierungshierarchie erzwungen auch wenn das Konzept eigentlich aufgrund einer erfolgreichen Bearbeitung der Pre-Attribute instanziiert werden könnte.

Obwohl der Instanziierungsprozess in Kapitel 7 noch eingehender beschrieben wird, soll an dieser Stelle zumindest eine grobe Beschreibung der Auswertung der einzelnen Konzeptslots in Form des nachstehenden Algorithmus gegeben werden.

```
Instanziiere Konzept
    bearbeite Pre-Attribute
    if Bewertung > Qmin and Ziel = true then
        bilde Instanz
    else
        Instanziiere Spezialisierungen, Teile, Aspekte
        bearbeite Attribute
        if Bewertung > Qmin then
            bilde Instanz
```

Algorithmus 6.1: Instanziierung eines Konzeptes

Hierdurch wird noch einmal verdeutlicht, wie die einzelnen Slots der Konzeptbeschreibung ausgewertet werden, um eine Instanz des Konzeptes zu erzeugen. So wird durch Auswertung der Bewertungsslots und des Zielslots entschieden, ob allein durch die in den Pre-Attributen enthaltenen Informationen bereits eine Instanz des Konzeptes erzeugt werden kann oder ob es notwendig ist zunächst noch die Konzepte auf niedrigerer Hierarchieebene der Modellierung zu instanziieren.

6.8 Gültigkeitsbereiche von Bezeichnern

Eine wichtige Bedingung für die Korrektheit einer Modellierung ist neben ihrer syntaktischen auch die semantische Gültigkeit. Hiervon sind im wesentlichen die Verweise auf andere Attribute oder Konzepte betroffen. An dieser Stelle soll nun zunächst einmal exemplarisch eine Attributberechnung betrachtet werden, die typischerweise einen Operanden verwendet, der das Ergebnis einer vorherigen Attributberechnung ist.

Über den in der Modellierung verwendeten Namen könnte innerhalb des gesamten semantischen Netzes auf das benötigte Ergebnis zugegriffen werden. Hierbei ergibt sich jedoch ein die Modularität des Netzes betreffendes Problem. Auch in großen Netzen dürften dann keine Namen mehrfach verwendet werden, um Mehrdeutigkeiten zu verhindern. Dies ist jedoch nur sehr schwierig einzuhalten und führt zu unnötig komplizierten Modellierungen. Aus diesem Grunde wurden die Gültigkeitsbereiche stärker eingeschränkt. So ist ein impliziter Austausch von Attributergebnissen nur möglich zwischen Konzepten, die in direkter hierarchischer Beziehung über Standardrelationen stehen, oder zwischen Konzepten, die auf der übergeordneten Hierarchiestufe gemeinsame Vorgängerkonzepte besitzen. Dabei erfolgt hier eine Einschränkung auf die zuvor modellierten d.h. weiter links stehenden Konzepte.

Wenn es für die Modellierung notwendig sein sollte, auch auf entferntere Ergebnisse zuzugreifen, so können diese durch die spezielle

Operation "TRANS" von einem Konzept zu einem anderen Konzept transferiert werden, um dort dann weiterverarbeitet zu werden. Während Abbildung 6.7 die Gültigkeitsbereiche im Falle einer Berechnung von Pre-Attributen eines Konzeptes A beschreibt, stellt Abbildung 6.8 diese für den Fall der Attributberechnung dar. Gültig für die Suche der Operanden sind jeweils die zu den dargestellten Konzepten gehörigen beschrifteten (Pre-)Attributblöcke.

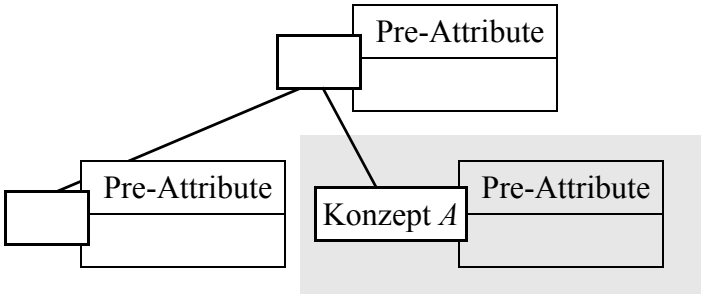


Abb. 6.7: Gültigkeitsbereiche der Bezeichner in Pre-Attributbeschreibungen des Konzeptes *A*

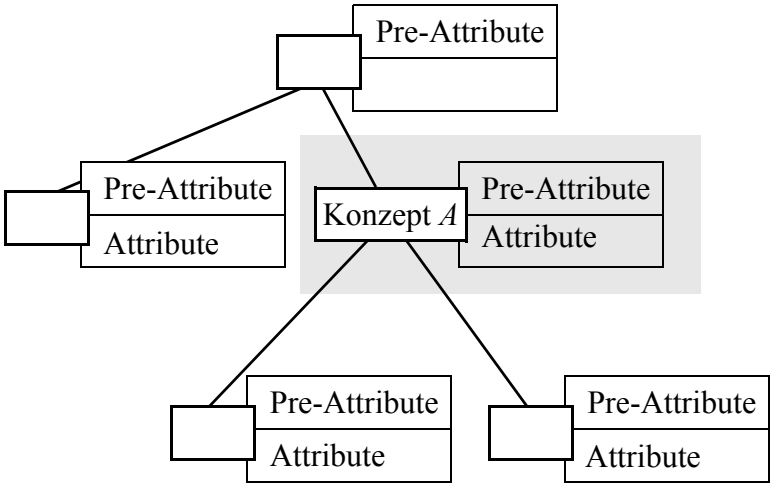


Abb. 6.8: Gültigkeitsbereiche der Bezeichner in Attributbeschreibungen des Konzeptes *A*

Die hier beschriebenen Gültigkeitsregeln für Zugriffe bei der Attributberechnung gelten in gleicher Weise auch für die Bestimmung von Bewertungen. Bei der Auswertung des Zielslots sind die Gültigkeitsbe-

reiche noch weiter eingeschränkt, und zwar auf Attribute des gerade betrachteten Konzeptes.

Formal lassen sich diese Gültigkeitsregeln für Operandenbezeichner wie folgt beschreiben:

Es sei $A^k = (A_1^k,\ldots A_{n_k}^k)$ die Folge der (Pre-)Attribute eines Konzeptes $k \in K$. Es seien o_j^{ki} die Operandenbezeichner in A_i^k, r^{ki} die Ergebnisbezeichner in A_i^k. Dann gilt $\forall o_j^{ki}$ eine der folgenden vier Aussagen.

1. $\exists l \in \{1, \ldots n\}: l < i \wedge o_j^{ki} = r^{kl}$

   d.h.: der Operand ist Ergebnis in einem weiter vorn stehenden Attribut A_l^k des gleichen Konzeptes.

2. $\exists k' \in K, (k', k) \in E \wedge \exists l \in \{1, \ldots n^{k'}\}: o_j^{ki} = r^{k'l}$

   d.h.: der Operand ist Ergebnis in einem Attribut eines Vorgängerkonzeptes k'.

3. $\exists k', k'' \in K, (k'', k') \in E \wedge (k'', k) \in E \wedge \exists l \in \{1, \ldots n^{k'}\}: o_j^{ki} = r^{k'l}$

   d.h. der Operand ist Ergebnis in einem Attribut $A_l^{k'}$ eines Konzeptes mit einem gemeinsamen Vorgängerkonzept k''.

Nur für Attribute (nicht jedoch für Pre-Attribute) besteht noch die folgende Möglichkeit:

4. $\exists k' \in K, (k, k') \in E \wedge \exists l \in \{1, \ldots n^{k'}\}: o_j^{ki} = r^{k'l}$

   d.h.: der Operand ist Ergebnis in einem Attribut eines Nachfolgerkonzeptes k'.

Diese vierte Variante macht bei der Berechnung der Pre-Attribute keinen Sinn, da zum Zeitpunkt der Auswertung der Pre-Attribute noch keine Nachfolgekonzepte bearbeitet worden sind und somit dort auch noch keine Ergebnisse vorliegen können.

6.9 Objektmodellierung

Nachdem die Modellierungssprache nun vollständig vorgestellt wurde, soll ein erstes Beispiel für die Konzeptmodellierung gegeben werden. Dabei muss natürlich zunächst festgelegt werden, welche Bildverarbeitungsoperationen zur Verfügung stehen und verwendet werden sollen. Im hier vorgestellten System besteht die Erkennung einer Objektansicht oder einer Detailansicht typischerweise aus vier Schritten, die sich entsprechend in den Attributbeschreibungen wiederfinden.

1. bestimme Fovealisierungspunkte
2. verfahre die Kamera mit Hilfe des Roboters entsprechend der zuvor ermittelten Fovealisierungspunkte und nimm neue Bilder auf
3. berechne Merkmalsvektoren
4. klassifiziere die Merkmalsvektoren

Da im ersten Schritt üblicherweise mehrere interessante Bildpunkte bestimmt werden können, die genauer analysiert werden sollten, sorgt das Suchverfahren im Instanziierungsprozess dafür, das die Schritte 2 bis 4 für jeden dieser Punkte ausgeführt werden, solange bis eine erfolgreiche Erkennung erfolgt und somit eine Instanz des Konzeptes erzeugt werden kann.

Eine Initialisierung des Systems bei der ein erster globaler Blick in die Szene erfolgt, wird auf der obersten Hierachieebene durchgeführt, so dass dann anschließend von diesem initialen Bild ausgehend die aufgeführten vier Schritte durchgeführt werden können.

Darüber hinaus besteht natürlich auch die Möglichkeit nicht nur datengetrieben Fovealisierungspunkte zu bestimmen. Es können auch modellgetrieben Fovealisierungspunkte erzeugt werden, die dann ebenfalls mit den Schritten 2 bis 4 näher analysiert werden. In Abb. 6.9 wird beispielhaft die Seitenansicht eines Autos modelliert.

Es wird an diesem kleinen Beispiel sicherlich schon deutlich, wie ein Ergebnis bei der Attributberechnung als Operand für eine andere Attributberechnung dient und somit eine typische Bildverarbeitungs-

KONZEPT	Ferrari Seitenansicht
Pre-Attribut	Segment
Typ	BOOLEAN
Wertebereich	{TRUE}
Operation	SEGMENTATION
Operand	<Initial Bild>
Parameter	Size>200
Formales Ergebnis	<PositionSeiteFerrari>
Pre-Attribut	FovBild
Typ	BOOLEAN
Wertebereich	{TRUE}
Operation	MOVE_AND_GRAB
Operand	<PositionSeiteFerrari>
Parameter	Huethres=15,Blurring=1
formales Ergebnis	<SeitenansichtFerrari>
Pre-Attribut	Merkmal
Typ	BOOLEAN
Wertebereich	{TRUE}
Operation	TRANSFORMATION
Operand	<SeitenansichtFerrari>
Parameter	
formales Ergebnis	MerkmalsvektorFerrari>
Pre-Attribut	Klasse
Typ	STRING
Wertebereich	{SeiteFerrari}
Operation	CLAN
Operand	<MerkmalsvektorFerrari>
Parameter	
fomales Ergebnis	<KlassifikationFerrari>

Abb. 6.9: Ein typisches Konzept, bei dem die Pre-Attribute die holistische Erkennung beschreiben, während die Attribute die topologischen Beziehungen zwischen einzelnen Ansichten enthalten.

Attribut	PositionFerrari
Typ	BOOLEAN
Wertebereich	{TRUE}
Operation	CALC_POSITION
Operand	<PositionSeiteFerrari>
Parameter	$\Delta x=0$, $\Delta y=50$, $\Delta z=0$, $\upsilon=90$, $\theta=0$, $\eta=0$, dist=750
formales Ergebnis	<PositionFerrari>
Bewertung	[FerrariVorderrad]/2
Bewertung	[FerrariHinterrad]/2
Bewertung	<KlassifikationFerrari>
Ziel	

END

Abb. 6.9: Ein typisches Konzept, bei dem die Pre-Attribute die holistische Erkennung beschreiben, während die Attribute die topologischen Beziehungen zwischen einzelnen Ansichten enthalten.

kette aus Bildaufnahme, Merkmalsextraktion und Klassifikation entsteht. Am Attribut „PositionFerrari“ sieht man weiterhin, wie auch modellbasiert Fovealisierungspunkte berechnet werden können. Wenn die Seitenansicht des Ferrari erfolgreich erkannt werden konnte, dann besteht natürlich die Möglichkeit, von der entsprechenden Aufnahmeposition ausgehend z.B. eine Nullposition des Autos zu bestimmen und diese dann auch für weitere Analysezwecke zu verwenden. Dies wird in Kapitel 9 noch eingehender beschrieben.

Die hier vorgestellte Modellierung ist offensichtlich eine rein deskriptive Vorgehensweise. Für die Objekte einer Domäne muss festgelegt werden, in welcher Granularität sie modelliert werden sollen, d.h. welche Objektansichten und Detailansichten von Interesse sind und mit welchen Operationen diese Ansichten ausgewertet werden können. Da die in Kapitel 4.2 vorgestellten Methoden zur ansichtenbasierten Erkennung unabhängig von der Domäne eingesetzt werden können, stellt sich also bei der Modellierung nur noch die Frage, in welche Ansichten ein Objekt zerlegt werden soll. Bei allen bislang durchgeführten Objektmodellierungen konnten daher sehr große strukturelle Ähnlichkei-

ten der Objektmodelle festgestellt werden. Diese Beobachtung führte dann auch zur Entwicklung eines Lernmoduls zur automatischen Netzgenerierung (siehe Kapitel 11).

Im folgenden Kapitel wird nun der Instanziierungsprozess, der die Modellbeschreibung abarbeitet, näher beschrieben.

7 Wissensverarbeitung in einem hybriden System

7.1 Instanziierung als Suchprozess

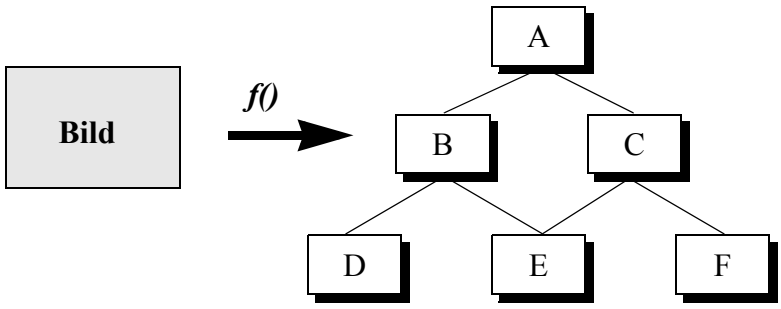
Im letzten Kapitel wurde ausführlich beschrieben, wie eine Modellierung von Objekten mit Hilfe von semantischen Netzen erfolgen kann. Die Objektmodellierung ist aber natürlich nur ein Teil eines Objekterkennungssystems. Im Sinne der in Kapitel 4.1 beschriebenen Trennung von deklarativem und prozeduralem Wissen in einem KI-System muss nun noch das prozedurale Element ergänzt werden. Aufgabe dieses prozeduralen Elementes, das unabhängig von den modellierten Objekten gestalten sein muss, ist es, Signalinformation - die Bildelemente - den symbolischen Beschreibungselementen des Objektmodells zuzuordnen.

Es ist also nun die Aufgabe, ein bestes Mapping der oben beschriebenen Art zu finden. Wie Niemann in [137] beschreibt, kann diese Aufgabenstellung als Suchproblem oder als Optimierungsproblem betrachtet werden. Im ersten Fall wird im Suchraum aller möglichen Mappings eine optimale oder eventuell auch suboptimale Lösung gesucht. Im zweiten Fall wird mit einem initial gewählten Mapping begonnen und dieses in einem Optimierungsverfahren weiter verbessert, bis die gewünschte Lösung erreicht ist. Die zweite Vorgehensweise besitzt den Vorteil, dass zu jeder Zeit ein Lösungsvorschlag für das gestellte Problem vorliegt, auch wenn dieses noch nicht unbedingt die gewünschte Qualität aufweist. Dies ist eine Randbedingung, die vor allem in Systemen mit Echtzeitanforderung von wesentlicher Bedeutung ist.

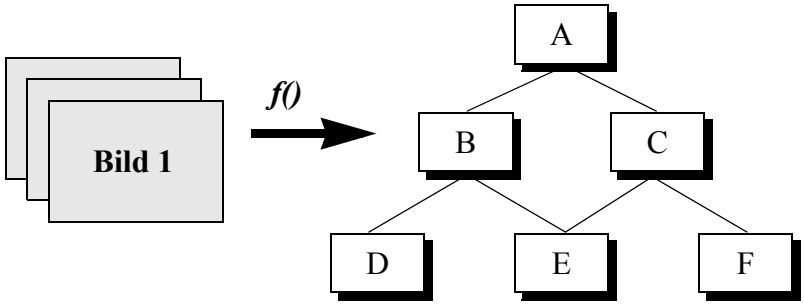
In dem hier vorgestellten System eignet sich diese Vorgehensweise aber nur bedingt, da hier die Problemstellung in einem ganz wesentlichen Punkt von der üblichen Formulierung abweicht. Hier wird ein Mapping gesucht, das nicht nur die Bildinhalte eines Bildes auf die Modellierungselemente abbildet. Es wird ein Mapping gesucht, das darüber hinaus gehend die Abbildungsinhalte mehrerer Bilder, die zu Beginn der Lösungssuche noch gar nicht existieren, auf die Modellstrukturen abbildet. Für die Lösung dieses *active vision* Problems favorisieren wir daher die Suche im Suchraum aller möglichen Mappings. Dazu soll im folgenden zunächst beschrieben werden, wie die Instanziierung eines semantischen Netzes konkret in ein Suchproblem überführt werden kann und welche Suchverfahren zur Lösung herangezogen werden können.

Bevor das erwähnte Suchproblem konkretisiert werden kann, muss zunächst festgelegt werden, wann eine Instanziierung eines Konzeptes durchgeführt werden kann. Im Sinne der hybriden, alternativ ganzheitlichen und dekompositorischen Erkennung, kann dies durch zwei Instanziierungsregeln beschrieben werden.

1. WENN die Berechnung der Pre-Attribute eines Konzeptes k zu einer ausreichend guten Bewertung b führt
   DANN bilde eine Instanz I_k mit den berechneten Pre-Attributen



Auswertung eines Bildes



Auswertung einer Szene durch mehrere Bilder

Abb. 7.1: Die Zuordnung der Bildelemente einer Szene zu einer symbolischen Beschreibung mittels einer Abbildung *f*()

2. WENN Instanzen von Nachfolgerkonzepten existieren und die Berechnung der Attribute eines Konzeptes k zu einer ausreichend guten Bewertung b_k führt
   DANN bilde eine Instanz I_k mit den berechneten Attributen

Diese Instanziierungsstrategie ist auf die holistische Modellierung mit Hilfe der Pre-Attribute und die dekompositorische Modellierung mit Hilfe der Attribute abgestimmt. Sie ist gleichzeitig aber unabhängig von der Art der Objekte, so dass sie - einmal festgelegt und implementiert - für verschiedene Anwendungen allgemeingültig eingesetzt werden kann.

Dabei beschreibt Regel 1 die holistische Erkennung. Ist sie erfolgreich, was durch die modellierte Bewertung festgelegt ist, so kann eine

Instanz des untersuchten Konzeptes gebildet werden. Eine weitere Betrachtung der Nachfolgekonzepte ist dann nicht notwendig.

Regel 2 beschreibt die dekompositorische Erkennung, bei der Teilstrukturen des Objektes erkannt werden konnten. In der anschließenden Attributberechnung wird die Topologie dieser Teilstrukturen auf ihre Gültigkeit überprüft und es kann gegebenenfalls eine Instanziierung erfolgen. Diese zweite Regel führt zu einer rekursiven Bearbeitung des semantischen Netzes in Form einer *TOP-DOWN*-Expansion und einer *BOTTOM-UP*-Instanziierung. Hierbei erfolgt rekursiv ein Abstieg in der Netzhierarchie bis einzelne Details erkannt werden können. Beim Wiederaufsteigen in der Hierarchie durch das Auflösen der Rekursion erfolgt dann die Instanziierung der Konzepte der höheren Hierarchieebenen.

In der Instanziierungsphase wird die modellierte Bewertungsfunktion $b()$ verwendet, um die Qualität einer Instanz zu bestimmen. Je höher die erzielte Bewertung, desto besser „passt“ die Bildstruktur zur Modellbeschreibung. Ziel ist es also, diese Bewertungsfunktion zu maximieren, um in diesem Sinne eine beste Abbildung von Szenenelemente zur Modellbeschreibung zu finden. Typischerweise liefert das Distanzmaß $d(\boldsymbol{t_L}, \boldsymbol{t_P})$ der holistischen Erkennung einen wesentlichen Teil zur Bewertung einer Instanz. Im Falle einer dekompositorischen Erkennung wird darüber hinaus noch die Bewertung der beteiligten Teilkonzepte verwendet, die je nach Bedeutung der einzelnen Teile für die Erkennung gewichtet werden können. Mehrere Bewertungsslots eines Konzeptes werden im Instanziierungsprozess durch einen Dempster-Shafer-Mechanismus akkumuliert.

Dabei werden die einzelnen Bewertungsslots als Sensoren interpretiert, die Hinweise auf die Existenz der gesuchten Instanz geben. Für zwei Evidenzvektoren m_1, m_2 gilt dann nach [166] als Akkumulationsregel:

$$m_3(Z_u) = \frac{\sum\limits_{\substack{i,j \\ X_i \cap Y_j = Z_u}} m_1(X_i) \cdot m_2(Y_j)}{\sum\limits_{\substack{i,j \\ X_i \cap Y_j \neq \varnothing}} m_1(X_i) \cdot m_2(Y_j)} \tag{7.1}$$

Im folgenden Abschnitt soll noch näher erläutert werden, wie sich aus der Instanziierungsstrategie direkt ein Suchbaum herleiten lässt.

7.2 Erzeugung eines Suchbaums

Die zwei im vorigen Abschnitt beschriebenen Instanziierungsregeln legen fest, wann die Instanz eines Konzeptes erzeugt werden kann. Da es das Ziel der Instanziierung ist, die beste Instanz eines vorgegebenen Konzeptes *A* zu finden, werden nun diese beiden Regeln auf *A* angewendet. Eine direkte Instanziierung von *A* ist möglich, wenn die Prämisse von Regel 1 erfüllt ist. Diese verlangt, dass die Pre-Attribute berechnet werden, um dann anhand der Bewertungsfunktion zu entscheiden, ob eine Instanz erzeugt werden kann oder nicht. Regel 1 beschreibt also eine direkte Instanziierung aufgrund der im Konzept modellierten Pre-Attribute. Sie soll daher auch als erstes angewendet werden. Nur beim Fehlschlagen aufgrund einer nicht erfüllten Prämisse soll Regel 2 getestet werden. Hierbei wird untersucht, ob die Nachfolgekonzepte von *A* erfolgreich instanziiert werden konnten und ob mit diesen eine korrekte Attributberechnung durchgeführt werden kann. Dies verlangt einen rekursiven Abstieg in der Modellierungshierarchie zu den Nachfolgekonzepten von *A*. Obwohl die deklarative Modellierung keine Aussagen darüber macht, in welcher Reihenfolge die Nachfolgekonzepte untersucht werden sollen, wird hier intern eine Reihenfolge festgelegt, die der Modellierungsreihenfolge entspricht. In der grafischen Repräsentation ist dies eine Verarbeitung von links nach rechts.

Besitzt nun das Konzept *A*, wie in der Abb. 7.2 gezeigt, zwei Nachfolgekonzepte *B* und *C*, so ergibt sich, dass zunächst die Pre-Attribute von *A* betrachtet werden. Falls keine Instanziierung möglich ist, wird dann zunächst in das Nachfolgekonzept *B* abgestiegen. Hier können nun wieder beide Regeln angewendet werden, so dass also nun die Pre-Attribute von *B* untersucht werden, bevor versucht wird, in die Nachfolgekonzepte von *B* abzusteigen. Dies kann sich rekursiv fortsetzen, bis letztlich die unterste Modellierungsebene erreicht ist. Wenn dann zu einem späteren Zeitpunkt die Bearbeitung von *B* abgeschlossen ist, wird zum Konzept *C* gewechselt. Dieses wird dann in der gleichen Art und Weise behandelt. Somit sind anschließend beide Nachfolgekonzepte von *A* bearbeitet. Unabhängig davon, ob nun eine erfolgreiche Instanziierung von *B* und *C* möglich war, werden dann die Attribute von *A* ausgewertet und es wird eine Bewertung einer möglichen Instanz I_A durchgeführt. Bei Überschreiten des Schwellwertes $Q_{min} = 0.7$ wird eine Instanz von *A* erzeugt.

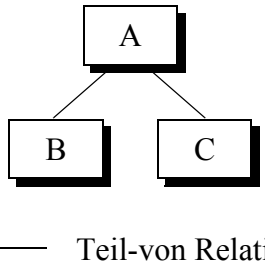


Abb. 7.2: Beispielnetz zur Erläuterung der Instanziierungsregeln

Hierbei sollte noch erwähnt werden, dass bei einem erfolglosen Versuch, die Nachfolgekonzepte *B* und *C* zu instanziieren, typischerweise auch keine Instanziierung von *A* möglich ist. Sinnvollerweise erlaubt jedoch die Modellierung von *A* eine Instanziierung, wenn nicht alle Nachfolgekonzepte erfolgreich instanziiert werden konnten, damit auch bei partieller Verdeckung eines Objektes eine Instanziierung und somit Erkennung des Objektes möglich ist.

Durch die soeben beschriebene Vorgehensweise bei der Auswertung der Instanziierungsregeln ist also eine Verarbeitungsreihenfolge der Konzepte im Netzwerk eindeutig festgelegt. Diese wird lediglich dann verändert, wenn bereits Regel 1 zu einer erfolgreichen Instanziierung führt und somit Regel 2 nicht zur Anwendung kommt. In diesem Fall wird auf die rekursive Bearbeitung eines Teils des Netzes verzichtet. Dies ergibt dann eine Verkürzung der Verarbeitungskette. Abbil-

dung 7.3 zeigt, wie sich aus einem Netzwerk die Verarbeitungskette herleiten lässt und wie sich Verkürzungen auswirken.

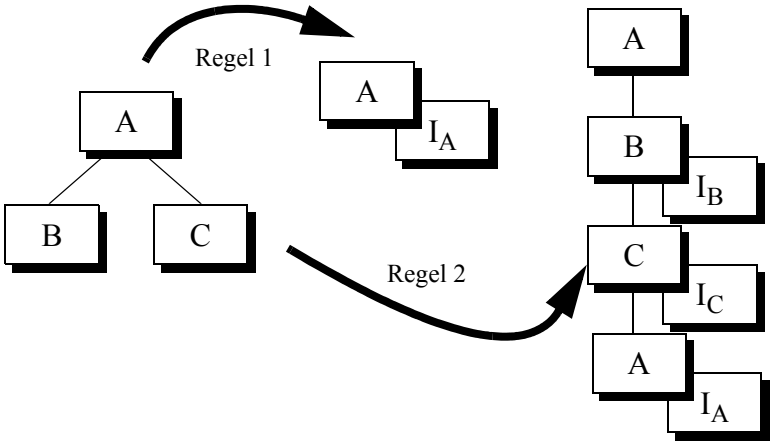


Abb. 7.3: Beispielnetz und seine Bearbeitung

In algorithmischer Form lässt sich die Verarbeitung des Netzwerkes wie in Algorithmus 7.1 dargestellt beschreiben.

Dieses Verarbeitungsschema lässt sich nun direkt in ein Suchproblem überführen. Da üblicherweise mehrere Bildstrukturen für eine Instanziierung zur Verfügung stehen, entsteht aus der Verarbeitungskette eine baumartige Struktur, die auch als Suchbaum interpretiert werden kann. Dies soll an einem kleinen Beispiel verdeutlicht werden. Es soll das Konzept *A* mit seinen Nachfolgekonzepten *B* und *C* instanziiert werden. Dabei sei *A* ein Konzept auf Objektebene, *B* und *C* seien Konzepte auf Teilobjektebene. Befinden sich nun *n* verschiedene Objekte im Bild, so sind zumindest *n* verschiedene Instanziierungen von *A* mit Hilfe der Regel 1 möglich. Weitere Instanziierungen könnten durch Bildrauschen oder Hintergrundstrukturen entstehen. Das gleiche gilt auch für die Konzepte *B* und *C*, für die je nach Art der Objekte mehrere Instanziierungsmöglichkeiten existieren. Wird das Konzept *A* dann mit Hilfe von Regel 2 instanziiert, so potenziert sich die Anzahl möglicher Instanziierungen. Wenn nicht durch geeignete Attributmodellierung eine Einschränkung erfolgt, so ergibt sich durch Regel 2, dass die Anzahl der Instanzen von *B* und *C* zu multiplizieren sind. Im allgemeinen ergibt sich für ein Konzept *A*, die Menge seiner Nachfolgekonzepte N_A und ihre Instanzenmengen *I*:

```
Instanziiere Konzept

  do
      berechne Pre-Attribute

      if Bewertung erreichbar then
          if Ziel erreicht then
              Konzept instanziiert
          else
              instanziiere Spezialisierungen
              instanziiere Aspekte
              instanziiere Teilkonzepte

              berechne Attribute

          if Bewertung erreicht then
              Konzept instanziiert
```

Algorithmus 7.1: Instanziierung eines Konzeptes des semantischen Netzes

$$|I(A)| = \prod_{N_A} |I(N_A)| \tag{7.2}$$

Dieses Problem der großen Anzahl möglicher Instanziierungen wurde bereits in einem früheren Abschnitt als *kombinatorische Explosion* des Suchraums bezeichnet. Abbildung 7.4 zeigt, wie sich durch alternative Instanziierungsmöglichkeiten aus der Verarbeitungskette von Konzepten und ihrer (Pre-)Attribute eine Baumstruktur entwickelt. Zur Vereinfachung sei in diesem Beispiel angenommen, dass zu jedem Konzept *K* ein Pre-Attribut (bezeichnet als *K*-1) und ein Attribut (bezeichnet als *K*-2) definiert sei. Verzweigungen im Suchbaum ergeben sich nun genau dann, wenn zu einer Attributberechnung mehrere Ergebnisse existieren. Dies ist zum Beispiel typischerweise bei Segmentierungen der Fall, wenn ein Bild in mehrere Segmente zerlegt werden kann, die dann alle alternativ, konkurrierend weiterverarbeitet werden müssen. Auf diese Weise entstehen dann konkurrierende Instanziierun-

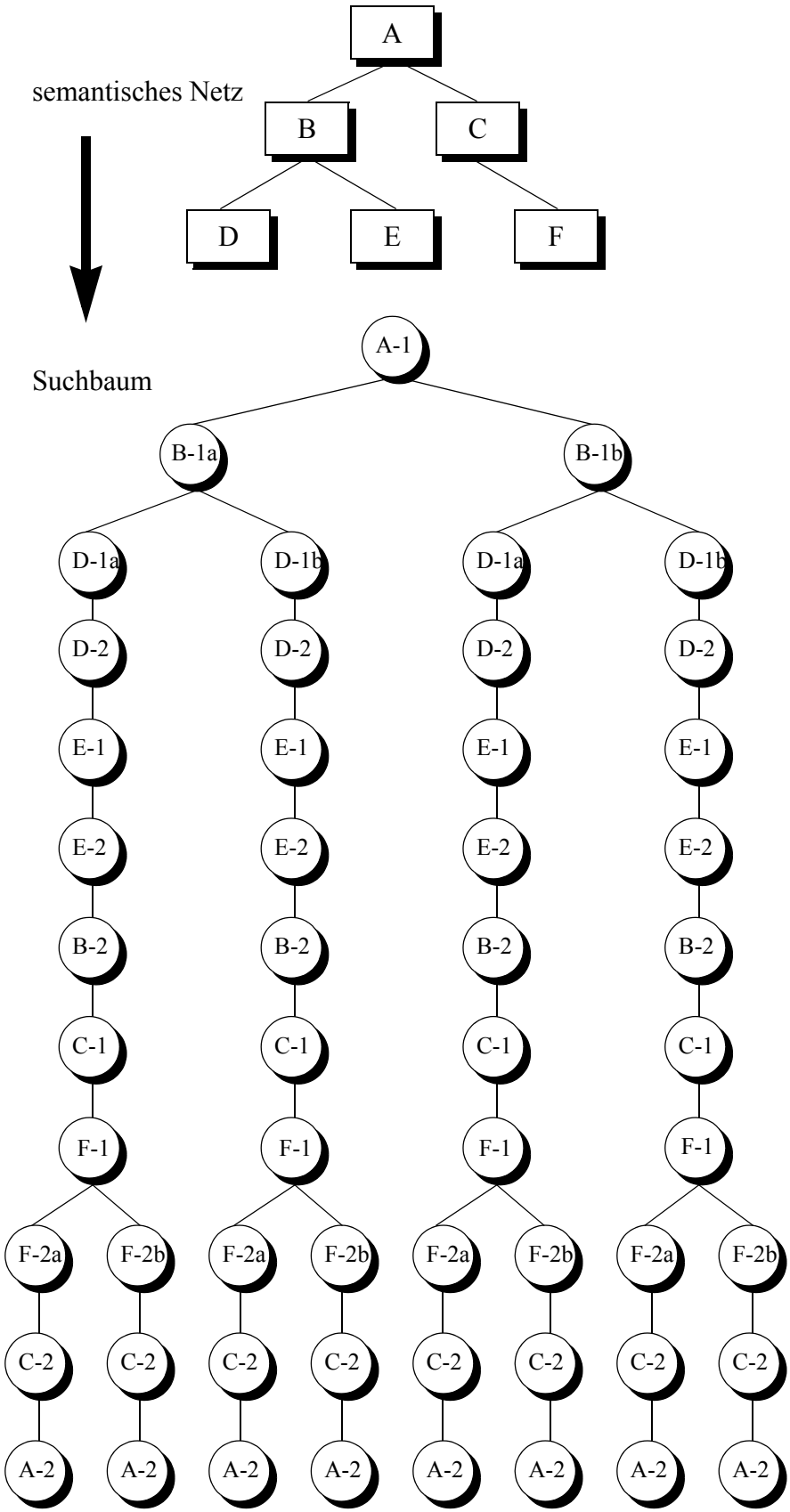


Abb. 7.4: Erzeugung eines Suchbaums bei der Auswertung der Netzbeschreibung

gen und jeder der Pfade von der Wurzel zu einem Blatt des Baumes stellt eine Abbildung von Bildstrukturen zu den Konzepten des semantischen Netzes dar. Dabei unterscheiden sich diese in den Pfaden repräsentierten Abbildungen in ihrer Qualität und die noch zu lösende Aufgabe ist es, möglichst effizient die beste Abbildung zu finden, d.h. den Pfad im Baum zu finden, der entsprechend der Bewertung der Instanzen die beste Zuordnung von Bildstrukturen zu Konzepten darstellt. In diesem Beispiel existieren solche konkurrierende Attributberechnungen für die Attribute *B*-1, *D*-1 und *F*-2, die jeweils mit *a* und *b* bezeichnet werden.

Im Instanziierungsprozess werden somit zwei Klassen von Operationen unterschieden.

Definition 7.1: eine Operation heißt **mehrdeutig**, wenn sie eine Ergebnisliste mit mehr als einem Element liefern kann.

Definition 7.2: Eine Operation heißt **eindeutig**, wenn sie nicht mehrdeutig ist.

Verzweigungen im Suchbaum können also immer dann entstehen, wenn zu einer Attributberechnung eine mehrdeutige Operation verwendet wird.

7.3 Suchverfahren

Im allgemeinen liegt, wie bei dem hier betrachteten Problem auch, die Problemstellung noch nicht in Form eines Suchgraphen vor, der dann lediglich noch nach einem Lösungspfad durchsucht werden müsste. Es ist vielmehr so, dass der Graph nur implizit durch z.B. die Objektmodellierung oder einige gegebene Fakten und Regeln repräsentiert wird. Die explizite Repräsentation wird dann erst während der Bearbeitung des Modells aufgebaut. Es werden also nach und nach die Knoten des Graphen generiert. Man spricht dann auch vom *Expandieren* eines Knotens, wenn alle seine unmittelbaren Nachfolger generiert werden.

Aus Kostengründen (hiermit sind sowohl Zeit- als auch Platzkosten gemeint) sollen dabei so wenige Knoten wie möglich generiert werden und es soll versucht werden, möglichst schnell zur Problemlösung zu gelangen. Für die Steuerung dieser Suche wird heuristisches Wissen eingesetzt, welches üblicherweise in Form einer Bewertungsfunktion *f* bereitgestellt wird. Diese Funktion soll die Kosten vom aktuell bearbeiteten Knoten bis zum Zielknoten abschätzen. Im folgenden sollen einige bekannte Suchverfahren vorgestellt werden. Je nach ihrer prinzipiellen Vorgehensweise werden dabei zwei Klassen von Suchverfahren unterschieden. Verfahren, die das sogenannte *irrevocable control* realisieren, fällen unwiderrufliche Entscheidungen bei der Auswahl eines Nachfolgerknotens, ohne Vorkehrungen für eine spätere Untersuchung von Alternativen zu treffen. Eine explizite Repräsentation des Graphen ist daher nicht notwendig. Diese Verfahren können jedoch nur in wenigen Fällen sinnvoll eingesetzt werden, da sie häufig nicht zu einer Problemlösung führen. Es sollen daher lediglich Verfahren des bekannteren *tentativ control* behandelt werden. Sie entscheiden sich versuchsweise für einen Nachfolgerknoten, treffen dabei jedoch durch entsprechendes Abspeichern der übrigen Knoten Vorkehrungen für die späteren Untersuchungen dieser Alternativen. Die Bewertungsfunktion *f* entscheidet dabei darüber, welcher der Nachfolgerknoten zunächst ausgesucht wird.

7.3.1 Breitensuche

Bei der Breitensuche handelt es sich um ein uninformiertes Verfahren. Sie kommt völlig ohne heuristisches Wissen über die Problemstellung aus. Dabei ist die Bewertungsfunktion *f* so gestaltet, dass jeweils der Knoten mit der geringsten Tiefe für die weitere Verarbeitung ausgesucht wird. Hierdurch erfolgt eine ebenenweise Bearbeitung des Suchbaums.

Da die Knoten einer Ebene die gleiche Tiefe besitzen, ist prinzipiell ein Auswahlkriterium (eine Heuristik) notwendig, welches über die Reihenfolge der Bearbeitung entscheidet. Dies hat jedoch quasi keinen Einfluss auf die Suchstrategien, da ja vor der Bearbeitung der neu generierten Nachfolgerknoten der Tiefe d+1 zunächst immer erst alle

Knoten der Ebene *d* bearbeitet werden. Dem Wurzelknoten wird dabei die Tiefe $d = 0$ zugewiesen.

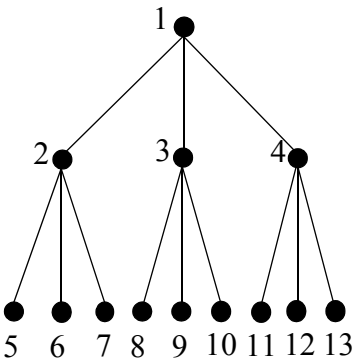


Abb. 7.5: Knotennumerierung bei der Breitensuche

Die Numerierung der Knoten in Abschnitt 7.5 zeigt die Reihenfolge ihrer Bearbeitung bei der Breitensuche. Es ist offensichtlich, dass die Breitensuche immer eine optimale Lösung findet, d.h. eine Lösung mit dem kürzesten Lösungspfad, da die Tiefe der bearbeiteten Knoten mit dem Fortschreiten des Algorithmus monoton wächst.

Für eine Abschätzung der Komplexität des Zeit- und des Speicherbedarfs müssen einige Annahmen getroffen werden. So geht man davon aus, dass der Zeitbedarf proportional zur Anzahl der zu untersuchenden Knoten ist. Der Speicherbedarf soll proportional zur Anzahl der Knoten sein, die gespeichert werden müssen. Die Anzahl der direkten Nachfolger eines Knotens wird vereinfachend als konstant betrachtet und sei *b*. Auf der Ebene *d* befinden sich damit b^d Knoten. Die Länge einer optimalen Lösung sei d_o.

Im ungünstigsten Fall müssen alle Knoten bis zur Tiefe d_o bearbeitet werden, bis eine Lösung gefunden wurde, im günstigsten Fall muss jedoch mindestens ein Knoten der Tiefe d_o bearbeitet werden. Somit ergibt sich für die Anzahl der untersuchten Knoten:

$$\text{im günstigsten Fall: } \sum_{i=0}^{d_0-1} b^i + 1 \tag{7.3}$$

$$\text{im schlechtesten Fall: } \sum_{i=0}^{d_0} b^i \tag{7.4}$$

Für die asymptotische Komplexität des Zeitbedarfs ergibt sich also $O(b^{d_o})$. Bezüglich des Speicherbedarfs ist leicht einsichtig, dass, bevor die Knoten der Ebene d_o bearbeitet werden können, zumindest alle b^{d_o-1} Knoten der Ebene d_o-1 gespeichert werden müssen, um das Prinzip des *tentativ control* zu realisieren. Dies bedeutet ebenfalls exponentielles Wachstum mit einer Komplexität von $O(b^{d_o})$.

7.3.2 Tiefensuche

Eine naheliegende Alternative zur Breitensuche stellt die sogenannte Tiefensuche dar. Wie der Name bereits sagt, wird hierbei ein Pfad bis zu einem noch zu erläuternden Abbruchkriterium in die Tiefe verfolgt. Dies bedeutet also, dass immer ein Knoten, der zuletzt generiert wurde, weiter bearbeitet wird, d.h. der jüngste Knoten wird weiter expandiert. Da auch hierbei mehrere gleichalte Knoten zur Auswahl stehen, kann wiederum eine Heuristik zur Auswahl einer der Knoten beitragen. Im Gegensatz zur Breitensuche ist jedoch in diesem Fall die gute Auswahl von besonderer Bedeutung, bestimmt sie doch in entscheidender Weise die Güte des Suchvorgangs. Abbildung 7.6 zeigt die Reihenfolge der Bearbeitung der Knoten eines Suchbaums im Falle der Tiefensuche.

