

Adaptive Erkennung von bewegten Fahrbahnobjekten in monokularen Bildfolgen mit Eigenbewegung

Włodzimierz Kasprzak

Kurzfassung

In dieser Arbeit wird ein System zur adaptiven, modellbasierten Objekterkennung in monokularen Bildfolgen von Verkehrsszenen vorgestellt, unter Annahme von mehreren, sich bewegenden Objekten und einer möglichen Eigenbewegung der Kamera. Als Anwendungsbereich wurden Szenen gewählt, wie sie im Straßenverkehr vorkommen können. Diese Arbeit befaßt sich dadurch mit perspektivisch verzerrten Verschiebungen im Bild, die durch die Fortbewegung der Kamera oder/und durch die Bewegung von zum Teil weit entfernten Objekten hauptsächlich entlang der Tiefenachse entstehen.

Es werden Verfahren zur Bewegungsdetektion im Bild, zur Fahrbahnerkennung und Eigenlageverfolgung sowie zur modellbasierten 3-D Objekterkennung vorgestellt. Für all diese Verfahren wird ein genereller, adaptiver Ansatz zur Objekterkennung definiert, der auf dem Prinzip der rekursiven Schätzung von Lage-, Bewegungs- und Formparametern basiert.

Die einzelnen Verfahren wurden integriert in ein prozedural implementiertes System, das auf einem Monoprozessorrechner getestet wurde. Das Prozeßmodell des Systems gliedert sich in drei Module: anwendungsunabhängige Segmentierung-Bewegungsdetektion, modellbasierte Fahrbahnerkennung und Eigenzustandsverfolgung sowie modellbasierte 3-D Objekterkennung. Das grundsätzlich für die Bildfolgenanalyse bei bewegter Kamera konzipierte System kann durch eine Rekonfigurierung für ähnliche Anwendungen mit stationärer Kamera eingesetzt werden. In diesem Fall können insbesondere die anwendungsunabhängigen Methoden zur Bewegungsschätzung im Bild stärker berücksichtigt werden.

Das Konzept einer wissensbasierten Realisierung des Erkennungssystems wird anschließend vorgestellt, das eine objektorientierte Modellierung des Wissens und den Einsatz einer universellen schritthaltenden Kontrolle vorsieht. Die Vorteile solcher Realisierung liegen in der leichteren Erweiterbarkeit, Konfigurierung und schnelleren Realisierung von komplexen Anwendungssystemen.

Verwendete Notation

Zeichen	Bedeutung
	<u>Koordinatensysteme</u>
$(X \ Y \ Z)^T$	Die Weltkoordinaten eines Punktes im 3-D Raum
$(X^w \ Y^w \ Z^w)^T$	Die Fahrbahnkoordinaten eines Punktes im 3-D Raum
$(X^k \ Y^k \ Z^k)^T$	Die Kamerakoordinaten eines Punktes im 3-D Raum
$(X^c \ Y^c \ Z^c)^T$	Die Eigenkoordinaten des Kamerafahrzeugs
$(x, y)^T$	Die Bildkoordinaten eines Punktes in Bildebene
F	Die Relation der Kamerabrennweite zu Pixelgröße
α	Die Rotation des Fahrbahnkoordinatensystems um die Y Achse
β	Die Rotation des Fahrbahnkoordinatensystems um die X Achse
FP	Der geometrische Fluchtpunkt
FOE	Der dynamische Fluchtpunkt
	<u>Konturveränderung in kurzer Bildfolge</u>
τ	Das Zeitintervall $t_{k+1} - t_k$ zwischen zwei aufeinanderfolgenden Bildern
\tilde{v}	Der kontinuierliche Konturveränderungsvektor in der Bildebene
v	Der diskrete Konturveränderungsvektor
v_z	Die relative Konturlängenveränderung
θ_i	Gewichtung der Konturattribute im i -ten Bild einer kurzen Bildfolge
	<u>Modellbasierte Objekterkennung</u>
$s = (x^d, \xi)$	Der Zustandsvektor eines 3-D Objektes
s^d	Der Trajektorienvektor eines 3-D Objektes
Θ	Der Richtungswinkel im Trajektorienvektor
V	Der Betrag der translatorischen Geschwindigkeit
ω	Die Winkelgeschwindigkeit
ξ	Der Formvektor eines 3-D Objektes
$B\tau, \kappa_1, \dots, \kappa_n$	Elemente des Formvektors
$B\tau$	Die maximale Breite einer Objekthypothese
m	Der Meßvektor
s_S	Der Zustandsvektor eines unbewegten Objektes
s_B	Der Zustandsvektor eines bewegten Objektes
\dot{p}	Die erste Ableitung bezüglich der Zeit einer Variable p
	<u>Rekursive Zustandsschätzung</u>
$s^*(k)$	Die Vorhersage des Zustandsvektors zum Zeitpunkt k
$s^+(k)$	Die Schätzung des Zustandsvektors zum Zeitpunkt k
F	Die Jacobimatrix der Zustandsübergangsfunktion
H	Die Jacobimatrix der Projektionsfunktion
I	Die Einheitsmatrix
K	Die Kalmanverstärkungsmatrix
$P^*(k)$	Die Vorhersage der Schätzwert-Kovarianzmatrix zum Zeitpunkt k
$P^+(k)$	Die Kovarianzmatrix des Schätzwertes zum Zeitpunkt k
E^+	Die Varianz eines Schätzwertes eines Skalars
E^*	Die Varianz einer Vorhersage eines Skalars
Q	Die Kovarianzmatrix des Prozeßrauschens
R	Die Kovarianzmatrix des Meßfehlers

Inhaltsverzeichnis

1	Einleitung	1
1.1	Das Problem: Objekterkennung in Bildfolgen	2
1.2	Bewegte Objekte – Stand der Technik	7
1.2.1	Pixelbasierte Verschiebung	7
1.2.2	Verschiebungsbasierte Bildsegmentierung	8
1.2.3	Merkmals- und 2-D Objektverfolgung im Bild	8
1.2.4	3-D Struktur und Bewegung aus Ansichten	9
1.2.5	Modellbasierte 3-D Objektverfolgung	10
1.2.6	Die Erkennung von mehreren Objekten	11
1.2.7	Schlußfolgerung: Bewegte Objekte	11
1.3	Systemansätze – Stand der Technik	11
1.4	Ziele und Lösungsansatz	16
1.4.1	Die Ziele	16
1.4.2	Der Lösungsansatz	17
1.5	Beitrag der Arbeit	18
1.5.1	Systemintegration und Konfigurierung	18
1.5.2	Neuentwicklungen für Teilprobleme	19
1.6	Gliederung der Arbeit	21
2	Der adaptive Ansatz zur Objekterkennung	23
2.1	Die funktionelle Systemstruktur	23
2.1.1	Das 2-D Modul	23
2.1.2	Das 2.5-D Modul	26
2.1.3	Das 3-D Modul	27
2.1.4	Der Analysefluß	28
2.1.5	Zur modellbasierten 3-D Bewegungserfassung	29
2.2	Das Modell des adaptiven Ansatzes	30
2.2.1	Ein dynamisches System	30
2.2.2	Die Initialisierung	34
2.2.3	Die rekursive Schätzung	34
2.2.4	Die Messung	37
2.2.5	Die Objektauswahl	38
2.3	Anwendung des adaptiven Ansatzes	39
2.3.1	Adaptives Differenzbild bei Eigenbewegung	39
2.3.2	Konturverfolgung	40
2.3.3	Adaptive Fluchtpunkterkennung	40
2.3.4	Adaptive Fahrbahnerkennung	41
2.3.5	Adaptive 3-D Objekterkennung	41
2.4	Systemkonfigurierung bei stationärer Kamera	43

3	Bildsegmentierung und Verschiebungsdetektion	45
3.1	Ikonik	45
3.1.1	Bildvorverarbeitung	45
3.1.2	Kantendetektor	47
3.1.3	Pixelbasierte Verschiebung	48
3.2	Linien- und Regionen-Detektion	50
3.2.1	Linienfinder	50
3.2.2	Regionenfinder	51
3.3	Konturverfolgung im Bild	52
3.3.1	Konturdetektion im Einzelbild	52
3.3.2	Stabile Messung von Konturen	52
3.3.3	Die dynamischen Konturmerkmale	53
3.3.4	Gewichtete Mittelung in kurzer Bildfolge	54
3.3.5	Diskretisierungsfehler der Konturverfolgung	55
4	Fahrbahnerkennung und Eigenzustandsverfolgung	61
4.1	Fluchtpunkte im Bild	61
4.1.1	Geometrischer Fluchtpunkt FP	61
4.1.2	Der dynamische Fluchtpunkt FOE	64
4.2	Konturklassifikation im Bild	65
4.2.1	Geometriebasierte Konturklassifikation	65
4.2.2	Verschiebungsbasierte Konturklassifikation	65
4.2.3	2.5-D Konturattribute	65
4.3	Dynamische Orientierung im 3-D Raum	69
4.3.1	Die Koordinatensysteme im 3-D Raum	69
4.3.2	Die Rotationswinkel	72
4.3.3	Die zeitlichen Trajektorien auf Ebene	72
4.4	Die Fahrbahnerkennung	72
4.4.1	Die Fahrbahnhypothesen	73
4.4.2	Detektion (Messung) der Parameter W, B	74
4.4.3	Erste Modifikation und Fahrbahnauswahl	75
4.4.4	Indirekte Parameter - Dynamik und Fahrbahnkrümmung	78
4.5	Der Eigenzustand	79
5	Modellbasierte Objekterkennung	81
5.1	Das Objektmodell	81
5.1.1	Die Trajektorien	82
5.1.2	Die Objektformen	83
5.1.3	Spezialisierte Formen	84
5.1.4	Aktivierung des Modells	85
5.2	Generierung von Objekthypothesen	85
5.2.1	Gruppierung von Segmenten	85
5.2.2	Initialisierung des Objektzustandes	85
5.3	Dynamische Übergangsfunktionen	86
5.3.1	Für unbewegte Fahrbahnstreifen	87
5.3.2	Für bewegte Fahrzeuge	87
5.3.3	Für Hindernisse/Umgebungsobjekte	89
5.4	Modifikation der Objekthypothese	89
5.5	Abbildungsfunktionen und Messungen	91

5.5.1	Die 3–D Messung	91
5.5.2	Die 2–D Messung	93
5.5.3	Der Ablauf der 2–D Messung	96
5.6	Objektauswahl	100
5.6.1	Der Konsistenztest	100
5.6.2	Der Erkennungstest	100
6	Eine wissensbasierte Realisierung	101
6.1	Die wissensbasierte Systemstruktur	101
6.1.1	Die Wissensbasis	101
6.1.2	Die Kontrolle	103
6.2	Die optimale Modell–zur–Kontrolle Relation	107
6.2.1	Der Analyseaufwand	107
6.2.2	Anforderungen an das Modell	107
6.2.3	Anforderungen an die Basiskontrolle	110
6.2.4	Anforderungen an die Bildfolgenkontrolle	111
6.3	Das Modell	114
6.3.1	Die Struktur des Modellnetzes	114
6.3.2	Die Zyklenfreiheit des Modells	115
6.4	Die schritthaltende Graphensuche	117
6.4.1	Das Verfahren	117
6.4.2	Die Suchbaumübernahme und Auffrischung	117
6.4.3	Das Bewertungsschema im Suchbaum	118
7	Testergebnisse	121
7.1	Verschiebungsschätzung im Bild	121
7.2	Konturverfolgung	125
7.3	Fahrbahnerkennung	128
7.3.1	Fluchtpunkt FP	128
7.3.2	Fahrbahnparameter	132
7.3.3	Fahrbahnkrümmung	135
7.4	Verfolgung von Fahrbahnstreifenobjekten	138
7.5	Verfolgung von bewegten Objekten	140
7.5.1	Qualitative Auswertung	140
7.5.2	Quantitative Auswertung	142
7.6	Rechenzeiten und Konfigurierung	148
8	Zusammenfassung und Diskussion	147
	Literaturverzeichnis	150
9	Anhang	157
9.1	Die dynamische Eigentransformation <i>DET</i>	158
9.2	Die Übergangsfunktion für unbewegte Objekte	160
9.3	Die Übergangsfunktion für bewegte Objekte	162
9.4	Die Abbildungsfunktion	164
9.5	Die Abstimmparameter	171

Abbildungsverzeichnis

1.1.1	Bildfolgen aus einer stationären Kamera	2
1.1.2	Bildfolge aus einer bewegten Kamera	2
1.1.3	Die Kameraneigung zur Fahrbahnebene	4
1.4.1	Das System zur adaptiven Objekterkennung auf Fahrbahnebene	17
2.1.1	Die Struktur des Erkennungssystems	24
2.1.2	Beispiele von Zwischenergebnissen	25
2.1.3	Der Analysefluß im System	28
2.1.4	Probleme der Verschiebungsschätzung bei Eigenbewegung	30
2.1.5	Das Aperturproblem	31
2.2.1	Die Struktur des adaptiven Ansatzes	32
2.2.2	Die rekursive Zustandsschätzung	35
2.2.3	Der Kalman Filter Algorithmus	36
2.2.4	Die Bedingungen für Verfolgung und Erkennung	38
2.2.5	Die Phasen einer Hypothese	39
2.4.1	Verfolgung eines bewegten Objektes mit stationärer Kamera	44
2.4.2	Analyse des Verkehrsaufkommens auf der Autobahn	45
2.4.3	Analyse des Verkehrsaufkommens auf einer Straßenkreuzung	45
3.1.1	Das 2-D Modul	46
3.1.2	Die histogrammbasierte Streckung	47
3.1.3	Ein Beispiel der Kantendetektion	47
3.1.4	Das Schema der Kantendetektion	49
3.1.5	Bewegungsschätzung im Bild bei Perspektive und Eigenbewegung	50
3.2.1	Detektion der Liniensegmente	51
3.2.2	Die Regionendetektion	51
3.3.1	Diskrete Kontur und ihre Merkmale	52
3.3.2	Dynamische Attribute einer Kontur	53
3.3.3	Diskretisierungsfehler von Geradensegment und Kreis	56
4.1.1	Das 2.5-D Modul	62
4.1.2	Detektion des Fluchtpunktes FP	63
4.1.3	Detektion des Fluchtpunktes FOE	64
4.2.1	Geometriebasierte Konturklassifikation	66
4.2.2	Klassifikation der Fahrbahnkonturen	66
4.2.3	Verschiebungsbasierte Konturklassifikation	67
4.2.4	Virtuelle Objektkante	68
4.3.1	Die Abbildungsgeometrie	70
4.3.2	Die Eigentrajektorie auf Fahrbahnebene	73
4.4.1	Ausgewählte Liniensegmente auf Fahrbahnebene	76

4.4.2 Häufungshistogramm für rückprojizierte Fahrbahnmarkierung	77
4.4.3 Die Gitter der Fahrbahnthesen	77
4.4.4 Die stabilisierten Fahrbahnparameter	78
5.1.1 Das Modul zur modellbasierten Objekterkennung	82
5.1.2 Drei Klassen von modellierten Objekten	83
5.1.3 Drei Spezialisierungsstufen des PKW-Modells	84
5.3.1 Die Objekttrajektorie bei Eigenbewegung	88
5.5.1 Der Objekt-zu-Objekt-Abgleich	92
5.5.2 Die Bedeutung der 2-D Messung	93
5.5.3 Die kantenbasierte Objektverfolgung	96
5.5.4 Konvergieren der Modellalternativen	99
6.1.1 Wissensbasierte Realisierung des Systems	102
6.1.2 Die "top-down" und "bottom-up" Inferenzen	103
6.1.3 Der Zustandsraum und der Entscheidungsbaum	105
6.1.4 Der A (A*) Algorithmus zur optimalen Graphensuche	106
6.2.1 Zwei syntaktisch unterschiedliche Modelle	108
6.2.2 Zwei semantisch unterschiedliche Modelle	109
6.2.3 Gleiche Instantiierungen von gleichen Knoteneinträgen	111
6.2.4 Globale Inferenzverwaltung	112
6.2.5 Beispiel von drei Objekten mit je 3 konkurrierenden Instanzen	112
6.2.6 Die suchaufwendige Analyse	113
6.2.7 Die benötigte Analyse	113
6.3.1 Das semantische Netzwerk des Modells	116
6.4.1 Das Verfahren der schritthaltenden Graphensuche	117
6.4.2 Gleichzeitiges Neuinitialisieren und Auffrischen	118
6.4.3 Globale Inferenzverweise und späteres Auffrischen	118
7.1.1 Einzelbilder aus sechs Testfolgen	123
7.1.2 Einfluss des Kameranicken auf Verschiebungsdetektion	124
7.1.3 Blockbasierter optischer Fluß	124
7.1.4 Gradientenbasierter optischer Fluß	124
7.1.5 Adaptive Bewegungsdetektion	125
7.3.1 Verfolgung des Fluchtpunktes in erster Bildfolge	130
7.3.2 Verfolgung des Fluchtpunktes in dritter Bildfolge	130
7.3.3 Die Varianz des FP-Schätzwertes	131
7.3.4 Stabilisierung der Fahrbahnparameter	133
7.3.5 Fahrbahnverfolgung in erster Bildfolge	134
7.3.6 Fahrbahnverfolgung in dritter Bildfolge	134
7.3.7 Eigengeschwindigkeit und Lenkeinschlag	135
7.3.8 Kamerarichtung und Eigenwinkelgeschwindigkeit	135
7.3.9 Die Fahrbahnkrümmung bei linearer Fahrbahn	136
7.3.10 Die Fahrbahnkrümmung bei Kurvenfahrt	137
7.4.1 Eine synthetische Bildfolge	138
7.5.1 Ein Beispiel der generierten Autohypothesen	140
7.5.2 Tiefenlagen für eine PKW- und LKW-Hypothese	142
7.5.3 Tiefenlageschätzung von zwei Autohypothesen	143
7.5.4 Parameter PX für eine PKW- und LKW-Hypothese	143
7.5.5 Stabilisierung des Richtungswinkels einer PKW- und LKW-Hypothese	144

7.5.6 Die Eigenfahrzeugrichtung 144

7.5.7 Translatorische Geschwindigkeit einer PKW- und LKW-Hypothese 145

7.5.8 Translatorische Geschwindigkeiten und Eigengeschwindigkeit 146

7.5.9 Rotationelle Geschwindigkeit der PKW- und LKW-Hypothese 146

7.5.10 Eigenwinkelgeschwindigkeit 147

7.5.11 Die Messung und Stabilisierung der Breite 147

Tabellenverzeichnis

1.1	Datenkomplexität zur Klassifikation der Objekterkennung	3
1.2	Kriterien zur Klassifikation von Analysestrategien	14
7.1	Rechenzeiten von pixelbasierten Detektoren	123
7.2	Einfluß der Schwellwerte auf die Konturdetektion	126
7.3	Stabilität einer Verschiebungsschätzung für einen Mittelstreifen	126
7.4	Quantitative Beurteilung einer Verschiebungsschätzung	127
7.5	Verfolgung des dynamischen Fluchtpunktes FOE	128
7.6	Mittlerer Fehler und Fehlervarianz der FP-Bestimmung	129
7.7	Fahrbahndetektion in 5 Bildfolgen	131
7.8	Genauigkeit der Fahrbahndetektion	133
7.9	Verfolgung eines unbewegten Objektes	139
7.10	Genauigkeit der Detektion und Verfolgung eines unbewegten Objektes	139
7.11	Qualitative Auswertung der Objektverfolgung	141
7.12	Rechenzeiten und Konfigurierung bei Eigenbewegung	148
7.13	Rechenzeiten und Konfigurierung ohne Eigenbewegung	149

Kapitel 1

Einleitung

Ausgangspunkt für die automatische maschinelle Bilddeutung (Bildauswertung) sind meistens zweidimensionale monokulare Bilder von betrachteten Szenen, die entweder als *Einzelbilder* ohne zeitlichen Bezug oder zu einer *Bildfolge* zusammengefaßt dem Analysesystem als Eigabedaten dienen. Bei der Auswertung jedes *einzelnen* Bildes ergibt sich die wesentliche Schwierigkeit, daß aufgrund der perspektivischen Projektion der dreidimensionalen Welt auf die Bildebene eine Dimension verlorenggeht. Zur Rekonstruktion dieser Dimension können zwei Kameras angewendet werden – man bezeichnet das als *Stereosehen*. Es können aber auch zwei Bildaufnahmen einer Kamera zu unterschiedlichen Zeitpunkten vorliegen – wir sprechen dann von *Bewegungsstereo*. Voraussetzung für eine Lösung bei der zweiten Methode ist, daß sich die relative Lage zwischen Kamera und Umgebung während der beiden Aufnahmen ändert. Als eine Verallgemeinerung des Bewegungstereo kann eine längere *Bildfolge* angesehen werden, die mehr als zwei aufeinanderfolgende Bilder enthält.

Das generelle Schema für automatische Bildauswertung geht von drei hierarchischen Verarbeitungsebenen aus: *Ikonik* (ikonische Verarbeitung) und *Segmentierung* (*niedrige Ebene*), *Objekterkennung* (*mittlere Ebene*) und *kognitive Ebene* (*hohe Ebene*). Die Verarbeitung auf der niedrigen Ebene ist weitgehend unabhängig von dem Weltausschnitt, aus dem die abgebildeten Szenen stammen. Als Ergebnis dieser Verarbeitung liegt dann eine *initiale symbolische Bildbeschreibung* vor, welche aus den im Bild detektierten Segmenten und ihren Relationen zueinander besteht. Die mittlere Ebene beruht auf der Kenntnis der Modelle von potentiell vorkommenden Objekten. Mit den Objektmodellen ist erstmals *anwendungsabhängiges* Wissen über den betrachteten Weltausschnitt in die Analyse eingebracht. Das Ergebnis der Objekterkennung ist eine *quantitative Szenenbeschreibung*, die durch die detektierten Objekte und ihre geometrisch-topologisch-zeitliche Merkmale gegeben ist. Auf der *kognitiven* Ebene, die auch modellbasiert verläuft, wird eine *qualitative Szenenbeschreibung* in Form von Ereignissen und Szenarien erzeugt, sowie ein aktives Vorgehen des autonomen Systems geplant und gesteuert.

Die vorliegende Arbeit befaßt sich mit der *Erkennung von bewegten Objekten* in einer *monokularen Bildfolge*, die mit einem *passiven* Sensor (einer Kamera) aufgenommen wurde. Bedingt durch die Art des Sensors unterscheidet sich unsere Aufgabe von Aufgaben solcher Bildanalyssysteme, die andere Bildklassen analysieren – Stereobilder, Tiefenbilder. In der Arbeit werden Verfahren zur anwendungsunabhängigen Bewegungserfassung in Bildebene und zur modellbasierten Erkennung von bewegten Objekten entwickelt, welche innerhalb von zwei Verarbeitungsebenen – der niedrigen Ebene und der Ebene der Objekterkennung – eines Bildauswertungssystems anzusiedeln sind. In folgenden Abschnitten dieses Kapitels wird die betreffende Problematik analysiert, eine kurze Übersicht zum Stand der Technik gegeben, die Ziele und der entwickelte Lösungsansatz angesprochen und der wissenschaftliche Beitrag der Arbeit erläutert.

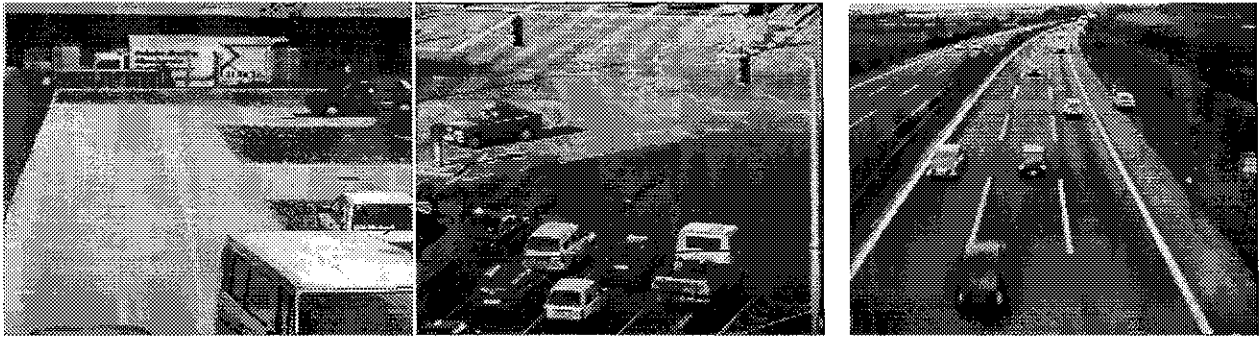


Bild 1.1.1: Ausschnitte aus drei Bildfolgen von Verkehrsszenen aufgenommen mit einer stationären Kamera: Parkplatzszenen (links), Kreuzung (mitte), Brückenszenen (rechts)

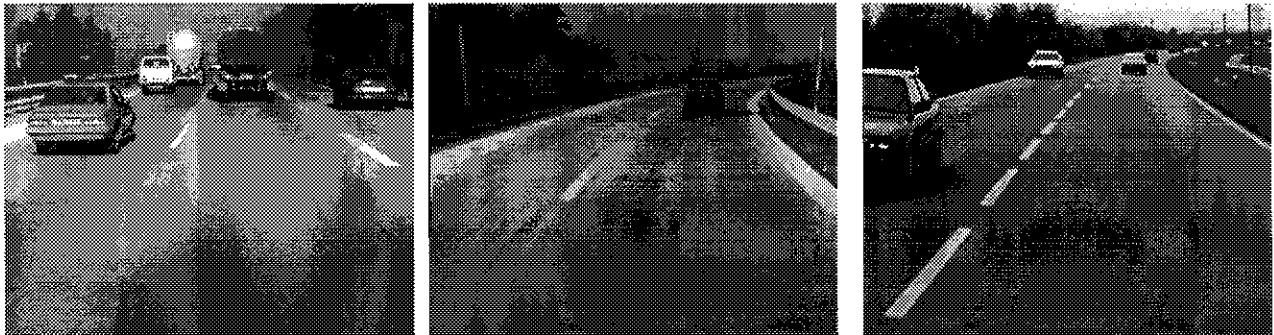


Bild 1.1.2: Ausschnitte aus Bildfolgen aufgenommen mit einer bewegten Kamera im Straßenverkehr

1.1 Das Problem: Objekterkennung in Bildfolgen

Grundsätzlich haben wir es in natürlichen Szenen mit 3-D Objekten zu tun, die aus verschiedenen Blickwinkeln betrachtet werden können und die einer unbekanntem Bewegung im Raum unterliegen können. Die Erkennungsaufgabe in diesem Fall in Szenen mit mehreren Objekten ist eines der noch nicht zufriedenstellend gelösten Probleme der automatischen Bildauswertung.

Das Problem der Objekterkennung in monokularen Bildern ist so umfassend, daß eine Klassifikation und Einschränkung der für diese Arbeit relevanten Problematik gleich am Anfang als notwendig erscheint. Als Kriterien zur Klassifikation von Ansätzen zur Objekterkennung dienen hier die in Tabelle 1.1 aufgeführten Datentypen.

Die Bilddaten und Anwendungen

Man unterscheidet gewöhnlich zwischen zwei Arten von monokularen Bildfolgen: (A) aus stationärer Kamera (bei bewegten Objekten) oder (B) aus bewegter Kamera (bei bewegten oder unbewegten Objekten). Diese Bildfolgen können entweder Laborszenen (künstliche Beleuchtung, vordefinierte Szenen) (Klasse k) oder natürliche Szenen (natürliche Beleuchtung, z.Teil unbekannte Szenenelemente) (Klasse n) darstellen.

Die Bilddaten		Relevanz
<i>Zeitfaktor</i>	Einzelbild	
	Bildfolge	Ja
<i>Szenentyp</i>	Laborszene	
	Natürliche Szene	Ja
<i>Bildtyp</i>	Grauwertbild	Ja
	Farbbild	
Das Kameramodell		
<i>Kamerabewegung</i>	starre, stationäre Kamera	Ja
	starre Kamera auf bewegtem Untergrund	Ja
	aktive Kamera	
<i>Projektion</i>	Parallelprojektion	
	Perspektive	Ja
<i>Objektauflösung im Bild</i>	klein – bis ca. 20 x 20 Pixel	
	mittelgroß – bis ca. 120 x 120 Pixel	Ja
	groß – über 120 x 120 Pixel	
<i>Blickwinkel (Kamerahöhe)</i>	kleiner Winkel (bis 2.0 m über Fahrbahn)	Ja
	mittelgroßer Winkel (2.0-5.0 m über Fahrbahn)	Ja
	gerader Winkel (z.B. Flugaufnahme)	
Segmentierung und Verschiebung im Bild		
<i>Segmenttyp</i>	Kanten	Ja
	Regionen	Ja
<i>Verschiebung im Bild</i>	pixelbasierte Verschiebung	Ja
	Merkmalsverfolgung	Ja
	2-D Objektverfolgung	
<i>Segmentkomplexität</i>	lineares Segment	Ja
	gekrümmtes Segment	
<i>Segmentattribute</i>	geometrisch-topologische	Ja
	zeitliche Veränderung	Ja
Objekterkennung		
<i>Objekttyp</i>	2-D Objekt	
	Einkörperobjekte	Ja
	Mehrkörperobjekte	Ja
<i>Objektfläche</i>	planare Flächen	Ja
	gekrümmte Flächen	
<i>Objektbewegung</i>	stationäre Objekte	Ja
	bewegte Objekte	Ja
<i>Objektanzahl</i>	Einobjektszene	
	Mehrobjektszene	Ja

Tabelle 1.1: Datenbasierte Kriterien zur Klassifikation der Ansätze zur 3-D Objekterkennung in monokularen Bildern

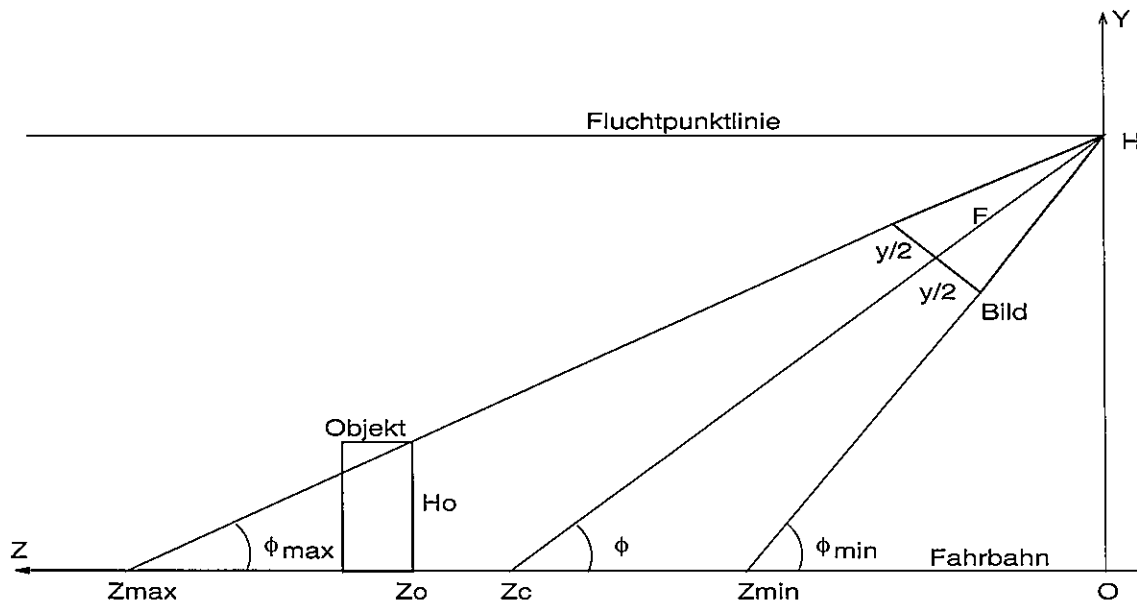


Bild 1.1.3: Die Kameraneigung zur Fahrbahnebene und die damit verbundenen Projektionsbedingungen und Objektgrößen im Bild

In der Umgebung (Ak) (stationäre Kamera, künstliche Beleuchtung) werden die Bildfolgen gewöhnlich eingesetzt zur Gewinnung der 3-D Struktur eines einzelnen Objektes aus seiner Bewegung. Dies setzt eine relativ große Bildprojektion des Objektes voraus (mindestens 100 x 100 Pixel). Bei der Klasse (An) gelten die Bildfolgen als Ausgangspunkt für visuelle Sicherheitssysteme bei Einfahrten, sowie für Verkehrsflußanalyse auf Kreuzungen und Autobahnen ([BA89], [FRB94], [Kil92]) usw. Ziel dieser Analyse ist die Detektion, Klassifikation oder Erkennung von bewegten Objekten in Verkehrsszenen.

Die Bildfolgen aus einer bewegten Kamera sind Ausgangspunkt für automatische Bildauswertung im Bereich der Robotik (Klasse Bk) (z.B. aktive Kamera), zur automatischen Andockung von Satelliten an Raumstationen ([DG88], [Wue88], [Gen92]), zur Fahrerunterstützung im Straßenverkehr ([Mas92], [Fra89], [Wet95]) usw.

Als Anwendungsfall der theoretischen Ausführungen dient in der vorliegenden Arbeit die maschinelle Bildauswertung von Verkehrsszenen, wie sie entweder zur Analyse des Verkehrsaufkommens (bei stationärer Kamera) (Bild 1.1.1) oder zur Steuerung von autonomen Fahrzeugen (bei bewegter Kamera) (Bild 1.1.2) verwendet werden kann.

Das Kameramodell

In dieser Arbeit wird grundsätzlich von einer starr montierten Kamera ausgegangen, die auf einem sich bewegendem Fahrzeug postiert ist oder auch stationär sein kann. Dadurch ist eine Eigenbewegung der Kamera möglich, die zum Teil unbekannt sein kann (Nickbewegung, verursacht durch Fahrbahnwellen oder Wind).

Die Voraussetzungen der Analyse werden bedingt durch folgende Kameraparameter: die Höhe der Anfertigung der Kamera über der Fahrbahn (H), die Neigung zur Fahrbahn (Φ), die Kamerabrennweite (F) und die Bildauflösung (Bildhöhe y) (Bild 1.1.3). Man muß von einer relativ geringen Höhe H ausgehen, zum Beispiel kleiner als 1.70 m bei Anbringung der Kamera in Fahrzeugen oder kleiner als 5 m bei Positionierung der Kamera auf Brücken- oder neben Straßenkreuzungen. Der aktuelle Standard auf dem Markt der Bildsensoren bedeutet eine Bildauflösung von $768h \times 576v$ [Pixel²] auf einem lichtempfindlichen Bereich von $4.9h \times$

3.7v [mm²]. Es gibt zwar bereits einzelne Kameras mit einer Pixelgröße von $5.1\mu\text{m} \times 5.1\mu\text{m}$ bei einer Bildauflösung von $5k \times 5k[\text{Pixel}^2]$, aber für eine breite technische Anwendung ist der Preis noch unakzeptabel hoch. Die zukünftigen technologischen Grenzen sollen bei einer Pixelgröße von $2\text{--}3\mu\text{m}$ und einer Bildauflösung von bis zu $20000 \times 20000[\text{Pixel}^2]$ liegen ([SVS95]).

Die verwendeten Objektive besitzen eine Brennweite von mindestens 20 mm . Die Relation F zwischen Kamerabrennweite f und Bildhöhe y beträgt:

in praktisch verfügbaren Kameras mit $y = 4.9[\text{mm}]$ (unter Annahme, daß die Kamera um 90° Grad um eigene Mittelachse gedreht wird) ist mindestens $F \simeq 4.082[y]$. Dies entspricht einem maximalem Öffnungswinkel der Kamera von 15.52° Grad.

in theoretisch verfügbaren Kameras mit $y = 5120[\text{Pixel}] \times 5.1[\mu\text{m}] = 26.112[\text{mm}]$ ist mindestens $F \simeq 0.766[y]$. Dies entspricht einem maximalem Öffnungswinkel der Kamera von 73.6° Grad.

Die Relation des Öffnungswinkels zur Höhe über der Fahrbahn ist entscheidend darüber, wie groß der ins Bild projizierte Bereich im Raum sein kann. Wenn wir davon ausgehen, das die Tiefe des zu analysierenden Bereiches bis zu $100[\text{m}]$ betragen soll bzw., daß der Kreuzungsbereich $20 \times 20[\text{m}^2]$ groß sein soll, dann ergeben sich folgende Beschränkungen des Neigungswinkels der Kamera (Bezeichnungen gemäß Bild 1.1.3):

1. Bei $H = 1.7[\text{m}]$, $H_o = 3.5[\text{m}]$ und $Z_o = 100[\text{m}]$ ist $\Phi_{max} \simeq -1.15[\text{Grad}]$. Damit ist $\Phi = 6.61[\text{Grad}]$ bei $F = 4.082[y]$, oder $\Phi = 35.6$ bei $F = 0.766[y]$.
2. Bei $H = 5.0[\text{m}]$, $H_o = 3.5[\text{m}]$ und $Z_o = 20 + Z_{min} = 25[\text{m}]$ ist $\Phi_{max} = 4.7[\text{Grad}]$. Der Öffnungswinkel muß der Tiefe Z_{min} angepaßt werden. Für $Z_{min} = 5[\text{m}]$ ist insgesamt $\Phi = 12.46[\text{Grad}]$ bei $F = 4.082[y]$ oder $\Phi = 41.5[\text{Grad}]$ bei $F = 0.766[y]$. Bei größeren Tiefen für Z_{min} muß aber der Öffnungswinkel kleiner gewählt werden, um den gewünschten Szenenbereich zu umfassen, so daß sich der Neigungswinkel verringert.

Aus der Tatsache eines begrenzten Neigungswinkels können zwei Schlüsse gezogen werden:

1. Wir haben es und werden es in absehbarer Zukunft mit Seitensichten von 3-D Objekten zu tun haben, insbesondere bei bewegter Kamera und bei Brückenansichten auf Autobahnen.
2. Die Projektionen der Objekte und ihre Bewegungen unterliegen einer *perspektivischen Verzerrung* in der Bildebene.

Entsprechend der Bildauflösung und Brennweite ergeben sich auch Grenzen für die projizierten Objektgrößen. Man kann von sichtbaren Fahrzeugsansichten von bis zu $2.5[\text{m}] \times 3.5[\text{m}]$ (bei Front- oder Heck-Ansicht) ausgehen. Bei $F = 768 \times 4.082 \simeq 3135[\text{Pixel}]$ beträgt die Projektion dieser Objekte im Bild:

- ca. $150 \times 210[\text{Pixel}^2]$ bei $Z = 50[\text{m}]$
- ca. $75 \times 105[\text{Pixel}^2]$ bei $Z = 100[\text{m}]$

In der Praxis arbeiten wir auf Halbbildern, da die in Zeilensprungtechnik aufgenommenen Bilder eine Unterabtastung erfordern. Dadurch sind die maximalen Größen im Bild nur halb so groß wie oben angegeben. Es wird erwartet, daß die überwiegende Anzahl von Objekten als *mittelgroß* im Bild erscheinen wird und daß nur die wesentlichen Objektmerkmale im Bild zu erkennen sein werden.

Segmentierung und Verschiebung im Bild

Zur Bildverarbeitung und Segmentierung stehen uns viele Methoden der Mustererkennung bereits zur Verfügung ([Nie81], [Nie90]) insbesondere auch die Methoden der Verschiebungsschätzung in der Bildebene ([AN88], [Die88]). Für die Segmentextraktion oder Segmentgruppierung kann eine Verschiebungsschätzung im Bild nützlich sein, falls die Objekte sich überwiegend parallel zur Bildfläche bewegen. Die Möglichkeit der Detektion von komplexen Segmenten ist direkt abhängig von der Größe der Objekte im Bild.

Erkennung von einzelnen 3-D Objekten

Wir gehen von drei generellen Ansätzen zur 3-D Objekterkennung in Einzelbildern aus, die je nach Komplexität der Szene und Objektformen erfolgreich eingesetzt werden können (für eine Übersicht siehe z.B. [BJ85]) :

- *prototypen-freie* Flächenrekonstruktion

Zur ersten Klasse werden hier die sogenannten Ansätze *shape-from-xxx* gezählt ([AB88], [IJ81], [Wit81]), sofern sie auf Einzelbildern arbeiten (d.h. die Methode *Struktur-aus-Bewegung*) zählen wir nicht dazu).

- *prototypen-freie* Rekonstruktion der Objekte

Im zweiten Fall gibt es für gewisse 3-D Körperklassen prozedurale Vorschriften, die zu einer Gruppierung von Regionen mit dazugehörigen Kanten und Vertices zu 3-D Körpern führen. Das ist eine rein datengetriebene Analyse in der *anwendungsunabhängiges* Wissen eingesetzt wird. Dieser Ansatz wurde bisher zur Erkennung von Polyedern in relativ schwach strukturierten Bildern (z.B. Strichzeichnungen) eingesetzt [Sug86].

- *modellbasierte* Objekterkennung

In Anwendungen, in denen man mit vielen kleinen und mehrdeutigen Bildelementen zu tun hat oder komplexe Objekte erkennen will, muß mit expliziten Objektprototypen gearbeitet werden. In diesem zweiten Ansatz wird versucht, die Objekte zu detektieren und zu lokalisieren, indem explizit definierte 2-D Projektionen von Modellelementen mit den im Bild gefundenen Merkmalen verglichen werden. Die Methode für die Erkennung einzelner Objekte ist gewöhnlich *Erkennung durch Lokalisierung*, d.h. Suche nach entsprechender Transformation der Bildpunkte in Modellpunkte ([Gri90]).

Im Falle der Erkennung von bewegten 3-D Objekten in Bildfolgen, den wir in dieser Arbeit verfolgen, können zwei generelle Ansätze unterschieden werden:

- Die prototypen-freie Rekonstruktion (der Ansatz *Struktur-aus-Bewegung*)

Es wird vorausgesetzt, daß einige aufeinanderfolgende Bilder signifikante Ansichten des projizierten Objektes enthalten und daß eine Korrespondenz von mehreren Merkmalen (Vertices oder Kanten) in diesen Bildern gefunden werden kann ([LH81], [TH84]). Neben diskreten Merkmalen kann auch ein dichter optischer Fluss als Ausgangspunkt einer 3-D Interpretation dienen ([WW88]). In beiden Fällen tragen diese Bildmessungen zur Lösung der nichtsingulären inversen 3-D Projektion des Objektes bei. Weitere vorgegebene Regularitätsbeschränkungen sind meistens notwendig, um robuste Verfahren zu entwickeln.

Eine vereinfachte Lösung wird möglich unter Annahme orthographischer Projektion ([Ull79], [HA94]) obwohl hier mindestens drei Bilder notwendig sind.

Hier kann man weiter zwischen der Gewinnung von globalen *Objekttiefekoordinaten* und der *3-D Objektstruktur* (relative Tiefe von Objektpunkten in Objektkoordinaten) unterscheiden ([Sub88]).

- Die modellbasierte Objektverfolgung (der *4-D Ansatz*)

Eine Integration von räumlichen und zeitlichen Aspekten der Bilderkennung. Mit Methoden der Regelungstechnik wird ein rückgekoppeltes System beschrieben [Dic88]). Die initialen 3-D Objektparameter werden in die Bildebene abgebildet und mit den im Bild detektierten Merkmalen verglichen. Entsprechend dem Ergebnis dieses Vergleiches werden die Systemwerte modifiziert. Eine Rekonstruktion für die ersten n Bilder kann zur Initialisierung des Objektzustandes genutzt werden.

Erkennung von mehreren Objekten

Die Szenenbeschreibung bedeutet eine Suche nach konsistenten Objekt-Konfigurationen, wobei die Eigenverdeckungen von Objekten und Verdeckungen zwischen Objekten analysiert werden müssen. Besonders jetzt ist eine modellbasierte Analyse gefragt und insbesondere der Einsatz eines 3-D Objektmodells wird erwartet. Dieses Vorgehen kann eine robuste Erkennung von Objektgrenzen bei teilweise verdeckten Objekten garantieren.

Schlußfolgerung: Problem

Die Entwicklung und der Einsatz von Verfahren zur Bewegungserkennung in monokularen Bildfolgen soll eines der wichtigsten Ziele dieser Arbeit sein. Die Integration dieser Verfahren mit klassischen Verfahren zur Objekterkennung in Einzelbildern in ein Gesamtsystem zur Bildfolgenanalyse von Verkehrsszenen ist eine wichtige Voraussetzung für praktische Tests des Ansatzes. Bevor die Ziele konkret formuliert werden können, sollen die wichtigsten hierbei bekannten alternativen Ansätze beschrieben werden. Deswegen widmen sich die anschließenden zwei Abschnitte dem Stand der Technik im Bereich der *bewegten Objekte* und des *Systemkonzeptes*.

1.2 Bewegte Objekte – Stand der Technik

In einem System zur Bildfolgenanalyse kann die Bewegungsinformation für unterschiedliche Datenstrukturen bestimmt werden und in unterschiedlicher Form vorliegen: als Verschiebung im Bild von einzelnen Pixeln und Pixelgruppen, als 2-D Bewegung von Bildsegmenten und 2-D Objekten oder als 3-D Bewegung von 3-D Objekten.

1.2.1 Pixelbasierte Verschiebung

Differenzbild

Die einfache Methode zur Bewegungsdetektion erzeugt sogenannte Differenzbilder. Ein Differenzbild wird im allgemeinen aus zwei aufeinanderfolgenden Bildern einer Bildfolge berechnet und enthält Informationen darüber, in welchen Bildbereichen Bewegung zwischen den Einzelbildern stattgefunden hat ([VJ90]).

Adaptives Differenzbild

Ein adaptive Form des Differenzbildes wurde in [KvB90] vorgeschlagen zur Detektion des stationären Hintergrundes einer Szene. Jeder Grauwert wird über die Zeit stabilisiert. Das adaptive

Differenzbild ist das Ergebnis der pixelbasierten Differenzen zwischen dem neuen und dem stabilisierten Bild.

Optischer Fluß

Eine Übersicht von gängigen pixelbasierten Verfahren zur quantitativen Bestimmung der Verschiebung im Bild (sogennater *optischer Fluß*) findet sich in [BFB94]. Es folgt eine Zusammenfassung der Klassen von diesen Verfahren:

- Gradientenbasierter optischer Fluß

Dieses Verfahren wurde vorgeschlagen durch Horn & Schunck und seine letzte Fassung findet sich in [Sch86]. Es gab seitdem eine Vielzahl von Modifikationen. Wichtige Beiträge sind der *Monotonieoperator* von Enkelman et al. [EKNZ88] und die Richtungsforderung von Nagel [Nag89].

- Blockabgleich, Regionenabgleich

Neben einem einfachen Abgleich zwischen je zwei Pixelblöcken im aktuellen und vorherigen Bild, von konstanter Lage und Größe, wurden auch mehr komplexe Verfahren vorgeschlagen: z.B. mehrstufiger Regionenabgleich auf der Basis einer Laplacian Pyramide (ein grob-zu-fein Abgleich) [Ana89], hierarchischer Blockabgleich [Kir89], [Kir93].

- Detektion im Frequenzbereich

Die Detektion erfolgt durch Minimierung der Ausgangsenergie von durch Geschwindigkeiten getunten Filtern [Hee88] oder durch Untersuchung der Phasenverschiebung im Ausgangssignal von Breitpaßfiltern [FJ90].

- Detektion im Volumenbild

Die Bewegung eines 2-D Merkmals wird als diskrete Fläche im 3-D Bild-Zeit-Raum interpretiert und mit Hilfe von Flächenmerkmalen detektiert ([LJ89]).

1.2.2 Verschiebungsbasierte Bildsegmentierung

Die Verschiebungsinformation kann zur Segmentierung des Bildes in bewegte und unbewegte Regionen oder andere Bildmerkmale führen. Diese Segmentierung wird relativ oft für Verkehrsszenenanalyse verwendet, da hier die projizierten Objekte relativ klein sind und eine detaillierte Erkennung nicht möglich ist (z.B. [BK94], [IRP94], [BF93]). Diese Ansätze sind insbesondere geeignet für Einzelobjekte, die vor einem homogenen Hintergrund fahren.

1.2.3 Merkmals- und 2-D Objektverfolgung im Bild

Durch die Verfolgung eines Merkmals im Bild oder eines 2-D Objektes kann seine 2-D Bewegung bestimmt werden. Die Verfolgung von Bildmerkmalen ist anwendungsunabhängig, d.h. sie greift lediglich auf die im Bild selbst vorhandene Information zurück. Bei der modellbasierten Objektverfolgung steht zusätzlich Hintergrundwissen (die 2-D Form eine Objektansicht) über die zu verfolgenden Objekte zur Verfügung.

Pixelblockverfolgung

Sei ein Pixelblock gegeben, daß ein 2-D Referenzmuster enthält. Die Verfolgung dieses Pixelblockes in einer Bildfolge kann als ein *Korrelationsproblem* ([ERTM94]) aufgefaßt werden,

daß gewöhnlich mit einem Korrelationsverfahren gelöst wird. Im Unterschied zum blockbasierten optischen Fluß wird die Verschiebung nicht für starre Pixelblöcke im Bild bestimmt, sondern die Lage eines Pixelmusters im Bild wird tatsächlich kontinuierlich verfolgt.

Kantenverfolgung

Ein Verfahren zur *Kantenverfolgung* (z.B. [DF90]) kann wie folgt zusammengefaßt werden: Es wird ein dynamisches Modell der Kantenbewegung entworfen. Die Zuordnung einer Kante zur Kante im nächsten Bild findet innerhalb eines Suchgebietes statt, um die erwartete Position herum. Es wird ein Abstandsmaß verwendet um aus der Menge der konkurrierenden Kantenpartner den Besten auszuwählen. Falls ein Partner gefunden ist, wird der neue Zustand der Kante berechnet. Als Abstandsmaß zur Korrespondenz von Bildsegmenten können statistische Maße eingesetzt werden aber die hohe Anzahl der möglichen Tests erfordert in der Praxis den Einsatz von heuristischen Beschränkungen ([Zha94]).

Verfolgung von Punkten und Konturen

Falls markante Punkte im Bild detektiert werden können und die Objektbewegung annähernd parallel zur Bildfläche verläuft empfiehlt sich die Verfolgung von 2-D Trajektorien dieser Punkte ([SS90]).

Die verfolgten Punkte können zu einer Bildkontur zusammengefaßt werden. Diese Hypothese kann sich während der Verfolgung elastisch verändern und verformen, was als *aktive Konturen* bezeichnet wird ([KWT88], [BCZ93]).

In [BK93] werden sogenannte lokale Spiegel (Symmetrieachsen) als Merkmale von Objektprimitiven herangezogen. Durch die Verwendung einfacher Formen, wie Quadrat und Kreisscheibe, wird versucht Fahrzeugdetails in Bildfolgen zu finden und zu verfolgen. Um die für dieses Verfahren benötigte Rechenzeit zu reduzieren, wurde auf Differenzbilder zurückgegriffen. Die Suche nach lokalen Spiegel-Symmetrieachsen wird auf die in Differenzbildern als bewegt markierte Bereiche reduziert.

Modellbasierte 2-D Objektverfolgung

In Anwendungen mit autonomen Fahrzeugen wird von weit entfernten Fahrzeugen ausgegangen, deren Heck- oder Frontansicht allein sichtbar ist. Das führt zu Ansätzen in denen ein generisches 2-D Kantenmodell dieser Fahrzeugansicht mit den Bilddaten abgeglichen wird ([RG94], [SNv92]). Dabei ist das primäre Ziel eine stabile Detektion und Verfolgung der Bildlage des Objektes und nicht die Erkenntnisse über seine Tiefe oder Geschwindigkeit.

1.2.4 3-D Struktur und Bewegung aus Ansichten

Für die Analyse von Verkehrsszenen in Bildfolgen wurde versucht, kurze Trajektorien von mehreren markanten Punkten eines Fahrzeuges als Eingabedaten zu nehmen für die Bestimmung der 3-D Struktur von einfachen Körpermodellen ([DN82]) oder zur initialen Schätzung der Tiefe und Bewegung des Fahrzeuges [TSB94]. Die Ergebnisse zeigen, daß diese Ansätze nur bei bestimmten Beschränkungen erfolgreich sein können. Entweder ist das zur erkennende Objekt als relativ großes Bildobjekt gegeben, das nicht fremdverdeckt ist (wie im ersten Ansatz), oder die Punkttrajektorien werden manuell bestimmt (wie im zweiten Fall).

1.2.5 Modellbasierte 3-D Objektverfolgung

Die Verfahren zur modellbasierten 3-D Objektverfolgung wurden anfangs für Anwendungen in der Luftfahrt und Raumfahrt entwickelt. Der Ansatz wurde später eingesetzt auch für Anwendungen im Bereich der autonomen Fahrzeuge im Straßenverkehr oder mobilen Roboter in technischer Umgebung. Er eignet sich insbesondere zur Verfolgung von einzelnen bewegten 3-D Objekten im Raum, deren genaue Form und Maße dem Verfolgungssystem bekannt sind.

Wünsche

Ein modellbasiertes Verfahren zur automatischen Satellitenandockung an Raumstationen wird in [Wue88] vorgestellt. Die Bildmessung enthält nur wenige signifikante Punkte der Raumstation, die aber eine genaue Erfassung der Station erlauben. Die mögliche Rotationsbewegung ist begrenzt auf eine Fläche und die Kamera bewegt sich nur relativ zur Raumstation (dies entspricht dem Fall einer stationären Kamera bei bewegtem Objekt oder umgekehrt ein stationäres Objekt bei sich bewegender Kamera).

Gennery

In [Gen92] wird dasselbe Anwendungsfeld wie bei Wünsche in Betracht gezogen. Es wird ein Verfahren zur Verfolgung von bekannten dreidimensionalen Objekten beschrieben, welche sich mit sechs Freiheitsgraden im Raum bewegen können. Das Objekt kann frei rotieren und die Kamera ist unbewegt. Die Bildmessung besteht entweder aus Punkten oder aus Kanten auf planaren Flächen. Das Objektmodell enthält dreidimensionale Eigenschaften – zumeist sind dies 3-D Kanten aber auch markante Modellpunkte sind gegeben. Es wird eine beliebige Anzahl von Kameras eingesetzt, welche alle stationär sind. Die Verwendung von mehreren Kameras bringt eine erhöhte Robustheit mit sich, was insbesondere dann wichtig ist, wenn Hintergrundmuster und Beleuchtung die Aufnahme stören. Das vorgestellte Verfahren ist in der Lage eine erhebliche Anzahl von fehlenden oder falsch detektierten Merkmalen zu verkraften. Zum einen liegt dies an der Redundanz der verschiedenen Kameraperspektiven, zum anderen daran, daß nach Merkmalen nur in der Nähe ihrer vorhergesagten Position gesucht wird.

Koller

In [Kol92] und [KDN93] wird ein Ansatz zur Objektverfolgung von Fahrzeugen mittels einer stationären Kamera beschrieben. Es wird ein (durch 12 Längenangaben) parametrisiertes Modell für Fahrzeuge verwendet. Aus diesem können dann durch Ändern der jeweiligen Längenparameter verschiedene Typen von Fahrzeugen erzeugt werden. Es werden 5 Beispiele für die häufigsten Fahrzeugtypen angegeben. Prinzipiell können mit dieser Vorgehensweise die meisten Fahrzeugformen approximiert werden. Ein sich innerhalb des Bildes kohärent bewegendes Gebiet dient unter folgenden Annahmen als erste Grobschätzung für eine Objekthypothese: die Bewegung wird von einer Objekthypothese verursacht, das hypothetische Objekt bewegt sich vorwärts, die Hauptachse des Objekts wird als parallel zur Bewegung angesehen. Aus der Objekthypothese werden Modellkanten ins Bild projiziert und die sichtbaren Kanten ermittelt. Die im Bild gefundenen Liniensegmente werden auf Übereinstimmung mit diesen erwarteten Modellkanten untersucht. Um fehlerhafte Übereinstimmungen mit Liniensegmenten, die sich aus dem Schattenwurf des Objektes ergeben, zu vermeiden, wird das Hintergrundwissen um zusätzliche Informationen über die aktuelle Beleuchtung erweitert.

1.2.6 Die Erkennung von mehreren Objekten

Bei der Verfolgung von diskreten Bildmerkmalen, 2-D und 3-D Objekten stehen wir generell vor der Aufgabe das Problem der *Zuordnung* (oder der *Korrespondenz*) zu lösen. Dieses Problem ist von exponentieller Komplexität, d.h. je mehr Merkmale oder Objekthypothesen desto schneller wächst der Aufwand der Lösung. Eine frühe Arbeit über das Entscheidungsproblem bei der Verfolgung von mehreren Objekten ist ([Rei94]). Eine Übersicht von statistischen Verfahren zur Entscheidung des Korrespondenzproblems in Bildfolgen wird in [Cox93] gegeben. Es zeigt sich, daß optimale Verfahren, welche die volle Statistik des Problemsfeldes überprüfen, einen erheblichen Aufwand besitzen, der ihren Einsatz in Anwendungssystemen fraglich macht. Schnellere suboptimale Lösungen bergen in sich die Gefahr einer falschen Hypothese "nachlaufen" zu wollen.

Das weitere Problem bei mehreren Objekten im Bild ist die Gruppierung von einzelnen Bildsegmenten zu zusammenhängenden Gruppen und die Interpretation dieser Gruppen als einzelne Objekte. Diese Aufgabe wird überwiegend durch anwendungsspezifische Analyseansätze (d.h. unter Kenntnis der gesuchten Objektklassen) gelöst. Dabei werden generell mehrere Objekthypothesen aufgebaut als die tatsächliche Anzahl der Objekte in der Szene. Die entscheidende Frage lautet in diesem Zusammenhang: wie kann eine konsistente Untermenge von Objekthypothesen ausgewählt werden.

1.2.7 Schlußfolgerung: Bewegte Objekte

Die Methoden der Verschiebungsschätzung im Bild bieten den Vorteil einer potentiell parallelen Verarbeitungsform mit mehreren einfachen Prozessoren. Der Nachteil jedoch ist die mangelnde Robustheit, da bereits kleine Nickbewegungen der Kamera bedeutende Fehler der Verschiebungsmessung verursachen können. Dieser Nachteil kann zum Teil in der Verfolgung von 2-D Merkmalen ausgeglichen werden. Diese Verfahren bedürfen aber bereits einer vorherigen Detektion und Zuordnung von Merkmalen im Bild, was wiederum zu mehreren konkurrierenden Analyseergebnissen führen kann. .

Ein schwerwiegender Nachteil sowohl der Verschiebungsschätzung im Bild wie auch der Merkmalsverfolgung liegt in der Interpretation der eigentlichen 3-D Objektbewegung als eine Verschiebung oder 2-D Bewegung. Aber auch die Berücksichtigung der Eigenbewegung ist nur bedingt möglich auf der Ebene der 2-D Erkennung. Aus diesen Gründen besteht in dieser Arbeit die Notwendigkeit einer modellbasierten 3-D Objekterkennung. Zwar ist auch hier das Korrespondenzproblem von Objekten zu lösen, aber die Anzahl der zu verfolgten Objekthypothesen ist um ein vielfaches geringer als die Anzahl der Bildsegmente.

1.3 Systemansätze – Stand der Technik

Die Verarbeitungsart

Im Falle dynamischer Bildanalyse muß der Aspekt der *Echtzeitverarbeitung* erwogen werden. Die Bilddeutung müßte im Prinzip in Echtzeit erfolgen. Dabei kann Echtzeit in jeder Anwendung unterschiedliche Zeiteinheit bedeuten, abhängig von der Dynamik der beobachteten Prozesse. Für die Analyse von Verkehrsszenen erlaubt uns der heutige Stand der Technik noch keine Erfüllung der Echtzeitbedingung auf einem Monoprozessor, so daß die gesamte Bildfolge vorerst im Rechner abgespeichert wird. Dabei kann man zwei Verarbeitungsarten unterscheiden – die *schritthaltende* und *retrospektive* Art. Bei der schritthaltenden Verarbeitung entspricht die Analysereihenfolge der Einzelbilder ihrer Reihenfolge in der Bildfolge. Bei einer retrospektiven

Auswertung ist die Bearbeitungsreihenfolge beliebig – sie kann beiderseitig variieren zwischen früheren und späteren Bildern. In unserer Arbeit fordern wir lediglich eine *schritthaltende* Verarbeitungsstrategie, d.h. die Einhaltung der Reihenfolge der verarbeiteten Daten und einen definierten (endlichen) Bedarf an Zeit und Speicherplatz je Verarbeitungsschritt (Bild).

Die Analysemodelle

Generelle Analysemodelle für Bildauswertung wurden vorgeschlagen:

- ein Arbeitszyklus für statische Auswertung (Kanade, Nagel) [Kan80], [Nag79]

Dieser Ansatz besteht aus einem generellen Zyklus von Analyseschritten: Hypothesen-Generierung und -Verifizierung. Die Abfolge ist fest vorgegeben, so daß von einer linearen Analysestrategie gesprochen werden kann.
- Transformationen im Suchraum (Niemann) [Nie81]

Die Analysestrategie ist als ein Pfad im Suchbaum zu sehen. Die Knoten stellen Analyseergebnisse dar, die Kanten entsprechen der Anwendung von Operatoren. Damit sind die Teilergebnisse lokal an einen Suchbaumknoten gebunden.
- eine "Blackboard" von bi-gerichteten Verarbeitungsprozessen (Hanson & Riseman) [HR78], [RH88]

Dieses Analysemodell kann als eine Mischung von beiden vorherigen gesehen werden. Verschiedene Analyseprozesse generieren Teilergebnisse, die auf einer allgemein zugänglichen "blackboard" abgelegt werden. Ein graphenbasiertes Suchverfahren steuert die Analyse durch Anstoßen von Analyseprozeduren, die sowohl auf das Modell, wie auch auf die globalen Analyseergebnisse zugreifen können. Entlang eines Pfades im Suchbaum wird eine modellbasierte Interpretation schrittweise entwickelt. Im Unterschied zu der vorherigen Strategie existiert hier ein globaler Zugang zu den Analyseergebnissen.

Diese Modelle wurden dann weiterentwickelt für den Fall der dynamischen Analyse: Interpretationszyklus mit generischer, zeitlicher Beschreibung ([Nag88]), und schritthaltende Graphensuche ([WN91]). Als wesentliches, gemeinsames Merkmal dieser dynamischen Analysemodelle ist die Propagation von Analyseergebnissen von einem Zeitpunkt zum nächsten zu sehen. Insbesondere die Ergebnisse der Segmentierung – ein Bild unabhängig vom anderen – zeichnen sich durch hohe Instabilität aus. Diskretisierungsfehler, Rauschen, Empfindlichkeit gegenüber Schwellwerten führen dazu, daß unterschiedliche Bildsegmente für dasselbe Objektteil in zwei aufeinanderfolgenden Bildern extrahiert werden. Der Einsatz von Modellwissen auf einer höheren Verarbeitungsebene und die damit verbundene Einschränkung der zu erkennenden Welt führt zu robusten Ergebnissen. Eine dynamische Rückkopplung zwischen aktuellen Ergebnissen und der Analyse des nächsten Bildes führt, z.B. durch Einsatz von adaptiven Methoden, zur einer weiteren Verbesserung der Ergebnisse.

Die Systemrealisierung

Die Systemrealisierung vom Gesichtspunkt der Informatik ist grundsätzlich in einer von drei folgenden Formen denkbar:

- Ein *prozedurales* Analysesystem

In diesen Systemen wird *anwendungsunabhängiges* Wissen (sogenannte Methoden) eingesetzt zur Extraktion von relativ einfachen Daten, die keine komplexe Struktur aufweisen (auch generell als einfache *Muster* in der Mustererkennung bezeichnet ([Nie81])

werden). Eine anwendungsunabhängige Verarbeitung bedeutet, daß die eingesetzten Methoden weitgehend unabhängig sind vom Typ der abgebildeten Szene. Zu den Methoden zählen gewöhnlich Verfahren zur ikonischen Bildverarbeitung und Segmentierung, aber auch die Extraktion der dynamischen Verschiebungsinformation oder Verfahren zur prototypen-freien Flächen- oder Objektrekonstruktion können hier zugeordnet werden.

Die Analysestrategie ist in diesem Fall meistens "fest verdrahtet" und prozedural definiert, d.h. die Analyse besteht aus einer fast determinierten Abfolge (eventuelle Optionen sind möglich) von prozedural definierten Analyseschritten.

Diese Form ist für relativ kleine Systeme geeignet, die höchstens zwei Datenaabstraktionsebenen umfassen.

- Ein *modellbasiertes* Analysesystem

Das Wissen über ein Anwendungsgebiet wird in Form eines *Modells* dem System zur Verfügung gestellt. Dabei ist hier zweitrangig ob das Modell überwiegend prozedural oder auch in einer deklarativ-prozeduralen Sprache dargestellt ist. Das Ziel der Analyse ist den, nach vordefinierten Kriterien optimalen, Abgleich zwischen den Bilddaten und dem Modell zu finden. Es wird dazu ein problemangepaßtes Kontrollverfahren verwendet.

Diese Systemform ist dann geeignet, wenn komplexe Objekte analysiert werden sollen (auch als *komplexe Muster* bezeichnet). Dazu wird Wissen auf mehreren Abstraktionsebenen eingesetzt, so daß eine Struktur des Modells erkennbar ist.

- Ein *wissensbasiertes* Analysesystem

Ein *wissensbasiertes* System dient als Gerüst für den Systementwurf und Integration einzelner Systemkomponenten, die das Wissen unterschiedlicher Art und auf verschiedenen Abstraktionsebenen verarbeiten. Die hier wichtigsten Systemmodule eines *wissensbasierten* Bildanalyseystems sind [Nie85], [NB87]:

1. *Modell* – Anwendungsabhängiges Wissen in *deklarativer* und *prozeduraler* Form über ein Anwendungsgebiet, als Ergebnis des Akquisitionsprozesses.
2. *Kontrolle* – Eine problemunabhängige Verarbeitungsstrategie, die entsprechende *Analyse*-Prozesse mit relevanten Datenmengen aktiviert. Zusätzlich zur Analyse können solche Prozesse gesteuert werden wie *Dialog* and *Erklärung*.
3. *Datenbasis* – Repräsentation von Zwischenergebnissen und Bildbeschreibungen.
4. *Methoden* – Modellunabhängiges Wissen für die ikonische Bildverarbeitung und Segmentierung.

Diese Systemform ist dann geeignet, wenn mehrere verwandte Anwendungen sich nur durch die verwendeten Modelle unterscheiden und ansonsten dieselbe Analysestrategie verwenden können. Insbesondere bei komplexen Systemen ist diese Form geeignet, da sie durch Austausch der Modellelemente und nur geringfügige Änderung von einigen Kontrollprozeduren einen schnellen Aufbau von Analysesystemen für verwandte Anwendungen unterstützt.

Viele *Wissensrepräsentationsmethoden* wurden bereits für die 3-D Objekterkennung vorgeschlagen: *Produktionenmengen* [HM86], *Grammatiken* [Fu82], und *strukturelle Methoden*. Unser Ansatz fällt in die letzte Kategorie. Hier können Unterklassen unterschieden werden: *Relationenstrukturen* [HS79], [Rad84], *Graph- und Struktur-Grammatiken* [Kas87] sowie *Rahmen* [Bro81] und *Semantische Netzwerke* [NSSK90].

Entsprechend den Wissensformen wurden *Kontrollverfahren* für wissensbasierte Systeme vorgeschlagen die in folgende Klassen fallen: *Suchverfahren* (Zustandsraumsuche,

UND/ODER Graphensuche), *linguistische Analyse* mit syntaktisch-*semantischen* Parsern, *Strukturabgleich* und *Relaxation*.

Die Analysestrategie

Einige Kriterien zur Beschreibung einer modell- oder wissenbasierten Analysestrategie sind in Tabelle 1.2 zusammengefaßt.

Die Analysestrategie	
Ablauf	deterministisch ("fest verdrahtet", linearer Zustandsraum)
	nicht deterministisch (ergebnisabhängig, verzweigter Zustandsraum)
Abgleichsrichtung	ein Modell-zu-Bild-Abgleich (modellabhängige Anzahl)
	ein Bild-zu-Modell-Abgleich (datenabhängige Anzahl)
	gemischter Abgleich (alternierend)
Datenbasis	zentrale Ergebnisse (auf der "blackboard")
	lokale Ergebnisse (in "Suchknoten")
Abgleichstyp	Relaxation
	Hough Transformation und Clusteranalyse
	Suchverfahren
Konsistenzverwaltung	stets konsistente Teilergebnisse
	teilweise inkonsistente Teilergebnisse
Einschränkungstest	einzelne Beschränkungen
	kombinierte Beschränkungen
	Transformationsmatrizen
Lösung der Verdeckung	Nullelement-Abgleich
	Partieller-Abgleich
Konsistenzauflösung	statistische Maße
	Konfigurationstest (Visibilitätstest)

Tabelle 1.2: Kriterien zur Klassifikation von Analysestrategien

Dabei muß die für die Analyse eines einzelnen Bildes optimale Strategie nicht gleich die beste Strategie bezüglich der Bildfolgenanalyse sein. Für die Konstruktion einer schritthaltenden Analysestrategie müssen neben obigen Kriterien für die Einzelbildanalyse insbesondere noch folgende Fragen beantwortet werden:

1. Wie und zum welchen Zeitpunkt entscheidet man zwischen konkurrierenden Analyseergebnissen ?
2. Wie wirken sich frühere Ergebnisse auf die Analyse zum nächsten Zeitpunkt aus ?

Es liegt nahe, daß optimale Entscheidungen zur Analysestrategie, die für die Analyse eines einzelnen Bildes getroffen wurden, nicht gleich als optimal für die Strategie bezüglich der Bildfolgenanalyse gelten müssen.

Beispiele

Als Beispiele von Systemen zur wissenbasierten Bildanalyse werden hier die Systeme *ACRONYM*, *VISIONS* und *SIGMA* kurz erläutert.

ACRONYM

Das *ACRONYM* System von Brooks [Bro81] baut sein Modell aus drei Graphen auf: *Objekt-Graph*, *Restriktionen-Graph* und *Prädiktions-Graph*. Diese werden durch *Rahmen* repräsentiert. Die *Erkennungsstrategie* ist eine Iteration von drei Prozessen: der *Suche* im Restriktionen-Graph, der *Prädiktion* von 2-D Objektinvarianten und Quasiinvarianten und der *Bildbeschreibung* (lokaler Abgleich der vorhergesagten Merkmale und der Bildmerkmale, sowie aus e Kombination der lokalen Ergebnisse zu *Clustern*). Die Prozesse der Prädiktion und Interpretation sind definiert als *Produktionen-Regeln*. Die Basis für Berechnungen bildet ein nichtlineares System zur Verarbeitung von Beschränkungen.

VISIONS

Das *VISIONS* System von Hanson & Risemann [HR78], [RH88] definiert eine generelle Methodik für wissensbasierte Bildauswertung. Das Wissen in Form eines semantischen Netzwerkes ist abgelegt als eine Menge von *Schemata* mit prozeduralen und deklarativen Teilen. Unter anderen enthalten sie auch Prozeduren zur Objekterkennung in Form von Hypothesen-Generierung und Verifikation. Eine Wissensbasis ist organisiert als eine Hierarchie mit Hilfe von PART-OF und IS-A Relationen. Die Kontrolle aktiviert Analyseprozesse aus drei Klassen: *Fokussierung*, *Expandierung* und *Verifikation*. Das *Analysemodul* erzeugt inkrementell die Bildbeschreibung. Dabei findet eine Auswahl innerhalb von drei konkurrierenden Mengen statt: zuerst wird ein Knoten im Interpretationsraum ausgewählt, dann die Hierarchiestufe der Wissensbasis und zum Schluß ein Schema auf dieser Stufe. Weil der Interpretationsprozeß über einzelne Schemata verteilt ist, findet die Kommunikation über eine *Blackboard* statt, d.h. alle Ergebnisse in der Datenbasis sind für jeden anderen Prozeß zugänglich.

SIGMA

Auch im System *SIGMA* von Hwang, Davis & Matsuyama [HM86] werden *bottom-up* und *top-down* Analyseprozesse in einem Erkennungsprozeß aktiviert. *SIGMA* gliedert sich in drei Module: niedrige Stufe (LL), hohe Stufe (HL) und Anfragebeantwortung (QA). Das HL Modul verarbeitet die Segmentierungsdaten basierend auf dem Objektmodell und generiert ein *Interpretationsnetz*. Dabei findet eine iterative Analyse statt, auf der Basis von Hypothesen-Generierung-Integrierung und Verifikation. Das QA Modul wird aktiviert zum Schluß jedes Iterationsschrittes um zu überprüfen, ob das Ziel der Analyse, dargestellt als eine Anfrage, erreicht ist. Das Szenenmodell ist abgelegt in einer kombinierten Repräsentation von Rahmen und Produktionen-Regeln, die einen abstrakten Graphen mit Knoten, Kanten und Regeln implementieren. Die Kanten beschreiben ausschließlich die Spezialisierungshierarchie. Die Teilhierarchie wird durch Regeln ausgedrückt. Die Analysestrategie besteht aus vier Schritten: Hypothesenbildung als *Vorhersage* von Objekten im Bild, *Integration* von Hypothesen durch Clusterbildung, *Abstrahierung* eines Clusters zu einer komplexen Hypothese, *Verifikation* der Hypothese durch Berechnung von restlichen Attributewerten.

Schlußfolgerung: Systemansatz

Eine objektorientierte Darstellung des Wissens ist einer prozeduralen vorzuziehen, weil:

- Anpassungen an andere Weltausschnitte leichter durchführbar sind
- Das deklarative Modell gegenüber dem prozeduralen leichter automatisch erlernbar ist

- Integrierung in ein kognitives System, das grundsätzlich wissensbasiert realisiert wird, wird erleichtert. Auch unterschiedliche Systeme, die z.B. verschiedene Signaldaten verarbeiten, können in ein System integriert werden.

Neben pragmatischen Aspekten ist die Anwendung eines wissensbasierten Systems von Bedeutung, wenn es darum geht, anwendungsübergreifende Merkmale der Wissensverarbeitung aufzugreifen und explizit darzustellen. Dank der klaren Strukturierung des Wissens und der expliziten Trennung zwischen Modell und Kontrolle können die Ergebnisse der Forschung auf ähnliche Anwendungen leicht übertragen werden.

Die Schlüsselfrage einer erfolgreichen Analysestrategie ist die optimale Berücksichtigung der daten- und modellbedingten Analyseschritte. Auch muß die sie implementierende Kontrollkomponente syntaktisch relativ einfach sein, damit eine erfolgreiche vielseitige Verwendung dieser Komponente möglich wird.

Für eine schritthaltende Kontrolle ergibt sich noch ein wichtiger zeitaspekt – die Stabilität der einmal getroffenen Entscheidungen sollte gewahrt sein. Im Unterschied zur statischen Analyse, wo eine optimale Lösung bereits während der Einzelbildanalyse erreicht sein muß, sollte dies hier erst dann erfolgen, wenn die Entscheidung als ausreichend sicher gilt.

Die Komplexität der Analyse wird auch bedingt durch die Art der Abbildung des Modells in syntaktische Datenstrukturen. Zwei syntaktisch gleiche Modelle, aber mit unterschiedlicher Komplexität ihrer Semantik, können entweder zu einer suchraumintensiven Suche führen oder es auch bei einer linearen Analyse belassen.

1.4 Ziele und Lösungsansatz

1.4.1 Die Ziele

Das Hauptziel der Arbeit besteht in der Entwicklung von Verfahren zur Erkennung von bewegten Objekten in monokularen Bildfolgen von Verkehrsszenen. Dabei soll der Fall einer bewegten Kamera im Vordergrund stehen, aber auch eine Rekonfigurierung des Systems für Anwendungen mit stationärer Kamera soll möglich sein und als ein Unterfall des generellen Systems gelten.

Die in der Zukunft angestrebte Anwendung des Systems zur Verkehrsszenenanalyse definiert die Klassen der zu erkennenden Objekte: Fahrbahnmarkierungen, Fahrzeuge und Hindernisse (nichterkannte Objekte auf der Fahrbahn). Sie setzt auch eine echtzeitnahe Arbeitsweise voraus, so daß in absehbarer Zeit eine Echtzeitanwendung technisch realisierbar wird. Insbesondere bei solchen technischen Systemen werden Grenzen der Verwendung vieler iterativ arbeitender oder mit hohem Rechenaufwand verbundener Verfahren sichtbar. Das zweite Hauptziel der Arbeit ist deswegen die Definition und Zerlegung des Erkennungsproblems in Verarbeitungsstufen, die parallel oder als eine Pipeline ausgeführt werden können, die Entwicklung von neuen, relativ robusten Verfahren dazu sowie eine effiziente Integration dieser Verfahren in ein leistungsfähiges und robustes Gesamtsystem.

Es werden folgende vereinfachende Annahmen über die Art der Eingabedaten und über die Struktur der analysierten Szene getroffen:

- Eine Szene besteht sowohl aus bewegten wie auch unbewegten Objekten, wobei nur Objekte auf der Fahrbahn erkannt werden sollen.
- Von der Fahrbahn wird angenommen, daß sie auf einer Ebene liegt. Dabei sind Kurvenfahrten möglich.
- Die Eigengeschwindigkeiten des Kamerafahrzeuges sind bekannt. Sie wird modelliert als translative Geschwindigkeit in der Fahrbahnebene und Winkelgeschwindigkeit um die

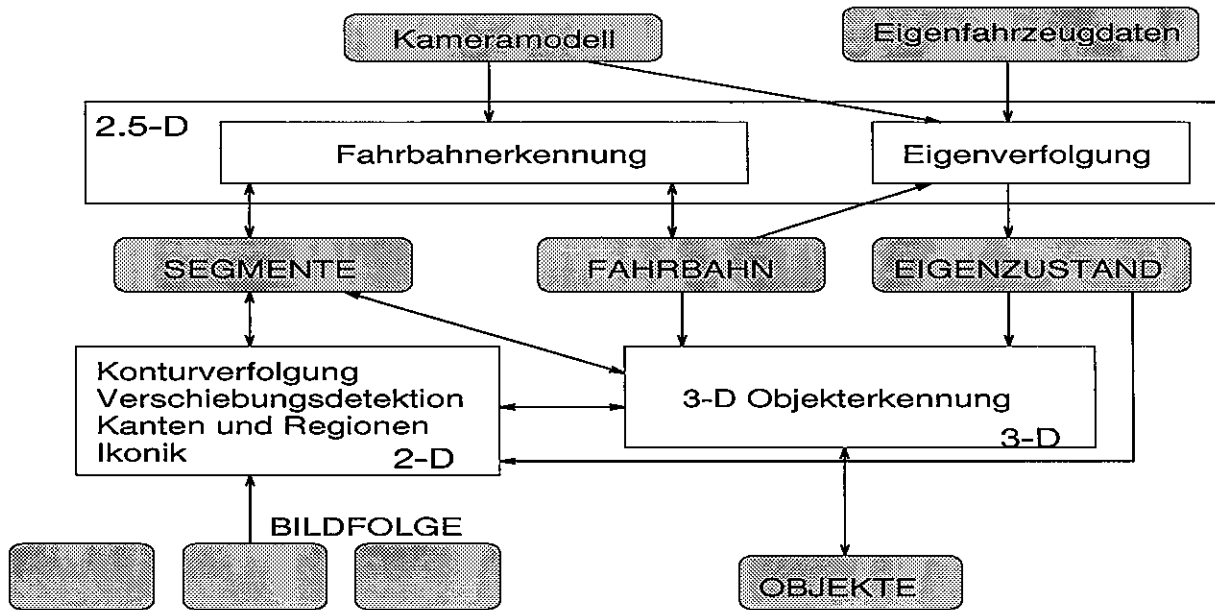


Bild 1.4.1: Das System zur adaptiven Objekterkennung auf Fahrbahnebene

Normale zur Fahrbahnebene. Die Höhe der Anordnung der Kamera über der Fahrbahn ist bekannt. Die Orientierung der Kamera gegenüber der Fahrbahn, sowie das Kameranicken sind im allgemeinen nicht bekannt.

- Eine Objektbewegung besteht aus einer Translation in der Straßenebene und einer Rotation um eine Normale zur Fahrbahnebene.
- Die Bewegungen der Objekte und des Kamerafahrzeuges unterliegen ausreichend glatten Änderungen.

1.4.2 Der Lösungsansatz

Den in dieser Arbeit gewählten Ansatz beschreiben wir wie folgt: *adaptive und schritthaltende Erkennung von bewegten Objekten auf Fahrbahnebene, mit perspektivisch verzerrter Bewegung im Bild, in monokularen Bildfolgen bei Eigenbewegung*. Das Ergebnis der Erkennung soll zu jedem Zeitpunkt eine *geometrisch-zeitliche Szenenbeschreibung* sein, d.h. Beschreibung der Anzahl der Objekte in der Szene, deren räumlicher Lage und deren Bewegung auf Fahrbahnebene.

Der adaptive Lösungsansatz wurde im Rahmen eines Testsystems auf einem Monoprozessor implementiert. Die funktionale Systemstruktur besteht aus drei Modulen (Bild 1.4.1):

2-D Segmentierung und Verschiebung im Bild

Dieses Modul enthält anwendungsunabhängige Prozeduren zur ikonischen Bildverarbeitung, zur Detektion von Bildsegmenten in Form von geschlossenen Konturen und zur deren Verfolgung im Bild. Eine Verschiebungsdetektion im Bild dient zur Klassifikation in bewegte und unbewegte Kanten.

2.5-D Fahrbahnerkennung und Eigenzustandsverfolgung

Das *Fahrbahnmodell* enthält anwendungsabhängiges Wissen über verschiedene Fahrbahnklassen. Die Ziele der zugehörigen Analyse ist die Bestimmung der Fahrbahnparameter und der Kameralage, sowie die Klassifikation der Segmente bezüglich der Fahrbahn.

3-D 3-D Objekterkennung

Dieses Modul enthält ein dynamisches Model der bewegten und unbewegten Objekte. Die Aktivierung dieses Wissens führt zur zur Initialisierung von Objekthypothesen, zur adaptiven Bestimmung der Objektzustände und zur Auswahl von konnsistenten Hypothesen.

1.5 Beitrag der Arbeit

Es wird ein Gesamtsystem zur Erkennung von bewegten Objekten in Verkehrsszenen vorgestellt, in dem auch mehrere neuentwickelte Ansätze zur Lösung von Teilaufgaben integriert sind. Diese Einzelelemente tragen zusammen mit weiteren, mehr oder weniger etablierten Verfahren der Mustererkennung zu einer robusten Realisierung des Objekterkennungsystems bei.

1.5.1 Systemintegration und Konfigurierung

Integration von Geometrie und Bewegung

Da die Bildanalyse der unteren Stufe für natürliche Szenen meistens nicht robust genug ist, suchen einige Autoren die Lösung in einem intensiven Nutzen der kognitiven Verarbeitungsebene ([Pau93]). Bis es aber soweit ist, das Objekte erkannt sind, muß erst eine erfolgreiche Integration der niedrigen Stufe und der Objekterkennungsstufe stattfinden. Hier fehlt es noch an robusten Lösungen. Bisherige Systeme zur Verkehrsszenenanalyse befassen sich meistens entweder nur mit einer low-level Bewegungsdetektions (z.B. [Kil92]) oder mit einer geometrisch bedingten modellbasierten Objekterkennung (z.B. [Wet95]) ohne ausreichende Zuhilfenahme und Integration von beiden Verarbeitungsebenen. In diesem Sinne bildet das hier vorgestellte System eine Weiterentwicklung des Ansatzes von Koller ([Kol92], [KDN93]), der eine modellbasierte Objektverfolgung mit einer bildbewegungsabhängigen Initialisierung koppelt, dies aber nur für eine stationäre Kamera tut.

Bei wenigen Bildmerkmalen muß die modellbasierte Objekterkennung optimale Hypothesen (Bewertungen) auch mit wenig zuverlässigen Analyseergebnissen treffen. Dadurch entstehen viele ähnlich bewertete Hypothesen, die zu hohem Analyseaufwand führen. Bewegungsbasierte Information kann zur Reduktion dieser Hypothesenmenge führen. Deswegen sollen die Methoden zur bewegungsabhängigen Tiefeschätzung auch berücksichtigt werden. In dieser Arbeit erfolgt eine Unterstützung der geometrisch bedingten Segmentierung durch eine pixelbasierte Bewegungsdetektion und Konturverfolgung im Bild. Dadurch werden zusätzliche Attribute eines Bildsegments generiert, was zu einer Einschränkung der Hypothesenmenge während der Objekterkennung führt. sowie zu einer robusteren Initialisierung von 3-D Zuständen von Objekthypothesen führt.

Der Ansatz zur adaptiven Erkennung

Als Grundlage von Lösungen auf mehreren Datenabstraktionsebenen dient der Ansatz der adaptiven Erkennung. Dieser Ansatz bedient sich des Schemas der parallelen Verfolgung von mehreren konkurrierenden Hypothesen verbunden mit rekursiver Schätzung von deren Zuständen und Bewertungen, sowie einer auf diesen Bewertungen basierenden Auswahl von besten Hypothesen. Der modellbasierte, adaptive Ansatz geht von einem dynamischen Modell aus, sowohl der Form wie auch der Bewegung, und wendet eine rekursive Schätzung während der Objektverfolgung an. Da hier die Störgrößen (Diskretisierungsfehler) nicht geringfügig sind, haben wir es mit dem *Filterproblem* [BS75] zu tun.

Bewegte und stationäre Kamera

Durch die Rekonfigurierung des Systems zwischen Analyse bei bewegter Kamera und Analyse bei stationärer Kamera (unter Beschränkung auf den Bereich der Verkehrsszenen) wird die Verallgemeinerung des Ansatzes gegenüber bisherigen Ansätzen praktisch belegt. Dadurch wird auch die breite Klasse von möglichen Anwendungen des Systems verdeutlicht.

Wie es im weiteren Verlauf der Arbeit deutlich wird, ist aber auch im Fall des Untersystems bei stationärer Kamera ein wesentlicher Unterschied zwischen dem in vorliegender Arbeit beschriebenen adaptiven Ansatz zur Objekterkennung und der modellbasierten Objektverfolgung im Sinne von [Kol92] festzustellen. Im Unterschied zu [Kol92] wird hier eine statistische Unabhängigkeit der Bewertungen von Einzelmessungen von der momentanen Objekthypothese erzielt.

1.5.2 Neuentwicklungen für Teilprobleme

Bei den neuentwickelten Lösungen von Teilproblemen handelt es sich um folgende Verarbeitungsschritte:

1. Pixelbasierte Bewegungsdetektion im Bild bei Eigenbewegung
2. Integrierte Detektion von Liniensegmenten und Regionen
3. Konturverfolgung im Bild mit 3-D Konturattributen
4. Adaptive Fahrbahnerkennung
5. Adaptive 3-D Objekterkennung auf Fahrbahnebene bei Eigenbewegung
6. Eine objektorientierte Modellform und eine schritthaltende Suchbaumkontrolle

Verschiebungsdetektion

Es muß betont werden, daß die Bewegungsschätzung im Bild bisher grundsätzlich nur für Anwendungen mit stationärer Kamera entwickelt und eingesetzt wurde. Verschiedene Versuche, diese Verfahren für natürliche Szenen auch bei bewegter Kamera erfolgreich anzuwenden, ohne Hinzunahme eines 3-D Modellwissen, haben sich als wenig brauchbar für die Detektion von tatsächlich bewegten Objekten erwiesen ([PM91], [EE91], [BK93]).

Eine Möglichkeit, diese Methoden auch bei bewegter Kamera zu verwenden, liegt in der Berücksichtigung der Eigenbewegung (falls diese bekannt ist) und der Ausgleichung der perspektivischen Verzerrung. In diesem Sinne folgt die in dieser Arbeit vorgenommene Entwicklung von low-level Bewegungsdetektoren der Arbeit in ([WZN93], [ZN92]). Überlegungen ähnlicher Art, vor der Bewegungsschätzung den Perspektiveffekt durch logarithmische Transformation der Bildebene auszugleichen, werden neulich auch für Anwendungen mit stationärer Kamera vorgeschlagen ([DK95]).

Konturverfolgung

Eine weitere Frage betrifft die Wahl der Bildelemente, für die eine Verschiebung im Bild ermittelt werden kann sowie die Anzahl der Bilder in denen nach Korrespondenzen gesucht wird. Die konventionellen Verfahren arbeiten entweder auf Pixelebene (*gradienten-basierte Verfahren*, *Blockvergleich*) oder auf Bildmerkmalen, wie Ecken und Kanten. Ein optischer Fluß wird gewöhnlich für zwei aufeinanderfolgende Bilder generiert. Enkelmann et al. [EKNZ88] und Salari & Sethi [SS90] haben die Problematik einer gleichzeitigen Verarbeitung von mehreren

Bildern für die Zuverlässigkeit der Bewegungsergebnisse unterstrichen. Der Begriff der *Raum-Zeit*-Verarbeitung wurde vorgeschlagen von Jain ([Jai88], [LJ89]) für eine Mehrbild-Umgebung. Eine rekursive Variante der Verschiebungsschätzung beruht auf der dynamischen Systemtheorie. Bei bewegter Kamera kann nur solche Bildverschiebung stabilisiert werden, die aufgrund signifikanter Bewegung eines Objektes relativ zur Kamera entsteht ([BC86]).

Adaptive Fahrbahnerkennung

Der adaptive Ansatz wird auch zur Fahrbahnerkennung eingesetzt, indem die Fahrbahn und die Kameraparameter als dynamische Systemzustände modelliert werden. Im Gegensatz zu bekannten Verfahren ([PJ92], [ST92]), die kantenbasiert arbeiten, erfolgt hier die Detektion der Fahrbahnmarkierungen sowohl aufgrund der Kantendaten, wie auch aufgrund einer homogenen Fahrbahnregion. Ein parallel zu dieser Arbeit entwickeltes Verfahren zur Straßenerkennung in [Ric95] basiert vorwiegend auf der Regionendetektion und -verfolgung.

Weitere wichtige Merkmale des Ansatzes sind die modellbasierte Detektion von Fahrbahnhyphothesen und das parallele Verfolgen von mehreren Fahrbahnklassen. Die Fahrbahnkrümmungsdetektion wird hier durch die dynamische Erkennungsart ermöglicht. Es werden gut stabilisierte Parameter des Eigenzustandes, wie die Eigenlage bezüglich der Fahrbahnmitte bzw. Kameraneigung bezüglich des Fluchtpunktes und die Eigenwinkelgeschwindigkeit, zur Messung der momentanen Fahrbahnkrümmung eingesetzt. Diese Methode ist besonders vorteilhaft gegenüber anderen Methoden in [PJ92], [ST92], und [Ric95], die solche Krümmungen direkt im Bildmaterial zu detektieren versuchen, wenn die Fahrbahnränder nur teilweise sichtbar sind.

Adaptive Objekterkennung

Eine prototypenfreie 3-D Objektrekonstruktion verlangt nach fehlerfreier Segmentierung. Für Szenen mit natürlicher Beleuchtung, teilweise unbekannter Kamerabewegung, sowie bei zu erwartenden großen Diskretisierungsfehlern ist das zur Zeit technisch nicht erreichbar. Deswegen wird hier die modellbasierte Objekterkennung angewendet.

Die zu analysierenden Bildfolgen eignen sich nicht für die "bootstrap" Ansätze zur bewegungsbasierten Objekterkennung. Das Problem liegt darin, daß "bootstrap" Algorithmen entweder ausgewählte Bildmerkmale über eine ausreichend lange Bildfolge verfolgen (eine ausreichende Anzahl von Lageparametern in dieser Bildfolge bildet die Voraussetzung für die Überdeterminierung der Strukturlösung) oder auf optischem Fluß aufbauen. In beiden Fällen können bei bewegter Kamera eine relativ lange Verfolgung der Lage solcher kleiner Bildelemente oder eine stabile Bewegungsschätzung nicht erwartet werden.

Wissensbasierte Realisierung

Wir gehen der Frage nach, ob eine suchbaumbasierte Kontrolle mit lokalen Analyseergebnissen geeignet ist für die dynamische Objekterkennung. Insbesondere wird ein Vergleich mit einer zentralen Verwaltung von Analyseergebnissen vorgenommen, wie es dies z.B. bei den Begründungsverwaltungssystemen der Fall ist ([DK86], [DF89]).

Als weiteres wird eine in diesem Zusammenhang optimale objektorientierte Modellstruktur vorgeschlagen. Die Optimalität des Modells bedeutet hier, daß eine relativ einfache schritthalte Kontrolle eingesetzt werden kann (z.B. einfacher als in [Wet95]) und daß eine Explosion der Analysekomplexität bei "unscharfer" Bewertungsfunktion vermieden wird (was z.B. beim Einsatz des Verfahrens in [Wei92] passieren könnte).

1.6 Gliederung der Arbeit

Die vorliegende Arbeit gliedert sich neben dieser Einführung in sieben weitere Kapitel und einen Anhang. Im Kapitel 2 wird der dynamische Ansatz zur Objekterkennung zusammenfassend beschrieben. Es wird eine funktionale Lösung des Problems entworfen und die Rekonfigurierung des Systems für Anwendungen mit stationärer Kamera wird angesprochen.

In den Kapiteln 3–5 werden die drei Module des Erkennungssystems in prozeduraler Form im einzelnen beschrieben. Der Segmentierer mit der 2–D Bewegungsdetektion sind Gegenstand des Kapitels 3. Die modellbasierte Fahrbahnerkennung und Eigenzustandsverfolgung werden in Kapitel 4 vorgestellt. Die adaptive 3–D Objekterkennung wird schließlich in Kapitel 5 erläutert. Das Kapitel 6 befaßt sich mit dem Konzept einer wissensbasierten Realisierung des beschriebenen Erkennungssystems. Insbesondere wird dabei detailliert auf das optimale Zusammenwirken des Modells und der schritthaltenden Kontrolle eingegangen.

Im Kapitel 7 werden die erzielten Testergebnisse vorgestellt und diskutiert. Im letzten Kapitel erfolgt die Zusammenfassung der Arbeit und ein Ausblick auf noch offene Fragen.

Kapitel 2

Der adaptive Ansatz zur Objekterkennung

In diesem Kapitel wird die funktionelle Struktur des realisierten Systems zur Erkennung von bewegten Objekten auf Fahrbahnebene bei Eigenbewegung vorgestellt (Abschnitt 2.1). Eine umfassende Beschreibung der drei Systemmodule – Bildsegmentierung–Bildverschiebung, Fahrbahnerkennung–Eigenzustandsverfolgung und 3–D Objekterkennung – erfolgt in den späteren Kapiteln 3 bis 5. In Abschnitt 2.2 werden der adaptive Erkennungsansatz und seine mathematischen Grundlagen erläutert. Der Ansatz wird an mehreren Stellen des Erkennungssystems angewendet, für unterschiedliche Datentypen. Eine zusammenfassende Beschreibung dieser Anwendungsstellen im System erfolgt in Abschnitt 2.3. Zum Schluß dieses Kapitels wird die Systemrekonfigurierung bei stationärer Kamera angesprochen (Abschnitt 2.4).

2.1 Die funktionelle Systemstruktur

Die funktionelle Systemstruktur zur adaptiven Objekterkennung bei Eigenbewegung ist in Bild 2.1.1 gegeben. Das System besteht aus drei Systemmodulen (kurz bezeichnet als die 2–D, 2.5–D und 3–D Blöcke). Das 2–D Modul enthält anwendungsunabhängige Verfahren zur Bildsegmentierung und Verschiebungsschätzung. Das 2.5–D Modul bedeutet eine modellbasierte Fahrbahnerkennung und Eigenzustandsverfolgung. Das 3–D Modul beinhaltet modellbasierte Verfahren zur 3–D Objekterkennung. Die einzelnen Verarbeitungsschritte sind durch Einträge in nicht angefüllten Kästchen dargestellt. Die zu verarbeiteten Daten und Analyseergebnisse sind durch gefüllte Boxen repräsentiert. Die Reihenfolge der Verarbeitung von einzelnen Schritten ist meistens sequentiell, obwohl einige Verarbeitungsschritte auch parallel zueinander ausgeführt werden können (z.B. die Regionen– und Liniensuche neben der Verschiebungsdetektion).

Um Objekte in der Bildfolge zu erkennen, benötigen wir neben dem *Objektmodell* und dem *Fahrbahnmodell* (die implizit in den entsprechenden Verarbeitungsschritten der 2.5–D und 3–D enthalten sind) auch Wissen über die Aufnahmebedingungen (das *Kameramodell*) und die *Eigengeschwindigkeiten* des Kamerafahrzeuges.

In jedem Modul erfolgt eine zeitlich abhängige Schätzung der Ergebniswerte, vorwiegend durch die Stabilisierung der detektierten Messwerte in Einzelbildern. Es folgt eine kurze Erläuterung der Aufgaben einzelner Verarbeitungsschritte. Beispiele von wichtigsten Zwischenergebnissen sind in Bild 2.1.2 graphisch dargestellt.

2.1.1 Das 2–D Modul

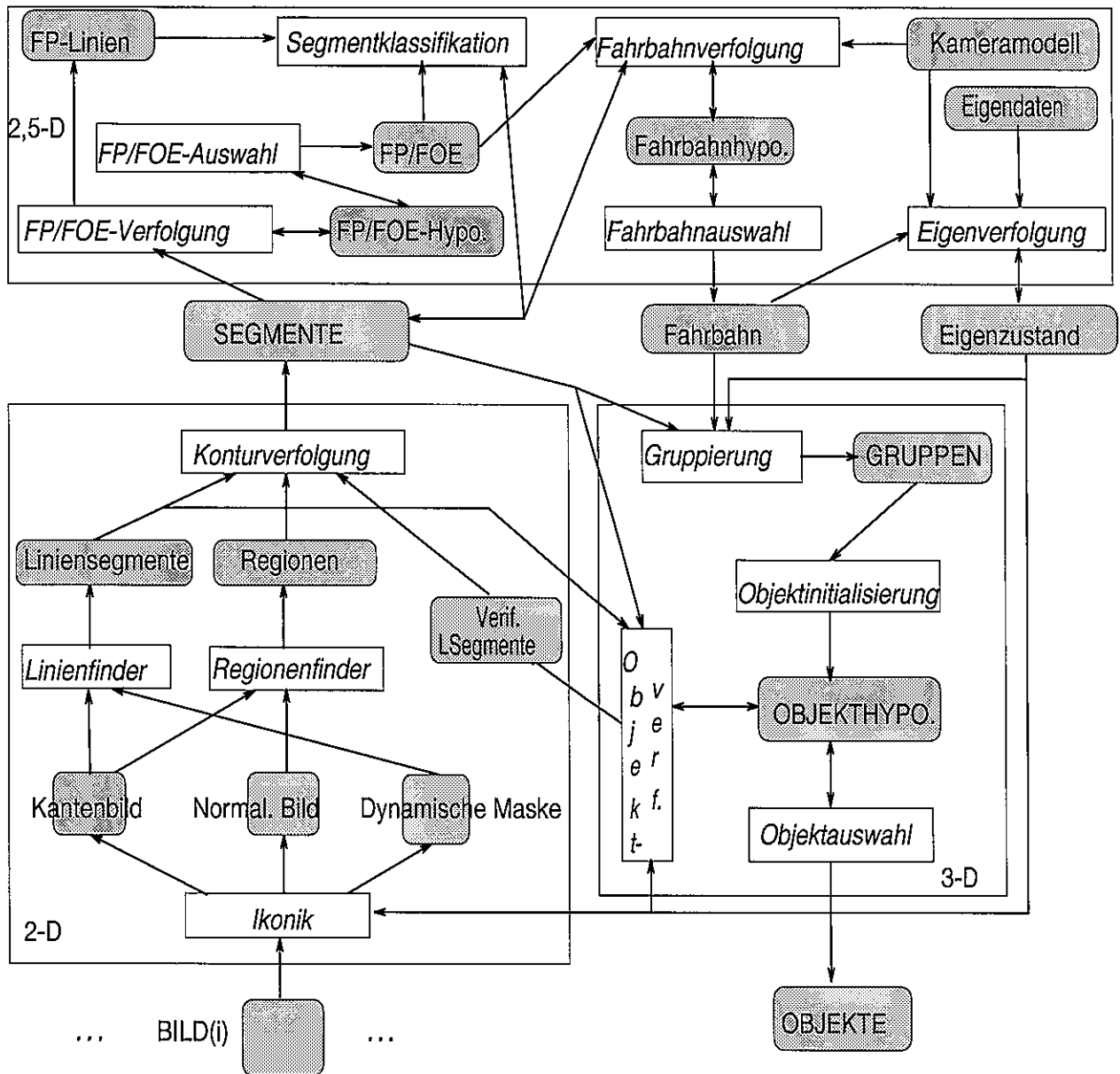


Bild 2.1.1: Die Struktur des Systems zur adaptiven Objekterkennung

Die wesentlichen Verarbeitungsschritte im 2-D Modul sind: *Ikonik*, *Linien-* und *Regionenfinder* sowie *Konturverfolgung*.

Ikonik

In diese Gruppe fallen Verfahren zur Bildunterabtastung und Bildnormalisierung aber auch ein Kantenoperator und die Erzeugung einer pixelbasierten Verschiebungsmaske bzw. eines optischen Flusses.

Nach einer später folgenden Segmentdetektion werden unter Einsatz des Kantenbildes und einer ausgewählten dynamischen Verschiebungsmaske die Segmente als bewegungsrelevant oder nichtrelevant klassifiziert.

Beispiele der drei Ergebnisbilder der ikonischen Verarbeitung sind in Bild 2.1.2 (a)–(c) gegeben.

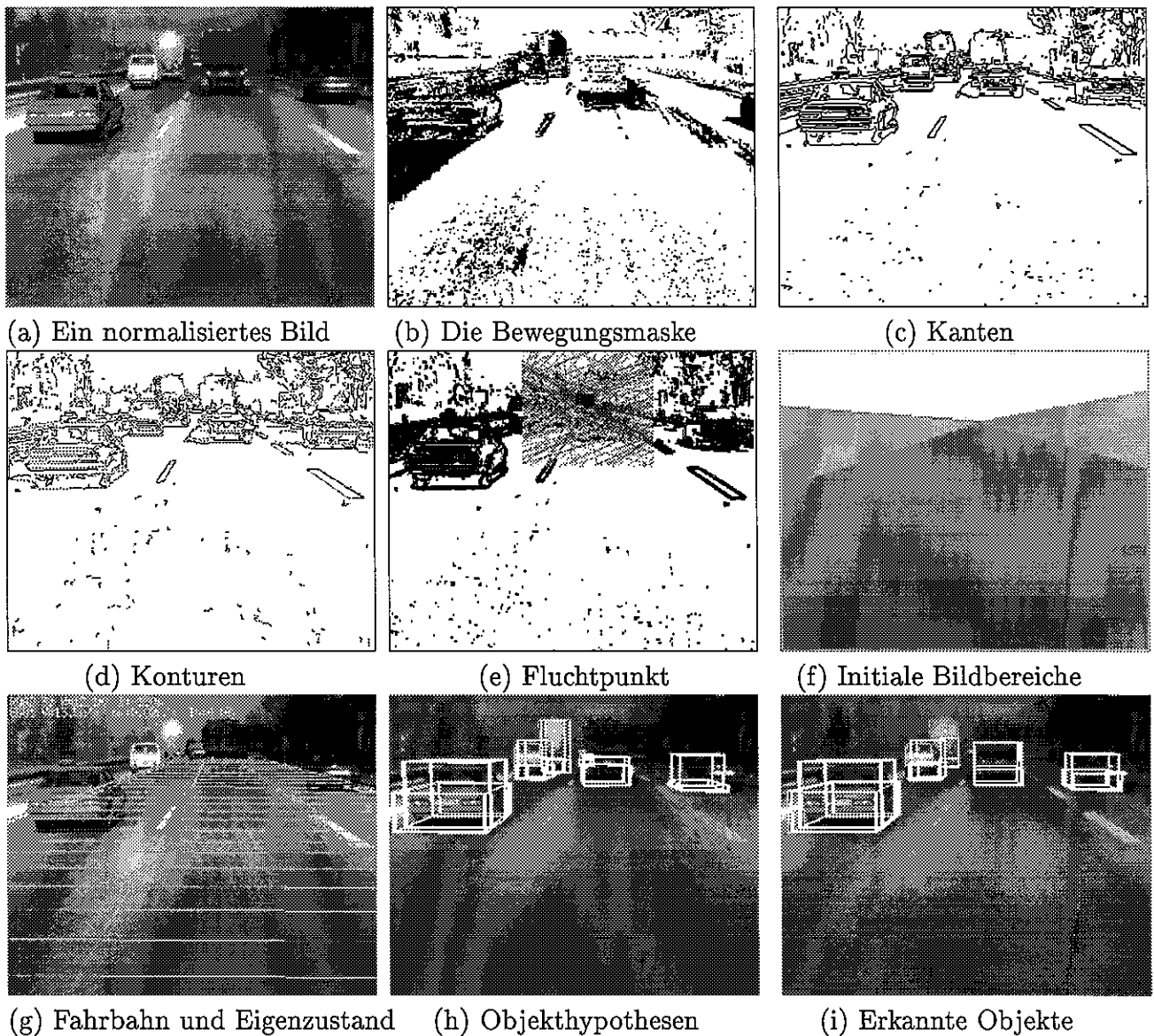


Bild 2.1.2: Beispiele von wichtigsten Zwischenergebnissen dargestellt in Form von Ergebnisbildern

Linien- und Regionenfinder

In diesem Schritt erfolgt die Detektion von Bildmerkmalen im aktuellen Bild. Es werden sowohl lineare Kantensegmente im Kantenbild gefunden, wie auch (im Prinzip parallel dazu) Regionen in einem durch das Kantenbild maskierten Originalbild.