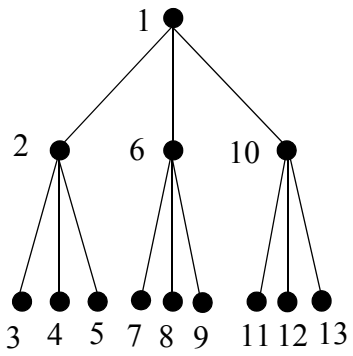


Abb. 7.6: Knotennumerierung bei der Tiefensuche

Da die Heuristik lediglich eine Auswahl zwischen den direkten Nachfolgern eines Knoten trifft, spricht man hierbei auch von einem lokalen Ordnungsprinzip. Global erfolgt somit die Steuerung über die Tiefensuche, lokal entscheidet eine Heuristik über die zu untersuchende Alternative. Über die lokale Ordnung sollte dabei versucht werden, die Suche möglichst frühzeitig in geeignete Teilbäume hineinzuleiten. Dies

ist um so wichtiger, da die Tiefensuche aufgrund des Verfolgens einzelner Pfade bis zu einem gewissen Endekriterium prinzipiell nicht sofort eine optimale Lösung findet. Es besteht im Gegenteil sogar die Gefahr, während der Suche in einen extrem langen (wenn nicht gar unendlich langen) Pfad hineinzulaufen. Aus diesem Grunde werden Bedingungen (*constraints*) eingeführt, die beschreiben, ob ein Pfad noch zu einer optimalen oder wenigstens überhaupt zu einer Lösung führen kann. Dies sollte auch beinhalten, dass eine maximale Suchtiefe vorgegeben wird. Desweiteren muss durch geeignete Vorkehrungen vermieden werden, dass Zyklen entstehen. So sollte z.B. beim Schachspiel eine Stellung, die bereits im Pfad repräsentiert ist, nicht noch einmal aufgenommen werden.

Für die Abschätzung der asymptotischen Komplexität ergibt sich bei Vorgabe einer maximalen Suchtiefe d_{max} ein Speicherbedarf der Ordnung $O(bd_{max})$ und ein Zeitbedarf von $O(b^{d_{max}})$ bei einer blinden Suche, d.h. einer Suche ohne lokales Ordnungsprinzip.

Im günstigsten Fall, also bei Einsatz einer guten Heuristik, kann jedoch eine optimale Lösung der Tiefe d_o bereits in einer Zeit von der Ordnung $O(d_o)$ gefunden werden. Hier sollte noch ergänzt werden, dass die Vorgabe der maximalen Suchtiefe d_{max} auch die Qualität der gefundenen Lösung beeinflusst. Wenn d_{max} auch die Tiefe einer optimalen Lösung ist (also $d_{max} = d_o$), so wird auf jeden Fall eine optimale Lösung gefunden. Wird d_{max} jedoch zu groß gewählt, so werden unter Umständen auch nicht-optimale Lösungen gefunden. Wird hingegen d_{max} zu klein gewählt, werden gar keine Lösungen gefunden.

7.3.3 Bestensuche

Im Zusammenhang mit der Tiefensuche war bereits die Bedeutung eines lokalen Ordnens der zu expandierenden Knoten besprochen worden. Dieses soll nun auf ein globales Ordnungsprinzip erweitert werden. Dabei werden nun nicht mehr nur die gleichalten Knoten, sondern alle noch zu expandierenden Knoten daraufhin untersucht, welcher am vielversprechendsten für die weitere Suche erscheint. Eine Bewertungsfunktion führt hierzu eine Schätzung der zu erwartenden Kosten durch. Der bekannteste in dieser Form aufgebaute Algorithmus ist der

*A**-Algorithmus. An seinem Beispiel soll der Aufbau der Bewertungsfunktion näher erläutert werden.

Die Grundidee ist, die Bewertung eines Pfades vom Wurzelknoten zu einem Zielknoten über den Knoten *n* durch zwei Summanden darzustellen. Der erste beschreibt die Kosten eines optimalen Pfades vom Wurzelknoten zum Knoten *n*, der andere die Kosten von *n* zum Zielknoten.

Es seien:

$g^*(n)$ Kosten eines optimalen Pfades vom Wurzelknoten zum Knoten *n* (optimal im Sinne minimaler Kosten)

$h^*(n)$ Kosten eines optimalen Pfades vom Knoten *n* zu einem Knoten, der der Endebedingung genügt, d.h. einem Knoten, der eine beste Problemlösung repräsentiert.

$f^*(n)$ Kosten eines optimalen Pfades durch den Knoten *n* vom Startknoten zu einem Knoten, der die Endebedingung erfüllt.

Dann gilt: $f^*(n) = g^*(n) + h^*(n)$

Da im allgemeinen die oben geschriebenen Kostenfunktionen nicht im voraus bekannt sind, wird versucht, sie durch eine Schätzung näherungsweise zu beschreiben.

Es seien:

$g(n)$ Kosten des "billigsten" bis zu diesem Zeitpunkt gefundenen Pfades vom Startknoten bis zum Knoten *n*.

$h(n)$ Schätzung von $h^*(n)$ unter Verwendung von heuristischem Domänenwissen. Es gilt $h(n) \geq 0$ für alle Knoten *n*.

$f(n)$ Bewertungsfunktion des Suchverfahrens

Es gilt hierbei: $f(n) = g(n) + h(n)$

Verfahren mit einer derartigen Bewertungsfunktion werden *A** genannt. Einige Autoren wie z.B. Nilsson [139] nennen diese Verfahren *A* und nur dann *A**, wenn *h* eine untere Schranke von *h** ist.

Wenn der zugrunde liegende Graph ein Baum ist, lässt sich die Funktion g sehr leicht berechnen und es gilt gleichzeitig $g = g^*$. Hierzu werden einfach die Kosten des im Suchbaum gespeicherten Pfades vom Startknoten zum Knoten n berechnet. Diese sind im allgemeinen proportional zur Tiefe des Knoten n. Da im Falle eines Baumes nur genau ein Pfad vom Startknoten zum Knoten n existiert, gibt es auch keinen kostengünstigeren Pfad. Somit ist der uns bekannte Pfad gleichzeitig auch der optimale Pfad, dessen Kosten wir als g^* bezeichnet hatten, und es gilt somit $g = g^*$. Für allgemeine Graphen gilt die Abschätzung $g \geq g^*$.

Schwieriger, aber von wichtiger Bedeutung ist die Schätzung des heuristischen Anteils h der Bewertungsfunktion. Hier muss eine für die konkrete Problemstellung möglichst gute Schätzung erfolgen, damit die Bewertungsfunktion f die tatsächlichen Kosten eines optimalen Pfades f^* gut annähert. Man spricht dabei von einer optimistischen Schätzung, falls $h \leq h^*$. Es existieren verschiedene Aussagen über die Eigenschaften des A^*, wenn die Bewertungsfunktion gewissen Bedingungen genügt. Die wichtigsten zwei seien hier dargestellt.

Zulässigkeit

Definition 7.3: Ein Suchverfahren heißt **zulässig**, wenn es immer mit einer optimalen Lösung terminiert, falls eine Lösung existiert.

Satz 7.1: Falls $h(n) \leq h^*(n)$ für alle Knoten n des Graphen gilt, dann ist A^* zulässig.

Informiertheit von Bewertungsfunktionen

Satz 7.2: A_1^* und A_2^* seien zwei Versionen von A^* mit $h_1(n) \leq h^*(n)$ und $h_2(n) \leq h^*(n)$ für alle Knoten $n \in V$ und mit $h_1(n) \leq h_2(n)$ für alle Knoten $n \in V \backslash \{t | t \text{ erfüllt Endebedingung}\}$. Dann gilt: Falls es eine Lösung gibt, so wird bis zur Terminierung jeder von A_2^* expandierte Knoten auch von A_1^* expandiert.

Man sagt auch: A_2* ist *besser informiert* als A_1* und daher folgt: A_2* *dominiert* A_1*.

Die Beweise zu den Sätzen 7.1 und 7.2 sowie einige weitere Aussagen über das Verhalten von A* und anderer Suchverfahren sind ausführlicher in [88] zu finden.

Während Satz 7.1 eine notwendige Bedingung für eine gute Schätzfunktion darstellt, wird durch Satz 7.2 ausgedrückt, dass h andererseits nicht beliebig klein gewählt werden sollte. So hängt also die Anzahl der besuchten Knoten und somit der Zeitaufwand für die Suche unmittelbar von der Qualität der Schätzfunktion ab, die zwar kleiner als h* aber eben doch möglichst nah an h* gewählt werden sollte. Wird einerseits $h = 0$ gewählt, so erhalten wir die Breitensuche, die zwar zulässig ist, aber in jedem Fall ein exponentielles Laufzeitverhalten zeigt. Mit $h = h$* wird andererseits der Idealfall mit einem linearen Laufzeitverhalten erzielt.

Im folgenden ist nun noch in einem Pseudo-Code die allgemeine Form einer Graphsuche wiedergegeben. Dabei geschieht die Suche im Suchbaum gleichzeitig zur Erzeugung desselben. Es werden darüberhinaus auch nur die Teile des Suchbaums erzeugt, die für die Suche benötigt werden. Ausgehend vom Startknoten des Suchbaums wird dieser in eine initial leere Menge der offenen, d.h. noch nicht expandierten Knoten eingetragen. Die Funktion *Elementwahl*() wählt aus der Menge der offenen Knoten den nächsten zu bearbeitenden Knoten aus. Dieser wird expandiert und in die Menge der geschlossenen, d.h. bereits bearbeiteten Knoten eingetragen. Die neu erzeugten Knoten werden zu der Menge der offenen Knoten hinzugefügt und stehen somit für eine weitere Bearbeitung zur Verfügung. Um Mehrfachbearbeitungen einzelner Knoten zu vermeiden, wird dabei überprüft, ob ein Knoten bereits in der Menge der bearbeiteten Knoten enthalten ist. Es sei an dieser Stelle darauf hingewiesen, dass die Funktion *Elementwahl*() die Art der Suche festlegt und die lokalen oder globalen Heuristiken hierfür beinhaltet.

Es stellt sich nun noch die Frage, wie

- der nächste zu expandierende Knoten ausgewählt wird, d.h. wie eine Bewertung der Suchbaumknoten erfolgt, und wie
- das Expandieren der Suchbaumknoten geschieht.

Algorithmus für die Graphsuche

```
function Graphsuche: boolean,
  found:= false
  offen:= {Start}
  geschlossen:= { }
  while not found and offen  ≠  { } do
      N:= Elementwahl (offen)
      offen:= offen \ {N};
      geschlossen:= geschlossen  ∪  {N};
      if Endebedingung (N) then
          found:= true
      else
          S:= expandiere (N);
          offen:= offen  ∪  (S\ geschlossen);
  end while
  Graphsuche:= found
end function
```

Algorithmus 7.2: allgemeine Graphsuche

Daher wird im folgenden Abschnitt erläutert, wie eine für den *A**-Algorithmus geeignete Bewertungsfunktion gewählt werden kann. Abschnitt 7.4 beschreibt dann das Expandieren der Suchbaumknoten, während die daran anschliessenden Abschnitte dieses Kapitels noch einige Möglichkeiten aufzeigen, das Expandieren zu beschleunigen und den Suchbaum zu verkleinern.

7.3.4 Auswahl einer Bewertungsfunktion

Um den *A**-Algorithmus zur Steuerung eines Analyseprozesses verwenden zu können, sind Schätzungen für die Kostenfunktion *f*, *g*, *h* notwendig, die den zuvor beschriebenen Bedingungen einer optimistischen Schätzung genügen.

Bevor eine geeignete Schätzfunktion betrachtet werden kann, muss jedoch zunächst einmal untersucht werden, wie die Kostenfunktion im Falle einer Bildanalyse aussehen kann. Wie im vorherigen Abschnitt an verschiedenen Beispielen gezeigt, ist die Tiefe eines Suchbaumknotens bereits durch die Struktur des semantischen Netzwerks und der verwendeten Strategie einer Top-Down-Expansion und Bottom-Up-Instanziierung gegeben. Eine Schätzung der zu erwartenden Tiefe für einen Lösungsknoten kann daher die Kostenfunktion nicht geeignet approximieren.

Da das Ziel des Analyseprozesses ja das Auffinden einer möglichst guten Zuordnung von Bildstrukturen zur Modellbeschreibung ist, also einer Zuordnung mit optimaler Bewertung, muss offensichtlich auch die Kostenfunktion f des A^*-Algorithmus an diese Bewertungsfunktion gekoppelt sein. Ein optimaler Pfad vom Startknoten zu einem Zielknoten ist dann der Pfad zu einem Blattknoten mit maximaler Bewertung.

Es muss also eine Schätzung der Bewertung durchgeführt werden. Diese kann aus den Bewertungen der Instanzen, die auf dem Pfad vom Startknoten zu einem Knoten K im Suchbaum generiert werden, berechnet werden. Eine hohe Bewertung entspricht dabei niedrigen Kosten und umgekehrt. Knoten auf dem Pfad für die bislang noch keine Instanz erzeugt wurde, werden mit einer maximalen Bewertung von 1,0 geschätzt, um die Monotoniebedingung und somit die Zulässigkeit des Verfahrens zu gewährleisten. Geschätzt wird jeweils die erreichbare Bewertung des Zielkonzeptes. Somit verringert sich diese bei Fortschreiten auf dem Pfade, da maximale Bewertungen durch tatsächlich erzielte Instanzbewertungen ersetzt werden.

Betrachtet man die Kostenfunktion als Umkehrung der Bewertung, so entspricht die Auswahl eines Knoten mit minimal geschätzten Kosten der Auswahl eines Knoten mit maximal geschätzter Bewertung.

Im vorherigen Abschnitt war bereits darauf hingewiesen worden, dass es im allgemeinen schwierig ist, eine gute Schätzfunktion h zu finden. Bei der in dieser Arbeit vorgeschlagenen Modellierung auf der Basis von Objektansichten und -teilansichten können wir uns bei der Su-

che jedoch gezielt die vorgestellten Vorteile des hybriden Systems zunutze machen. Dies sind insbesondere die Tatsache, dass

- ❒ Teilansichten objektspezifisch sind und somit nur als Bestandteil eines oder weniger Objekte auftreten und dass
- ❒ Teilansichten eine Lageinformation besitzen und es somit erlauben, direkt Hypothesen über die Lage des Objektes und anderer Teilansichten zu erstellen.

Wird also, die Schätzfunktion an die Bewertung von Instanzen gekoppelt, so wird im allgemeinen eine Schätzung sehr nahe an $h^*()$ erreicht und somit ein nahezu lineares Laufzeitverhalten erzielt. Dies wird an den Anwendungsbeispielen in Kapitel 12 noch näher untersucht.

7.4 Expandieren der Suchbaumknoten

Für das eingangs erwähnte Suchen einer besten Instanziierung des semantischen Netzwerkes muss im Suchbaum der Pfad gefunden werden, der diese beste Instanziierung repräsentiert. Dazu müssen die im Suchbaum enthaltenen Knoten näher analysiert werden. Jeder einzelne Knoten ist an eine (Pre-)Attributbeschreibung gebunden, die wie im vorangegangenen Kapitel beschrieben verschiedenen Slots enthält. Da diese Attributbeschreibung einen prozeduralen Anteil (die *Operation*) enthält, beinhaltet das Expandieren eines Suchbaumknotens das Ausführen dieser Operation, die üblicherweise eine Bildverarbeitungsfunktion ist.

Dazu werden zunächst die benötigten Operanden innerhalb der Knoten auf dem Pfades vom Wurzelknoten des Suchbaumes zum bearbeiteten Suchbaumknoten gesucht. Diese Suche beschränkt sich jedoch aus Gründen der Übersichtlichkeit und der besseren Modellierung wie in Abschnitt 6.8 erläutert auf ein lokales Umfeld des zu bearbeitenden Konzeptes.

Nach der Bestimmung der Operanden kann dann unter Verwendung des beschriebenen Operationsgebietes und eventuell zusätzlich definierter Parameter aus der Attributbeschreibung die angegebene Operation aufgerufen werden und eines oder mehrere potentielle Attributergebnisse berechnet werden. Nach einer Überprüfung bezüglich des festgelegten Wertebereiches erfolgt eine Bewertung der Ergebnisse durch Verknüpfen von minimaler Operandenbewertung mit einem von der Operation bestimmten Gütemaß (Gleichung 7.5).

$$bew(A) = min_i\{bew(Operand_i)\} \cdot bew(Operand)/100 \quad (7.5)$$

Dies ist eine besonders im Bereich regelbasierter Systeme häufig genutzte Vorgehensweise zur Bewertung neu geschlossenen Wissens. Dabei entsprechen dann die Operanden der Prämisse einer Regel und die Operation der Regel selbst, der vom Knowledge Engineer eine Bewertung zugeordnet wurde. In unserem Fall ist jedoch diese Bewertung nicht statisch festgelegt, sondern wird von der Operation während der Berechnung bestimmt. Die Bewertungen von Operationen und damit auch die der Attribute sind auf einen Wertebereich zwischen 0 und 100 normiert.

Im Falle einer eindeutigen Operation wird das bewertete Operationsergebnis dem Attribut zugewiesen und es kann zur Bearbeitung des nächsten Knotens übergegangen werden. Wenn es sich bei der ausgeführten Operation um eine mehrdeutige Operation handelt und diese eine Liste von Operationsergebnissen liefert, so wird für jedes Element der Liste eine Kopie des Suchbaumknotens angelegt und mit jeweils einem Element der Liste versehen. Hierdurch entstehen dann die bereits erwähnten Verzweigungen im Suchbaum.

Nach der Bearbeitung eines Attributes wird zum nächsten Knoten übergegangen, bis ein Attributblock eines Konzeptes vollständig bearbeitet ist. Ist dies der Fall, so werden die Bewertungsslots des Konzeptes analysiert - im Falle der Pre-Attribute erfolgt hierbei eine vorläufige Bestimmung der bis dahin erreichten Bewertung, wobei bislang noch nicht ermittelte Attribute und Teilkonzepte, auf die in den Slots zugegriffen wird, nicht verwendet werden. Wenn eine ausreichend gute Bewertung erreicht wurde, so wird die Analyse der Attribute eines Konzeptes zunächst abgeschlossen und in der Konzeptinstanziierung fortgefahren. Wurde eine solche Bewertung jedoch nicht erreicht, so ist

es nötig, eine andere Belegung der Attribute zu untersuchen. Dazu wird im Suchbaum wieder nach oben steigend überprüft, ob an einem Knoten eine alternative Attributbelegung möglich ist. Wird solch ein Knoten gefunden, so kann an dieser Stelle wieder nach unten gewandert werden. Dieser Vorgang wiederholt sich, bis eine ausreichend gut bewertete Attributbelegung gefunden wurde. Somit werden also die zu einem Konzept gehörenden Attribute immer als ein Block behandelt, der auch als *Attributgraph* bezeichnet wird, an dessen Bearbeitung sich ein Instanziierungsversuch anschließt. Dieser Versuch ist erfolgreich, wenn die Bewertung über der gefundenen Schwelle Q_{min} liegt. In diesem Fall wird dann ein Element der Datenstruktur *Instanz* erzeugt und mit den dazugehörigen berechneten Attributen verbunden. Über die Instanzrelation wird zudem die Instanz an das dazugehörige Konzept gebunden. Darüber hinaus werden auch die Standardrelationen zwischen den entstehenden Instanzen durch entsprechende Verzeigerungen repräsentiert. In Abbildung 7.7 wird ein Ausschnitt des Suchbaumes aus Abbildung 7.4 aufgegriffen und die Instanzbildung dargestellt.

Man kann der Abbildung entnehmen, dass in diesem Beispiel immer beide Attributblöcke bearbeitet werden mussten, bevor eine Instanziierung zustande kam.

Das Zusammenfassen der Attribute eine Konzeptes und deren blockweises Bearbeiten entspricht nicht der ursprünglichen Idee der A*-Suche, die ja nach jeder Knotenexpansion alle offenen Knoten des Netzwerkes in die nächste Auswahl einbezieht. Sie hat sich jedoch als sehr sinnvoll erwiesen, da die Bewertung der Knoten ja unmittelbar mit der Bewertung von erzeugten Instanzen zusammenhängt. Diese können jedoch erst bewertet werden, wenn die entsprechenden Attribute vollständig bearbeitet wurden. Andernfalls müssten für die Instanz zu erwartende Bewertungen geschätzt werden. Gute Schätzungen sind hierbei jedoch sehr schwierig und erst nach der Klassifizierung einer Bildstruktur möglich. Damit ist dann aber typischerweise auch bereits ein Attributblock vollständig ausgewertet. Somit wird also von der ursprünglichen Idee des *A** zugunsten eines verringerten Aufwandes bei der Kostenschätzung abgewichen.

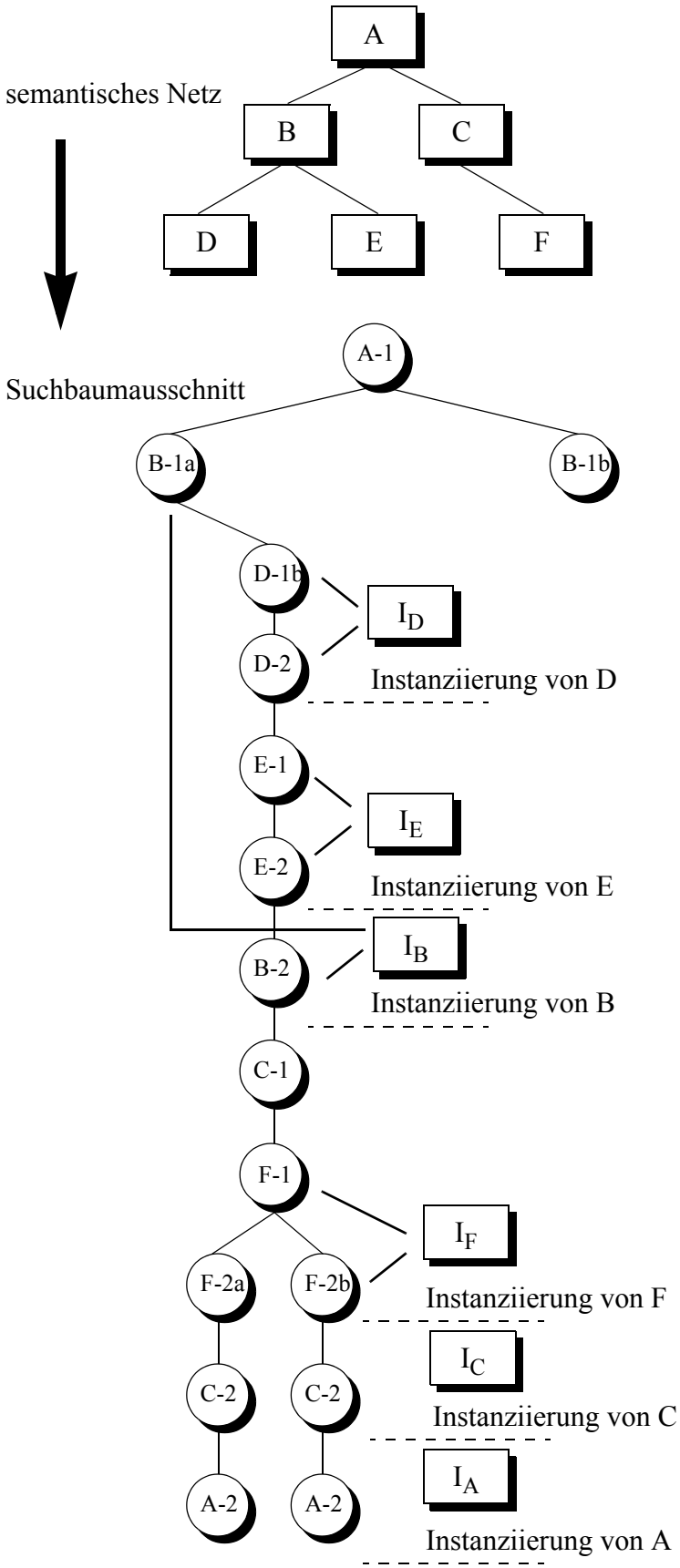


Abb. 7.7: Instanziierung von Konzepten beim Bearbeiten des Suchbaumes

Diese Vorgehensweise ist im folgenden in algorithmischer Schreibweise dargestellt (Algorithmen 7.3 und 7.4).

```
Attributgraphauswertung

begin
  if Blattknoten erreicht then
       schätze die erreichbare Bewertung
       if Bewertung ausreichend gut then
            return success
       else
            return failure
  else
       bearbeite aktuellen Attributgraphknoten

       Attributgraphauswertung des Restgraphen

       while nicht erfolgreich do
            ermittele alternativen Wert für den aktuellen Knoten

       if nicht erfolgreich then
            return failure

       Attributgraphauswertung des Restgraphen

end

Knoten, die nicht erfolgreich berechnet werden konnten, werden
markiert und dienen für spätere Aufsetzpunkte.
```

Algorithmus 7.3: Attributgraphauswertung

Es ist offensichtlich, dass mit zunehmender Komplexität der Szenen und der Modelle der Suchaufwand exponentiell ansteigt. Die hier vorgestellte modellgesteuerte Suche im Suchbaum wird daher durch verschiedene, auf die Problematik der Bildauswertung bezogene Regeln ergänzt, um diesen Suchaufwand so klein wie möglich zu halten. Einen wichtigen Punkt bildet dabei die Verwaltung der mehrdeutigen

```
Berechne Attribute

begin
  while  nicht beste Instanz gefunden and
         nicht zwecklos do

      Attributgraphauswertung
      if erfolgreich then
         if nicht alle Attribute erfolgreich berechnet then
            Speichere aktuellen Attributgraphen
         else
            beste Instanz gefunden = true
      else
         zwecklos = true

  if nicht beste Instanz gefunden then
      benutze den besten der abgespeicherten Attributgraphen
end
```

Algorithmus 7.4: Attributberechnung

Operationsergebnisse, also im wesentlichen die Verwaltung der zu untersuchenden Bildstrukturen. Hierauf wird im folgenden Abschnitt näher eingegangen.

7.5 Die Verwaltung von Bildstrukturen

Die vom Kontrollalgorithmus durchzuführende Zuordnung von Bildstrukturen zu Konzeptinstanzen kann durch einige einschränkende Regeln vereinfacht werden. Dabei können diese direkt in den Algorithmus einfließen, um seine Effizienz zu verbessern und um den Knowledge Engineer von einer immer wieder neu durchzuführenden Modellierung dieser Regeln zu entlasten. Die verwendeten Verfahren sollen im folgenden erläutert werden.

1. Eine Bildstruktur S gilt so lange als belegt, wie eine in den Suchbaum eingebundene Instanz I eines Konzeptes K existiert, in deren Attributen $A_1,...,A_n$ diese Bildstruktur verwendet wird.

Über die Lebensdauer einer solchen Instanz I kann weitergehend mit folgender Regel eine Aussage gemacht werden:

2. Eine in den Suchbaum eingebundene Instanz I eines Konzeptes K_1 gilt so lange als gültig, bis sie für das Scheitern der notwendigen Instanziierung eines anderen Konzeptes K_2 verantwortlich gemacht wird.

Für die Verwaltung der Bildstrukturen wird ein Metaspeicher verwendet, in dem zunächst alle Bildstrukturen eingetragen werden, die vom Segmentationsverfahren geliefert werden. Die erste dieser Strukturen wird dem bearbeiteten Attribut als formales Ergebnis zugewiesen und im Metaspeicher als benutzt markiert. Werden dann in der weiteren Bearbeitung des semantischen Netzes zusätzliche Strukturen benötigt, so erfolgt zuerst eine Suche innerhalb des Metaspeichers. Dabei wird aufgrund von Regel 1 eine zur Zeit nicht benutzte Bildstruktur als Ergebnis geliefert. Die Freisetzung der Bildstrukturen erfolgt, wenn Instanzen nicht zu einem gültigen Ergebnis geführt haben und wieder aus dem Suchbaum entfernt werden (Regel 2).

Die hier beschriebene eindeutige Zuordnung von Bildstrukturen zu Instanzen verhindert jedoch, in einer Ansammlung sich verdeckender Objekte alle hieran beteiligten identifizieren zu können. Diese bilden nämlich unter Umständen eine zusammenhängende Bildstruktur, welche als bearbeitet gilt, sobald sie aufgrund der Detailanalyse einer Instanz zugeordnet werden kann. Eine entsprechende Ergänzung der Regeln ist daher notwendig. Folgende Vorgehensweise bietet sich hierfür an: In dem Fall, dass mehrere Objekte durch Verdeckung eine gemeinsame Bildstruktur bilden, kann eine Erkennung auf Objektebene nicht erfolgen. Es ist daher eine Detailanalyse dieser Bildstruktur notwendig, zu deren Beginn sie wieder zur weiteren Bearbeitung freigegeben wird. Um die weitere Bearbeitung zu beschleunigen, ist eine Sperrung dieser Struktur notwendig, wenn alle Teilstrukturen in den aktuellen Instanzenbaum eingebunden sind. Diese Vorgehensweise lässt sich als Regel 3 zusammenfassen:

3. Eine Bildstruktur S, die auf Teilstrukturen analysiert wird, wird zunächst keiner Instanz zugeordnet und gilt genau dann als gesperrt, wenn alle Teilstrukturen $TS_1,...,TS_m$ von S einer Instanz $I_1,...,I_m$ zugeordnet werden konnten.

7.6 Die Verwaltung von Instanzen

Die während der Bearbeitung des Suchbaumes entstehenden Instanzen repräsentieren Objekte oder zumindest Teilobjekte, die bereits im Bild erkannt wurden. Wenn nun ein Suchpfad, auf dem bereits einzelne Instanzen gebildet wurden, nicht zum erwünschten Suchziel führt, soll das in diesem Suchpfad enthaltene Wissen über einzelne Bildbereiche auch anderen Suchpfaden zur Verfügung gestellt werden. Als eine Wissenseinheit bieten sich hierzu die erzeugten Instanzen an. Diese beschreiben ja bereits, dass z.B. ein im Objektmodell enthaltenes Teilobjekt im Bild lokalisiert und erkannt werden konnte. Wird nun an anderer Stelle im Suchbaum wiederum nach diesem Teilobjekt gesucht, so kann auf das vorhandene Wissen zurückgegriffen werden. Dazu werden die (Pre-)Attribute der Instanz an der entsprechenden Stelle in den aktuell bearbeiteten Suchpfad eingefügt.

Wichtig ist hierbei, dass die in der Instanz verwendeten Bildstrukturen entsprechend der in Abschnitt 7.5 beschriebenen Regeln auf ihre Verwendbarkeit innerhalb dieses Suchpfades überprüft werden. Der Zugriff auf geeignete Instanzen wird über die *Instanz*-Relation ermöglicht, die eine Verbindung zwischen den Konzepten und den während der Bearbeitung entstehenden Instanzen realisiert. Die bereits erzeugten Instanzen eines gesuchten Konzeptes werden dabei für einen Suchpfad in sogenannte *freie* und *benutzte* Instanzen unterschieden.

Definition 7.4: Eine Instanz I heißt dann *benutzt* bezüglich eines Suchpfades P, wenn sie in diesen Pfad eingebunden ist. Andernfalls heißt sie *frei*.

Bei diesem Einfügen muss natürlich beachtet werden, dass auch die Instanzen in Relation zueinander stehen, so wie dies für die entsprechenden Konzepte gilt. In Abbildung 7.8 wird dies deutlich, wenn im

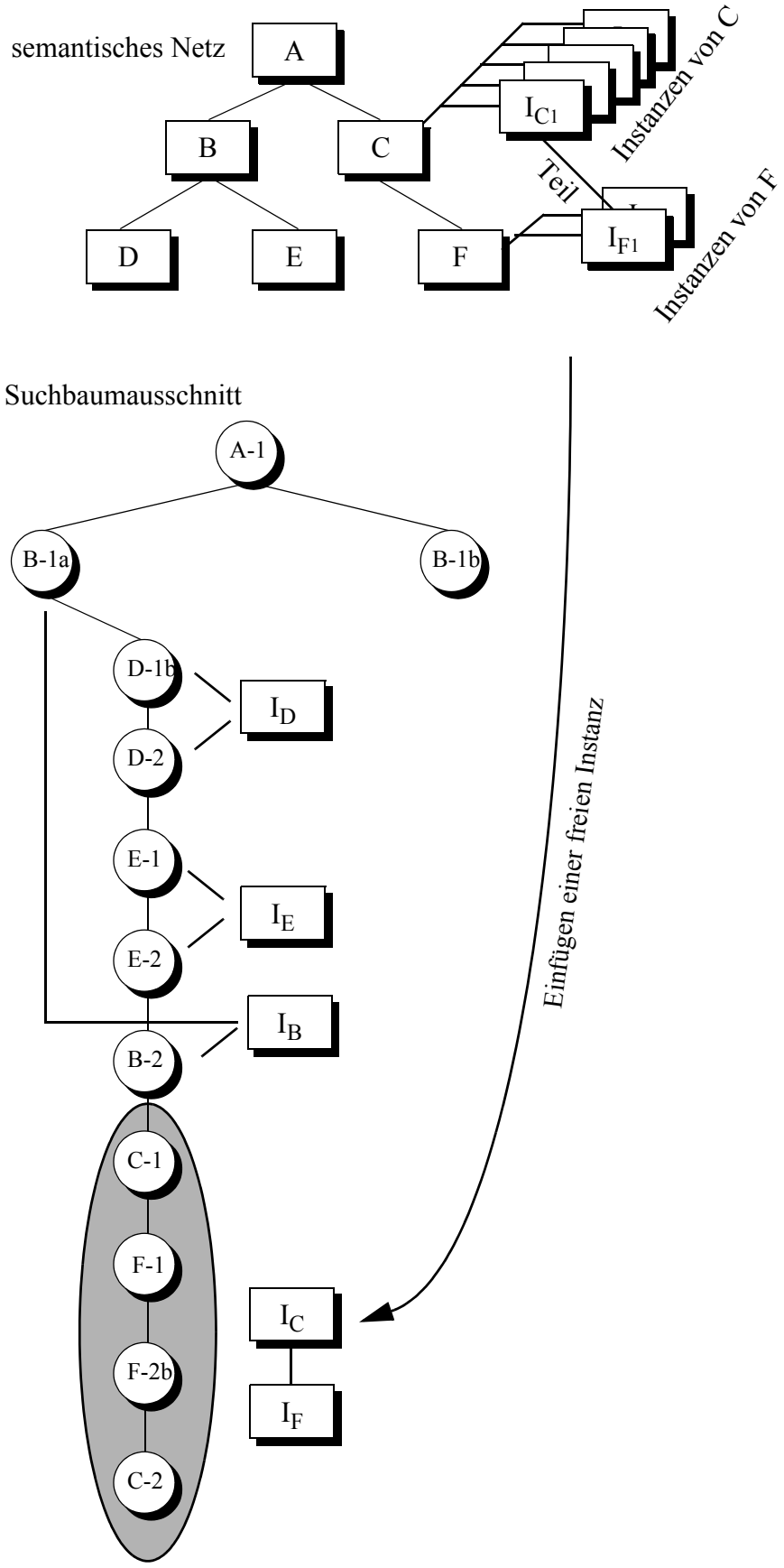


Abb. 7.8: Einfügen einer freien Instanz I_C mit gleichzeitigem Einfügen der Teilinstanz I_F ohne erneutes Berechnen der Attribute

Suchbaum eine freie Instanz des Konzeptes *C* eingefügt wird und gleichzeit damit auch eine Instanz von *F*, die eine Teilstruktur von *C* repräsentiert. Hierbei gilt:

Ist die Instanz I_C eine freie Instanz bezüglich des betrachteten Pfades *P*, so sind auch alle Instanzen *I*, die in Relation zu I_C stehen, freie Instanzen bezüglich *P*.

7.7 Die Verwaltung von Operationsergebnissen

Neben dem Suchaufwand im zu bearbeitenden Suchbaum bestimmt auch die eigentliche Attributberechnung wesentlich den Zeitaufwand für die Erkennung. Dies trifft insbesondere für die bildverarbeitenden Operationen zu. In vielen Fällen führen aber diese Berechnungen gar nicht zu einer erfolgreichen Instanziierung. Dies ist zum Beispiel der Fall, wenn ein Objekt gesucht wird, verschiedene Bildsegmente hierzu herangezogen werden und bei der Klassifikation dann festgestellt wird, dass sie alle nicht dem gesuchten Objekt entsprechen, dafür aber anderen Objektklassen zugeordnet werden. Diese Klassifikationsergebnisse müssen auch zu einem späteren Zeitpunkt noch den entsprechenden Bildsegmenten zugeordnet werden können, um für die Instanziierung anderer Konzepte ausgewertet werden zu können. Anders als bei herkömmlichen Logik- oder Backtrackingsystemen werden aus diesem Grunde Ergebnisse, die bei der Bearbeitung eines (Pre-)Attributgraphen berechnet wurden, auch dann gespeichert und weiter verwaltet, wenn sie nicht zu einer Instanziierung des Konzeptes geführt haben. Zu diesem Zweck werden die einzelnen Operationsergebnisse miteinander verkettet. Bei der Attributberechnung kann dann bereits am Operanden entschieden werden, ob ein Operationsaufruf notwendig ist oder ob das Ergebnis bereits vorliegt, da es schon zu einem früheren Zeitpunkt berechnet worden war. Dabei können einem Operanden natürlich auch mehrere Ergebnisse angehängt werden, wenn er zur Berechnung mehrerer Attribute dient. Um nun das geeignete Ergebnis auszuwählen, werden daher jeweils auch die verwendete Operation und ihre Parameter gespeichert. Weiterhin werden speziell bei den topologischen Operationen auch mehrere Operanden zur Attributberechnung verwendet,

so dass in diesem Fall auch der gesamte Kontext auf seine Gültigkeit überprüft werden muss. Abbildung 7.9 zeigt eine solche Ergebniskette, bei der an ein Bildsegment der resultierende Merkmalsvektor und das Klassifikationsergebnis angehängt sind.

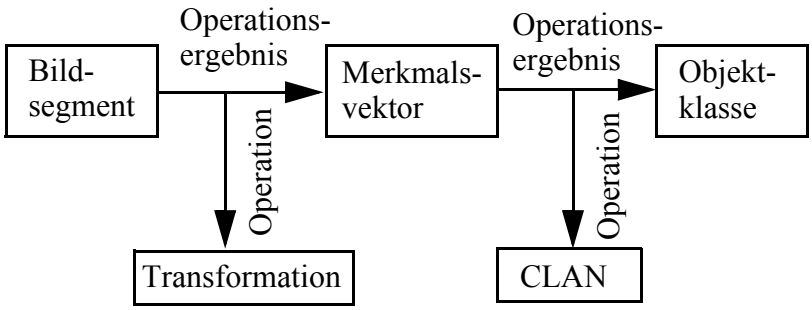


Abb. 7.9: Aufbau von Ergebnisketten, die für spätere Instanziierungsversuche zur Verfügung gestellt werden

Auf diese Weise wird sichergestellt, dass jede Bildstruktur tatsächlich nur ein einziges Mal die typischen Bildverarbeitungsschritte durchläuft.

8 Parallele Instanziierungsstrategien

Im vorigen Kapitel wurde beschrieben, wie die Instanziierung eines Objektmodells durchgeführt werden kann, indem diese auf ein Suchproblem abgebildet wird. Die Auswertung von Suchbäumen besitzt aber offensichtlich eine inhärente datengetriebene Parallelität, indem parallel alternative Zuweisungen, die sich als Verzweigungen im Suchbaum darstellen, untersucht werden. Daneben ergibt sich eine modellgetriebene Parallelität auf der Modellierungsebene, wenn ein Objekt durch mehrere Teilansichten beschrieben wird. Diese können gleichzeitig im Bild gesucht werden. Einen Überblick über mögliche Ansätze zur Parallelisierung von Kontrollalgorithmen findet man in [56].

In diesem Kapitel wird nun zunächst ein datengetriebener Ansatz auf Objekt- und Teilobjektebene beschrieben. Danach wird ein neuer modellgetriebener Ansatz erläutert, der parallel verschiedene Spezialisierungen oder Teile eines Objektes untersucht.

8.1 Datengetriebene Parallelität

Abbildung 8.1 zeigt noch einmal, wie der Kontrollalgorithmus aus einem semantischen Netzwerk (oben) einen Suchbaum (unten) generiert.

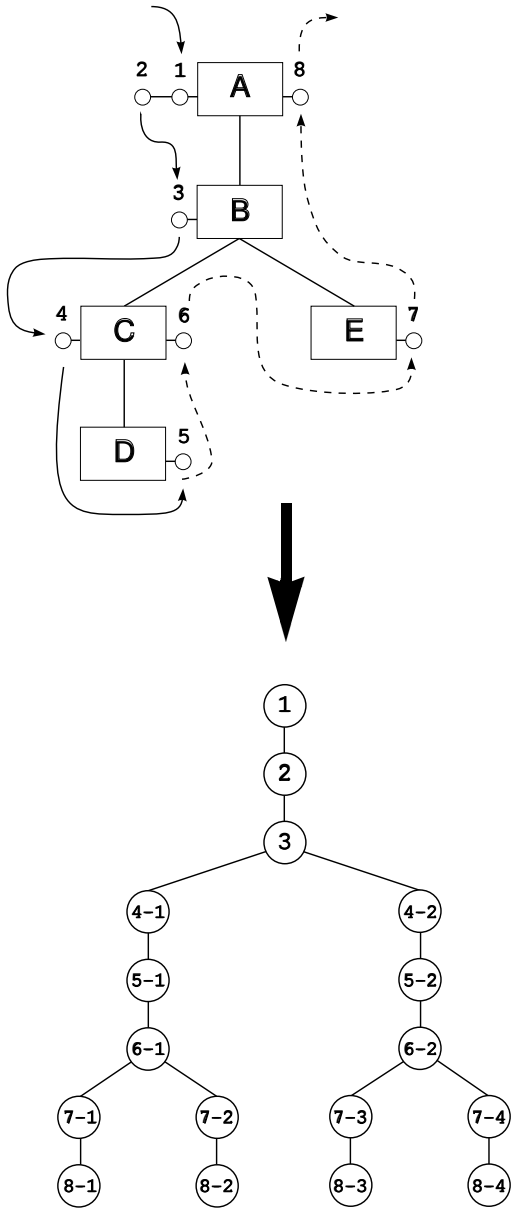


Abb. 8.1: Generierung des Suchbaums aus dem semantischen Netz

Die (Pre-)Attribute der einzelnen Konzepte werden der Reihe nach aufgegriffen und als Suchbaumknoten in dieser Reihenfolge miteinan-

der verkettet. Die Reihenfolge, in der die Konzepte ausgewertet werden, ergibt sich aus der hierarchischen Struktur des semantischen Netzwerkes. Die Pre-Attribute werden während der Expansion, die Attribute während der Instanziierung in den Suchbaum eingebunden.