Konturverfolgung

Als erstes erfolgt die Detektion von geschlossenen Konturen im aktuellen Bild. Dies läßt sich auf einen Konturschluß von solchen Liniensegmenten, die über dieselbe Region benachbart sind, zurückführen.

Als nächstes wird versucht eine Zuordnung der aktuellen Kontur zu Konturen, die in vorherigen Bildern gefunden wurden, zu finden. Es werden korrespondierende Konturen in bis zu N aufeinander folgenden Bildern ($N = 2 - 5$) ermittelt. Die entgeltigen 2-D Konturwerte werden auf Grund einer gewichteten Mittelung der Einzelwerte in einer kurzer Bildfolge von bis zu

N Bildern festgestellt. Die Konturattribute enthalten somit neben geometrischen Parametern, wie Lage, Länge und Fläche auch 2-D Verschiebungsvektoren von markanten Konturpunkten und den Wert der relativen Längenveränderung der Kontur.

2.1.2 Das 2.5-D Modul

Die Fahrbahnerkennung können wir als *Fluchtpunkterkennung*, *Segmentklassifikation* und *Fahrbahnerkennung* zusammenfassen. Die jeweilige Erkennung besteht aus zwei wesentlichen Schritten: Verfolgung und Auswahl. Daneben enthält das 2.5-D Modul auch eine Komponente zur Eigenzustandsverfolgung (*Eigenverfolgung*).

FP/FOE-Verfolgung und -Auswahl (Fluchtpunkterkennung)

Bedingt durch die unterschiedliche Art von Eingabedaten wird bei der Fluchtpunkterkennung unterschieden zwischen der *FP-Erkennung* und der *FOE-Erkennung*.

Aus der Menge der linearen Segmente werden hypothetische Fahrbahnmarkierungen im Bild detektiert, die auf sogenannten Fluchtpunktgeraden liegen. Diese Geraden sollen sich im Bild in einem Punkt schneiden, der als der geometrische Fluchtpunkt *FP* bezeichnet wird. Die adaptive Erkennung des Fluchtpunktes bedeutet, das bis zu drei Hypothesen parallel verfolgt werden und das die momentan bestbewertete Hypothese als der Fluchtpunkt erkannt wird. Ein ähnlicher Vorgang für Geraden auf denen die Verschiebungsvektoren von hypothetischen Hintergrundkonturen liegen führt zur Detektion des aktuellen dynamischen Fluchtpunktes *FOE* im Bild.

Konturklassifikation

Mit Hilfe des Fluchtpunktes *FP* erfolgt eine initiale Klassifikation der Konturen im Bild in Fahrbahnbereich (sowohl Fahrbahnmarkierungen wie auch Fahrzeuge), Umgebung und Horizont. Die Verschiebungsrichtung einer Kontur im Bild bezüglich des Fluchtpunktes *FOE* kann zu einer weiteren Konturklassifikation in bewegte und unbewegte Konturen beitragen. Die Genauigkeit dieser verschiebungsbasierten Schätzung ist aber bei bewegter Kamera im allgemeinen viel geringer als die der geometriebasierten Klassifikation.

Fahrbahnverfolgung und -Auswahl (Fahrbahnerkennung)

Es wird hier auf das *Kameramodell* zurückgegriffen. Dieses umfaßt ein Lochkameramodell – mit Angaben zur Bildfläche und Brennweite – und das Wissen über die Anfertigung der statischen Kamera im Fahrzeug und die möglichen Freiheitsgrade der Kamera bezüglich der Straße.

Unter Annahme ebener Fahrbahn und mit Kenntnis des Kameramodells können zwei Rotationswinkel der Kamera-zur-Fahrbahn-Transformation direkt aus dem Fluchtpunkt abgeleitet werden. Die detektierte Fahrbahnmarkierung wird in den 3-D Raum zurückprojiziert, um eine anschließende Detektion des Fahrbahntyps (z.B. zwei- oder dreispurige Fahrbahn), der Fahrbahnbreite und der Entfernung zwischen Fahrbahnmitte und Kamerafahrzeug zu ermöglichen. Somit wird eine vollständige momentane Transformation zwischen der Fahrbahn (mit Mittelpunkt der Fahrbahn als Ursprung der Koordinaten) und den Kamerakoordinaten gefunden.

Ähnlich wie bei der Fluchtpunkterkennung wird auch hier der adaptive Erkennungsansatz angewendet.

Eigenverfolgung

Für die Eigenzustandsverfolgung wird ein Bewegungsmodell der Kamera vorausgesetzt. Die Kamera wird als statisch montiert angenommen, so dass sich die Dynamik der Szenenabbildung auf die Bewegung des Eigenfahrzeuges und das zu ermittelnde (unbekannte) Kameranicken beschränkt. Die translatorische Geschwindigkeit und die Winkelgeschwindigkeit des Fahrzeuges auf Straßenebene werden als bekannt vorausgesetzt.

Falls die laufenden Angaben über die Eigengeschwindigkeit gegen Null tendieren, wird der Fall der stationären Kamera angenommen. Dies führt zu einer entsprechenden automatischen Rekonfigurierung des Systems.

2.1.3 Das 3-D Modul

Die 3-D Objekterkennung (das 3-D Modul) beinhaltet die Schritte der *Gruppierung*, *Objektinitialisierung*, *Objektverfolgung* und *Objektauswahl*. Der adaptive Erkennungsansatz findet dabei seine vollständigste Anwendung.

Gruppierung

Auf Grund der geometrischen und verschiebungsbasierten Konturmerkmale (ergänzt durch die 3-D Tiefenschätzung für Objekte auf der Fahrbahn) werden die Konturen zu Gruppen zusammengefaßt. Dabei werden drei Klassen von Gruppen unterschieden: Fahrbahnstreifengruppe, Gruppe für ein bewegtes Fahrbahnobjekt (Fahrzeug) und Gruppe für ein unbewegtes Objekt (Hindernis).

Objektinitialisierung

Es wird versucht auf der Basis jeder gefundenen Konturgruppe eine Objekthypothese zu generieren. Dieser Vorgang wird als Initialisierung des Zustandsvektors einer Objekthypothese aufgefaßt. Der Zustandsvektor besteht aus einem dynamischen Teil, welcher die Lage und Bewegung im 3-D Raum beschreibt, und aus einem Formvektor, der die Objektgröße und Form wiedergibt. Die Initialisierung des Zustandsvektors beruht sowohl auf den geometrischen Merkmalen, wie auch auf den Verschiebungsdaten.

Objektverfolgung

Die Objekthypothesen werden über die Zeit verfolgt und dabei werden ihre Zustände aufgefrischt und stabilisiert. Die als sicher geltenden Objekte unterliegen einer spezialisierten Objektverfolgung (mit sogenannter 2-D Messung), in der die Objektprimitiva in die Bildebene projiziert und mit den neu ermittelten Bildsegmenten verglichen werden. Die noch relativ unsicheren Hypothesen werden einer generellen Objektverfolgung zugeführt (mit sogenannter 3-D Messung), die einen Objekt-zu-Objekt-Abgleich vornimmt und die jeweilige Hypothese entsprechend auffrischt.

Falls die Objektverfolgung erfolgreich verläuft, was durch die ebenfalls laufend aufgefrischte Güte gegeben ist, wird die Hypothese weitergeführt. Andernfalls wird sie nach einer gewissen Zeit verworfen.

Objektauswahl

Die erfolgreiche Stabilisierung eines Objektzustandes bedeutet zugleich die Verifikation der zugehörigen Hypothese. Die Varianzen der Objektschätzung und die Länge der Verfolgung bilden zusammen das entscheidende Kriterium für die Auswahl zwischen konkurrierenden Objekthypothesen.

2.1.4 Der Analysefluß

Das *Zusammenwirken* der drei Module erfolgt durch einen bidirektionalen Informationsfluß. Der Austausch von Analyseergebnissen zwischen zwei Modulen in der "bottom-up" Richtung ist bekannt – die Segmentierungsdaten für das aktuelle Bild werden weitergereicht zur Fahrbahnerkennung und Objekterkennung. Die aus der Fahrbahnerkennung resultierende Transformation zwischen den Bild- und Fahrbahnkoordinaten trägt wesentlich zur Objekterkennung bei. Umgekehrt wird diese Transformation auch der low-level Bewegungsdetektion zugeführt um somit den Einfluß der Eigenbewegung im Bild auszugleichen. Die Fahrbahnerkennung bildet auch eine Voraussetzung für die Klassifikation von Segmenten. Die "top-down" Rückkopplung von der Objekterkennung zur Segmentierung erfolgt durch eine gezielte Suche nach Liniensegmenten im Bild, die durch Vorhersage der verfolgten Objekthypothesen gesteuert wird.

Der Analysefluß im System wird als eine sich für jedes Bild k wiederholende Verarbeitung von Analyseschritten gesehen (Bild 2.1.3).

1. Ikonische Bildverarbeitung (Bild(k), Bild ($k-1$))
2. Regionenfinder
3. Linienfinder
4. FP-Verfolgung
5. FP-Auswahl
6(a) Vorhersage des aktuellen Eigenzustandes(k)
FOR jede spezialisierte Objekthypothese $s(k)$
7. SPEZ.OBJ.VERFOLGUNG($s(k)$, Segmente(k))
8. Konturverfolgung – für übrige Segmente
9. FOE Verfolgung
10. FOE Auswahl
11. Konturklassifikation
12. Fahrbahnverfolgung
13. Fahrbahnauswahl
6(b) Eigenzustandsmodifikation
14. Gruppierung
15. Objektinitialisierung
FOR jede generelle Objekthypothese $s(k)$
16. OBJ.VERFOLGUNG($s^*(k)$, Neue_Objekte(k))
17. Objektauswahl

Bild 2.1.3: Der Analysefluß im System

2.1.5 Zur modellbasierten 3-D Bewegungserfassung

In unserem Ansatz haben wir uns für eine modellbasierte 3-D Bewegungserfassung entschieden in der die pixelbasierte Bewegungsschätzung und eine 2-D Segmentverfolgung nur untergeordnete Plätze einnehmen.

Probleme der pixelbasierten Verschiebung

In einer praktischen Untersuchung von pixelbasierten Bewegungsdetektoren (vgl. Abschnitt 7.1) hat sich gezeigt, daß diese Methoden zur Lösung unseres Problems bei kleinen Objekten und bei Eigenbewegung aus verschiedenen Gründen nur bedingt eingesetzt werden können. Neben schwacher Qualität und Stabilität der Ergebnisse ist auch der zeitliche Aufwand zu groß.

Es sollen hier drei Probleme der Verschiebungsschätzung im Bild angesprochen werden: die objektüberschreitende Glättung des optischen Flußes das *Kameranicken* als eigentliche Objektbewegung im Bild und das *Aperturproblem*.

In Bild 2.1.4 (a) ist ein Bild einer Bildfolge dargestellt, in der eine Relativbewegung der Fahrbahnmarkierung sowie des vorherfahrenden PKW-s festzustellen ist. Bild 2.1.4 (b) zeigt eine fehlerhafte Glättung eines optischen Flußes, die ohne Kenntnis von Objektkanten durchgeführt wurde. Das Ergebnis ist eine sehr vage Darstellung von bewegten und unbewegten Regionen, in der wichtige Objektkanten verschwunden sind.

In Bildfolgen, die aus einem fahrenden Fahrzeug aufgenommen wurden, haben wir es mit Relativbewegungen zu tun. Neben translativer Eigenbewegung des Fahrzeuges, die bekannt sein kann, gibt es vor allem das unbekanntes *Kameranicken* sowie seitliche Überdrehung des Fahrzeuges. Beide Schwierigkeiten der Messung werden durch Fahrbahnwellen und Kurvenneigungen verursacht. Dadurch sind die Verschiebungsvektoren in längerer Bildfolge nicht stabil genug, um über sie die 3-D Objektattribute zu erkennen und zu stabilisieren. Die Länge der Verfolgung von Bildmerkmalen, die zu einer 2-D Bewegungsschätzung herangezogen werden, ist damit begrenzt durch *Kameranicken*. Für die Bildfolge aus Bild 2.1.4(a) wurde ein gradientenbasierter optischer Fluß berechnet (Merkmale haben die Form einzelner Pixel). Die negativen Werte der vertikalen Komponente (entlang der vertikalen Bildachse) nach 25 Bildern ist in Bild 2.1.4(c) dargestellt in Form eines Grauwertbildes. Der Grauwert eines Pixels ist proportional zur Länge der negativen y -Verschiebungskomponente dieses Pixels. Man kann feststellen, daß die vertikalen Verschiebungen von Fahrbahnmarkierungen in derselben Größenordnung liegen, wie die Verschiebungen im Bereich des Vorgängerautos. Dies entspricht nicht den unterschiedlichen relativen Geschwindigkeiten dieser beiden Objekte.

Probleme der Merkmalsverfolgung im Bild

Abhängig von der Auswahl der Merkmale, für die der optische Fluß berechnet wird, können beide Komponenten oder nur eine Komponente des Verschiebungsvektors (entlang des Intensitätsgradienten) geschätzt werden. Es ist bekannt, daß für eine Ecke im Bild ein voller Bewegungsvektor ermittelbar ist, daß aber entlang einer linearen Kante nur die Verschiebungskomponente vertikal zur Kante geschätzt werden kann (vgl. Bild 2.1.5).

Die Verfolgung von geschlossenen Konturen leidet nicht unter dem Aperturproblem. Schwierigkeiten dieses Ansatzes entstehen durch die relativ hohe Anzahl der Segmente im Bild und durch die überwiegend geringe Relativbewegung der bewegten Objekte im Bild. Optimale Verfahren zur Korrespondenzauflösung verursachen einen erheblichen Aufwand und die schnelleren suboptimalen Lösungen sind unter vorgegebenen Voraussetzungen nicht robust genug.

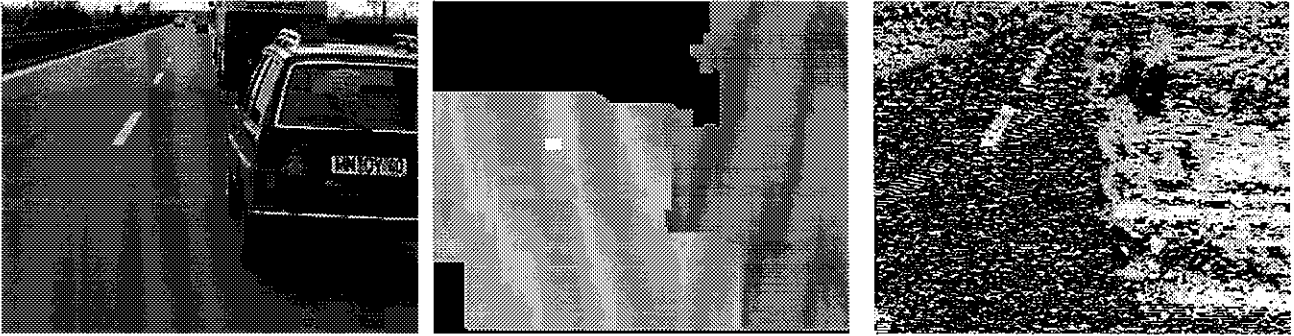


Bild 2.1.4: Das 25. Bild einer monokularen Bildfolge (links); das Beispiel einer überzogenen Glättung des optischen Flusses (die negative vertikale Verschiebung wird kodiert durch Grauwerte) (mitte); ähnliche Verschiebung im Bild von Fahrbahnmarkierung und bewegtes Objekt (rechts)

2.2 Das Modell des adaptiven Ansatzes

Der in dieser Arbeit vorgeschlagene adaptive Erkennungsansatz kann wie folgt zusammengefaßt werden (Bild 2.2.1):

1. dynamische Systemmodellierung

Die zu erkennende Größe (bzw. das Objekt) wird als ein *dynamisches System* modelliert, dessen Auswirkungen im Bild detektiert werden können (als *Messung*), aber dessen unbekannter *Zustand* zu ermitteln ist.

2. die Initialisierung

Eine gewisse Anzahl von schritthaltenden *Einzelmessungen* $M_j(k)$, ($k = 0, \dots, i$), die potentiell als Auswirkungen eines Objektes *Objekt_j* im Bild zu beobachten sind, führt zur Initialisierung eines hypothetischen Zustandes $Z_j(i)$.

3. die Verfolgung

Ein paralleles Verfolgen von vorher initialisierten, mehreren *konkurrierenden Hypothesen* (*Zuständen*) bestehend aus einer *rekursiven Schätzung* der damit verbundenen Zustände $Z_j(i)$ über die Zeit, unter Einbeziehung der schritthaltenden *Einzelmessungen* $M_j(i)$.

4. der Konsistenztest

Die *Auflösung* von Konkurrenzen zwischen Hypothesen, sobald eine sichere Beurteilung möglich ist.

5. der Erkennungstest

Die Auswahl von *erkannten Objekten* aus der Menge der verfolgten Hypothesen.

Im weiteren werden die oben erwähnten Punkte im einzelnen erläutert.

2.2.1 Ein dynamisches System

Ein lineares dynamisches System

Gegeben sei ein dynamisches System in der Form:

$$\mathbf{s}(t) = \mathbf{A}(t) \mathbf{s}(t_0) + \mathbf{B}(t) \mathbf{u}(t) \quad (2.2.1)$$

$$\mathbf{m}(t) = \mathbf{C}(t) \mathbf{s}(t) \quad (2.2.2)$$

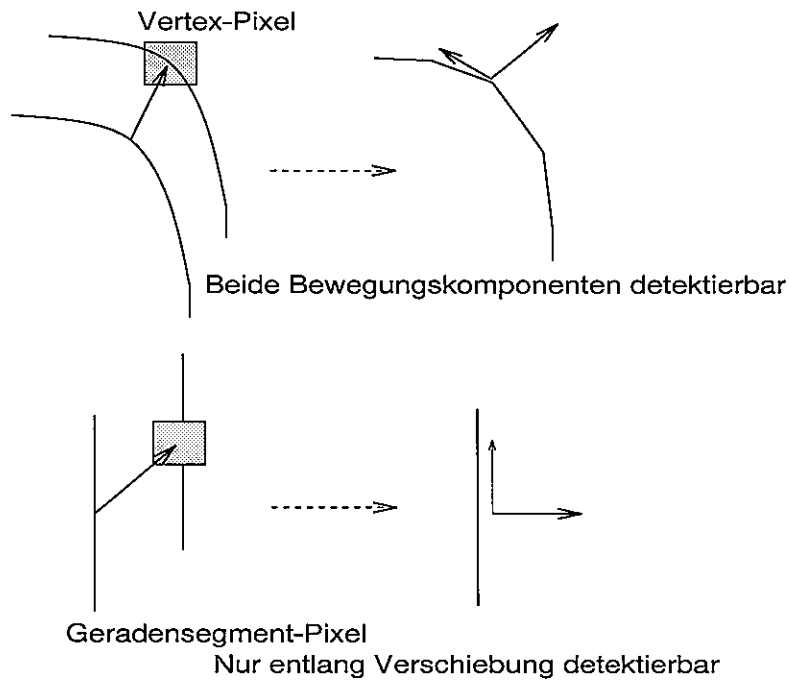


Bild 2.1.5: Das Aperturproblem.

wobei sind: \mathbf{s} der n -dimensionale Zustandsvektor, \mathbf{u} der p -gliedrige bekannte Stellvektor und \mathbf{m} der m -dimensionale Vektor der Ausgangsgrößen. Die *Beobachtungsaufgabe* besteht darin, eine Zustandsschätzung $\mathbf{s}^+(t)$ des dynamischen Systems $\mathbf{s}(t)$ auf Grund von Messungen der Ausgangsgröße $\mathbf{m}(t)$ und der Eingangsgröße $\mathbf{u}(t)$ vorzunehmen.

Die Lösung der Gleichung 2.2.1 hat die allgemeine Form des *einfachen Beobachters*:

$$\mathbf{s}(t) = \Phi(t, t_0) \mathbf{s}(t_0) + \int_{t_0}^t \Phi(t, \tau) \mathbf{B}(\tau) \mathbf{u}(\tau) d\tau \quad (2.2.3)$$

Dabei ist $\Phi(t, t_0)$ die Transitionsmatrix zu $\mathbf{A}(t)$. Der Nachteil des einfachen Beobachters liegt darin, daß das zeitliche Verhalten des Schätzfehlers

$$\bar{\mathbf{s}}(t) = \mathbf{s}(t) - \mathbf{s}^+(t)$$

ausschließlich von der Dynamik des beobachteten Systems abhängt und insofern nicht beeinflussbar ist. Dieser Nachteil wird behoben indem das Modell um die Ausgangsgleichung

$$\mathbf{m}^+(t) = \mathbf{C}(t) \mathbf{s}^+(t)$$

ergänzt wird. Die Differenz $\mathbf{m}(t) - \mathbf{m}^+(t)$ wird zur Verbesserung des Schätzwertes $\mathbf{s}^+(t)$ zurückgeführt. Die Beobachterform ist dann:

$$\frac{d}{dt} \mathbf{s}^+(t) = \mathbf{A}(t) \mathbf{s}^+(t) + \mathbf{B}(t) \mathbf{u}(t) + \mathbf{K}(t) \{ \mathbf{m}(t) - \mathbf{C}(t) \mathbf{s}^+(t) \} \quad (2.2.4)$$

Um das Verhalten des Schätzfehlers zu untersuchen bilden wir:

$$\frac{d}{dt} \bar{\mathbf{s}}(t) = \mathbf{A}(t) \bar{\mathbf{s}}(t) - \mathbf{K}(t) \{ \mathbf{C}(t) \mathbf{s}(t) - \mathbf{C}(t) \mathbf{s}^+(t) \} \quad (2.2.5)$$

$$\frac{d}{dt} \bar{\mathbf{s}}(t) = \{ \mathbf{A}(t) - \mathbf{K}(t) \mathbf{C}(t) \} \bar{\mathbf{s}}(t) \quad (2.2.6)$$

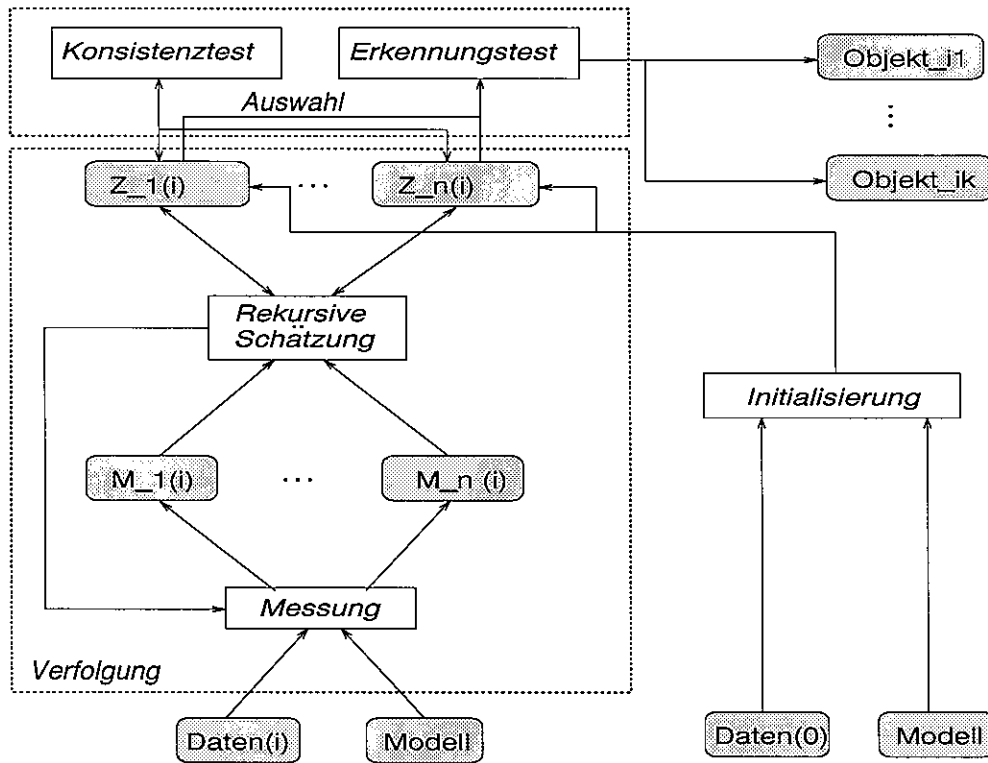


Bild 2.2.1: Die Struktur des adaptiven Erkennungsansatzes

Jetzt ist es möglich, die Dynamik des Beobachters durch geeignete Wahl der Verstärkungsmatrix $\mathbf{K}(t)$ so einzurichten, daß der Beobachtungsfehler $\vec{s}(t)$ genügend schnell abklingt. Leider ist es nicht trivial, zu einer Strecke n -ter Ordnung mit m Ausgangsgrößen sämtliche $n \times m$ Verstärkungsgrade $K_{ij}(t)$ des Beobachters für alle t adäquat einzustellen.

Ein diskretes lineares System

Der Fall *diskreter* Zeit ist bei der Beobachtung mathematisch einfacher, weil Ableitungen durch Differenzen und Integrale durch Summen ersetzt werden. Anstatt $\mathbf{s}(t = k)$, $\mathbf{m}(t = k + 1)$ usw. schreiben wir im Falle eines zeitlich diskreten Systems: $\mathbf{s}(k)$, $\mathbf{m}(k + 1)$ usw. Der Zusammenhang zwischen den Werten des Zustandes \mathbf{s} an zwei aufeinanderfolgenden Testzeitpunkten wird durch die allgemeine Lösungsformel der Differentialgleichung beschrieben:

$$\mathbf{s}(k + 1) = \Phi(k + 1, k) \mathbf{s}(k) + \int_k^{k+1} \Phi(k + 1, \tau) \overline{\mathbf{B}}(\tau) \overline{\mathbf{u}}(\tau) d\tau \quad (2.2.7)$$

Es wird die folgende Bezeichnung eingeführt: $\Phi(k + 1, k) = \mathbf{A}(k)$. Als Transitionsmatrix $\mathbf{A}(k)$ ist stets regulär. Zur Aufbringung der Stellgröße sind zwei Verfahren üblich:

- impulsförmige Stellgröße – $\mathbf{u}(t) = \overline{\mathbf{u}}(k) \delta(t - k - \Delta t)$

Bei impulsförmiger Stellgröße wird die Ausblendeigenschaft der Deltafunktion beim Integral benutzt:

$$\Phi(k + 1, \tau) \overline{\mathbf{B}}(k) \overline{\mathbf{u}}(k) = \mathbf{B}(k) \mathbf{u}(k) \quad (2.2.8)$$

- stückweise konstante Stellgröße – $\mathbf{u}(t) = \mathbf{u}(k)$ für $k \leq t \leq k + 1$

Bei stückweiser konstanter Stellgröße kann $\bar{u}(k)$ aus dem Integral herausgezogen werden und es wird definiert:

$$\int_k^{k+1} \Phi(k+1, \tau) \bar{B}(\tau) d\tau = \mathbf{B}(k) \quad (2.2.9)$$

Gemäß diesen Vereinbarungen geht das kontinuierliche dynamische System über in die *diskrete* Version:

$$\mathbf{s}(k+1) = \mathbf{A}(k) \mathbf{s}(k) + \mathbf{B}(k) \mathbf{u}(k) \quad (2.2.10)$$

$$\mathbf{m}(k) = \mathbf{C}(k) \mathbf{s}(k) \quad (2.2.11)$$

Die allgemeine Lösung der Vektordifferenzgleichung hat die Form:

$$\mathbf{s}(k) = \Phi(k, k_0) \mathbf{s}(k_0) + \sum_{\kappa=k_0}^{k-1} \Phi(k, \kappa+1) \mathbf{B}(\kappa) \mathbf{u}(\kappa), \quad k_0 < k$$

Die Eingangsgröße $\mathbf{u}(k)$ erzeugt den erzwungenen Lösungsanteil, der sich dem freien durch die Anfangsbedingung $\mathbf{s}(k_0)$ hervorgerufenen Lösungsanteil additiv überlagert. Wir nehmen an, daß die Wirkung der bekannten Stellfolge $\mathbf{u}(k_0), \mathbf{u}(k_0+1), \dots, \mathbf{u}(k-1)$ auf $\mathbf{s}(k)$ und somit auf $\mathbf{m}(k)$ gesondert berechnet und von den Meßgrößen abgezogen wird. Es genügt also die Betrachtung nur des *freien* Systems:

$$\mathbf{s}(k+1) = \mathbf{A}(k) \mathbf{s}(k) \quad (2.2.12)$$

$$\mathbf{m}(k) = \mathbf{C}(k) \mathbf{s}(k) \quad (2.2.13)$$

Ein stochastisch gestörtes System

Die *Filteraufgabe* ist eine Verallgemeinerung der *Beobachtungsaufgabe*. Bei der Filterung werden die stochastischen Störgrößen und Meßfehler explizit berücksichtigt. Das Filter hat die gleiche Struktur wie der entsprechende Beobachter. Der Unterschied besteht darin, daß die Verstärkungsgrade des Filters optimal bezüglich der gegebenen statistischen Eigenschaften der stochastischen Stör- und Meßgeräusche sind, während die Verstärkungsgrade des Beobachters nach anderen Gesichtspunkten ausgewählt werden können.

Wir nehmen weiter an, daß der Einfluß der bekannten Eingangsgrößen bereits eliminiert sei. Das mathematische Modell des beobachteten Systems lautet:

$$\mathbf{s}(k+1) = \mathbf{A}(k) \mathbf{s}(k) + \mathbf{v}(k); \quad k \geq k_0 \quad (2.2.14)$$

$$\mathbf{m}(k) = \mathbf{C}(k) \mathbf{s}(k) + \mathbf{w}(k) \quad (2.2.15)$$

wobei $\mathbf{v}(k)$ das Systemrauschen, $\mathbf{w}(k)$ der Meßfehler ist. Der unzugängliche Zustandsvektor $\mathbf{s}(k)$ ist n -dimensional, sein Anfangswert $\mathbf{s}(k_0)$ sei ein Zufallsvektor mit dem Erwartungswert Null und gegebener Kovarianzmatrix $\mathbf{P}(k_0)$:

$$E\left\{ \left[\mathbf{s}(k_0) - E\{\mathbf{s}(k_0)\} \right] \left[\mathbf{s}(k_0) - E\{\mathbf{s}(k_0)\} \right]^T \right\} = \mathbf{P}(k_0)$$

Die Meßgröße $\mathbf{m}(k)$ ist m -dimensional. Die Störgröße $\mathbf{v}(k)$ und der Meßfehler $\mathbf{w}(k)$ sind vektorielle weiße Zufallsprozesse in diskreter Zeit, mit gegebenen symmetrischen, positiv semidefiniten Kovarianzmatrizen:

$$E\{\mathbf{v}(k)\} = \mathbf{0} \quad E\{\mathbf{v}(k) \mathbf{v}^T(\kappa)\} = \mathbf{Q}(k) \delta_{k\kappa}$$

$$E\{\mathbf{w}(k)\} = \mathbf{0} \quad E\{\mathbf{w}(k) \mathbf{w}^T(\kappa)\} = \mathbf{R}(k) \delta_{k\kappa}$$

($\delta_{k\kappa}$ - das Kronecker-Delta: $\delta_{k\kappa} = 1$ für $k = \kappa$, $\delta_{k\kappa} = 0$ für $k \neq \kappa$)

Der Anfangszustand, der Stör- und der Meßfehler seien gegenseitig unkorreliert:

$$E\{\mathbf{s}(k_0)\mathbf{v}^T(k)\} = \mathbf{0}, \quad E\{\mathbf{s}(k_0)\mathbf{w}^T(k)\} = \mathbf{0}, \quad E\{\mathbf{v}(k)\mathbf{w}^T(\kappa)\} = \mathbf{0}$$

Zusätzlich wird angenommen, daß über den Anfangszustand des Systems keinerlei a-priori-Kenntnisse vorliegen ($\mathbf{P}(k_0) \rightarrow \infty$) und daß die Kovarianzmatrix der Meßfehler $\mathbf{R}(k)$ regulär ist.

2.2.2 Die Initialisierung

Bei fortschreitender Beobachtung kommt in jedem folgenden Tastzeitpunkt k ein neuer Satz von Messungen $\mathbf{m}(k)$ hinzu. Es wird solange abgewartet, bis die Gesamtzahl der Einzelmesswerte $m_j(k)$, ($j = 1, \dots, |\mathbf{m}|$; $k = 0, \dots, i$) größer oder gleich der Zahl $n = |\mathbf{s}|$ der Zustandsvariablen ist. Dann kann der Wert von $\mathbf{s}(i)$ mittels einer Gauschen Ausgleichsrechnung ermittelt werden.

Gegeben sind die Meßwerte $\mathbf{m}(0), \mathbf{m}(1), \dots, \mathbf{m}(i)$; gesucht ist $\mathbf{s}(i)$. Wir eliminieren die Zustandsvektoren $\mathbf{s}(0) \dots$ indem wir sie durch den gesuchten Zustand $\mathbf{s}(i)$ ausdrücken:

$$\mathbf{s}(1) = \Phi(i, k) \mathbf{s}(k); \quad \mathbf{s}(k) = \Phi(k, i) \mathbf{s}(i) \quad (2.2.16)$$

Es ergibt sich ein Gleichungssystem:

$$\mathbf{m}(0) = \mathbf{C}(0) \mathbf{s}(0) = \mathbf{C}(0) \Phi(0, i) \mathbf{s}(i) \quad (2.2.17)$$

$$\mathbf{m}(1) = \mathbf{C}(1) \mathbf{s}(1) = \mathbf{C}(1) \Phi(1, i) \mathbf{s}(i) \quad (2.2.18)$$

$$\dots = \dots \quad (2.2.19)$$

$$\mathbf{m}(i) = \mathbf{C}(i) \mathbf{s}(i) = \mathbf{C}(i) \Phi(i, i) \mathbf{s}(i) \quad (2.2.20)$$

Eine äquivalente Matrix-Gleichungsform ist

$$\mathbf{z}(i, 0) = \mathbf{D}(i, 0) \mathbf{s}(i) \quad (2.2.21)$$

wobei ist \mathbf{z} der Vektor der Meßvektoren $\mathbf{m}(k)$ und \mathbf{D} der Vektor der Zeilen $\mathbf{C} \Phi$. Der Vektor \mathbf{z} hat die Dimension $(i-1) \times |\mathbf{m}|$. Wenn diese Dimension gerade gleich n ist (der Ordnung des Systems - der Anzahl von unbekanntem Zustandsvariablen $s_j(i)$) und außerdem \mathbf{D} regulär ist, dann läßt sich das Gleichungssystem in eindeutiger Weise nach $\mathbf{s}(i)$ auflösen. Sind mehr als n skalare Messungen akkumuliert, dann kann mit Hilfe der klassischen *Ausgleichsrechnung* von Gauß (*Methode der kleinsten Quadrate* [BS75]) ein Schätzwert \mathbf{s}^+ berechnet werden.

Im Falle eines stochastisch gestörten Systems führt man die Kovarianzmatrix \mathbf{S} ein, die durch $\mathbf{Q}(\kappa)$ und $\mathbf{R}(\kappa)$ ausgedrückt werden kann ($k_0 \leq \kappa \leq k$). Mit den so ermittelten \mathbf{D} und \mathbf{S} kann der Schätzwert $\mathbf{s}^+(k)$ nach der sogenannten *Gauß-Markoffschen Formel* ([BS75]):

$$\mathbf{s}^+ = (\mathbf{D}^T \mathbf{S}^{-1} \mathbf{D})^{-1} \mathbf{D}^T \mathbf{S}^{-1} \mathbf{z} \quad (2.2.22)$$

erstmalig bestimmt werden. Gleichzeitig ist der erste Wert von $\mathbf{P}^+(k)$ zu berechnen:

$$\mathbf{P}^+ = E\{\mathbf{s}^+(\mathbf{s}^+)^T\} = (\mathbf{D}^T \mathbf{S}^{-1} \mathbf{D})^{-1} \quad (2.2.23)$$

2.2.3 Die rekursive Schätzung

Wir wollen vermeiden, den mühsamen Vorgang einer Ausgleichsrechnung in jedem Intervall zu wiederholen. Für eine schritthaltende Beobachtung würde sich besonders gut eine *rekursive* Lösung eignen, die es erlaubt, den alten Schätzwert $\mathbf{s}^+(k)$ nach Eintreffen einer zusätzlichen Beobachtung $\mathbf{m}(k+1)$ durch geringe Korrekturen zu erneuern. Bereits Gauß hat eine rekursive Lösung angegeben, die es erlaubt, den alten Schätzwert $\mathbf{s}(k)$ nach Eintreffen einer zusätzlichen Beobachtung $y(k+1)$ durch geringe Korrekturen zu erneuern. Plackett hat diese Methode auf ein m -Tupel gleichzeitiger Messungen \mathbf{m} erweitert.

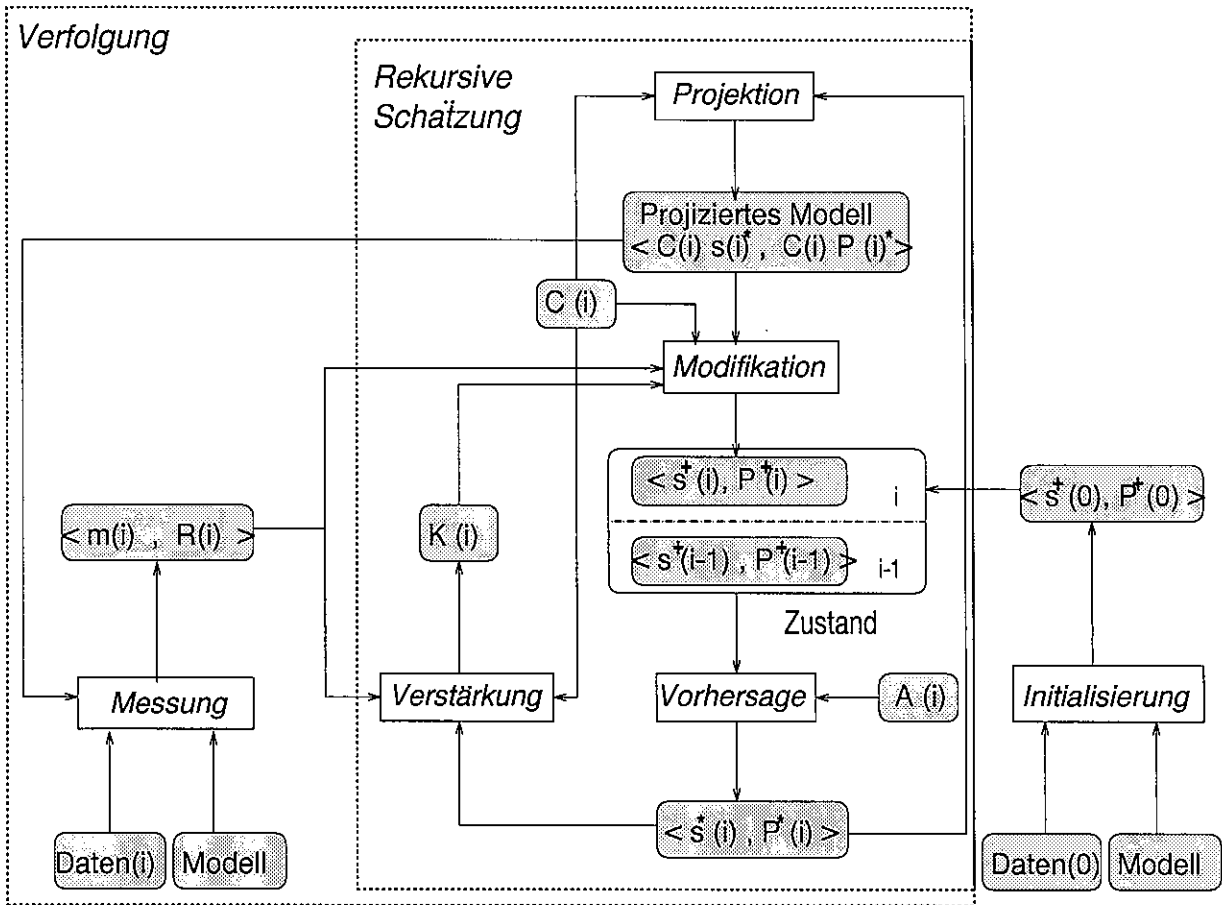


Bild 2.2.2: Die rekursive Schätzung eines einzelnen Zustandes während der Verfolgung

Das Kalman Filter

Der Algorithmus für die *rekursive Gauß-Markoffsche* Filterung des zeitlich diskreten, stochastisch gestörten Systems 2.2.14 ist identisch mit dem *Kalman Filter* [BS75] und wird in Bild 2.2.3 definiert. Ein graphischer Überblick der Einbettung des Kalman Filters in die Objektverfolgung folgt in Bild 2.2.2.

Während der Schätzung ist zu entscheiden, wie sicher die neue Messung $\mu(k)$ und wie sicher der extrapolierte Vektor $s^+(k)$ ist. Die Sicherheit eines Vektorwertes wird durch seine Kovarianzmatrix ausgedrückt. Aus den Kovarianzmatrizen $P^*(k)$, $R^*(k)$ und indirekt aus $Q(k)$ wird die Kalman-Verstärkungsmatrix-Matrix $K(k)$ berechnet.

Das Kalman Filter ist für vektorielle Gauß-Markov Prozesse das optimale Filter überhaupt, für alle anderen Prozesse das optimale lineare Filter ([BS75]). Optimal bedeutet hier, daß der Fehler $\bar{s}(k)$ zwischen dem tatsächlichen Zustandswert und dem geschätzten Zustandswert minimiert wird und es kein anderes Verfahren gibt, daß einen noch kleineren Fehler liefert als das lineare Kalman Filter.

Das Erweiterte Kalman Filter

Falls mindestens eine von den Übergangsfunktionen, Zustandsübergangsfunktion und Projektionsfunktion, nicht linear ist, können u.U. keine Matrizen A und C definiert werden. Stattdessen ist das folgende zeitlich diskrete, stochastisch gestörte, dynamische System gegeben:

$$s(k + 1) = f(s(k)) + v(k), \quad E\{s(k_0)\} = \xi \tag{2.2.24}$$

1. Anfangsbedingungen: Nachdem die Gesamtzahl der Einzelmeßwerte $m_j(k)$, $k = 1, \dots, k_0$ größer oder gleich der Zahl n der Zustandsvariablen ist wird der erste Schätzwert $\mathbf{s}^+(k_0)$ und der erste Wert der Kovarianzmatrix $\mathbf{P}^+(k_0)$ bestimmt. Annahmen über den Anfangszustand: erwartungstreu (Biasfreiheit) – $E\{\mathbf{s}(k_0)\} = \mathbf{o}$ und unendliche Streuung – $\mathbf{P}^{-1}(k_0) = \mathbf{o}$. Anschliessend gehe zu Punkt 6.
2. Zu jedem folgenden Zeitpunkt $k > k_0$ wird eine Messung $\mathbf{m}(k)$ und die Kovarianzmatrix $\mathbf{R}(k)$ des Meßfehlers ermittelt. Zum Zeitpunkt k muß auch die Kovarianzmatrix des Systemrauschens $\mathbf{Q}(k)$ verfügbar sein.
3. Die Schätzung der aktuellen Kalman-Verstärkungsmatrix \mathbf{K} : $\mathbf{K}(k) = \mathbf{P}^*(k) \mathbf{C}^T(k) \{ \mathbf{C}(k) \mathbf{P}^*(k) \mathbf{C}^T(k) + \mathbf{R}(k) \}^{-1}$
4. Die Modifikation (Schätzung, Innovation) des Zustandes: $\mathbf{s}^+(k) = \mathbf{s}^*(k) + \mathbf{K}(k) \{ \mathbf{m}(k) - \mathbf{C}(k) \mathbf{s}^*(k) \}$
5. Die Modifikation der Schätzfehler-Kovarianzmatrix $\bar{\mathbf{P}}$: $\mathbf{P}^+(k) = \mathbf{P}^*(k) - \mathbf{K}(k) \mathbf{C}(k) \mathbf{P}^*(k) - \text{a-posteriori Varianz}$
6. Die Vorhersage (Prädiktion) des nächsten Zustandes: $\mathbf{s}^*(k+1) = \mathbf{A}(k) \mathbf{s}^+(k)$
7. Die Vorhersage (Prädiktion) der nächsten Kovarianzmatrix \mathbf{P} des Schätzfehlers: $\mathbf{P}^*(k+1) = \mathbf{A}(k) \mathbf{P}^+(k) \mathbf{A}^T(k) + \mathbf{Q}(k) - \text{a-priori Varianz}$
8. Mit $k \leftarrow k+1$ wiederhole ab Punkt 2.

Bild 2.2.3: Der Kalman Filter Algorithmus

$$\mathbf{m}(k) = \mathbf{h}(\mathbf{s}(k)) + \mathbf{w}(k) \quad (2.2.25)$$

wobei \mathbf{f}, \mathbf{g} Funktionen sind, die vom aktuellen Zustand abhängen.

Eine Modifikation des Kalman Filters, genannt das *Erweiterte Kalman Filter* (EKF), erhält man durch Linearisierung der Übergangsfunktionen $f(\cdot)$ und $h(\cdot)$ ([WRCG88]). Die Zustandsübergangsfunktion $f(\cdot)$ wird um den geschätzten Zustandswert herum linearisiert und ihre *Jacobimatrix*

$$\mathbf{F}(k) = \left. \frac{\delta \mathbf{f}}{\delta \mathbf{s}} \right|_{\mathbf{s}^+(k)} \quad (2.2.26)$$

wird zur Prädiktion der nächsten Kovarianzmatrix, anstatt der Matrix $\mathbf{A}(k)$ im linearen Kalman Filter verwendet. Die Projektionsfunktion $h(\cdot)$ wird um den vorhergesagten Zustandswert herum linearisiert und ihre *Jacobimatrix*

$$\mathbf{H}(k) = \left. \frac{\delta \mathbf{h}}{\delta \mathbf{s}} \right|_{\mathbf{s}^*(k)} \quad (2.2.27)$$

wird zur Berechnung der Kalmanverstärkung \mathbf{K} verwendet.

Durch Linearisierung von $\mathbf{f}(\mathbf{s}(k))$ in der Umgebung von $\mathbf{s}^+(k)$ und von $\mathbf{h}(\mathbf{s}(k))$ in $\mathbf{s}^*(k)$, sowie mit der Definition $\mathbf{e}(k) = \mathbf{s}(k) - \mathbf{s}^*(k)$ erhält man ein lineares System bezüglich des Fehlers $\mathbf{e}(k)$:

$$\mathbf{e}(k+1) = \mathbf{s}(k+1) - \mathbf{s}^*(k+1) = \mathbf{F}(k)(\mathbf{s}(k) - \mathbf{s}^+(k)) + \mathbf{v}(k) = \quad (2.2.28)$$

$$= \mathbf{F}(k)\mathbf{e}(k) - \mathbf{F}(k)\mathbf{e}^+(k) + \mathbf{v}(k) \quad (2.2.29)$$

$$\mathbf{g}(k+1) = \mathbf{m}(k+1) - \mathbf{h}\mathbf{s}^*(k+1) = \mathbf{H}(k+1)\mathbf{e}(k+1) + \mathbf{w}(k+1) \quad (2.2.30)$$

Daraus ergibt sich folgende Fehlerschätzung:

$$\mathbf{e}(k+1) = \mathbf{e}^*(k+1) + \mathbf{K}(k+1)[\mathbf{g}(k+1) - \mathbf{H}(k+1) \mathbf{e}^*(k+1)] = \quad (2.2.31)$$

$$= \mathbf{K}(k+1)[\mathbf{m}(k+1) - \mathbf{h} \mathbf{s}^*(k+1)] \quad (2.2.32)$$

$$\mathbf{s}^+(k+1) = \mathbf{K}(k+1)[\mathbf{m}(k+1) - \mathbf{h}(\mathbf{s}^*(k+1))] \quad (2.2.33)$$

wobei sind:

$$\mathbf{Q}(k) = E\{\mathbf{v}(k)\mathbf{v}^T(k)\}; \quad (2.2.34)$$

$$\mathbf{R}(k+1) = E\{\mathbf{w}(k)\mathbf{w}^T(k)\}; \quad (2.2.35)$$

Der Algorithmus des erweiterten Kalman Filters ist Wortgleich mit einem Algorithmus des Kalman Filters, in dem $\mathbf{A}(k)$ durch $\mathbf{F}(k)$ ersetzt wurde und $\mathbf{C}(k)$ durch $\mathbf{H}(k)$ ersetzt wurde.

Um den mühsamen Weg einer Matrixinvertierung bei der Berechnung der Kalmanverstärkung zu vermeiden, wurde ein *EKF mit sequentieller Innovation* implementiert [Wue91]. Durch eine getrennte Betrachtung der unkorrelierten Zeilen wird die Matrixumkehrung durch eine Iteration von sequentiellen Divisionen (bei zeilenweisem Vorgehen) ersetzt (vgl. Abschnitt 5.4).

2.2.4 Die Messung

Die Messung bedeutet in unseren Anwendungen des adaptiven Ansatzes den Einsatz von problemangepaßten Methoden (bzw. model-basierten Verfahren) zur Detektion von entsprechenden Bildmerkmalen, Bildstrukturen, Fahrbahnparametern oder Objektparametern. Insofern ist unser "Meßgerät" von besonderer Natur. Für eine erfolgreiche Stabilisierung ist es entscheidend

1. eine problemangepaßte Methode zu finden, mit der die Meßvarianzen in $\mathbf{P}(k)$ zuverlässig bestimmt werden können,
2. eine entsprechenden Relation zwischen dem besten Meßwert (also mit niedrigster Varianz) und dem Wert des Systemrauschen in $\mathbf{Q}(k)$ einzuhalten.

Um den Bedingungen der Biasfreiheit der Messungen zu genügen, sollten die Einzelmessungen unabhängig voneinander sein. Diese Bedingung wird oftmals in Anwendungen des Kalman Filters zur Objektverfolgung unterlaufen. In solchen Ansätzen wird versucht eine fast optimale Hypothese zu initialisieren (z.B. [KDN93]) und diese dann im Bild zu verfolgen. Die anschließenden Messungen und, was wichtiger ist, die Bewertungen und Auswahl der konkurrierenden Hypothesen sind direkt von der Hypothese abhängig – wie gut die Messung die Erwartungen der Hypothese widerspiegelt. Dieser Ansatz kann als pure *Objektverfolgung* bezeichnet werden.

Der in dieser Arbeit verfolgte Ansatz sieht eine hypothesenunabhängige Bewertung der Messung vor. Durch die Hypothese wird lediglich der Suchbereich im Bild oder Raum eingeschränkt. Die Bewertung spiegelt aber nur den Erfüllungsgrad der Modellerwartung wieder und die Messung verläuft im vorgegebenem Suchbereich unabhängig von der Hypothese. Dadurch bleibt die statistische Unabhängigkeit der Messungen gewahrt. Der adaptive Ansatz kann auch als *Erkennung durch Verfolgung* bezeichnet werden.

In dieser Arbeit werden je nach Anwendungsbereich zum Teil unterschiedliche Methoden zur Bestimmung der Varianz einer Messung entwickelt. Zum Beispiel für die Messung des Fluchtpunktes im Bild ist die Relation der Anzahl der Häufungspunkte im Suchbereich zur gesamten Anzahl der Schnittpunkte der Fluchtpunktgeraden maßgebend für die Varianzbestimmung. Das Systemrauschen wird durch Defaultwerte modelliert. Gewöhnlich wird auch eine minimale und maximale Schranke für jede Meßvarianz eingeführt.

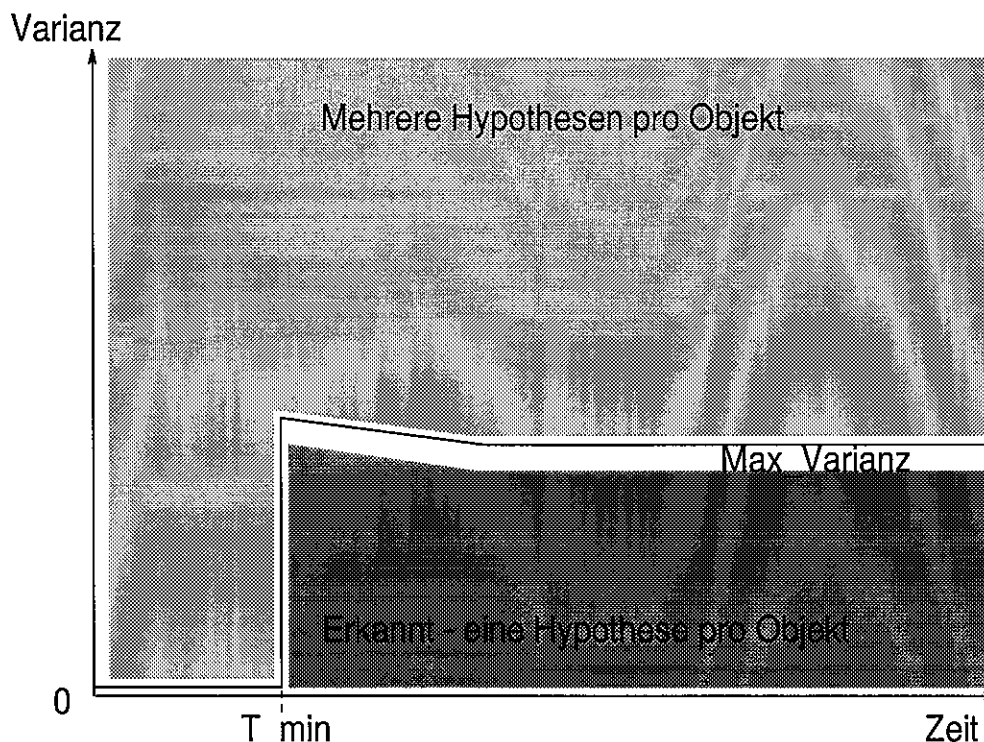


Bild 2.2.4: Die Bedingungen für die Verfolgung von potentiell inkonsistenten Hypothesen und den Auswahltest zur Erkennung.

2.2.5 Die Objektauswahl

Dieser Schritt besteht aus dem *Konsistenztest* zwischen Paaren von Hypothesen und aus dem *Erkennungstest* für jede nach dem Konsistenztest noch verbliebene Hypothese.

Die Bedingungen für den Konsistenztest werden definiert durch zwei folgende Parameter: die minimale Verfolgungszeit T_{min} und die *Maximum_Var* (Varianz) Schwelle. Falls die Verfolgungszeiten von zwei potentiell inkonsistenten Hypothesen beide länger sind als T_{min} oder eine von diesen Hypothesen sich bereits in der Erkennungsphase befindet (d.h. für die gilt daß die Verfolgungszeit $> T_{min}$ und ihre Varianz $< Maximum_Var$) dann findet der Konsistenztest zwischen beiden Hypothesen statt (Bild 2.2.4).

Eine Hypothese befindet sich entweder in ihrer *Verfolgungsphase* oder in einer von ihren drei *Erkennungsphasen*. Die Verfolgungsphase besteht dann, wenn die Verfolgungszeit der Hypothese kürzer als T_{min} ist oder ihre Varianz größer als *Maximum_Var* ist. Andererseits befindet sich die Hypothese in einer Erkennungsphase. Die drei Erkennungsphasen unterscheiden sich durch den Einsatz von unterschiedlichen Modellspezialisierungen. Die generelle Erkennungsphase charakterisiert sich durch die Modellfamilie *Erkennung_Form*. Dieses Modell wird angewendet solange die Varianzfunktion der Hypothese über der Varianzschwelle liegt. Nachdem sich die Hypothese weiter "befestigt hat" und die Varianz unter diese Schwelle gesunken ist das *Erkennung_Fein*-Modell wird angewendet. Die Anwendung der letzten Spezialisierung (das *Erkennung_Typ*-Modell) hängt von der Verfügbarkeit von zusätzlichen Rechenressourcen für sicher erkannte Einzelobjekte ab aber auch von der Größe dieser Objekte im Bild und von deren Wichtigkeit für den Beobachter (Bild 2.2.5). Die Lösung dieses Problems liegt außerhalb dieser Arbeit. In der später folgenden detaillierten 3-D Objekterkennung wird dieses Problem definiert und eine mögliche Lösung nur theoretisch vorgestellt.

Der allgemeine Fall der Objektauswahl kommt in dem Erkennungssystem nur für 3-D Ob-

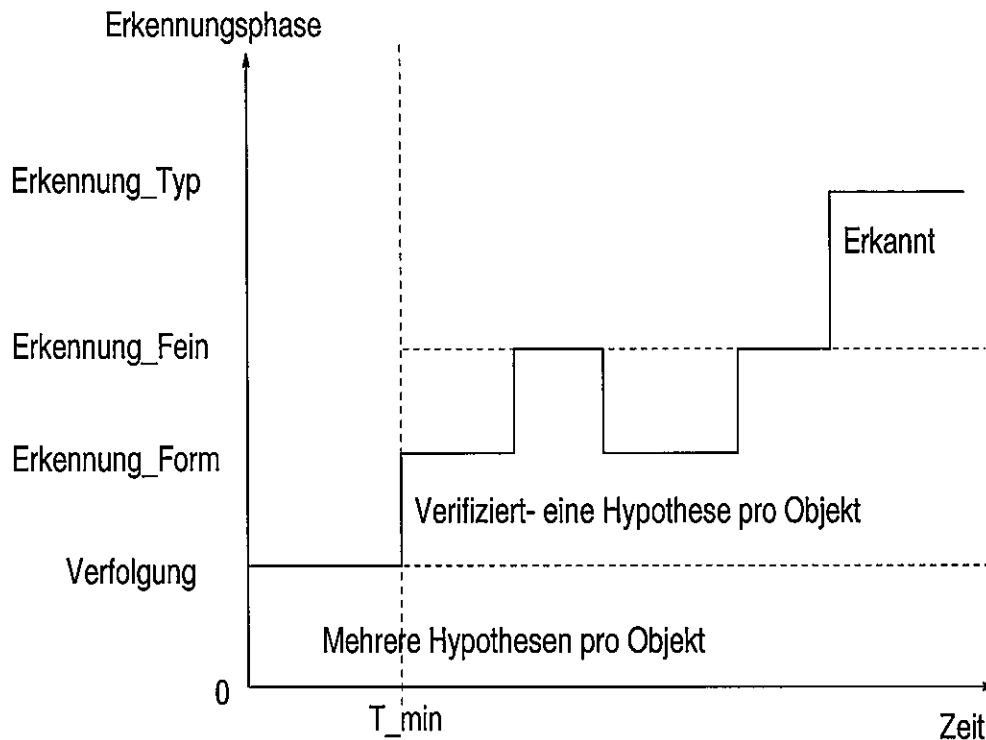


Bild 2.2.5: Die Phasen einer Hypothese.

jekthypothesen vor. Bei anderen Anwendungsstellen des adaptiven Ansatzes werden entweder keine konkurrierenden Hypothesen generiert oder es wird nur jeweils eine Hypothese gebraucht, d.h. alle verfolgten Hypothesen sind dann per Definition konkurrierend zueinander. Deswegen sprechen wir an manchen Stellen nur von einer *Verfolgung* und nicht von der *Erkennung* (das letzte bezeichnet einen Oberbegriff).

2.3 Anwendung des adaptiven Ansatzes

Wie bereits angedeutet wird der adaptive Ansatz in mehreren Verarbeitungsschritten des Systems verwendet. Im weiteren folgt eine generelle Übersicht dieser Anwendungsstellen.

2.3.1 Adaptives Differenzbild bei Eigenbewegung

Um die Nachteile einer perspektivischen Projektion in Standardverfahren zur Verschiebungsschätzung auszugleichen wurden z.B. die log-polar Transformation der Bildebene vorgeschlagen ([DK95]). Bei der Verschiebungsschätzung im Bild wird hier die Eigenbewegung und die momentane Transformation zwischen Fahrbahn und Kamera berücksichtigt. Die momentanen Abbildungsbedingungen zwischen Raum und Bild sind in dem *Eigenzustand* zum Zeitpunkt k zusammengefaßt. Neben der Kenntnis der Fahrbahnebene und der Projektionsparameter, werden hier auch das Kameranicken und die Eigengeschwindigkeiten detektiert (vergleiche dazu Abschnitt 4.5).

Das pixelbasierte Kalman Filter zur adaptiven Differenzmaske ist eine vereinfachte Form eines skalaren linearen Filters mit konstanten Werten der Verstärkung und Varianzen (vergleiche dazu Abschnitt 3.1.3):

- $s_{\text{Pixel}} = [\text{Grauwert}, \delta\text{Grauwert}]^T$

- $m_{Pixel} = [Grauwert]$
- $E_{Pixel} = \sigma_k$, falls ein von der Vorhersage sehr abweichender Grauwert oder $E_{Pixel} = \sigma_g$, falls ein ähnlicher Grauwert wie die Vorhersage

Dabei ist σ_k um eine Größenordnung kleiner als σ_g (z.B. $\sigma_k = 0.01$, $\sigma_g = 0.1$). Die Entscheidung darüber welcher Varianzwert genommen wird hängt mit einer vorgegebenen Schwelle zusammen. Die Tatsache, daß die Meßvarianz direkt von der Vorhersage abhängig ist, bedeutet eine Stabilisierung des Grauwertes um den Schätzwert herum oder einen zeitlich verzögerten Übergang auf einen neuen Schätzwert.

2.3.2 Konturverfolgung

Für die Verfolgung von einzelnen Bildkonturen in einer längeren Bildfolge (länger als 5 Bilder) kann ein lineares Kalman Filter eingesetzt werden. Der Zustandsvektor enthält die Positionen von markanten Punkten, die Verschiebungen der Punkte im Bild, die Länge der Kontur und ihre relative Veränderung. Es wird eine gewichtete Messung für jedes dieser Merkmale bestimmt, die aus bis zu 5 aufeinanderfolgenden Einzelmessungen besteht. Die Meßvarianzen entsprechen den Varianzen der Einzelmessungen in dieser kurzen Bildfolge.

2.3.3 Adaptive Fluchtpunkterkennung

Sowohl für den geometrischen Fluchtpunkt FP wie auch den dynamischen Fluchtpunkt FOE werden identische Filter eingesetzt. Hier wird die Beschreibung der Verfahren anhand des Fluchtpunktes FP gegeben. Der Zustandsvektor besteht aus der Lage und Verschiebung im Bild:

$$s_{FP}(k) = [x_{FP}(k), y_{FP}(k), \delta x(k), \delta y(k)]^T \quad (2.3.1)$$

mit den Varianzwerten:

$$P_{FP}(k) = [E_x(k), E_y(k), E_{\delta x}(k), E_{\delta y}(k)]^T \quad (2.3.2)$$

Der Meßvektor besteht aus den Lageparametern im Bild (vergleiche Bild 4.3.1) und ist äquivalent mit einem reduzierten Zustandsvektor:

$$m_{FP}(k) = [x_{FP}(k), y_{FP}(k)]^T \leftrightarrow s_{FP-R}(k) \quad (2.3.3)$$

mit den Meßvarianzen:

$$R_{FP}(k) = [R_x(k), R_y(k)]^T \quad (2.3.4)$$

Die Initialisierung eines Zustandes entspricht der FP -Messung im ersten Bild. Es sind bis zu drei konkurrierende FP -Hypothesen zugelassen.

Ein lineares Kalman Filter wird zur rekursiven Lageschätzung verwendet. Da diese Parameter als unkorreliert gelten, werden die Kalmanverstärkungen $K_R = [K_x(k), K_y(k)]^T$ durch einfache Divisionen berechnet, welche den Einfluß der neuen Messung auf den nächsten Schätzwert gewichten:

$$K_x(k) = \frac{E_x^+(k)}{E_x^+(k) + R_x(k)}; \quad K_y(k) = \frac{E_y^+(k)}{E_y^+(k) + R_y(k)} \quad (2.3.5)$$

Die *Modifizierung (Innovation)* $\mathbf{s}_R^+(k)$ der geschätzten Lageparameter und ihrer Varianzen erfolgt dann durch:

$$\mathbf{s}_R^+(k) = \mathbf{s}_R^*(k) + \mathbf{K}_R(k) ((\mathbf{m}(k) - \mathbf{s}_R^*(k))) \quad (2.3.6)$$

$$\mathbf{P}_R^+(k) = (1 - \mathbf{K}_R(k)) \mathbf{P}_R^*(k) \quad (2.3.7)$$

Anschließend werden synthetische Verschiebungswerte berechnet:

$$\delta x(k) = x_{FP}^+(k) - x_{FP}^+(k-1) \text{ und } \delta y(k) = y_{FP}^+(k) - y_{FP}^+(k-1) \quad (2.3.8)$$

Sie tragen dann zur Modifizierung der geschätzten Verschiebungen $\delta x^*(k), \delta y^*(k)$ bei, in einem ähnlichen Verfahren, wie es der Fall bei den Lageparametern ist. Für die Destimmung der Varianzen in $\mathbf{R}(k)$ ist die Anzahl der Häufungspunkte im gegebenen Bereich bei bekannter Anzahl von allen Schnittpunkten der Fluchtpunktgeraden maßgebend. Eine detaillierte Beschreibung der Fluchtpunkterkennung erfolgt im Abschnitt 4.1.

2.3.4 Adaptive Fahrbahnerkennung

Der Zustandsvektor einer Fahrbahnhypothese besteht aus der Breite W , der seitlichen Verschiebung zwischen der Mittellinie der Fahrbahn und der Kamera B (vergleiche Bild 4.4.4), sowie aus der translatorischen Geschwindigkeit in seitlicher Richtung δB und der Rotationsgeschwindigkeit ω_F , die indirekt die Fahrbahnkrümmung ausdrückt:

$$\mathbf{s}_F(k) = [W(k), B(k), \delta B(k), \omega_F]^T \quad (2.3.9)$$

Die individuelle Messung entspricht dem reduzierten Zustand und enthält die Breite und Lage:

$$\mu_F(k) = \mathbf{s}_{F-R}(k) = [W(k), B(k)]^T \quad (2.3.10)$$

Die abhängige Messung enthält entsprechende Parameter für seitliche Geschwindigkeit δB und Rotationsgeschwindigkeit der Fahrbahnkrümmung ω_F .

Es werden bis zu drei Fahrbahnypothesen parallel verfolgt. Genauso wie bei der Fluchtpunktverfolgung ist der Konsistenztest trivial, da nur eine Fahrbahn momentan im Bild existieren kann. Eine detaillierte Beschreibung der Fahrbahnerkennung erfolgt im Abschnitt 4.4.

2.3.5 Adaptive 3-D Objekterkennung

Die Anwendung der adaptiven Theorie findet im Modul zur 3-D Objekterkennung ihre volle Form wieder. Da dieses Modul Gegenstand des Kapitels 5 ist, wird in diesem Abschnitt nur eine allgemeine Beschreibung des 3-D Moduls als Anwendung des adaptiven Ansatzes gegeben.

Das Objektmodell

Jede Objekthypothese einer vorgegebenen Objektklasse wird durch ihren Zustandsvektor \mathbf{s} eindeutig beschrieben. Dieser Vektor besteht aus zwei Teilen: einem *dynamischen Zustandsvektor* \mathbf{s}^d und einem *Formvektor* ξ .

Im Allgemeinen gibt es sechs *Freiheitsgrade* der *Lage* eines starren Körpers (definiert durch den Ursprung seines Koordinatensystems) im 3-D Raum:

- drei Translationskomponenten – $\mathbf{T} = [T_X, T_Y, T_Z]$
- drei Rotationskomponenten um den eigenen Koordinatenursprung – $\Theta = [\Theta_X, \Theta_Y, \Theta_Z]$

Sei ein Objektpunkt $\mathbf{p}^o = [X^o, Y^o, Z^o]$ in Objektkoordinaten gegeben. Die Lage dieses Objektpunktes im 3-D Raum in Kamerakoordinaten ergibt sich aus einem Satz der momentanen Transformationsparameter:

$$\mathbf{p}^k(k) = \mathbf{R}(k)\mathbf{X}^o + \mathbf{T}(k) \quad (2.3.11)$$

wobei $\mathbf{R}(k)$ eine Rotationsmatrix ist:

$$\mathbf{R}(\Theta) = \mathbf{R}(Z, \Theta_Z) \mathbf{R}(Y, \Theta_Y) \mathbf{R}(X, \Theta_X) \quad (2.3.12)$$

Die sechs *Bewegungsparameter* eines starren Körpers im 3-D Raum sind wie folgt:

- drei translatorische Verschiebungen – $\Delta\mathbf{T} = [\Delta p_X, \Delta p_Y, \Delta p_Z]$
- drei Rotationskomponenten – $\Delta\Theta = [\Delta\Theta_X, \Delta\Theta_Y, \Delta\Theta_Z]$

Im allgemeinen Fall hätten wir einen dynamischen Vektor mit 12 Parametern. Dazu kämen noch die Formparameter, deren Anzahl von der Komplexität der Modellform abhängig ist. Für unseren Problembereich kann die Anzahl der Freiheitsgrade verringert werden durch die Beschränkung der Objektbewegungen auf Fahrbahnebene. Wir definieren den dynamischen Zustandsvektor $\mathbf{s}^d(k)$ eines Objektes zum Zeitpunkt des k -ten Bildes einer Bildfolge als einen fünfstelligen Vektor (vergleiche Bild 5.3.1 und Bild 5.1.2):

$$\mathbf{s}^d(k) = [p_X(k), p_Z(k), \Theta(k), V(k), \omega(k)]^T \quad (2.3.13)$$

wobei $V = |\Delta\mathbf{p}| = \sqrt{(\Delta p_X)^2 + (\Delta p_Z)^2}$ den Wert der translatorischen Bewegung in der Ebene OXZ entlang der Richtung Θ bedeutet und ω die Winkelgeschwindigkeit bezeichnet.

Die Komponenten des Formvektors einer Objekthypothese sind die Breite B_r und mehrere Parameter κ_i (vergleiche Bild 5.1.2 und Bild 5.1.3):

$$\boldsymbol{\xi}(k) = [B_r(k), \kappa_1(k), \kappa_2(k), \dots, \kappa_j(k)]^T \quad (2.3.14)$$

Die Anzahl und Bedeutung der Komponenten κ_i hängt von der Spezialisierung der Formrepräsentation und von der Objektklasse ab.

Die Vorhersage

Die dynamischen Zustandsgleichungen für einen Körper im 3-D Raum verbinden Lage und Bewegung des Körpers miteinander:

$$\mathbf{R}(k+1) = \mathbf{R}(k)\Delta\mathbf{R}(k) \quad (2.3.15)$$

$$\mathbf{p}(k+1) = \mathbf{p}(k) + \Delta\mathbf{p}(k) \quad (2.3.16)$$

Die Anzahl der Zustände und dadurch der Übergangsgleichungen ist zwar kleiner als im generellen Fall, aber es sind nichtlineare Abbildungen in der Übergangsfunktion $f(\mathbf{s})$ enthalten, so daß ein *EKF* verwendet wird.

Die Messung

Die Abbildungsfunktion $h(\mathbf{s})$ hängt direkt mit der Komplexität des Objektmodells sowie mit der Art der Messung zusammen, d.h. mit der Art der Zuordnung der Modellelemente zu Bildelementen.

- Eine 3-D Messung bedeutet eine wiederholte Objekthypothesengenerierung unabhängig von vorherigen Ergebnissen für das vorherige Bild. Die Modifizierung einer Objekthypothese erfolgt durch einen Objekt-zu-Objekt-Abgleich. In diesem Fall ist der Meßvektor in der Form identisch mit einem reduzierten Zustandsvektor:

$$\mathbf{m}(k) = \mathbf{s}_R(k) \cup \mathbf{s}_g(k) = \quad (2.3.17)$$

$$= [P_X(k), P_Z(k), Br(k), h_1(k), h_2(k))]^T \cup [\Theta_g(k), l_{1g}(k), l_{2g}(k)]^T \quad (2.3.18)$$

- Während der Verfolgung wird die Lage der Modellelemente (markante Punkte oder Liniensegmente) im 3-D Raum vorhergesagt und anschliessend werden diese Elemente in die Bildebene projiziert und einem Vergleich mit neu detektierten Bildsegmenten unterzogen (Gruppenmerkmale oder Kanten). Bei einer 2-D Messung gibt es eine nichttriviale Projektion $h(\mathbf{s}^+(k))$: auf der Basis der Lage von mehreren Modellpunkten im 3-D Raum (P_i -s, $i = 1, \dots, n$) wird die Bildlage und Richtung vorhergesagt; diese Punkte werden dann mit mehreren zusammengeführten Punkten im aktuellen Bild abgeglichen:

$$\mathbf{m}(k)^T = [p_1, p_2, \dots, p_n]^T \quad (2.3.19)$$

Im einfachsten Fall sind die Objekte durch fünf markante Modellkanten ihrer umgebenden 3-D Hülle spezifiziert. Diese Kanten werden auf die Bildebene projiziert und mit den Bildkanten im nächsten Bild verglichen (vergleiche Bild 5.5.2).

Die Modifikation

Die Modifikation wird in zwei Schritten durchgeführt. Direkt nach der Messung erfolgt die neue Schätzung des reduzierten Zustandes $\mathbf{s}_R^+(k)$. Anschließend können aus

$$\delta \mathbf{s}_R^+(k) = [\mathbf{s}_R^+(k) - \mathbf{s}_R^+(k-1)] \quad (2.3.20)$$

die neuen synthetischen Messungen der Geschwindigkeit $V(k), \omega(k)$, sowie die bewegungsabhängigen Messungen der Formparameter vorgenommen werden. Diese Messungen führen zur Modifikation des übriggebliebenen Zustandsvektors

$$\mathbf{s}_E(k) = [V(k), \Theta(k), \omega(k), \kappa_1(k), \dots, \kappa_i(k)]^T \quad (2.3.21)$$

Die Objektauswahl

Während der Objektverfolgung wird die volle Form des Konsistenztests und Erkennungstests angewandt. Die Anzahl der Objekte im Bild ist nicht bekannt. Es können damit mehrere konkurrierende Hypothesen für ein Objekt erzeugt werden, zwischen denen ausgewählt werden muß.

2.4 Systemkonfigurierung bei stationärer Kamera

Das System kann mit Hilfe von zwei Systemparametern EIGENBEWEGUNG und PERSPEKTIVE in drei Untersysteme für unterschiedliche Anwendungen konfiguriert werden. Die Belegung dieser Parameter hat folgende Bedeutung:

- EIGENBEWEGUNG = 0 – stationäre Kamera (alle Fahrzeugansichten möglich)

- EIGENBEWEGUNG = 1 – bewegte Kamera (die Ansichten überwiegend auf Front/Heck begrenzt)
- PERSPEKTIVE = 1 – perspektivische Verzerrung (kleiner Winkel Φ in Bild refKamera-neigung)
- PERSPEKTIVE = 0 – relativ großer Winkel Φ und annähernd parallele Projektion

Diese Parameter werden automatisch gesetzt beim Testen von folgenden Bedingungen:

1. IF ($V_{Eigenzustand} < 0.1m/sec$) THEN EIGENBEWEGUNG = 0
ELSE EIGENBEWEGUNG = 1
2. IF ($H/f > 0.10$) THEN PERSPEKTIVE = 0
ELSE PERSPEKTIVE = 1
wobei H die Höhe über der Fahrbahn und f die Brennweite bedeuten
3. IF ($\omega_{Eigenzustand} < 0.1Rad/sec$) THEN lineare Eigenbewegung
ELSE nichtlineare Eigenbewegung

Die Kombination der beiden Parametr führt zur folgender Rekonfigurierung des Gesamtsystems:

1. IF ((EIGENBEWEGUNG = 0) AND (PERSPEKTIVE = 0))
THEN Straßenkreuzung mit einer stationären Kamera (keine mehrfachen FP-Hypothesen, die Fahrbahn ist bekannt, 4 Richtungen der bewegten Objekte)
2. IF ((EIGENBEWEGUNG = 0) AND (PERSPEKTIVE = 1))
THEN stationäre Kamera auf einer Brücke (keine mehrfachen FP-Hypothesen, Fahrbahn ist bekannt, überwiegend zwei Richtungen der Objektbewegung)
3. IF (EIGENBEWEGUNG = 1)
THEN Kamera im bewegten Eigenfahrzeug (volle FP- und Fahrbahn-Erkennung, überwiegend zwei Richtungen der Objektbewegung)

Ein vollautomatisches System zum Bildverstehen für autonome Fahrzeuge sollte selbstverständlich zwischen diesen drei Zuständen schalten können. Als Beispiele von Verkehrsszenen die mit stationärer Kamera aufgenommen werden sollen hier Parkplatzszenen, Brücken-Autobahnszenen und Straßenkreuzungsszenen erwähnt werden (Bild2.4.1) (Bild2.4.2), (Bild2.4.3).



Bild 2.4.1: Objekterkennung in einer stationären Bildfolge einer Parkplatzszene.



Bild 2.4.2: Objekterkennung in einer stationären Bildfolge einer Autobahnszene mit mehreren Objekten auf drei Fahrspuren (Ermittlung der mittleren Geschwindigkeiten und Anzahl der Fahrzeuge pro Fahrspur).



Bild 2.4.3: Objekterkennung in einer stationären Bildfolge einer Straßenkreuzung mit mehreren Objekten (Ermittlung der Anzahl von Fahrzeugen pro Richtung) .

Kapitel 3

Bildsegmentierung und Verschiebungsdetektion

In diesem Kapitel werden Verfahren beschrieben, die auf der "untersten" Verarbeitungsebene eingesetzt werden (Bild 3.1.1). Der erste Abschnitt behandelt die ikonische Verarbeitung, sowohl die Einzelbildverarbeitung wie auch die pixelbasierten Methoden zur Bestimmung des optischen Flußes oder der Verschiebungsdetektion, unter Annahme einer perspektivisch verzerrten Objektbewegung auf Fahrbahnebene. Der Übergang von den ikonischen Daten zur initialer symbolischer Bildfolgenbeschreibung wird im Abschnitt 3.2 dargestellt (Detektion von Liniensegmenten und Regionen sowie deren Überführung in Konturen). Im Abschnitt 3.3 erfolgt die Beschreibung der entwickelten Methoden zur Konturverfolgung im Bild und zur Detektion von 3-D Attributen aus der Konturveränderung.

3.1 Ikonik

Diese Verarbeitungsstufe bekommt ein monokulares Bild sowie den momentanen Eigenzustand am Eingang und liefert mehrere Ergebnisbilder: das normalisierte Bild, das Kantenbild, und eine dynamische Bewegungsmaske (bzw. einen optischen Fluß). In Bild 3.1.3 sind Beispiele von Ergebnisbildern gegeben, die in den ersten Schritten erzeugt werden.

3.1.1 Bildvorverarbeitung

Diese Stufe enthält nur ein Minimum an brauchbaren Verfahren zur Vorverarbeitung eines Bildes. Damit wird die Nützlichkeit von anderen denkbaren Verfahren, wie z.B. zur Rauschunterdrückung oder zur Unterabtastung in Zeilensprungbildern, nicht negiert. Es kann sicherlich weiter untersucht werden in wieweit dieser zusätzlicher Aufwand zur einer Verbesserung der Erkennungsqualität des Systems beiträgt.

Unterabtastung

Im ersten Schritt wird das aktuelle Originalbild der verarbeiteten Bildfolge eingelesen und *unterabgetastet*, falls es sich um ein Bild in Zeilensprungtechnik handelt.

Der einfachste Weg, besteht aus der Reduktion der Auflösung um Faktor zwei, d.h. aus einem Bild der Größe $(2 * X \times 2 * Y)$ wird ein Bild der Größe $(X \times Y)$ erzeugt. Diese Methode der Unterabtastung bedeutet die Auswahl der Pixel in jeder zweiten Spalte und jeder zweiten Zeile. Um wegen der Eliminierung von "ungeraden" oder "geraden" Spalten, keine eventuellen Kanten zu verlieren, wird die willkürliche Auswahl in einer Zeile ersetzt durch eine selektive

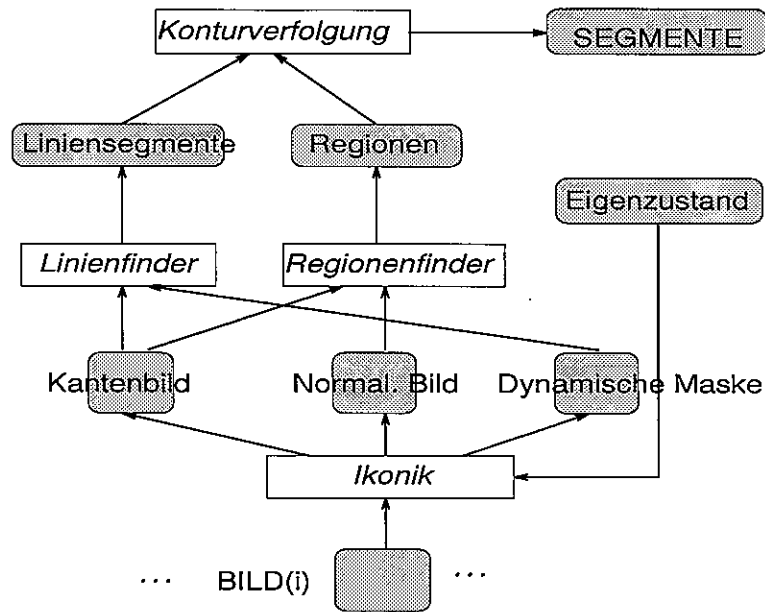


Bild 3.1.1: Die Verarbeitungsschritte des 2-D Moduls

Auswahl. Von zwei benachbarten Pixeln in einer Zeile wird jeweils jener behalten, der einen größeren Gradienten entlang der Zeile aufweist:

$$Pixel_{i,j} = Pixel_{2i,2j} \text{ oder } Pixel_{2i+1,2j}, \quad (i = 0, \dots, X - 1), (j = 0, \dots, Y - 1) \quad (3.1.1)$$

Es kann hier wohl angenommen werden, daß die Fahrzeuge in der Regel heller als die Fahrbahn und Umgebung aber dunkler als der Horizont sind. Deswegen wäre, durch die Anwendung alleine des Grauwertes als Auswahlkriterium, das zu erkennende Objekt nicht immer bevorteilt. Wie Untersuchungen in [Wet95] gezeigt haben, bringt eine vorherige Tiefpaßfilterung des Originalbildes keine wesentliche Verbesserung des unterabgetasteten Bildes. Sie ist aber mit einem wesentlichen Rechenaufwand verbunden und deswegen wurde hier darauf verzichtet.

Die Analyse von Bildfolgen mit Zeilensprungtechnik mit Beibehaltung der originellen Größe, wie sie in [Ott94] verfolgt wird, betrachten wir als einen umständlichen Versuch der "Verbesserung" von eigentlichen Sensordaten. Die Auflösung der zur Zeit verfügbaren Sensoren hat bei weitem noch nicht die physikalischen Grenzen erreicht, so daß mit ständig wachsender Auflösung der auf dem Markt verfügbaren Kameras in unmittelbarer Zukunft zu rechnen ist [SVS95].

Normalisierung

Zur Normalisierung des Bildes wird als erstes das Grauwert-Histogramm des aktuellen Bildes berechnet und entsprechende zwei Schwellen im Histogramm ermittelt. Das Histogramm stellt die Häufungen der 256 Grauwerte im Bild dar. Daraus werden die Schwellen H_{min} , H_{max} aus dem Intervall $\langle 0, 255 \rangle$ wie folgt ermittelt:

$$HSum = \sum_{i=0}^{255} \text{Häufung}(i) \quad (3.1.2)$$

$$H_{min} \in \langle 0, 255 \rangle := \left(\sum_{i=0}^{H_{min}} \text{Häufung}(i) \right) > (HTeil * HSum) \quad (3.1.3)$$

$$H_{max} \in \langle 0, 255 \rangle := \left(\sum_{i=H_{max}}^{255} \text{Häufung}(i) \right) > (HTeil * HSum) \quad (3.1.4)$$

Je $H_{\text{Teil}}\%$ (z.B. 1 %) der gesamten Pixelmenge besitzt Grauwerte, die kleiner sind als H_{min} bzw. größer als H_{max} .

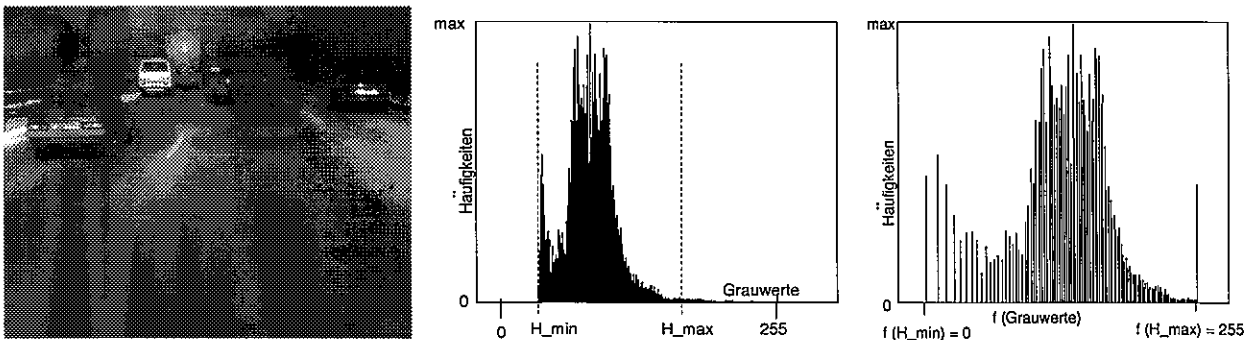


Bild 3.1.2: Die histogrammbasierte Streckung: das Originalbild nach Unterabtastung (links); das Histogrammbild mit den zwei gefundenen Schwellwerten (mitte); das Histogramm des gestreckten Bildes (rechts).

Die Grauwerte aus dem Intervall $\langle 0, H_{\text{min}} \rangle$ werden auf den Grauwert 0 im gestreckten Bild abgebildet, und die Grauwerte aus dem Intervall $\langle H_{\text{max}}, 255 \rangle$ werden auf 255 abgebildet. Der Bereich $\langle H_{\text{min}} + 1, H_{\text{max}} - 1 \rangle$ findet eine lineare Streckung statt, d.h. er wird linear auf den größeren Bereich $\langle 1, 254 \rangle$ abgebildet.

$$\text{GrauwertNeu} = \text{HFunktion}(H_{\text{min}}, H_{\text{max}}, \text{GrauwertAlt}) \quad (3.1.5)$$

Ziel dieser Normalisierung ist es (1) zu einem regelmäßigerem Energiespektrum des Bildes zu kommen und (2) eine weitere Verarbeitung mit beleuchtungsunabhängigen Schwellen (Homogenitätsschwelle für Konturen und Kantenstärkeschwelle für Kanten) zu ermöglichen.

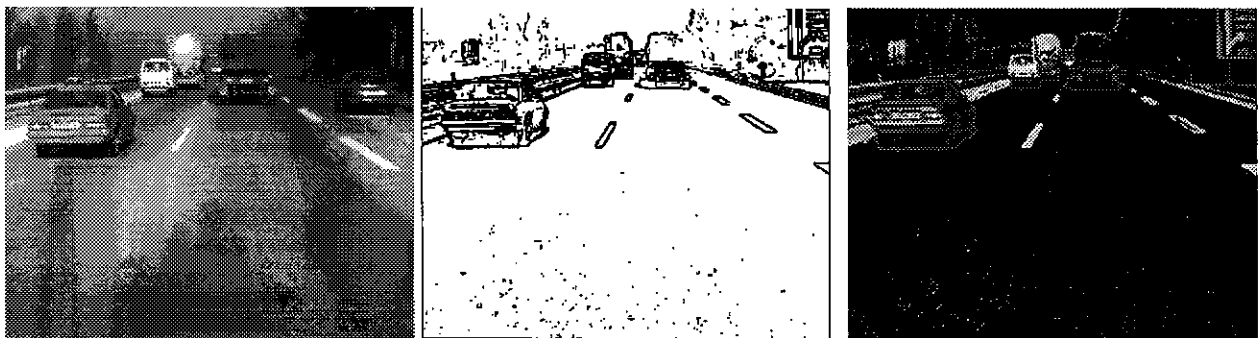


Bild 3.1.3: Beispiele der Kantendetektion: ein normalisiertes Bild (links), das Kantenbild (mitte), das durch das Kantenbild maskierte Bild (rechts)

3.1.2 Kantendetektor

Der Detektion von Kantenelementen sind viele Methoden der Bildverarbeitung gewidmet (z.B. [Can86], [Der87]). Diese Methoden können in zwei Kategorien eingeteilt werden, abhängig davon ob sie die Maxima der ersten Ableitung der Grauwertfunktion oder die Nullstellen der zweiten Ableitung der Grauwertfunktion zur Kantendetektion verwenden.

Die erste Kategorie von Kantendetektoren kann entweder durch Einsatz von relativ einfachen Gradientenoperatoren (wie Roberts, Sobel oder Prewitt-Operatoren) die Kantenstärke

für jedes Pixel berechnen oder auch durch Verwendung von mehreren Masken in größeren Umgebungen (wie der Nevatia–Babu Operator) noch zusätzlich für jedes Pixel die Kantenrichtung ermitteln. Ein Nachteil dieser Kategorie von Verfahren ist die Notwendigkeit einer anschließenden Kantenverdünnung, da meistens eine Kantenbreite von mehreren Pixeln erzeugt wird, und bei den komplexen Masken kommt der Rechenaufwand noch dazu, der benötigt wird um perfekte Kantenrichtungen zu ermitteln.

Die Nulldurchgangsverfahren berechnen die zweite Ableitung der Grauwertfunktion (z.B. [MH80], [Che87]) und detektieren deren Nulldurchgänge als Kantenelemente. Deren Nachteil gegenüber den vorherigen Verfahren liegt in größerer Rauschempfindlichkeit und eventuell auch darin, daß hier keine Angaben über Kantenrichtung und Kantenstärke geliefert werden. Zur Rauschunterdrückung wird oft vorher eine Gaußfilterung vorgenommen, was wiederum den Aufwand steigert.

Meistens wird ein Schwellwert eingeführt, um Kantenelemente mit kleinem Gradientenbeitrag zu unterdrücken. Solche Kantenelemente werden als rauschbedingt angesehen, was aber auch dazu führen kann, daß Kantenelemente in kontrastarmen Gebieten nicht detektiert werden. Deswegen wird häufig ein zusätzlicher Schritt nach der eigentlichen Kantendetektion zugefügt, der eine Verbesserung des Kantenbildes aufgrund einer Analyse der lokalen Umgebung jedes Kantenelementes erreichen soll. Insbesondere bei überwiegend schräg zur Bildebene verlaufenden Szene und somit für mit hohem Perspektiveffekt behafteten Bildern läßt sich aber ein generelles lokales Umgebungskriterium modellunabhängig nicht definieren.

Realisierung

Das integrierte Verfahren zur Kantendetektion und –Verdünnung wird in Bild 3.1.4 zusammengefaßt. Das Ziel der Kantendetektion in dem vorliegenden System ist nicht nur die Detektion der Kantenelemente im Bild, sondern auch eine Vorbereitung zur anschließenden Linien- und Regionendetektion. Insofern sind wir hier sowohl an den Gradientenstärken selbst interessiert, aber auch an jeweiligen Maxima der Grauwertfunktion. Was die Kantenrichtung anbetrifft, so wird nur die Klassifikation in eine von vier Richtungen (positiv oder negativ entlang der Bildzeile oder Spalte) erwartet.

Es wird ein Kantendetektor eingesetzt, der mit Masken des Sobel–Operators arbeitet, der aber die Kantenstärken getrennt in vertikaler und horizontaler Richtung berechnet. Es wird somit zwar ein *Kantenstärkebild* berechnet, der aber intern auch gleich zur Generierung eines speziellen *Kantenbildes* verwendet wird. Dieses Kantenbild enthält sowohl Kantelemente mit der Breite von einem Pixel, die also den Nulldurchgängen der diskreten Realisierung der ersten Ableitung der Grauwertfunktion entsprechen, aber mit zusätzlicher Angabe der Kantenrichtung, wie auch die restlichen Kantenpixel (mit Richtungsangabe), deren Kantenstärke eine vorgegebene Schwelle übersteigt.

Das Kantenbild eignet sich jetzt sowohl zur Liniensuche im Bild wie auch zur Maskierung des Grauwertbildes und zur gezielten Regionensuche in solchem Bild. Das Kantenstärkebild dient zusammen mit der dynamischen Maske zur Wichtigkeitsbeurteilung von später gefundenen Liniensegmenten.

3.1.3 Pixelbasierte Verschiebung

In diesem Schritt wird die *dynamische Maske* erzeugt, die pro Pixel entweder nur eine binäre Information (Bewegung ja oder nein) oder auch eine kodierte Verschiebungsmessung enthalten kann. Mit Hilfe der dynamischen Bildmaske wird die Wichtigkeit der später ermittelten Konturen eingestuft.

FOR jede Zeile im Bild	
FOR jedes Pixel von links nach rechts (oder umgekehrt)	
berechne horizontale Kantenstärke	
IF	lokales Maximum oder Minimum entlang der Zeile
THEN	belege entsprechendes Kantenlement mit Richtungskod und kodiere benachbarte Pixel als auf aufsteigender oder fallender Gradientenseite gelegen
FOR jede Spalte im Bild	
FOR jedes Pixel von oben nach unten (oder umgekehrt)	
berechne vertikale Kantenstärke	
IF	lokales Maximum oder Minimum entlang der Spalte
THEN	belege entsprechendes Kantenlement mit Richtungskod und kodiere benachbarte Pixel als auf aufsteigender oder fallender Gradientenseite gelegen

Bild 3.1.4: Das Schema der Kantendetektion

Durch die Kenntnis des aktuellen Eigenzustandes gegenüber der Fahrbahn wird der Zusammenhang zwischen jedem Pixel im aktuellen Bild und vorherigen Bild festgestellt. Dabei wird ein Pixel, daß "unter" dem Fluchtpunkt im Bild liegt als zur Fahrbahnebene angehörend angenommen und ein Pixel über dem Fluchtpunkt als zur Umgebung angehörend angenommen (Höhe über der Fahrbahn ist dann größer als die maximale Höhe eines Fahrzeuges):

$$Pixel_{i,j}(k) = Funktion(Pixel_{l,m}(k-1), Eigenzustand) \quad (3.1.6)$$

Bei der Anwendung von Standardverfahren zur Bestimmung der Verschiebung in Bildebene unterscheiden wir drei Schwierigkeitsstufen:

1. Parallele Projektion, stationäre Kamera
2. Perspektivische Projektion, stationäre Kamera
3. Perspektivische Projektion, überwiegend entlang der Sehachse sich bewegende Kamera

Um die Nachteile einer perspektivischen Projektion in Standardverfahren zur Verschiebungsschätzung auszugleichen wurden z.B. die log-polar Transformation der Bildebene vorgeschlagen ([DK95]).

Um eine nichttriviale Eigenbewegung bei der Verschiebungsschätzung im Bild zu berücksichtigen, bedarf es neben der Kenntnis der Fahrbahnebene und der Projektionsparameter, noch einer Detektion der Kameranickens sowie der Kenntnis der Eigenbewegung (vergleiche dazu Bild 3.1.5).

Adaptives Differenzbild

Hier sind wiederum zwei Optionen zulässig: es wird entweder ein Differenzbild oder ein adaptives Differenzbild ermittelt. Im zweiten Fall bedeutet dies eine Reimplementierung der Methode von Karman & von Brandt [KvB88] zur adaptiven Detektion des ruhenden Hintergrundes, hier aber mit Berücksichtigung der Eigenbewegung. Für jedes einzelne Pixel wird ein Veränderungswert von Bild zu Bild ermittelt. Falls dieser Wert unterhalb einer vorgegebenen Schwelle liegt, wird das entsprechende Pixel der neu erzeugten dynamischen Bildmaske auf 0 gesetzt, sonst erhält

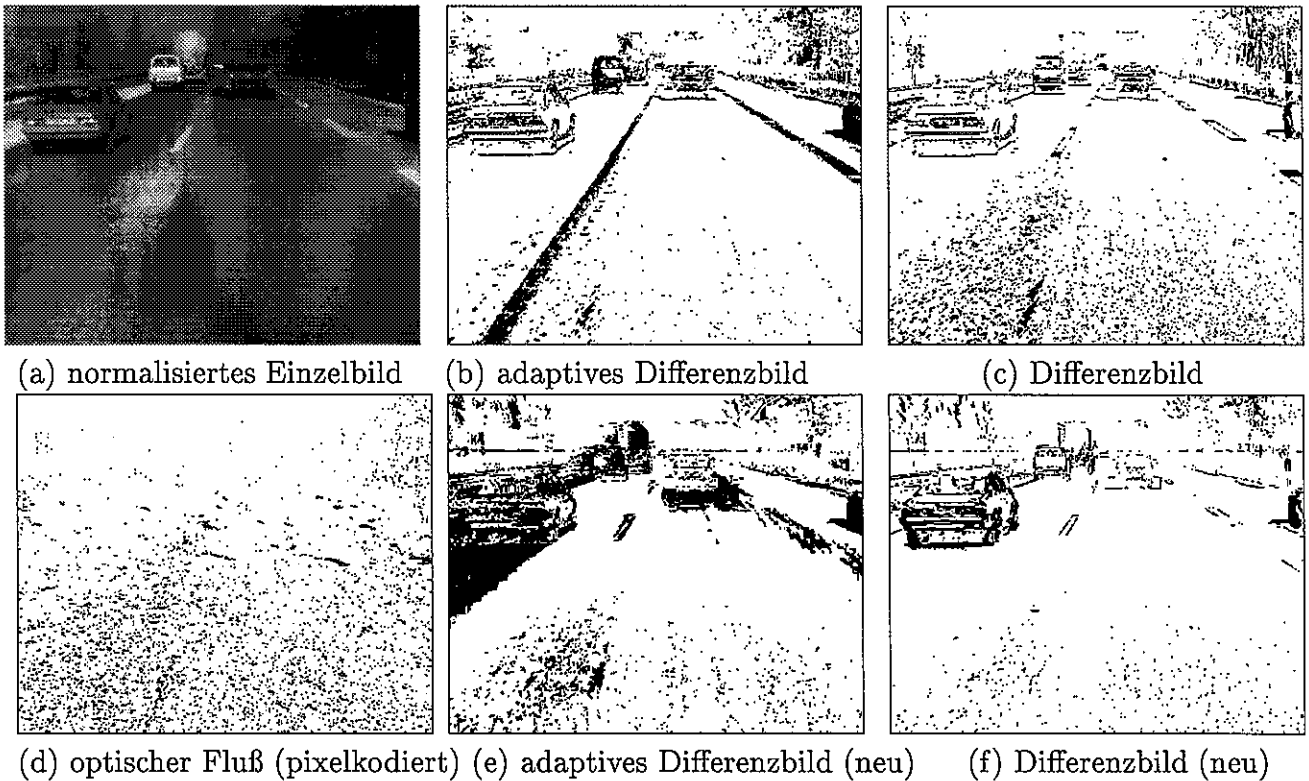


Bild 3.1.5: Beispiele der dynamischen Bildmaske erzielt durch pixelbasierte Verschiebungsschätzung: die drei Verfahren ohne Eigenbewegung (b,c,d), zwei Verfahren mit Berücksichtigung des Eigenzustandes (e,f)

es den Wert 1 (dies entspricht den dunklen und hellen Bereichen in Bild 3.1.3 rechts. Die Vorhersage und Modifikation des Zustandes erfolgt mit einem skalaren Kalman Filter, unter Annahme einer konstanten Mess- und Rausch-Varianz.

Optischer Fluß

Hier wurde eine klassische gradienten-basierte Methode zur Berechnung des optischen Flußes implementiert ([Nie90]). Sie wurde durch die Berücksichtigung der Eigenbewegung ergänzt. Auch hier wird vorher der Zusammenhang zwischen jedem Pixel im aktuellen Bild und vorherigen Bild festgestellt. Die Verschiebungsvektoren werden zu einem kodierten Bild zusammengefaßt. Die Kodierung der einzelnen Pixelwerte setzt sich zusammen aus einer diskreten Richtung und diskreter Länge des Verschiebungsvektors.

3.2 Linien- und Regionen-Detektion

3.2.1 Linienfinder

Ein Liniendetektor sucht im Kantenbild nach linearen und zusammenhängenden Kantenzügen, die als *Liniensegmente* bezeichnet werden.

Es wurde hier ein Suchverfahren im Kantenbild entwickelt, daß entlang der Kantenlemente eine topologische Suche durchführt und dabei nach jeder Zunahme eines Pixels eine Linearitätsbedingung überprüft. Dieses Kriterium basiert auf einer vorgegebenen maximalen mittleren Entfernung der gefundenen diskreten Kantenlemente von einer ausgleichenden Geraden.



Bild 3.2.1: Die detektierten Liniensegmente (links) und Vergleich zwischen Kantenelementen im Kantenbild (mitte) und Liniensegmenten (rechts) am Beispiel eines PKWs.



Bild 3.2.2: Die Regionendetektion: die Fahrbahnregion im Grauwertbild (links), das kantenmaskeierte Grauwertbild zur restlichen Regionensuche (mitte), das Regionensuche-Ergebnis (rechts)

Ein zusätzliches Kriterium prüft die eventuelle Längenkürzung des linearen Ausgleichsegmentes zwischen dem Anfangs- und Endpunkt.

Während der Liniensuche wird auch die topologische Nachbarschaft jedes Liniensegmentes zu einer vorher detektierten homogenen Region ermittelt.

3.2.2 Regionensucher

Vor der Regionensuche wird das Grauwertbild durch das Kantenbild maskiert. Es werden alle Pixel aus dem Grauwertbild entfernt (d.h. deren Grauwerte werden auf Null gesetzt), die zu keiner Umgebung eines Kantenelementes gehören. Die eigentlichen Kantenelemente werden auch entfernt, um eine Begrenzung für die Regionen einzuführen und eine fehlerhafte Verschmelzung von zwei Regionen zu vermeiden. Dadurch wird die Regionensuche auf relevante Bereiche begrenzt.

Das eingesetzte Verfahren erinnert stark an die topologische Liniensuche, wie sie vorher zur Liniendetektion eingesetzt wurde. Mit dem Unterschied, daß jetzt ein Homogenitätskriterium für die Grauwertfunktion die Suche steuert was zur detektion von geschlossenen Konturen führt. Um Missverständnissen vorzubeugen mit den späteren kantenbasierten Konturen, werden diese geschlossenen Regionenkonturen einfach als Regionen bezeichnet. Die Grauwerte der von innen oder außen an ein Kantenelement grenzenden Region müssen alle im Intervall $< I_m - \text{HOMOGEN}, I_m + \text{HOMOGEN} >$ liegen, wobei I_m der Anfangswert für die Suche nach dieser Region ist und HOMOGEN eine vorgegebene Homogenitätsschwelle ist.