Verzweigungen im Suchbaum entstehen nun, wenn mehrdeutige Operationen zur Attributberechnung verwendet werden. Ein paralleler Kontrollalgorithmus kann jeden neu entstehenden Pfad als separate Teilaufgabe behandeln. Folglich werden mehrere mögliche Lösungspfade ausgewertet, wobei der Pfad mit der höchsten Bewertung das beste Ergebnis liefert. Der allgemeine Suchalgorithmus (Algorithmus 7.2) muss daher lediglich um die parallele Weiterverarbeitung mehrerer expandierter Knoten erweitert werden.

Ein paralleler Algorithmus für die Graphsuche

```
function parallele Graphsuche : boolean
  found:= false
  offen:= {Start}
  geschlossen:= { }
  for i=1 to max_threads do in parallel
      found = Graphsuche(offen, geschlossen);
end function

function Graphsuche(offen, geschlossen): boolean
  while not found and offen ≠ { } do
      N:= Elementwahl (offen)
      offen:= offen \ {N};
      geschlossen:= geschlossen ∪ {N};
      if Endebedingung (N) then
          found:= true
      else
          S:= expandiere (N);
          offen:= offen ∪ (S\ geschlossen);
  end while
  Graphsuche:= found
end function
```

Algorithmus 8.1: Paralleler Suchalgorithmus

Der parallele Suchalgorithmus besteht also aus mehreren nebenläufigen Prozessen, die in dieser Implemntierung (Solaris 7, SUN-Workstations) als *Threads* (*Programmfäden*) bezeichnet werden. Diese Threads arbeiten mit einem gemeinsamen Hauptspeicher und können somit jeder auf die Mengen der offenen und geschlossenen Suchbaumknoten zugreifen.

Dabei muss beachtet werden:

- dass über Synchronisationsmechanismen sichergestellt wird, dass nicht zwei Threads gleichzeitig auf die gemeinsamen Daten zugreifen;
- dass Kamera und Roboter Ressourcen darstellen, auf die ebenfalls nicht gleichzeitig zugegriffen werden darf;
- dass Wartezeiten beim synchronisierten Zugriff auf diese Ressourcen minimiert werden;
- dass bei der Attributberechnung, also dem Expandieren eines Knotens, sichergestellt wird, dass Mehrfachberechnungen - wie in Kapitel 7.7 beschrieben - vermieden werden.

Die erste Bedingung des sequentialisierten Zugriffs auf gemeinsame Speicherbereiche kann recht einfach durch Semaphore realisiert werden. Auch die zweite Bedingung kann auf diese Weise umgesetzt werden, jedoch muss beachtet werden, dass hierbei zwei verschiedene Ressourcen - Roboter und Kamera - in einem kritischen Bereich verwendet werden können. Dies ist genau dann der Fall, wenn zunächst der Roboter zu einer neuen Position verfahren wird und dann anschliessend von dort ein Bild aufgenommen wird. Um dabei Deadlock-Situationen zu vermeiden, wird festgelegt, dass diese beiden Verarbeitungsschritte zusammenhängend in einer Operation MOVE_AND_GRAB durchgeführt werden müssen. Dadurch kann diese Operation als Ganzes als kritischer Abschnitt betrachtet werden und durch Semaphore geschützt werden.

Schwieriger sicherzustellen sind die beiden anderen Bedingungen. Wenn mehrere Suchbaumknoten gleichzeitig bearbeitet werden sollen, so kann davon ausgegangen werden, dass bei mehr als nur einem von diesen bei der Expansion auf Roboter und Kamera zugegriffen wird. Dies hätte aber zur Folge, dass mehrere der Prozesse in einen Wartezustand geraten und darauf warten, dass ihnen diese exklusiven Ressour-

cen zugewiesen werden. Um dies zu vermeiden, wird eine zusätzliche Aufteilung der offenen Knoten durchgeführt. Die offenen Knoten werden unterteilt in eine Menge von Knoten, die Roboter und Kamera zur Expansion benötigen, und solche, die ohne diese Ressource auskommen. Bei der Auswahl der offenen Knoten wird dann darauf geachtet, dass nur einer der Threads einen Knoten auswählt, der auf Roboter und Kamera zugreift. Daher wird Algorithmus 8.1 zu Algorithmus 8.2 modifiziert.

Es werden nun zwei Mengen offener Knoten unterschieden: offene Knoten ohne Ressourcenzugriff (*offen_ohne*) und offene Knoten mit Ressourcenzugriff (*offen_mit*). Einer der Threads bearbeitet nun bevorzugt die offenen Knoten, die die exklusive Ressource *Roboter mit Kamera* benötigen, während die anderen Threads unverändert die offenen Knoten ohne Ressourcenzugriff bearbeiten. Diese Threads müssen nur beim Einordnen der neu erzeugten Suchbaumknoten anhand der modellierten Operation entscheiden, in welche der Mengen offener Knoten diese eingefügt werden müssen.

Da das Expandieren eines Suchbaumknotens immer mit dem relativ zeitaufwendigen Ausführen einer Bildverarbeitungsfunktion verbunden ist, war in Kapitel 7.7 bereits diskutiert worden, wie Mehrfachberechnungen - also z.B. das mehrfache Klassifizieren einer Bildstruktur - vermieden werden können. Die dort vorgestellte Variante des Verknüpfens von Operanden, Operation und resultierendem Ergebnis muss jedoch für die parallele Instanziierung erweitert werden. Hier ist es nun nämlich möglich, dass auf einen Operanden zugegriffen wird, der gleichzeitig auch von einem anderen Thread mit der gleichen Operation bearbeitet wird. Aus diesem Grund wird die Datenstruktur, in der Operanden abgelegt sind, um eine zusätzliche Liste erweitert. In dieser wird vermerkt,

- ob ein Operand zur Zeit schon von einem anderen Thread bearbeitet wird und wenn ja
- welche Operationen gerade auf diesen Operanden ausgeführt werden.

Modifikation der parallelen Graphsuche

```
function parallele Graphsuche : boolean
  found:= false
  offen:= {Start}
  geschlossen:= { }
  found = Graphsuche_spezial(offen, geschlossen);
  for i=2 to max_threads do in parallel
      found = Graphsuche(offen, geschlossen);
end function

function Graphsuche_spezial(offen_mit, offen_ohne,
                                          geschlossen): boolean
  while not found and offen_ohne ≠ { } and offen_mit ≠ { } do
      if offen_mit  ≠ { } then
          N:= Elementwahl (offen_mit)
          offen_mit:= offen_mit \ {N};
      else
          N:= Elementwahl (offen_ohne)
          offen_ohne:= offen_ohne \ {N};
      geschlossen:= geschlossen  ∪  {N};

      if Endebedingung (N) then
          found:= true
      else
          S:= expandiere (N);
          for each s in S do
              if s not in geschlossen then
                  if s benötigt Ressource then
                      offen_mit:= offen_mit  ∪  {s};
                  else
                      offen_ohne:= offen_ohne  ∪  {s};
  end while
  Graphsuche:= found
end function
```

Algorithmus 8.2: Modifizierter paralleler Suchalgorithmus

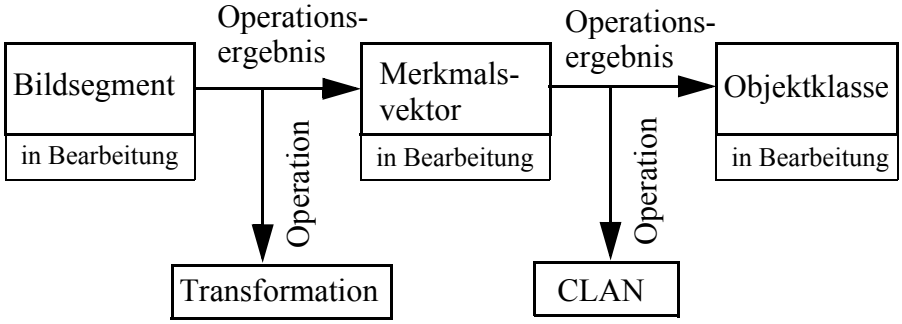


Abb. 8.2: Modifizierte Operandenverkettung zur Vermeidung von Mehrfachberechnungen durch Einfügen der Kennung *in Bearbeitung*

Stellt ein Thread t_1 nun fest, dass die ihm zugeteilte Operation auf dem entsprechenden Operanden bereits von einem anderen Thread t_2 bearbeitet wird, so wartet dieser nicht einfach auf das Bereitstellen des Ergebnisses. Er startet vielmehr einen weiteren Thread t_3 , der seinerseits in den Suchprozess eingreift und sich den nächsten offenen Suchbaumknoten auswählt. Der wartende Thread t_1 suspendiert sich anschliessend, so dass er keine weitere CPU-Zeit verbraucht und wartet auf ein Aktivierungssignal durch den Thread t_2 , welches dieser sendet, wenn er mit seiner Bearbeitung fertig ist.

Diese Vorgehensweise führt zu einem modifizierten Expandieren der Suchbaumknoten, wie in Algorithmus 8.3 beschrieben. Bei der Bearbeitung eines Suchbaumknotens wird zunächst markiert, dass die dazugehörige Struktur bearbeitet wird. Anschliessend werden die notwendigen Operanden gesucht und es wird geprüft, ob Operationsergebnisse bereits vorliegen oder gerade von einem anderen Thread berechnet werden. Ist letzteres der Fall, so erfolgt das Suspendieren. Nach Abschluss der Attributberechnung werden eventuell auf diese Berechnung wartende threads durch Versenden eines Signals wieder aktiviert.

Damit bei dieser Vorgehensweise des zusätzlichen Erzeugens von Threads nicht beliebig viele Threads im System aktiv werden, überprüft jeder Thread vor Entnahme eines Suchbaumknoten aus der Menge der offenen Knoten, ob zur Zeit überzählige Threads aktiv sind. Wenn dies

Modifiziertes Expandieren von Suchbaumknoten

```
function expandiere( N ) : Knotenmenge
  N.in_Bearbeitung := true;
  O := suche_Operanden( N );
  if in_Bearbeitung( O ) then
       suspend( O.flag );
  if N noch nicht berechnet then
       E := berechene_Attribut( N );
       verkette( N, E );
       signal( N.flag );
  else
       E := extrahiere an N gebundene Ergebnisse;
  S := { };
  for each e in E do
       erzeuge_Suchbaumknoten( e );
       S := S ∪ {e}
  expandiere := S;
end function
```

Algorithmus 8.3: Suspendieren von Threads beim Warten auf Operationsergebnisse

der Fall ist - d.h. diese Threads sind tatsächlich aktiv und nicht wartend oder suspendiert - so beendet sich der Thread selbst. Somit bewegt sich die Anzahl der aktiven Threads recht konstant um eine zu Programmstart festgelegte Maximalanzahl. Diese entspricht in etwa der Anzahl der im System zur Verfügung stehenden Prozessoren (siehe hierzu auch Abschnitt 8.4).

8.2 Modellgetriebene Parallelität

Die datengetriebene Parallelisierung erweist sich nicht immer als ausreichend, um eine gleichmäßige Auslastung der Prozessoren zu erzielen. Da, wie eingangs erwähnt, die Objektmodellierung auf der Basis

holistisch verarbeiteter Teilansichten durchgeführt wird, kann oftmals bereits nach erfolgreicher Erkennung einer einzelnen Teilansicht eine genaue Lageschätzung des Objektes und damit der restlichen Teilansichten erfolgen. In diesem Fall mutiert ein Großteil des Suchbaums schon sehr frühzeitig zu einem einzelnen Pfad, der zur Hypothesenverifikation bearbeitet werden muss. Aus diesem Grund wird auch der Ansatz der modellgetriebenen Parallelität verfolgt, der in anderen Kontrollalgorithmen in der Bildverarbeitung, wie sie z.B. in [57] vorgestellt werden, bislang nicht realisiert wurde.

Um diesen Ansatz detailiert beschreiben zu können, ist es zunächst notwendig, einige Begriffe festzulegen.

Definition 8.1: Suchbaumknoten, die sich auf das gleiche Attribut eines Konzeptes der Modellierung beziehen und durch Mehrfachoperationen beim Expandieren des Suchbaums entstanden sind, werden *Mehrlinge* genannt. Dies sind die Knoten im Suchbaum, die einen gemeinsamen Vorgängerknoten haben.

Definition 8.2: Die Konzepte im semantischen Netzwerk, die einen gemeinsamen Vorgänger in der Modellierungshierarchie besitzen, werden *Bruderkonzepte* genannt. Dies sind also zum Beispiel die Teile eines Objektes.

Definition 8.3: Bei der parallelen Bearbeitung von Bruderkonzepten werden temporäre Suchbaumzweige erzeugt, mit denen diese im Suchbaum auf der gleichen Ebene eingefügt werden. Diese temporären Suchbaumzweige werden *Bruderzweige* genannt.

Bei der modellgetriebenen Parallelisierung wird nun der Ansatz verfolgt, vom Objektmodell - also dem semantischen Netzwerk - ausgehend nach Möglichkeiten einer Parallelisierung zu suchen. Dabei stellt man fest, dass *Bruderkonzepte* in der Modellbeschreibung relativ unabhängig voneinander bearbeitet werden können. Daher kann deren Bearbeitung auch parallel erfolgen.

Da jedoch die Algorithmen nicht primär von dem Objektmodell ausgehen, sondern sich in erster Linie an der Datenstruktur des Such-

baums orientieren, ist die Realisierung dieses Ansatzes deutlich aufwendiger als der der datengetriebenen Parallelisierung. Betrachtet man noch einmal Abbildung 8.1, so wird deutlich, dass Bruderkonzepte im Suchbaum nicht auf einer Ebene auftreten, sondern in einem Pfad des Suchbaumes nacheinander auftreten und somit gar nicht gleichzeitig für eine parallele Auswertung zur Verfügung stehen. Um nun aber die modellgetriebene mit der datengetriebenen Parallelität zu vereinen, werden beim Expandieren eines Suchbaumknotens temporäre Verzweigungen im Suchbaum erzeugt, die zu den Knoten führen, die die verschiedenen Bruderkonzepte repräsentieren.

Die Auswertung dieser temporären Teilzweige erfolgt nun durch den bestehenden Algorithmus analog zu den übrigen Teilzweigen durch je einen Thread. Nach Abschluss der Auswertung verbindet der letzte Thread den bearbeiteten Teilzweig mit seinen Bruderzweigen, so dass der Suchbaum am Ende die gleiche Gestalt hat, wie es ohne die temporären Verzweigungen der Fall gewesen wäre.

Abbildung 8.3 verdeutlicht diese Arbeitsweise. Der Suchbaum auf der linken Seite zeigt eine Szene, welche die Seitenplatte des Cranfield-Satzes zum Inhalt hat. In der vereinfachten Modellierung besteht diese Seitenplatte aus einer Aussparung (Zweig links) und einer 12mm-Bohrung (Zweig rechts). Beide Teilaspekte werden unabhängig voneinander betrachtet (Bewegen der Roboterkamera und Aufnahme eines neuen Bildes, Auswertung und Erkennung des Teilobjektes). Sobald ein Teilzweig ausgewertet wurde, fügt er sich mit seinen Bruderzweigen zu einem Zweig zusammen. Diesen Vorgang zeigt der rechte Suchbaum in Abbildung 8.3. Dort ist die temporäre Verbindung des rechten Teilzweiges zum Startknoten durch die endgültige Verbindung zum linken Nachbarn ersetzt worden. Am Ende des Suchbaumpfades steht die Erkennung der Gesamtszene („Seitenplatte“) mit Hilfe der erkannten Teilobjekte.

Dieses einfache Beispiel unterschlägt, dass modellgetriebene und datengetriebene Verzweigungen oftmals gemeinsam auftreten. Das Verbinden von Teilzweigen wird dann zu einem komplexen Vorgang, den Abbildung 8.4 verdeutlicht.

Ein temporärer Suchbaumzweig, der n alternative Eingangsdaten zugewiesen bekommt, teilt sich in n Äste. Jeder dieser n Mehrlinge

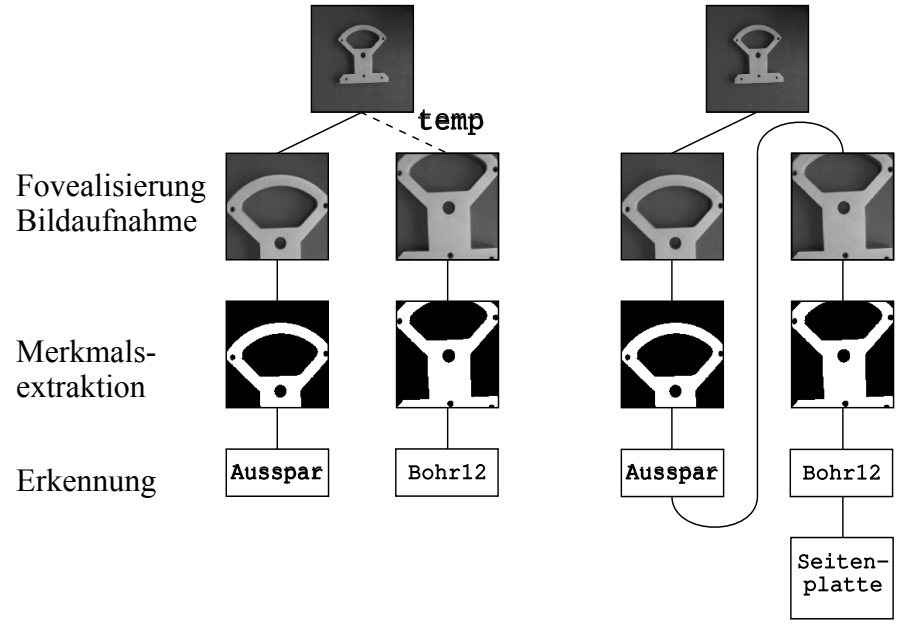


Abb. 8.3: Verwendung temporärer Suchbaumzweige und anschließendes Verketten der Bruderzweige

muss am Ende der Bearbeitung in den endgültigen Suchbaumpfad rekombiniert werden. Da aber nur (maximal) ein linker bzw. rechter Bruderzweig vorhanden ist (den sich diese n Äste teilen müssten), müssen n-1 Kopien der Nachbarbäume erstellt werden. Diese Aufgabe erfüllt ebenfalls der Kontrollalgorithmus. Allgemein gilt für die Anzahl der zu erstellenden Kopien: $k = (m-1) \cdot (n-1)$ mit k: Anzahl der benötigten Zweigkopien; m: Anzahl der Bruderzweige; n: Anzahl der Mehrlinge eines Bruderzweiges.

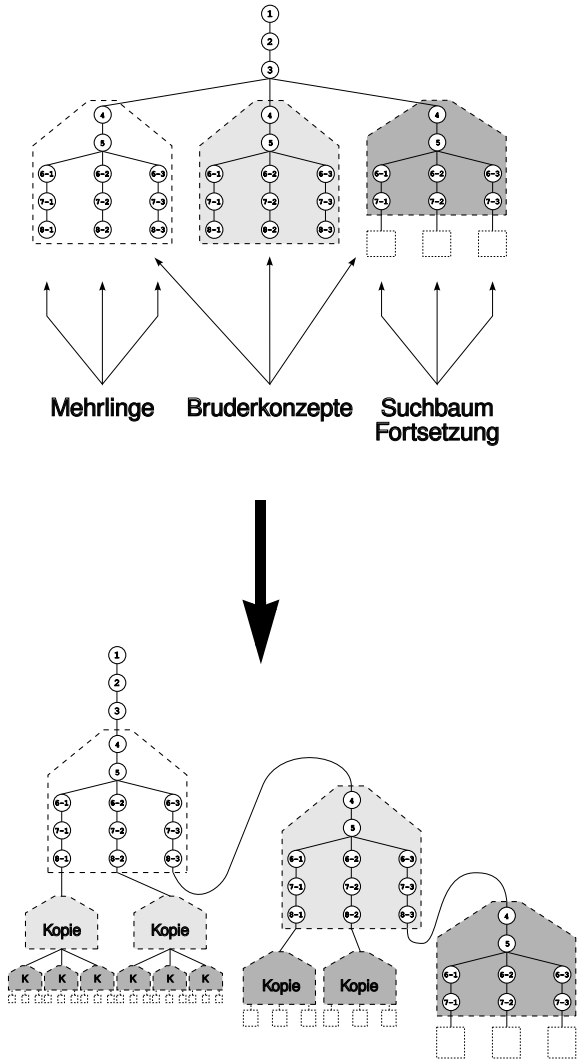


Abb. 8.4: Rekombinieren eines Suchbaums aus temporären Teilzweigen und das dabei notwendige Erzeugen von Kopien

8.3 Strategien zur Suchraumbegrenzung

Um bei der Auswertung komplexer Szenarien durch das oben beschriebene Kopieren von Suchbaumbereichen keine zu stark wachsen-

den Suchbäume zu erzeugen, wurden einige Heuristiken zum Beschneiden (*Pruning*) der Suchbäume in das System integriert. Dabei erfolgt das Pruning im wesentlichen in zwei Fällen:

- einzelne Pfade scheinen nicht zu einer guten Problemlösung zu führen und werden daher nicht weiter verfolgt;
- ein Pfad hat bereits eine so hohe Bewertung erzielt, das andere Pfade wahrscheinlich keine bessere Bewertung erreichen werden.

Für die hierzu notwendigen Entscheidungen werden zwei Schwellwerte herangezogen, die vom Knowledge-Engineer applikationsspezifisch gesetzt werden können. Hierdurch wird der Suchbaum deutlich kleiner gehalten, was sich in einer verbesserten Laufzeit und in einem geringeren Speicherbedarf widerspiegelt. Es muss jedoch ergänzt werden, dass die Zulässigkeit des Verfahrens im Sinne von Definition 7.3 nicht in jedem Falle garantiert werden kann. Bei den durchgeführten Experimenten hat sich dies jedoch als unproblematisch herausgestellt. Offensichtlich ist aber hierbei ein Trade-off zwischen starkem Pruning und damit verbundenem Beschleunigen der Suche und dem Erhalt der Zulässigkeit zu fällen. Wichtig ist aber in jedem Falle, dass für die Erkennung besonders wichtige Teilstrukturen möglichst frühzeitig untersucht werden und das Pruning daher auf der Basis der Erkennung dieser Teile durchgeführt wird. Bei der Objektmodellierung müssen daher die besonders wichtigen Teilstrukturen als erstes beschrieben werden.

Durch die Eingabe der Abbruchparameter durch einen Knowledge-Engineer bei der Objektmodellierung kann das System gezielt für eine bestimmte Umgebung, z.B. Objekte auf einem Transportband oder in einer industrielle Montagezelle, optimiert werden.

In diesem Bereich sind noch weitere Arbeiten geplant, um bei der Modellierung in Abhängigkeit des erstellten Modells geeignete Schwellwerte automatisch vorzuschlagen. Dies würde die Modellbildung in diesem Punkt wesentlich vereinfachen.

8.4 Ergebnisse

Die PAWIAN-Umgebung wurde auf einer Sun Sparc1000 mit 4 Prozessoren implementiert, wobei die Parallelisierung mittels Multithreading (MT) realisiert wurde. In dem dabei verwendeten Zwei-Schichten-MT-Modell des Betriebssystems Solaris hat der Programmierer keinen Einfluss auf die Verteilung der Threads auf die vorhandenen Prozessoren [104]. Diese Aufgabe übernimmt die Thread Library, die neben den Threads des Kontrollalgorithmus auch noch sämtliche anderen MT-Anwendungen, wie z.B. den Solaris Kernel oder den Fenstermanager, schedulen muss. Da alle vorhandenen Threads um die Hardware-Ressourcen konkurrieren, läuft die Bilderkennung auch nicht auf allen vier Prozessoren gleichzeitig ab.

Für alle Testreihen wurde die Seitenplatte des Cranfield-Montagesatzes verwendet. Da die Verwendung des Roboterarmes den Einfluss der maximalen Threadanzahl auf die gesamte Bearbeitungszeit verfälscht, wurde zur Messung der relativen Beschleunigung eine Messumgebung geschaffen, welche die verwendeten Operationen simuliert. Dadurch sind wir in der Lage, die Leistung des parallelen Kernels weitestgehend unabhängig von äußeren Einflüssen zu messen. Die Messungen ergeben, dass nach einem erwartungsgemäß rapiden Abfall der Laufzeit bei steigender Threadzahl bald ein Sättigungseffekt auftritt. Diese Sättigung muss eintreten, da die Anzahl der Prozessoren limitiert ist. Außerdem nehmen Verwaltungsaufwand und Kollisionen bei mehreren Threads verständlicherweise zu und verhindern eine weitere Beschleunigung der Bearbeitung bzw. kehren sie um. Abb. 8.5 zeigt die normierte Laufzeit bei Verwendung von 1 bis 10 Threads für den parallelen Kontrollalgorithmus für zwei unterschiedlich komplexe Szenen.

Erwartungsgemäß wird die beste Laufzeit bereits bei einer kleinen Threadanzahl erreicht, da nur vier Prozessoren zur Verfügung stehen. Dass dies nicht bei vier Threads der Fall ist, liegt zum einen daran, dass die Prozessoren auch durch Systemprozesse belastet werden. Der theoretisch mögliche Beschleunigungswert von vier wird zum anderen nicht erreicht, da der Kontrollalgorithmus natürlich auch einen sequentiellen Anteil besitzt, der stark von der Modellierung der Objekte abhängt. Der Einfluss des sequentiellen Anteils auf die Performanz des

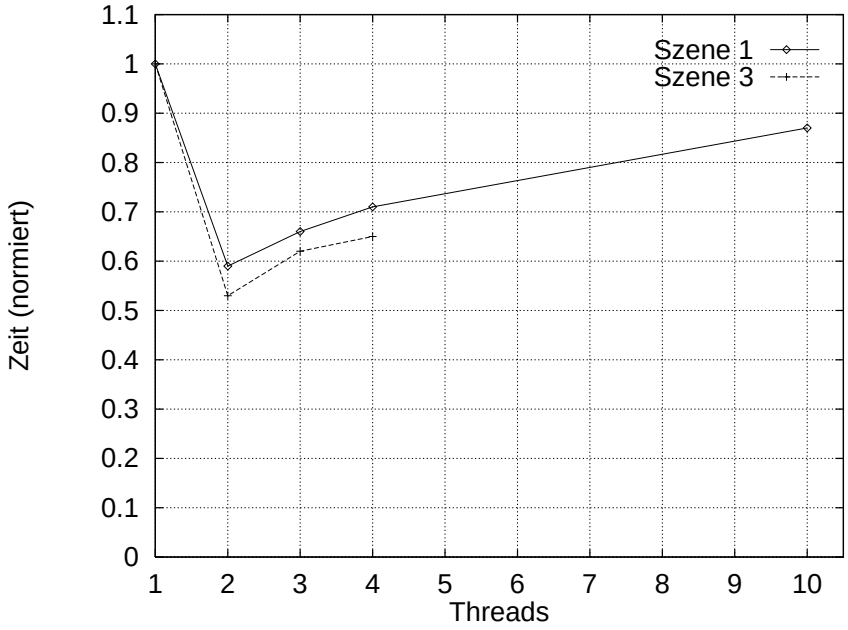


Abb. 8.5: Beschleunigung des Kontrollalgorithmus (inkl. sequentieller Programmteile) für zwei unterschiedlich komplexe Szenen

Kontrollalgorithmus ist umso größer, je einfacher die behandelten Szenen und Objektmodelle gehalten werden, d.h. je geringer der Anteil daten- und modellgetriebener Parallelität ist. Darüber hinaus wird in den existierenden Objektmodellen sehr stark vom Roboter Gebrauch gemacht. D.h. es wird sehr stark ein aktiver Erkennungsansatz verfolgt, bei dem die Kamera häufig zu neuen Fovealisierungspunkten verfahren wird. Hierbei kann zwangsläufig als optimale Parallelisierung lediglich eine pipeline-artige Struktur erzielt werden, in der ein Bild ausgewertet wird, während gleichzeitig die Kamera zu einem neuen Aufnahmepunkt verfahren wird. Da die Bewegung des Roboters im Vergleich zur Bildauswertung relativ langsam ist, können in diesen Fällen nur zwei Prozessoren effektiv ausgelastet werden (siehe Abb. 8.6).

Es muss ebenfalls berücksichtigt werden, dass die Vergleichsmessung für ein bzw. zwei Threads nicht auf einer Ein- bzw. Zwei-Prozessormaschine, sondern ebenfalls auf der Sparc1000 mit 4 Prozessoren erfolgt ist. Somit ist ein Vergleich der ermittelten Werte zwar zulässig, aber nicht unter dem Gesichtspunkt des Speedups (Verhältnis von Bearbeitungszeit bei Verwendung von *n* Prozessoren zur Bearbeitungszeit bei Verwendung von einem Prozessor) zu sehen.

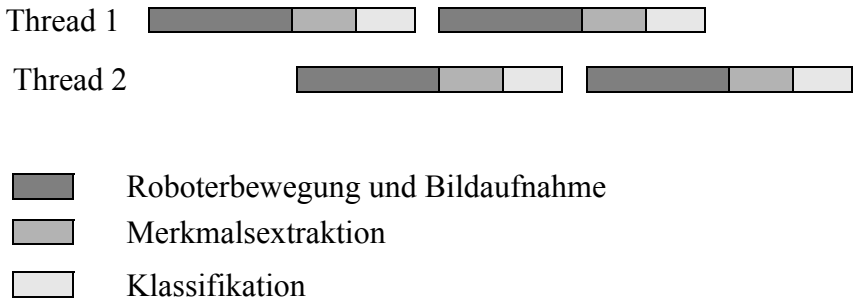


Abb. 8.6: Pipeline zur Bildauswertung

Die Auslastung der vier Prozessoren ist in Abb. 8.7 dargestellt. Man erkennt neben der unterschiedlichen Dauer der drei Testreihen, dass mit zunehmender Threadzahl die vorhandenen Prozessoren auch besser ausgelastet werden. Die Threadanzahl (1, 4, 5) gibt dabei an, wieviele Threads maximal nebeneinander existieren dürfen. Tatsächlich wird die maximale Anzahl - gerade bei einfachen Szenen und Objektmodellen - nur selten erreicht. Somit ist zu erklären, dass bei maximal 4 Threads zur Spitzenlast nur 3 der 4 Prozessoren auslastet werden. Werden maximal 5 Threads zugelassen, so werden bis zu 90% der vorhandenen Prozessorkapazität ausgelastet. Dass die Bearbeitungszeit trotz der besseren Auslastung der Prozessoren geringfügig höher ist, liegt an der Anzahl der vorhandenen Aufgaben. So kommt es bei 5 Threads zu Wartezeiten, die bei 4 Threads gerade noch nicht auftreten.

Ein weiterer Aspekt, der bereits bei der Beurteilung der Beschleunigung angesprochen wurde, zeigt sich bei der Kurve für die Verwendung von nur einem Thread. Hier wird eine Auslastung der Prozessorkapazität von etwa 45% erreicht, d.h. es werden fast zwei Prozessoren benutzt. Der Grund hierfür liegt in der Tatsache, dass neben dem Bilderkennungsprogramm auch noch Betriebssystemroutinen, Fenstermanager, etc. die Prozessoren belasten. Diese Belastung beträgt etwa 5-15% der Gesamtkapazität von 4 Prozessoren. Da der Kontrollalgorithmus im Singlethreaded-Betrieb nicht auf einen Prozessor beschränkt ist, ist die Gesamtlaufzeit tatsächlich geringer, als dies auf einem Ein-Prozessor-System der Fall wäre. Leider stand für die Messungen kein vergleichbares Ein-Prozessor-System zur Verfügung.

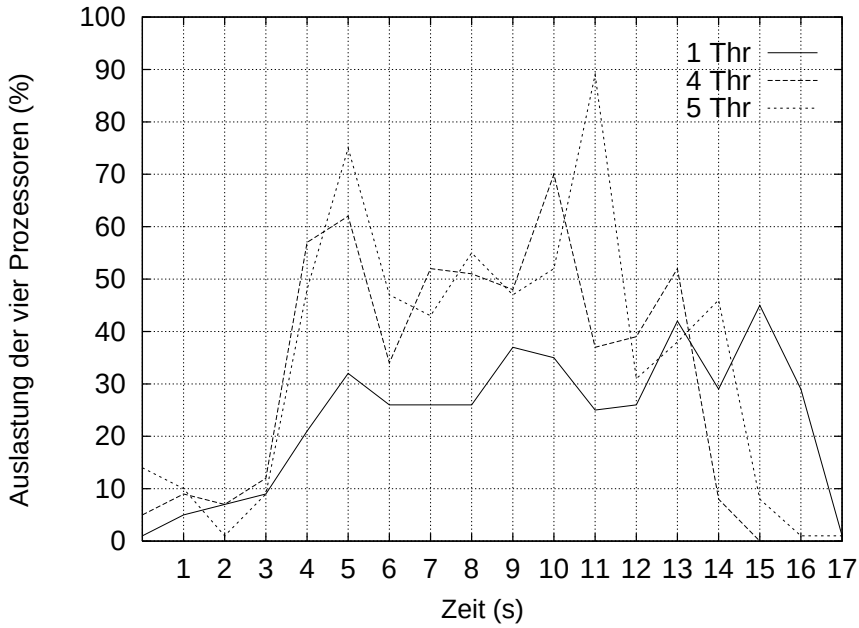


Abb. 8.7: Prozessor-Auslastung während eines Programmdurchlaufs bei Bearbeitung einer Seitenplatte

Die Ergebnisse zeigen aber bereits sehr deutlich, dass eine Parallelisierung für kleine Parallelrechner-Systeme (2-4 Prozessoren), wie sie heute bereits im Massenbereich mit Intel-Prozessoren angeboten werden, zu einer guten Beschleunigung führt.

9 Aktive Objekterkennung mit dem hybriden System

Bis zu diesem Punkt der Arbeit wurde ein hybrides Erkennungssystem vorgestellt, das zur Auswertung eines einzelnen Bildes eines Objektes oder einer Szene verwendet wird. Im folgenden Kapitel soll nun herausgearbeitet werden, wieso und wie ein aktiver Ansatz verfolgt wird. Dies geschieht im wesentlichen aus den folgenden Gründen:

- ❒ die Auswertung verschiedener Blickrichtungen auf ein Objekt zur robusteren Erkennung
- ❒ die Auswertung von Objektdetails durch Zoom oder Heranfahren der Kamera zur robusteren Unterscheidung ähnlicher Objekte
- ❒ zur detaillierten Vermessung des Objektes

Im folgenden wird beschrieben, wie sich die Ansteuerung aktiver Mechanismen in die Objektmodelle integrieren lässt und wie speziell in diesem System die Roboterkinematik integriert wurde. Darüber hinaus muss auch die Wissensbeschreibungssprache die Modellierung unterschiedlicher Objektansichten und der dazu gehörigen Blickwinkel und ähnlicher Parameter erlauben.

In Kapitel 2 und 3 war schon ausführlich auf die Vorteile und Eigenschaften aktiver Sehsysteme eingegangen worden. Für die aktive Objekterkennung sind dabei insbesondere von Bedeutung:

- die Blicksteuerung: Wohin soll geschaut werden, wohin soll die Kamera bewegt werden?
- die Robotik: Wie kann die Kamera zur gewünschten Position verfahren werden?
- die Fusion der verschiedenen Bildauswertungen: Wie sollen die Erkennungsergebnisse fusioniert werden?

Diese drei Punkte sollen in den folgenden Abschnitten näher untersucht werden.

9.1 Die Blicksteuerung

Blicksteuerungen werden in verschiedenen Arbeiten vorgeschlagen, so von Milanese, Moravec, Reisfeld und Yeshurun [122], [127], [152], [195]. Hierbei lassen sich zwei Schwerpunkte unterscheiden, zum einen wird Aufmerksamkeit durch Bewegung innerhalb von Bildsequenzen gesteuert, zum anderen werden Aufmerksamkeitsregionen innerhalb eines auszuwertenden Bildes bestimmt. Beide Vorgehensweisen haben ihre Berechtigung und ergänzen sich gegenseitig. In unserem Fall betrachten wir eine Aufmerksamkeitssteuerung für die Erkennung unbewegter Objekte. Auch hierbei lassen sich die verschiedenen Verfahren wieder in zwei Klassen einteilen:

- datengetriebene, bottom-up Strategien
- modellgetriebene, top-down Strategien.

Im folgenden wird eine Strategie vorgestellt, die datengetrieben arbeitet und daher prinzipiell ohne a priori Wissen auskommt. Ihr Vorteil gegenüber anderen Mechanismen liegt im wesentlichen darin, dass sie auf Merkmale zugreift, die auch für die Erkennung eingesetzt werden und daher keinen nennenswerten Zusatzaufwand mit sich bringt. Durch die übergeordnete Kontrollschicht ergibt sich zudem eine modellgetriebene Rückkopplung, so dass letztlich hier ein Ansatz vorgestellt wird, indem bottom-up und top-down Strategien zur Aufmerksamkeitssteuerung integriert sind. Diese sich ergänzenden Strategien werden auch im Blockschaltbild des Systems in Form zweier Rückkopplungen gezeigt (Abb. 5.4).

9.1.1 Regionenbasierte Fovealisierung

Eine häufig eingesetzte Fovealisierungsstrategie zur Blicksteuerung basiert auf der Segmentierung eines Bildes in Regionen gleicher bzw. ähnlicher Helligkeit oder Farbwertes [2], [7]. Dabei wird z.B. durch Schwellwertbildung oder durch ein Regionenwachstumsverfahren das Bild in mehrere Regionen zerlegt. Besonders auffällige Regionen, die sich z.B. durch einen starken Kontrast zu benachbarten Regionen auszeichnen oder die einem gesuchten Helligkeits- bzw. Farbwert entsprechen, werden dann im nächsten Schritt weiter analysiert. Dazu kann dann z.B. auf den Flächenschwerpunkt der Region fovealisiert werden, so dass dieser anschließend im Bildmittelpunkt liegt. Zudem kann durch Heranfahren oder Heranzoomen der Bildausschnitt so gewählt werden, dass die zu untersuchende Region anschließend bildfüllend aufgenommen wird.

Diese Vorgehensweise wird auch im hier beschriebenen System verwendet. Je nach Applikation und dem zu erwartenden Bildmaterial wird dabei entweder eine grauwertbasierte oder aber eine farbbasierte Segmentierung des Bildes durchgeführt[1]. Im folgenden soll die farbbasierte Segmentierung näher erläutert werden. Bei diesem Segmentierungsansatz wird davon ausgegangen, dass sich ein oder mehrere Ob-

1. Somit können für ein Objekt je nach Applikation verschiedene Modelle eingesetzt werden, die sich durch die verwendeten Operationen voneinander unterscheiden. Hier ergibt sich also ein Unterschied der Modelle auf der problemabhängigen intensionalen Ebene, wobei die konzeptionelle Ebene unberührt bleibt.

jekte vor einem homogenen Hintergrund befinden. Dies ist z.B. bei der Erkennung von Industrieteilen auf einem Fließband oder einem anderen Objektträger einfach sicherzustellen. Die Objekt-Hintergrund-Trennung findet dann - wie von Austermeier in [2] vorgeschlagen - in vier Schritten statt.

1. Transformation des RGB-Eingangsbildes in den HSI-Farbraum, bestehend aus den Komponenten *Farbwinkel* (*Hue*), *Sättigung* und *Intensität*, da der Farbwinkel eine recht stabile Segmentierung erlaubt. Die Transformationsvorschrift und einige Besonderheiten des HSI-Farbraumes werden im Anhang 2 beschrieben.
2. Mit Hilfe eines probabilistischen Abtastrasters mit einer zum Bildrand zunehmenden Abtastdichte wird nach einer Cluster-Analyse der Farbwinkel des Hintergrundes bestimmt.
3. Für jeden Bildpunkt wird dann die Differenz zur ermittelten Hintergrundfarbe bestimmt und durch eine Schwellwertoperation festgelegt, ob der Bildpunkt zum Hintergrund oder zu einem Objekt gehört.
4. Die zum Objekt gehörenden Bildpunkte werden zu Regionen zusammengefasst.

Die hier beschriebene Vorgehensweise wurde für die Erkennung verschiedener Modellautos eingesetzt. Diese sind aus Metall gefertigt, lackiert und besitzen eine stark reflektierende Oberfläche. Derartige Reflexionen stellen typischerweise eines der schwierigsten Probleme bei der Segmentierung dar, da die auftretenden Glanzlichter nur eine sehr geringe Sättigung aufweisen. Desweiteren müssen für die Anwendung des vorstehend beschriebenen Segmentierungsverfahrens noch die verwendeten Schwellwerte für die Objekt-Hintergrund-Trennung und für die Regionenbildung der zum Objekt gehörenden Bildpunkte festgelegt werden. Dazu wurde anhand einer Reihe von Testbildern der Objekte die Auswirkung verschiedener Schwellwerte untersucht. Die nachstehenden Abbildungen verdeutlichen dies. Die Autos, die hierzu verwendet wurden, sind ein roter Ferrari, ein dunkellila gefärbter Ferrari, ein silberner Porsche und ein gelbgrüner Punto. In der folgenden Abbildung ist zunächst die Zerlegung des RGB-Bildes in die Kompo-

nenten des HSI-Farbraumes - *Farbwinkel*, *Sättigung* und *Intensität* - dargestellt.