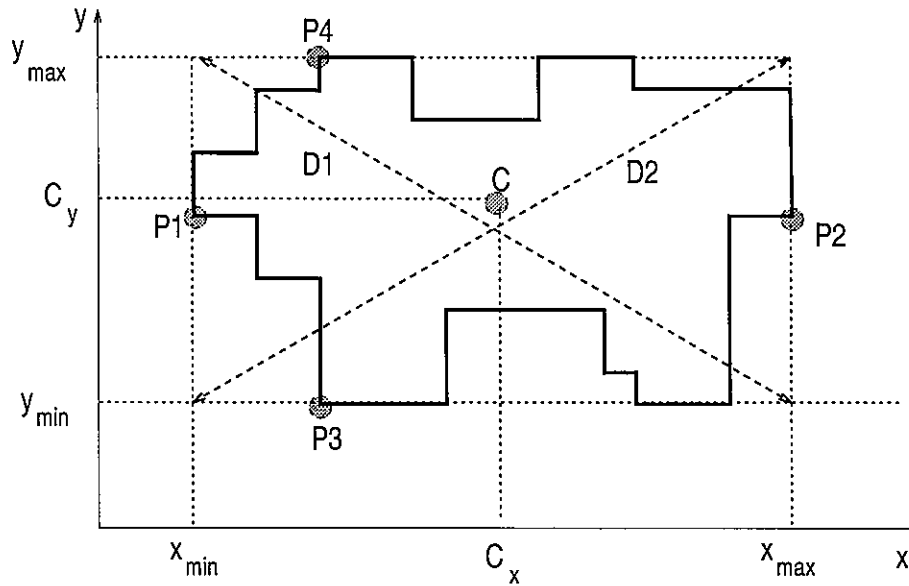


Bild 3.3.1: Diskrete Kontur und ihre Merkmale

3.3 Konturverfolgung im Bild

3.3.1 Konturdetektion im Einzelbild

Nachdem die Liniensegmente und Regionen ermittelt wurden, erfolgt der Zusammenschluß der benachbarten Segmente zu Konturen. Diese Nachbarschaft wird durch Zugehörigkeit zur gemeinsamer Region bestimmt.

Dabei werden für jede Kontur folgende Merkmale im Bild ermittelt (Bild 3.3.1):

- die Lage im Bild des Schwerpunktes $C = (C_x, C_y)$,
- die Lage der Punkte $P1, P2, P3, P4$ der maximalen Ausdehnung der Kontur in jede der 4 Richtungen $x+, x-, y+, y-$
- das umschreibende Rechteck $(x_{min}, x_{max}, y_{min}, y_{max})$
- Konturlänge l
- Summe der zwei Diagonallängen $d = |\vec{D1}| + |\vec{D2}|$
- der Anteil der Kantenelemente in der Kontur

Aus dem dynamischen Maskenbild wird für jede Kontur ein zusätzliches Attribut gewonnen: der Anteil der Pixel der dynamischen Maske in der Kontur an der Gesamtlänge der Kontur.

3.3.2 Stabile Messung von Konturen

Für Bildfolgen ähnlicher Qualität und Anwendung wurden mehrere Versuche durchgeführt, um die Messungen von Bildmerkmalen in langer Folge zu stabilisieren. Dabei handelte es sich um Positionswerte von Kantensegmenten [PM91] oder spiegelsymmetrische Kantenmerkmale [BK93]. Die Bildverschiebung wurde lediglich als Rauschen modelliert. Diese Modellierung kann aber nur für relativ unbewegte Objekte im Bild überzeugen. Sie erlaubt keine Aussagen über die Eigendynamik von Objekten mit signifikanter Bewegung. Das Kameranicken bei fahrendem Fahrzeug verursacht aber beträchtliche Fehler in der Verfolgung einer dynamischen 2-D Kontur. Deswegen werden in dieser Arbeit in einer längeren Bildfolge ($N > 5$), nur die Lagewerte im

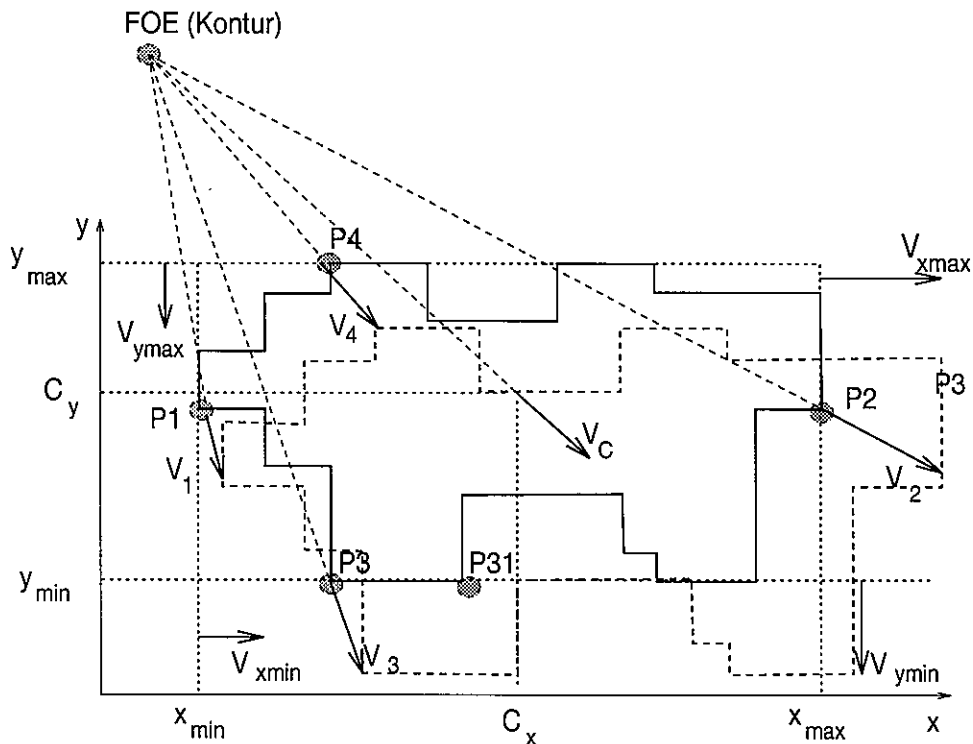


Bild 3.3.2: Dynamische Attribute einer Kontur

Bild stabilisiert. Eine Verfolgung und Stabilisierung der Dynamik von Objekten in längerer Bildfolge erfolgt erst auf der Ebene der 3-D Objekte.

Die Messungen einer einzelnen Kontur unterliegen Störungen, die auf Diskretisierungsfehler und Detektionsfehler zurückzuführen sind. Jeder Kontur im aktuellen Bild sind aber bis zu $N - 1$ frühere Konturen zugeordnet. Dies ermöglicht eine gewichtete Mittelung der detektierten Werte und eine Bewertung der Qualität dieser gemittelten Messung mit Hilfe der statistischen Varianz der (bis zu N) Einzelmessungen.

Dieser Vorgang zur Konturdetektion kann mit einer Oberflächensegmentierung von medizinischen Volumendaten verglichen werden [Kas89]. Ein normalisiertes Bild entspricht dann einem Schnittbild in den Volumendaten. Allerdings sind unsere Bilder nicht alle auf einmal verfügbar, sondern kommen in einer zeitlichen Reihenfolge, so daß unsere, durch Vorwissen gesteuerte, Konturdetektion einer iterativen Oberflächenerweiterung im Bild-Zeit-Raum entspricht [Kas93].

3.3.3 Die dynamischen Konturmerkmale

Eine Kontur wird abgespeichert mitsamt ihren Korrespondenzen in bis zu $N - 1$ ($N = 2 - 5$) früheren Bildern, d.h. die Datenstruktur einer Kontur besteht aus bis zu N korrespondierenden Einzelkonturbeschreibungen und den ermittelten Veränderungen. Falls in einer Konturstruktur mindestens zwei (höchstens aber N) korrespondierende Einzelkonturen zusammengefaßt sind, kann der aktuelle *Konturveränderungsvektor* (oder allgemein der *Konturzustandsvektor*) zum Zeitpunkt des aktuellen Bildes bestimmt werden (Bild 3.3.2). Er berechnet sich aus den Unterschieden zwischen Merkmalen der bis zu N korrespondierenden Konturen. Dies sind die bildlichen Verschiebungen von Punkten und die Veränderung der Konturlänge.

- Verschiebungen von markanten Punkten:

- Verschiebung des Schwerpunktes: $v_C = (v_{Cx}, v_{Cy})$
- Verschiebung des umschreibenden Rechtecks: $(v_{xmin}, v_{xmax}, v_{ymin}, v_{ymax})$;
- Verschiebung der (bis zu 4) ausgewählten Konturpunkte:
 $v^{P1} = (v_x^{P1}, v_y^{P1}); v^{P2} = (v_x^{P2}, v_y^{P2}); v^{P3} = (v_x^{P3}, v_y^{P3}); v^{P4} = (v_x^{P4}, v_y^{P4});$
- Veränderung der Länge:
 - Relative Veränderung der Konturlänge: v_z^l
 - Relative Veränderung der zwei Diagonallängen: v_z^d

3.3.4 Gewichtete Mittelung in kurzer Bildfolge

Der *Konturveränderungsvektor* besteht damit aus der Menge der Verschiebungen markanter Konturpunkte bzw. Veränderungen der Längen und Fläche zwischen den Zeitpunkten t_{j+1} und t_j ($j = 0, 1, 2, \dots$) der zwei aufeinanderfolgenden Bilder. Da wir aber nur diskrete und fehlerhafte Werte haben, wird die tatsächliche Konturveränderung (einer kontinuierlichen Kontur in Bildebene) angenähert durch eine gewichtete Mittelung von bis zu $N - 1$ diskreten Veränderungen.

Seien δp – die Entfernung im Bild zwischen zwei benachbarten Pixeln und τ – das Zeitintervall zwischen zwei Bildern (z.B. $1/25s$). Die Komponenten der Konturveränderung sind:

$$v_x = \frac{1}{(N-1)\tau} \sum_{i=1}^{N-1} \phi_i (x(i) - x(0)) \quad [\delta p / \delta t] \quad (3.3.1)$$

$$v_y = \frac{1}{(N-1)\tau} \sum_{i=1}^{N-1} \phi_i (y(i) - y(0)) \quad [\delta p / \tau] \quad (3.3.2)$$

$$v_z^l = \frac{1}{(N-1)\tau} \frac{\sum_{i=1}^{N-1} \phi_i (l(i) - l(0))}{l(0)} \quad [1/\tau] \quad (3.3.3)$$

$$v_z^d = \frac{1}{(N-1)\tau} \frac{\sum_{i=1}^{N-1} \phi_i (d(i) - d(0))}{d(0)} \quad [1/\tau] \quad (3.3.4)$$

$$(3.3.5)$$

Die Längenveränderungen $\{v_z^l, v_z^d\}$ sind damit im Gegensatz zu Verschiebungen $\{v_x, v_y\}$ keine Geschwindigkeiten sondern nur relative zeitliche Veränderungen.

Die Gewichte $\{\phi_i | i = 1, \dots, N - 1\}$ sollen den Tiefen $Z_C(i)$ (entlang der Z Kameraachse) der auf die Schwerpunkte $(x_C(i), y_C(i))$ der korrespondierenden Konturen abgebildeten Objektpunkte entsprechen:

$$\phi_i = \frac{x_C(1) - x_C(0)}{x_C(i) - x_C(0)} = \frac{y_C(1) - y_C(0)}{y_C(i) - y_C(0)} = \frac{Z_C(i)}{i Z_C(1)} \quad (3.3.6)$$

Da aber diese Tiefen zu diesem Zeitpunkt nicht bekannt sind, werden diese Gewichte auf Grund der relativen Konturveränderung v_z geschätzt. Aus der Definition von v_z und der Projektionsgleichung kann nämlich folgender Zusammenhang festgestellt werden zwischen v_z und Z_C für eine Objektfläche, die annähernd parallel zur Bildfläche ist:

$$\Delta Z = Z_C(i+1) - Z_C(i) \simeq Z_C(0) \left(\frac{1}{1+v_z} - 1 \right) \quad (3.3.7)$$

Da aber v_z selbst von den Gewichten abhängig ist, findet eine gleichzeitige iterative Berechnung von v_z und $\{\phi_i | i = 1, \dots, N - 1\}$ statt. Sie beginnt mit den initialen Werten:

$$\phi_i(\text{Init}) = \frac{1}{i}; \quad v_z(\text{Init}) = \frac{1}{N-1} \sum_{i=1}^{N-1} \phi_i(\text{Init}) \frac{(l(i) - l(0))}{l(0)}; \quad (3.3.8)$$

Die Gewichte werden wie folgt geschätzt:

$$\phi_i = \frac{Z_C(i)}{i Z_C(1)} \simeq \frac{1 + i\gamma}{i(1 + \gamma)}; \quad \text{wobei: } \gamma = \left(\frac{1}{1 + v_z} - 1 \right) \quad (3.3.9)$$

Die Messungen einer einzelnen Kontur unterliegen großen Störungen, die auf Diskretisierungsfehler und Detektionsfehler zurückzuführen sind. Da wir von mehreren korrespondierenden Konturen ausgehen und Veränderungswerte für Lage und Länge berechnen, kann für jede einzelne Kontur ihre Abweichung zwischen gemitteltem Wert und tatsächlich gemessenem Wert bestimmt werden. Wenn wir das als Fehlervariable auffassen, kann die Varianz der Fehlervariable in bis zu $N - 1$ Messungen bestimmt werden. Neben der Konturgröße bietet diese Varianz eine Voraussetzung zur Berechnung der Qualität Q einer bewegten Kontur.

$$Q = 1 / \left(\frac{l_i}{\text{Konturanzahl}} \sum_{i=1}^{\text{Konturanzahl}} f_{var}(i) \right) \quad (3.3.10)$$

wobei ist: $f_{var}(i) = \text{Varianz}(i)/l(i)$

3.3.5 Diskretisierungsfehler der Konturverfolgung

Sei mit $l(i)$ die diskrete Länge einer Kontur im Bild i bezeichnet und mit $L(i)$ die tatsächliche Länge der stetigen Kontur in Bildebene. Es können zwei Diskretisierungsfehler unterschieden werden:

- der *Systemfehler*
- der *Toleranzfehler*

Der Systemfehler hängt von der Form and Richtung der Kontur ab. Dieser Fehler könnte ausgeglichen werden, wenn die genaue Transformation der diskreten Kontur in ihre stetige Form bekannt wäre. Der Toleranzfehler hängt unmittelbar mit der Pixelgröße zusammen. Die Verschiebung der Kontur innerhalb von einer Pixellänge oder Pixelbreite verursacht Unterschiede in ihrer diskreten Darstellung.

Längenfehler

Ein Liniensegment der Länge L

In diesem Fall hängt der Systemfehler von der Richtung des Liniensegmentes im Bild ab. Für ein Segment, daß parallel zu einer der Bildachsen ist (der Winkel $\gamma = 0^\circ$ in Bild 3.3.3) gibt es keinen Systemfehler. Umgekehrt ist dieser Fehler am größten falls $\gamma = 45^\circ$. Der Toleranzfehler wird mit $\pm 0.5 \cos \gamma [\text{Pixel}]$ an jedem Segmentende angenommen (aber für $L > 1/2 [\text{Pixel}]$). Generell kann folgender Zusammenhang zwischen tatsächlicher Länge L und diskreter Länge l bestimmt werden:

$$L = \frac{l}{\cos \gamma + \sin \gamma} \pm \frac{1}{\cos \gamma}, \quad (0 \leq \gamma \leq 45^\circ) \quad (3.3.11)$$

Aus dieser Gleichung resultiert folgender Fehler im schlechtesten Fall:

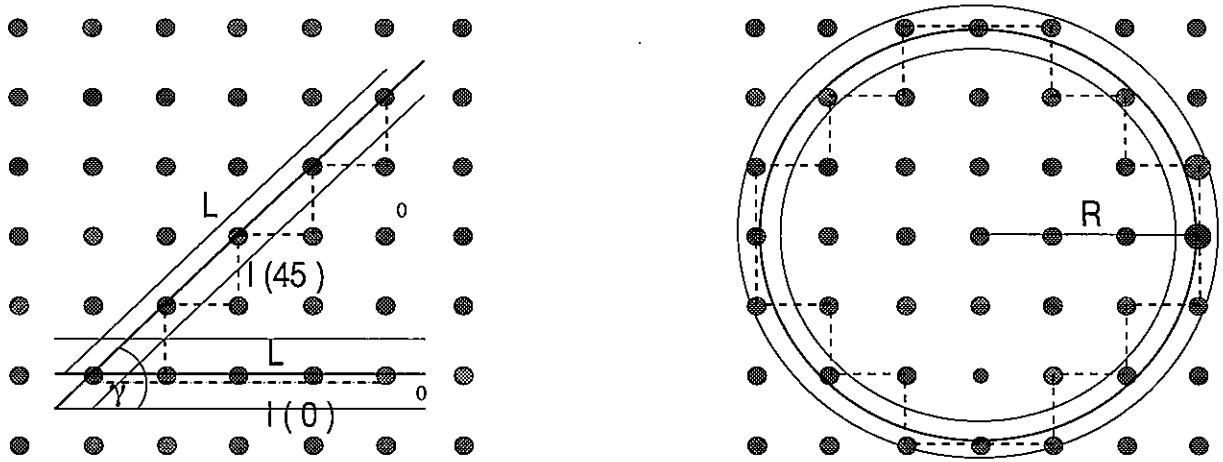


Bild 3.3.3: Diskretisierungsfehler von Geradensegment und Kreis

$$l \frac{\sqrt{2}}{2} - \sqrt{2} < L < l + 1, \quad (l > 2)$$

$$\frac{\sqrt{2}}{2} < L < 3, \quad (l = 2)$$

Ein Polygon mit Länge L

Wenn ein Polygon aus n Liniensegmenten besteht mit Längen $L_i (i = 0, \dots, n - 1)$, dann ist der Zusammenhang zwischen L und seiner diskreten Abbildung l wie folgt:

$$L = \sum_{i=0}^{n-1} \left(\frac{l_i}{\cos \gamma_i + \sin \gamma_i} \right) \pm \sum_{i=0}^{n-1} \frac{1}{\cos \gamma_i}, \quad (0 \leq \gamma_i \leq 45^\circ; i = 0, \dots, n - 1) \quad (3.3.12)$$

oder in Kurzform;

$$L = a l \pm b, \quad (0 \leq \gamma_i \leq 45^\circ; i = 0, \dots, n - 1) \quad (3.3.13)$$

mit

$$a = \frac{\sum_{i=0}^{n-1} l_i a_i}{\sum_{i=0}^{n-1} l_i}; \quad b = \sum_{i=0}^{n-1} \frac{1}{\cos \gamma_i}, \quad (0 \leq \gamma_i \leq 45^\circ; i = 0, \dots, n - 1)$$

Bei unbekanntenen Werten γ_i -s muß der Fehler im schlechtesten Fall berücksichtigt werden:

$$l \frac{\sqrt{2}}{2} - n \sqrt{2} < L < l + n, \quad (l > 2n)$$

$$\frac{n\sqrt{2}}{2} < L < 3n, \quad (l = 2n)$$

$$\frac{n}{2} < L < \frac{3n}{2}, \quad (l < 2n)$$
(3.3.14)

Ein Kreis

Nach Voss ([Vos88], S.121) kann die Länge l einer diskreten Abbildung eines Kreises mit Radius r approximiert werden durch folgende Gleichung:

$$l(r) \simeq 8(\text{ent}(r)) \quad (3.3.15)$$

wobei $\text{ent}(r)$ liefert den am nächsten gelegenen Integerwert für r . Die obige Gleichung entspricht dem Systemfehler der Kreisdiskretisierung. Der Toleranzfehler von r oder für die Lage des Mittelpunktes ist $\pm 1/2[\text{Pixel}]$. Sei $L = 2\pi r$ die tatsächliche Kreislänge. Dann ergibt sich folgender Diskretisierungsfehler für einen Kreis:

$$\frac{\pi l}{4} - \pi < L < \frac{\pi l}{4} + \pi, \quad (l \geq 8)$$

$$\frac{\pi}{2} < L < 2\pi, \quad (l = 4)$$
(3.3.16)

Diskretisierungsfehler der Konturverschiebung

Sei eine nichtnegative stetige Verschiebungskomponente \tilde{v}_x (oder \tilde{v}_y) gegeben. Der größte positive Fehler v_x^{max} (oder v_y^{max}) von \tilde{v}_x (oder \tilde{v}_y) erfolgt dann, wenn der Mittelpunkt $x_C(0)$ (oder $y_C(0)$) ist um $(-1/2 + \epsilon)$ von dem tatsächlichen Mittelpunkt $\tilde{x}_C(0)$ (oder $\tilde{y}_C(0)$) entfernt und alle anderen $x_C(i)$ -s (oder $y_C(i)$ -s) ($i = 1, \dots, N-1$) sind um $+1/2$ von den jeweiligen tatsächlichen $\tilde{x}_C(i)$ (oder $\tilde{y}_C(i)$) entfernt. Falls $x_C(0) = \tilde{x}_C(0) - \frac{1}{2}$; $x_C(i) = \tilde{x}_C(i) + \frac{1}{2}$ ($i = 1, \dots, N-1$) dann wird eine maximale diskrete Verschiebung ermittelt:

$$v_x^{max} = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} (\tilde{x}_C(i) + \frac{1}{2} - (\tilde{x}_C(0) - \frac{1}{2}))$$

$$\boxed{v_x^{max} = \tilde{v}_x + \frac{2}{N}; v_y^{max} = \tilde{v}_y + \frac{2}{N}} \quad (3.3.17)$$

Umgekehrt kann ein minimaler Verschiebungswert (negativer) v_x^{min} (oder v_y^{min}) für ein negatives \tilde{v}_x (oder \tilde{v}_y) bei

$$x_C(0) = \tilde{x}_C(0) + \frac{1}{2}; x_C(i) = \tilde{x}_C(i) - \frac{1}{2} \quad (i = 1, \dots, N-1)$$

erfolgen, definiert durch:

$$\boxed{v_x^{min} = \tilde{v}_x - \frac{2}{N}; v_y^{min} = \tilde{v}_y - \frac{2}{N}} \quad (3.3.18)$$

Tatsächlich ist der Diskretisierungsfehler von v_x und v_y nur von der Pixelgröße abhängig (abgesehen von Fehlern der Segmentdetektion) und nicht von der Konturform. Im folgenden wird gezeigt, daß auch der Diskretisierungsfehler von \tilde{v}_z nur ein Toleranzfehler ist.

Diskretisierungsfehler der Längenveränderung

Der maximale Wert v_z^{max} der diskreten Abbildung einer stetigen Konturveränderung mit positivem Wert \tilde{v}_z erfolgt unter folgenden Voraussetzungen:

$$v_z^{max} = \frac{2}{N(N-1)} \frac{\sum_{i=1}^{N-1} (l(i)_{max} - l(0)_{min})}{l(0)_{min}}$$

wobei $l(0)_{min}$ ist die minimale Entfernung zwischen $L(0)$ und $l(0)$ für gegebene Konturform und bei Winkel γ , sowie $l(i)_{max}$ ist die maximale Entfernung zwischen $L(i)$ und $l(i)$ ($i = 1, \dots, N-1$).

Umgekehrt ist der minimale Wert v_z^{min} der diskreten Abbildung einer stetigen Konturveränderung mit positivem Wert \tilde{v}_z verursacht durch folgende Bedingungen:

$$v_z^{min} = \frac{2}{N(N-1)} \frac{\sum_{i=1}^{N-1} (l(i)_{min} - l(0)_{max})}{l(0)_{max}}$$

wobei $l(0)_{max}$ ist die maximale Entfernung zwischen $L(0)$ und $l(0)$ für gegebene Konturform und bei Winkel γ , sowie $l(i)_{min}$ ist die minimale Entfernung zwischen $L(i)$ und $l(i)$ ($i = 1, \dots, N-1$).

Polygon

Die generelle Form der Diskretisierungsabbildung d für ein Polygon mit n Seiten

$$a(0)l(0)_{min} = L(0) - b(0) + \epsilon; \quad a(i)l(i)_{max} = L(i) + b(i) - \epsilon; \quad (i = 1, \dots, N-1).$$

führt zur folgendem maximalen Wert $v_{z_poly}^{max}$:

$$v_{z_poly}^{max} = \frac{2}{N(N-1)} \frac{\sum_{i=1}^{N-1} \left(\frac{L^i + \beta^i - \epsilon}{\alpha^i} - \frac{L^0 - \beta^0 + \epsilon}{\alpha^0} \right)}{\frac{L^0 - \beta^0 + \epsilon}{\alpha^0}} \quad (3.3.19)$$

Für den Fall, daß keine Rotation vorkommt sind die $a(i)$ -s ($i = 0, \dots, N-1$) identisch und sie könne aus der obigen Gleichung eliminiert werden. Dadurch ist der Systemfehler der v_z Messung gleich Null. Ohne Rotation sind auch die $b(i)$ -s alle gleich. In diesem Fall kann die Gleichung in folgende Form umgewandelt werden:

$$v_{z_poly}^{max} = (1 + \mathcal{K}_{poly}) \tilde{v}_z + \frac{4}{N} \mathcal{K}_{poly} \quad (3.3.20)$$

wobei sind

$$\mathcal{K}_{poly}(\gamma = 0^\circ) < \mathcal{K}_{poly} < \mathcal{K}_{poly}(\gamma = 45^\circ) \quad (3.3.21)$$

$$\mathcal{K}_{poly} = \frac{\frac{n}{\cos \gamma} - \epsilon}{L(0) - \left(\frac{n}{\cos \gamma} - \epsilon\right)} \quad (3.3.22)$$

$$\mathcal{K}_{poly}(\gamma = 0^\circ) = \frac{n - \epsilon}{L^0 - (n - \epsilon)} \quad ; \quad \mathcal{K}_{poly}(\gamma = 45^\circ) = \frac{n\sqrt{2} - \epsilon}{L^0 - (n\sqrt{2} - \epsilon)} \quad (3.3.23)$$

Ähnliche Form besitzt die Bestimmung des minimalen Wertes

$$v_{z_poly}^{min} = (1 - \mathcal{K}_{poly}) \tilde{v}_z - \frac{4}{N} \mathcal{K}_{poly} \quad (3.3.24)$$

Kreis

Änliche Überlegungen für einen Kreis mit Radius r (bei $l \geq 8$):

$$l(0)_{min} = \frac{4}{\pi} (L(0) - \pi(0) + \epsilon);$$

$$l(i)_{max} = \frac{4}{\pi} (L(i) + \pi(i) - \epsilon); \quad (i = 1, \dots, N-1).$$

führen zu folgenden maximalen $v_{z_kreis}^{max}$ und minimalen Werten $v_{z_kreis}^{min}$ für eine positive Konturveränderung \tilde{v}_z :

$$v_{z_kreis}^{max} = (1 + \mathcal{K}_{kreis}) \tilde{v}_z + \frac{4}{N} \mathcal{K}_{kreis} \quad v_{z_kreis}^{min} = (1 - \mathcal{K}_{kreis}) \tilde{v}_z - \frac{4}{N} \mathcal{K}_{kreis} \quad (3.3.25)$$

mit

$$\mathcal{K}_{kreis} = \frac{4}{\pi} \frac{\pi - \epsilon}{l^0} = \frac{\pi - \epsilon}{L^0 - (\pi - \epsilon)}$$

Vergleich

Die Qualität der gewichteten Messung von v_z ist im schlechtesten Fall besser als die Qualität der Messung von v_x (oder x_y) im schlechtesten Fall, wenn:

$$\left| \frac{v_z - \tilde{v}_z}{\tilde{v}_z} \right| < \left| \frac{v_x - \tilde{v}_x}{\tilde{v}_x} \right| \quad \text{oder} \quad \left| \frac{v_z - \tilde{v}_z}{\tilde{v}_z} \right| < \left| \frac{v_y - \tilde{v}_y}{\tilde{v}_y} \right| \quad (3.3.26)$$

Beispiel 3.1 (für $N = 5$)

1. Für ein Polygon mit 4 Seiten und stetiger Länge $L_{poly} = 84$ gilt:

$$0.928\tilde{v}_z - 0.058 < v_z < 1.072\tilde{v}_z + 0.058$$

$$\mathcal{K}_{poly} = \frac{n\sqrt{2} - \epsilon}{L_{poly} - (n\sqrt{2} - \epsilon)} < \frac{4\sqrt{2}}{84 - 4\sqrt{2}} \simeq 0.072$$

2. Für ein Kreis mit Umfang 21π ($r = 10.5$) gilt:

$$0.95\tilde{v}_z - 0.04 < v_{z_kreis} < 1.05\tilde{v}_z + 0.04$$

$$\mathcal{K}_{kreis} = \frac{\pi - \epsilon}{L_{kreis} - (\pi - \epsilon)} < \frac{\pi}{21\pi - \pi} = 0.05$$

3. Für das Polygon und den Kreis gilt zugleich:

$$\tilde{v}_x - 0.4 < v_x < \tilde{v}_x + 0.4 \quad ; \quad \tilde{v}_y - 0.4 < v_y < \tilde{v}_y + 0.4$$

Für den Kreis besitzt die gewichtete Messung von v_z eine bessere Qualität bezüglich der Diskretisierungsfehler als die gewichtete Messung von v_x (oder v_y) wenn:

$$\tilde{v}_x < \frac{0.4 \tilde{v}_z}{0.04 + 0.05\tilde{v}_z}$$

Z.B. für $\tilde{v}_z = 0.1$ muß sein $\tilde{v}_x < 0.889$; und für $\tilde{v}_z = 0.01$ muß sein $\tilde{v}_x < 0.099$

Aus der vorausgegangenen Analyse der Diskretisierungsfehler geht hervor, daß bei ausreichend großen L und \tilde{v}_z die gewichtete Messung von v_z mehr robuster erfolgt als die der Länge l . Z.B. für ein Polygon mit 4 Seiten und Länge $L = 80$ gilt bei $|\tilde{v}_z| = 0.2$ folgendes ($N=5$):

$$0.95L < l < 1.44L \quad ; \quad 0.62|\tilde{v}_z| < |v_z| < 1.38|\tilde{v}_z|$$

Dieselbe Berechnung für $L = 160$ und $|\tilde{v}_z| = 0.1$ ($N=5$) ergibt:

$$0.975L < l < 1.427L \quad ; \quad 0.67|\tilde{v}_z| < |v_z| < 1.33|\tilde{v}_z|$$

Die hier geführte Diskussion unterstreicht die Wichtigkeit der gewichteten Messung und der adaptiven Schätzung der Konturverschiebung und -veränderung. Der Diskretisierungsfehler der Länge, ist bei rotationsfreier Bewegung hauptsächlich ein Systemfehler, und ändert sich wenig in einer kurzen Bildfolge. Anders dagegen der Fehler von v_z , der bei gleicher Annahme ein Toleranzfehler ist. Die Wahrscheinlichkeit, daß der schlechteste Fehlerfall in einer längeren Bildfolge auftritt, ist gering. Deswegen wird in der Praxis ein deutlich niedriger Fehler erwartet, als der theoretisch schlechteste Diskretisierungsfehler.



Kapitel 4

Fahrbahnerkennung und Eigenzustandsverfolgung

In diesem Kapitel wird das 2,5-D Systemmodul zur Fahrbahnerkennung und Eigenzustands-schätzung beschrieben. Die einzelnen zu ermittelnden Parameter sind: die geometrischen und dynamischen Fluchtpunkte im Bild, der Fahrbahnparameter mit der Eigenfahrzeugposition gegenüber der Fahrbahnmitte sowie die Eigentrajektorie und Fahrbahntrajektorie im Raum. Die Verarbeitungsschritte dieses Moduls sind in Bild 4.1.1 zusammengefaßt.

4.1 Fluchtpunkte im Bild

Wir unterscheiden hier zwischen dem *geometrischen Fluchtpunkt* FP und dem *dynamischen Fluchtpunkt* FOE , wobei beide in der Bildebene liegen. FP wird definiert als der perspektivische Fluchtpunkt der Geraden auf denen die länglichen Kanten, d.h. entlang der Fahrbahn, liegen. FOE ist der perspektivische Fluchtpunkt der Geraden auf denen Verschiebungsvektoren von unbewegtem Hintergrund liegen, d.h. Verschiebung verursacht durch Eigenbewegung.

Durch die Bestimmung der beiden Fluchtpunkte werden unterschiedliche Ziele verfolgt. Der geometrische Fluchtpunkt dient zur laufenden Orientierungsbestimmung der Kamera bezüglich der Fahrbahn, was eine sehr wichtige Voraussetzung für die Erkennung im 3-D Raum bildet. Der dynamische Fluchtpunkt dient zur Klassifikation der Segmente als zu bewegten Objekten gehörend oder zum unbewegten Hintergrund.

4.1.1 Geometrischer Fluchtpunkt FP

Das Ziel der Detektion und adaptiven Schätzung des geometrischen Fluchtpunktes ist die Bestimmung, bzw. Korrektur, der Neigung der Kamera zur Fahrbahnebene und die Bestimmung der Richtung der Kamera bezüglich der Mittelaxis der Fahrbahn, bzw. der momentanen Tangente dieser Mittelaxis. Diese Aufgabe kann sowohl bei bewegter Kamera wie auch bei stationärer Kamera gestellt werden. Die Kamera ist zwar starr montiert und eine vorherige interaktive Messung der Fluchtpunktlage im Bild ist durchaus möglich und erwünscht, bei fahrendem Eigenfahrzeug kann sich aber die Lage ändern, verursacht durch Richtungsänderungen des Fahrzeuidges oder der Fahrbahn, sowie durch Auf- und Abwärtsbewegung beim Bremsen, Beschleunigen oder Überfahren von Bodenwellen.

Der Verarbeitungsschritt zur laufenden Bestimmung des Fluchtpunktes FP in der Bildebene besteht aus folgenden sekventiellen Teilschritten:

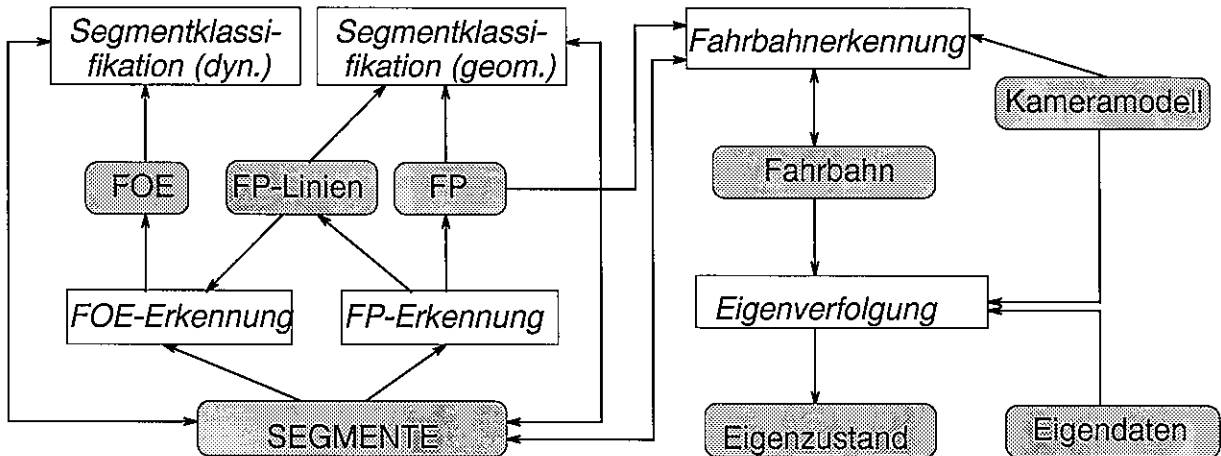


Bild 4.1.1: Das Modul zur Fahrbahnerkennung und Eigenzustandsverfolgung

1. die Bestimmung von Fluchtpunktgeraden der Fahrbahnmarkierung für eine ausgewählte Untermenge der Liniensegmente im aktuellen Bild
2. eine neue Messung (Detektion) des Fluchtpunktes als maximaler Häufungspunkt der Fluchtpunktgeraden in Bildebene
3. eine adaptive Schätzung des Fluchtpunktes als Modifizierung der vorherigen Schätzung und anschließende Vorhersage des nächsten Fluchtpunktes

Bestimmung von Fluchtpunktgeraden

Aus der Menge der Liniensegmenten werden sogenannte FP-Segmente ausgewählt. Diese Segmente sind überwiegend vertikal im Bild gelegen (der Verhältnis deren Breite zu Länge ist größer als eine vorgegebene Schwelle), deren benachbarte Regionengrauwerte sind relativ hell (die Grauwerte sind größer als eine weitere minimale Schwelle) und sie liegen im Bild unterhalb der bisher geschätzten Fluchtpunktlage. Ein Beispiel der hypothetischen Fluchtpunktsegmente ist in Bild 4.1.2(oben) gegeben.

Innerhalb der ausgewählten Konturen wird nach potentiellen Fluchtpunktanten gesucht. Es sind solche Kantenteile der FP-Konturen, die nicht allzu nahe an dem geschätzten Fluchtpunkt liegen (d.h. höchstens ein Endpunkt der Kante liegt im Bild innerhalb eines Kreises mit Zentrum in dem geschätzten Fluchtpunkt und mit vorgegebenem minimalen Radius), aber die Geraden auf denen diese Kanten liegen sind nicht weit entfernt von der geschätzten Fluchtpunktlage. (die Distanz zwischen einer FP-Geraden und dem geschätzten Fluchtpunkt soll geringer sein als eine weitere vorgegebene Schwelle).

Detektion des Fluchtpunktes (Messung)

Die Schnittpunkte der potentiellen Fluchtpunktgeraden werden in einem Schnittpunktfeld mit Radius FP_RADIUS (der variieren kann je nach Stabilität der bisherigen FP-Suche) um den geschätzten Fluchtpunkt herum gewichtet aufaddiert. Das Gewicht ist um so höher, je länger die beteiligte Kante ist. Der größte Häufungsbereich des Schnittpunktfeldes wird ermittelt. Der Mittelpunkt dieser Häufung bildet die aktuelle Fluchtpunktmessung (Bild 4.1.2(unten))

$$\mathbf{m}_{FP}(k) = [x_{FP}(k), y_{FP}(k)]^T \quad (4.1.1)$$

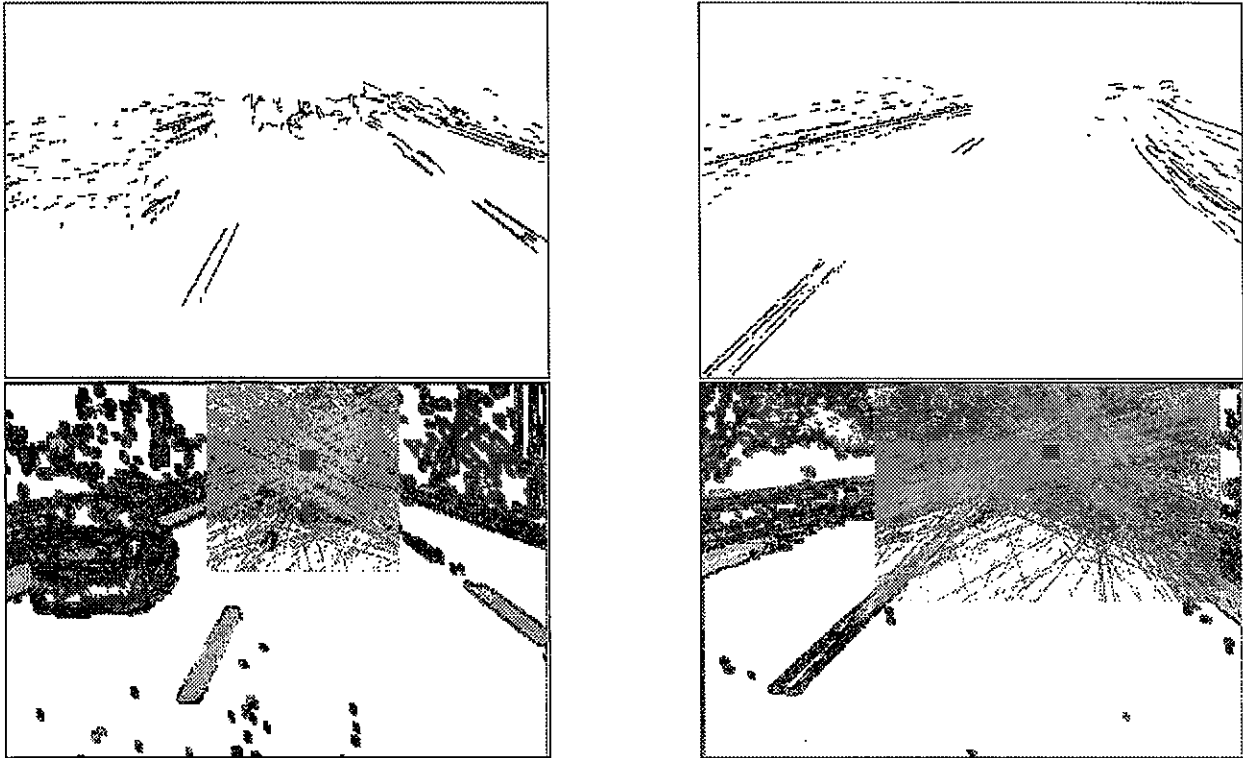


Bild 4.1.2: Liniensegmente zur Fluchtpunktbestimmung (oben) – bei linearem Fahrbahnverlauf (links) und bei Kurvenfahrt (rechts). Der geometrische Fluchtpunkt FP als maximale Häufung im Schnittpunktfeld der FP -Geraden (unten).

Die Genauigkeit dieses Wertes wird auf Grund des Häufungswertes geschätzt und in Form eines Messvarianzvektors

$$\mathbf{R}_{FP}(k) = [R_x(k), R_y(k)]^T \quad (4.1.2)$$

dargestellt.

Schätzung des Fluchtpunktes

Die Lage des geometrischen Fluchtpunktes FP im Bild wird über die Zeit mit einem linearen Kalman Filter stabilisiert. Es handelt sich um einen zeitlich diskreten Ansatz, deswegen setzen wir $t_k = k$. Zum Zeitpunkt des k -ten Bildes ist einerseits die aktuelle *Messung* $\mathbf{m}_{FP}(k)$ vorhanden, sowie die Varianzen der Messwerte $R_x(k)$ und $R_y(k)$. Andererseits existiert bereits eine *Vorhersage*

$$\mathbf{s}_{FP}^*(k) = [x_{FP}^*, \delta x^*, y_{FP}^*, \delta y^*]^T \quad (4.1.3)$$

mit der Kovarianzmatrix $\mathbf{P}_{FP}^*(k)$, die aus der vorherigen Schätzung durch Einsatz der aktuellen Vorhersagegleichung bestimmt wurden.

Die Messwerte führen zuerst zu einer neuen Schätzung der Lageparameter:

$$\mathbf{s}_R^+(k) = [x_{FP}^+(k), y_{FP}^+(k)]^T \quad (4.1.4)$$

Anschließend werden synthetische Verschiebungswerte

$$\delta x(k) = x^+(k) - x^+(k-1) \text{ und } \delta y(k) = y^+(k) - y^+(k-1) \quad (4.1.5)$$

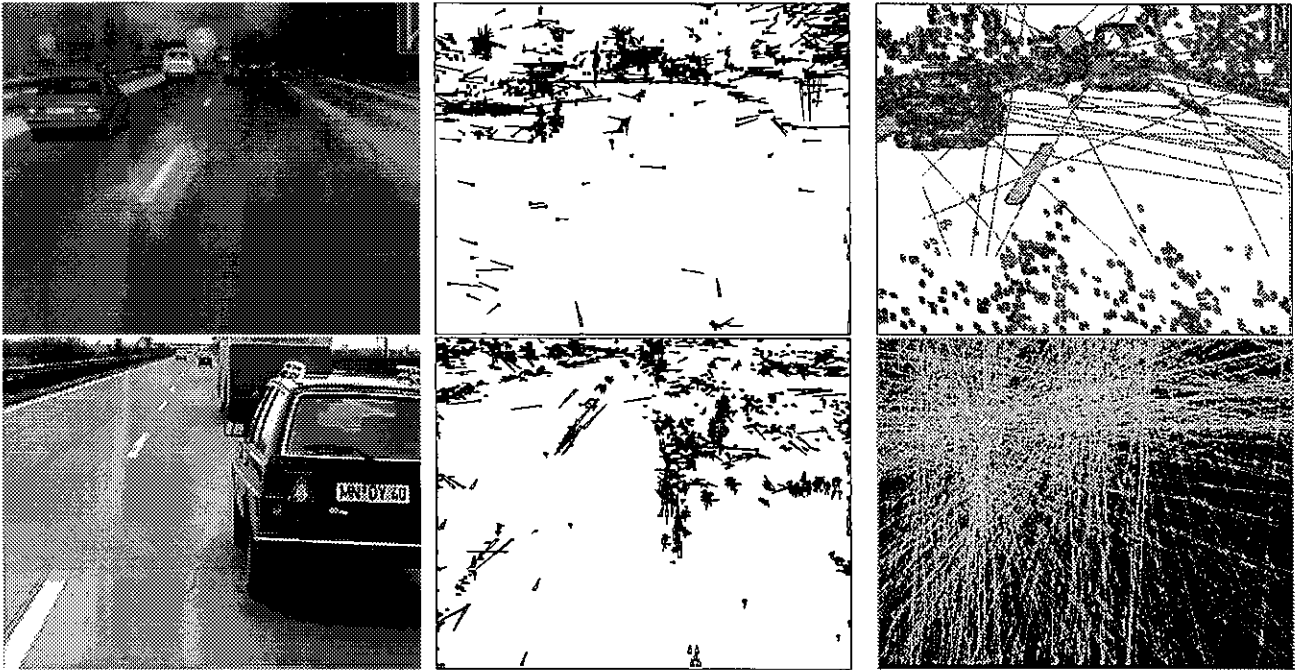


Bild 4.1.3: Detektion des Fluchtpunktes FOE : Originalbilder (links), Verschiebungsvektoren von Liniensegmenten (mitte), FOE als maximale Häufung der Verschiebungsvektoren von Fluchtpunktgeraden (rechts)

gemessen. Sie tragen dann zur Modifizierung der geschätzten Verschiebungen $\delta x^+(k), \delta y^+(k)$ bei.

Für eine erfolgreiche Stabilisierung ist es entscheidend eine Methode zu finden, mit der die Varianzen in $\mathbf{P}_{FP}(k)$ zuverlässig bestimmt werden können. In dieser Arbeit ist dafür die Relation der Anzahl der Häufungspunkte im gefundenen Bildbereich zur gesamten Anzahl der Häufungspunkte im Suchbereich maßgebend.

Um einen Vergleich zwischen bis zu drei konkurrierenden FP-Hypothesen zu erreichen, müssen die Schätzvarianzen durch jeweilige Häufungszahlen normiert werden. Dadurch können die momentanen gemittelten globalen Häufungen für jede Hypothese bestimmt und miteinander verglichen werden.

4.1.2 Der dynamische Fluchtpunkt FOE

Während einer Gruppierung von Konturen zu Objekthypothesen ist es wichtig entscheiden zu können, ob eine Kontur zu einem bewegten oder stationärem Objekt gehört. Dies ist ein Ziel der Schätzung des *dynamischen* Fluchtpunktes im Bild $FOE = (FOE_x, FOE_y)$ (Bild 4.1.3). Bis auf die unterschiedliche Art der Eingangsdaten unterscheidet sich dieser Verarbeitungsschritt nicht von dem vorherigen Schritt zur Bestimmung des Fluchtpunktes FP . Der Fluchtpunkt FOE wird bestimmt als der maximale Häufungspunkt von Schnittpunkten solcher Geraden im Bild, auf denen die Verschiebungsvektoren der Konturen mit (geometrischen) FP-Kanten liegen.

Der Einfluß der Kamerabewegung (Kamerafahrzeugbewegung und Kameranicken), relativ zur Straßentrajektorie, wird durch die Winkel $\alpha_{FOE}, \beta_{FOE}$ beschrieben:

$$\beta_{FOE} = \arctan\left(\frac{FOE_x}{F}\right); \quad \alpha_{FOE} = -\arctan\left(\frac{FOE_y}{F}\right); \quad (4.1.6)$$

wobei F die bekannte Brennweite der Kamera ist. Im Idealfall geradliniger Bewegung auf gerader Straße bzw. gleicher Krümmungen beider Trajektorien und ohne Kameranicken, müßten sich die beiden Fluchtpunkte FP und FOE überlagern (unter Annahme fehlerfreier Detektion). Die Differenzen der entsprechenden Winkel $\Delta\beta$ und $\Delta\alpha$ entsprechen der Krümmungsdifferenz und der Nickrotation in Zeit τ :

$$\Delta\beta(k) = \beta(k) - \beta_{FOE}(k) ; \Delta\alpha(k) = \alpha(k) - \alpha_{FOE}(k); \quad (4.1.7)$$

$$(4.1.8)$$

4.2 Konturklassifikation im Bild

4.2.1 Geometriebasierte Konturklassifikation

Dieser Schritt bedeutet eine Vorbereitung zur Konturgruppierung und sein Ablauf ist wie folgt:

1. Es wird die Entfernung aller bisherigen Fluchtpunktgeraden vom neu geschätzten Fluchtpunkt berechnet. Nur solche Geraden, deren Entfernungswert unter einer vorgegebenen Schwelle liegt, werden berücksichtigt.
2. Jedes noch verbliebene Segment S erhält eine Gewichtung, welche die Länge des Segmentes mit der Entfernung seines Mittelpunktes von dem Fluchtpunkt kombiniert.
3. Auf der Basis der verbliebenen FP-Segmente im Bild wird der linke und rechte Straßenrand im Bild ermittelt. Es werden die am meisten nach außen gerichteten FP-Geraden gewählt, die eine gewisse Gewichtungsschwelle überschreiten.
4. Die Bildbereiche können jetzt klassifiziert werden. Es wird eine Mask *Klassen_Bild* erzeugt, die drei Bereiche enthält: Straße, Umgebung, Horizont (Bild 4.2.1).
5. Jede Kontur, deren innere Fläche einen gewissen Anteil von Straßenpixeln in *Klassen_Bild* besitzt, wird als zur Straßenklasse gehörend klassifiziert. Konturen ohne gewisser Straßenanteile aber mit Umgebungsanteilen, werden der Umgebungs-klasse zugewiesen. Die restlichen Konturen verbleiben in der Horizontklasse (Bild 4.2.1).
6. Nachdem die Lage des Fluchtpunktes FP bekannt ist findet eine weitere Klassifikation der Fahrbahnkonturen in zwei spezialisierte Klassen statt: Fahrbahnstreifen und Hindernis (Bild 4.2.2).

4.2.2 Verschiebungsbasierte Konturklassifikation

Insbesondere ohne Eigenbewegung wird eine robuste Bestimmung der bewegten Konturen im Bild durch den Einsatz der dynamischen Maske ermöglicht (Bild 4.2.3). Die Segmente werden je nach Überdeckung ihrer Konturen durch die Pixel der dynamischen Maske als bewegt oder unbewegt klassifiziert. Bei Eigenbewegung wird überwiegend von der geometrischen Konturklassifikation gebraucht gemacht. Durch die Berücksichtigung des dynamischen Fluchtpunktes FOE, wird aber auch dort die Verschiebungsinformation zur Unterscheidung zwischen Hintergrundkonturen und Konturen von bewegten Objekten herangezogen.

4.2.3 2.5-D Konturattribute

Für ausgewählte Konturen, die relativ groß im Bild sind und für die mehrere Merkmale zuverlässig geschätzt werden können, ist eine Anwendung von Methoden der Klasse *Struktur-aus-Bewegung* denkbar. Wir beschränken uns weiterhin auf Fälle, wo die Rotation von Objekten im

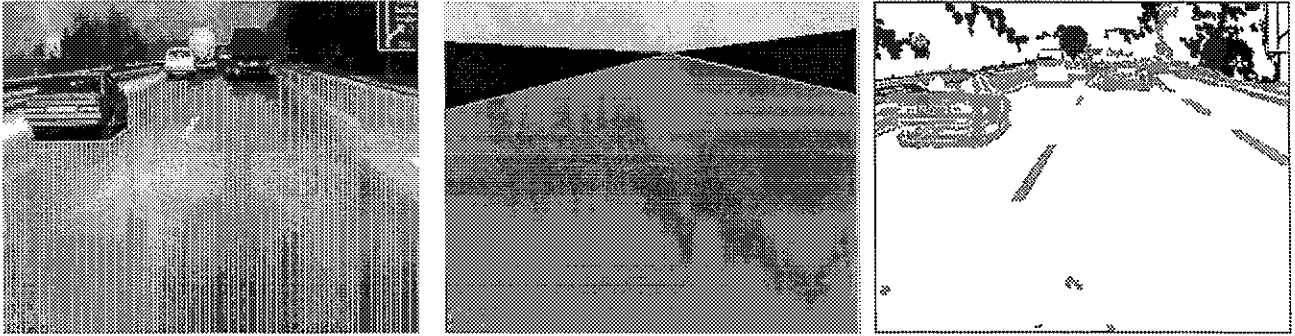


Bild 4.2.1: Geometriebasierte Konturklassifikation: die Fahrbahn- und Horizontregionen (links) erzeugen zusammen mit dem Fluchtpunkt eine Dreiklassen-Bildmaske bestehend aus – Fahrbahn, Umgebung und Horizont (mitte); gemäß der Überdeckung von entsprechenden Bildbereichen wird jede Kontur einer von drei initialen Klassen zugeordnet (Fahrbahnbereich, Umgebung, Horizont) (rechts).

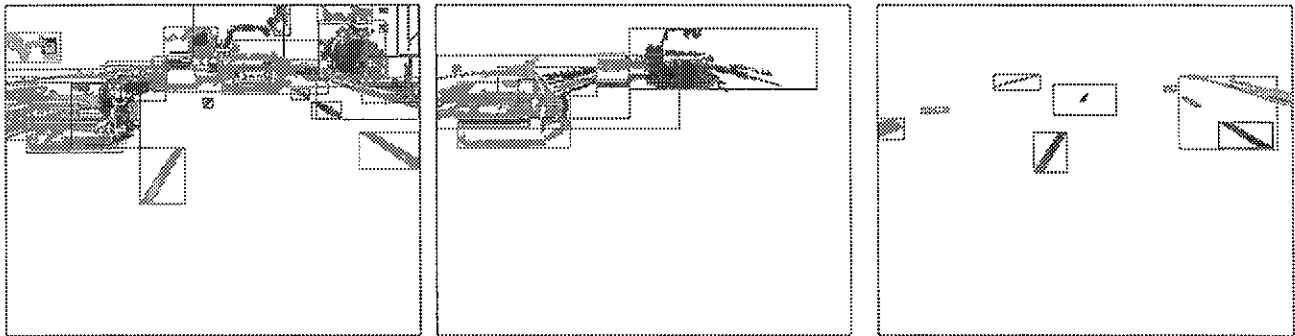


Bild 4.2.2: Klassifikation von Fahrbahnkonturen: aus der Menge der Konturen (links) wird die Untergruppe der potentiellen Hinderniskonturen (mitte) und potentiellen Fahrbahnstreifenkonturen (rechts) bestimmt.

Vergleich zur Verschiebung vernachlässigt werden kann. Zweckes weiterer Vereinfachung sollen die projizierten Flächen planar sein.

Längenbasierte Schätzung der Geschwindigkeit

Am Anfang nehmen wir an, daß die projizierte Fläche parallel zur Bildebene verläuft. Sei mit $\mathbf{V}^k = (\Delta X^k, \Delta Y^k, \Delta Z^k)^T$ die unbekannte translatorische Objektverschiebung im Zeitintervall τ in Kamerakoordinaten bezeichnet. Die Abbildung eines Punktes P mit Kamerakoordinaten $P^k = (X^k, Y^k, Z^k)$ auf die Bildebene ist ein Punkt $p = (x, y, 0)$ mit:

$$X^k = x \frac{Z^k}{F} \quad ; \quad Y^k = y \frac{Z^k}{F}$$

wobei F die Relation der Brennweite der Kamera zur Pixelgröße bedeutet (d.h. die Brennweite wird ausgedrückt in Pixeln).

Sei ein Geradensegment der Fläche mit den Endpunkten P_0, P_1 gegeben. Mit \mathcal{L} bezeichnen wir die Entfernung zwischen P_0 und P_1 , und mit L die Entfernung zwischen den Projektionen der Endpunkte p_0 und p_1 . Es gilt

$$\mathcal{L} = \frac{Z^k}{F} L$$

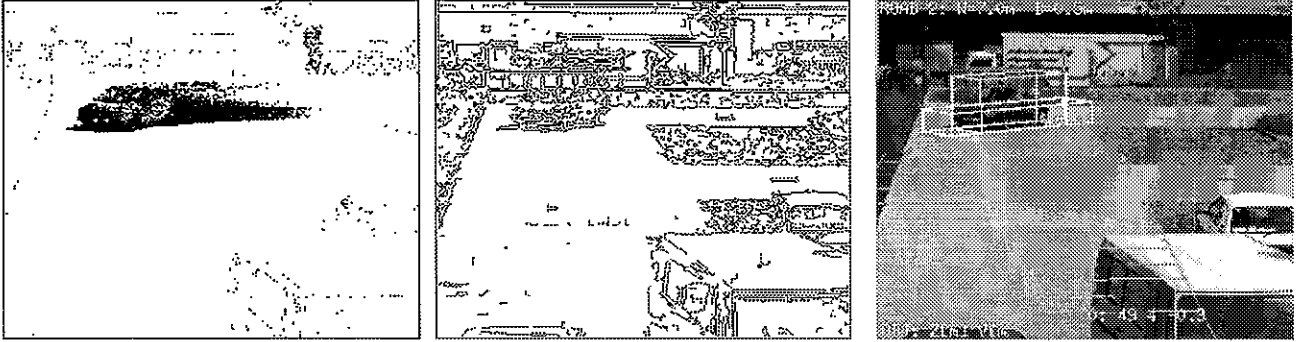


Bild 4.2.3: Verschiebungsbasierte Klassifikation: die dynamische Maske (links) führt von der gesamten Segmentmenge (mitte) zur Auswahl der bewegten Segmente, die später zu einer Objekthypothese zusammengefaßt werden (rechts).

Mit $Z^k(k)$ und $Z^k(k+1)$ bezeichnen wir die Tiefe der Fläche zu Zeitpunkten t_k und t_{k+1} , wobei $t_{k+1} - t_k = \tau$. Es gilt:

$$\mathcal{L}(k) = \mathcal{L}(k+1) \Leftrightarrow \left[\left(\frac{Z^k(k)}{F} \right) * L(k) \right] = \left[\left(\frac{Z^k(k+1)}{F} \right) * L(k+1) \right]$$

$$\frac{L(k+1)}{L(k)} = \frac{Z^k(k)}{Z^k(k+1)} = \frac{Z^k(k)}{Z^k(k) + \Delta Z^k} \quad (4.2.1)$$

Für die diskrete Approximation l der kontinuierlichen Kontur L gilt aufgrund der Definition von v_z in Kapitel 3. folgendes:

$$\frac{L(k+1)}{L(k)} \simeq \frac{l(k+1)}{l(k)} = \frac{l(k) + \Delta l}{l(k)} = 1 + v_z \quad (4.2.2)$$

Die rechten Seiten der beiden letzten Gleichungen beschreiben den Zusammenhang zwischen der Komponente v_z der Konturveränderung und der Relation von Objektverschiebung zur Tiefe des auf den Konturschwerpunkt projizierten Objektpunktes:

$$\Delta Z^k \simeq Z^k(k) \left(\frac{1}{1 + v_z} - 1 \right) \quad (4.2.3)$$

Die beiden restlichen Komponenten der Objektverschiebung können auch relativ zur Tiefe geschätzt werden:

$$\Delta X^k = \frac{1}{F} [\Delta Z^k(x(k) + v_x) + v_x Z^k(k)] \quad (4.2.4)$$

$$\Delta Y^k = \frac{1}{F} [\Delta Z^k(y(k) + v_y) + v_y Z^k(k)] \quad (4.2.5)$$

Virtuelles Segment

Im allgemeinen verlaufen die projizierten Liniensegmente nicht parallel zur Bildebene. Wir lösen diesen allgemeinen Fall, indem für das Originalsegment zu jedem Zeitpunkt ein *virtuelles* Liniensegment definiert wird, das parallel zur Bildebene verläuft. Dieses virtuelle Segment soll auf dieselbe Bildkante abgebildet werden, wie das originelle Liniensegment.

Sei \mathcal{L} die Länge eines Liniensegmentes zwischen den Punkten P_1 und P_2 . Seien Z_1 und Z_2 die Tiefen der Endpunkte. Das *virtuelle* Segment mit Länge \mathcal{L}_v verläuft zwischen den Punkten $P_{v1} = (X_{v1}, Y_{v1}, Z_C)$ und $P_{v2} = (X_{v2}, Y_{v2}, Z_C)$ mit $Z_C = (Z_1 + Z_2)/2$ (Bild 4.2.4).

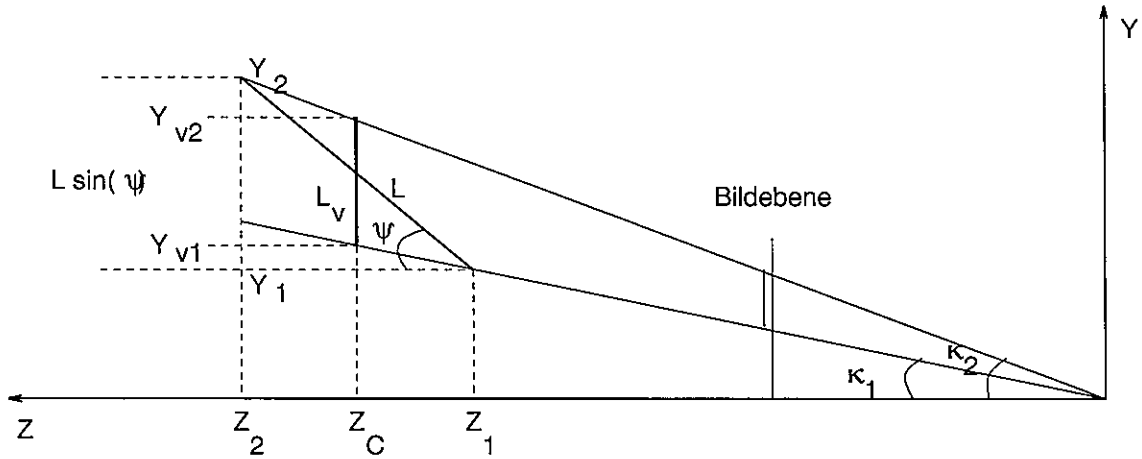


Bild 4.2.4: Virtuelle Objektkante

$$\mathcal{L}_v = \mathcal{L} \sin(\psi) - \text{sign}(Y_2 - Y_1) * \left[\text{sign}(Y_1) \tan(k_1)(Z_1 - Z_C) + \text{sign}(Y_2) \tan(k_2)(Z_C - Z_1) \right]$$

$$\text{mit : } \psi = \arctan\left(\frac{\sqrt{(Y_2 - Y_1)^2 + (X_2 - X_1)^2}}{|Z_2 - Z_1|}\right); \quad k_i = \arctan\left(\frac{\sqrt{Y_i^2 + X_i^2}}{|Z_i|}\right) \quad (4.2.6)$$

Für Tiefen $Z_C(k)$ die relativ groß sind im Vergleich zur Segmentlänge oder für Segmente, die parallel zur Bildebene liegen, bleibt $\mathcal{L}_v(k)$ annähernd konstant mit der Zeit. Andernfalls muß der Einfluß der sich ändernden Winkel $\kappa_i(k)$ -s berücksichtigt werden:

$$\mathcal{L}_v(k+1) \simeq \mathcal{L}_v(k) \Rightarrow \Delta Z(k) \simeq Z_C(k) \left(\frac{1}{1 + v_z(k)} - 1 \right) \quad (4.2.7)$$

$$\mathcal{L}_v(k+1) = \theta(k) \mathcal{L}_v(k) \Rightarrow \Delta Z(k) \simeq Z_C(k) \left(\frac{\theta(k)}{1 + v_z(k)} - 1 \right) \quad (4.2.8)$$

Wenn die Kontur parallel zur Bildebene verläuft dann ist $\theta = 1.0$ und wenn die Kontur nahezu vertikal zur Bildebene ist dann ist $\theta \simeq 1 - \gamma$.

Punkte-basierte Geschwindigkeit

Durch Verfolgung der Lage von mehreren Punkten der Kontur, können wir gemäß dem Prinzip des Bewegungsstereo eine Schätzung der 3-D Bewegung und Lage vornehmen.

Sei wie bereits vorher mit $V = (\Delta X, \Delta Y, \Delta Z)^T$ die gesuchte translatorische Objektverschiebung (in Kamerakoordinaten) im Zeitraum $\tau = t_{k+1} - t_k$ bezeichnet. Ein Punkt $P(k) = (X, Y, Z)^T$ (in Kamerakoordinaten) wird auf einen Bildpunkt $p(k) = (x, y, 0)^T$ abgebildet. Sei $P(k+1) = (X', Y', Z')$ die Lage des Punktes P(k) nach dem Zeitraum τ und $p(k+1) = (x', y', 0)$ die Lage seiner Projektion im nächsten Bild. Der Verschiebungsvektor $(u, v)^T$ von p im Bild ergibt sich aus:

$$u = x' - x = \frac{-F\Delta X - x'\Delta Z}{Z} \quad (4.2.9)$$

$$v = y' - y = \frac{-F\Delta Y - y'\Delta Z}{Z} \quad (4.2.10)$$

Nach Eliminierung von Z führt das zur folgenden Gleichung:

$$-Fv\Delta X + Fu\Delta Y = \Delta Z(xv - uy) \quad (4.2.11)$$

Bis zu 5 Punkte einer Kontur stehen zur Verfügung. Für je zwei Punkte P_1, P_2 einer Kontur kann ein System von zwei Gleichungen dieser Form mit den drei Unbekannten $\Delta X, \Delta Y, \Delta Z$ aufgebaut werden. Die Lösungen für ΔX und ΔY sind proportional zu ΔZ

$$\Delta X = \frac{u_2(x_1v_1 - u_1y_1) - u_1(x_2v_2 - u_2y_2)}{F(u_1v_2 - v_1u_2)} \Delta Z; \quad (4.2.12)$$

$$\Delta Y = \frac{v_2(x_1v_1 - u_1y_1) - v_1(x_2v_2 - u_2y_2)}{F(u_1v_2 - v_1u_2)} \Delta Z; \quad (4.2.13)$$

falls $u_1v_2 - v_1u_2 \neq 0$.

Die Lage der Punkte $P_i, i = 0, 1$ wird auch als Funktion von ΔZ berechnet:

$$Z_i = \frac{-F\Delta X - (x_i + u_i)\Delta Z}{u_i}; \quad \text{oder} \quad Z_i = \frac{-F\Delta Y - (y_i + v_i)\Delta Z}{v_i} \quad (4.2.14)$$

$$Y_i = -\frac{Z_i y_i}{F}; \quad X = -\frac{Z_i x_i}{F} \quad (4.2.15)$$

falls $u_i \neq 0$ oder $v_i \neq 0$:

4.3 Dynamische Orientierung im 3-D Raum

In diesem Abschnitt befassen wir uns mit der Transformation zwischen den Bildkoordinaten und den 3-D Koordinatensystemen im Raum. Wir betrachten sowohl die Transformationen zu einem diskreten Zeitpunkt, wie auch die Dynamik der Koordinatensysteme, die sich in Form von Trajektorien im Raum (bezüglich der stabilen Weltkoordinaten) ausdrückt.

4.3.1 Die Koordinatensysteme im 3-D Raum

Wir unterscheiden hier zwischen drei sich dynamisch ändernden Koordinatensystemen im 3-D Raum:

1. das *Kamerakoordinatensystem* $X^k Y^k Z^k$ mit Ursprung O^k ,
2. die *Eigenkoordinaten* $X^c Y^c Z^c$ mit Ursprung O^c ,
3. die *Fahrbahnkoordinaten* $X^w Y^w Z^w$, wobei der Ursprung O^w dieses Systems zum Teil an die Eigenfahrzeuglage gebunden ist,

Die Dynamik von all diesen Koordinatensystemen wird relativ zu einem konstanten Weltkoordinatensystem XYZ beschrieben.

Die Bildkoordinaten und Projektionsabbildung

Für die Projektion des 3-D Raumes in die Bildebene wird ein *Lochkameramodell* verwendet. Im *Brennpunkt* der Kamera wird der Ursprung des Kamerakoordinatensystems gesetzt und die Tiefenachse $O^k \vec{Z}^k$ verläuft vertikal zur Bildebene. Der Ursprung der Bildkoordinaten entspricht dem Punkt $o = (0, 0, -f)^T$ in Kamerakoordinaten, wobei f die *Brennweite* bedeutet. Die Achsen

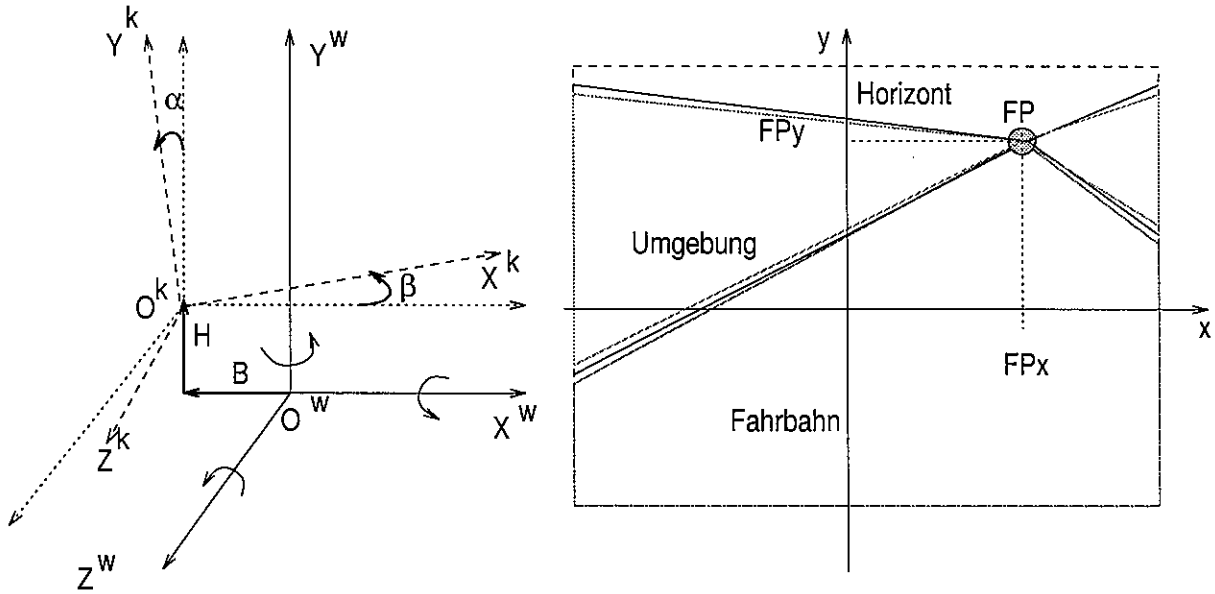


Bild 4.3.1: Die Transformationen im 3-D Raum: die Kamera- und Fahrbahnkoordinaten (links), die Detektion der Kameraneigung und Kamerarichtung aus der Lage des Fluchtpunktes FP (rechts).

der Bildkoordinaten $\vec{o}\hat{x}$ und $\vec{o}\hat{y}$ verlaufen parallel zu entsprechenden Achsen $O^k\vec{X}^k$ und $O^k\vec{Y}^k$ der Kamerakoordinaten.

Ein Punkt in Bildebene mit Bildkoordinaten $(x, y)^T$ entspricht dabei einem Punkt im 3-D Raum mit Kamerakoordinaten $(X^k, Y^k, Z^k)^T$ wie folgt:

$$X^k = x * \delta x; Y^k = y * \delta y; Z^k = -f; \quad (4.3.1)$$

wobei $\delta x, \delta y$ die Breite bzw. die Höhe eines Pixels im Bild bezeichnen.

Durch die Relation der Brennweite zur Größe der lichtempfindlichen Fläche wird der Öffnungswinkel der Kamera bestimmt. Dieser Wert ist maßgebend für den Perspektiveffekt der Abbildung. Wir setzen $F_x = f/\delta x$ und $F_y = f/\delta y$, um diese Relationenwerte in beiden Richtungen zu bestimmen und dadurch die Brennweite in Bildeinheiten auszudrücken.

Zwischen einem Punkt $P = (X^k, Y^k, Z^k)^T$ (gegeben in Kamerakoordinaten) und seiner Projektion in die Bildebene $p = (x, y)^T$ besteht folgender Zusammenhang:

$$x = -F_x \frac{X^k}{Z^k}; y = -F_y \frac{Y^k}{Z^k} \quad (4.3.2)$$

Die Fahrbahnkoordinaten

Der Ursprung der Fahrbahnkoordinaten (X^w, Y^w, Z^w) ist durch den Fahrbahnquerschnitt und durch die momentane Lage des Kamerafahrzeuges bestimmt. Der Ursprung liegt in der Mitte der Fahrbahn, auf der Fahrbahnebene und auf demselben Fahrbahnquerschnitt, wie der Ursprung der Kamerakoordinaten. Die Richtung der $O^w\vec{Z}^w$ Achse entspricht der Tangente der Fahrbahnkrümmung auf der Ebene OX^wY^w die durch den momentanen Koordinatenursprung verläuft.

Zu jedem Zeitpunkt t ist die Transformation des Fahrbahnkoordinatensystems $X^wY^wZ^w$ in das Kamerakoordinatensystem $X^kY^kZ^k$ durch das Aneinanderreihen von folgenden Transformationsschritten gegeben (Bild 4.3.1(a)):

- Verschiebung des Ursprungs entlang der Achse Y von $Y^w = 0$ nach $Y^w = H$ (H ist die Höhe des Bildursprungs über der Fahrbahn).
- Verschiebung des Ursprungs entlang der Achse X von $X^w = 0$ nach $X^w = B$ (B ist die horizontale Entfernung des Bildursprungs von der Fahrbahnmitte).
- Rotation bezüglich der Y Achse um Winkel β
- Rotation bezüglich der X Achse um Winkel α
- Rotation bezüglich der Z Achse um Winkel γ

Die Transformation der Fahrbahnkoordinaten in die Kamerakoordinaten eines konkreten Punktes im Raum entspricht derselben Abfolge von Einzeltransformationen, aber mit umgekehrten Zeichen. Sie setzt sich zusammen aus entsprechenden Verschiebungen um $-H$ und $-B$, sowie Rotationen um $-\beta$, $-\alpha$ und $-\gamma$.

Der Winkel γ beschreibt in der Praxis das seitliche Schaukeln des Fahrzeuges, das im weiteren Verlauf der Arbeit vernachlässigt wird. Diese Vereinfachung ist insofern zulässig, daß der somit eventuell eingeführte Fehler bei relativ langsam befahrenen Kurven relativ klein im Vergleich zu anderen Fehlerquellen ist. Anders als bei den zwei verbleibenden Winkeln α und β findet sich hier auch keine robuste Methode der Bildanalyse, welche diesen Winkelbetrag automatisch messen könnte. Unter der Annahme $\gamma = 0$ ist die Koordinatentransformation eines Punktes (von Fahrbahn- in Kamerakoordinaten) wie folgt gegeben:

$$\begin{pmatrix} X^k \\ Y^k \\ Z^k \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-\alpha) & -\sin(-\alpha) & 0 \\ 0 & \sin(-\alpha) & \cos(-\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(-\beta) & 0 & \sin(-\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\beta) & 0 & \cos(-\beta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -B \\ 0 & 1 & 0 & -H \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{pmatrix}$$

Dies wird ausmultipliziert zu folgender Form:

$$X^k = \cos(\beta)X^w - \sin(\beta)Z^w - B \cos(\beta) \quad (4.3.3)$$

$$Y^k = \sin(\alpha) \sin(\beta)X^w + \cos(\alpha)Y^w + \sin(\alpha) \cos(\beta)Z^w - B \sin(\alpha) \sin(\beta) - H \cos(\alpha) \quad (4.3.4)$$

$$Z^k = \cos(\alpha) \sin(\beta)X^w - \sin(\alpha)Y^w + \cos(\alpha) \cos(\beta)Z^w - B \cos(\alpha) \sin(\beta) + H \sin(\alpha) \quad (4.3.5)$$

Eine Rücktransformation des Kamerakoordinatensystems in das Fahrbahnkoordinatensystem ist durch die inverse Transformation gegeben. Im Falle der Rücktransformation der Koordinaten eines Punktes (d.h. Transformation der Kamerakoordinaten- in Fahrbahnkoordinaten) ist sie folgendermaßen definiert:

$$\begin{pmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & B \\ 0 & 1 & 0 & H \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X^k \\ Y^k \\ Z^k \\ 1 \end{pmatrix}$$

Die ausmultiplizierte Form ist wie folgt:

$$X^w = X^k \cos(\beta) + Y^k \sin(\alpha) \sin(\beta) + \cos(\alpha) \sin(\beta)Z^k + B \quad (4.3.6)$$

$$Y^w = Y^k \cos(\alpha) - Z^k \sin(\alpha) + H \quad (4.3.7)$$

$$Z^w = -X^k \sin(\beta) + Y^k \sin(\alpha) \cos(\beta) + Z^k \cos(\alpha) \cos(\beta) \quad (4.3.8)$$

Die Eigenkoordinaten

Die Eigenkoordinaten bilden eine Zwischenstufe während der Transformation von Kamerakoordinaten in die Fahrbahnkoordinaten. Die Eigenkoordinaten unterscheiden sich von den Kamerakoordinaten nur in der Neigung zur Fahrbahnebene und der Höhe über der Fahrbahn. Die Überführung der Fahrbahnkoordinaten in die Eigenkoordinaten besteht aus folgenden Schritten:

- Verschiebung des Ursprungs entlang der Achse X von $X^w = 0$ nach $X^c = B$ (B ist die horizontale Entfernung des Bildursprungs von der Fahrbahnmitte)
- Rotation bezüglich der Y Achse um Winkel β

Dies entspricht einer vereinfachten Transformation zwischen der Fahrbahn und der Kamera, bei der $H = 0$ und $\alpha = 0$.

4.3.2 Die Rotationswinkel

Die momentanen Werte der Rotationswinkel α und β , der Kameraorientierung gegenüber der Fahrbahn, werden aus den Bildkoordinaten des *geometrischen Fluchtpunktes* $FP(t)$ gewonnen (Bild 4.3.1(b)). Im Bildpunkt FP schneiden sich die Bildgeraden auf denen die Projektionen der Straßenbegrenzungen und der Mittelstreifen liegen. Bei signifikanter Fahrbahnkrümmung wechseln die relevanten Geraden in Tangenten der Straßenmarkierungen unmittelbar vor dem Fahrzeug. Der Winkel α entspricht der Neigung der Kamera zur Fahrbahnebene und β dem Winkel, in dem die Kamera seitlich geschwenkt wird, bezüglich der Tiefenachse der Fahrbahn. Diese Winkel werden geschätzt wie folgt:

$$\alpha(k) = -\arctan\left(\frac{FP_y(k)}{F_y}\right); \quad \beta(k) = \arctan\left(\frac{FP_x(k)}{F_x}\right) \quad (4.3.9)$$

Der Winkel γ , in dem die Kamera zur Seite gedreht ist, könnte aus der Neigung des Horizonts im Bild bestimmt werden. Bei stark strukturierter Landschaft und mehreren bewegten Objekten ist der Horizontverlauf aber sehr schwer zu ermitteln. Wie bereits erwähnt wird sein Einfluß auf die Ergebnisse in dieser Arbeit vernachlässigt.

4.3.3 Die zeitlichen Trajektorien auf Ebene

Die Transformationsparameter zwischen den Fahrbahnkoordinaten und den Eigenkoordinaten unterliegen dynamischen Änderungen. Der Grund dafür sind die unterschiedlichen Verläufe der Fahrbahn und der Richtung des Eigenfahrzeuges. Dies wird in (Bild 4.3.2) in Form von zwei Trajektorien der zwei Koordinatensysteme – Fahrbahnkoordinaten und Eigenkoordinaten – dargestellt:

- die Trajektorie des Kamerafahrzeuges mit Krümmungszentrum C^c und Radius R^c (die Eigentrajektorie)
- die Trajektorie des Ursprungs des Fahrbahnkoordinatensystems mit Krümmungszentrum C^w und Radius R^w

Die virtuellen Geschwindigkeiten (Betrag der translatorischen Geschwindigkeit und die Rotationsgeschwindigkeit) mit denen der Ursprung der Fahrbahnkoordinaten sich bewegt sind das Ergebnis der Fahrbahnkrümmung und der Kamerafahrzeugbewegung.

4.4 Die Fahrbahnerkennung

Die Fahrbahnerkennung hat zum Ziel eine schritthaltende und adaptive Bestimmung von folgenden Parametern:

- die Detektion der Fahrbahnklasse, d.h. Erkenntnisse über die Anzahl der Fahrbahnsuren,

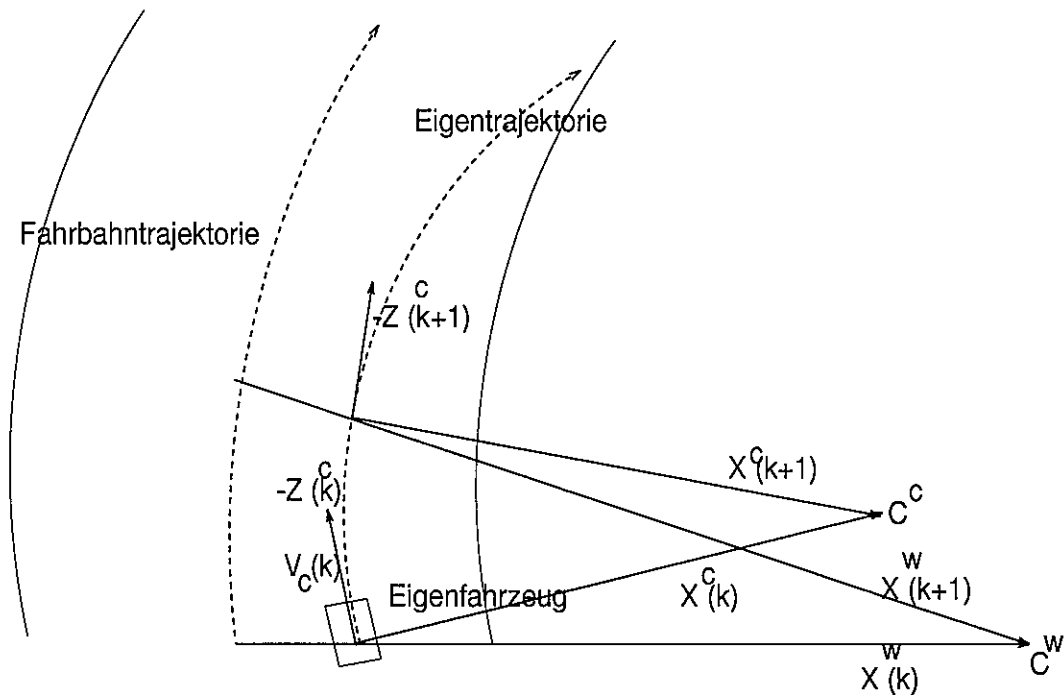


Bild 4.3.2: Die Eigentrajektorie und Fahrbantrajektorie auf Fahrbannebene.

- die adaptive Schätzung der Fahrbahnbreite $Breite(k)$ und der Entfernung der Fahrbahnmittle von dem Ursprung der Eigenkoordinaten, je nach Fahrbahnklasse,
- die Verfolgung der Fahrbahnrichtung in Weltkoordinaten (dies entspricht der adaptiven Schätzung der Fahrbahnkrümmung).

4.4.1 Die Fahrbahnthesen

Es werden drei Fahrbahnthesen parallel verfolgt: T_2, T_3 und T_4 , welche die Klassen der 2-, 3- und 4-spurigen Fahrbahn entsprechend modellieren. Der Zustandsvektor einer Hypothese T_i ($i = 2, 3, 4$) ist wie folgt:

$$s_{T_i}(k) = [W_i(k), B_i(k), \delta B_i(k), \omega_{F_i}]^T \quad (4.4.1)$$

Dabei ist W_i die Breite der Fahrbahn und B_i die Translation entlang der X-Achse zwischen der Mittellinie der Fahrbahn und dem Ursprung der Eigenkoordinaten O^c . δB_i beschreibt die Dynamik von B_i und ω_{F_i} ist die Rotationsgeschwindigkeit die indirekt die Fahrbahnkrümmung ausdrückt.

Das verwendete Modellwissen über die Fahrbahn enthält folgende Angaben:

- die minimale und maximale Breite einer Fahrspur
- die minimale und maximale Breite einer Fahrbahnmarkierung am linken und rechten Rand oder innerhalb der Fahrbahn

Bei der Vorhersage des aktuellen Zustandes $s_{T_i}^*(k)$ geht man von unveränderten Parameterwerten aus, bis auf $B_i^+(k)$ das sich entsprechend der momentanen Dynamik $\delta B_i^+(k-1)$ ändert.

4.4.2 Detektion (Messung) der Parameter W, B

Als erstes beschreiben wir die direkte Detektion der Fahrbahnbreite und der momentanen Lage der Kamera bezüglich der Fahrbahnmitte für jede Fahrbahnhypothese, sowie die Bewertung der einzelnen Messungen. Die bisherigen Fahrbahnypothesen haben keinen Einfluß auf die einzelnen Messungen.

Liniensegmente von potentiellen Fahrbahnstreifen

Eine initiale Kamera-zur-Straße-Transformation kann aufgrund der momentanen Fluchtpunktschätzung berechnet werden (d.h. die Winkel α und β), bei bekanntem H und unter Annahme $B = 0$.

Solche Konturen, die zur Klasse *Fahrbahn* gehören und die früher ermittelten FP-Segmente enthalten, werden als potentielle Fahrbahnmarkierungen detektiert (vgl. Konturklassifikation) (Bild 4.4.1(a, b)).

Die FP-Segmente und die Konturschwerpunkte werden durch Anwendung der initialen Kamera-zur-Straße-Transformation in die initialen Fahrbahnkoordinaten umgerechnet, unter der Annahme, daß die Konturpunkte auf der Fahrbahnebene liegen (Bild 4.4.1(c, d)).

Häufungspuffer entlang des Fahrbahnquerschnittes

Jedes ausgewählte Liniensegment erhält eine Gewichtung, die proportional zur Länge des Segmentes entlang der Tiefenachse ist und umgekehrt proportional zur Tiefenlage ist. Der zugehörige Grauwert des Segmentes fließt auch in die Gewichtung hinein. Anschliessend werden diese Liniensegmente auf einen Fahrbahnquerschnitt parallel zurückprojiziert und ihre Gewichte in entsprechenden Feldern des Häufungsvektors aufaddiert (Bild 4.4.2).

Gitterabgleich im Häufungspuffer

Die vorherige Messung ist für jede Fahrbahnhypothese gleich und wird deswegen nur einmal durchgeführt. Jetzt folgt ein Abgleich zwischen einem Modellgitter und dem Häufungspuffer, der für jede Fahrbahnhypothese mit einem entsprechendem Modellgitter wiederholt wird. Für die Hypothese T_i ($i = 2, 3, 4$) wird ein Gitter mit $i + 1$ Modellstreifen eingesetzt. Es wird für jeden Modellstreifen ein bester Abgleich mit einem Paar von Häufungssäulen im Häufungspuffer bestimmt. Das als individuelle Messung detektierte Meßgitter besteht aus $i + 1$ Meßstreifen, die insgesamt am besten bewertet wurden (Bild 4.4.3).

Da die Meßstreifen diskreten Koordinaten der X -Achse entsprechen kann aus dem Gitter direkt sowohl die Fahrbahnbreite wie auch die Lage der Kamera bezüglich der Mitte bestimmt werden.

Gewichtete Messung und Varianz der Messung

Die individuellen Messungen von B_i und W_i ($i = 2, 3, 4$) im aktuellen Bild und in bis zu 4 unmittelbar vorherigen Bildern werden zum Schluß der direkten Messung zu gewichteten Meßwerten zusammengefaßt.

Die Varianz der individuellen Messungen innerhalb dieser kurzen Verfolgung bestimmt die aktuelle Meßvarianz. Bevor sie aber weitergegeben wird sind noch andere Unsicherheiten des Meßprozesses zu berücksichtigen. Ein fehlender Streifen im aktuellen individuellen Gitter trägt zur größeren Unsicherheit durch aufaddieren einer "Strafvarianz". Insbesondere werden die Randstreifen schlechter bewertet, die außerhalb der momentanen Breite des Häufungsvektors

herausragen und deswegen nur hypothetisiert wurden. Dieses Vorgehen allein würde aber eine Hypothese mit wenigeren Streifen stets bevorzugen. Als Ausgleich dazu werden die im 2-D Modul gefundene Fahrbahnregion und die mit ihr verbundenen Werte der minimalen Fahrbahnausprägung herangezogen. Für jeden Randstreifen, der innerhalb dieser minimalen Werte liegt, wird auch eine "Strafvarianz" aufaddiert.

4.4.3 Erste Modifikation und Fahrbahnauswahl

Die direkt gemessenen Parameter W_i, B_i tragen zur Modifikation dieser Werte im Zustandsvektor s_{Fi} bei.

Die Varianzen von allen drei Hypothesen, gewichtet durch jeweilige Fahrbahnbreite, werden jetzt zur Auswahl der besten Fahrbahnhypothese genommen. Die Schätzwerte der momentan besten Hypothese werden zur weiteren Verarbeitung genommen (Bild 4.4.4). Dadurch ist die aktuelle Schätzung einer kompletten Kamera-zur-Fahrbahn-Transformation gegeben.

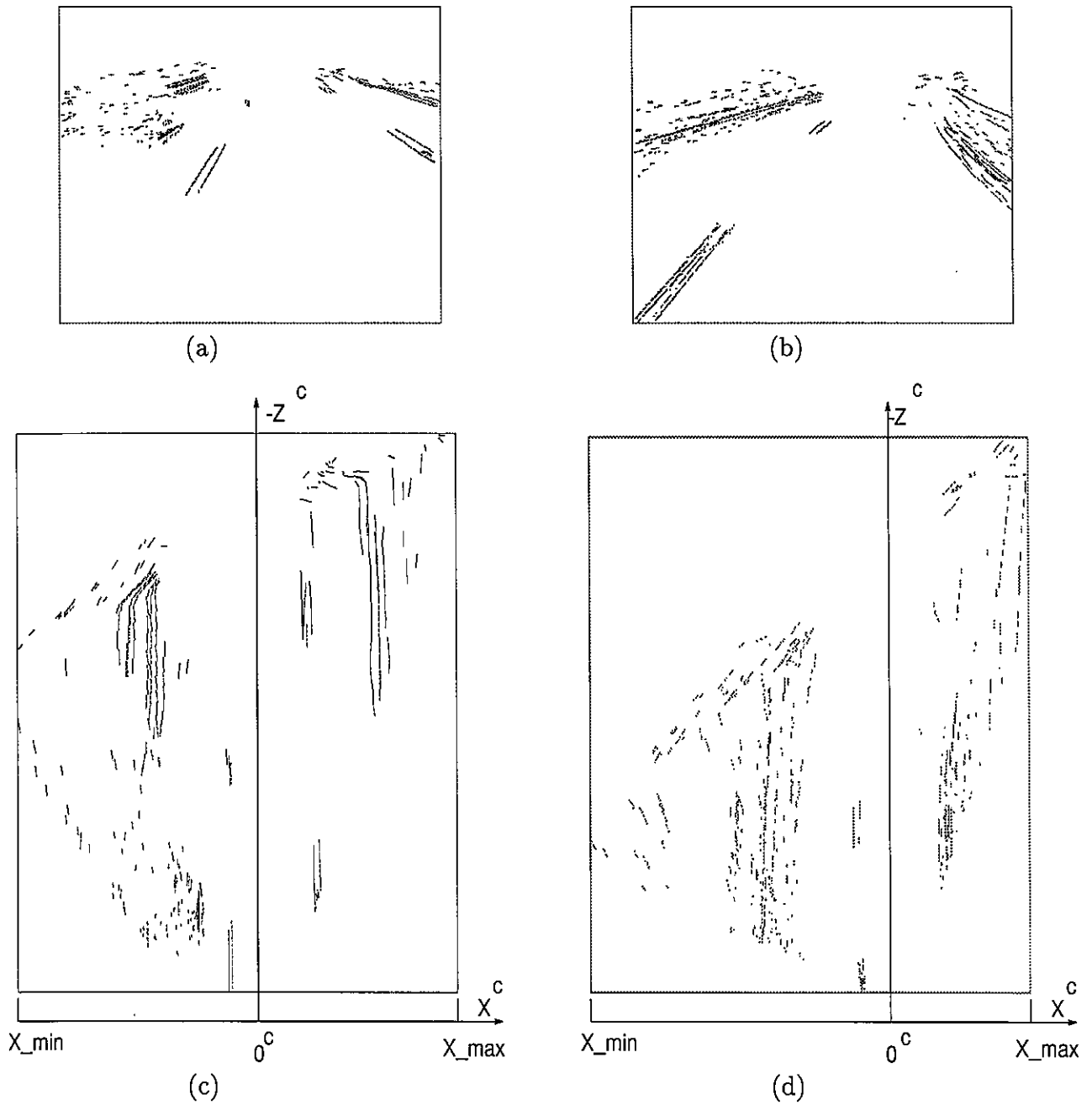


Bild 4.4.1: Liniensegmente die zur Fahrbahnerkennung ausgewählt wurden (a, b) und ihre Rückprojektion auf Fahrbahnebene (Draufsichten) (c, d) – (a,c) geradlinige Fahrbahn, (b,d) Kurvenfahrt.

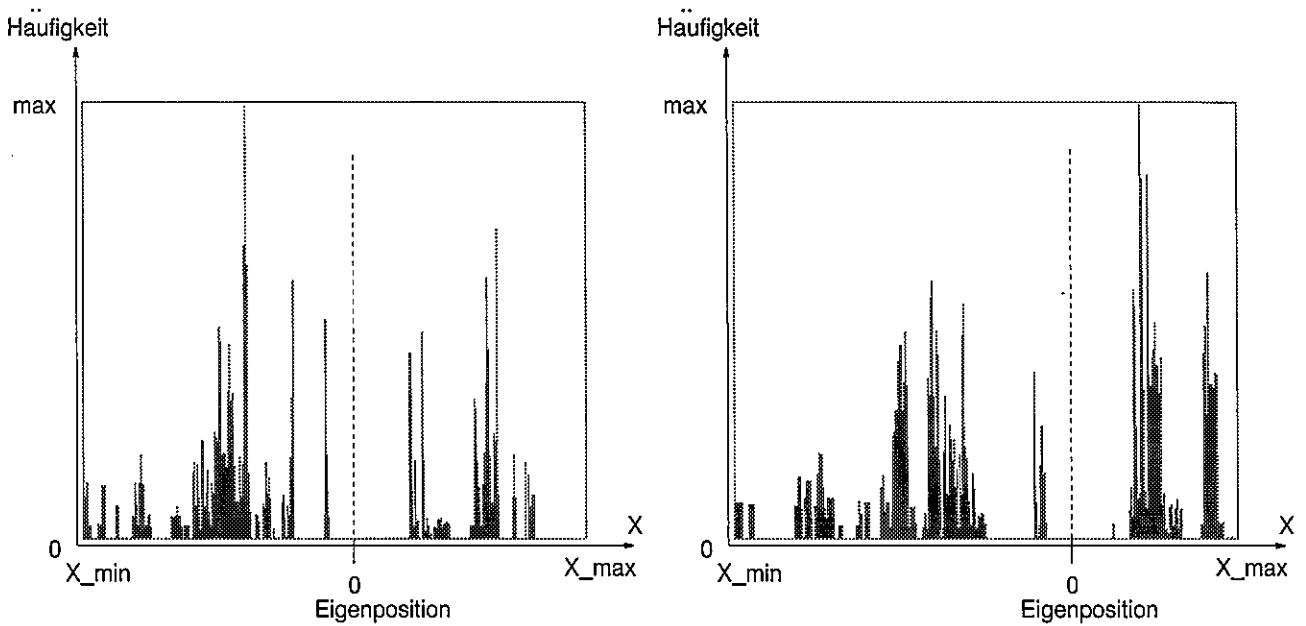


Bild 4.4.2: Häufungshistogramme für einen Querschnitt der rückprojizierten Fahrbahnmarkierung (entlang der X^w -Achse): bei geradliniger Fahrbahn (links) und bei Kurvenfahrt (rechts).

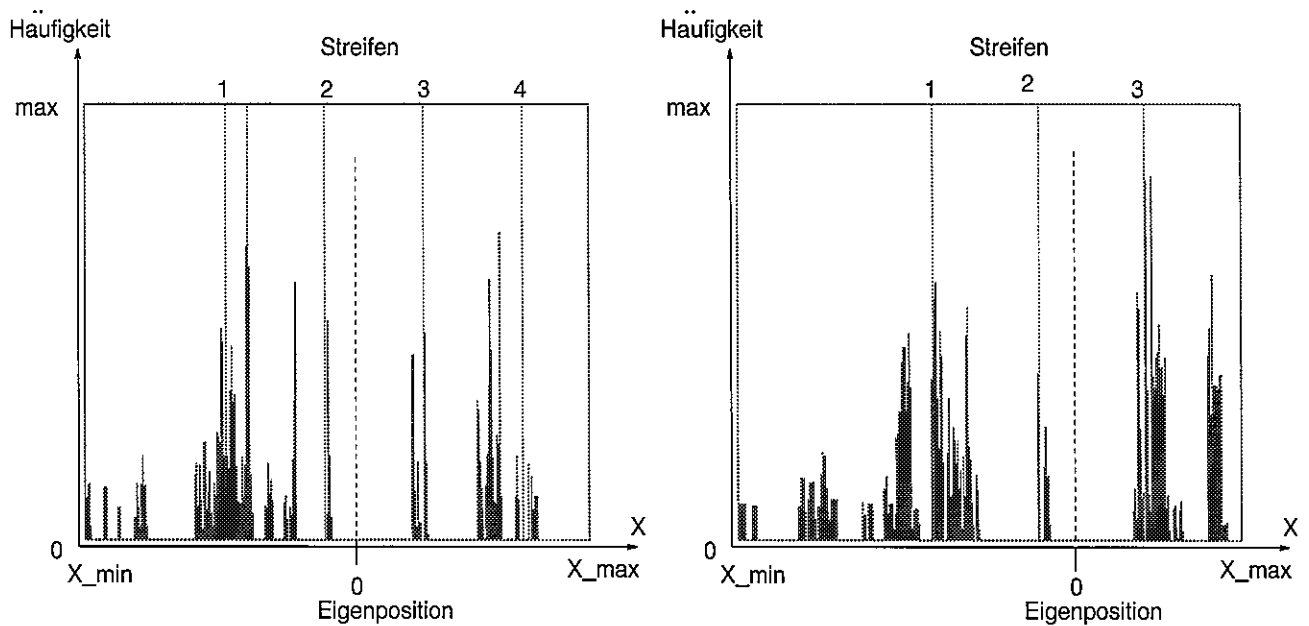


Bild 4.4.3: Die Gitter der Fahrbahnypothesen (ein Gitterelement markiert die Lage einer Fahrbahnmarkierung entlang der X^w -Achse: für 3-spurige Fahrbahn (links) und 2-spurige Fahrbahn (rechts)).

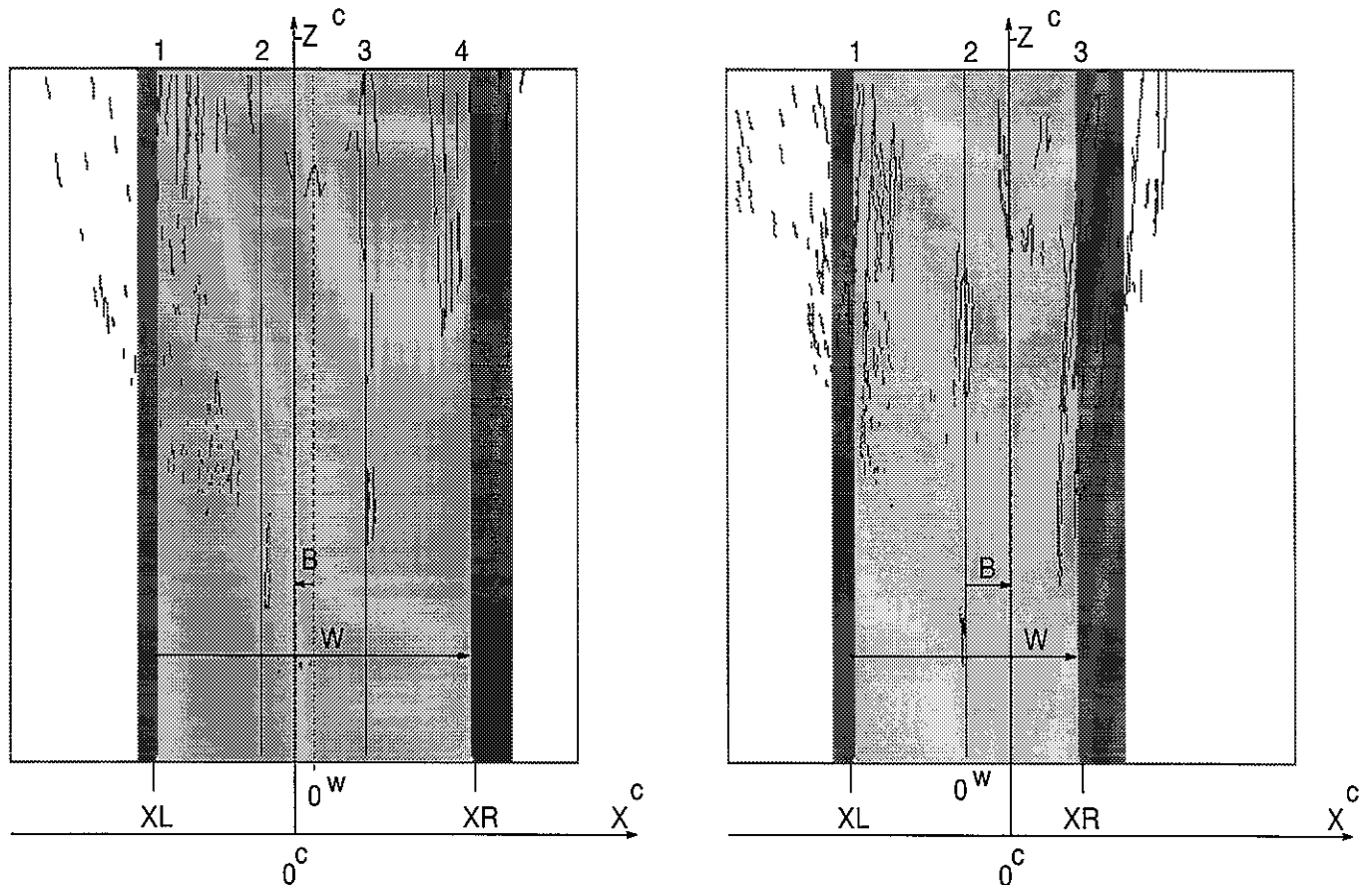


Bild 4.4.4: Draufansicht der stabilisierten Fahrbahnparameter (Breite W , Translation B und Streifenanzahl) für die erkannte Fahrbahnklasse: bei geradliniger Fahrbahn (links) und bei Kurvenfahrt (rechts).

4.4.4 Indirekte Parameter – Dynamik und Fahrbahnkrümmung

In der Literatur sind zwei folgende Ansätze zur Fahrbahnkrümmungsdetektion bekannt:

1. Eine modellunabhängige Analyse von verschiedenen Fluchtpunkten im Bild. Diese Fluchtpunkte werden als Häufungsbereiche von FP-Kanten ermittelt, die sich in einzelnen horizontalen Querschnittssektionen des Bildes befinden ([PJ92]).
2. Ein modellgesteuerter Ansatz zur Approximierung der detektierten Fahrbahnbegrenzungskanten durch Kurven zweiter Ordnung ([ST92]).

Die oben genannten Ansätze führen zum Erfolg nur dann wenn eine gute Sicht von weit entfernten Fahrbahnrandern möglich ist. Dies ist aber nur dann gewährleistet wenn zum einen keine Fremdobjekte vorkommen (die eine massive Verdeckung dieser kanten im verursachen können) und zum anderen die Kamera relativ hoch über der Fahrbahn positioniert ist, um die gewöhnlich lang gezogenen Kurven auch gut ins Bild zu bekommen.