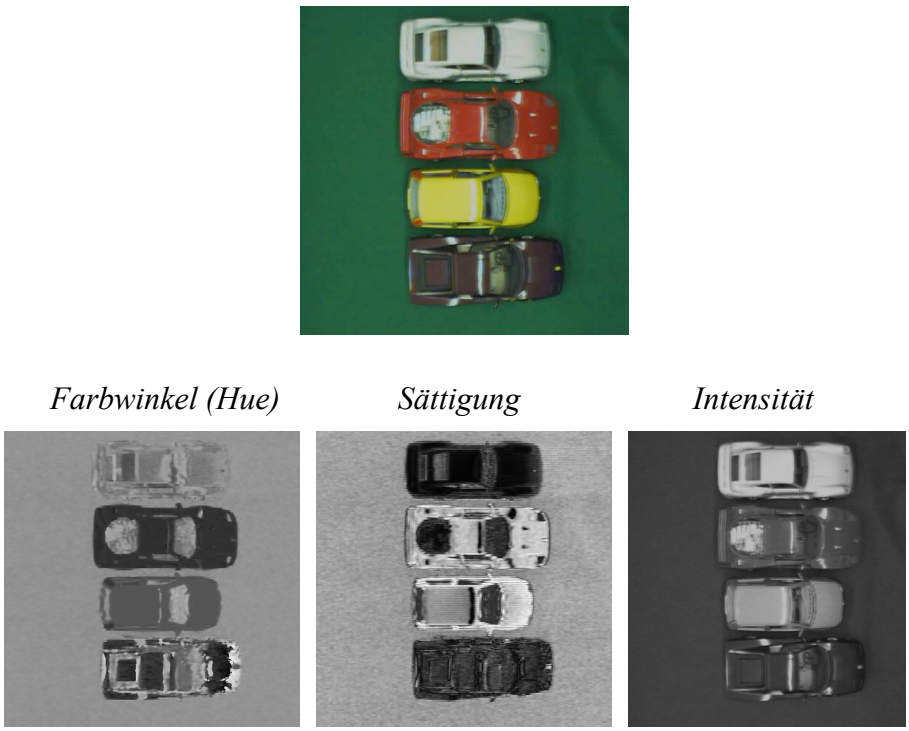


Abb. 9.1: Farbwinkel, Sättigung und Intensität eines Testbildes mit vier verschiedenen Modellautos

Die Abbildung 9.2 zeigt die Zwischenergebnisse und Endergebnisse, die mit festen Schwellwerten für die Objekt-Hintergrund-Trennung berechnet wurden. Mit Hilfe der Schwellwerte für Sättigung und Intensität wird festgelegt, ob der Farbwinkel für die Auswertung verwendet werden kann oder ob eine Grauwertsegmentation an der jeweiligen Bildkoordinate durchgeführt wird. Dabei wird mindestens eine vorgegebene Sättigung erwartet, um den Farbwinkel auswerten zu können. Ist diese nicht gegeben, so erfolgt die Segmentation durch Auswertung der Intensität. Eine Objekt-Hintergrund-Trennung erfolgt ebenfalls durch Auswertung der Intensitäts-Komponente, wenn die Intensitätsabweichung vom Hintergrund oberhalb einer gegebenen Intensitätsschwelle liegt. Der Schwellwert für den Farbwinkel legt fest, wie stark sich Objekt und Hintergrund in ihren Farbwinkeln unterscheiden müssen, um ein Pixel zum Objekt oder zum Hintergrund zugehörig zu definieren.

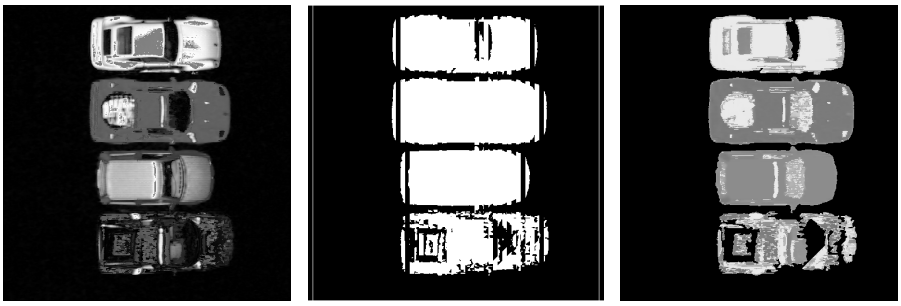


Abb. 9.2: Zwischenergebnisse mit extrahiertem Hintergrund und Endergebnis der Segmentation

In den Experimenten erwiesen sich ein Schwellwert von 50 für die benötigte Sättigung als sehr gut geeignet. Für die erlaubte Intensitätsdifferenz wurde ein Schwellwert von 70 festgelegt. Eine leichte Variation dieser beiden Parameter erwies sich als unkritisch, so dass hiermit tatsächlich eine recht robuste Segmentation zur Verfügung steht. Für die Farbwinkel-Differenz wurde ein Schwellwert von 15 festgelegt.

Im Ergebnisbild werden dann alle Regionen durch Grauwerte zwischen 0 und 255 repräsentiert. Dabei wird für den Hintergrund der Grauwert 0 (schwarz) verwendet. Für Regionen mit ausreichend großer Sättigung wird der mittlere Regionenfarbwinkel linear auf Grauwerte zwischen 100 und 220 abgebildet. Ist die Sättigung zu gering, wird die mittlere Intensität der Region linear auf Grauwerte zwischen 230 und 236 abgebildet.

Die in diesen Tests ermittelten Schwellwerte werden dann als Parameter der Operationen in der Modellierung eingetragen und bei der Operationsausführung ausgewertet. Somit kann für jedes Objekt und jede Applikation entsprechend der intensionalen Betrachtung des Problems eine geeignete Modellierung durchgeführt werden.

Da in Kapitel 12 noch eingehender auf die Modellierung der Autos und die resultierenden Erkennungsergebnisse eingegangen wird, soll an dieser Stelle hierauf verzichtet werden. Es soll stattdessen im folgenden Abschnitt ein weiteres Fovealisierungsverfahren vorgestellt werden, dass sich direkt auf die Objektrepräsentation bezieht und daher ohne Zusatzaufwand zur Verfügung steht. Dies ist die Fovealisierung auf markante Ecken.

9.1.2 Eckenbasierte Fovealisierung

In diesem Abschnitt wird nun ein eckenbasierter Ansatz vorgestellt, der in engem Zusammenhang mit den von uns entwickelten und verwendeten Erkennungsstrategien steht. Dazu soll zunächst kurz die Erzeugung von Eckenmerkmalen vorgestellt werden, deren Verwendung im Erkennungsprozess und die daraus resultierenden Mechanismen zur Generierung von Aufmerksamkeitspunkten. Die hierauf aufbauende Aufmerksamkeitssteuerung ist in einen wissensbasierten Erkennungsvorgang integriert.

Neben den zuvor erwähnten simplen und komplexen Neuronen mit unterschiedlich großen Einzugsgebieten existieren im visuellen Kortex weitere Neuronentypen. Diese weisen eine besonders hohe Aktivität auf, wenn eine Kante zwar innerhalb des rezeptiven Feldes verläuft, diese sich jedoch nicht im gesamten rezeptiven Feld befindet. An einem oder gar an beiden Enden des rezeptiven Feldes zeigen sich hier inhibitorische Wirkungen. Diese Zellen sprechen dadurch besonders gut auf Ecken oder Kantenenden an. Dieses Verhalten wird in unserem technischen System dadurch simuliert, dass während der von uns verwendeten Kontinuitätsüberprüfung zur Erzeugung der komplexen Neurone eine Überprüfung der Orientierung der Neurone mit besonders hoher Aktivität erfolgt. Weisen hierbei zwei benachbarte aktive Neurone eine Orientierungsdifferenz oberhalb einer vorgegebenen Schwelle (z.B. 45°) auf, so wird ein hyperkomplexes Eckenneuron aktiviert. In gleicher Weise wie die Kantenrepräsentation liegen hierbei natürlich auch diese Eckenstrukturen in wolkenartiger Form vor.

Die Eignung dieser "Eckenwolken" für eine robuste Erkennung konnte bereits in verschiedenen Anwendungen nachgewiesen werden. Die Eckenwolken dienen dabei als eine zusätzliche Information, die eine verbesserte Diskriminierung ähnlicher Objekte ermöglicht. Abbildung 9.3 zeigt zwei ähnliche Objekte, die aufgrund ihrer Kantenrepräsentation nicht in allen Fällen deutlich unterschieden werden können, da beide Objekte ein sehr großes Maß an Übereinstimmung in ihrer Kantenrepräsentation besitzen. Erst durch die Hinzunahme der Eckenrepräsentation wird dies fehlerfrei möglich (Abb. 9.4), da das zweite Objekt sowohl an den Bohrungen als auch an den abgeschrägten oberen

Ecken sehr ausgeprägte Eckenwolken bildet, die sich deutlich von denen des ersten Objektes unterscheiden.

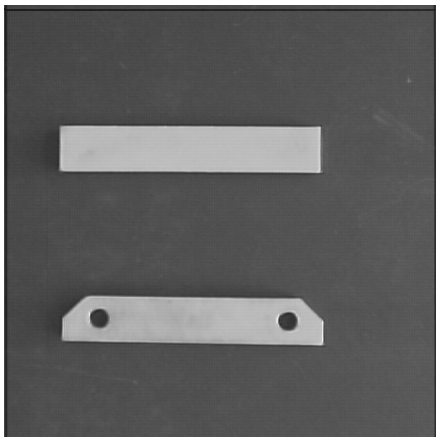


Abb. 9.3: Grauwertbild zweier Objekte

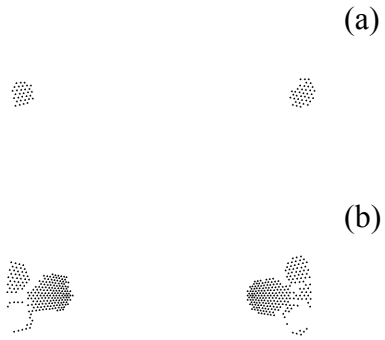


Abb. 9.4: Übereinstimmende (a) und nicht übereinstimmende Elemente (b) der Eckenrepräsentation

Die Eignung von Eckenwolken zur Erkennung mit Hilfe einer einfachen Matching-Klassifikation zeigt bereits, dass die Eckenrepräsentation ein robustes Merkmal darstellt, und deutet darauf hin, dass auch eine weitere Nutzung zur Fovealisierung auf Teilstrukturen eines Objektes möglich ist. Eine solche Auswertung von Teilstrukturen wird in den Fällen eingesetzt, in denen aufgrund der Komplexität des Objektes oder wegen Verdeckungen durch andere Objekte eine ganzheitliche Objekterkennung nicht möglich ist.

Um eine sichere Fovealisierung zu gewährleisten, muss dabei sichergestellt sein, dass

1. prägnante Objektecken hinreichend häufig zu einer Eckenkodierung führen;
2. eine moderate Anzahl an Fovealisierungspunkten erzeugt wird;
3. eine Positionsbestimmung mit einer Ungenauigkeit von max. vier Pixeln möglich ist.

Die Gründe für die erste Bedingung sind offensichtlich. Sie stellt sicher, dass eine Fovealisierung überhaupt möglich ist. Dabei ist es nicht

zwingend notwendig, dass eine Ecke tatsächlich in allen Fällen als solche kodiert wird. Das Fehlen führt zu einem Nichterkennen einer Teilstruktur des gesuchten Objektes kann aber durch die erfolgreiche Erkennung ausreichend vieler anderer Teilstrukturen kompensiert werden. Auf der anderen Seite soll aber natürlich die verwendete Fovealisierungsstrategie lediglich die interessanten Bildbereiche herausarbeiten und somit zu einer erheblichen Daten- und Aufwandsreduktion beitragen - je nach Komplexität der Szene von 512x512 möglichen Aufmerksamkeitspunkten auf deutlich weniger als hundert. Damit jedoch eine Erkennung überhaupt möglich wird, verlangt ein fovealisierendes System, dass die zu untersuchende Struktur tatsächlich im Zentrum des Bildes oder der Retina vorliegt. Die wolkenartige Kodierung der Kanten- und Eckenmerkmale in unserem System erlaubt dabei eine Verschiebungstoleranz von bis zu vier Pixeln, wenn auf eine explizite Translationsinvarianz verzichtet werden soll [94].

An einer Testreihe von 81 Bildern wurde daher zunächst überprüft, wie häufig die einzelnen Objektecken kodiert wurden und mit welcher Genauigkeit eine Lokalisierung der Ecke mit Hilfe des Schwerpunktes der dazugehörigen Eckenwolke möglich ist. Dabei wurde ein Objekt in verschiedenen Positionen und Drehungen aufgenommen und die jeweiligen Merkmalsvektoren gebildet. Um eine vergleichende Darstellung zu ermöglichen, erfolgte eine Normierung bezüglich Position und Rotation, so dass die Eckenwolken und ihre Schwerpunkte überlagert werden konnten. Abb. 9.5 zeigt zunächst das untersuchte Objekt (a), seine Eckenwolken (b) sowie eine Überlagerung aller Eckenschwerpunkte der Testreihe in normierter zweidimensionaler (c) und dreidimensionaler Darstellung (d). Es wird ein vergrößerter Ausschnitt aus dem Originalbild gezeigt. Der PKW nimmt im 256x256 Farbbild eine Fläche von etwa 75x25 Pixeln ein.

Es wird deutlich, dass sich sechs Ecken herauskristallisieren, die in fast jeder Testaufnahme mit nur geringen Verschiebungen des Schwerpunktes detektiert wurden und dass nur in wenigen Bildern einzelne unerwünschte Ecken kodiert wurden. Einige detailliertere Informationen stellt noch die folgende Tabelle vor, in der für alle in der Testreihe aufgetretenen Eckenwolken die Erwartungswerte, Standardabweichung und max. Abweichung der normierten Schwerpunkte aufgelistet wer-

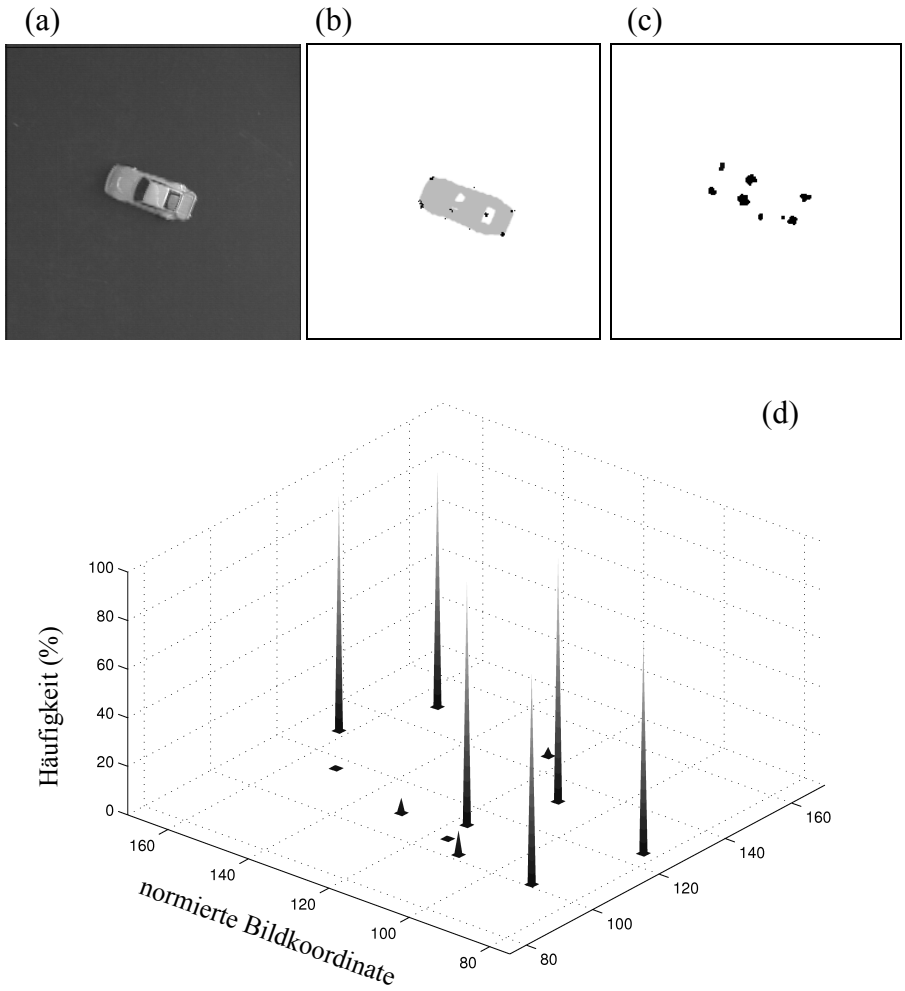


Abb. 9.5: Grauwertbild eines Testobjektes (a), Eckenwolken (b), Überlagerung der Schwerpunkte der Eckenwolken aller 81 Testbilder (c) und ihre Verteilung (d)

den. Auch hierbei wird nochmals deutlich, dass die Positionsgenauigkeit klar innerhalb der Toleranzgrenze von vier Pixeln liegt.

Aufgrund der Ergebnisse der Testreihe wurde eine Modellierung gewählt, die sich auf die zu erwartenden sechs Fovealisierungspunkte stützt. Dabei wurde jeweils ein Bildausschnitt um den Schwerpunkt der Eckenwolke ausgewertet. Gleichzeitig wurde auch die Kamera näher an die Struktur herangefahren. Abb. 9.6 zeigt eine anderes Auto, während die sich aus der Auswertung des Modells ergebenden acht fovealisierten Teilansichten in Abb. 9.7 dargestellt werden.

In Szenen mit mehreren Objekten, die sich gegenseitig teilweise verdecken, entstehen natürlich zusätzliche Eckenwolken an den Berüh-

Tabelle 2: Auswertung der Versuchsreihe

Ecke	Häufigkeit	Erwartungswert		Standardabweichung		Max. Abweichung
		Zeile	Spalte	σ_{Zeile}	σ_{Spalte}	
1	86%	101,6	91,4	0,721	0,816	2,7
2	10%	102,1	110,6	0,696	0,927	1,9
3	1%	106,0	116,0	0,000	0,000	0,0
4	6%	109,2	130,2	0,400	0,400	1,1
5	100%	113,4	116,7	0,609	0,766	3,4
6	1%	116,0	152,0	0,000	0,000	0,0
7	86%	124,9	83,3	0,805	1,049	2,9
8	96%	128,9	162,2	0,926	0,906	2,3
9	100%	131,7	109,6	0,700	0,786	2,4
10	4%	146,0	124,0	0,000	0,000	0,0
11	96%	148,7	153,9	0,876	0,728	2,6

rungspunkten der einzelnen Objekte. Auch hiervon werden verschiedene vom Roboter angefahren und die dazugehörigen Teilansichten werden ausgewertet. Die Auswertung ganzer Teilansichten erlaubt jedoch bereits nach erfolgreichem Erkennen einer der modellierten Teilansichten Rückschlüsse auf die Lage des gesuchten Objektes, so dass die restlichen Teilansichten zielgerichtet angefahren werden können, ohne alle Fovealisierungspunkte auswerten zu müssen. Dies wird in Abschnitt 9.5 noch detailliert diskutiert. Ein Ausschnitt aus der Modellierung der linken Frontpartie (in seiner Ansicht von oben) wird in Abbildung 9.8 gezeigt.

Es wird dabei von einem Initialbild ausgegangen, das nun für die Erkennung der linken Frontpartie auf seine Eckenstrukturen untersucht wird. Dazu wird die bereits für den Versuch der ganzheitliche Erkennung des Autos erzeugte Eckenrepräsentation einer einfachen Cluster-Analyse unterzogen, so dass die einzelnen Eckenwolken bestimmt werden können. Die Mittelpunkte der verschiedenen Cluster dienen dann

Abb. 9.6: Untersuchtes Testobjekt

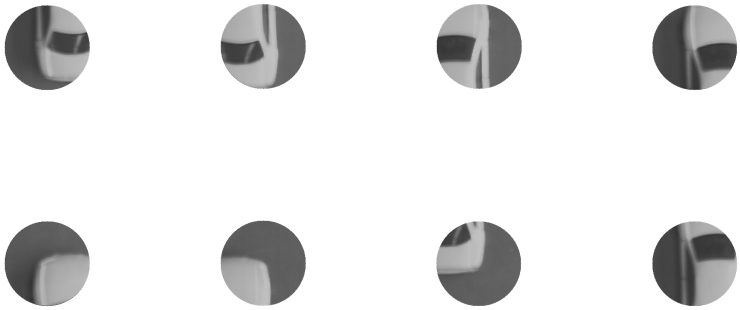


Abb. 9.7: Teilansichten bei Fovealisierung auf Eckenwolken

als Anfahrpunkte für eine neue detaillierte Bildaufnahme. Durch diese Fovealisierung liegt nun der Objektbereich, der zur Bildung der Eckenwolke geführt hatte, in der Bildmitte. Um eine weitere Konzentration auf diesen Bereich zu ermöglichen, wird ein Bildausschnitt aus der Bildmitte der weiteren Analyse durch Merkmalsextraktion und Klassifikation zugeführt.

```
Konzeptname :          linke Frontpartie
  Teil von :           Auto
  Attribut :           Eckenbild
      Typ :            BOOLEAN
      Wertebereich :   {TRUE}
      Operation :      FEATURE_TO_IMAGE
          Operand :    <Merkmal.Auto>
      Formales Erg.:   <Eckenbild>
  Attribut :           Struktur
      Typ :            BOOLEAN
      Wertebereich :   {TRUE}
      Operation :      SEGMENTATION
          Operand :    <Eckenbild>
          Parameter :  Size>20
      Formales Erg.:   <Eckenposition>
  Attribut :           Bild
      Typ :            BOOLEAN
      Wertebereich :   {TRUE}
      Operation :      MOVE_AND_GRAB
          Operand :    <Eckenposition>
      Formales Erg.:   <Bild>
  Attribut :           Ausschnitt
      Typ :            BOOLEAN
      Wertebereich :   {TRUE}
      Operation :      CUT_IMAGE
          Operand :    <Bild>
          Parameter :  50
       Formales Erg.:  <Detailbild>
  Attribut :           Merkmalsextraktion
      Typ :            BOOLEAN
      Wertebereich :   {TRUE}
      Operation :      TRANSFORMATION
          Operand :    <Detailbild>
      Formales Erg.:   <Merkmal>
  Attribut :           Klassifizierung
      Typ :            STRING
      Wertebereich :   {linke Frontpartie}
      Operation :      CLAN
          Operand :    <Merkmal>
      Formales Erg.:   <Form>
  Bewertung :          <Form>
ENDE :                 linke Frontpartie
```

Abb. 9.8: Modellierung der eckenbasierten Fovealisierung; optionale Einträge wie Operationsgebiet und Parameter, die nicht besetzt sind, werden nicht dargestellt.

9.2 Integration der Motorik in den Erkennungsprozess

Nachdem im vorangegangenen Abschnitt zwei alternative Vorgehensweisen vorgestellt wurden, um Fovealisierungspunkte in einem Bild zu bestimmen, ist es nun noch nötig, die Kamerabewegungen zu bestimmen, die dazu führen, dass zum einen die Fovealisierungspunkte gezielt angefahren werden können und zum anderen auch modellbasiert neue Objektansichten erreicht werden. Typischerweise geschieht dies in sechs Schritten:

1. interessante Blickpunkte werden im Bild ermittelt
2. der Abstand zwischen Kamera und Objekt wird bestimmt
3. eine neue Kameraposition wird berechnet
4. die Roboterkinematik wird überprüft und es werden eventuell alternative Positionen bestimmt
5. die Kamera wird verfahren, ein neues Bild wird aufgenommen
6. aus erkannter Teilobjektposition wird auf die Objektposition geschlossen und umgekehrt

Hierzu nun noch einige Erläuterungen:
Der erste Arbeitsschritt war bereits im vorangegangenen Abschnitt vorgestellt worden. Ergebnis dieser Blickpunktbestimmung ist eine Menge von Bildkoordinaten (x, y) interessanter Bildpunkte, die näher analysiert werden sollen. Aus der Differenz einer solchen Bildkoordinate zum Bildmittelpunkt, der Brennweite des Objektivs, der aktuellen Kameraposition und dem aktuellen Abstand der Kamera zum Objekt kann dann ermittelt werden, wie die Kamera verfahren werden muss, um sie mittig über dem Fovealisierungspunkt zu positionieren. Hierzu wird eine sechsdimensionale Weltkoordinate berechnet. Im vierten Schritt wird mit Hilfe eines Kinematikmodells des Roboters überprüft, ob die gewünschte Kameraposition überhaupt angefahren werden kann. So kann zum Beispiel die neue Kameraposition auch außerhalb des Arbeitsbereiches des Roboters liegen, oder einzelne Gelenkwinkel sind nicht einstellbar. Wenn jedoch die Position gültig ist, so wird die Kamera verfahren und ein neues Bild wird aufgenommen.

Diese ersten fünf Arbeitsschritte beziehen sich bislang noch auf eine rein datengetriebene Blicksteuerung. Bei der eine Bewegung der Kamera nur innerhalb der Bildebene durchgeführt wird. Dabei wird also die Position der Kamera verändert, nicht aber die Blickrichtung, mit der die Kamera in die Szene hineingerichtet ist. Da im hier vorgestellten System aber auch explizites Modellwissen zur Verfügung steht und im semantischen Netz topologische Relationen zwischen einzelnen Objektansichten und Objektteilansichten modelliert sind, besteht darüber hinaus die Möglichkeit, nach Erkennen einer Ansicht und der dabei verwendeten Kameraposition auf eine Normlage des Objektes und somit auf eine Normlage verschiedener anderer Ansichten zu schließen. Dies wird durch den sechsten Arbeitsschritt angedeutet.

In den folgenden Abschnitten soll nun näher erläutert werden, wie zunächst bei der datengetriebenen Fovealisierung neue Kamerapositionen bestimmt werden (Schritte zwei und drei). Danach wird beschrieben, wie modellgetrieben neue Blickpositionen berechnet werden und wie hierzu die Modellierung aussieht (Schritt sechs) bevor dann abschließend noch auf das Kinematikmodell des Roboters eingegangen wird (Schritt vier). Diesen Erläuterungen werden die wichtigsten mathematischen Grundlagen zunächst noch vorangestellt.

9.3 Roboter und Kamerasystem

Für verschiedene Anwendungen des Erkennungssystems wird im Laborbetrieb ein Sechsgelenk-Arm-Roboter der Firma Siemens (Modell *manutec R2*) verwendet. Dieser dient sowohl dazu, Objekte zu greifen und zum Beispiel zu montieren als auch dazu, die Kamera zu bewegen, um so beim Erkennungsvorgang zu verschiedenen Ansichten des Objektes zu gelangen. Dazu wurde die Kamera mit einer speziellen Halterung am Tool-Center-Point neben einem Greifer-Wechslersystem befestigt. Abbildung 9.9 zeigt den Roboter mit Kamerahalterung.

Der Roboter wird mit seiner serienmäßigen Steuerung betrieben. Diese ist in der Lage, über eine serielle Schnittstelle Anfahrpositionen im Roboterkoordinatensystem einzulesen und auszuwerten. Wenn also

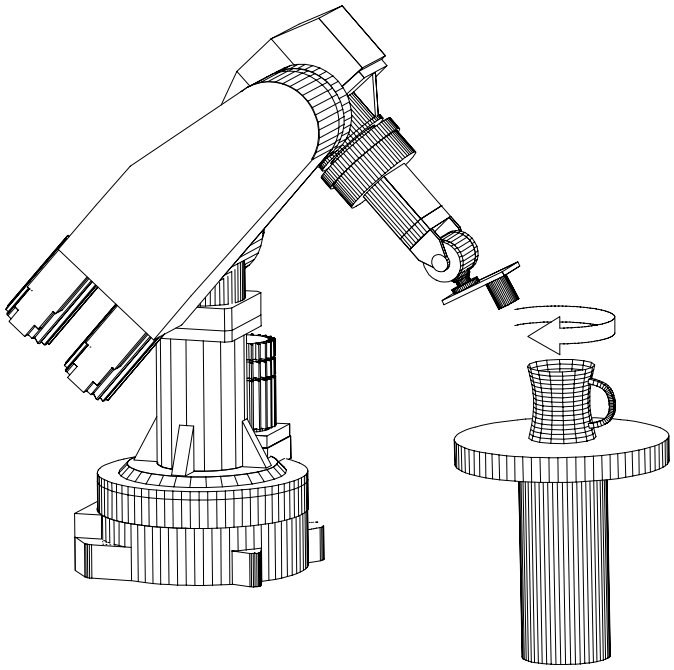


Abb. 9.9: Roboter *manutec R2* mit angeflanschter Kamera

die Kamera an einer bestimmten Raumkoordinate (x, y, z) eine gewünschte Blickrichtung (ϑ, φ, ε) einnehmen soll, so ist es notwendig, die korrespondierenden TCP-Koordinaten im Roboterkoordinatensystem zu bestimmen und diese an die Steuerung zu übertragen. Die daraus resultierende Bewegung (Trajektorie, Geschwindigkeiten, Gelenkstellungen, etc.) wird dann in der Steuerung bestimmt und geregelt.

Der Roboter wird also in kartesischen Koordinaten gesteuert. Die resultierenden Gelenkwinkel werden in der Steuerung des Roboters berechnet. Ein Satz von Transformationsmatrizen ${}^0\boldsymbol{A}$ bis ${}^5\boldsymbol{A}$ beschreibt, wie die den Gelenken zugeordneten Koordinatensysteme ineinander überführt werden können [164]. Da die Kamera zusätzlich montiert wurde, wird eine weitere Transformationsmatrix ${}^6\boldsymbol{A}$ eingeführt, die den Übergang vom TCP-Koordinatensystem zum Kamerakoordinatensystem beschreibt. Es gilt dabei für die Transformation eines Punktes $\boldsymbol{p}_i$ im Koordinatensystem des Gelenkes i in das Koordinatensystem des Gelenkes i+1:

$$\boldsymbol{p}_{i+1} = {}^{i}\boldsymbol{A}\ \boldsymbol{p}_i \tag{9.1}$$

Als Beispiel für den Aufbau einer Transformationsmatrix soll die Transformation vom Kamerakoordinatensystem zum Koordinatensystem im Tool-Center-Point (TCP) des Roboters dienen. Unabhängig

vom verwendeten Robotertyp und seiner Steuerung muss diese Transformation in jedem Falle ausgeführt werden, da die Kamerahalterung nachträglich an den Roboter angesetzt wurde und nicht in die Steuerung integriert ist.

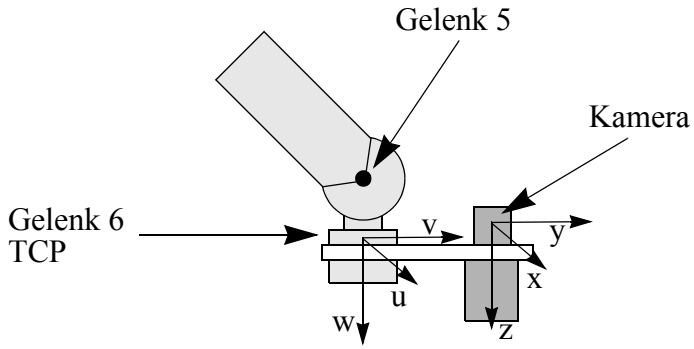


Abb. 9.10: Koordinatensysteme in TCP und Kamera

Die beiden zu betrachtenden Koordinatensysteme sind im allgemeinen gegeneinander gedreht und verschoben. Ein Vektor $\boldsymbol{p}_{Kamera} = \boldsymbol{p}_7$ im Kamerakoordinatensystem (x, y, z) kann durch eine Multiplikation mit einer Rotationsmatrix $\boldsymbol{R}$ und eine anschließende Addition eines Translationsvektors in den entsprechenden Vektor $\boldsymbol{p}_{TCP} = \boldsymbol{p}_6$ im TCP-Koordinatensystem (u, v, w) transformiert werden. Da dies in vielen Fällen unhandlich ist, werden üblicherweise homogene Koordinaten verwendet, in denen die vierte Komponente des Vektors $\boldsymbol{p}_7$ seine Skalierung beschreibt. Dann kann die Translation auch durch Multiplikation mit einer Translationsmatrix $\boldsymbol{T}$ beschrieben werden.

Für das Beispiel in Abbildung 9.10 ergibt sich zunächst für die drei benötigten Drehungen, die zunächst um die z-Achse, dann um die so entstehende y-Achse und anschließend um die x-Achse um die Winkel a, b und c erfolgen:

$$\boldsymbol{R}_{z,\,a} = \begin{bmatrix} \cos a & \sin a & 0 \\ -\sin a & \cos a & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9.2}$$

$$\boldsymbol{R}_{y,\,b} = \begin{bmatrix} \cos b & 0 & -\sin b \\ 0 & 1 & 0 \\ \sin b & 0 & \cos b \end{bmatrix} \tag{9.3}$$

$$\boldsymbol{R}_{x,\,c} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos c & -\sin c \\ 0 & \sin c & \cos c \end{bmatrix} \tag{9.4}$$

Somit erhält man durch Multiplikation der drei einzelnen Rotationen und durch Umwandlung in homogene Koordinaten

$$\boldsymbol{R} = \boldsymbol{R}_{x,\,c} \cdot \boldsymbol{R}_{y,\,b} \cdot \boldsymbol{R}_{z,\,a} = \tag{9.5}$$

$$\begin{bmatrix} \cos a \cos b & \sin a \cos b & \sin b & 0 \\ -\sin a \cos c - \sin b \sin c \cos a & \cos a \cos c - \sin a \sin b \sin c & \sin c \cos b & 0 \\ \sin a \sin c - \sin b \cos a \cos c & -\sin c \cos a - \sin a \sin b \cos c & \cos b \cos c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

und

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.6}$$

wobei die vierte Ziele der Translationsmatrix eine Skalierung des Koordinatensystems beschreibt. Da diese hier nicht von Bedeutung ist, müssen in die Einheitsmatrix lediglich die drei Verschiebungskomponenten eingetragen werden. Es gilt dann für die gesamte Transformation

$$\boldsymbol{p}_6 = \boldsymbol{T} \cdot \boldsymbol{R} \cdot \boldsymbol{p}_7 \tag{9.7}$$

Die beiden Transformationsmatrizen können natürlich auch durch Multiplikation zusammengefasst werden und man erhält als sogenannte homogene Transformationsmatrix

$${}^{6}\boldsymbol{A}^{T} = \boldsymbol{T} \cdot \boldsymbol{R} \tag{9.8}$$

Wie man der Zeichnung entnehmen kann, ist in diesem Fall die Kamera so angebracht, dass die Koordinatensysteme lediglich gegeneinander verschoben sind. Die Rotationsmatrix ist daher die Einheitsmatrix. Auch die zu den übrigen Gelenken gehörenden Rotationsmatrizen weisen ähnliche Einschränkungen auf, da die Gelenkachsen eines Industrieroboters normalerweise nicht windschief zueinander liegen, sondern häufig parallel oder rechtwinklig.

Mit Hilfe der zuvor beschriebenen Transformation kann also ein Punkt, der ursprünglich im Bildkoordinaten als Fovealisierungspunkt detektiert worden war, zunächst in ein Kamerakoordinatensystem (Abschnitt 9.4) und von dort in ein Koordinatensystem des Roboters überführt werden. Dabei kann dieser zweite Schritt wie beschrieben durch die verschiedenen Gelenk-Koordinatensysteme bis in das Basis-Koordinatensystem des Roboters propagiert werden.

Die für die Transformation von einem Objektkoordinatensystem in das Roboterkoordinatensystem notwendigen Berechnungen, stehen als Operationen für die Modellierung zur Verfügung und werden in den Attributbeschreibungen verwendet.

9.4 Die datengetriebene Bestimmung neuer Blickpunkte

Nach der Bestimmung interessanter Bildpunkte (x, y), die als Bildkoordinaten vorliegen, ist es notwendig, hieraus Roboterbewegungen abzuleiten. Dabei wird zunächst einmal davon ausgegangen, dass die Blickrichtung bei der Bildaufnahme beibehalten werden soll. Somit muss im ersten Schritt aus der gewünschten Bewegung im Bild die notwendige Kamerabewegung bestimmt werden, um dann anschließend auf die Roboterbewegung überzugehen.

Wenn unter Beibehaltung der Blickrichtung ein Bildpunkt (x, y) durch Verschiebung der Kamera in den Bildmittelpunkt gerückt werden soll, so ist es notwendig, mit Hilfe der Abbildungseigenschaften der Kameraoptik, den Verschiebungsvektor im Bild in eine Bewegung in

Kamerakoordinaten umzurechnen. Im allgemeinen reicht es dabei aus, für die Optik das bekannte *Lochkameramodell* anzusetzen.

Die Abbildungseigenschaften werden durch Abbildung 9.11 verdeutlicht.

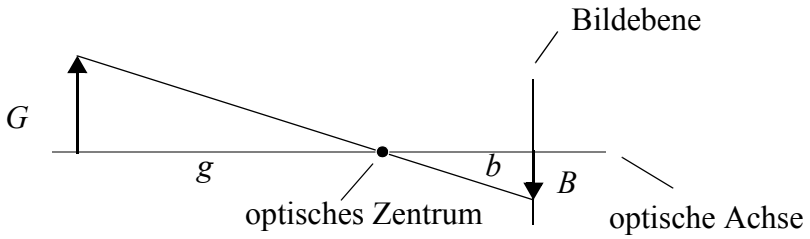


Abb. 9.11: Lochkameramodell mit Brennweite *b*, Objektentfernung *g,* Objektgröße *G* sowie der Größe des Objektbildes *B*

Im verwendeten Lochkameramodell gilt:

$$\frac{B}{G} = \frac{b}{g} \tag{9.9}$$

und es folgt somit unmittelbar für den Betrag der Verschiebung der Kamerakoordinaten:

$$G = \frac{g}{b} \cdot B \tag{9.10}$$

Soll zudem eine interessante Bildregion um den Fovealisierungspunkt in einer gewünschten Größe abgebildet werden, so muss zusätzlich der Aufnahmeabstand *g* wie in Abb. 9.12 dargestellt variiert werden.

Aus

$$\frac{G}{g} = \frac{B}{b} \quad \text{und} \quad \frac{G}{g'} = \frac{B'}{b} \tag{9.11}$$

folgt wiederum unmittelbar

$$g' = \frac{B}{B'} \cdot g \tag{9.12}$$

und somit für die Variation des Aufnahmeabstandes

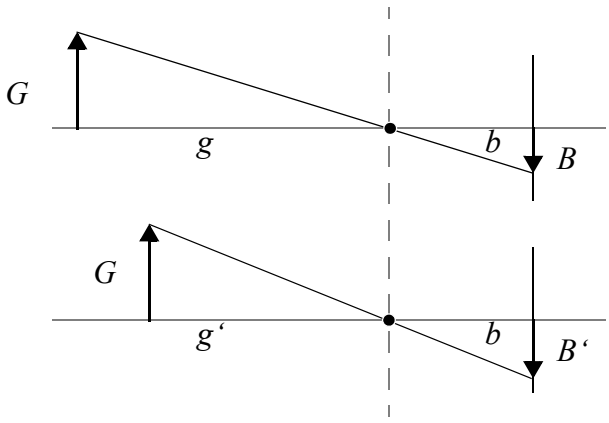


Abb. 9.12: Variation des Aufnahmeabstandes bei konstanter Brennweite

$$\Delta g = g - g' = \left(1 - \frac{B}{B'}\right) \cdot g \tag{9.13}$$

Es ergibt sich also als notwendige Bewegung im Kamerakoordinatensystem der Bewegungsvektor

$$\Delta \boldsymbol{p}_{Kamera} = \Delta \boldsymbol{p}_7 = (\Delta x, \Delta y, \Delta z)^T = \left(\frac{g}{b} \cdot x, \frac{g}{b} \cdot y, \left(1 - \frac{B}{B'}\right) \cdot z\right)^T \tag{9.14}$$

Um die im vorigen Abschnitt bestimmte Bewegung der Kamera auch ausführen zu können, muss sie auf das Koordinatensystem des Manipulators übertragen werden. Dies geschieht zunächst mittels einer Translation entlang der Kamerahalterung in den Ursprung des TCP-Koordinatensystems und eine anschließende Drehung entsprechend der Verdrehung und Verkippung der Kamera bezüglich des Tool-Center-Points. Dies wird durch die entsprechende Transformationsmatrix ${}^6A$ festgelegt.

$$\Delta \boldsymbol{p}_{TCP} = \Delta \boldsymbol{p}_6 = {}^6\boldsymbol{A} \cdot \Delta \boldsymbol{p}_{Kamera} \tag{9.15}$$

Mit Hilfe der Bewegung $\Delta \boldsymbol{p}_{TCP}$ ist nun eine Steuerung des Roboters entweder als Relativbewegung oder aber als absolute Positionierung $\boldsymbol{p}^{neu}{}_{TCP} = \boldsymbol{p}^{alt}{}_{TCP} + \Delta \boldsymbol{p}_{TCP}$ im Roboterkoordinatensystem möglich.

9.5 Die modellgetriebene Bestimmung neuer Blickpunkte

Neben einer datengetriebenen Bestimmung von Blickpunkten bietet ein wissensbasierter Modellierungsansatz die Möglichkeit, durch den Vergleich von bereits aus einer Szene extrahiertem Wissen und dem Modellwissen auch modellgetrieben interessante Blickpunkte zu bestimmen. So kann zum Beispiel nach Erkennen einer Ansicht oder einer Teilansicht eines Objektes auf die Positionen weiterer markanter Ansichten geschlossen werden. Diese Vorgehensweise und ihre Integration in die Objektmodelle soll in diesem Abschnitt näher erläutert werden.