In dieser Arbeit wurde ein anderer Ansatz entwickelt, der auf der Verfolgung von zwei Trajektorien basiert – der Eigentrajektorie und der Trajektorie des Fahrbahnursprungs. Dabei können zwei Berechnungsmethoden eingesetzt werden:

1. Die Krümmung wird abgeleitet aus der Verfolgung eines einzigen Fluchtpunktes bei Kenntnis der eigenen Winkelgeschwindigkeit ω_{own} :

$S = 0$
 FOR ($i = t - 9, t - 8, \dots, t$) DO $S = S + \omega_{ego}(i)$

$$Krümmung(t) = \frac{S - [\beta(t) - \beta(t - 9)]}{10} \quad (4.4.2)$$

- Die Krümmung wird abgeleitet aus der Verfolgung der Eigenlage B relativ zur Fahrbahnmitte bei Kenntnis der Eigenwinkelgeschwindigkeit ω_{ego} .

Die Eigenwinkelgeschwindigkeit $\omega_{ego}(t)$ wird aus den Fahrzeugmeßdaten – translatorische Eigengeschwindigkeit V_{ego} und Lenkeinschlagswinkel der Vorderräder δ_{ego} – gewonnen. Für kleine δ_{ego} gilt nämlich

$$\delta_{ego} = \frac{L}{Radius}, \quad (4.4.3)$$

wobei L die Distanz zwischen Vorder- und Hinterachse ist, z.B. $L \simeq 2.6[m]$. Aus

$$\omega_{ego} = \frac{V_{ego}}{Radius}$$

folgt dann

$$\omega_{ego} = \frac{V_{ego} * \delta_{ego}}{L} \quad (4.4.4)$$

4.5 Der Eigenzustand

Der sogenannte *Eigenzustand* besteht aus zwei sich dynamisch ändernden Zustandsvektoren:

- Der Vektor der *Eigenbewegung*, der die Bewegung des Eigenfahrzeuges entlang der Eigen trajektorie auf Fahrbahnebene beschreibt. Er ist wie folgt gegeben:

$$s_c^d(k) = [O_X(k), O_Z(k), \Theta_c(k), V_c(k), \omega_c(k)]^T, \quad (4.5.1)$$

wobei (O_X, O_Z) sind die Koordinaten des Ursprungs O_C des Eigenkoordinatensystems auf Fahrbahnebene (ausgedrückt in Weltkoordinaten), Θ_c ist der Richtungswinkel der Z^c -Achse (in Weltkoordinaten), V_c ist der Betrag der translatorischen Geschwindigkeit und ω_c ist die Winkelgeschwindigkeit des Kamerafahrzeuges.

- Der aktuelle Transformationsvektor zwischen den Kamerakoordinaten und den Fahrbahnkoordinaten wird zusammengefaßt zu:

$$s_T(k) = [H, B(k), \alpha(k), \beta(k)]^T \quad (4.5.2)$$

Die Kenntnis des Eigenzustandes ist eine generelle Voraussetzung für die modellbasierte Erkennung von 3-D Objekten. Aber auch bei der Verschiebungsschätzung im Bild wird er berücksichtigt.

Die dynamische Eigentransformation

Es wird eine initiale Lage des Ursprungs und ein initialer Richtungswinkel vorausgesetzt. Die Geschwindigkeiten V_c, ω_c sollen mit Hilfe des Fahrzeugmodells und der aktuellen Eigenfahrzeugdaten (Geschwindigkeit, Lenkradwinkel) außerhalb des Erkennungssystems ermittelt werden. Was das dynamische Fahrzeugmodell anbetrifft, so muß berücksichtigt werden, daß die Tangente der Kreiskurve im Schwerpunkt des Fahrzeuges (dieser Punkt beschreibt einen Kreis mit Bahnradius *Radius*) mit der Fahrzeuginnenachse den Schwimmwinkel γ bildet. Bei langsamer Kurvenfahrt, d.h. grosser Bahnradius gegenüber hinterem Radstand LH , laesst sich der Schwimmwinkel γ approximieren durch: $\gamma = LH/R$.

Seien jetzt mit $P^c(k) = (P_X^c(k), P_Y^c(k), P_Z^c(k))^T$ die Eigenkoordinaten eines Punktes P im 3-D Raum zum Zeitpunkt t_k bezeichnet. Auf Grund der Eigenbewegung erfolgt folgende Koordinatentransformation des unbewegten Punktes im Zeitintervall $\tau = t_{k+1} - t_k$ zwischen den diskreten Zeitpunkten t_k und t_{k+1} ,

- Translation um Vektor $(-l_X^c(k), -l_Y^c(k), -l_Z^c(k))^T$

$$l_X^c(k) = -\frac{V_c(k)}{\omega_c(k)}(1 - \cos(\omega_c(k)\tau)) \quad (4.5.3)$$

$$l_Y^c(k) = 0 \quad (4.5.4)$$

$$l_Z^c(k) = -\frac{V_c(k)}{\omega_c(k)}(\sin(\omega_c(k)\tau)) \quad (4.5.5)$$

- und Rotation um Winkel $-\omega_c\tau$ um die Y^c Achse

Diese Transformation wird weiterhin als die *dynamische Eigentransformation* (kurz $DET(k)$) bezeichnet:

$$\begin{pmatrix} p_X^c(k+1) \\ p_Y^c(k+1) \\ p_Z^c(k+1) \\ 1 \end{pmatrix} = [DET(k)] \begin{pmatrix} p_X^c(k) \\ p_Y^c(k) \\ p_Z^c(k) \\ 1 \end{pmatrix}$$

In ausmultiplizierter Form und nach Vereinfachung ergibt sich folgende Form der Transformation $DET(k)$: (die k -s auf der rechten Seite der Gleichungen wurden jetzt aus Gründen der Übersichtlichkeit weggelassen):

$$p_X^c(k+1) = p_X^c \cos(\omega_c\tau) - p_Z^c \sin(\omega_c\tau) + \frac{V_c}{\omega_c}(\cos(\omega_c\tau) - 1) \quad (4.5.6)$$

$$p_Y^c(k+1) = p_Y^c \quad (4.5.7)$$

$$p_Z^c(k+1) = p_X^c \sin(\omega_c\tau) + p_Z^c \cos(\omega_c\tau) + \frac{V_c}{\omega_c}(\sin(\omega_c\tau)) \quad (4.5.8)$$

Eine genaue Herleitung der Transformation $DET(k)$ findet sich im Anhang 9.1.

Kapitel 5

Modellbasierte Objekterkennung

In diesem Kapitel wird das Objektmodell (Abschnitt 5.1) und das mit ihm verbundene 3–D Modul detailliert erläutert (Abschnitte 5.2–5.6) Das 3–D Modul zur modellbasierten, adaptiven 3–D Objekterkennung (Bild 5.1.1) enthält folgende Schritte: Generierung von Objekthypothesen (Abschnitt 5.2), Verfolgung von Objekthypothesen (Abschnitte 5.3–5.5) und Objektauswahl zwischen konkurrierenden Hypothesen (Abschnitt 5.6).

Während der Verfolgungsphase wird der Zustand jedes vorher initialisierten Objektes rekursiv aufgefrischt, indem ein Abgleich zwischen projizierten Modellprimitiva des vorhergesagten Zustandes und den neuen Messungen im Bild vorgenommen wird. Die Objektverfolgung wird durch die dynamische Übergangsfunktion (Vorhersage) (Abschnitt 5.3), die Modifikation des Zustandes (Abschnitt 5.4) und die Projektionsabbildung und Meßwerterfassung (Abschnitt 5.5) beschrieben.

5.1 Das Objektmodell

Wie bereits in Abschnitt 2.3 eingeführt wird jede Objekthypothese einer von mehreren vorgegebenen Objektklassen zugeordnet und durch ihren Zustandsvektor s eindeutig beschrieben. Dieser Vektor besteht aus zwei Teilen – einem *dynamischen Vektor* s^d , der die Trajektorie des Objektes auf Fahrbahnebene beschreibt und einem *Formvektor* ξ , der im Prinzip stabile Formparameter enthält:

$$s(k) = [s^d(k), \xi(k)] \quad (5.1.1)$$

Im allgemeinen Fall hätten wir es mit 12 Parametern der Lage, Bewegung und Größe einer bekannten Objektform zu tun. Die Objektformen innerhalb einer Klasse, z.B. Klasse *PKW*, können aber selbst noch variieren. Dadurch kämen noch weitere Parameter zum Zustandsvektor dazu. Um den Aufwand auf einem akzeptablem Niveau zu halten, sowie die Komplexität des Modells den tatsächlich im Bild detektierbaren Merkmalen anzupassen haben wir für unseren Problembereich die Anzahl der Freiheitsgrade verringert und dadurch auch der Aufwand für die Intialisierung und rekursive Schätzung des Zustandsvektors der Objekte. Die wesentliche Vereinfachung beruht auf folgenden Annahmen über die Fahrbahn und Bewegungen:

- Die Fahrbahnoberfläche wird durch eine Ebene modelliert
- Die bewegten Objekte stehen und bewegen sich nur auf der Fahrbahn
- Die Eigenbewegung des Kamerafahrzeugs findet auch nur auf der Fahrbahn statt
- Die translatorische Verschiebungsrichtung überdeckt sich mit der momentanen Orientierung des Objektes im Raum.

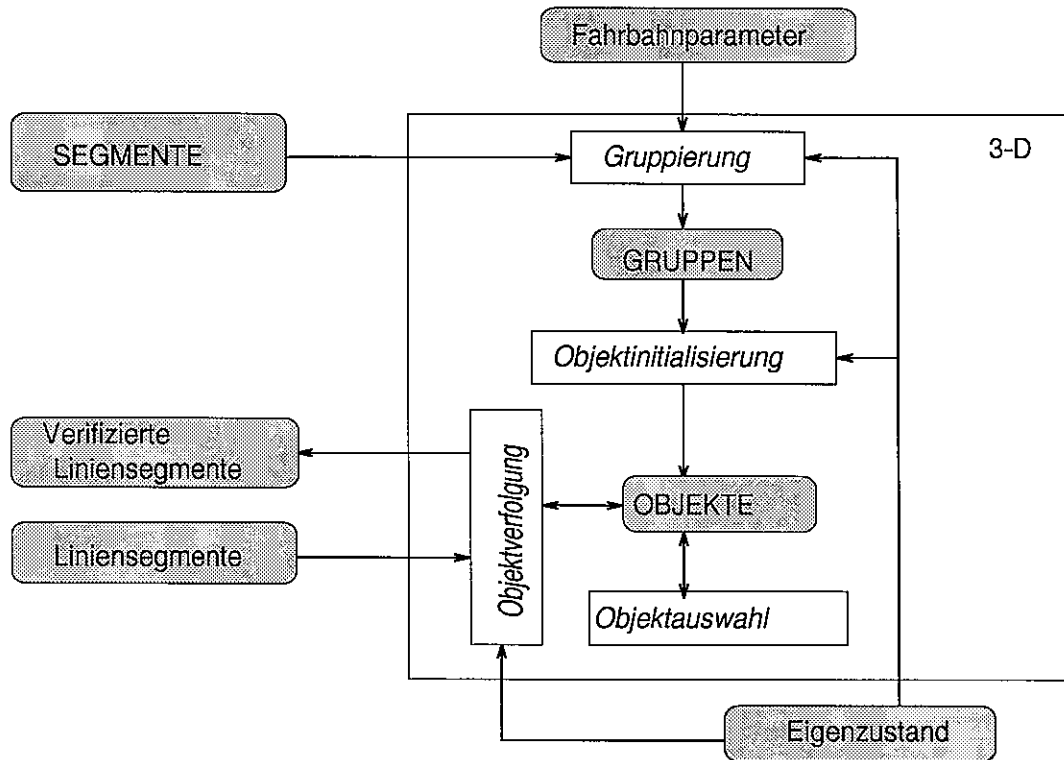


Bild 5.1.1: Das Modul zur modellbasierten Objekterkennung

5.1.1 Die Trajektorien

Wir definieren den dynamischen Vektor $s^d(k)$ des Objektzustandes zum Zeitpunkt des k -ten Bildes einer Bildfolge als einen fünfstelligen Vektor:

$$s^d(k) = [p_X(k), p_Z(k), \Theta(k), V(k), \omega(k)]^T \quad (5.1.2)$$

wobei $V = |\Delta p| = \sqrt{(\Delta p_X)^2 + (\Delta p_Z)^2}$ den Wert der translatorischen Bewegung in der Ebene $0XZ$ entlang der Richtung Θ bedeutet und ω die Winkelgeschwindigkeit bezeichnet.

Die einzelnen Parameter des dynamischen Vektors können je nach Objektklasse unterschiedliche Bedeutung haben. Wir unterscheiden im folgenden zwei grundsätzliche Objekttypen, was den dynamischen Teil anbetrifft: unbewegte (dynamischer Vektor s_S) und bewegte Objekte (dynamischer Vektor s_B).

Der dynamische Vektor eines unbewegten Objektes auf der Fahrbahnebene ist wie folgt:

$$s_S^d(k) = [p_{SX}^c(k), p_{SZ}^c(k), \Theta_S^c(k), V_S(k), \omega_S(k)]^T \quad (5.1.3)$$

Dabei ist zu beachten, daß die drei ersten Parameter stets in den Eigenkoordinaten (d.h. den Eigenfahrzeugkoordinaten) spezifiziert sind.

Der dynamische Vektor eines bewegten Objektes besteht aus folgenden Komponenten:

$$s_B^d(k) = [p_{BX}^c(k), p_{BZ}^c(k), \Theta_B(k), V_B(k), \omega_B(k)]^T \quad (5.1.4)$$

wobei hier nur die Lageparameter in den Eigenkoordinaten gegeben sind, aber der Richtungswinkel in globalen Weltkoordinaten spezifiziert ist.

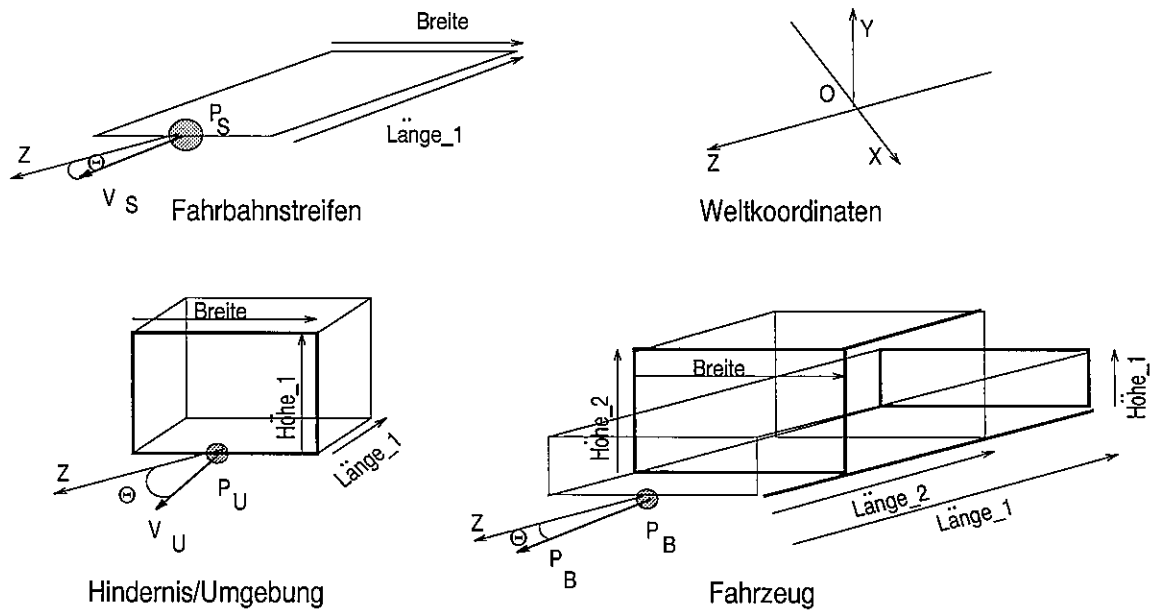


Bild 5.1.2: Drei Klassen von modellierten Objekten : (a) Fahrbahnstreifen, (b) Hindernis oder Umgebungsobjekt, (c) Fahrzeug

5.1.2 Die Objektformen

Die Komponenten des Formvektors einer Objekthypothese sind die Breite Br und mehrere Parameter κ_i :

$$\xi(k) = [Br(k), \kappa_1(k), \kappa_2(k), \dots, \kappa_j(k)]^T \quad (5.1.5)$$

Die Anzahl und Bedeutung der Komponenten κ_i hängt von der Spezialisierung der Formrepräsentation und von der Objektklasse ab. Es gibt drei Objektformen in der allgemeinen Modellfamilie *Streifen*, *Hindernis/Umgebung* und *Fahrzeug*. Die erste Form wird nur durch zwei Formparameter bestimmt – Breite und Länge (also mit Hilfe von Br und κ_1). Ein Hindernisobjekt oder Umgebungsobjekt wird nur durch seine 3-D Hülle beschrieben, d.h. drei Formparameter – *Breite*, *Länge₁* und *Höhe₁* (also mit Br , κ_1 und κ_2). Ein Fahrzeugobjekt wird durch zwei Quader modelliert mit gleicher Breite – d.h. mit fünf Formparametern: Breite, Länge 1 und 2, Höhe 1 und 2 (also Br und vier κ -s) (Bild 5.1.2).

ξ_S bezeichnet den Formvektor einer Objekthypothese für einen Fahrbahnstreifen:

$$\xi_S = [Br_S, \kappa_1] \quad (5.1.6)$$

Br_S ist die Breite der 3-D umschreibenden Hülle. Die Länge der 3-D Hülle ergibt sich aus $Länge = \kappa_1 Br_S$. Die *Höhe* wird durch einen konstanten (niedrigen) Wert angenommen.

Der Formvektor ξ_U für ein Hindernis oder ein Umgebungsobjekt ist ein dreistelliger Vektor:

$$\xi_U = [Br_U, \kappa_1, \kappa_2] \quad (5.1.7)$$

Br_U bedeutet die Breite der 3-D Objekthülle. Die Länge und Höhe dieser Hülle sind proportional zur Breite: $Länge = \kappa_1 Br_U$, $Höhe_1 = \kappa_2 Br_U$ und $Höhe_2 = \kappa_4 Br_U$.

ξ_B bezeichnet den Formvektor der Objekthypothese für ein bewegtes Fahrzeug:

$$\xi_B = [Br_B, \kappa_1, \kappa_2, \kappa_3, \kappa_4] \quad (5.1.8)$$

Br_B bedeutet die Breite des Objektes. Die vier restlichen Formparameter sind proportional zur Breite: $Länge_1 = \kappa_1 Br_B$, $Länge_2 = \kappa_2 Br_B$.

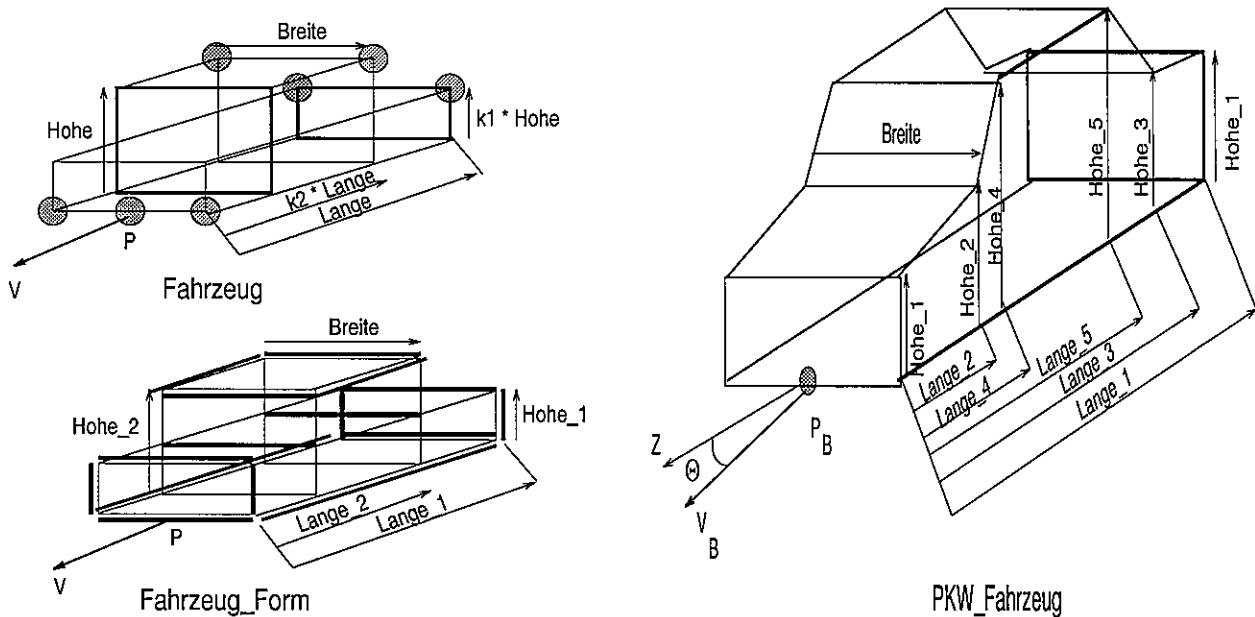


Bild 5.1.3: Drei Spezialisierungsstufen der *Fahrzeug*-Modelle am Beispiel eines PKW-Objektes

5.1.3 Spezialisierte Formen

Neben der drei allgemeinen Formen, die als Objektklasse *Verfolgung* zusammengefaßt werden können, sind zwei weitere Spezialisierungsstufen im Modell vorgesehen – die als Klassen *Verfolgung_Form* und *Verfolgung_Fein* beschrieben werden können. Die drei Spezialisierungsstufen eines Objektes *PKW* sind in Bild 5.1.3 dargestellt.

Der Unterschied zwischen den Objektmodellen der Klasse *Verfolgung* und *Verfolgung_Form* liegt in der Aktivierung dieser Modelle für die Objektverfolgung und der Anzahl der unabhängigen Formparameter. In der äußeren Form gibt es hier keinen Unterschied. Im ersten Fall werden Objekthypothesen im nächsten Bild wiederholt generiert und es findet ein 3-D punktebasierter Abgleich mit den vorherigen Objekthypothesen statt. Die zwei Formparameter *Hohe_2* und *Länge_2* sind durch konstante Faktoren an die unabhängigen Parameter *Hohe* und *Länge* gebunden. Im zweiten Fall wird eine gezielte Messung im 2-D Bild für jede verfolgte Objekthypothese durchgeführt, so daß der Abgleich zwischen modellbasierter Vorhersage und Messung auf der Ebene der 2-D Kanten stattfindet. Der Schritt zur Objektverfolgung wird vor der Konturdetektion durchgeführt. Die neuen Kanten, die eine weiterverfolgte Objekthypothese unterstützen, gehören zu der Menge der *Verifizierten Kanten* und werden aus der weiteren Verarbeitung der aktuellen Bilddaten hinausgenommen.

Die Modelle der Klasse *Verfolgung_Fein* arbeiten ausschließlich mit dem kantenbasierten Abgleich zusammen. Sie dienen einer detaillierten Fahrzeugerkennung, falls die Bildauflösung dazu ausreichend fein ist. Dabei wird an solche Formen der *Fahrzeug_Fein*-Klasse gedacht, wie *Pkw* und *Lkw*. Die Objektform ist hier durch ein erweitertes parametrisches Drahtmodell dargestellt, das noch mehr Kanten als bisher enthält. Diese spezialisierten Formen unterscheiden sich von der entsprechenden Form der Klasse *Fahrzeug_Form* nur durch ihren größeren Detailgrad und den damit verbundenen erhöhten Rechenaufwand für die Objektverfolgung. Diese Art der Modellierung ist nur für einzelne Objekte gedacht, die entweder mit großer Auflösung ins Bild projiziert werden, oder zu deren Projektion eine zusätzliche Kamera mit Teleobjektiv verwendet wurde.

Andere stationäre oder bewegte Objekte, die nicht zur Fahrzeugklasse gehören, werden bisher als *Hindernis* eingestuft. Vom Prinzip her können aber alle realisierten Elemente der

adaptiven Schätzung auch zur Verfolgung von solchen bewegten Objekten, wie z.B. *Radfahrer* eingesetzt werden. Dies bedarf aber vorher einer Erweiterung der Wissensbasis um das Modell des Radfahrers.

5.1.4 Aktivierung des Modells

Der Aktivierung von allen drei Modellklassen liegt der Ansatz zur adaptiven Erkennung zugrunde, wie er in Abschnitten 2.2 und 2.3.5 eingeführt wurde. Die Klassen unterscheiden sich nur durch die Komplexität der Zustandsvektoren und Übergangs- und Projektionsfunktionen sowie der benötigten Messungen. Auf diese Verfahren wird im einzelnen in den folgenden Abschnitten 5.2–5.6 detailliert eingegangen.

5.2 Generierung von Objekthypothesen

Die Hypothesengenerierung bedeutet eine modellbasierte Gruppierung von Segmenten und die Initialisierung eines Objektzustandes für jede entsprechende Gruppe.

5.2.1 Gruppierung von Segmenten

Auf der Basis der klassifizierten Konturen und mit Kenntnis der Straßenlage werden Konturgruppen generiert. Die *Gruppen*-Klasse umfaßt drei Typen von Gruppen: *Streifengruppe*, *Hindernisgruppe* und *Fahrzeuggruppe*.

Die Konturen werden am Anfang eingeteilt in drei Mengen: überwiegend horizontal, vertikal und andere. Die eigentliche Gruppierung beginnt mit einer am untersten im Bild gelegenen horizontalen Kontur. Als erstes werden vertikale Konturen dieser Kontur zugeordnet, die über ihr liegen. Anschließend werden erneut horizontale Konturen, für eine Begrenzung der Gruppe von oben, gesucht. Zum Schluß werden restliche Konturen, die innerhalb der Aussenkontur liegen, der Gruppe zugewiesen.

Während der Gruppierung wird das Wissen über die Straßentransformation und die Einschränkungen der 3-D Objektgröße intensiv genutzt. Indem mit dem Segment begonnen wird, das auf der Straßenebene liegen soll, kann gleich die Anfangstiefe geschätzt werden. Mögliche Hindernisse sollten in etwa die Breite eines Fahrbahnstreifen nicht überschreiten. Das Breite-Höhe-Verhältnis und Breite-Länge-Verhältnis sollten im vorgegebenen Rahmen liegen. Diese Modellzwänge werden auf die Bildebene zurückgerechnet, und somit wird die Gruppierung zu relevanten Bereichen eingeschränkt. Auf der Basis einer im Bild gefundenen Konturgruppe kann ein neues Objekt hypothetisiert oder ein früheres verifiziert werden.

5.2.2 Initialisierung des Objektzustandes

In der Initialisierungsphase wird der Anfangszustand eines Objektes aufgrund des Eigenzustandes und der Attribute der zugehörigen Konturgruppe initialisiert. Es gibt kleine Unterschiede in der Initialisierung von Fahrbahnstreifen-, Hindernis- und Fahrzeug-Hypothesen.

Initialisierung der Fahrbahnstreifen

Je nach aktueller Fahrbahnhypothese werden bis zu 6 Streifenobjekte initialisiert: linker und rechter Randstreifen sowie bis zu zwei Streifen pro Fahrspurbegrenzung.

Für Streifen besteht die Möglichkeit sowohl einer FP-basierten Tiefenschätzung, wie auch einer bewegungsbasierten Tiefenschätzung. Die Richtung wird aus der Rückprojektion der Konturpunkte in die Straßenkoordinaten gewonnen. Eine Menge von aufeinanderfolgenden Mittelstreifen wird quasi als ein Objekt abstrahiert, so daß die Werte des Vorgängerstreifen bei der Initialisierung der nächsten Streifenhypothese übernommen werden.

Bewegte Objekte

Bewegte Objekte müssen auch auf der Fahrbahnebene stehen. Hier aber ist die Bewegung nicht bekannt, so daß sie weniger zuverlässig ist als für Straßenstreifen. Jedes Hindernis bekommt einen Tiefenbereich zugeordnet. Dieser Bereich wird ermittelt, indem die Aufstandsfläche des Objektes in die Straßenebene projiziert wird. Die Richtungshypothese wird aus der die Konturgruppe umschließenden Konturform gewonnen.

Hindernisobjekte

Die Hindernisobjekte werden nur aufgrund der Eigenbewegung und der Verschiebungsvektoren initialisiert. Sie werden wiederum als vertikal zur Straßenebene angenommen, sowie parallel zur Y^w - oder Z^w -Achse des Straßenkoordinatensystems (im allgemeinen zur Krümmungstangente der Fahrbahn).

Ablauf

Die Objektinitialisierung läuft nach folgendem Schema ab:

1. Eine FP-basierte Tiefenschätzung von $p_Z^w(0)$ unter Annahme $p_Y^w = 0$.
2. Eine Korrektur der Höhenschätzung wenn $p_X^w(0)$, weit außerhalb der Straße liegen sollte.
3. Der Betrag der translatorischen Geschwindigkeit $V(0)$ wird entweder aus der Tiefe und der relativen Konturveränderung v_z , oder aus den unterschiedlichen Verschiebungen v_x, v_y von Konturpunkten gewonnen.
4. Der Richtungswinkel $\Theta(0)$ ergibt sich aus der Rückprojektion der Konturgruppe in die Straßenkoordinaten.
5. Die Umgebungsobjekte werden als vertikal zur Straßenebene angenommen.
6. Die Winkelgeschwindigkeit $\omega(0)$ entspricht der aktuellen Fahrbahnkrümmung.

5.3 Dynamische Übergangsfunktionen

In diesem Abschnitt werden die Übergangsfunktionen $f_S(s_S)$, $f_B(s_B)$ und $f_U(s_U)$ für die drei verschiedenen Objektklassen definiert – unbewegte Fahrbahnstreifen, bewegte Objekte und unbewegte Hindernisse/Umgebungsobjekte. Die funktion $f_U(s_U)$ ist eine Kombination von Elementen aus den beiden anderen Funktionen. Für detaillierte Angaben zur Herleitung der Übergangsfunktionen und die Jacobimatrizen dieser Funktionen wird auf den Anhang (Abschnitte 9.2.2, 9.2.3) verwiesen.

5.3.1 Für unbewegte Fahrbahnstreifen

Mit $\mathbf{s}_S(k) = [p_{SX}^c(k), p_{SZ}^c(k), \Theta_S^c(k), V_S(k), \omega_S(k), \xi_S]^T$ bezeichnen wir der Zustandsvektor eines unbewegten Fahrbahnstreifen-Objektes.

Die Parameter Θ_S , V_S und ω_S sind direkt abhängig von den Parametern Θ_c und ω_c der Eigenbewegung:

$$\begin{aligned}\Theta_S^c(k) &= \Theta_{S0} - \Theta_c(k) \\ V_S(k) &= V_c(k) \\ \omega_S(k) &= -\omega_c(k)\end{aligned}$$

wobei Θ_{S0} der Richtungswinkel der Objekthypothese in Weltkoordinaten ist, der bei der Objektinitialisierung bestimmt wird. Die Übergangsfunktion der Lageparameter ist durch die dynamische Eigentransformation (vgl. Abschnitt 4.5) DET gegeben, in der die Parameter der Eigenbewegung durch die entsprechenden Parameter des Zustandsvektors \mathbf{s}_S ersetzt wurden. Die diskreten *Übergangsgleichungen* sind wie folgt:

$$\begin{aligned}p_S^c(k+1) &= [DET(-l_X^c(k), -l_Z^c(k), -\omega_c(k)\tau)]p_S^c(k) \\ V_S(k+1) &= V_S(k) = V_c(k) \\ \Theta_S(k+1) &= \Theta_S(k) + \omega_S(k)\tau \\ \omega_S(k+1) &= \omega_S(k) \\ \xi_S(k+1) &= \xi_S(k)\end{aligned}$$

$$\mathbf{s}_S(k+1) = \begin{pmatrix} p_{SX}^c(k+1) \\ p_{SZ}^c(k+1) \\ \Theta_S^c(k+1) \\ V_S(k+1) \\ \omega_S(k+1) \\ \xi_S(k+1) \end{pmatrix} = \begin{pmatrix} p_{SX}^c \cos(\omega_S \tau) + p_{SZ}^c \sin(\omega_S \tau) - \frac{V_S}{\omega_S} (\cos(\omega_S \tau) - 1) \\ -p_{SX}^c \sin(\omega_S \tau) + p_{SZ}^c \cos(\omega_S \tau) + \frac{V_S}{\omega_S} (\sin(\omega_S \tau)) \\ \Theta_S^c + \omega_S \tau \\ V_S \\ \omega_S \\ \xi_S \end{pmatrix} \quad (5.3.1)$$

Bei kleinen Winkelgeschwindigkeiten $\omega_S \rightarrow 0$ reduziert sich das Bewegungsmodell zu einer geradlinigen Bewegung und somit geht die Übergangsfunktion in folgende Funktion $f_S(\text{lin})$ über:

$$\mathbf{s}_S(k+1) = \begin{pmatrix} p_{SX}^c(k) \\ p_{SZ}^c(k) \\ \Theta_S^c(k) \\ V_S(k) \\ \omega_S(k) \\ \xi_S(k) \end{pmatrix} + \begin{pmatrix} 0 \\ V_S \tau \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (5.3.2)$$

Die Jacobimatrix für unbewegte Objekte $\mathbf{F}_S(k)$ wird im Anhang (Abschnitt 9.2) spezifiziert.

5.3.2 Für bewegte Fahrzeuge

Sei mit $\mathbf{s}_B(k) = [p_{BX}^c(k), P_{BZ}^c(k), \Theta_B(k), V_B(k), \omega_B(k), \xi_B]^T$ der Zustandsvektor eines bewegten Fahrzeuges bezeichnet.

Die Bewegung des Eigenfahrzeuges summiert sich hier mit der Bewegung des Objektes in der Straßenebene (vgl. Bild 5.3.1). Nur die summarische Bewegung unterliegt unserer Beobachtung. Nehmen wir an, das die unbekannte Eigenbewegung des Objektes im Raum V_o und

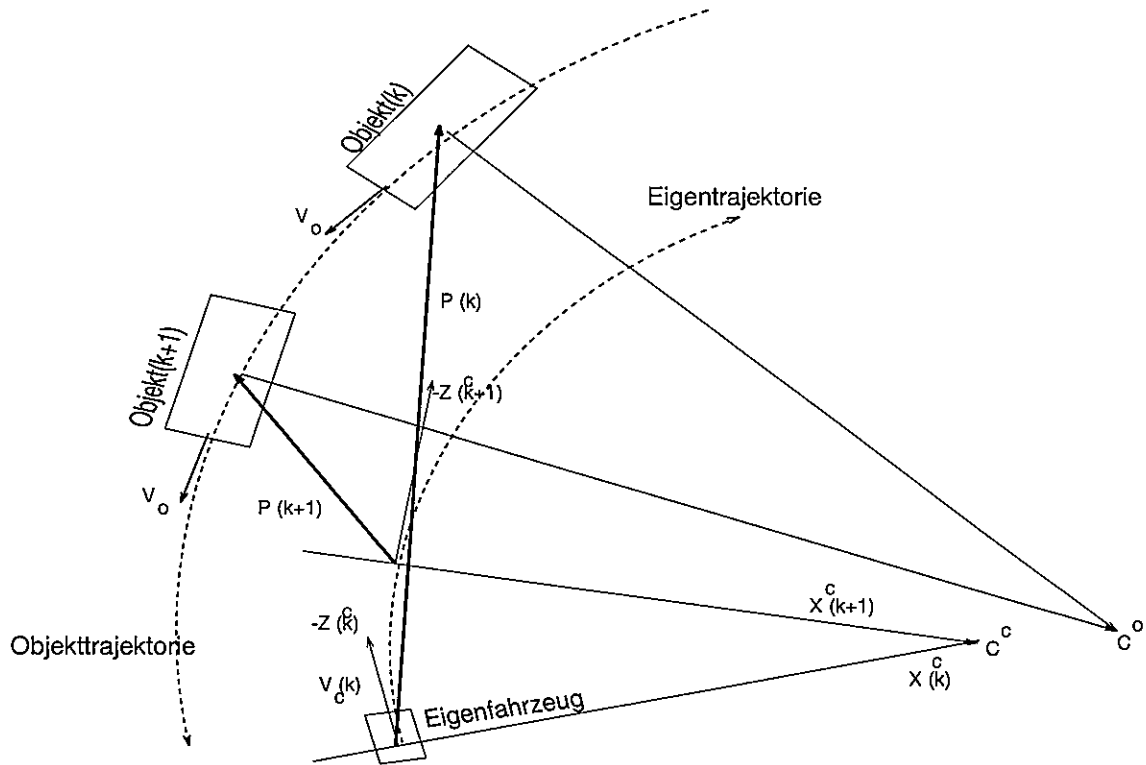


Bild 5.3.1: Die Objekttrajektorie auf Fahrbahnebene bei Eigenbewegung.

$\dot{\theta}_o = \omega_o$ beträgt. Für die Eigenbewegung des Objektes bei unbewegter Kamera gelten ähnliche Bewegungsgleichungen, wie für die Eigenbewegung des Kamerafahrzeuges:

$$\begin{aligned}\dot{p}_X^c(k) &= V_B \cos \dot{\theta}_o \\ \dot{p}_Z^c(k) &= V_B \sin \dot{\theta}_o \\ \dot{\Theta}_B(k) &= \omega_B \\ \dot{V}_B(k) &= 0; \\ \dot{\omega}_B(k) &= 0; \\ \dot{\xi}_B(k) &= 0;\end{aligned}$$

Die Lösung der Differentialrechnung für diese Bewegungsgleichungen im Zeitraum $\langle t_k, t_{k+1} \rangle$ und die Berücksichtigung der Eigenbewegung des Kamerafahrzeuges (in Form von Transformation $DET(k)$) führen zur folgenden (summarischen) diskreten Zustandsübergangsfunktion $f_B(s_B)$:

$$\begin{aligned}\mathbf{p}_B^c(k+1) &= [DET(k)] \mathbf{p}_o^c(k+1) = [DET(k)] [OBT(k)] \mathbf{p}_B^c(k) \\ V_B(k+1) &= V_B(k) \\ \Theta_B(k+1) &= \Theta_B(k) + \omega_B T \\ \omega_B(k+1) &= \omega_B(k) \\ \xi_B(k+1) &= \xi_B(k)\end{aligned}$$

Nach Ausmultiplizierung der Koordinatentransformationen ergibt sich folgende Zustandsübergangsfunktion $f_B(s_B)$ für ein bewegtes Objekt auf Fahrbahnebene:

$$\mathbf{s}_B(k+1) = \begin{pmatrix} p_{BX}^c(k+1) \\ p_{BZ}^c(k+1) \\ \Theta_B(k+1) \\ V_B(k+1) \\ \omega_B(k+1) \\ \xi_B(k+1) \end{pmatrix} \quad (5.3.3)$$

$$= \begin{pmatrix} p_{BX}^c \cos(\omega_c \tau) - p_{BZ}^c \sin(\omega_c \tau) + \frac{V_c}{\omega_c} (C-1) + \frac{V_B}{\omega_B} (C2-C1) \\ p_{BX}^c \sin(\omega_c \tau) + p_{BZ}^c \cos(\omega_c \tau) - \frac{V_c}{\omega_c} S + \frac{V_B}{\omega_B} (S1-S2) \\ \Theta_B(k) + \omega_B(k) \tau \\ V_B(k) \\ \omega_B(k) \\ \xi_B(k) \end{pmatrix} \quad (5.3.4)$$

mit

$$\begin{aligned} S1 &= \sin(\Theta_B^c - \omega_c \tau) \\ C1 &= \cos(\Theta_B^c - \omega_c \tau) \\ S2 &= \sin(\Theta_B^c - \omega_c \tau + \omega_B \tau) \\ C2 &= \cos(\Theta_B^c - \omega_c \tau + \omega_B \tau) \end{aligned}$$

Bei kleinen Winkelgeschwindigkeiten ω_c und ω_B reduziert sich das Bewegungsmodell zu einer geradlinigen Bewegung:

$$\mathbf{s}_B(k+1) = \begin{pmatrix} p_{BX}^c(k) \\ p_{BZ}^c(k) \\ \Theta_B(k) \\ V_B(k) \\ \omega_B(k) \\ \mathbf{F}_B(k) \end{pmatrix} + \begin{pmatrix} -V_B(k) \tau \sin(\Theta_B^c(k)) \\ V_c(k) \tau - V_B(k) \tau \cos(\Theta_B^c(k)) \\ 0 \\ 0 \\ 0 \\ \mathbf{0} \end{pmatrix} \quad (5.3.5)$$

Die Herleitung der Transformation $OBT(k)$, der Zustandsübergangsfunktion $f_B(\mathbf{s}_B)$ und die Jacobimatrix für bewegte Objekte $\mathbf{F}_B(k)$ erfolgt im Anhang (Abschnitt 9.3).

5.3.3 Für Hindernisse/Umgebungsobjekte

Im Prinzip können die Übergangsfunktionen für Hindernis- oder Umgebungsobjekte direkt aus den Funktionen für unbewegte Streifen oder bewegte Fahrzeuge abgeleitet werden, je nachdem ob sie stationär sind oder sich bewegen. Für Hindernisse gilt der dynamische Vektor des Fahrzeugmodells und für Umgebungsobjekte der dynamische Vektor des Streifenmodells. Der Formvektor bildet eine Zwischenstufe der beiden anderen Formvektoren.

5.4 Modifikation der Objekthypothese

Die Zustandsprojektion ist nicht direkt abhängig von den Bewegungsparametern V und ω des Objektzustandes. Andererseits gibt es eine gegenseitige Abhängigkeit in den Meßwerten der Parameter Richtung Θ und der Längen l_1 und l_2 voneinander. Diese Problem tritt sowohl bei der 2-D Messung, wie auch bei der 3-D Messung auf.

Deshalb wurde eine zweistufige Modifikation realisiert. Als erstes erfolgt die neue Schätzung des reduzierten Zustandes

$$\mathbf{s}_R^+(k) = \mathbf{s}^+(k) - \{V, \omega\} \quad (5.4.1)$$

Dabei werden auch die gegenseitig abhängigen Richtungs- und Längenparameter Θ, l_1, l_2 neu geschätzt.

Aus dem Vektor $\delta \mathbf{s}_R^+(k) = [\mathbf{s}_R^+(k) - \mathbf{s}_R^+(k-1)]$ werden neue Werte von $V(k), \omega(k)$ gemessen. Aus diesen Messungen ergeben sich Grenzwerte der gegenseitig abhängigen Parameter, die als eine zweite synthetische Messung aufgefaßt werden

$$\mathbf{s}_m(k) = [\Theta_m(k), l_{1m}(k), l_{2m}(k)]^T \quad (5.4.2)$$

Diese synthetischen Messungen führen zusammen zur zweiten Modifikation des Zustandes

$$\mathbf{s}_E(k) = [V(k), \Theta(k), \omega(k), l_1(k), l_2(k)]^T \quad (5.4.3)$$

Es wird also ein *Erweitertes Kalman Filter* (gemäß Abschnitt 2.2.3) mit einer zweistufigen Modifikation und zusätzlicher synthetischer Zwischenmessung eingesetzt. Die Schritte 2,3,4 und 5 des Kalman Filters in Bild 2.2.3 zerfallen in je zwei Teilschritte. Die Reihenfolge der Verarbeitung dieser 8 Teilschritte für eine Objekthypothese zum Zeitpunkt k ist damit wie folgt:

2.a Messung im Bild der Lageparameter

3.a Berechnung der reduzierten Kalmanverstärkung

$$\mathbf{K}(k) = \mathbf{P}_R^*(k) \mathbf{H}^T(k) \{ \mathbf{H}(k) \mathbf{P}_R^*(k) \mathbf{H}^T(k) + \mathbf{R}(k) \}^{-1} \quad \text{mit } \mathbf{H}(k) = \left. \frac{\delta \mathbf{h}}{\delta \mathbf{s}_R} \right|_{\mathbf{s}_R^+(k)} \text{ der Jacobi-Matrix der Funktion } \mathbf{h}(\cdot).$$

4.a Modifikation des reduzierten Zustandes

$$\mathbf{s}_R^+(k) = \mathbf{s}_R^*(k) + \mathbf{K}(k) \{ \mathbf{m}(k) - \mathbf{h}(\mathbf{s}_R^*(k)) \};$$

5.a Modifikation der reduzierten Kovarianzmatrix

$$\mathbf{P}_R^{\dagger}(k) = \mathbf{P}_R^*(k) - \mathbf{K}(k) \mathbf{H}(k) \mathbf{P}_R^*(k)$$

2.b Synthetische Messung der Bewegung und die bewegungsabhängige Messung der abhängigen Parameter auf der Basis der Differenz $\mathbf{s}_R^+(k) - \mathbf{s}_R^+(k-1)$

3.b Berechnung der restlichen Kalmanverstärkung

4.b Modifikation des restlichen Zustandes Modifikation des restlichen Zustandes $\mathbf{s}_E^{\dagger}(k)$ auf der Basis der Vorhersage $\mathbf{s}_E^*(k)$, der synthetischen Messung von V, ω und der abhängigen Messung $\mathbf{s}_m(k)$.

5.b Modifikation der restlichen Kovarianzmatrix

Die sequentielle Innovation

Um den mühsamen Weg einer Matrixinvertierung bei die Berechnung der Kalmanverstärkung zu vermeiden, wurde ein *EKF mit sequentieller Innovation* implementiert [Wue91]. Durch eine getrennte Betrachtung der unkorrelierten Zeilen wird die Matrixumkehrung durch eine sequentielle Division (bei zeilenweisem Vorgehen) ersetzt.

1. Die Projektion der Zustandsvorhersage \mathbf{s}^* :

$$\text{FOR } i = 1, \dots, |\mathbf{m}| \text{ DO } \mathbf{mproj}[i] = h_i(\mathbf{s}^*);$$

2. Meßwerterfassung: bestimme den aktuellen Meßvektor \mathbf{m} und Kovarianzmatrix \mathbf{R} .

3. Die Modifikationsschleife (Innovationsschleife):

FOR $j = 1, \dots, |\mathbf{s}|$ DO $\delta \mathbf{s}[j] = 0$;

FOR $i = 1, \dots, |\mathbf{m}|$ DO

(a) $\mathbf{m}_{proj}[i] = \mathbf{m}_{proj}[i] + \mathbf{H}[i] * \delta \mathbf{s}$;

(b) $nenner = \mathbf{H}[i] * \mathbf{P} * \mathbf{H}[i]^T + \mathbf{R}[i][i]$;

(c) Die Berechnung der aktuellen Kalman-Verstärkungsmatrix \mathbf{K} :

FOR $j = 1, \dots, |\mathbf{s}|$ DO $\mathbf{K}[j][i] = \mathbf{P}[j] * \mathbf{H}^T[i] / nenner$;

(d) $\delta \mathbf{m} = \mathbf{m}[i] - \mathbf{m}_{proj}[i]$;

(e) Die Teilmodifikation des neuen Zustandes:

FOR $j = 1, \dots, |\mathbf{s}|$ DO : $\delta \mathbf{s}[j] = \delta \mathbf{s}[j] + \mathbf{K}[j][i] * \delta \mathbf{m}$;

(f) Die Modifikation (Innovation) der Schätzfehler-Kovarianzmatrix $\tilde{\mathbf{P}}$:

$\mathbf{P}^+ = \mathbf{P}^* - \mathbf{K}[[i] * \mathbf{H}[i] * \mathbf{P}^*$

4. Die entgültige Modifikation des Zustandes:

FOR $j = 1, \dots, |\mathbf{s}|$ DO $\mathbf{s}^+[j] = \mathbf{s}^*[j] + \delta \mathbf{s}[j]$

5.5 Abbildungsfunktionen und Messungen

Die Abbildungsfunktion $h(\mathbf{s})$ hängt direkt mit der Komplexität des Objektmodells zusammen, d.h. mit der Art der Zuordnung der Modellelemente zu Bildelementen. Wir unterscheiden hier zwischen zwei Meßarten:

1. die 3-D Messung bedeutet, daß eine wiederholte Objektinitialisierung für jedes komplette Bild vorgenommen wird; die Modifikation der Objekthypothese bedeutet dann eine Objekt-zu-Objekt-Abgleich.
2. die 2-D Messung bedeutet eine gezielte Segmentdetektion für jedes bereits existierende Objekthypothese und eine Modifikation der Objekthypothese aufgrund der Unterschiede zwischen Modellprojektion und Messung.

Die 3-D Messung wird für die Verfolgung von Objekthypothesen der generellen Modellklasse *Verfolgung* eingesetzt. Die 2-D Messungen werden für spezialisierte Modellklassen *Verfolgung_Form* und *Verfolgung_Fein* eingesetzt.

5.5.1 Die 3-D Messung

Die 3-D Messung bedeutet eine wiederholte Objekthypothesengenerierung, die unabhängig von Ergebnissen für das vorherige Bild vorgenommen wird. Die Modifizierung einer Objekthypothese erfolgt durch einen Objekt-zu-Objekt-Abgleich (Bild 5.5.1).

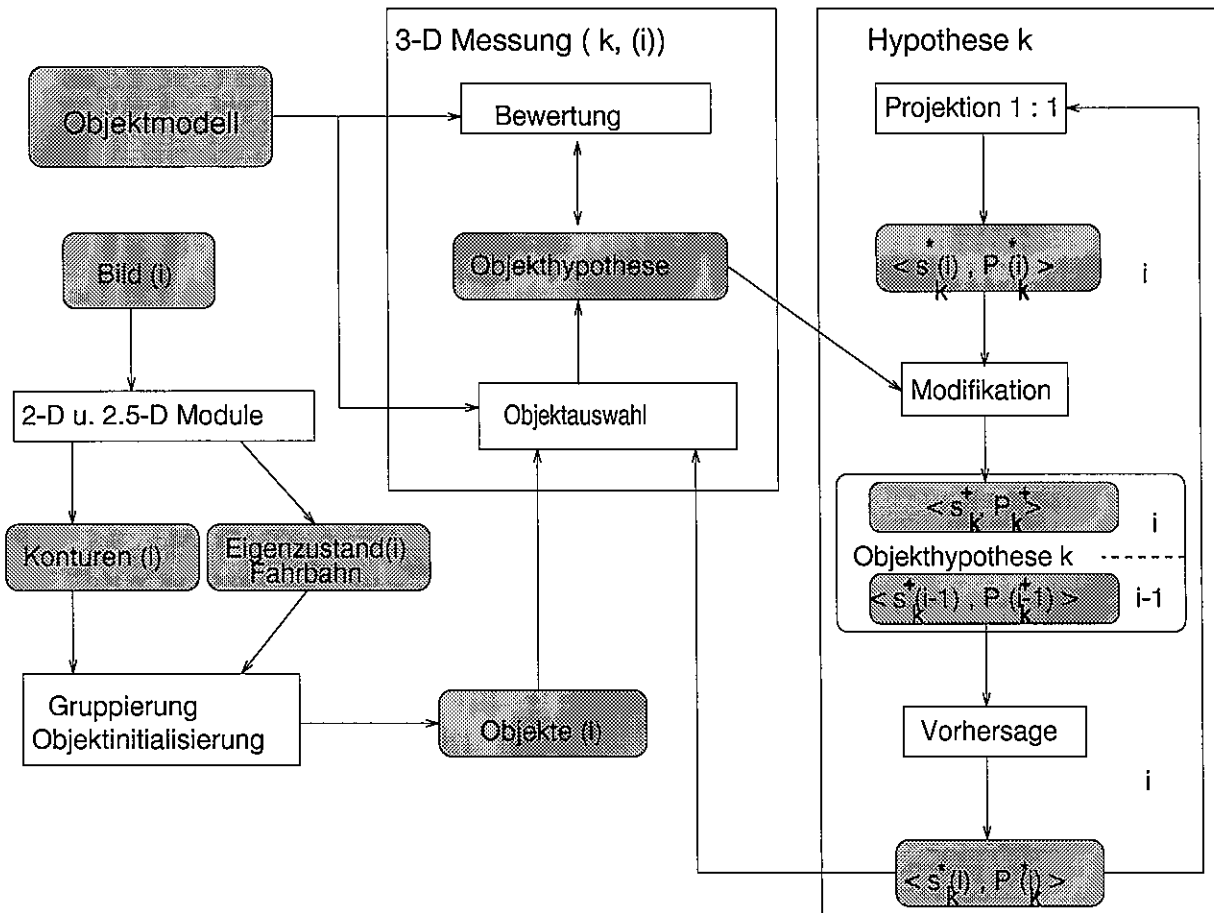


Bild 5.5.1: Die Objektverfolgung basierend auf einem Objekt-zu-Objekt-Abgleich.