Dabei stellt sich zunächst die Frage, welche Informationen nach dem Erkennen einer einzelnen Ansicht oder Teilansicht eines Objektes zur weiteren Auswertung zur Verfügung stehen. Dazu noch einmal ein kleiner Rückblick auf die verwendeten Operationen zur Analyse einzelner Ansichten. In Kapitel 4.2.4 war festgehalten worden, dass zunächst einmal beim Vergleich eines gelernten Prototypen mit dem präsentierten Bild die Orientierung ε des Objektes sowie dessen Position (x, y) im Bild ermittelt wird. Außerdem war davon ausgegangen, dass die Entfernung zum Objekt geschätzt ist. Desweiteren kann, wie in Kapitel 4.2.5 beschrieben, durch das Lernen verschiedener Objektansichten die komplette Ansichtensphäre eines Objektes abgedeckt werden und es werden beim Erkennen die Lagewinkel (φ, ϑ) ermittelt. Wie bereits in den vorangegangenen Abschnitten beschrieben, kann aus der Entfernung und der Bildposition bei einem kalibrierten Kamerasystem auf die Koordinaten des Objektes im Kamerakoordinatensystem geschlossen werden. Somit ist eine vollständige Bestimmung der Position $(x, y, z, \vartheta, \varphi, \varepsilon)$ der erkannten Objektansicht im Kamerakoordinatensystem möglich.

Im Objektmodell müssen nun die Informationen zur Verfügung gestellt werden, die benötigt werden, um aus der Position dieser Objektansicht auf eine definierte Referenz-Position des Objektes zu schließen. Dadurch steht anschließend genau diese Objektposition dem Erkennungssystem zur Verfügung. Somit kann dann auch auf Positionen weiterer Ansichten geschlossen werden und es können aus dem Objektmodell heraus beliebige Blickpunkte zur weiteren Analyse generiert

werden. Geklärt werden muss nun also, welche Informationen im Modell hierfür bereitgestellt werden müssen, wie diese dort repräsentiert werden und wie sie verarbeitet werden müssen.

9.5.1 Von der Ansicht zum Objekt

Zunächst einmal wird im Objekt und in jeder seiner Ansichten ein Koordinatensystem verankert. Für Ansichten, zu denen auch datengetrieben Fovealisierungspunkte geliefert werden, wird zur Vereinfachung der Ursprung dieses Koordinatensystems in den Fovealisierungspunkt gelegt. Die Lage der Achsen des Koordinatensystems für jede Ansicht bestimmt sich aus der gewünschten Blickrichtung auf das Objekt. Dabei gilt[1]:

- die *z*-Achse entspricht der Blickrichtung auf das Objekt
- die *y*-Achse entspricht der Vorzugsorientierung in der Bildebene
- die *x*-Achse ergibt sich aus *y* und *z* zum Rechtssystem

Als Referenzkoordinatensystem des Objektes wird das Koordinatensystem einer beliebigen Ansicht ausgewählt. Es hat sich hierbei als vorteilhaft herausgestellt, eine Ansicht auszuwählen, die in der betrachteten Anwendung mit großer Wahrscheinlichkeit bereits im Initialbild auftritt, da sich hierdurch die Modellbildung geringfügig vereinfacht.

Die Beziehung eines beliebigen Ansichten-Koordinatensystems zu dem Referenzkoordinatensystem des Objektes kann wiederum durch eine Verschiebung und eine Drehung des Koordinatensystems beschrieben werden. Aus diesem Grund müssen im Objektmodell genau diese Verschiebungs- und Rotationsparameter (Δx, Δy, Δz, υ, θ, η) festgelegt werden. Dabei beschreiben die Rotationsparameter folgende Drehungen:

1. Rotation um die *z*-Achse um den Winkel υ (positiv *x* nach *y*)
2. Rotation um die *y*‘-Achse um den Winkel θ (positiv *z*‘ nach *x*‘)

1. Es kann natürlich auch jede andere Lage des Koordinatensystems gewählt werden. Die hier vorgeschlagene Kopplung der *x*-*y*-Ebene an die Bildebene hat sich aber als sehr gut handhabbar erwiesen.

3. Rotation um die x‘‘-Achse um den Winkel η (positiv z‘‘ nach y‘‘)

Diese Parameter können gleichzeitig natürlich auch genutzt werden, um von einem einmal ermittelten Referenzkoordinatensystem wieder auf die Koordinatensysteme anderer Ansichten zu schließen. Die Vermessung und Festlegung der Translations- und Rotationsparameter erfolgt dabei immer im Referenzkoordinatensystem.

An einem kleinen Beispiel soll diese Vorgehensweise verdeutlicht werden. In der Abbildung 9.13 sind sowohl das Referenzkoordinatensystem (Mitte des Autos), als auch das Zielkoordinatensystem (hinteres rechtes Rad) dargestellt. Als Transformationsparameter ergibt sich daher: Δx=-20mm, Δy=-25mm, Δz=10mm, wobei der Offset in z positiv ist, weil der Ausgangspunkt hier auf der Höhe des Daches liegt, während der Zielpunkt auf Höhe der Räder liegt.

Eine Möglichkeit der Transformation des Ausgangssystems in das Zielsystem ist:

1. Drehung des Ausgangssystems um z, so dass x‘ der alten y-Achse entspricht ($\upsilon = 90°$)
2. Drehung um y‘ nicht nötig ($\theta = 0°$)
3. Drehung um x‘‘, so dass z‘‘ der alten x-Achse entspricht ($\eta = -90°$)

Mit Hilfe dieser sechs Parameter kann also vom Referenzsystem des Autos eine geeignete Blickrichtung auf das rechte Hinterrad des Autos bestimmt werden. Soll umgekehrt vom Hinterrad auf das Referenzsystem geschlossen werden, so muss beachtet werden, dass die hierfür notwendigen Umrechnungsparameter nicht einfach durch Negation bestimmt werden können. Ihre Bestimmung hat in dem Koordinatensystem des rechten hinteren Rades zu erfolgen. Die Offsetwerte würden dann folgendermaßen aussehen: Δx=-25mm, Δy=-10mm, Δz=20mm.

Für die Kamerasteuerung wird darüber hinaus neben den gerade erwähnten sechs Parametern noch die gewünschte Distanz *dist* zwischen dem Ursprung des Koordinatensystems und der Kamera benötigt.

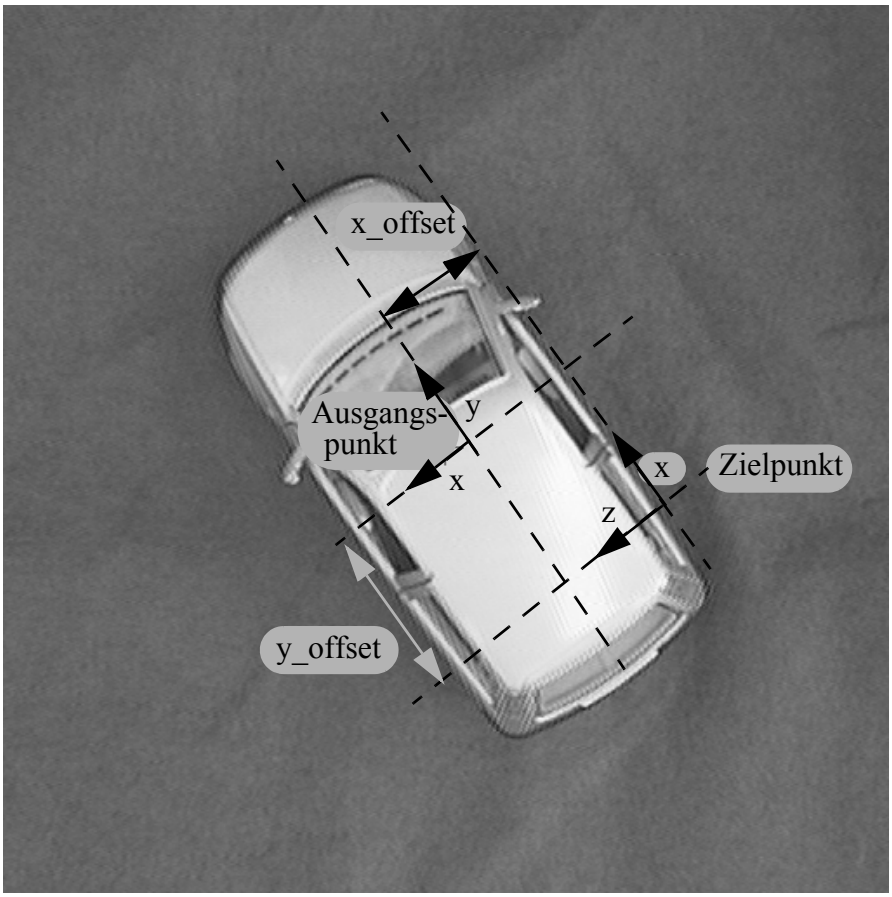


Abb. 9.13: Parameterbestimmung an dem Beispiel eines Autos

9.5.2 Die Modellierung der Fovealisierungsstrategie

Bei den vorstehenden Überlegungen war bereits deutlich geworden, dass insgesamt sieben Parameter notwendig sind, um modellgetrieben die Kamera auf neue Fovealisierungspunkte richten zu können. Die sechs Parameter Δx, Δy, Δz, υ, θ, η beschreiben dabei die Beziehung zwischen einer Objektansicht und einem Referenzkoordinatensystem, der siebte Parameter *dist* legt den Abstand zwischen Kamera und Objekt bei der Aufnahme fest. Die hierzu notwendigen Berechnungen werden über die prozedurale Schnittstelle in den (Pre-)Attributberechnungen integriert. Wenn von einer Ansicht auf das Referenzkoordinatensystem geschlossen werden soll, dient das Erkennungsergebnis, welches ja zusätzlich zur Klassifikation einer Ansicht auch Lageinformation beinhaltet, als Operand für diese Berechnung. Das re-

sultierende Operationsergebnis kann dann an anderer Stelle wieder als Operand genutzt werden, wenn vom Referenzkoordinatensystem auf eine neue Ansicht geschlossen werden soll. Die oben aufgeführten hierzu notwendigen Parameter werden dann als solche im Parameterslot der Operationsbeschreibung festgelegt.

Attribut	PositionRadHintenLinks
Typ	BOOLEAN
Wertebereich	{TRUE}
Operation	CALC_POSITION
Operand	<PositionPunto>
Parameter	Δx=-20, Δy=-25, Δz=10, υ=90, θ=0, η=-90, dist=750
formales Ergebnis	<PositionFerrari>

Abb. 9.14: Modellierung der Blickrichtung

9.6 Erreichbarkeit von Blickpunkten

Bei dem Versuch, die daten- oder modellgetrieben erzeugten Blickpunkte mit dem Roboter anzufahren, ergibt sich je nach Größe und Position des Objektes im Arbeitsraum das Problem, das die gewünschte Kameraposition nicht eingenommen werden kann. Dies ist der Fall, wenn

- die notwendige Effektorposition außerhalb des Arbeitsraumes liegt
- einzelne Gelenkwinkel aufgrund mechanischer Begrenzungen nicht eingestellt werden können
- der Effektor oder die am Effektor befestigte Kamera mit Hindernissen oder den Roboterarmen selbst kollidieren.

Da in diesen Fällen eine Beschädigung des Roboters oder zumindest ein von der Robotersteuerung hervorgerufener Stillstand des Sy-

stems eintreten würde, ist es notwendig, vor der Mitteilung einer gewünschten Position an die Robotersteuerung selbst die Erreichbarkeit des Blickpunktes zu überprüfen. Dazu wird mit Hilfe einer Rückwärtsrechnung von der notwendigen Kameraposition auf die einzelnen Gelenkstellungen des Roboters geschlossen.

Es werden die Denavit-Hartenberg-Regeln [48] zur Beschreibung der Transformationsmatrizen ${}^0\boldsymbol{A}$, ${}^1\boldsymbol{A}$, ... ${}^6\boldsymbol{A}$ verwendet. Wie bereits zuvor erwähnt beschreiben ${}^0\boldsymbol{A}$ ... ${}^5\boldsymbol{A}$ die sechs Gelenkpositionen des Roboters, während ${}^6\boldsymbol{A}$ den Übergang zur Kamera festlegt. Da diese Transformationsmatrizen für jeden Roboter jeweils wieder neu festgelegt werden müssen, sollen an dieser Stelle nur einige allgemeine Anmerkungen zu dieser Vorgehensweise gemacht werden.

Durch Multiplikation der Transformationsmatrizen und Gleichsetzen mit der gewünschten Kameraposition entsteht ein Gleichungssystem, das dann nach den gesuchten Winkeln aufgelöst wird. Diese legen nun fest, wie die einzelnen Gelenke des Roboters eingestellt werden müssen, um die gewünschte Effektorposition zu erreichen. Dabei können verschiedene Fälle unterschieden werden:

- Es kann keine Lösung des Gleichungssystems gefunden werden. Dann handelt es sich um eine *unerreichbare* Stellung. Eine solche Stellung ist beispielsweise ein Punkt im Raum, der vom Roboter soweit entfernt ist, dass er auch bei völlig ausgestrecktem Arm nicht erreicht werden kann.
- Es existiert genau eine Lösung des Gleichungssystems. Dann wird von einer *prinzipiell erreichbaren* Effektorstellung gesprochen.
- Es existieren mehrere Lösungen. Dies ist typischerweise dann der Fall, wenn der Roboter mehr als sechs Freiheitsgrade besitzt. Dann können einige der Gelenkvariablen frei gewählt werden. Aber auch bei Robotern mit kleineren Freiheitsgraden können für einzelne Effektorstellungen mehrere Lösungen existieren. In diesem Fall spricht man von *Reduktionsstellungen*.

Die Tatsache, dass eine Stellung prinzipiell erreichbar ist, bedeutet jedoch noch nicht, dass sie vom Roboter auch tatsächlich erreichbar ist, da bislang noch keine mechanischen Randbedingungen betrachtet wurden. Es muss also noch zusätzlich getestet werden, ob die berechneten

Gelenkwinkel auch tatsächlich eingestellt werden können und ob es zu keinen Kollisionen von Kamera, Effektor und Armen kommt.

Die vollständige Rückwärtsrechnung für den verwendeten Roboter vom Typ *Siemens manutec R2* findet sich in [102]. Ausführlichere Hinweise zum Thema Roboter-Kinematik sind u.a. in [164] zu finden.

10 Module zur Evaluation der Wissensbasen

Um den Benutzer beim Aufbau und der eventuell notwendigen Fehlersuche in seinen Wissensbasen zu unterstützen, steht eine graphische Benutzeroberfläche zur Verfügung, die zum einen die Darstellung der erzeugten Wissensbasen erlaubt, zum anderen aber auch eine flexible Möglichkeit bietet, die automatische Bearbeitung mit Unterbrechungspunkten zu instrumentieren, an denen die Bearbeitung stoppt oder an denen graphische Ausgaben von Operanden oder Operationsergebnissen erfolgen sollen. Diese Oberfläche wird im folgenden in ihren wesentlichen Eigenschaften dargestellt und erläutert. Da sie jedoch nicht Hauptbestandteil dieser Arbeit ist, sollen sich diese Ausführungen auf die wesentlichen Eigenschaften beschränken. Die Bedeutung einer graphischen Wissensdarstellung und graphisch unter-

stützten Auswertung eines Bearbeitungsdurchganges in wissensbasierten Bildanalysesystemen wird auch von Kummert beschrieben [99].

10.1 Die Netzwerkdarstellung

Die Modellierung komplexer Objekte oder Szenen mit Hilfe von Wissensbeschreibungssprachen ermöglicht eine explizite und strukturierte Beschreibung von Beziehungen zwischen Objekten oder Objektteilen. Besonders die Verwendung semantischer Netzwerke zur Modellierung impliziert eine grafische Darstellung dieser Beziehungen, die zu einer wesentlich verbesserten Wartbarkeit großer Wissensbasen führt.

Im Falle der in dieser Arbeit verwendeten Netzbeschreibungen lassen sich sehr gut die Standard-Relationen zwischen den verschiedenen Konzepten darstellen. An verschiedenen Stellen wurden diese Darstellungen bereits genutzt, um beispielhaft einige Modellierungen zu zeigen (vgl. etwa Kapitel 6 bis 9 sowie die folgenden Abschnitte dieses Kapitels). Dabei erfolgt - ausgehend von einem Startkonzept - eine Zuordnung der Konzepte zu Hierarchieebenen derart, dass keine horizontalen Kanten in der Netzgrafik auftreten können. Diese Zuordnung folgt im wesentlichen kanonisch der hierarchischen Modellierung, in der Strukturen durch immer feinere Strukturen beschrieben werden. Im Anschluss an die Bestimmung der breitesten Hierarchieebene erfolgt eine zentrierte Anordnung der Konzepte innerhalb der übrigen Ebenen. Zwei Kantentypen zwischen den durch beschriftete Rechtecke beschriebenen Konzepten symbolisieren die beiden bereits erwähnten Standardrelationen. Auf eine völlige Kreuzungsfreiheit der Kanten wurde verzichtet, da die grafische Anordnung der Konzepte von links nach rechts innerhalb einer Hierarchieebene eine wesentliche Information der Modellierung und der damit verbundenen automatischen Abarbeitung des Netzes darstellt. In einzelnen Fällen können bei der Modellierung von Objektstrukturen auch Hierarchieebenen in Teilnetzen übersprungen werden. Dies würde unter Umständen zu einem Kreuzen einer Kante mit einem Konzept führen. Um dies vermeiden zu können, wird eine solche Kante geteilt und in der entsprechenden Ebene ein nicht sichtbares virtuelles Konzept eingeführt, durch dessen Position

innerhalb der Hierarchieebene die Kante läuft. Sie kann unter Umständen auch eine Richtungsänderung erfahren. Die Kanten der Standardrelationen sind zusätzlich mit einer Beschriftung versehen, die die modellierte minimale und maximale Anzahl von Teil- oder Spezialisierungsinstanzen angibt sowie während der Bearbeitung die jeweils zu einem bestimmten Zeitpunkt erzeugte Anzahl. Die Detailmodellierung eines Konzeptes bestehend aus den Attributbeschreibungen, den Bewertungsslots und dem Zielslot wird bei Bedarf in einem zusätzlichen Fenster textuell dargestellt. Eine ausführliche Beschreibung der verwendeten Verfahren findet sich in [15].

Aus der Literatur bekannte Verfahren zur automatischen Graphendarstellung - einen sehr guten Überblick zu diesem Thema gibt [186] - gingen nur eingeschränkt in die verwendeten Verfahren ein, da die in dieser Arbeit vorgestellte Wissensbeschreibungssprache mit ihren Standardrelationen eine sehr spezielle Graphenklasse erzeugt. Zudem verdeutlicht die Anordnung der Konzepte bereits deren semantische Bedeutung, so dass besondere Restriktionen für ihre Darstellung gegeben sind. Das beschriebene Verfahren greift jedoch den von Sugiyama vorgeschlagenen Ansatz zur Formatierung gerichteter zyklenfreier Graphen auf [185].

Bis hierher wurde lediglich eine statische Darstellung der Wissensbasen vorgestellt. Im folgenden soll nun auch auf eine dynamische Darstellung zur Programmausführung eingegangen werden, die eine grafische und damit benutzerfreundliche Kontrolle der automatischen Bearbeitung ermöglicht.

10.2 Unterbrechungspunkte und Online-Analyse

Bevor ein geeignetes Verfahren zur Beeinflussung der Netzauswertung ausgewählt werden kann, ist es notwendig zu formulieren, welcher Zweck hiermit verfolgt wird. Die bislang mit dem Umgang von automatisierten Auswerteverfahren gemachten Erfahrungen zeigen sehr deutlich, dass es notwendig ist, dem Knowledge Engineer ein Hilfsmittel zur Verfügung zu stellen, mit dem er sich schnell einen Überblick über die von ihm erzeugte Wissensbasis verschaffen kann. Im Bereich der semantischen Netzwerke bietet sich hierbei zwangsläufig eine gra-

fische Darstellung der Netze an. Diese ermöglicht eine einfache Übersicht über die modellierten Objekte und ihre Beziehungen untereinander. Dies betrifft zunächst einmal eine statische Betrachtung der Wissensbasis. Aber auch während der Auswertung kann eine grafische Darstellung der Aktivitäten die Fehlersuche erleichtern und zu einem größeren Verständnis für die implementierte Vorgehensweise beitragen, wenn es gelingt, die einzelnen Bearbeitungsschritte sichtbar zu machen. Aus diesem Grund erfolgen an den wesentlichen Bearbeitungspunkten Veränderungen in der grafischen Darstellung der Wissensbasis, die durch geeignete Ausgaben im Grafikfenster unterstützt werden. Sie ermöglichen es, den Kontrollfluss zu beobachten sowie die Berechnung der Attribute zu verfolgen. Um dies im Detail nachvollziehbar zu machen, ist es jedoch notwendig, die Bearbeitung bei Bedarf zu unterbrechen oder zu verlangsamen. Die Testumgebung dient somit nicht zu einem Debuggen auf Sourcecode-Ebene innerhalb des Kontrollmoduls, sondern zu einem Debuggen auf Ebene der Wissensbasis. Die wichtigsten Bearbeitungspunkte werden im folgenden kurz beschrieben.

Auf Konzeptebene stehen für jedes Konzept einzeln einstellbar folgende markante Bearbeitungspunkte zur Verfügung:

- ❒ Suchen nach einer berechneten, aber zur Zeit freien Instanz
- ❒ Start des Instanziierungsversuchs eines Konzeptes
- ❒ Erfolgreiche bzw. fehlgeschlagene Instanziierung des Konzeptes
- ❒ Freigabe einer Instanz

Außerdem wurden auf Attributebene ebenfalls für jedes Attribut einzeln steuerbar verschiedene Punkte ausgewählt:

- ❒ Start der Bearbeitung eines Attributes
- ❒ Suche nach den Operanden des Attributes
- ❒ Suche nach dem potentiellen Ergebnis im Metaspeicher
- ❒ Operationsaufruf
- ❒ Wertebereichsüberprüfung
- ❒ Erfolgreiche bzw. erfolglose Beendigung der Berechnung

Zusätzlich kann die Darstellung der Operanden und der Operationsergebnisse in einem weiteren Fenster aktiviert werden, sofern es sich hierbei um darstellbare Werte handelt, wie z.B. aufgenommene Kamerabilder oder berechnete Merkmalsvektoren. Als Unterbrechungstypen wird unterschieden zwischen einem Fortsetzen nach Tastatur- oder Mauseingabe oder einer einstellbaren zeitlichen Verzögerung der Bearbeitung. Letzteres bietet sich vor allem zu Vorführungszwecken an.

An den erwähnten Unterbrechungspunkten können zusätzliche Ausgaben gewählt werden. So kann z.B. explizit dargestellt werden, welche Attribute von welchem Konzept als Operanden für die Attributberechnung dienen, welches Ergebnis die Operation liefert und wie der Wertebereich für dieses Attribut definiert worden war. Diese Ausgaben bieten dem Knowledge Engineer eine sehr gute Testumgebung für die Evaluierung seiner Modelle.

Neben der Möglichkeit, diese Einstellungen für jedes Konzept bzw. Attribut einzeln durchzuführen, können auch netzwerkweite Einstellungen oder operationsspezifische Einstellungen gewählt werden. Dem Benutzer werden hierfür jeweils Listen der im Netzwerk modellierten Konzepte und Attribute sowie der verwendeten Operationen angeboten, aus denen eine Auswahl getroffen werden kann, für die die entsprechenden Einstellungen durchgeführt werden. Letzteres bietet sich vor allem dann an, wenn man am Verhalten einer bestimmten Operation besonders interessiert ist. So kann z.B. für jede Bildaufnahme die Darstellung des Kamerabildes veranlasst werden.

Die zuvor beschriebenen Unterbrechungspunkte und die durch diese möglichen zusätzlichen Ausgaben sind ein wichtiger Bestandteil der Online-Analyse der Netzauswertung, die optional vom Knowledge Engineer eingestellt wird. Durch eine fest installierte Darstellung der aktiven Konzepte und der untersuchten Pfade dorthin ist es möglich, sich einen Überblick über das Fortschreiten der Auswertung zu verschaffen. Hierzu werden die bearbeiteten Konzepte farblich markiert, so dass zwischen folgenden Zuständen unterschieden werden kann:

- ❒ Das Konzept wurde noch nicht bearbeitet.
- ❒ Das Konzept wurde zuletzt erfolgreich bearbeitet.
- ❒ Das Konzept wurde erfolglos bearbeitet.

Weiterhin wird online in der Konzeptdarstellung notiert, wie viele Instanzen eines Konzeptes bislang erzeugt wurden, aufgeschlüsselt nach der Anzahl der aktuell in das Netz eingebundenen und derzeit nicht verwendeten Instanzen.

Abbildung 10.1 zeigt beispielhaft die Netzdarstellung während der Bearbeitung des Konzeptes *Rad*. Zudem wurde für den Unterbrechungspunkt festgelegt, dass das Kamerabild dargestellt wird und dass das Detektionsergebnis in das Bild eingeblendet wird. Man erkennt, dass das Attribut *Detect* zur groben Lokalisierung eines Rades in der Szene erfolgreich berechnet werden konnte, d.h. es wurden eine oder mehrere mögliche Radpositionen bestimmt.

In Abbildung 10.2 ist die Auswertung der Szene bereits weiter fortgeschritten. Über die Konzepte *Szenenmodell*, *Szene*, *Rad*, *Kranz-4-Loch* wurde das Konzept *Schrauben* erreicht, von dem zur Zeit eine erste Instanz von vier benötigten bearbeitet wird. Auch hier konnte eine erste Position in verbesserter Auflösung detektiert werden. Das Resultat der Operation zur Schraubendetektion *True* wird mit dem modellierten Wertebereich verglichen. Bei Verwendung eines Farbmonitors werden zusätzlich noch die Konzeptrahmen je nach Instanziierungszustand in unterschiedlichen Farben dargestellt.

Im weiteren Verlauf des Instanziierungsprozesses muss dann nach erfolgreicher Detektion der Schrauben im Konzept *Kranz-4-Loch* die geometrische Anordnung der Schraubenhypothesen überprüft werden. Dies geschieht im Attribut *Testgeo*. Abbildung 10.3 zeigt, dass die ersten vier ausgewählten Schraubeninstanzen nicht die modellierten Eigenschaften aufweisen. Die Überprüfung der Geometrie liefert nicht das gewünschte Ergebnis. Der Suchalgorithmus wird daher später einen anderen Zweig des Suchbaums verfolgen.

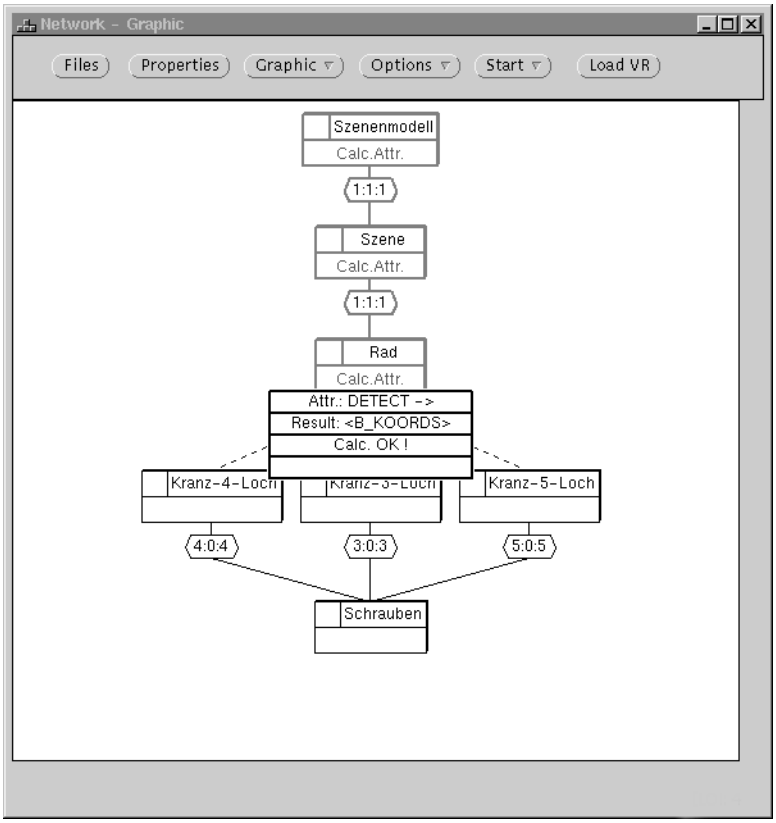


Abb. 10.1: : Netzdarstellung mit Unterbrechungspunkt während der Erkennung eines Rades mit Einblenden des detektierten Radmittelpunktes in das Kamerabild

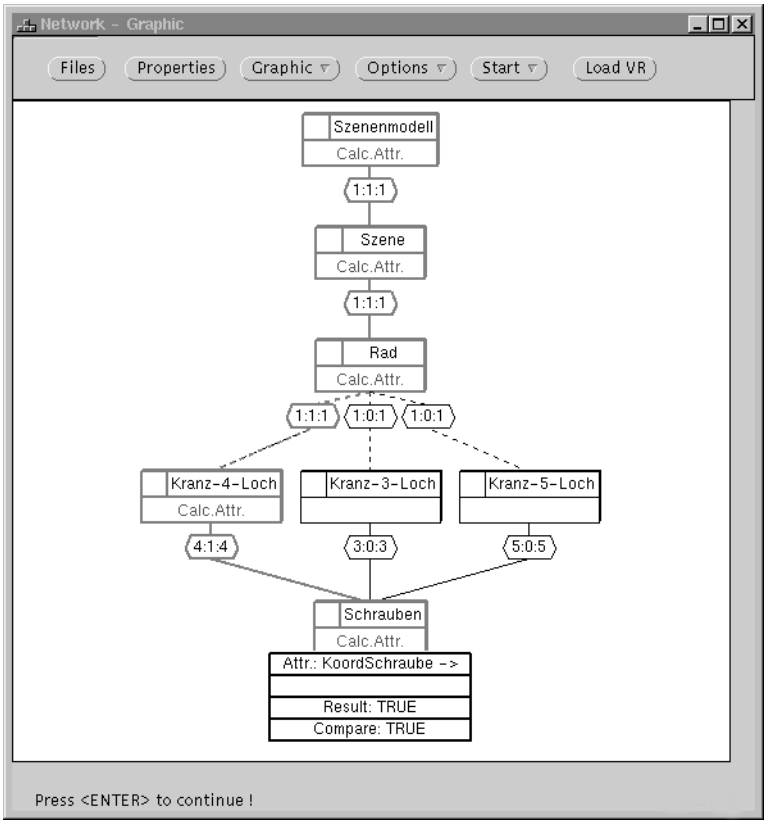


Abb. 10.2: : Netzdarstellung bei der Wertebereichsüberprüfung der Schraubendetektion

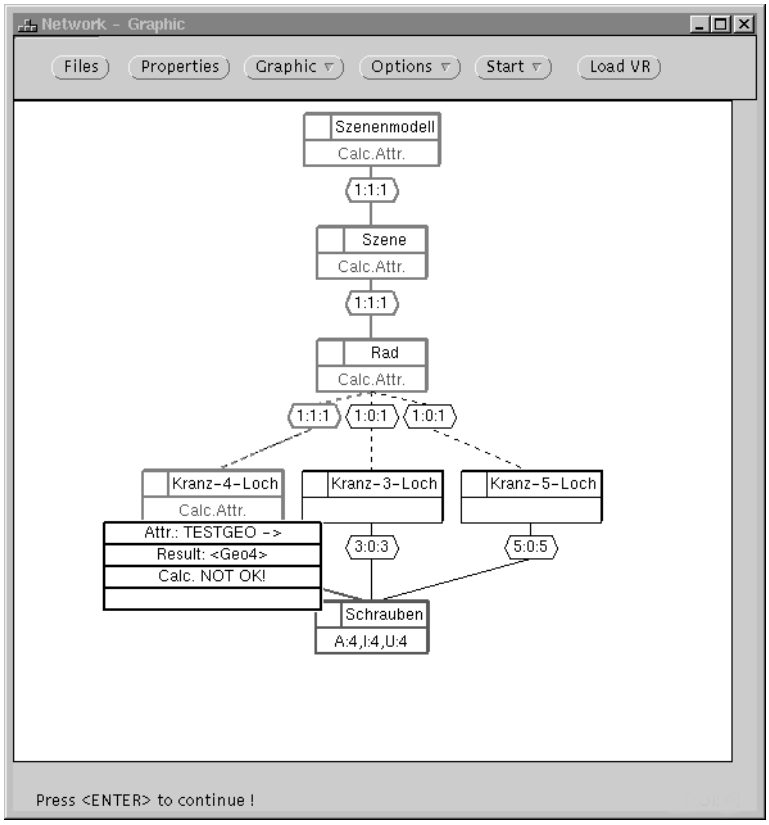


Abb. 10.3: : Netzdarstellung bei der Geometrieüberprüfung

10.3 Offline-Analyse instanziierter Netzwerke

Zusätzlich zur beschriebenen Online-Verfolgung der Bearbeitung steht eine Offline-Analyse des Instanzenbaumes zur Verfügung. Nach der Instanziierung der Konzepte können einzelne Instanzen eines gewünschten Konzeptes selektiert und detailliert untersucht werden. Hierzu gehören folgende Punkte:

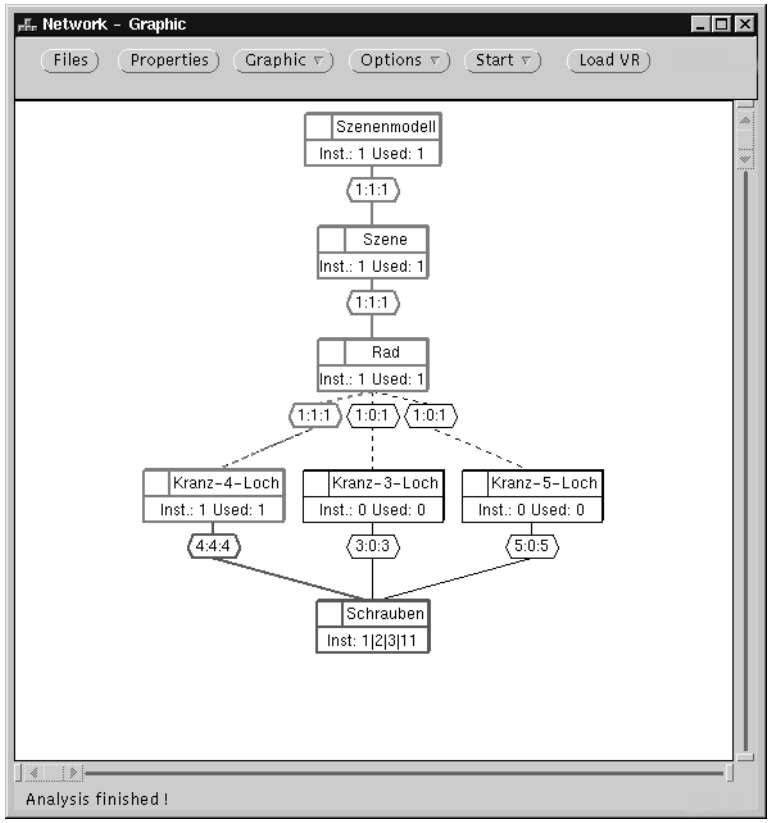
- ❒ Darstellung des Instanziierungspfades
- ❒ Darstellung der Teilinstanzen
- ❒ Darstellung der berechneten Bewertung
- ❒ Werte und Bewertungen der zu dieser Instanz gehörenden Attribute

- ❒ Grafische Darstellung von Attributergebnissen (z.B. Kamerabild, segmentiertes Bild, Merkmalsvektoren)
- ❒ Speicherung von Attributergebnissen in Dateien mit festgelegtem Dateiformat

Abbildung 10.4 zeigt an einem Beispiel einige der oben angeführten Möglichkeiten nach Beendigung der automatischen Netzbearbeitung.

Es wurde hierzu Instanz Nummer 1 des Konzeptes *Kranz-4-Loch* ausgewählt. Neben der Bewertung der Instanz wird auch der Instanziierungspfad dargestellt. Es wird deutlich, dass zur Instanzbildung auf vier verschiedene Schraubeninstanzen zugegriffen wurde. Es sind dies die Instanzen 1, 2, 3 und 11. Bei Bedarf besteht zudem die Möglichkeit, sich die einzelnen Attributergebnisse der Instanz anzeigen zu lassen. Dies geschieht zum einen textuell, bei Bildern (Kamerabild, Filterantworten, etc.) aber auch graphisch.

Eine ausführliche Beschreibung von Online- und Offline-Auswertung wird in [16] gegeben. Als Erweiterung zu der hier beschriebenen Funktionalität des Grafikmoduls existiert zusätzlich ein vollständig grafikbasierter Editor zur Erzeugung der Wissensbasen.



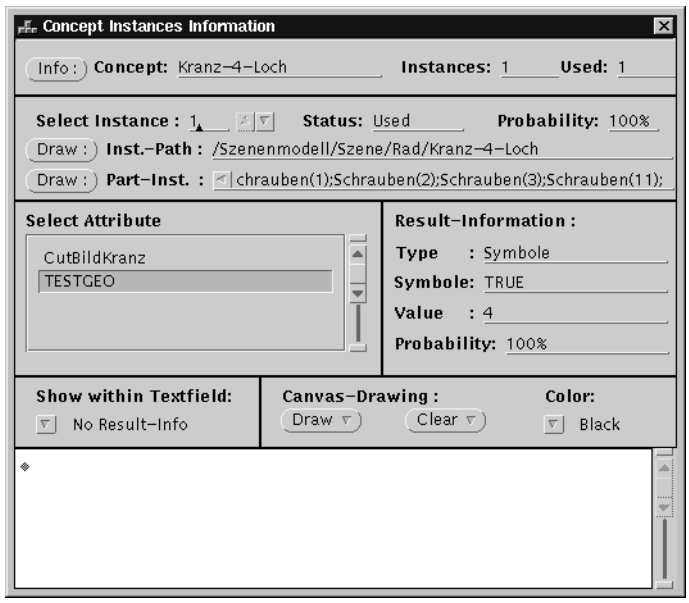


Abb. 10.4: : Offline-Informationen zur ersten Instanz des Konzeptes *Kranz-4-Loch*

10.4 Simulation der aktiven Bildaufnahme

Bei dem Aufbau des aktiven Bilderkennungssystems hat sich neben der Notwendigkeit von Hilfsmitteln zur Analyse der Wissensbasis ein weiteres sehr wichtiges Problem herauskristallisiert. Für Tests und Probeläufe muss in einem aktiven System jeweils auf die entsprechenden Ressourcen wie Roboter, Schwenk-Neigekopf und Kameras zugegriffen werden, die aber, da leider nur in einmaliger Ausfertigung vorhanden, von verschiedenen Arbeitsgruppen und Projekten verwendet werden. Aus diesem Grunde wurde eine Simulationsumgebung geschaffen, in der die aktive Kamera - sei dies nun der Kamerastereokopf oder die am Roboter befestigte Kamera - durch ein VR-Modul ersetzt wird. Dazu können Objekte, die im DXF-Format vorliegen, zu Szenen komponiert werden. Das VR-Modul berechnet dann anhand der aktuellen Kameraparameter (Position, Blickrichtung, Brennweite) das aus der Szene sich ergebende Bild. Dadurch kann auf den Einsatz der realen Kameras verzichtet werden, wenn es darum geht, die prinzipielle Funktionsfähigkeit einer neuen Operation oder einer Änderung an der Ablaufsteuerung oder dem Suchverfahren zu testen. Dies ersetzt natürlich nicht die Tests mit der realen Kamera, wenn es um die Bestimmung von Erkennungsraten geht, da die virtuell erzeugten Bilder nicht die realen Beleuchtungsvariationen, Oberflächenbeschaffenheiten, etc. widerspiegeln. Abbildung 10.5 zeigt einzelne Bilder einer aus drei Objekten komponierten Szene.

Bei Einsatz der Roboterkamera besteht darüber hinaus die Möglichkeit, die Bewegungen des Roboters beim Anfahren der verschiedenen Blickpunkte im Erkennungsprozess zu simulieren. Auch hierzu wird auf das VR-Modul zugegriffen, wobei zur Bestimmung der Gelenkwinkel in der Animation auf die in Kapitel 9.6 beschriebene Rückwärtsrechnung des Roboters eingesetzt wird. Mit Hilfe dieser Animation kann während der Tests bereits beobachtet werden, welche Bewegungen durchgeführt werden, ob diese plausibel sind und ob Probleme mit der Erreichbarkeit einzelner Blickpunkte zu erwarten sind.

Die hier kurz in ihren wesentlichen Eigenschaften vorgestellte grafische Benutzerschnittstelle des wissensbasierten Systems mit seinen Debbuging-Eigenschaften hat wesentlich zum Erfolg der Anbindung

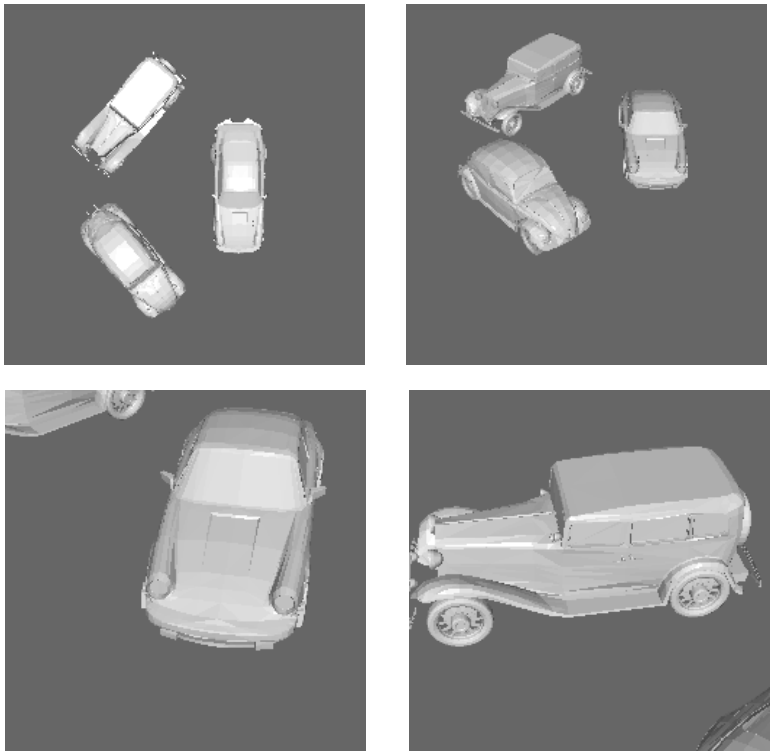


Abb. 10.5: Bildmaterial aus einer virtuellen Szene.

der holistischen Erkennungsmethoden beigetragen. Die Möglichkeit, die automatische Bearbeitung der semantischen Netzwerke online mit grafischen Hilfsmitteln zu verfolgen und den Prozess an interessanten Bearbeitungspunkten zu unterbrechen, hat die Entwicklung der notwendigen Algorithmen sowie deren Testphase wesentlich beschleunigt. Darüber hinaus hat sie sich auch beim täglichen Umgang mit dem Erkennungssystem und bei der Modellierung der verschiedenen Objekte als unverzichtbares Hilfsmittel erwiesen.

Die Verwendung künstlicher, mit Hilfe des VR-Moduls erzeugter Bilder hat darüber hinaus den Vorteil bei der Fehlersuche Abläufe reproduzierbar gestalten zu können. Dies ist bei natürlichem Bildmaterial als Eingabe aufgrund der normalen Variationen im Bild durch Rauschen oder Beleuchtungsvariation nicht immer möglich.

11 Automatische Erzeugung hybrider Objektmodelle

Neben der zentralen Frage, wie das Wissen über Objekte in einem Bilderkennungssystem repräsentiert werden soll, ist auch die Frage, wie dieses Wissen in das System eingebracht werden kann, von entscheidender Bedeutung für die Akzeptanz und Einsetzbarkeit eines solchen Systems. Dies betrifft im besonderen die Erkennung komplexer dreidimensionaler Objekte. Das Erzeugen von Objektmodellen benötigt dabei häufig sehr umfangreiche Kenntnisse über die zu untersuchende Domäne, um das benötigte Wissen entsprechend strukturieren zu können.

11.1 Lernverfahren

Verfahren zur automatischen Wissensakquisition sollen hierbei unterstützend wirken oder sogar diese Aufgabe vollständig übernehmen. Bei der Einteilung der existierenden Lernverfahren können zwei Klassen unterschieden werden:

- *symbolische Lernverfahren*, die das in den Beispielen enthaltenen Wissen in eine explizite, symbolische Beschreibung der Objekte transformieren, z.B. in der Form von Regeln, Entscheidungsbäumen, semantischen Netzwerken, etc. [35], [121], [128]
- *subsymbolische Lernverfahren*, die das Wissen aus den Beispielen in eine implizite Repräsentation überführen, wie dies z.B. bei den verschiedenen neuronalen Netzwerktypen der Fall ist [74], [155], [156].

Beiden Lerntypen gemeinsam ist jedoch, dass üblicherweise eine große Zahl an Trainingsdaten zur Verfügung gestellt werden muss, was den Lernvorgang entsprechend aufwendig macht. Dagegen ist es unser Ziel, bereits durch einmaliges Präsentieren eines Objektes, ein Objektmodell aufzubauen, das für das Wiedererkennen unter veränderten Bedingungen (Position, Beleuchtung, etc.) geeignet ist. Es handelt sich somit um ein *induktives Lernverfahren*, bei dem das Modellwissen aus Beispielen extrahiert wird. Einen Überblick über verschiedene Lernverfahren geben [34] und [129]. Aus der nachstehenden Beschreibung des gewählten Verfahrens wird hervorgehen, dass es sich dabei um ein *unüberwachtes Lernverfahren* handelt, bei dem also kein Lehrer benötigt wird[1]. Hervorzuheben sind zwei wichtige Eigenschaften des Lernvorgangs.

1. Sowohl auf der subsymbolischen als auch auf der symbolischen Ebene der Modellbildung im hybriden System erfolgt ein

1. Da bei dem hier besprochenen Lernverfahren symbolische Modellbeschreibungen erzeugt werden, ist es aber natürlich sinnvoll, wenn ein Benutzer während des Lernvorganges für die erzeugten Konzepte geeignete, d.h. dem Sprachgebrauch angepasste symbolische Bezeichner vergibt.

unüberwachtes, induktives Lernen durch Präsentieren nur eines einzigen Beispiels.

2. Der Lernvorgang selbst ist als ein *aktiver Vorgang* ausgelegt, bei dem sich das System selbst die notwendigen Ansichten auswählt und die hierzu gehörenden Bilder aufnimmt.

11.2 Charakteristische Ansichten

Aufgrund der Modellierung der zu erkennenden Objekte auf der Basis einiger charakteristischer Objekt- und Detailansichten, liegt die Hauptaufgabe der automatischen Modellgenerierung zunächst einmal in der Extraktion und Auswahl dieser charakteristischen Ansichten. Dabei geht es nicht darum, eine Auswahl von Ansichten zu finden, die die Kontur oder die Oberfläche eines Objektes vollständig abdeckt und beschreibt. Es geht auch nicht darum, eine Auswahl zu treffen, die keinerlei redundante Information über ein Objekt enthält. Es ist vielmehr Ziel, eine kleine Menge von Ansichten auszuwählen, die geeignet ist, ein Objekt von den anderen Objekten der Domäne zu unterscheiden. Hierzu ist eine vollständige Beschreibung nicht notwendig, während sich überschneidende Ansichten geeignet sind, im Falle von teilweisen Verdeckungen die Objekterkennung zu unterstützen. In der Verwendung von Detailansichten für die Objekterkennung unterscheidet sich dieser Ansatz auch von dem in [69] beschriebenen Ansatz. Er wurde erstmalig in [29] vorgestellt.

Im folgenden seien zunächst einmal einige Begriffs-Definitionen gegeben.

Definition 11.1: Das Bild eines Objektes aus einer beliebigen Blickrichtung nennen wir **Objektansicht**, wenn das Objekt vollständig im Kamerabild enthalten ist.

Definition 11.2: Das Bild eines Objektes aus einer beliebigen Blickrichtung nennen wir **Detailansicht**, wenn lediglich ein Teil des Objektes im Kamerabild enthalten ist. Dabei ist die Aufnahmeentfernung typischerweise

kleiner als bei einer Objektansicht. Dies ist aber nicht zwingende Voraussetzung.

Definition 11.3: Wir sprechen allgemein von einer **Ansicht** eines Objektes, wenn es sich um eine Objektansicht oder eine Detailansicht handelt.

Definition 11.4: Wir nennen eine Ansicht A **hart charakteristisch** für eine Ansichtenmenge $A(K)$ einer Objektklasse K bezüglich eines Ähnlichkeitsmaßes d, wenn

1. zu jedem Objekt der Klasse K genau eine Ansicht $A' \in \mathrm{A}(K)$ mit großer Ähnlichkeit zu A existiert und wenn
2. für alle Objekte aus einer anderen Klasse $K' \neq K$ keine Ansicht $A' \in \mathrm{A}(K')$ mit hinreichend großem Ähnlichkeitsmaß zu A existiert.

Hinweis: Für die Festlegung eines großen Ähnlichkeitsmaßes wird verlangt, dass $d(A', A) > d_{min}$. Das Ähnlichkeitsmaß d und der Schwellwert d_{min} werden dabei üblicherweise durch den bei der Erkennung verwendeten Klassifikator bestimmt.

Definition 11.5: Wir nennen eine Ansicht A **weich-** oder **k-charakteristisch**, wenn in den Bedingungen 11.4.1 und 11.4.2 jeweils nur wenige, k Ansichten A_i' mit hinreichend großem Ähnlichkeitsmaß existieren.

Hinweis: Wenn im folgenden allgemein von charakteristischen Ansichten eines Objektes gesprochen wird, so sind damit weich- oder k-charakteristische Ansichten gemeint.

Es sei an dieser Stelle darauf hingewiesen, dass somit der Ansichten-Begriff vom Aspekt-Begriff unterschieden wird. Die Menge der Aspekte eines Objektes ist eine echte Teilmenge der Ansichtenmenge.

11.3 Auswahl charakteristischer Ansichten

Im folgenden wird nun betrachtet, wie näherungsweise aus der Ansichtenmenge eine geeignete Menge charakteristischer Ansichten für die Modellbildung extrahiert werden kann. Da sich der Lernprozess natürlich an den Erkennungsvorgang anlehnt, werden zur Erzeugung der Ansichten im ersten Schritt die Verfahren der subsymbolischen Erkennungsebene verwendet. Es werden also mit Hilfe der Fovealisierungsroutinen Aufmerksamkeitspunkte bestimmt, die von der Kamera angefahren werden. Die auf diese Weise aufgenommenen Detailansichten werden dann wie in Kapitel 4.2.4 beschrieben transformiert und gelernt. Mit Hilfe des Klassifikators wird anschließend bestimmt, wie wahrscheinlich eine Fehlklassifikation eines einzelnen Merkmalsvektors innerhalb der betrachteten Domäne ist. Lediglich die Merkmalsvektoren, die nur sehr selten fehlklassifiziert werden, werden als charakteristisch für ein Objekt betrachtet (*weich*- oder *k*-charakteristisch nach Definition 11.5). Diese Vorgehensweise beschränkt sich zunächst einmal auf die Auswertung von Bildern aus einer Blickrichtung.

Für die Modellierung komplexer 3D Objekte sollen jedoch auch Objektansichten aus verschiedenen Blickrichtungen verwendet werden. Aus diesem Grunde wird zusätzlich mit einer vom Benutzer vorzugebenden Schrittweite die Ansichtensphäre eines Objektes abgetastet. Für jede dieser Objektansichten wird dann die oben beschriebene Vorgehensweise zur Erzeugung weiterer Detailansichten und deren Auswahl durchgeführt.

Auf diese Weise werden zunächst einmal verschiedene Objekt- und Detailansichten eines oder mehrerer Objekte der Domäne erzeugt. Dieses sind natürlich nicht alle Ansichten nach Definition 11.3 sondern nur eine kleine Teilmenge. Die Erzeugung aller Ansichten würde bei kleiner Diskretisierung der Ansichtensphäre zu einer nicht handhabbaren Menge von Ansichten führen. Wir müssen uns daher auf eine Teilmenge begrenzen. Die gewählte Einschränkung auf die Ansichten, die mit Hilfe von Fovealisierungsroutinen erzeugt werden können, ist sinnvoll, da auch bei der Erkennung primär diese „fovealisierbaren“ Ansichten

ausgewertet werden[1]. Abbildung 11.1 verdeutlicht den Prozess der Ansichtenauswahl.

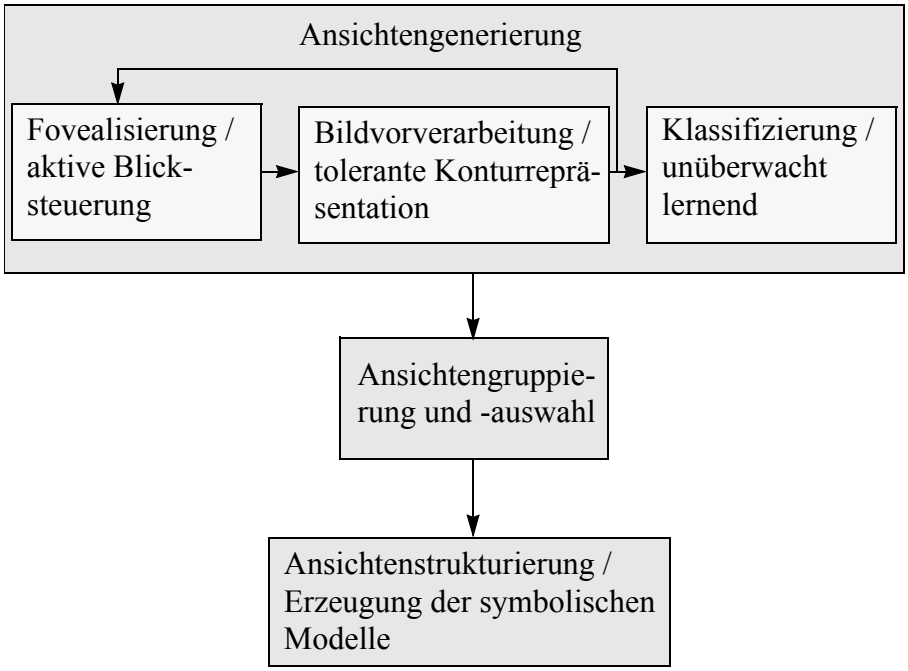


Abb. 11.1: Struktur des Lernvorganges

Im folgenden soll die Auswahl der Ansichten in formaler Form beschrieben werden. Wir betrachten zunächst einmal die Verarbeitung einer einzelnen Ansicht A eines Objektes O. Es seien nun $F(A) = \{f_1, f_2, \ldots, f_n\}$ die Menge der Fovealisierungspunkte im Bild der Ansicht A, die für weitere detaillierte Aufnahmen betrachtet werden sollen. Es sei $N(A)$ die Menge der normalisierten Detailansichten, die sich durch Anfahren der Fovealisierungspunkte ergeben. Hierbei werden Detailbilder mit konstanter Blickrichtung, aber veränderter Distanz aufgenommen. Die Normalisierung erfolgt bei den Detailbildern auf eine verringerte Referenzdistanz r', so dass eine bessere Auflösung der Details gewonnen wird. Die Merkmalsextraktion liefert uns Aktivitätsmuster der komplexen Neurone als Merkmalsvektoren $T(A) = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_n\}$. In Analogie zur subsymbolischen Klassifikation verwenden wir für die Auswertung der Merkmalsvektoren das Ähnlichkeitsmaß $d_{ij} = d(\boldsymbol{t}_i, \boldsymbol{t}_j)$, das sich aus dem *inneren Produkt* minus der

1. Die Beschränkung auf fovealisierbare Ansichten ist nicht zwingend, da die Erkennung mit Hilfe der General-Hough-Transformation (Kap. 4.2.7) translationsinvariant erfolgt. Bei einer Beschränkung auf die Umgebung eines Fovealisierungspunktes verkleinert sich jedoch sowohl die Größe des Hough-Akkumulators als auch die Laufzeit wesentlich.

Hamming-Distanz der beiden binären Merkmalsvektoren ergibt. Dieses Ähnlichkeitsmaß wird als Näherung für die Wahrscheinlichkeit betrachtet, dass zwei Merkmalsvektoren, respektive die zugrundeliegenden Ansichten, verwechselt werden können. Mit $d_{ij} = d(\boldsymbol{t}_i, \boldsymbol{t}_j)$ ergibt sich die Wahrscheinlichkeitsmatrix $\boldsymbol{D} = [d_{ij}]$. Der Zeile i dieser Matrix kann man nun also entnehmen, wie ähnlich die durch den Merkmalsvektor $\boldsymbol{t}_i$ repräsentierte Ansicht zu anderen Ansichten ist.

Durch eine Binarisierung mit Hilfe der Schwellwertfunktion

$$u : \Re \rightarrow \{0,1\} \text{ mit } u(x) = \begin{cases} 1 \text{ für } x > 0 \\ 0 \text{ für } x \leq 0 \end{cases} \tag{11.1}$$

erhalten wir durch Verwendung des Schwellwertes d_{min} die Matrix

$$\boldsymbol{S} = [s_{ij}] \in \{0,1\}^{n \times n} \text{ mit } s_{ij} = u(d_{ij} - d_{min}) \tag{11.2}$$

Für unsere Experimente verwenden wir dabei $d_{min} = 0.8$. Hierdurch kann nun in einer Zeile i durch einfache Summation der Zeilenelemente „gezählt“ werden, zu wievielen Verwechslungen die entsprechende Ansicht neigt. Wir betrachten hierzu nun die Zeile $\boldsymbol{s}_i = (s_{i1}, s_{i2},...,s_{in})$ mit der Norm

$$\|\boldsymbol{s}_i\|_1 = \sum_j s_{ij} . \tag{11.3}$$

Da zumindest für die Diagonalelemente $s_{ii} = 1$ gilt, folgt unmittelbar $\|\boldsymbol{s}_i\|_1 \geq 1$.

Mit Hilfe dieser Werte $\|\boldsymbol{s}_i\|_1$ können wir nun drei verschiedene Klassen von Merkmalsvektoren $\boldsymbol{t}_i$ unterscheiden:

1. $\|\boldsymbol{s}_i\|_1 = 1$: $\boldsymbol{t}_i$ wird nicht mit anderen Merkmalsvektoren verwechselt und trägt damit Information zur Orientierung und Lageschätzung des Objektes O.
2. $1 < \|\boldsymbol{s}_i\|_1 < s_{max}$ mit einem Schwellwert s_{max}: Die Verwendung von $\boldsymbol{t}_i$ zur Orientierungs- und Lageschätzung von O kann zu falschen Hypothesen führen, da eine Verwechselungsgefahr mit anderen Merkmalsvektoren besteht. Der Merkmalsvektor $\boldsymbol{t}_i$ eig-

net sich jedoch zur Überprüfung mit Hilfe zuvor aufgestellter Hypothesen, da nur eine geringe Wahrscheinlichkeit zu Verwechslungen besteht. Der Einfluss dieser Ansichten auf die Erkennung wird durch eine Verringerung der Glaubwürdigkeit (Evidenz) in den Bewertungsslots verkleinert. Wir verwenden als Schwellwert $s_{max} = n/10$.

3. $\|\boldsymbol{s}_i\|_1 \geq s_{max}$: Aufgrund der großen Verwechslungsgefahr sollte $\boldsymbol{t}_i$ nicht für die Erkennung von O verwendet werden.

Offensichtlich lässt sich dieses Verfahren direkt auf die Bearbeitung mehrerer Objekte O_k verallgemeinern. Dabei werden zunächst zu jedem Objekt eine Objektansicht und die hieraus resultierenden Detailansichten zu allen Objekten betrachtet. Für die 3D Verarbeitung werden darüber hinaus auch mehrere Objekt- und Detailansichten A_{kl} dieser Objekte in die Auswahl einbezogen.

11.4 Generierung von Objektmodellen

Die Zweiteilung des Generierungsprozesses in eine erste Phase zur Auswahl der charakteristischen Ansichten und in eine zweite Phase der eigentlichen Modellbildung ermöglicht die einfache und schnelle Anpassung an Änderungen der Modellierungssyntax, an veränderte Modellierungsstrategien und auch die Verwendung für völlig andere Modellierungssprachen. Bei der in dieser Arbeit vorgestellten Wahl von semantischen Netzwerken für die Objektmodellierung werden die charakteristischen Objektansichten über eine *Aspekt-Relation* integriert, während die charakteristischen Teilansichten jeweils über eine *Teil-von-Relation* an die zugehörige Objektansicht gebunden werden.

Jeder Ansicht wird dabei ein Konzept des Netzwerkes zugeordnet. Dabei werden die charakteristischten Detailansichten - d.h. die Detailansichten, die die wenigsten Verwechslungen aufweisen - als erstes eingetragen. Wie bei der Beschreibung des Instanziierungsalgorithmus dargestellt, werden die als erstes eingetragenen Teil-Konzepte auch als erstes bearbeitet. Somit werden Details mit geringster Verwechslungs-

wahrscheinlichkeit als erstes bearbeitet und es können sehr schnell aussagekräftige Hypothesen gebildet werden.

In den Attributen der Konzepte wird vermerkt, mit welchen Operationen (Fovealisierung, Merkmalsextraktion, Klassifikator) die jeweilige Ansicht bearbeitet wurde. Die bei der Ansichtengenerierung verwendeten Blickrichtungen werden ebenfalls in den Attributen vermerkt, so dass auf diese Weise die topologischen Relationen der verschiedenen Ansichten gespeichert werden und bei der Erkennung ausgewertet werden können. Abbildung 11.2 zeigt einige der in das Netzwerk eingebundenen Objekt- und Detailansichten eines PKW. Die topologischen Relationen werden hierbei durch die geometrische Anordnung der Ansichten verdeutlicht.

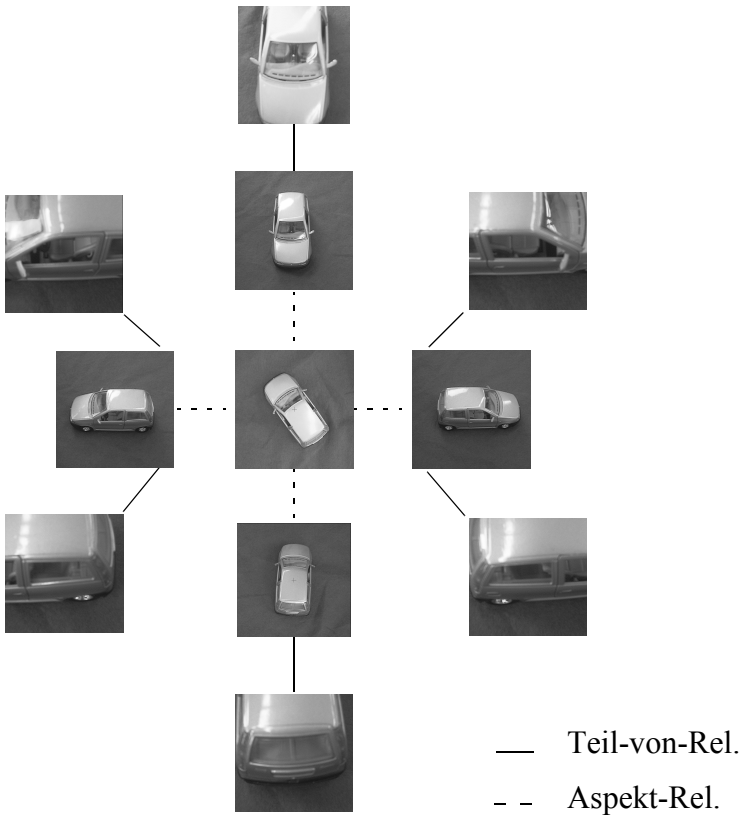


Abb. 11.2: Einige charakteristische Objekt- und Detailansichten eines Versuchsobjektes

Details der Modellierung der einzelnen Konzepte können den Beispielen aus Kapitel 6 entnommen werden. Die Notwendigkeit einer solchen automatischen Netzgenerierung wird auch dadurch deutlich, dass

das vollständige Netzwerk, in dem drei Autos modelliert sind, aus über 2000 Einträgen besteht. Auch wenn sich viele dieser Einträge sehr stark ähneln und ein neuer Autotyp zum Beispiel durch Kopieren eines bereits existierenden Modells und anschliessendem Anpassen der Wertebereiche in das Netz aufgenommen werden kann, so erleichtert die automatische Modellgenerierung doch erheblich die Arbeit mit dem Erkennungssystem.

11.5 Diskussion und Ausblick

Die erzeugten Netzwerke ähneln sehr stark den manuell erzeugten, enthalten jedoch bislang typischerweise mehr Konzepte und Ansichten. Dies macht offensichtlich, dass zum einen die manuelle Eingabe der Netzwerke zeitaufwendig ist und daher generell die Tendenz besteht, möglichst wenige Konzepte für die Modellierung zu verwenden. Zum anderen existieren jedoch möglicherweise beim Experten noch weitere Kriterien zur Auswahl der Ansichten, die in unser Verfahren noch nicht eingeflossen sind. Interessanterweise konnten jedoch auch einige Ansichten ermittelt werden, die bislang vom Anwender als charakteristisch und nicht verwechselbar betrachtet wurden, die aber doch zu Verwechselungen mit anderen Ansichten neigen.

Eine nachträgliche Bearbeitung der Modelle war bei unseren Tests nicht notwendig. Es wurden quasi-zweidimensionale Werkstücke (Cranfield-Satz) und komplexe dreidimensionale Automodelle (Maßstab 1:24) untersucht. Sollte eine weitere Minimierung des Objektmodells gewünscht sein, so kann diese manuell problemlos durchgeführt werden. Ein Löschen einzelner Netzteile ist dabei natürlich deutlich einfacher als das manuelle Erstellen. Prinzipiell hat jedoch die Tatsache, dass die automatisch erzeugten Netze zusätzliche Konzepte enthalten, keinen Einfluss auf die Erkennungsleistung sondern lediglich auf die Laufzeit der Erkennung. So wurden die automatisch generierten Modelle von Autos bereits in Vorführungen des Erkennungssystems eingesetzt.

Eine automatische Minimierung der für die Modellierung verwendeten Ansichten soll durch eine Überarbeitung des Netzwerkes nach Auswertung mehrerer Erkennungsdurchgänge erzielt werden, bei denen das Erkennungssystem Erfahrung über die betrachtete Domäne und das Instanziierungsverhalten der einzelnen Konzepte sammelt. Erst dann ist es auch möglich, dass das System selbständig Generalisierungen erzeugt. Da zur Zeit aus lediglich einer einzigen Präsentation eines Objektes sein Modell erzeugt wird, können somit auch nur die Bestandteil- und die Aspekt-Hierarchie automatisch generiert werden.

12 Anwendungen des Erkennungs-systems

In diesem Kapitel sollen an zwei unterschiedlichen Aufgabenstellungen die Einsatzmöglichkeiten des vorgestellten Systems aufgezeigt und die Tragfähigkeit der Beschreibungssprache verdeutlicht werden. Dazu wird zunächst die Modellierung flacher Szenen beschrieben. Es geht dabei um eine Demontageanwendung, in der der Bildverarbeitung die Aufgabe zu kommt, die zu demontierenden Objekte zu erkennen und zu vermessen. Im zweiten Anwendungsbeispiel geht es um die aktive Erkennung komplexer dreidimensionaler Objekte, bei der die Möglichkeiten des Erkennungssystems besonders gut zum Einsatz kommen.

12.1 Visuell gesteuerte Demontage von Altautos

Bei der ersten Anwendung des Erkennungssystems handelt es sich um eine Demontageaufgabe, in der exemplarisch die Demontage der Räder von Altautos untersucht wurde. Um akzeptable Laufzeiten bei gleichzeitig ausreichender Genauigkeit zu erzielen, ist es hierbei besonders wichtig, in einer geeigneten Strategie vorzugehen. Es hat sich gezeigt, dass in einer Grob-Fein-Strategie in Verbindung mit einer aktiven Kamerasteuerung die gewünschten Ergebnisse zu erzielen sind. Im folgenden wird nun beschrieben, wie diese Vorgehensweise im wissensbasierten System modelliert wird.

12.1.1 Modellierung der Räder

Da für diese Anwendung davon ausgegangen werden kann, dass die Position des zu demontierenden Autos in der Demontageanlage im wesentlichen bis auf geringfügige Abweichungen von einigen Zentimetern bekannt ist, kann bei der Modellierung der Räder auf die Auswertung verschiedener Aspekte verzichtet werden. Dadurch kann auf den Einsatz einer Kamera am Roboterarm verzichtet werden und es wird nur ein stationärer Stereokopf eingesetzt, der dem Aufbau der Demontageanlage entsprechend positioniert ist. Durch den Verzicht auf die Aspekt-Hierarchie ergibt sich ein Modell, das sich auf die Beschreibung verschiedener Felgentypen als Spezialisierungen und eine verwendete Grob-Fein-Strategie in Form einer initialen Lokalisierung auf der Szenenebene, einer Auswertung des Rades als Ganzes sowie der Untersuchung jeder einzelnen Schraube im Detail durch eine Teil-von-Hierarchie stützt. Dies wird durch Abbildung 12.1 verdeutlicht.

Bei der Auswertung des Netzes wird auf Szenenebene in einem ersten Schritt durch ein von Trapp entwickeltes Stereoverfahren [176] eine grobe Tiefenschätzung des Arbeitsraumes vorgenommen. Mögliche Hindernisse im Bewegungsbereich des Roboters werden lokalisiert und bei der Trajektorienplanung berücksichtigt. Parallel hierzu wird das zu demontierende Objekt relativ zum Kamerakopf lokalisiert. Eine genügend große Entfernung des Kamerakopfes zu möglichen Objektpositionen ermöglicht hierbei eine kollisionsfreie Szenenuntersuchung

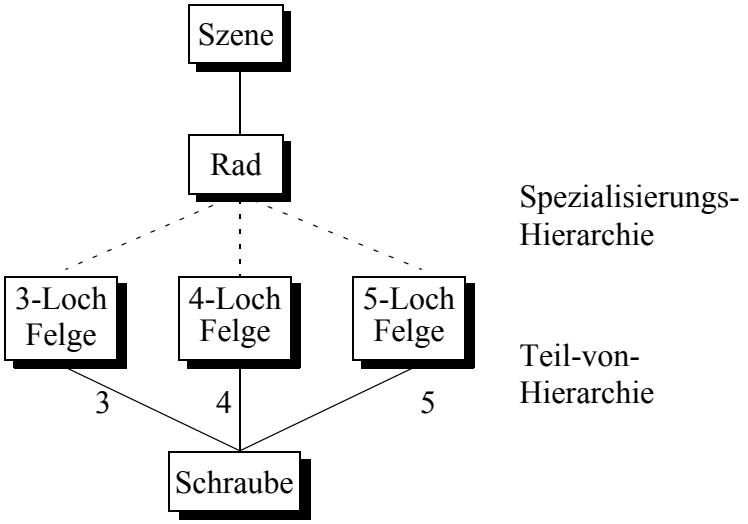


Abb. 12.1: Objektmodell zur Raderkennung

und erhöht gegenüber Nahaufnahmen die Übersicht in der Szene. Mit einem großem Abstand des Kamerakopfes zum Objekt sind bei Weitwinkelaufnahmen aber auch entsprechend hohe absolute Fehler in der Schätzung der Objektposition verbunden. Zur Einhaltung von Mindesttoleranzen können deswegen die Parameter der Kamerakopfeinheit (Zoom, Vergenz, etc.), basierend auf der Verknüpfung bisheriger Verarbeitungsergebnisse mit a priori Wissen des Objektmodells so verändert werden, dass die gesuchte Objektposition mit der erforderlichen Genauigkeit bestimmt wird. Dabei wird auf Objektebene des Modells eine formatfüllende Aufnahme des Rades generiert und ausgewertet[1]. Auf Detailebene werden dann Bildausschnitte in höchster Auflösung ausgewertet, um die zuvor erzielten Schraubenhypothesen zu verifizieren und um die Vermessung im Millimeterbereich durchzuführen.

12.1.2 Tiefenrekonstruktion durch Stereoverfahren

Die Positionsbestimmung der zu demontierenden Objekte in Weltkoordinaten wird parallel zur Erkennung (s. Abschnitt 2.4) durchgeführt. Für die Rekonstruktion der Tiefeninformation ist es notwendig,

1. Wenn hierzu die Freiheitsgrade des stationären Kamerakopfes nicht mehr ausreichen, kann zusätzlich eine am Roboter montierte Kamera aufgrund der ersten Messung gezielt in solche Aufnahmepositionen gefahren werden, die sich in nächster Nähe der jeweils zu vermessenden Objekte (Rad bzw. Radmuttern) befinden.

die Verschiebung korrespondierender Bildpunkte in den Bildern zu ermitteln. Im allgemeinen Fall sind diese in horizontaler und vertikaler Richtung ungleich Null. Zur Reduktion der Suche auf eine Dimension und somit des Rechenaufwandes werden die Bilder vor der Erkennung und Stereobildverarbeitung rektifiziert. Die Bilder, die mit nicht parallel ausgerichteten Vergenzachsen aufgenommen wurden, werden in Bilder eines Sehsystems mit parallelen Achsen transformiert [3].

Analog zur Erkennung verfolgt das Stereoverfahren einen konturbasierten Ansatz. Die Merkmalsextraktion wird mit einem Satz von orientierten Gabor-Filtern in verschiedenen Auflösungen durchgeführt (vgl. auch Kap. 4.2.6). In einem ersten Schritt werden die Filterantworten von linkem und rechtem Kamerabild miteinander korreliert. Die resultierenden Ähnlichkeitsmaße an den einzelnen Bildpositionen dienen dann als Initialisierung für einen Selbstorganisationsprozess, der in Anhang 3 ausführlicher beschrieben ist. Grundidee dieses Selbstorganisationsprozesses ist es, durch das Einbringen zusätzlicher Einschränkungen die nach der Korrelation noch vorhandenen Mehrdeutigkeiten aufzulösen. Hierzu gehören

- die Eindeutigkeitsbedingung - jedem Bildpunkt im dem einen Bild darf nur genau ein Bildpunkt im anderen Bild zugeordent werden,
- die Kontinuitätsbdingung - benachbarten Bildpunkten des einen Bildes sollen auch benachbarte Bildpunkte des anderen Bildes zugeordnet werden.

Zusätzlich erlaubt das Verfahren auch die Detektion von Verdeckungsbereichen, denen dann keine Disparität zugeordnet wird. Nach Erreichen des stationären Zustandes des Relaxationsprozesses wird der Imaginärteil des Ähnlichkeitsmaßes genutzt, um die Disparitätskarte genauer als das Abtastraster zu berechnen. An Positionen mit Variablen ungleich Null, die mit hoher Wahrscheinlichkeit mit Maxima des Realteils korrespondieren, wird der Verlauf des Imaginärteils interpoliert. Der Nulldurchgang gibt nun die neue subpixelgenaue Disparität an. Der Einsatz des subpixelgenauen Verfahrens kann den mittleren quadratischen Fehler durch die Quantisierung um bis zu 60% reduzieren [175].

12.1.3 Fusion von Bild- und Tiefendaten

Um einen Greifer kollisionsfrei an das zu demontierende Objekt heranzufahren, muss dessen Position und Lage in Bezug auf das Roboterkoordinatensystem bestimmt werden. Dazu werden die zweidimensionalen Informationen der Erkennung fusioniert mit den dreidimensionalen Tiefeninformationen. Die extrahierten dreidimensionalen Koordinaten bilden eine Punktmenge, die als eine verrauschte Abtastung der Objektkonturen oder -oberflächen interpretiert werden kann. Durch eine „geometrische Filterung“ werden fehlerhafte Punkte, die aufgrund der Kontinuitätsbedingung nicht zur Kontur oder Fläche gehören können, herausgefiltert. Zur Bestimmung der Oberflächennormale der Objekte wird eine Ebene mit der Methode der kleinsten Fehlerquadrate durch die extrahierte Punktmenge gelegt.

Da sich alle bisher ermittelten Positionen und Flächennormalen auf ein Koordinatensystem beziehen, dessen Ursprung mittig auf der Verbindungslinie der Projektionszentren der Kameras liegt, müssen abschließend Punkte und Vektoren vor einer Roboteraktion mittels einer affinen Transformation in Roboterkoordinaten umgerechnet werden:

$$\boldsymbol{p}_{Roboter} = \boldsymbol{A} \cdot \boldsymbol{p}_{Kamera} \tag{12.1}$$

Dies entspricht der in Kapitel 9.3 beschriebenen Umrechnung eines Koordinatensystems der Handkamera in das Roboterkoordinatensystem.

Die Parameter der Transformation werden mit einem Verfahren ähnlich wie bei der Kamerakalibrierung bestimmt. Eine Messplatte mit neun Quadraten wird an der Roboterhand befestigt. Da die geometrischen Beziehungen zwischen den Eckpunkten der Quadrate und dem Tool-Center-Point fest definiert sind, sind diese Markeneckpunkte in Roboterkoordinaten bekannt. Mit dem Stereokamerakopf werden nun die Eckpunkte im Grauwertbild subpixelgenau detektiert, die jeweiligen dreidimensionalen Weltkoordinaten bestimmt und mit Hilfe des Gauß-Newton-Minimierungsverfahrens werden die Transformationsparameter abschließend optimiert.

12.1.4 Modellbeispiele

Exemplarisch sollen in der folgenden Abbildung einige Konzeptbeschreibungen vorgestellt werden. Da die Hierarchiebeziehungen zwischen den Konzepten bereits aus der Abbildung 12.1 eindeutig hervorgehen, werden diese hier nicht mehr aufgeführt.

Man kann der Modellierung des Konzeptes *Rad* sehr gut entnehmen, wie die Verarbeitung des Aufgangsbildes <*STEREOBILD*> der Szene in mehreren Schritten in der besprochenen Grob-Fein-Strategie erfolgt. Dazu erfolgt jeweils eine Gaborfilterung (Operation *GABOR_FILTER*) in verschiedenen Auflösungsebenen (Parameter *sampling* = *n*). Nach einem entsprechenden Erkennungsschritt (Operation *DETECTION*) erfolgt im Bereich der detektierten Bildkoordinaten bei Bedarf ein erneutes Ausrichten der Kamera mit anschließender Bildaufnahme (Operation *GRAB_STH_IMAGES*) oder ein Ausschneiden eines geeigneten Bildbereiches um den Fixationspunkt herum (Operation *CUT_STH_IMAGES* mit Parameter *size* = *n*). Das so gewonnene Bildmaterial wird dann in verbesserter Auflösung erneut gefiltert.

In den drei Felgen-Spezialisierungen wird dann nur noch eine Region aus der fixierten Felge herausgeschnitten, in der sich die Schrauben befinden. Die Erkennung der Schrauben erfolgt dann im Konzept *Schraube*. Im Attribut der Felgen-Spezialisierung werden die detektierten Schrauben-Positionen (typischerweise ca. 15 Positionen) auf ihre Geometrie überprüft und entweder die 3, 4 oder 5 geeigneten Schrauben ausgewählt.

Durch diese Vorgehensweise kann der Aufwand für die Suche nach der korrekten Schraubenkonstellation beträchtlich eingeschränkt werden. Dazu eine kurze formale Betrachtung:

n sei die Anzahl der detektierten Schraubenhypothesen; *k* sei die Anzahl der gesuchten Schrauben (typischerweise gilt: $k = 3$, 4 oder 5). Dann ergeben sich daraus $M = n!/(n-k)!$ Möglichkeiten, die auf die geeignete Geometrie getestet werden müssen. dies lässt sich noch weiter reduzieren auf

Konzeptname: Rad

Pre-Attribut:	GABOR
Typ:	BOOLEAN
Wertebereich:	{TRUE}
Anzahl:	1,1
Operation:	GABOR_FILTER
Operand:	<STEREOBILD>
Operationsgebiet:	
Parameter:	sampling=4
formales Ergebnis:	<GAB_BILD>

Pre-Attribut:	DETECT
Typ:	BOOLEAN
Wertebereich:	{TRUE}
Anzahl:	1,1
Operation:	DETECTION
Operand:	<GAB_BILD>
Operationsgebiet:	
Parameter:	r=25,file=Prototypes/list128
formales Ergebnis:	<B_KOORDS>

Pre-Attribut:	CUT
Typ:	BOOLEAN
Wertebereich:	{TRUE}
Anzahl:	1,1
Operation:	CUT_STH_IMAGES
Operand:	<STEREOBILD><B_KOORDS>
Operationsgebiet:	
Parameter:	size=64
formales Ergebnis:	<CUT_STEREOBILD>

Abb. 12.2: Modellierung des Konzeptes *RAD* (wird fortgeführt)

Konzeptname: Rad (fortgeführt)

Pre-Attribut: GABOR_CUT
Typ: BOOLEAN
Wertebereich: {TRUE}
Anzahl: 1,1
Operation: GABOR_FILTER
Operand: <CUT_STEREOBILD>
Operationsgebiet:
Parameter: sampling=1
formales Ergebnis:<GAB_CUT>

Pre-Attribut: DISP_CUT
Typ: BOOLEAN
Wertebereich: {TRUE}
Anzahl: 1,1
Operation: DISPARITY_MAP
Operand: <GAB_CUT><B_KOORDS>
Operationsgebiet:
Parameter:
formales Ergebnis:<DISP_CUT>

Pre-Attribut: FixedRad
Typ: BOOLEAN
Wertebereich: {TRUE}
Anzahl: 1,1
Operation: GRAB_STH_IMAGES
Operand: <B_KOORDS><DISP_CUT>
Operationsgebiet:
Parameter:
formales Ergebnis:<BildFixedRad>

Abb. 12.3: Modellierung des Konzeptes *RAD* (wird fortgeführt)

Konzeptname: Rad (fortgeführt)

Pre-Attribut:	GabFixedRad
Typ:	BOOLEAN
Wertebereich:	{TRUE}
Anzahl:	1,1
Operation:	GABOR_FILTER
Operand:	<BildFixedRad>
Operationsgebiet:	
Parameter:	sampling=8
formales Ergebnis:	<GabFixedRad>
Pre-Attribut:	DetectFixedRad
Typ:	BOOLEAN
Wertebereich:	{TRUE}
Anzahl:	1,1
Operation:	DETECTION
Operand:	<GabFixedRad>
Operationsgebiet:	
Parameter:	r=22,file=Prototypes/list64
formales Ergebnis:	<KoordFixedRad>
Bewertung:	[3-Loch Felge]
Bewertung:	[4-Loch Felge]
Bewertung:	[5-Loch Felge]

Ziel:

ENDE Rad

Abb. 12.4: Modellierung des Konzeptes *RAD*

Konzeptname:3-Loch Felge

Pre-Attribut:	CUT_BILD_KRANZ
Typ:	BOOLEAN
Wertebereich:	{TRUE}
Anzahl:	1,1
Operation:	CUT_STH_IMAGES
Operand:	<BildFixedRad>
Operationsgebiet:	
Parameter:	size=256
formales Ergebnis:	<CutBildKranz>
Attribut:	TESTGEO5
Typ:	BOOLEAN
Wertebereich:	{TRUE}
Anzahl:	1,1
Operation:	PIXGEOMETRY
Operand:	<KoordSchraube><KoordSchraube><KoordSchraube>
Operationsgebiet:	
Parameter:	3
formales Ergebnis:	<Geo3>
Bewertung:	<Geo3>
Ziel:	

ENDE 3-Loch Felge

Abb. 12.5: Modellierung des Konzeptes *3-Loch Felge*

$$M = \binom{n}{k} = \frac{n!}{(n-k)!k!}, \tag{12.2}$$

wenn die Permutationen innerhalb der k Schrauben belanglos sind.

In der modellgestützten Grob-Fein-Strategie wird vor der eigentlichen Schraubenerkennung noch eine grobe Suche nach potentiellen Rädern im Bild durchgeführt. m sei nun die Anzahl der detektierten Radhypothesen. Durch Umweltwissen im Modell gilt: $m \approx 1$, da die grobe Entfernung und Position des Rades in der Demontagezelle bereits bekannt ist. Da sich aber aus Radposition und detektierter erster Schraubenposition die restlichen Schraubenpositionen vorhersagen lassen, ergibt sich nun als Suchaufwand, dass für jede Felgenhypothese aus den n potentiellen Schrauben jeweils eine ausgewählt wird und getestet wird, welche der restlichen Schrauben mit den restlichen k-1 gesuchten Positionen übereinstimmen. Im O-Kalkül ergibt sich somit ein Aufwand von:

$$O(m \cdot n \cdot k) = O(n) \tag{12.3}$$
da k konstant und $m \ll n$, m quasi konstant

Es wird also ein linearer Suchaufwand erzielt, der lediglich von der Anzahl der detektierten Schraubenpositionen abhängt. Da hierbei üblicherweise die korrekten Schraubenpositionen eine bessere Bewertung erfahren, als falsche Hypothesen, die z. B. durch Rost oder Schmutzflecken auf der Felge entstehen, verringert sich der Suchaufwand nochmals. Die Schätzfunktion des zugrunde liegenden A^* Algorithmus war ja gerade so gewählt, dass gut bewertete Instanzen zu bevorzugten Suchpfaden führen (Kap. 7.3.4).

12.1.5 Ergebnisse

Der beschriebene Demontageprozess wurde in einer ersten Implementierung zu Demonstrationszwecken in einem Dauertestlauf mehrfach geprüft. Bei einem nahezu senkrecht stehenden Rad in verschiedenen Positionen konnte eine Schraube in ca. 97% der Durchläufe entnommen werden. In den fehlerhaften Versuchen wurde die Schraube aufgrund eines falsch geschätzten Abstandes nicht oder nicht tief genug

gegriffen. Dagegen war die Genauigkeit in der *x*-*y* Ebene immer ausreichend, so dass eine Kollision des Greifers mit der Felge oder Schraube nicht auftrat (vgl. auch [32]).

Einige der getesteten Autoräder werden in Abbildung 12.6 gezeigt.

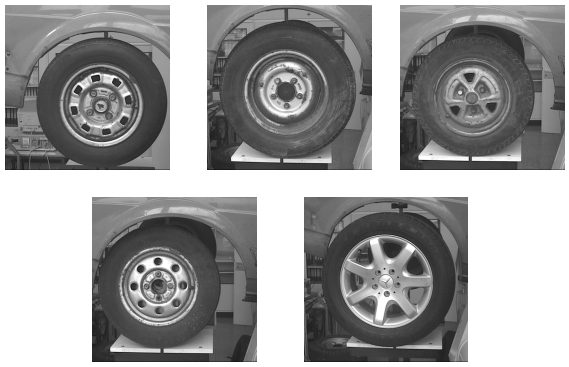


Abb. 12.6: Beispiele der getesteten Autoräder

Die verwendete Grob-Fein-Strategie wird in Abbildung 12.7 nochmals verdeutlicht. Das Stereo-Verfahren zeigt eine ausreichend gute Genauigkeit für den Demontageprozess. Selbst bei einem Abstand von 1,7 Metern zwischen der Kamera und den Rädern, der gewählt wurde, um den Arbeitsraum des Roboters frei zu halten, wird eine Genauigkeit in *x*-*y*-Richtung in einer Ebene parallel zur Stereobasis und zur Vergenzachse von besser als 1 mm erreicht. Die Tiefenschätzung (*z*-Richtung) erfolgt bei diesem Abstand mit einem Fehler von weniger als 2 mm. Die erzielbaren Genauigkeiten sind im wesentlichen linear abhängig vom Abstand zwischen Kamera und Objekt. Die folgende Abbildung 12.8 verdeutlicht diese Abhängigkeit. Sie zeigt den Fehler der Tiefenschätzung in Abhängigkeit von der Objektentfernung. Die gepunktete Linie beschreibt dabei den theoretisch möglichen Fehler, die Punkt-Strich-Linie zeigt den gemessenen Maximalfehler, während die durchgezogenen Linie den mittleren Tiefenfehler zeigt.

Zusätzlich wurden Untersuchungen zur Genauigkeit der Winkelbestimmungen durchgeführt (Abb. 12.9). Sowohl der Sturz als auch die Spur des Rades konnten in einem Bereich von ±20° mit einer Genauigkeit von 1° bestimmt werden. Diese Genauigkeit entspricht im wesentlichen der mechanischen Auflösung der Testkonstruktion.

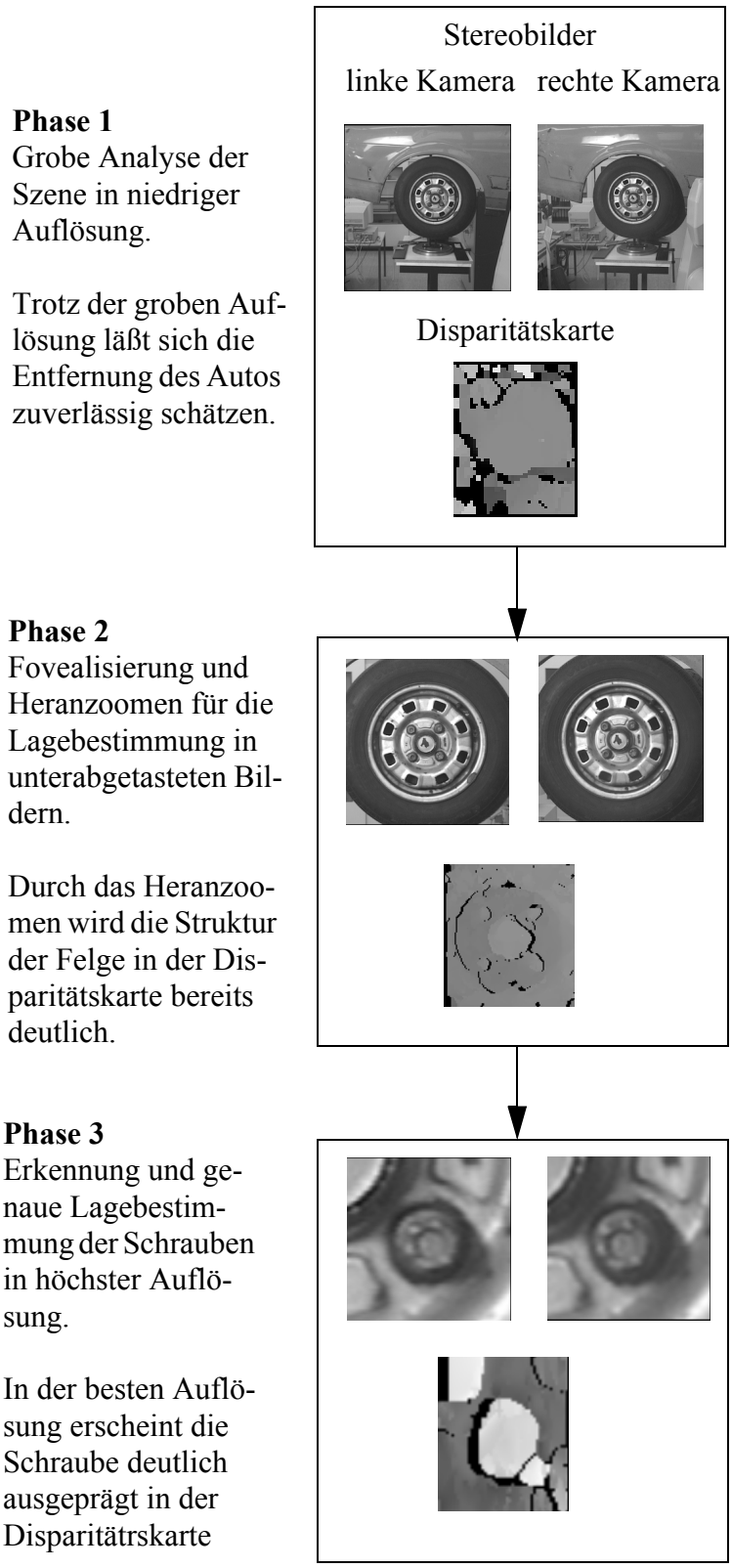


Abb. 12.7: Die verschiedenen Phasen der Grob-Fein-Strategie

Bei den Versuchsdurchführungen konnte gezeigt werden, dass die beschriebene Erkennungs- und Vermessungsstrategie hinreichend gute

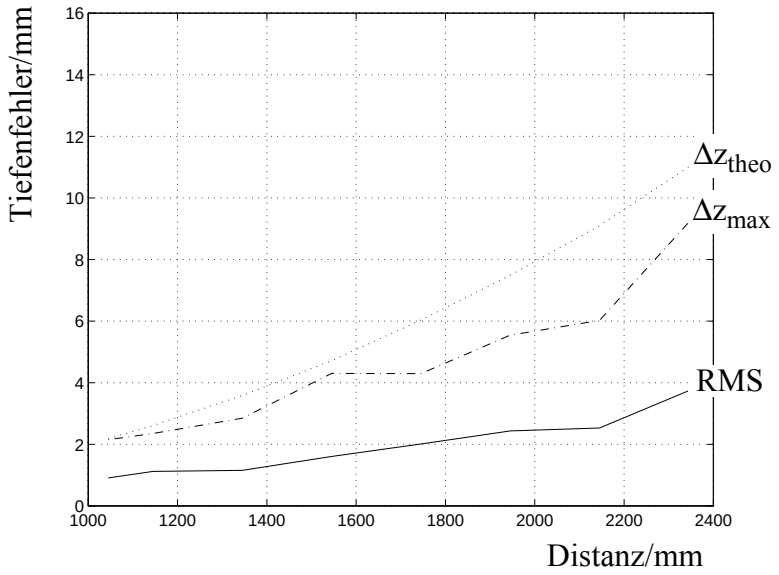


Abb. 12.8: Messfehler bei der Tiefenrekonstruktion

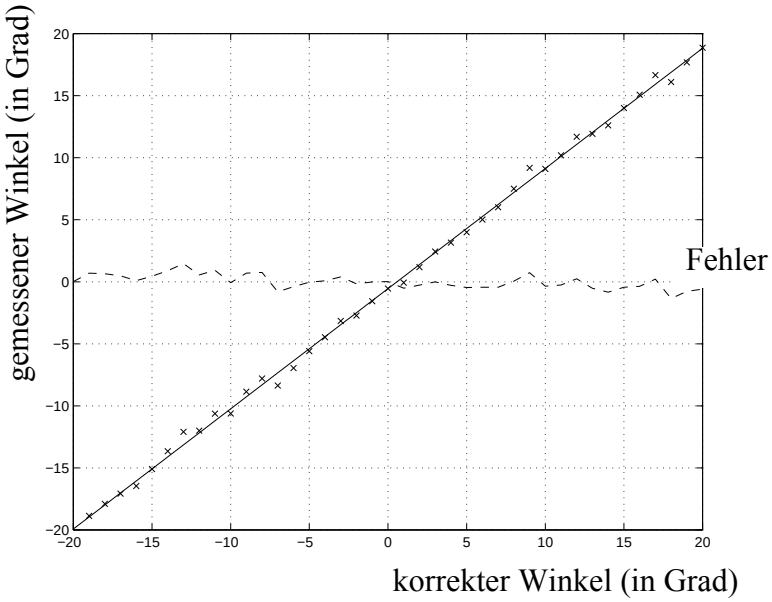


Abb. 12.9: Bestimmung von Sturz und Spur

Ergebnisse für den automatischen Demontageprozess liefert. Es wurde deutlich, dass speziell die gewählte Grob-Fein-Strategie sich sehr gut in den semantischen Netzwerken abbilden lässt. Auch die Einbindung des aktiven Stereokamerakopfes war problemlos über die prozeduralen Schnittstellen in den Attributbeschreibungen möglich.

Der Aspekt des aktiven Sehens und seine Einbindung in das Erkennungssystem wird am zweiten Anwendungsbeispiel der Erkennung komplexer dreidimensionaler Objekte noch wesentlich deutlicher. Dabei wurden beispielhaft verschiedene Autos modelliert.

12.2 Modellierung dreidimensionaler Objekte

Bei der aktiven Erkennung komplexer dreidimensionaler Objekte ist der Einfluss expliziten Modellwissens über das Objekt um so wichtiger, da im 3D Fall im allgemeinen objektspezifische Details nur durch starke Änderungen der Blickrichtung und der Aufnahmeposition erkannt werden können. Rein datengetrieben können nicht alle notwendigen Fovealisierungspunkte und speziell nicht die dann dazugehörigen unterschiedlichen Blickrichtungen generiert werden, da in einem initialen Bild nicht alle Details enthalten sein können.

12.2.1 Modellierung komplexer 3D-Objekte

Für die Erkennung von 3D-Objekten wurden daher in der Wissensbasis verschiedene typische 2D-Ansichten (Aspekte) des Objektes aufgenommen (Abb. 12.10).

In der Bearbeitungsphase erfolgt dann durch den visuell geführten Roboter eine aktive Exploration des Objektes, bei der die modellierten Ansichten und Teilansichten ausgewertet werden. Dabei wird ausgenutzt, dass die holistischen Erkennungsverfahren bereits auf Objektebene der Modellierungshierarchie eine hinreichend genaue Lageschätzung des Objektes liefern. Entsprechend dieser Lageschätzung werden die weiteren Aufnahmepositionen der Aspekt-Hierarchie berechnet und für eine detaillierte Objekterkennung oder -verifikation von der an der Roboterhand montierten Kamera eingenommen. Die an diesen Positionen aufgenommenen Bilder werden dann wiederum den neuronalen Verarbeitungsschritten zugeführt.

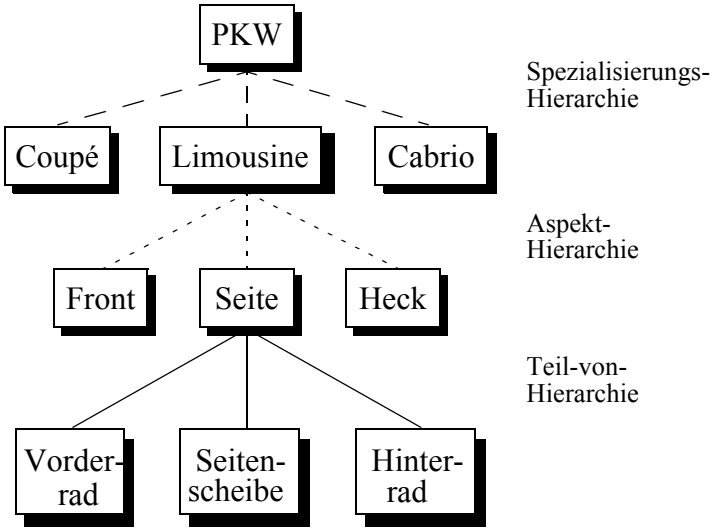


Abb. 12.10: 3D-Objektmodell in Form eines semantischen Netzwerkes

Erst durch diese aktive Objekterkennung können einander ähnliche, komplexe dreidimensionale Objekte erkannt und unterschieden werden. Am Beispiel der Erkennung eines PKWs soll die Auswertungsstrategie näher erläutert werden. Aus der initialen Ansicht ergibt sich mit Hilfe der neuronalen Verfahren eine Hypothese für Art und Lage des Objektes. Im semantischen Netzwerk wird daher im Subnetz "PKW" eine detailliertere Untersuchung angestoßen. Entsprechend der Modellbeschreibung und der initial geschätzten Lage werden nun Bilder von Front, Seite und Heck des PKW aufgenommen und wiederum mit den neuronalen Verfahren analysiert. Genauso wird auch mit den Details der Seitenansicht verfahren, wenn diese nicht zuvor mit ausreichender Sicherheit bereits ganzheitlich erkannt werden konnte (Abb. 12.11).

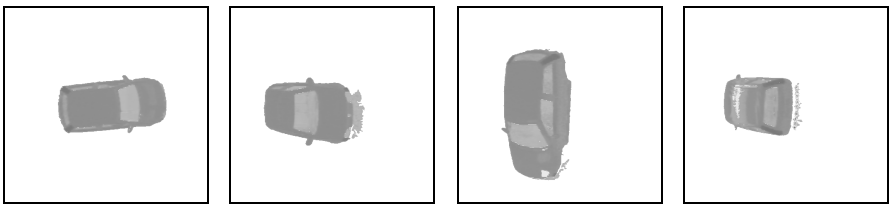


Abb. 12.11: Segmentierte Initialansicht, Front-, Seiten- und Heckansicht eines Fiat Punto

Für die Bestimmung der Anfahrpunkte ist es dabei ausreichend, wenn die Modellbeschreibung lediglich prinzipielle Positionen festlegt, die für alle PKW-Typen verwendet werden. Mit Hilfe der Bildverarbeitungsroutinen - in diesem Fall der Farbsegmentierung - wird dann eine verbesserte Kameraposition berechnet und es wird neu fovealisiert. Hierdurch reduziert sich nicht nur der Aufwand bei der Erstellung der Objektmodelle, sondern es können auch unvermeidliche Abweichungen in der Lageschätzung toleriert werden.

12.2.2 Suchaufwand

Auch für diese Anwendung lässt sich analog zur Demontageanwendung eine Aufwandsabschätzung durchführen: Es seien initial m Hypothesen für Objektpositionen gebildet, es seien n Ansichten des Objektes modelliert. Dann ergibt sich hieraus ein Suchaufwand $O(mn)$, wenn für jede der m Hypothesen alle n Ansichten getestet werden. Da für falsche Hypothesen jedoch typischerweise keine der n Ansichten instanziiert werden kann, folgt wieder durch die Wahl der Schätzfunktion im A^* Algorithmus, dass bereits frühzeitig, d.h. nach der ersten nicht instanzierbaren Ansicht auf andere Hypothesen „umgeschaltet“ wird. Somit ergibt sich auch hier wieder ein linearer Suchaufwand, der im wesentlichen von der Anzahl der modellierten Ansichten abhängt. Im O-Kalkül lässt er sich charakterisieren durch $O(m+n)$. Dies drückt aus, dass im ungünstigen Fall jede der m Hypothesen - und die dadurch entstehenden Suchpfade - angetestet werden, bevor die richtige Hypothese gefunden wurde und deren n Ansichten ausgewertet werden.

12.2.3 Ergebnisse

Anhand einer Versuchsreihe konnte die Leistungsfähigkeit des Systems nachgewiesen werden. Hier sei nun zunächst die Versuchsumgebung kurz skizziert. Es wurden zwei Modellautos (ein Fiat Punto und ein Ferrari F40) im Maßstab 1:24 verwendet. Die Modellautos sind lackiert und besitzen eine stark glänzende Oberfläche. Ihre Größe erlaubt es, trotz der eingeschränkten Reichweite eines relativ kleinen Roboters Bilder der Autos in vielen Positionierungen von verschiedenen Seiten aus aufzunehmen. Gleichzeitig zeigen sie aber noch eine Vielzahl von

Details, deren Auswertung von Interesse ist. Diese Autos werden dem System auf einem Tisch mit einer wenig strukturierten Unterlage präsentiert. Da der ganze Aufbau direkt an einer Fensterfront steht, stellt die Beleuchtungsvariation aufgrund sich häufig ändernden Tageslichtes (direkte Sonneneinstrahlung, diffuses Licht durch Bewölkung, etc.) ein Problem dar. Hieraus resultierende Segmentierungsungenauigkeiten werden jedoch recht gut durch die wolkenartige Merkmalsrepräsentation aufgefangen; trotzdem sind hier weiterführende Arbeiten notwendig. Dies spielt besonders dann eine große Rolle, wenn im aktiven Erkennungsprozess die Kamera in Richtung der Fensterfront steht und dabei durch die Gegenlichtsituation sehr schwierige Lichtverhältnisse entstehen. Speziell die Heckansichten erwiesen sich in diesen Situtationen als problematisch[1].

Die folgende Tabelle zeigt die Erkennungsraten für die Gesamtobjekte, sowie für einige ausgewählte Details. Es wird deutlich, dass es zwar zu Fehlklassifikationen und Rückweisungen bei einzelnen Teilansichten kommt, dass aber unter Einbeziehung der gesamten Modellinformation die Autos sehr häufig korrekt erkannt werden konnten. Insbesondere konnten hierdurch Fehlklassifikationen vermieden werden. Die zwei Modellautos wurden dem Erkennungssystem insgesamt 100 Mal in verschiedenen Positionen und unter unterschiedlichen Beleuchtungsbedingungen bei zum Teil direkter Sonneneinstrahlung präsentiert.

Tabelle 3: Erkennungsergebnisse

Fiat Punto	Gesamt-objekt	Front	linke Seite	rechte Seite	Heck
korrekte Klassifikation	88%	76%	88%	86%	68%
Fehlklassifikation	0%	10%	0%	0%	14%
Rückweisung	4%	4%	4%	10%	8%
ungeeignete Aufnahmeposition	-	10%	8%	4%	10%
Generalisierung „Auto“	8%				

1. Hier zeigt sich für die Anwendung als aktives Sehsystem die Notwendigkeit neuer Kameratechnologien, die z.B. eine deutlich verbesserte Dynamik aufweisen, wie dies etwa bei der CMOS-basierten Technologie der Fall ist.

Ferrari F40	Gesamt-objekt	Front	linke Seite	rechte Seite	Heck
korrekte Klassifikation	80%	86%	82%	76%	72%
Fehlklassifikation	0%	4%	10%	20%	20%
Rückweisung	0%	0%	0%	0%	0%
ungeeignete Aufnahmeposition	-	10%	8%	4%	8%
Generalisierung „Auto“	20%				

Zusätzlich zum Modellwissen über die Autotypen *Punto* und *F40* wurde im semantischen Netz auch ein generalisierendes Konzept *Auto* verwendet, welches immer dann instanziiert wurde, wenn verschiedene Details erkannt wurden, aber keine hinreichend gute Erkennung eines speziellen Autotyps möglich war. Somit kann bei den Erkennungsergebnissen auf Objektebene unterschieden werden zwischen korrekter Erkennung, Fehlklassifikation, Rückweisung oder Zuordnung zum generalisierenden Konzept *Auto*. Darüber hinaus wird in der Tabelle noch aufgeführt, wie häufig die Erkennung scheiterte, weil aufgrund der Roboterkinematik eine geeignete Position zur Bildaufnahme nicht eingenommen werden konnte.

13 Schlussbetrachtung und Ausblick

Können Maschinen sehen? Erwartet uns eine neue Generation sehender und autonom arbeitender, vielleicht gar intelligenter Roboter? Vieles deutet daraufhin, dass sich in wenigen Jahren völlig neue Möglichkeiten für die Automatisierung von Produktionsabläufen ergeben werden. Immer leistungsfähigere Hardware ermöglicht es, Fragestellungen zu untersuchen und Systeme zu entwickeln, die tatsächlich zu sehenden Robotern führen werden. Die in dieser Arbeit vorgestellte Anwendung der visuell gesteuerten Demontage von Altautos stellt ein eindrucksvolles Beispiel für die sich abzeichnenden Entwicklungen dar.

Gegenstand der Arbeit war es, ein für die visuell gesteuerte Automation geeignetes Bilderkennungssystem vorzustellen. Wesentlichstes Element des diskutierten Systems ist seine Architektur als hybrides, subsymbolisch und symbolisch arbeitendes aktives Objekterkennungssystem. Die allgemeine Verwendbarkeit der Methoden auf beiden Ebenen des Systems ermöglicht seinen Einsatz in ver-

schiedenen Applikationen. Sowohl die Montage als auch die noch deutlich schwierigere Demontage unterschiedlicher Objekte werden hiermit ermöglicht. Von besonderer Bedeutung ist dabei die Einbeziehung robuster, biologisch motivierter Konturdetektoren und eines hierauf aufbauenden holistisch arbeitenden Klassifikationsmoduls in eine dekompositorische, symbolische Objektbeschreibung. Hierdurch entsteht die Möglichkeit, domänenunabhängige Verfahren auf der subsymbolischen Ebene mit speziellem Domänenwissen auf der symbolischen Ebene zu koppeln. Erst durch Einbeziehung dieses Domänenwissens kann die dargestellte Leistungsfähigkeit erreicht werden. In ein solches hybrides System kann gleichzeitig auch die Idee des aktiven Sehens integriert werden. Somit wurden drei unterschiedliche Paradigmen homogen in ein Gesamtsystem integriert:

- die GANZHEITLICHE ERKENNUNG,
- die DEKOMPOSITORISCHE ERKENNUNG,
- das AKTIVE SEHEN.

Es konnte gezeigt werden, dass sich diese verschiedenen Paradigmen gegenseitig sinnvoll und optimal ergänzen. Verschiedene Probleme der einzelnen Ansätze, wie z.B. die Anfälligkeit bei Verdeckungen oder die exponentiell wachsende Größe des Suchraumes, wie sie in Kapitel 3 und 4 diskutiert wurden, konnten überwunden und so die Leistungsfähigkeit des Gesamtsystems deutlich verbessert werden.

Mit der hier entwickelten Symbiose von subsymbolischer und symbolischer Verarbeitung in Verbindung mit aktiven Mechanismen können neue, komplexe Anwendungsgebiete für die Bilderkennung erschlossen werden. Damit vermag dieser Ansatz dem *sehenden Roboter* einen neuen Baustein hinzuzufügen. Dem Ziel, autonom arbeitende, visuell gesteuerte Roboter in der industriellen Produktion einzusetzen, sind wir dadurch einen beträchtlichen Schritt näher gekommen.

Für den hier vorgestellten sehenden Roboter eröffnen sich vielfältige Aufgabenfelder, die weit über die industrielle Produktion hinausgehen. So ergeben sich auch im Dienstleistungssektor bislang ungeahnte Einsatzmöglichkeiten. Denkbar ist z.B. ein Tankroboter, der nebenbei noch das Reifenprofil und den Ölstand kontrolliert. Auch ein autonom arbeitender Rasenmäher, der nicht nur den Rasen mäht, sondern auch allein den Weg vom Vorgarten hinter das Haus findet, ist im Bereich

des möglichen. Wenn der in dieser Arbeit aufgezeigte Weg und seine Methoden von der Industrie aufgegriffen und in Produkte umgesetzt werden, wird sich unser Lebensumfeld in den kommenden Jahren drastisch ändern. Angebot und Nachfrage werden darüber entscheiden, in welchen Bereichen sehende Roboter zuerst Einzug halten werden. Wie bei allen Neuerungen sollten wir dabei nicht vergessen, dass sie in erster Linie den Menschen bei der Bewältigung ihrer Aufgaben unterstützen sollten. Das technisch Interessante sollte nicht allein seiner Machbarkeit wegen umgesetzt werden. Die soziale und ethische Komponente eines technischen Systems und seiner Auswirkungen muss daher bei der konkreten Produktgestaltung mitberücksichtigt werden. Der in Japan bereits angedachte Klinikroboter [169], der die Patienten mit Mahlzeiten und Getränken versorgt, möge uns darum hoffentlich erspart bleiben.

Die zukünftigen Aufgaben werden nun die Konsolidierung des Erreichten, der Erschließung weiterer Anwendungsgebiete, aber auch die Weiterentwicklung einzelner Systemkomponenten sein. Von besonderer Bedeutung für eine weite Verbreitung des Ansatzes ist hierbei das Lernmodul, das bislang als ein erster Prototyp zu betrachten ist. Es ist in der Lage, aus einer einzelnen Objektpräsentation auf beiden Systemebenen eine Beschreibung des Objektes zu erzeugen und in die Wissensbasis des Systems einzufügen. Ziel sollte jedoch ein „lebenslanges“ Lernen des Systems sein, so dass sich dieses an neue Umgebungsbedingungen selbständig adaptieren kann. Dies betrifft sowohl die subsymbolische Ebene, auf der die holistische Erkennung durch Vergleich mit zuvor gelernten Prototypen stattfindet, als auch die symbolische Ebene, die beschreibt, welche Ansichten eines Objektes für die Erkennung relevant sind und in welchen Relationen diese zueinander stehen.

Die Kombination des Erkennungssystems mit einer mobilen Plattform würde darüber hinaus noch weitere Freiheitsgrade schaffen und außerdem dazu beitragen, dass zusätzliche Anwendungen bearbeitet werden können. Erste Untersuchungen zur Kooperation mehrerer autonom arbeitender Erkennungssysteme werden zur Zeit bereits durchgeführt und dienen ebenfalls der weiteren Leistungssteigerung sowie der Erhöhung der Flexibilität.

Mit diesem kurzen Ausblick auf zukünftige Arbeiten sollte deutlich gemacht werden, dass trotz der sehr vielversprechenden Resultate, die

erzielt werden konnten, noch Einiges zu tun ist, wenn in Zukunft *sehende Roboter* flexibel einsetzbar sein sollen.

Mit dem in dieser Arbeit vorgestellten hybriden Objekterkennungssystem und dessen Anwendungen konnte aufgezeigt werden, in welche Richtung Wege zu diesem Ziel beschritten werden können - Wege, die nicht nur die Ingenieurwissenschaft künftig vor neue, interessante und herausfordernde Aufgaben stellen werden.

14 Anhang

14.1 Anhang 1: Gaborfilter

Gaborfilter werden im hier vorgestellten System zur auflösungs- und orientierungsselektiven Kantenextraktion eingesetzt. Diese erfolgt in M=3 Auflösungsebenen und L=12 verschiedenen Orientierungen.

Die komplexe Impulsantwort des verwendeten Gaborfilters ergibt sich dabei für eine zweidimensionale Bildkoordinate $\boldsymbol{x}$ mit $m \in [0,\ldots,M-1]$ und $l \in [0,\ldots,L-1]$ zu:

$$g_{ml}(\boldsymbol{x}) = \frac{1}{2\pi a_m b_m} \mathrm{e}^{-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{A}_{ml} \boldsymbol{x}} \mathrm{e}^{j\boldsymbol{k}_{0ml}^T \boldsymbol{x}} \tag{14.1}$$

mit

$$\boldsymbol{A}_{ml} = \begin{bmatrix} \cos\phi_l & -\sin\phi_l \\ \sin\phi_l & \cos\phi_l \end{bmatrix} \begin{bmatrix} a_m^{-2} & 0 \\ 0 & b_m^{-2} \end{bmatrix} \begin{bmatrix} \cos\phi_l & \sin\phi_l \\ -\sin\phi_l & \cos\phi_l \end{bmatrix} \tag{14.2}$$

und

$$\boldsymbol{k}_{0ml} = k_{0m}\begin{bmatrix}\cos\phi_l\\ \sin\phi_l\end{bmatrix} \tag{14.3}$$

Im Frequenzbereich ergibt sich daher:

$$G_{ml}(k) = \mathrm{e}^{-\frac{1}{2}(\boldsymbol{k}-\boldsymbol{k}_{0ml})^T \boldsymbol{A}_{ml}^{-1\mathrm{T}}(\boldsymbol{k}-\boldsymbol{k}_{0ml})} \tag{14.4}$$

Bei einer Betrachtung von $M = 3$ Auflösungsebenen und $L = 12$ verschiedener Orientierungen werden k_{0m} und ϕ_l wie folgt gewählt:

$$k_{0m} = \frac{\pi}{2^{m+1}}; \qquad m \in [0, \ldots, M-1] \tag{14.5}$$

$$\phi_l = l\Delta\phi; \qquad l \in [0, \ldots, L-1] \tag{14.6}$$

mit $\Delta\phi = 15°$.

In unserem Ansatz wurde die relative Bandbreite der Filter - wie von Trapp in [176] vorgeschlagen - auf 0.6 Oktaven festgelegt. In diesem Fall besitzen benachbarte Filter in ihrem Schnittpunkt einen Wert von 0.5 (Abb. 14.1).

Aus dem gleichen Grund wird auch das Verhältnis zwischen a_m und b_m gewählt:

$$\lambda = \frac{a_m}{b_m} = 0.78; \qquad \forall m \in [0 \ldots M-1] \tag{14.7}$$

Bei dieser Wahl besitzen benachbarte Filter unterschiedlicher Orientierung in ihrem Schnittpunkt den Wert 0.5 auf einem Kreis gleicher Orts-Frequenz um den Mittelpunkt (Abb. 14.2).

Aufgrund der geringen Bandbreite können die Filterantworten mit einen Faktor

$$s_m = 2^m; \quad m \in [0, \ldots, M-1] \tag{14.8}$$

unterabgetastet werden, ohne dass Aliasing-Effekte auftreten.

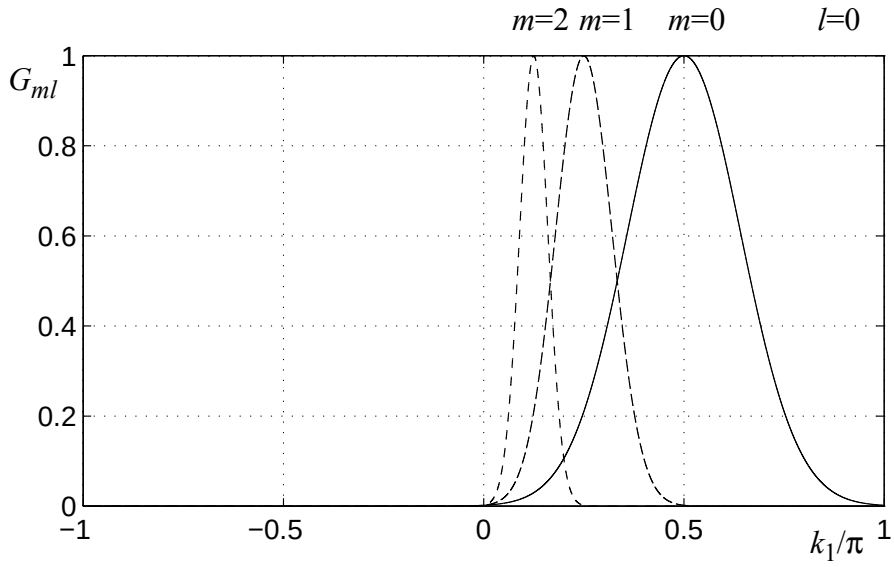


Abb. 14.1: Der Betrag der drei Gabor-Filter im Frequenzbereich

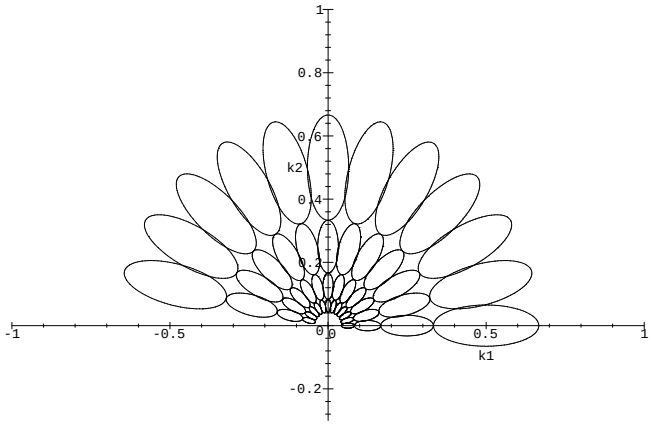


Abb. 14.2: Anordnung der Filter in der Frequenzebene

Für die Erzeugung der Auflösungspyramide werden auf den Bildmittelpunkt zentrierte Regionen unterschiedlicher Größe gefiltert und derart unterabgetastet, dass jede Region durch gleichviele Abtastpunkte repräsentiert wird. Somit bekommen wir kleine Regionen in der Bildmitte in hoher Auflösung und große Regionen in grober Auflösung.

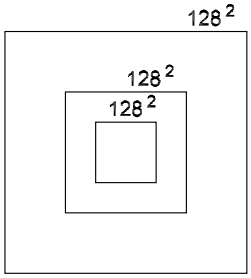


Abb. 14.3: Ausgehend von einem 512^2 Bild werden verschieden große Bildausschnitte jeweils mit 128^2 Pixeln repräsentiert.

Durch diese Wahl der Parameter zeigen die Filter ein Verhalten, das dem von Quadratur-Filtern ähnelt. Der Realteil und der Imaginärteil der Filterantwort zeigen eine Phasendifferenz von etwa $\pi/2$. Dies wird auf zwei Weisen ausgenutzt: das lokale Maximum des Betrages der Filterantwort wird als Konturinformation extrahiert und für die Erkennung verwendet. Darüber hinaus kann durch den Phasenversatz in Stereoanwendungen eine Tiefenkarte mit Subpixelgenauigkeit berechnet werden.

14.2 Anhang 2: Der HSI-Farbraum

Ein in der Fabbildverarbeitung sehr häufig verwendeter Farbraum ist der HSI-Farbraum. Ebenso wie der bekanntere RGB-Farbraum wird auch dieser durch drei Komponenten (*Hue*, *Saturation*, *Intensity*) aufgespannt. Sein wesentlicher Vorteil gegenüber dem RGB-Farbraum ist, dass die Hue-Komponente nur sehr wenig auf Beleuchtungsveränderung reagiert. Während sich im RGB-Farbraum bei Veränderung der Beleuchtungsintensität alle drei Komponenten verändern, bleibt im HSI-Farbraum die *Hue*-Komponente stabil und die Veränderung wird nur durch die *Intensitäts*-Komponente repräsentiert.

Die in Transformation vom RGB in den HSI-Farbraum wird dabei nach folgender Transformationsvorschrift durchgeführt.

$$\begin{aligned}
&\text{Farbwinkel (Hue)} && H = \operatorname{acos} \frac{1/2 \cdot (2R - G - B)}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \\
&\text{Sättigung (Saturation)} && S = 1 - 3 \cdot \frac{min\{R, G, B\}}{R + G + B} \\
&\text{Intensität (Intensity)} && I = \frac{R + G + B}{3}
\end{aligned} \tag{14.9}$$

Bei der Auswertung eines entsprechend transformierten Farbbildes muss jedoch berücksichtigt werden, dass aufgrund von Singularitäten die *Hue*-Komponente nur bei ausreichend großer Sättigung und Intensität verwendet werden kann. Ebenso zeigt auch die *Sättigungs*-Komponente nur bei ausreichend großer *Intensität* ein stabiles Verhalten gegen Rauscheinflüsse. Daher kann also bei ungenügender Sättigung oder Intensität keine wirkliche Farbsegmentierung erfolgen. In diesen Fällen wird daher in unserem System an diesen Bildpunkten nur die *Intensitäts*-Komponente ausgewertet. Diese Vorgehensweise wird ausführlich in [2] beschrieben. Abbildung 14.4 zeigt den Doppelkegel des HSI-Farbraums mit den Bereichen ungenügender Sättigung und Intensität.

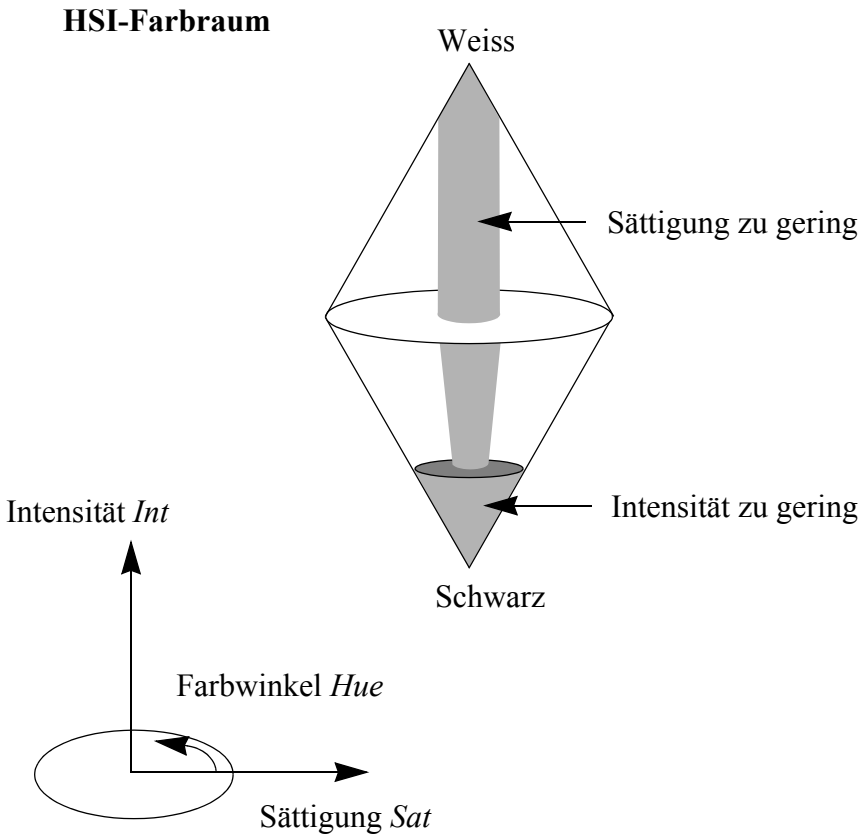


Abb. 14.4: Doppelkegel des HSI-Farbraums (in Anlehnung an [2], S. 106)

14.3 Anhang 3: Das Stereoverfahren

Analog zur Erkennung verfolgt das Stereoverfahren einen konturbasierten Ansatz. Die Merkmalsextraktion wird mit einem Satz von orientierten Gabor-Filtern in verschiedenen Auflösungen durchgeführt (vgl. auch Kap. 4.2.6). Die Faltung des Produktes von linker Filterantwort r_l und räumlich verschobener konjugiert komplexer Filterantwort r_r mit einem reellwertigen Fenster w liefert ein komplexes Ähnlichkeitsmaß ρ_{lr} zwischen den gefilterten Bildern. Das Ähnlichkeitsmaß wird durch die lokale Signalenergie der Filterantworten normiert.

$$\rho_{lr}(\boldsymbol{x}, \boldsymbol{d}) = \frac{w(\boldsymbol{x})_T(r_l(\boldsymbol{x})r_r^*(\boldsymbol{x}+\boldsymbol{d}))}{\sqrt{w(\boldsymbol{x})_T|r_l(\boldsymbol{x})|^2}\sqrt{w(\boldsymbol{x})_T|r_r(\boldsymbol{x}+\boldsymbol{d})|^2}} \tag{14.10}$$

Aufgrund der Rektifikation ist die vertikale Komponente des Disparitätsvektors gleich Null. Somit reduziert sich die Faltung auf eine Dimension. Die reelle Fensterfunktion w wird durch eine eindimensionale Gauß-Funktion, deren Ausdehnung gleich der Impulsantwort der Filter ist, gebildet. Sind die Filterantworten in einem lokalen Bereich ähnlich, so nimmt der Realteil des Ähnlichkeitsmaßes ein Maximum und der Imaginärteil ein betragsmäßiges Minimum (Nulldurchgang) an.

Um Fehlzuweisungen durch interokuläre Differenzen wie Rauschen und Verzerrungen und durch periodische Strukturen zu minimieren, wird nach der Korrelation ein Selbstorganisationsprozess gestartet. In diesem Relaxationsansatz werden anhand von Bedingungen der physikalischen Umwelt, wie der Eindeutigkeits- und Kontinuitätsbedingung, Mehrdeutigkeiten aufgelöst [111]. Für jede Bildkoordinate $\boldsymbol{x}$ und jede mögliche Disparität $\boldsymbol{d}$ wird eine reelle, zeitabhängige Variable $\xi(\boldsymbol{x}, \boldsymbol{d}, t)$ definiert. Der Realteil der komplexen Korrelation dient zur Initialisierung des Selbstorganisationsprozesses. Basierend auf einem Ansatz von Reimann und Haken [151] und erweitert um eine implizite Detektion von Okklusionen [175] wird ein Selbstorganisationsprozess gestartet, der durch folgende gekoppelte, nicht lineare Differentialgleichung beschrieben wird:

$$\begin{aligned}\dot{\xi}(\boldsymbol{x}, \boldsymbol{d}, t) = \{ \quad & A - C\xi^2(\boldsymbol{x}, \boldsymbol{d}, t) \\ & - \frac{B}{3K} \sum_{\boldsymbol{d}' \neq \boldsymbol{d}} \xi^2(\boldsymbol{x}, \boldsymbol{d}', t) \\ & - \frac{B}{3K} \sum_{\boldsymbol{d}' \neq \boldsymbol{d}} \xi^2(\boldsymbol{x} + \boldsymbol{d} - \boldsymbol{d}', \boldsymbol{d}', t) + \xi^2\left(\boldsymbol{x} + \frac{1}{2}(\boldsymbol{d} - \boldsymbol{d}'), \boldsymbol{d}', t\right) \\ & + \frac{D}{L} \sum_{x' \in U} \xi(\boldsymbol{x}', \boldsymbol{d}, t) \Big\} \xi(\boldsymbol{x}, \boldsymbol{d}, t)\end{aligned} \tag{14.11}$$

Dabei sind A, B, C, D positive Konstanten. K und L sind die Anzahl der an dem Selbstorganisationsprozess beteiligten Variablen und werden nur zur Normierung benötigt. Während der erste Term in der ersten Zeile ein exponentielles Wachstum beschreibt, beschränkt der zweite Term die Amplitude von $\xi(\boldsymbol{x}, \boldsymbol{d}, t)$. Der Term der zweiten Zeile wird benötigt, um das Eindeutigkeitsbedingung zu beschreiben und führt zu einem Wettbewerb zwischen allen Variablen mit gleicher Bildkoordinate. Durch den Term in der dritten Zeile wird verhindert, dass in Verdeckungsbereichen fälschlicherweise eine Disparität ermittelt wird. Zur Wahrung der Kontinuitätsbedingung werden letztlich noch alle Variablen einer kleine Nachbarschaft U mit gleichem Disparitätswert durch den Term der vierten Zeile gekoppelt.

15 Literatur

[1] Aloimonos, J.; Weiss, I.; Bandyopadhyay, A.: Active Vision. In: Proc. of DARPA Image Understanding Workshop, 1987, S. 552-573.

[2] Austermeier, H.: Farbkonstanz, Segmentierung, Klassifikation - ein Gesamtsystem zur effizienten Farbbildanalyse in einer Robot-Vision-Applikation. Berlin (Logos-Verlag), 1998.

[3] Ayache, N.; Hansen, C.: Rectification of Images for Binocular and Trinocular Stereovision. In: Proceedings of the 9th Intern. Conf. on Pattern Recognition, 1988, S. . 11-16.

[4] Bajcsy, R.; Solina, F.: Three-dimensional object representation revisited. In: Proc. Int. Conf. Computer Vision, London, 1987, S. 231-240

[5] Bajcsy, R.: Active perception. Proc. IEEE, Vol. 76, 8, 1988, S. 996-1006.

[6] Ballard, D.H.: Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition, Vol. 13, 2, 1981, S. 111-122.

[7] Ballard, D.H.; Brown, C.M.: Computer Vision. Englewood Cliffs N.J. (Prentice-Hall), 1982.

[8] Ballard, D.H.: Animate Vision, Artificial Intelligence, Vol. 48, 1991, S. 57-86.

[9] Barr, A.; Feigenbaum, E.A.: The Handbook of Artificial Intelligence, Vol.1. Los Altos (William Kaufmann), 1981

[10] Besl, P.J.; Jain, R.C.: Three-dimensional object recognition. Computing Surveys, Vol. 17, 1, 1985, S. 75-145.

[11] Biederman, I.: Recognition by components: a theory of human image understanding. Psychological Review, Vol. 94, 1987, S. 115-145.

[12] Biederman, I; Gerhardstein, P.C.: Viewpoint-dependent mechanisms in visual object recognition. Journal of Experimental Psychology: Human Perception and Performance, Vol. 21, 6, 1995, S. 1506-1514.

[13] Binford, T.O.: Body-centered representation and perception. In: [80], S. 207-216.

[14] Binford, T.O.: Visual perception by computer. IEEE Conference on Systems and Control. Miami, Florida, 1971.

[15] Book, M.: Ein Grafikprogramm zur Darstellung von semantischen HSC-Netzwerken. Studienarbeit (unveröffentlicht), Universität-Gesamthochschule Paderborn, 1992

[16] Book, M.: Konzipierung und Implementierung einer grafischen Animation für das Bilderkennungssystem PANTER. Diplomarbeit (unveröffentlicht), Universität-Gesamthochschule Paderborn, 1993

[17] Borgefors, G.: Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, 6, S. 849-865.

[18] Brachman, R.J.: What's in a Concept: Structural Foundations for Semantic Networks. Cambridge u.a. (Bolt Beranek & Newman), 1977.

[19] Brachman, R.J.: On the Epistemological Status of Semantic Networks. In: Findler, N.V. Associative Networks: Representation and Use of Knowledge by Computers. New York u.a. (Academic Press) 1979, S .3-50.

[20] Bradski, G.; Carpenter, G.; Grossberg, S.: Working memory networks for learning temporal order with application to three-dimensional visual object recognition. Neural Comput., Vol. 4. 1992, S. 270-286.

[21] Bradski, G.; Grossberg, S.: Fast-Learning VIEWNET Architectures for Recognizing Three-dimensional Objects from Multiple Two-dimensional Views. In: Neural Networks, Vol. 8, 7/8, 1995, S. 1053-1080.

[22] Bräunl, T., Feyrer, S., Rapf, W., Reinhardt, M.: *Parallele Bildverarbeitung*; Addison-Wesley, Bonn,1995

[23] Brown, C.: Gaze behaviors for robotics. In: Sood, A.K.; Wechsler, H. (Hrsg.): Active Perception and Robot Vision. Berlin (Springer), 1992, S. 115-139.

[24] Bülthoff, H.H.; Little, J.J.; Poggio, T.: A parallel algorithm for real-time computation of optical flow. Nature, 337, 1989, S. 549-553.

[25] Bülthoff, H.H.; Edelman, S.: Psychophysical support for a two-dimensional interpolation theory of object recognition. Proceedings of the National Academy of Sciences, USA, 89, 1992, S. 60-64.

[26] Büker, U.; Hartmann, G.: Wissensbasierte Bilderkennung mit neuronal repräsentierten Merkmalen. In: Sagerer, G.; Kummert, F.; Posch, S. (Hrsg.): Mustererkennung 1995; Springer-Verlag, Berlin, 1995, S.586-593

[27] Büker, U.; Hartmann, G.: Knowledge based view control of a neural 3-D object recognition system. In: Proceedings of 13th Conference on Pattern Recognition (ICPR'96 Wien). Los Alamitos (IEEE Computer Society Press) 1996, Vol. IV, S. 24-29.

[28] Büker, U.: Wissensbasierte Bilderkennung mit symbolischen und neuronal repräsentierten Merkmalen; Fortschritt-Berichte, Reihe 10, Nr. 425, Düsseldorf, VDI-Verlag, 1996.

[29] Büker, U.; Kalkreuter, B.: Learning in an Active Hybrid Vision System. In: Jain, A.K.; Vekatesh, S.; Lovell, B. C. (Hg): Proceedings of the 14th International Conference on Pattern Recognition. ICPR'98, Los Alamitos (IEEE Computer Society) 1998, S. 178 - 181.

[30] Büker, U.: Hybrid Object Models for Robot Vision. In: Proceedings of the 24th Annual Conference on the IEEE Industrial Electronics Society, Vol. 4/4. IECON'98, Aachen. Piscataway, NJ (IEEE) 1998, S. 2045 - 2050.

[31] Büker, U.: Hybrid Object Models: Combining Symbolic and Subsymbolic Object Recognition Strategies. In: Callaos, N.; Omolayole, O.; Wang, L. (Hg.): Proceedings of the 4th International Conference on Information Systems, Analysis and Synthesis, Volume 1. ISAS'98. Orlando (IIIS) 1998, S. 444-451.

[32] Büker, U.; Drüe, S.; Götze, N.; Hartmann, G.; Kalkreuter, R. Stemmer, R.; Trapp, R.: An Active Object Recognition System for Disassembly Task. In: Proceedings of the 7th IEEE International Conference on Emerging Technologies and Factory Automation. ETFA '99, Piscataway (IEEE), 1999, S. 79 - 88.

[33] Cahn v. Seelen, U.; Madden, B.: Binocular Camera Heads. Internet URL: http://www.cis.upenn.edu/~grasp/head/headpage/heads.html

[34] Carbonell, J.G. (Hrsg): Special volume on machine learning. Artificial Intelligence, 40, 1989.

[35] Carbonell, J., G.: Machine learning. Cambridge, MA (MIT-Press), 1990.

[36] Carpenter, G.A.; Grossberg, S.: A massively parallel architecture for self-organizing neural pattern recognition machine. Computer Vision, Graphics, and Image Processing, Vol. 37, 1987, S. 54-115.

[37] Carpenter, G.A.; Grossberg, S.: ART2: Selforganization of stable category recognition codes for analog input patterns. Applied Optics, Vol. 26, 1987, S. 4916-4930.

[38] Carpenter, G.A.; Grossberg, S.; Rosen, D.: Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks, Vol. 4,1, 1990, S. 759-771.

[39] Carpenter, G.; Grossberg, S.; Rosen, D.B.: Art 2-A: an adaptive resonance algorithm for rapid category learning and recognition. Neural networks, Vol. 4, 1991, S. 493-504.

[40] Carpenter, G.A.; Grossberg, S.; Reynolds, J.: ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. Neural Networks, Vol. 4, 1991, S. 565-588.

[41] Carpenter, G.A.; Grossberg, S.; Markuzon, N.; Reynolds, J.: Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans. onNeural networks. Vol. 3, 1992, S. 698-713.

[42] Charniak, E.: Baysian networks without tears. AI Magazine, 12, 4, 1991, S.50-63.

[43] Chaudhary, V., Aggarwal, J.K.: Parallelism in Computer Vision: A Review. In: Kumar; Gopalakrishnan; Kanal (Hrsg.): Parallel Algorithms for Machine Intelligence and Vision; Springer-Verlag, New York, 1990, S. 271-309

[44] Chen, C.H.; Honavar, V.: Neural Network Automata. In: World Congress on Neural Networks Vol.4, S.470-477

[45] Chin, R.T.; Dyer, C.R.: Model-Based Recognition in Robot Vision. In: Computing Surveys, Vol. 15, 1, 1986, S. 67-108.

[46] Christensen, H.L.: The AUC robot camera head. Proc. of the SPIE, Vol. 1708, 1992, S. 26-33.

[47] Chua, L.O.; Roska, T.: The CNN Paradigm. IEEE Transactions on Circuits and Systems (Part I), Vol. 40, 3, 1993, S. 559-577.

[48] Denavit, J.; Hartenberg, R.S.: A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. ASME Journal of Appl. Mechanics, 77, 1955, S. 215-221.

[49] Dickinson,, S.J.; Pentland, A.P.; Rosenfeld, A.: From volumes to views: an approach to 3-D object recognition. CVGIP: Image Understanding, Vol. 55, 2, 1992, S. 130-154.

[50] Dickinson, S.J.; Christensen, H.I.; Tsotsos, J.K.; Olofsson, G.: Active object recognition integrating attention and viewpoint control. Computer Vision and Image Understanding, Vol. 67, 3, 1997, S. 239-260.

[51] Dickmanns, E.D.; Gräfe, V.: a) Dynamic monocular machine vision, b) Application of dynamic monocular machine vision. J. Machine Vision & Application, 1988, S. 223-261.

[52] Dunker, J.; Hartmann, G.; Stöhr, M.: A multiple-view approach to 3D recognition based on complex model neurons. In: Proc. of the Int. Conf. on Artificial Neural Networks (ICANN '95), Vol. 2, 1995, S.281-286.

[53] Dunker, J.; Hartmann, G.; Stöhr, M: Single view recognition and pose estimation of 3D-objects using sets of prototypical views and spatially tolerant contour representations. In: Proceedings of 13th Conference on Pattern Recognition (ICPR'96 Wien). Los Alamitos (IEEE Computer Society Press) 1996, Vol. IV, S. 14-18.

[54] Dunker, J.: Prototypengestützte Objekterkennung auf der Basis ortstoleranter Konturrepräsentationen. Berlin (Logos), 1998.

[55] Edelman, S.; Bülthoff, H.H.: Orientation dependence in the recognition of familiar and novel views of three-dimensional objects. Vision Research, Vol. 32, 1992, S. 2385-2400.

[56] Fischer, V.: Parallelverarbeitung in einem semantischen Netzwerk für die wissensbasierte Musteranalyse; Dissertation, Universität Erlangen, 1994

[57] Fischer, V., Niemann, H., Paulus, D., Winzen, A.: Ein paralleler Kontrollalgorithmus für die wissensbasierte Bildanalyse; in Reichel (Hrsg.): Informatik-Wirtschaft-Gesellschaft, 23.GI-Jahrestagung, Dresden; Springer-Verlag, Berlin, 1993, S. 515-519

[58] Fleet, D.J.; Jepson, A.D.; Jenkin, M.R.M.: Phase-Based Disparity Measurement. Computer Vision, Graphics and Image Processing: Image Understanding, Vol. 53, 2, 1991, S. 198-210.

[59] Förstner, W.: Quality assessment of object location and point transfer using digital image correlation techniques. In: Proc. of the 15th ISPRS Congress, Rio de Janeiro. 1984, S. 169-191.

[60] Forsyth, D.A.; Zisserman, A.: Reflections on Shading. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13, 7, 1991, S. 671-679.

[61] Fukunaga, K.: Introduction to statistical Pattern Recognition. London (Academic Press), 1990.

[62] Fukushima, K.: A neural network model for selective attention in visual pattern recognition. Biological Cybernetics, Vol. 55, 1986, S. 5-15.

[63] Gabor, D.: Theory of communication. J. IEE, Vol. 93, 1946, S. 429-457.

[64] Genesereth, M.R.; Nilsson, N.J.: Logische Grundlagen der Künstlichen Intelligenz. Braunschweig u.a. (Vieweg-Verlag), 1989

[65] Giefing, G.J.; Janssen, H.; Mallot, H.: A saccadic camera movement system for object recognition. In: Kohonen, T. u.a. (Hrsg.): International Conference on Artificial Neural Networks, Amsterdam (North-Holland), 1991, S.63-68

[66] Giles, C.L.; Omlin, C.W.: Rule refinement with recurrent neural networks. In: 1993 IEEE International Conference on Neural Networks Vol.2, 1993, S.801-806.

[67] Grimson, W.E.L.: A computer implementation of a theory of human stereo vision. Philosophical Transactions of the Royal Society of London, Vol. B292, 1981, S. 217-253.

[68] Gremban, K.D.; Ikeuchi,K.: Appearance-Based Vision and the Automatic Generation of Object Recognition Programs. In: Jain, A.K.; Flynn, P.J. (Hrsg.): Three-Dimensional Object Recognition Systems. Amsterdam (Elsevier), 1993, S. 229-258.

[69] Gremban, K.D.; Ikeuchi,K.: Planning multiple observations for object recognition. Int. Journal of Computer Vision, Vol. 12, 2/3, 1994, S. 137-172.

[70] Gross, A.D.; Boult, T.E.: Error fit measures for recovering parametric solids. In: Proc. International Conference on computer Vision, 1988, S. 690-694.

[71] Gross, H.-M.; Franke, R.; Böhme, H.-J.; Beck, C.: A neural network hierarchy for data driven and knowledge controlled selective visual attention. In Fuchs, S.; Hoffmann, R.: Mustererkennung 1992, Berlin (Springer), 1992, S. 341-346.

[72] Grossberg, S.: Adaptive pattern classification und universal recoding, I: Parallel development and coding of neural feature detectors. Biological Cybernetics, Vol. 23, 1976, S. 121-134.

[73] Grossberg, S.: Adaptive pattern classification und universal recoding, II: Feedback, expectation, olfaction, and illusions. Biological Cybernetics, Vol. 23, 1976, S. 187-202.

[74] Grossberg, S. (Hrsg.): The adaptive brain, Vol. 1 und 2. Amsterdam (Elsevier / North Holland), 1987.

[75] Grosso, E.; Sandini, G.; Tistarelli, M.: 3-D Object REcognition Using Stereo and Motion. IEEE Trans. on Systems, Man, and Cybernetics, Vol. 19, 6, 1989, S. 1465-1476.

[76] Hager, G.D.; Hutchinson, S.; Corke, P.I.: A tutorial on visual servo control. IEEE Int. Conf. on Robotics and Automation. Tutorial Notes TT3, 1996.

[77] Hartmann, G.; Kräuter, K.O.; Wiemers, H.; Seidenberg, E.; Drüe, S.: Ein distanz- und orientierungsinvariantes lernfähiges Erkennungssystem für Robotikanwendungen. In: Pöppl,S.; Handels, H. (Hrsg.): Mustererkennung 1993, Berlin (Springer), 1993, S. 375-382.

[78] Hartmann, G., Niemann, H., Sagerer, G., Kummert, F., Mertsching, B.: Semantische Netzwerksysteme in der Musteranalyse. Künstliche Intelligenz (KI), 3, Themenheft Mustererkennung, S. 23-29.

[79] Hartmann, G.; Büker, U.; Drüe, S.: A Hybrid Neuro-Artificial Intelligence-Architecture. In: Jähne, B. et al. (Hrsg.): Handbook on Computer Vision and Applications, Vol 3. San Diego (Academic Press), 1999, S. 153-196.

[80] Hebert, M.; Ponce, J.; Boult, T.; Gross, A. (Eds.): Object Representation in Computer Vision. Berlin (Springer), 1995.

[81] Heidemann, G.; Ritter, H.: Combining multiple neural nets for visual feature selction and classification. In: Proc. of the 9th Int. Conf. on Artificial Neural Networks, ICANN 99, S. 365-370, 1999.

[82] Hempel, O.; Büker, U.: A Parallel Control Algorithm for Hybrid Image Recognition. In: Yi Pan; Selim G. Akl; Keqin Li: Parallel and Distributed Computing and Systems. Proceedings of the 10th IASTED International Conference. Anaheim (IASTED/ACTA Press) 1998, S. 206 - 209.

[83] Hempel, O.; Büker, U.: Parallelisierung eines wissensbasierten Bilderkennungssystems mit integrierten neuronalen Netzwerken. Heinz Nixdorf Institut, Universität Paderborn, interner Bericht PAWIAN 10/99.

[84] Henrion, M.; Breese, J.S.; Horvitz, E.J.: Decision analysis and expert systems. AI Magazine, 12,4, 1991, S.64-91.

[85] Huttenlocher, D.; Ullman, S.: Object recognition using alignment. In Int. Conf. on Computer Vision. 1987, S. 102-111.

[86] Jacobs, D.W.: Grouping for Recognition. M.I.T. AI Lab Memo, 1177, 1989.

[87] Jacobs, D.W.: Space efficient 3D model indexing. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1992, S. 439-444.

[88] Kaindl, H.: Problemlösen durch heuristische Suche in der Artificial Intelligence. Wien (Springer), 1989.

[89] Kane, R.; Milgram, M.: Extraction of semantic rules from trained multilayer neural networks. In: IEEE International Conference on Neural Networks Vol.3, 1993, S. 1397-1401

[90] Khosla, R.; Dillon, T.S.: Task decomposition and competing expert system-artificial neural net objects for reliable and real time inference. In: 1993 IEEE International Conference on Neural Networks Vol.2, 1993, S. 794-800.

[91] Koendering, J.J.; van Doorn, A.J.: The internal representation of solid shape with respect to vision. Biological Cybernetics, Vol. 32, 1979, S. 211-216.

[92] Konen, W.; v.d. Malsburg, C.: Learning to generalize from single examples in the dynamic link architecture. Neural Computation, 5, 1993, S. 719-735

[93] Konen, W.; Maurer, T.; v.d. Malsburg, C.: A fast link matching algorithm for invariant pattern recognition. Neural Networks, Vol. 7, 1994, S. 1019-1030.

[94] Kräuter, K.-O.: Distanz- und orientierungsinvaiante Extraktion von Konturinformation aus den Kameradaten eines Robot-Vision-Systems. Dissertation, Universität-GH Paderborn, 1995.

[95] Kriegman, D.; Ponce, J.: On recognizing and positioning curved 3D objects from image contours. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12,12, 1990, S. 1127-1137.

[96] Kriegman, D.J.; Ponce, J.: Repräsentations for recognizing complex curved 3D Objects. In: [80], S. 125-138.

[97] Kumar, S.; Han, S.; Goldgof, D.; Bowyer, K.: On recovering hyperquadrics from range data. IEEE Trans. Pattern Analysis and Mchine Intelligence, Vol. 17, 1995, S. 1079-1083.

[98] Kummert, F. u.a.: A Hybrid Approach to Signal Interpretation Using Neural and Semantic Networks. In: Pöppl, S.J.; Handels, H. (Hrsg.): Mustererkennung 1993, Berlin (Springer), 1993, S. 245-252.

[99] Kummert, F.; Niemann, H.; Prechtel, R.; Sagerer, G.: Control and explanation in a signal understanding environment. Signal Processing, 32, 1993, S. 111-145.

[100] Kummert, F.: Interpretation von Bild- und Sprachsignalen - ein hybrider Ansatz. Aachen (Shaker Verlag), 1998.

[101] Kummert, F.; Fink, A.; Sagerer, G.: Schritthaltende hybride Objektdetektion. In: Paulus, E.; Wahl, F.M. (Hrsg.): Mustererkennung 1997. Berlin (Springer), 1997, S. 137-144.

[102] Kunde, M.: Wissensbasierte Modellierung dreidimensionaler Objekte für ein Active Vision System. Diplomarbeit (unveröffentlicht), Universität-GH Paderborn, 1996.

[103] Lanser, S.; Zierl, Ch.; Munkelt, O.; Radig, B.: MORAL - A Vision-based Objekt Recognition System for Autonomous Mobile Systems. In: Proc. of 7th CAIP, Berlin, Springer, 1997, S. 33-41.

[104] Lewis, B., Berg, D.J.: Threads Primer - A Guide to Multithreaded Programming, Sunsoft Press/Prentice Hall, 1996

[105] Little, J.J.; Boyd, J.E.: Recognizing people by their gate: the shape of motion. Videre, 1, 2, 1998.

[106] Liu, D.; Michel, A.: Sparsely Interconnected Neural Networks for Associative Memories with Applications to Cellular Neural Networks. IEEE Transactions on Circuits and Systems. Vol. 41, 4, 1994, S. 295-307.

[107] Lowe, D.G.: Perceptual Organization and Visual Recognition, Boston (Kluwer Academic Publishers), 1985.

[108] Lowe, D.G.: Three-dimensional object recognition from single two-dimensional images. Artificial Intelligence, 31, 1987, S 355-395.

[109] Lowe, D.G.: The viewpoint consistency constraint. Int. Journal on Computer Vision, Vol.1,1, 1987, S. 57-72.

[110] Madden, B.C.; Cahn v. Seelen, U.M.: PennEyes - A Binocular Active Vision System. Technical Report MS-CIS-95-37/Grasp LAB 396. University of Pennsylvania, Philadelphia, GRASP Laboratory, 1996.

[111] Marr, D.; Poggio, T.: A computational theory of human stereo vision. In: Proceedings of Royal Society of London, Vol. B 204, 1979, S. 301-328.

[112] Marr, D.C.: Vision. Freeman (San Francisco, CA), 1982.

[113] Massad, A.; Mertsching, B.; Schmalz, S.: Utilizing Temporal Associations for View-Based 3-D Object Recognition. In: Proc. of the 24th Conf. of the IEEE Industrial Electronics Society (IECON '98), Vol. 4, 1998, S. 2074-2078.

[114] Maver, J; Bajcsy, R.: Occlusion as a guide for planning the next view. IEEE Trans. Pattern Analysis and Machine Intelligence, 15, 5, 1993, S. 417-433.

[115] McGarry, K.; Wermter, S.; MacIntyre, J.: Hybrid Neural Systems: From Simple Coupling to Fully Integrated Neural networks. Neural Computing Surveys, 2, 1999, S. 62-93.

[116] McMillan, C.; Mozer, M.C.; Smolensky, P.: Dynamic Conflict Resolution in a Connectionist Rule-Based System. In: 13th International Joint Conference on Artificial Intelligence, San Mateo (Morgan Kaufmann), 1993, S. 1366-1373

[117] Medsker, L.: Hybrid Neural Network and Expert Systems. Boston (Kluwer Academic Publishers), 1994.

[118] Meng, M.: A neural production system and its application in visual-guided mobile robot navigation. In: IEEE International Conference on Neural Networks Vol.2, 1993, S. 807-812.

[119] Mertsching, B.: Lernfähiges wissensbasiertes Bilderkennungssystem auf der Grundlage des Hierarchischen Strukturcodes. Fortschrittberichte VDI 10, Nr.191, Düsseldorf (VDI-Verlag), 1991

[120] Metaxas, D.: A physics-based framework for segmentation, shape and motion estimation. In: [80]. S. 233-248.

[121] Michalski, R.; Carbonell, J.G.; Mitchell, T. (Hrsg.): Machine Learning: An Artificial Intelligence Approach. Vol. 1, Palo Alto (Tioga), 1983.

[122] Milanese, R.; Bost, J.-M.; Pun, T.: A relaxation network for a feature-driven visual attention system. SPIE Neural and Stochastic Methods in Image and Signal Processing, Vol. 1766, S.542-553, 1992.

[123] Milanova, M.; Büker, U.: Object Recognition in Image Sequences with Cellular Neural Networks. to appear in: Neurocomputing, 1999.

[124] Milanova, M.; Büker, U.: Cellular Neural Networks for Complex Object Recognition. Int. ICSC/IFAC Symposium on Neural Computation (NC '98), Wien, 1998, S. 304-310.

[125] Moghaddam, B.; Pentland, A.: Probabilistic Visual Learning for Object Representation. IEEE Transactions on pattern analysis and Machine Intelligence, Vol. 19/2, 1997, S. 696-710.

[126] Moratz, R.; Posch, S.; Sagerer, G.: Controlling Multiple Neural Nets with Semantic Networks. In: Kropatsch, W.G.; Bishof, H: Mustererkennung 1994, Technische Universität Wien, 1994, S. 288-295

[127] Moravec, H.P.: Towards automatic visual obstacle avoidance. In: 5th Int. Joint Conf. Artificial Intelligence. Cambridge, MA, S. 584-590.

[128] Morik, K.: Knowledge Acquisition and machine learning. London (Academic Press), 1993.

[129] Morik, K.: Maschinelles Lernen. In: Görz, G. (Hrsg.): Einführung in die künstliche Intelligenz. Bonn (Addison-Wesley), 1995.

[130] Murase, H.; Kimura, F.; Yoshimura, M.; Miyake, Y.: An improvement of the Auto-Correlation Matrix Pattern Matching Method and its Application to Handprinted HIRAGANA. Trans. IECE, Vol. J64-D, 3, 1981, S. 276-283.

[131] Murase, H.; Nayar, S.K.: Visual Learning and Recognition of 3-D Objects from Appearance. Int. Journal of Computer Vision, Vol. 14, 1995, S. 5-24.

[132] Nayar, S.K.; Murase, H.: Dimensionality of illumination manifolds in appearance matching. In: Ponce, J.; Zisserman, A.; Hebert, M. (Eds.): Object Representation in Computer Vision II. Berlin (Springer), 1996, S. 165-178.

[133] Nayar, S.K.; Nene, S.A.; Murase, H.: Subspace methods for Robot Vision. IEEE Trans. on Robotics and Automation, Vol. 12, 5, 1996, S. 750-758.

[134] Nevatia, R.; Binford, T.: Description and recognition of complex curved objects. Artificial Intelligence, Vol.8, 1977, S. 77-98.

[135] Niemann, H., Bunke, H.: *Künstliche Intelligenz in Bild- und Sprachanalyse*; B.G.Teubner, Stuttgart, 1987

[136] Niemann, H.; Sagerer, G.; Schroder, S.; Kummert, F.: ERNEST: A Semantic Network System for Pattern Understanding. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 12, 1990, S. 883-905.

[137] Niemann, H.: Knowledge-Based Interpretation of Images. In: Jähne, B. et al. (Hrsg.): Handbook on Computer Vision and Applications, Vol II, San Diego, Ca. (Academic Press), 1999, S. 855-874.

[138] Hornegger, J.; Nöth, E.; Fischer, V.; Niemann, H.: Semantic networks meet bayesian classifiers. In: Jähne, B. u.a. (Hrsg.): Mustererkennung 1996. Berlin (Springer), S. 260-267.

[139] Nilsson, N.J.: Principles of Artificial Intelligence. Berlin u.a. (Springer), 1982

[140] Nissan, E.; Siegelmann, H.; Galperin, A.: An Integrated Symbolic and Neural Network Archtiecture for Machine Learning in the Domain of Nuclear Engineering. In: International Conference on Pattern Recognition Vol.2, Los Alamitos (IEEE Computer Society Press), 1994, S. 494-496

[141] Noda, I.: A Model of Recurrent Neural Networks that Learn State-Transitions of Finite State Transducers. In: World Congress on Neural Networks Vol.4, S. 447-452

[142] Olson, C.F.; Huttenlocher, D.P.: Automatic Target Recognition by Matching Oriented Edge Pixels. IEEE Transactions on Image Processing, Vol. 6, 1, 1997, S. 103-113.

[143] Opitz, D.W.; Shavlik, J.W.: Heuristically Expanding Knowledge-Based Neural Networks. In: 13th International Joint conference on Artificial Intelligence, San Mateo (Morgan Kaufmann), 1993, S. 1360-1365

[144] Pahlavan, K.; Eklundh, J.-O.: A head-eye system - analysis and design. CVGIP: Image Understanding, Vol. 56, 1, 1992, S. 41-56.

[145] Paulus, D.; Ahlrichs, U.; Heigl, B.; Niemann, H.:: Wissensbasierte aktive Szenenanalyse. In: Levi, P.; Ahlers, R.J.; May, F.; Schanz, M. (Hrsg.): Mustererkennung 1998, Berlin (Springer), 1998, S. 185-192.

[146] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan kaufman, 1988.

[147] Perret,D.I.; Harries, M.H.; Bevan, R.; Thomas, S.; Benson, P.J.; Mistlin, A.; Chitty, A.J.; Hietanen, J.K.; Ortega, J.E.: Frameworks of analysis for the neural representation of animate objects and actions. Journal of Experimental Biology, 146, 1989, S. 87-113.

[148] Poggio, T.; Edelman, S.: A network that learns to recognize three-dimensional objects. Nature, Vol. 343, 1990, S. 263-266.

[149] Pomerleau, D.A.: Neural Networks for Intelligent Vehicles. In:Proc. of the Intelligent Vehicles Symposium 1993, Tokyo, 1993, S. 19-24.

[150] Quillian, M.R.: Semantic Memory. In: Minsky, M. (Hrsg.) Semantic Information Processing. Cambridge u.a. (MIT Press), 1986, S. 227-270.

[151] Reimann, D.; Haken, H.: Stereovision by Self-Organisation. Biological Cybernetics, 71, 1994, S. 17-26.

[152] Reisfeld, D.; Wolfson, H.; Yeshurun, Y.: Context-Free Attentional Operators: The Generalized Symmetry Transform. Int. Journal of Computer Vision, Vol. 14, S.119-130, 1995.

[153] Rimey, R.D.; Brown, C.M.: Selective attention as sequential behavior: Modeling eye movements with an augmented hidden Markov model. In: Proc. Image Understanding Workshop. San Mateo (Morgan Kaufmann), 1990, S. 840-849.

[154] Rimey, R.D., Brown, C.M.: Where to look next using a Bayes net: Incorporating geometric relations. In: Sandini, G. (Hrsg.): Computer Vision - ECCV '92, Berlin (Springer), 1992, S.542-550.

[155] Ritter, H.; Martinetz, T.; Schulten, K.: Neuronale Netze. Bonn (Addison-Wesley), 1994.

[156] Rojas, R.: Theorie der neuronalen Netze. Berlin (Springer), 1993

[157] Rosenfeld, A.: Robot Vision. In: Wong, A.K.C.; Pugh, A.: Machine Intelligence and Knowledge Engineering for Robotic Applications, Berlin (Springer), 1987, S. 1-19.

[158] Sagerer, G.: Darstellung und Nutzung von Expertenwissen für ein Bildanalysesystem. Berlin u.a. (Springer), 1985.

[159] Sagerer, G.: Automatisches Verstehen gesprochener Sprache. Mannheim (BI Wissenschaftsverlag), 1990.

[160] Sagerer, G.; Niemann, H.: Semantic Networks for Understanding Scenes. New York (Plenum Press), 1997.

[161] Sanfeliu, A.; Alquezar, R.: Active Grammatical Inference: A New Learning Methodology. In: Dori, D.; Bruckstein, A. (Hrsg.): Shape, Structure and Pattern Recognition. Singapur (World Scientific), 1994, S. 191-200.

[162] Shapiro, L.G.; Neal, P.J.; Ponder, K.: Relational Models for View Class Construction in 3D Object Recognition. In: Bunke, H. (Hrsg.): Advances in Structural and Syntactic pattern Recognition. Singapur (World Scientific), 1992, S. 401-410.

[163] Seibert, M.; Waxman, A.M.: Adaptive 3-d object recognition from multiple views. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, 2, 1992, S. 107-124.

[164] Siegert, H.-J.; Bocionek, S.: Robotik: Programmierung intelligenter Roboter. Berlin (Springer), 1996.

[165] Sirovich, L.; Kirby, M.: Low dimensional procedure for the characterization of human faces. Journal of Optical Society of America, Vol. 4, 3, 1987, S. 512-524.

[166] Shafer, G.: A Mathematical Theory of Evidence. Prinston (Prinston University Press), 1976.

[167] Shafer, S.: Shadows and Silhouettes in Computer Vision. Boston (Kluwer Acadamic Pub.), 1985.

[168] Shepard, R.N.; Cooper, L.A.: Mental images and their transformation. Cambridge, MA (MIT Press), 1982.

[169] Takahashi,Y.; Iizuka, T.; Ninomiya, H.: Standing-on-floor type tea serving welfare robot using voice instruction system. In: Proc. of the 24th annual Conf. of the IEEE Industrial Electronics Society (IECON ‘98), Vol. 2, Piscataway (IEEE), 1998, S. 1208-1213.

[170] Tarr, M.J.: Orientation dependence in three-dimensional object recognition. Dissertation, MIT, Department of Brain and Cognitive Sciences, 1989.

[171] Tarr, M.J.; Bülthoff, H.H.: Is human object recognition better described by geon structural descriptions or by multiple views? Journal of Experimental Psychology: Human Perception and Performance, Vol. 21, 6, 1995, S. 1494-1505.

[172] Terzopoulos, D.: From Physics-Based Representation to Functional Modeling of Highly Complex Objects. In: [80], S.347-359.

[173] Trapp, R., Drüe, S.; Mertsching, B.: Korrespondenz in der Stereoskopie bei räumlich verteilten Merkmalsrepräsentationen im Neuronalen-Active Vision System NAVIS. In: Sagerer, G.; Posch, S.; Kummert, F. (Hrsg.): Mustererkennung 1995, Berlin (Springer), 1995, S. 492-499.

[174] Trapp, R.; Drüe, S.: Ein flexibles binokulares Sehsystem: Konstruktion und Kalibrierung. In: Mertsching, B. (ed.): Proceedings in Artificial Intelligence., Sankt Augustin (Infix), 1996, S. 32-39

[175] Trapp, R.; Drüe, S.; Hartmann, G.: Stereo Matching with Implicit Detection of Occlussions. In: Burkhard, H.; Neumann, B.: Proceedings of the Fifth European Conference on Computer Vision, Berlin (Springer), S. 17-33, 1998.

[176] Trapp, R.: Stereoskopische Korrespondenzbestimmung mit impliziter Detektion von Okklusionen. Paderborn (Heinz Nixdorf Institut, Universität Paderborn), 1998.

[177] Trost, H.: Wissensrepräsentation in der AI am Beispiel Semantischer Netze. In: Retti, J. u.a. Artificial Intelligence - eine Einführung. Stuttgart (Teubner), 1986.

[178] Tsotsos, J.K.: On the relative complexity of active vs. passiv visual search. Int. Journal of Computer Vision, Vol. 7, 2, 1992, S. 127-141.

[179] Turk, M.A.; Pentland, A.P.: Face Recognition Using Eigenfaces. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Los Alamitos (IEEE Comp. Soc. Press), 1991, S. 586-591.

[180] Ullman, S.: Visual routines. Cognition, Vol. 13, 1984, S. 97-160.

[181] v.d. Malsburg, C.: The correlation theory of brain function. Internal report, Max-Planck-institut für Biophysikalische Chemie, Göttingen, 1981.

[182] v.d. Malsburg, C.: How are nervous structures organized? In: Basar, E., u.a. (Hrsg.): Synergetics of the Brain. Proc. of the Int. Symposium on Synergetics. Berlin (Springer), 1983, S. 238-249.

[183] v. Seelen, W.: A neural archtiecture for autonomous visually guided robots - results of the NAMOS project. Düsseldorf (VDI-Verlag), 1995.

[184] v. Seelen, W.; Bohrer, S.; Kopecz, J.; Theimer, W.M.: A Neural Architecture for Visual Information Processing. Int. Journal of Computer Vision, Vol. 16, 1995, S. 229-260.

[185] Sugiyama, K.; Tagawa, S.; Toda, M.: Methods for Visual Understanding of Hierarchical System Structures. In: IEEE Trans. Systems, Man, and Cybernetics 11, 2, 1981, S. 109-125.

[186] Tamassia, R.; Di Batista, G.; Batini, C.: Automatic Graph Drawing and Readability of Diagrams. In: IEEE Trans. Systems, Man, and Cybernetics 18, 11, 1988, S. 61-69.

[187] Waxman, A.M.; Seibert, M.; Gove, A.; Fay, D.A.; Bernadron, A.M.; Lazott, C.; Steele, W.R.; Cunningham, R.K.: Neural Processing of targets in visible, multispectral IR and SAR imagery. Neural Networks, Vol. 8, 7/8, 1995, S. 1029-1051.

[188] Weiss, L.E.; Sanderson, A.C.; Neumann, C.P.: Dynamic sensor-based control of robots with visual feedback. Journal of Robotics and Automation, Vol. 3, 5, S. 404-417.

[189] Whaite, P.; Ferrie, F.P.: From uncertainty to visual exploration. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 13, 1991, S. 1038-1049.

[190] Wilkes, D.; Tsotsos, J.K.: Active Object Recognition. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Los Alamitos (IEEE Comp. Soc. Press), 1992, S. 136-141.

[191] Winston, P.H.: Künstliche Intelligenz, Reading u.a. (Addison-Wesley), 1987

[192] Witkin, A.P.; Tenenbaum, M.: On the role of structure in vision. In: Rosenfeld, A.; Beck, J. (Hrsg.): Human and Machine Vision, New York (Academic Press), 1983, S. 481-543.

[193] Woods, W.A.: What's in a Link: Foundations for Semantic Networks. In: Bobrow, D.; collins, A. (Hrsg): Representation and Understanding. New York u.a. (Academic Press) 1975, S. 35-82.

[194] Wunsch, P.; Hirzinger, G.: Echtzeit-Lagebestimmung dreidimensionaler Objekte aus Bildsequenzen zur visuellen Lageregelung eines Industrieroboters. In: Mertsching, B. (Hrsg.): Aktives Sehen in technischen und biologischen Systemen. Proceedings in Artificial Intelligence, Vol.4. Sankt Augustin (infix-Verlag) 1996, S. 166-173.

[195] Yeshurun, Y; Schwartz, E.L.: Shape description with a space-variant sensor: Algorithm for scan-path, fusion, and convergence over multiple scans. IEEE Trans. on Pattern Analysis and Machine Intelligence, 11, S.1217-1222, 1989.