Der Meßvektor

Für die zweistufige Modifikation unterscheiden wir zwischen zwei Meßvektoren, dem direkten $m_R(k)$ und dem synthetischen Vektor $m_E(k)$. Der direkte Meßvektor $m_R(k)$ ist äquivalent zu einem reduzierten Zustandsvektor (die Geschwindigkeiten werden nicht direkt gemessen):

$$m_R(k) = [P_X(k), P_Z(k), Br(k), h_1(k), h_2(k), l_1(k), l_2(k))]^T \tag{5.5.1}$$

Im Vergleich zu einem vollen Zustandsvektor sind keine Geschwindigkeiten gegeben. Diese werden indirekt gemessen werden, durch die Verfolgung der Lage- und Richtungs-differenzen der Objekthypothese in bis zu 5 Bildern. Es werden auch bewegungsabhängige Messungen der Länge und Richtung vorgenommen, die als zusätzlich zu den geometriebasierten Messungen dieser Parameter gelten. Damit ist der indirekte Meßvektor $m_E(k)$ wie folgt:

$$m_E(k) = [V(k), \omega(k), \Theta_m(k), l_{1m}(k), l_{2m}(k)]^T \tag{5.5.2}$$

Die Projektionsabbildung

Die Projektionsabbildung $h(t)$ reduziert sich in diesem Fall zu einer Äquivalenzfunktion, da sowohl Zustände wie Messungen Variablen derselben Klassen im 3-D Raum sind.

Der Ablauf der 3-D Messung

Die Verarbeitung der 3-D Messung für eine vorgegebene Objekthypothese reduziert sich in diesem Fall zu einem Objekt-zu-Objekt-Abgleich zwischen der aktuellen Objektvorhersage

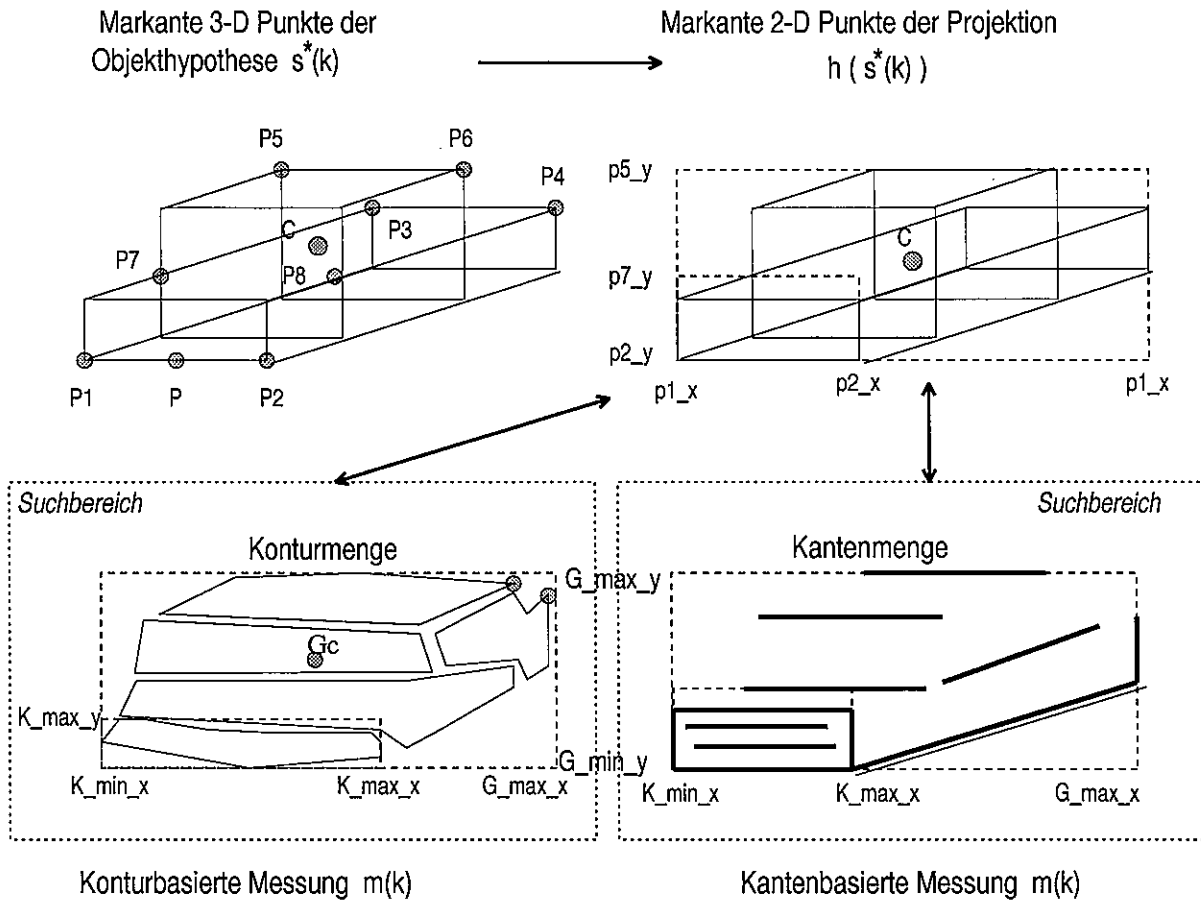


Bild 5.5.2: Die Bedeutung und Gewinnung der 2-D Messung für die Objektverfolgung: (a) konturbasierte Messung, (b) kantenbasierte Messung

und den neu initialisierten Objekten im aktuellen Bild.

Nachdem alle vorherigen Objekthypothesen modifiziert wurden (oder auch nicht), werden ihre gefundenen Partnerobjekte aus der Menge der neu generierten Hypothesen eliminiert. Die übriggebliebenen neuen Hypothesen werden in die Menge der verfolgten Hypothesen übernommen.

5.5.2 Die 2-D Messung

In diesem und im nächsten Abschnitt beschreiben wir die 2-D Messungen für die Objektverfolgung der spezialisierten Modellklasse *Verfolgung_Form*.

Der Abgleich mit den Bildmerkmalen kann prinzipiell auf zwei unterschiedlichen Segmentebenen verlaufen – entweder in Form des Abgleiches der Modellpunkte mit konturbasierten Messwerten oder mit liniensegmentbasierten Messwerten (Bild 5.5.2). In beiden Fällen werden Meßvektoren generiert, die die Lage von markanten Punkten im Bild der ausgewählten Segmentgruppen darstellen.

Der Meßvektor

Die projizierten Modellpunkte werden abgeglichen mit neuen 2-D Meßpunkten aus folgendem Meßvektor:

$$m(k)^T = [K_{minx}, K_{maxx}, G_{Cx}, G_{Cy}, G_{minx}, G_{maxx}, G_{miny}, G_{maxy}, K_x, K_y]^T \quad (5.5.3)$$

Der 2-D Meßvektor wird durch die x -Komponenten des umschreibenden Rechtecks der Startkontur (K_{minx}, K_{maxx}), sowie durch den Schwerpunkt $G = (G_x, G_y)$ und das umschreibende Rechteck ($(G_{minx}, G_{maxx}, G_{miny}, G_{maxy})$) der Konturgruppe im Bild gebildet (Bild 5.5.2):

$$\mathbf{m}(k) = [K_{minx}, K_{maxx}, G_x, G_y, G_{minx}, G_{maxx}, G_{miny}, G_{maxy}]^T \quad (5.5.4)$$

Es ist somit notwendig auch in der Verfolgungsphase die Konturmenge im Bild zu gruppieren, so wie es schon für die Hypothesengenerierung der Fall war. Nicht alle Elemente des Meßvektors sind relevant für bewegte und unbewegte Objekte. Der allgemeine Meßvektor reduziert sich zu folgenden zwei Untervektoren:

1. für bewegte Hindernisse:

$$\mathbf{m}_B(k) = [K_{minx}, K_{maxx}, G_{minx}, G_{maxx}, G_{miny}, G_{maxy}]^T \quad (5.5.5)$$

2. für unbewegte Fahrbahnstreifen:

$$\mathbf{m}_S(k) = [G_x, G_y, G_{minx}, G_{maxx}, G_{miny}, G_{maxy}]^T \quad (5.5.6)$$

Im einfachsten Fall sind die Objekte durch fünf markante Modellkanten ihrer umgebenden 3-D Hülle spezifiziert. Diese Kanten werden auf die Bildebene projiziert und mit den Bildkanten im nächsten Bild verglichen. Damit können nach erfolgreicher Verifikation einer Hypothese, die korrespondierenden Kanten aus der Konturdetektion ausgeschlossen werden. Die Bild- und Modellkanten werden gewöhnlich durch die Lage ihres Mittelpunktes $C_i (i = 1, \dots, 5)$, ihre Neigung gegenüber der x -Achse ϕ_i und ihre Länge $l_i (i = 1, \dots, 5)$ beschrieben. Der Messvektor besteht somit aus fünf Kantenbeschreibungen der Form:

$$\mathbf{m}(k) = [C_{ix}(k), C_{iy}(k), \phi_i(k), l_i(k)] (i = 1, \dots, 5) \quad (5.5.7)$$

Für die detaillierten Objektmodelle von Fahrzeugen werden längere Formvektoren als die bisherigen 5-stelligen Formvektoren der bewegten Objekte gebraucht. Vom Prinzip her ist aber die Objektverfolgung identisch mit der Verfolgung, die durch die Konzepte der Familie Verfolgung-Form modelliert wird. Der Strukturvergleich der projizierten Modellkanten mit Bildkanten erfordert die Existenz von Bildstrukturen mit relativ hoher Auflösung. In der Praxis, für das vorhandene Bildmaterial, wird dieses Modell im Durchschnitt für höchstens je ein Fahrzeug im Bild verwendet werden. Von größerer Bedeutung ist diese feine Modellierungsform falls eine Telekamera eingesetzt wird. Nach einer Fixierung solcher Kamera auf ein Objekt, können die im Bild detektierten Segmente mit Hilfe des detaillierten Modells ausgewertet werden.

Die Abbildungsfunktion

Die Abbildungsfunktion $h(\mathbf{s})$ wird in drei Schritten bestimmt:

1. *Auswahl der Modellpunkte*

Abhängig von der Richtung der Längsachse des Objektes relativ zur Z^c Achse der projizierten Kamerakoordinaten $\Theta_B^c(k) = \Theta_B(k) - \Theta_c(k)$ (für bewegte Objekte) bzw. $\Theta_S^c(k) = \Theta_{S0}(k) - \Theta_c(k)$ und der Art des Objektes werden entsprechende Modellpunkte aus der Menge $\{P1, P2, P3, P4, P5, P6, P7, P8G\}$ ausgewählt.

Sei mit Θ weiterhin entweder $\Theta_B^c(k)$, $\Theta_S^c(k)$ oder $\Theta_U^c(k)$ bezeichnet, je nach der Art des Objektes. Wenn das Objekt nach links vom Beobachter gedreht ist, dann werden die Punkte $P1, P2, P3, P4, P5, P7$ und G zur Abbildung in die Bildebene ausgewählt (Ansicht LINKS oben). Wenn das Objekt nach rechts gedreht ist, dann werden die Punkte $P1, P2, P3, P4, P6, P8$ und G ausgewählt (Ansicht RECHTS oben).

2. *Bestimmung der aktuellen Lage der Modellpunkte in Eigenkoordinaten*

Es wird die aktuelle Lage der im Schritt 1 ausgewählten Modellpunkte in Eigenkoordinaten bestimmt, als Funktion des aktuellen Zustandes $\mathbf{s}(k)$:

$$P1_X^c = p_X^c - \frac{Br}{2} \cos(\Theta) ; P1_Z^c = p_Z^c + \frac{Br}{2} \sin(\Theta) ; P1_Y^c = 0 ; \quad (5.5.8)$$

$$P2_X^c = p_X^c + \frac{Br}{2} \cos(\Theta) ; P2_Z^c = p_Z^c - \frac{Br}{2} \sin(\Theta) ; P2_Y^c = 0 ; \quad (5.5.9)$$

$$P3_X^c = P1_X^c - \kappa_1 Br \sin(\Theta) ; P3_Z^c = P1_Z^c - \kappa_1 Br \cos(\Theta) ; P3_Y^c = \kappa_2 Br ; \quad (5.5.10)$$

$$P4_X^c = P2_X^c - \kappa_1 Br \sin(\Theta) ; P4_Z^c = P2_Z^c - \kappa_1 Br \cos(\Theta) ; P4_Y^c = \kappa_2 Br ; \quad (5.5.11)$$

$$P5_X^c = P1_X^c - \kappa_3 Br \sin(\Theta) ; P5_Z^c = P1_Z^c - \kappa_3 Br \cos(\Theta) ; P5_Y^c = \kappa_4 Br ; \quad (5.5.12)$$

$$P6_X^c = P2_X^c - \kappa_3 Br \sin(\Theta) ; P6_Z^c = P2_Z^c - \kappa_3 Br \cos(\Theta) ; P6_Y^c = \kappa_4 Br ; \quad (5.5.13)$$

$$P7_X^c = P1_X^c ; P7_Z^c = P1_Z^c ; P7_Y^c = \kappa_2 Br ; \quad (5.5.14)$$

$$P8_X^c = P2_X^c ; P8_Z^c = P2_Z^c ; P8_Y^c = \kappa_2 Br ; \quad (5.5.15)$$

Der Punkt G wird nur für unbewegte Fahrbahnstreifen bestimmt:

$$G_{SX}^c = p_X^c - \frac{\kappa_1 Br}{2} \sin(\Theta) ; G_{SZ}^c = p_Z^c - \frac{\kappa_1 Br}{2} \cos(\Theta) ; G_{SY}^c = 0 ; \quad (5.5.16)$$

3. *Die Transformation der Eigenkoordinaten in die Bildebene*

Die im Schritt 2 ermittelten Koordinaten werden in Kamerakoordinaten umgewandelt

$$\begin{aligned} X^k &= X^c \\ Y^k &= \cos(\alpha)Y^c + \sin(\alpha)Z^c - H \cos(\alpha) \\ Z^k &= -\sin(\alpha)Y^c + \cos(\alpha)Z^c + H \sin(\alpha) \end{aligned} \quad (5.5.17)$$

und in die Bildebene projiziert

$$x = -F_x \frac{X^k}{Z^k} ; y = -F_y \frac{Y^k}{Z^k} \quad (5.5.18)$$

Die komplette Transformation zwischen Eigenkoordinaten (P_X^c, P_Y^c, P_Z^c) eines Punktes P und den Bildkoordinaten (p_x, p_y) seiner Projektion ist somit wie folgt:

$$p_x = -F_x \frac{P_X^c}{-\sin(\alpha)P_Y^c + \cos(\alpha)P_Z^c + H \sin(\alpha)} \quad (5.5.19)$$

$$p_y = -F_y \frac{\cos(\alpha)P_Y^c + \sin(\alpha)P_Z^c - H \cos(\alpha)}{-\sin(\alpha)P_Y^c + \cos(\alpha)P_Z^c + H \sin(\alpha)} \quad (5.5.20)$$

Der Messvektor $\mathbf{m}(k)$ besteht aus den Koordinaten von einzelnen Punkten einer Konturgruppe im Bild. Die gemessenen Werte korrespondieren mit den projizierten Werten wie folgt:

$$\mathbf{m}(k) = \begin{pmatrix} K_{minx} \\ K_{maxx} \\ G_x \\ G_y \\ G_{minx} \\ G_{maxx} \\ G_{miny} \\ G_{maxy} \\ K_{maxy} \end{pmatrix} \leftrightarrow \begin{pmatrix} p1_x \\ p2_x \\ g_x \\ g_y \\ p1_x \text{ oder } p3_x \\ p4_x \text{ oder } p2_x \\ p2_y \text{ oder } p1_y \\ MAX(p3_y, p5_y) \text{ oder } MAX(p4_y, p6_y) \\ p7_y \text{ oder } p8_y \end{pmatrix} = h(\mathbf{s}^*) \quad (5.5.21)$$

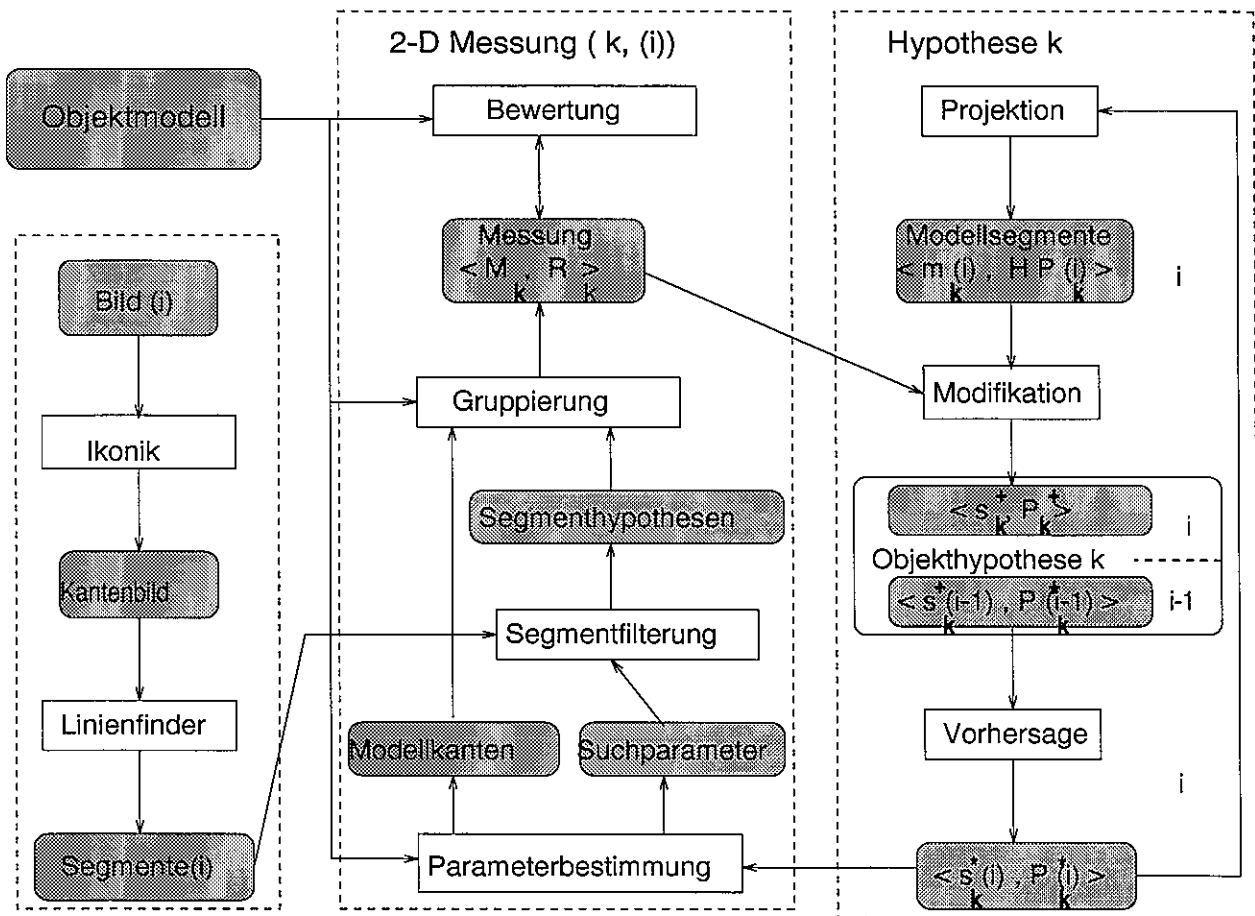


Bild 5.5.3: Die 2-D kantenbasierte Objektverfolgung für eine Objekthypothese

Wenn das Objekt nach links vom Beobachter gedreht ist, dann wird im Bild nach folgenden Zuordnungen gesucht:

$$p1_x \leftrightarrow (K_{minx} = G_{minx}), (p2_x, p2_y) \leftrightarrow (K_{maxx}, G_{miny}), \tag{5.5.22}$$

$$MAX(p3_y, p5_y) \leftrightarrow G_{maxy}, p4_x \leftrightarrow G_{maxx}, p7_y \leftrightarrow K_{maxy} \tag{5.5.23}$$

Wenn das Objekt nach rechts gedreht ist, dann wird somit im Bild nach folgenden Zuordnungen gesucht:

$$p2_x \leftrightarrow (K_{maxx} = G_{maxx}), (p1_x, p1_y) \leftrightarrow (K_{minx}, G_{miny}), \tag{5.5.24}$$

$$MAX(p4_y, p6_y) \leftrightarrow G_{maxy}, p3_x \leftrightarrow G_{minx}, p8_y \leftrightarrow K_{maxy} \tag{5.5.25}$$

Für eine genauere Beschreibung der Abbildungsfunktion und ihrer Jacobimatrix wird auf den Abschnitt 9.4 im Anhang verwiesen. Die Abstimmparameter im dynamischen Modell werden in Abschnitt 9.5 erläutert.

5.5.3 Der Ablauf der 2-D Messung

Der Ablauf der 2-D Messung und ihre Einbettung in die Objektverfolgung für eine Objekthypothese k zum Zeitpunkt i ist schematisch in Bild 5.5.3 dargestellt. Folgende Schritte können dabei unterschieden werden:

1. Suchbereichbestimmung für Objektvorhersage und Objektmodell
2. Modellkantenbestimmung

3. Ausfiltern von relevanten Liniensegmenten
4. Gruppierung von ausgewählten Liniensegmenten
5. Bewertung dieser Gruppe

Suchbereichbestimmung

Im ersten Schritt soll der Suchbereich und die zu erwarteten Richtungen im Bild geeignet eingeschränkt werden. Dabei gilt es zwei Forderungen optimal zu berücksichtigen:

1. Der ermittelte Bereich muß auch bei ungenauen aber prinzipiell richtigen Objekthypothesen eine Messung ermöglichen, damit eine Objektverfolgung nicht unterbrochen wird.
2. Eine unsichere Objekthypothese darf nicht zu einem zu großen Suchbereich führen, um Objektsprünge auf benachbarte Objekte zu vermeiden.

Punkt 1 bedeutet, daß eine Objekthypothese, die falsch (zu klein/groß, falsche Neigung oder Position im Raum) vorhergesagt wurde, die Möglichkeit haben muß, sich an das tatsächlich vorhandene Objekt anzupassen. Daher muß der Suchbereich, in dem nach relevanten Liniensegmenten gesucht wird, entsprechend groß sein. Punkt 2 beinhaltet eine gegensätzliche Forderung. Wird der Suchbereich zu groß, so steigt die Gefahr, daß falsche Vorhersagen eine gute Messung erhalten, da u.U. Liniensegmente zur Messung herangezogen werden, die jedoch zu anderen Objekten gehören.

Der Suchbereich von Liniensegmenten einer verfolgten Objekthypothese wird innerhalb der Bildebene durch folgende Faktoren bestimmt:

- Entfernung der Objekthypothese.
- Maximal mögliche Größe der Objekthypothese.
- Maximale Fehler der Schätzung von Θ , V und der Formparameter für die zu verfolgende Objekthypothese.

Hieraus wird errechnet, in welchem 3-D Gebiet sich die Objekthypothese aufhalten kann. Die Projektion der Eckpunkte dieses Gebietes in die Bildebene bestimmt die Größe des Suchbereichs.

Modellkantenbestimmung

Nicht alle Kanten eines Modells sind immer sichtbar. Wir betrachten hier nicht den Fall, daß das Objekt selbst von einem anderen Objekt verdeckt ist. Dies kann auf der Abstraktionsebene, auf der gearbeitet wird, nicht entschieden werden. Vielmehr sollen die Kanten des Modells eliminiert werden, die durch das Modell selbst verdeckt werden.

Das Streifenmodell ist von diesen Überlegungen nicht betroffen, da hier keine Kanten verdeckt werden. Anders ist dies beim Hindernismodell. Da – abhängig von der Betrachtung – einzelne Flächen nicht sichtbar sind (3 bis 5 der vorhandenen 6), werden mit Hilfe eines Algorithmus zur Auffindung nicht sichtbarer Flächen, die zugehörigen Kanten eliminiert.

Das Fahrzeugmodell besteht aus zwei Quadern. Für den oberen Quader wird der erwähnte Algorithmus ausgeführt. Für den unteren Quader nur für die Front- und Seitenflächen. Da dieses Modell konkav ist, kann es auch vorkommen, daß Teile des Modells Teile von Flächen verdecken. Aus diesem Grund würde man im allgemeinen Fall noch einen Algorithmus zur Bestimmung der Schnittpunkte zweier Linien benötigen. Da die hiervon betroffenen Kanten ohnehin wegen ihrer geringen sichtbaren Länge kaum ins Gewicht fallen, wird auf Sie verzichtet.

Ausfiltern relevanter Liniensegmente

Nachdem der Suchbereich ermittelt wurde, gilt es alle relevanten Liniensegmente, die zu einer Objekthypothese gehören können, aus der Menge aller im aktuellen Bild ermittelten Liniensegmente zu extrahieren. Diese Liniensegmente werden in einem lokalen Puffer gehalten und müssen bestimmten Anforderungen genügen:

- Die Menge der ausgewählten Liniensegmente muß möglichst klein sein, da die Effizienz des Abgleichs mit den Modellkanten hiervon abhängt.
- Es müssen möglichst alle zum Modell einer Objekthypothese passenden Liniensegmente ausgewählt werden.

Es ist offensichtlich, daß diese Forderungen gegensätzliche Ziele verfolgen. Die ersten Ansätze hierzu ordneten einer Linie eine "Sicherheit" zu, mit der ausgedrückt werden sollte, wie relevant sie für den Suchbereich sei. Hierzu wurde der Mittelpunkt der Linie und seine Lage zum Suchbereich betrachtet. Es wird also jedes Liniensegment ausgesucht, dessen Mittelpunkt sich im Suchbereich befindet.

Gruppierung von Segmenten – Modellabgleich

Nachdem sowohl die Daten der für den Abgleich relevanten Liniensegmente, sowie der sichtbaren Modellkanten ermittelt wurden, kann eine Messung durchgeführt werden. Diese soll unter den vorhandenen Liniensegmenten (Index s) diejenigen bestimmen, die den in die Bildebene projizierten Modellkanten (Index m) am ähnlichsten sind. Der Grad dieser Ähnlichkeit wird durch ein Abstandsmaß bestimmt.

$$L_d = c_1(\sqrt{(mx_m - mx_s)^2 + (my_m - my_s)^2} + |l_m - l_s|) + c_2 d(\Theta_m, \theta_s)^2 \quad (5.5.26)$$

wobei der Wertebereich von Θ im Intervall $[-\pi/2, +\pi/2]$ liegt und

$$d(\Theta_m, \Theta_s) = \begin{cases} |\Theta_m| + |\Theta_s| & , \text{ falls Summe} \leq \frac{\pi}{2} \\ \pi - (|\Theta_m| + |\Theta_s|) & , \text{ sonst} \end{cases} \quad (5.5.27)$$

Der Abstand L_d zweier Linien wird somit aus dem Abstand ihrer Mittelpunkte, ihrer Längendifferenz und ihrer Differenz der Ausrichtung errechnet. Die Faktoren c_1 und c_2 sorgen für eine Normierung und Gewichtung der einzelnen Größen. Will man die Größen normieren, so ist zunächst deren Wertebereich zu ermitteln. Hierbei gilt im Einzelnen:

$$d(\theta_m, \theta_s)^2 \leq (\pi/2)^2 = n_1; \quad \Delta m \leq \sqrt{S_x^2 + S_y^2} = n_2 \quad (5.5.28)$$

S_x und S_y kennzeichnen hierbei die Ausdehnung des Suchbereichs in x - und y -Richtung. Die unter 5.5.26 angegebenen Faktoren c_1 bis c_4 wurden wie folgt gewählt:

$$c_1 = \frac{1}{n_1}; \quad c_2 = \frac{1}{n_2}; \quad (5.5.29)$$

Die Ermittlung des Abstands eines Modells einer Objekthypothese zu den ausgesuchten Liniensegmenten geschieht mittels der gewichteten Summe der Abstände seiner Modellkanten, wobei $Lk_{ges} = \sum_{i=1}^n l_m(i)$ die Gesamtlänge der sichtbaren n Modellkanten l_i ist:

$$D_d = \frac{1}{Lk_{ges}} \sum_{i=1}^n L_d(i) l_m(i) \quad (5.5.30)$$

Somit haben wir unabhängig von der Größe eines Objektmodells den durchschnittlichen Abstand seiner Modellkanten. Dies entspricht aber genau dem Erwartungswert für einen Abstand Modellkante–Liniensegment. Die zugehörige Varianz ist entsprechend:

$$E_d = \frac{1}{Lk_{ges}} \sum_{i=1}^n l_m(i) (L_d(i) - D_d)^2 \quad (5.5.31)$$

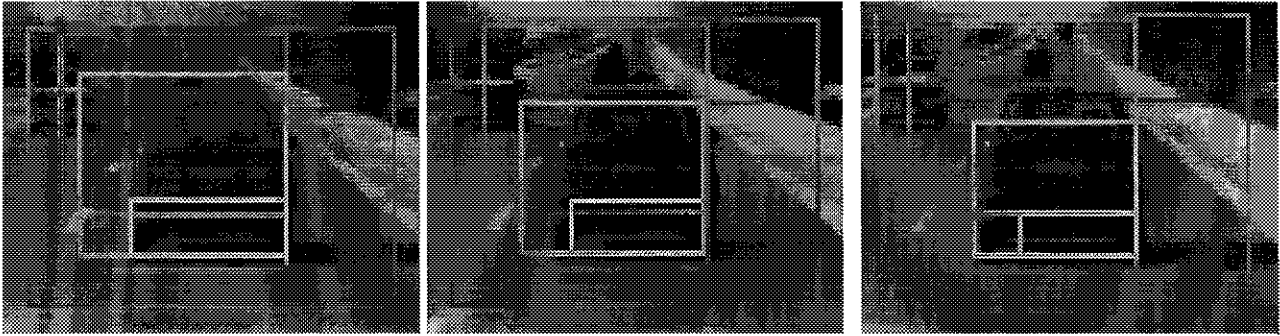


Bild 5.5.4: Konvergieren der Projektion von Modellalternativen im Bild: nach einer Iteration (links), nach zwei (mitte) und nach drei Iterationen (rechts) (das äußerste Rechteck enthält den Suchbereich, das kleinere die Vorhersage und das hellste die Messung).

Die Meßvarianz

Die weitere wichtige Aufgabe besteht darin, die Güte der gefundenen Messung zu bewerten. Da beide Größen in die Bewertung der Güte der Messung eingehen, ist es notwendig ein Verfahren zu finden, mit dem diese Messung möglichst unabhängig von der vorhergesagten Objekthypothese wird. Angenommen, es wird eine falsche Vorhersage einer Objekthypothese geliefert, die beim Modellabgleich eine gute Messung ergibt. Würde man sich nun lediglich auf die Messung der Vorhersage beziehen und sie bewerten, so käme man nur schwer von einer schlechten Vorhersage weg.

Aus diesem Grund werden durch Ändern aller Parameter (bis auf ω und V) des Zustandsvektors s^d sogenannte Modell-Alternativen generiert, die gleichberechtigt in Konkurrenz zur Vorhersage stehen. Von jeder dieser Alternativen wird ihr Modellabstand zu den vorhandenen Liniensegmenten gemessen. Die Modell-Alternative mit dem kleinsten Abstand wird ausgewählt. Abbildung 5.5.4 zeigt die Adaption der Vorhersage an das tatsächlich vorhandene Objekt. Dieses Verfahren läßt sich natürlich beliebig oft iterieren. Hierbei wird dann der modifizierte Zustandsvektor als Ausgangsbasis für weitere Modell-Alternativen hergenommen.

Fahrzeuge und Fahrbahnstreifen haben eine bestimmte Form. Deswegen werden mit dem vorhandenen Modellwissen weitere Tests auf Plausibilität der Objekthypothese durchgeführt (z.B. . Test auf Breite oder Höhe, Test auf das Breite-zur-Höhe-Verhältnis).

Wie bereits erwähnt, gehen zunächst 2 Größen in unsere Bewertung ein:

- Der durchschnittliche Abstand der Modellkanten.
- Die Varianz des Abstandes zwischen den Modellkanten und gewählten Liniensegmenten

Die Güte der Messung (Meßvarianz E_m) läßt sich folgendermaßen ausdrücken:

$$E_m = f_1 D_d + f_2 E_d; \quad (5.5.32)$$

Es ist darauf zu achten, daß die Werte der Kovarianzmatrix Q des Systemrauschens in entsprechender Relation zu den Meßvarianzen stehen. Da diese Meßvarianzen als Angabe der Güte einer Messung in den Kalmanfilter eingehen, dürfen sie insbesondere nicht kleiner werden als das Systemrauschen. Dies würde bedeuten, daß dann die zugehörige Messung besser wäre als die Vorhersage. Insbesondere muß gelten:

$$HQH^T < R \quad (5.5.33)$$

\mathbf{H} ist hierbei die Jakobimatrix der Zustandübergangsfunktion h des Messvektors \mathbf{m} und \mathbf{R} seine Kovarianzmatrix. Durch Auswertung der Zeilen der Jakobimatrix läßt sich ein Zusammenhang zwischen \mathbf{Q} und \mathbf{R} ermitteln. Für die richtige Wahl der Elemente von \mathbf{R} muß dann die Größenordnung der Summanden D_d und E_m skaliert werden.

5.6 Objektauswahl

Die Menge der Objekthypothesen wird zum Schluß der Verarbeitung daraufhin untersucht, ob konkurrierende Objekthypothesen vorkommen, ob eine Konkurrenzaufloßung möglich und sinnvoll ist und welche Objekthypothesen aus der Phase der Hypothesenverfolgung in die Phase des erkannten Objektes übernommen werden können.

5.6.1 Der Konsistenztest

In drei Schritten erfolgt die Auflösung von konkurrierenden Hypothesen;

1. Auswahl von Hypothesen die Konsistenzfähig sind, d.h. welche die Erkennungsbedingung der Zeitlänge erfüllen
2. Anwendungsabhängige Detektion der Konkurrenz von Hypothesenpaaren
3. Auswahl der besseren Hypothese und Eliminierung der schlechteren

Dank der Anwendung des Kalman filters bilden die über die Zeitlänge gewichteten Varianzen der Schätzwerte ein unmittelbares Auswahlkriterium. Falls zwei Hypothesen $Hypo_1, Hypo_2$ in t_1, t_2 Bildern verfolgt werden ($t_1 \geq t_2$) mit Varianzen E_1^*, E_2^* dann:

$$\text{IF } t_i > t_{max} \text{ (} i = 1, 2 \text{) THEN } T_i = t_{max} \text{ ELSE } T_i = t_i \text{ (} i = 1, 2 \text{)} \quad (5.6.1)$$

$$\text{IF } (E_{(1)}^* * T_1 \leq E_{(2)}^* * T_2) \text{ THEN wähle } Hypo_1 \text{ ELSE wähle } Hypo_2 \quad (5.6.2)$$

5.6.2 Der Erkennungstest

Der Übergang von der Phase *Hypothese* zu der Phase *erkanntes Objekt* hängt zusammen mit der Länge der Verfolgung und der Varianz des Objektzustandes. Ein erkanntes Objekt erfüllt zwei Bedingungen:

1. sie wurde ausreichend lange verfolgt
2. ihre Varianz fällt schnell nach unten und stabilisiert sich oberhalb des Wertes für Systemrauschen aber unterhalb einer vorgegebenen Varianzschwelle.

Schlecht bewertete Messungen für eine Hypothese tragen zur Erhöhung der Varianz bei, was beim Überschreiten dieser Schwelle diese Hypothese in die Verfolgungsphase zurückwirft. Der Ausfall von Messungen für eine Hypothese bedeutet neben der Erhöhung der Schätzvarianz auch eine schnelle Kürzung der Verfolgungszeit. Dadurch wird die Zurückweisung der Hypothese noch bekräftigt.

Im Unterschied zum Konsistenztest für Hypothesen in der Verfolgungsphase, werden alle Hypothesen zum Konsistenztest gegen eine erkanntes Objekt herangezogen. Dadurch wird das Generieren von neuen konkurrierenden Hypothesen eines gut erkannten Objektes erschwert.

Kapitel 6

Eine wissensbasierte Realisierung

In diesem Kapitel wird das Konzept eines wissensbasierten Systems zur adaptiven Objekterkennung vorgestellt. Als erstes wird die Struktur des Systems, basierend auf der Wissensform der semantischen Netze, erläutert. Die Probleme einer optimalen Zusammensetzung von Modell und Kontrolle werden in Abschnitt 6.2 behandelt. Die aus diesen Überlegungen resultierenden Entwürfe des Modells und der schritthaltenden Kontrolle sind in Abschnitten 6.3 und 6.4 beschrieben.

6.1 Die wissensbasierte Systemstruktur

Als Entwicklungsumgebung dient die System-Shell *ERNEST* [NSSK90], welche wissensbasierte Realisierungen auf der Basis der *Prozeduralen Semantischen Netzwerke* unterstützt. Ein solches System gliedert sich in *Wissensbasis* und *Kontrolle* (Bild 6.1.1).

6.1.1 Die Wissensbasis

In der Wissensbasis wird zwischen dem *Modell* und den *Daten* sowie den *Inferenzregeln* zur Aktivierung des Modells unterschieden. In einem prozeduralen semantischen Netzwerk werden sowohl das spezifische Wissen zur Fahrbahn- und Objekterkennung, wie auch die anwendungsunabhängigen Methoden des Segmentierers-Bewegungsdetektors strukturiert dargestellt. Das Netz ist implementiert in einer Programmiersprache und dabei abgebildet auf Datenstrukturen, genannt *Konzepte*, mit zugehörigen Prozeduren.

Der deklarative Teil des Modells ist eine rahmen-artige Implementierung eines semantischen Netzwerkes. In das Modell wird prozedurales Wissen eingebunden, das aus sog. Berechnungs- und Bewertungsfunktionen besteht. Die erstgenannten dienen zur Ermittlung von Attributen- und Relationenwerten der Konzepte in der aktuellen, derzeit analysierten Szene. Die Aufgabe der Bewertungsprozeduren besteht darin, ein Maß für die Güte der Inferferenzergebnisse zu erzeugen, anhand derer die Kontrolle ihre Suche nach der besten Interpretation steuert.

Die Generierung von Analyseergebnissen (*Daten* in Form von *Instanzen* und *modifizierten Konzepten*) erfolgt auf Grund der Verknüpfung der Konzepte mit den Eingabedaten und bisherigen Daten. Diese Verknüpfung findet mit Hilfe von einigen anwendungsunabhängigen *Inferenzregeln* statt. Das Modell wird durch einige anwendungsunabhängige Inferenzregeln aktiviert, die einen Abgleich zwischen dem Modell und bereits vorhandenen Daten durchführen. Als Ergebnis der Verknüpfung von Modell und Daten werden neue Zwischenergebnisse erzeugt (Instanzen, modifizierte Konzepte) [Sag90]. Es stehen uns fünf IF-THEN Regeln zur Verfügung, die entlang der *Bestandteil-* und *Konkretisierungskanten* agieren und eine Vererbungsregel, die entlang der *Spezialisierungskanten* aktiv ist. Die Inferenzregeln sind syntaktisch orientiert und

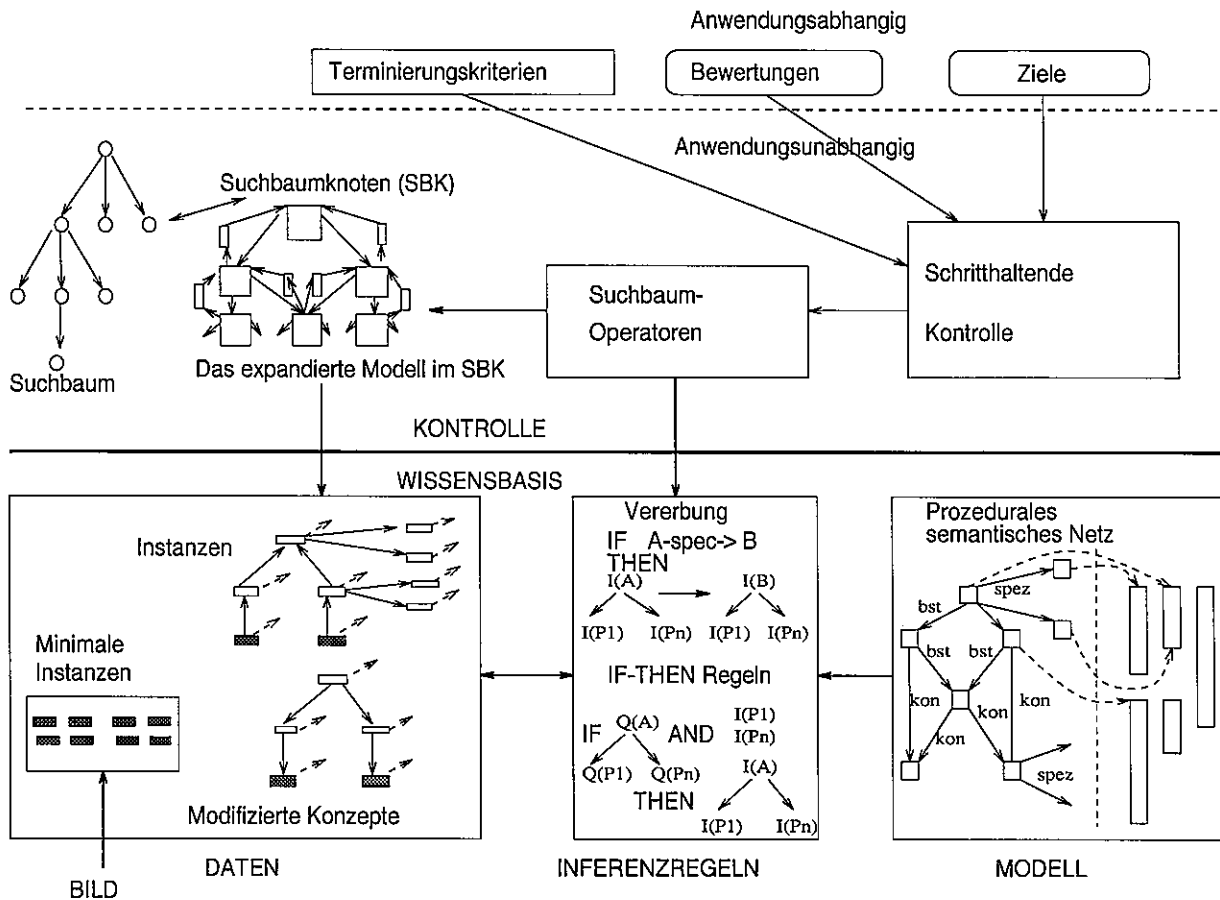


Bild 6.1.1: Wissensbasierte Realisierung des Erkennungssystems

gelten für das gesamte Netzwerk. Drei von den IF-THEN Regeln sind sogenannte *Instantiierungsregeln* (sie sind "bottom-up" orientiert) und die beiden restlichen sind *Modifizierungsregeln* ("top-down" oder "bottom-up").

Beispiel 6.1: In Bild 6.1.2 sind ein exemplarisches Netz mit anfänglichen Daten (links) und mögliche Analyseergebnisse (rechts) schematisch dargestellt. Das vorläufige Analyseziel besteht in der Instantiierung des Konzeptes *A*. Während der Initialisierung der Analyse wird ein modifiziertes Konzept $Q(A)$ des Konzeptes *A* generiert (Schritt *H1*). Im weiteren Verlauf der Analyse wird drei mal die top-down Modifizierungsregel für (modifizierte) Konzepte eingesetzt – dargestellt als die Inferenzen *R1*, *R2* und *R3*. Dabei werden die modifizierten Konzepte $Q(B)$ und $Q(C)$ als erste generiert, dann das modifizierte Konzept $Q_1(D)$ und zum Schluß $Q_2(D)$. Daneben werden fünf mal die "bottom-up" Instantiierungsregeln aufgerufen (Inferenzen *V1*, ..., *V5*), welche die Instanzen $I_1(D)$, $I(B)$, $I_2(D)$, $I(C)$ und $I(A)$ erzeugen. Dabei werden die bereits möglichen Instantiierungen gleich durchgeführt, ohne die vollständige Expandierung des Modells abzuwarten. In unserem Beispiel bedeutet das, daß die Inferenzen *V1* und *V2* unmittelbar nach Ablauf von *R2* und vor *R3* stattfinden. Nachdem das vorläufige Analyseziel – die Instantiierung des Konzeptes *A* – erreicht ist, werden im Schritt *H2* modifizierte Konzepte von übergeordneten Analysezielen generiert. Durch Aufruf einer "bottom-up" Modifizierungsregel werden aus $I(A)$ die modifizierten Konzepte $Q_1(E)$, $Q_2(E)$ und $Q(F)$ generiert.

Beachten wir, daß die Inferenz *H2* drei konkurrierende Analyseziele liefert. Grundsätzlich ist es möglich in jedem Inferenzschritt konkurrierende Instanzen bzw. modifizierte Konzepte

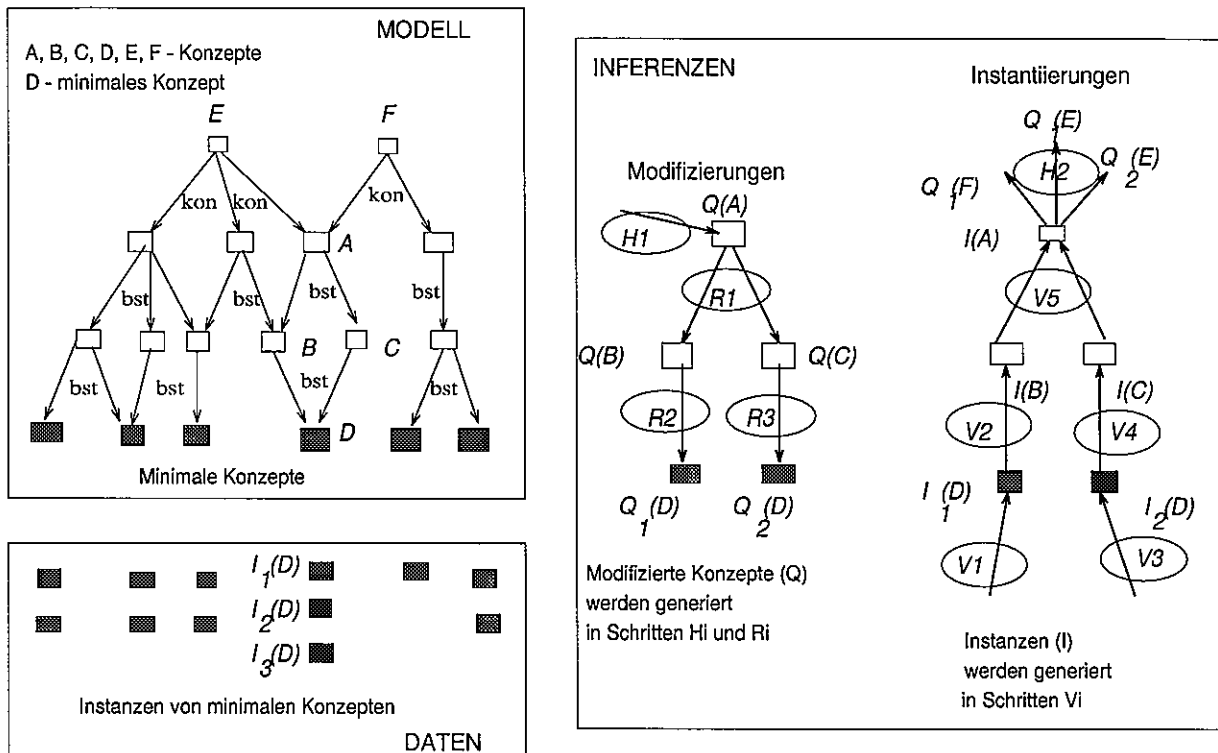


Bild 6.1.2: Die "top-down" und "bottom-up" Inferenzen

zu generieren. Das hängt von den Daten und der Gestaltung des Modells ab. Diese Tatsache haben wir in unserem Beispiel bisher nicht berücksichtigt, weil sie direkt mit der Kontrolle zusammenhängt. Die modifizierten Konzepte $Q(B)$ und $Q(C)$ waren nicht konkurrierend, sondern implikativ zueinander.

6.1.2 Die Kontrolle

Die allgemeine Verwendbarkeit der Kontrolle für verwandte Problembereiche, die in derselben wissensbasierten Sprache modelliert werden, ist eine Voraussetzung eines wissensbasierten Anwendungssystems.

Die Kontrolle besteht aus einem *anwendungsunabhängigen* Teil zur Verwaltung des Suchbaumes mit Suchbaumoperatoren zum Aufruf der Inferenzregeln, sowie aus *anwendungsabhängigen* Kontrollprozeduren. Ihre Aufgabe liegt in der Bestimmung der Reihenfolge von Inferenzschritten mit entsprechenden Modellelementen und Daten. Die Kontrolle wählt die passenden Bilddaten aus, stellt eine Zuordnung zwischen Modell und Bilddaten her, verfolgt je nach Genauigkeit der Zuordnung verschiedene Alternativen weiter oder verwirft sie und entscheidet schließlich, ob die ausgewählten Bilddaten dem Modell hinreichend genau entsprechen.

Die Aufgabe der Kontrolle liegt in der Verwaltung von konkurrierenden Analyseergebnissen, in der Auswahl der besten Ergebnisse im Sinne einer *Bewertungsfunktion* und in dem Anstoßen einer entsprechenden Inferenzregel. Benutzerabhängiges Kontrollwissen wird eingebracht durch die Strukturierung des Modells, durch die Attributberechnungsfunktionen und durch die Bewertungsfunktionen. Die Strukturierung des Modells führt während der Analyse zu einem *expandierten* Modell in Form eines attributierten UND/ODER Graphen ([NK90]. Da die Analyse meistens mit unvollständiger Information arbeitet, wird diese Ungewissheit in die Prozeduren der Konzepte übertragen. Das führt zur Generierung mehrerer konkurrierender Instanzen bzw. modifizierter Konzepte während der Analyse.

Die Basiskontrolle

Es steht uns eine überwiegend anwendungsunabhängige *Basiskontrolle* für statische Analyse zur Verfügung, welche die Reihenfolge der Inferenzen bestimmt und die Verwaltung des Suchraumes und der Ergebnisse übernimmt [Kum92]. Die Basiskontrolle zeichnet sich wie folgt aus:

1. *nicht deterministisch* – keine starre Reihenfolge der Verarbeitung, sondern abhängig von aktuellen Zwischenergebnissen.
2. *akternierende* Analysestrategie im Bezug auf das Modell.

Der Verarbeitungsfluß variiert ständig zwischen "top-down" Expandierung des Modells und "bottom-up" Instantiierung von expandierten Konzepten, sowie "bottom-up" Auswahl von übergeordneten Zielkonzepten.

Die Analysestrategie als Suche im Zustandsraum

Die alternierende Analysestrategie erklären wir mit Hilfe des allgemein bekannten Begriffes der Suche im *Zustandsraum* [Nil71]. Bei der Bildanalyse haben wir es überwiegend mit Problemen zu tun, für die jeweils mehrere Lösungen existieren können, d.h. wir können kein streng *deterministisches* Vorgehen angeben. Deswegen wird der sogenannte *Zustandsraum* als fundamentale Organisationsform aller zielorientierten symbolischen Aktivitäten angenommen. Er besteht aus einer Menge von Zuständen und Umformungsschritten. Der aktuelle Zustand ist eine Sammlung von Fakten und Zwischenergebnissen. Die Umformungsschritte sind Anwendungen von *Operatoren*, z.B. von Regeln in einem regelbasierten System. Im Zusammenhang mit der Basiskontrolle können wir auch von *Bereichen* im Zustandsraum und von Operatoren *zwischen* Bereichen sprechen. Die Bereiche werden implizit spezifiziert durch modellbedingte Beschränkungen (Expandierung des Modells verbunden mit Modifizierung von Konzepten). Die Lösung eines Problems entspricht einer Bewegungsstrategie innerhalb des Zustandsraumes, die einen Anfangszustand mit einem Zielzustand verbindet. Es gibt unterschiedliche *Problemlösungsstrategien* [Nil82], z.B.: die Vorwärtsstrategie, die Rückwärtsstrategie, "Hypothesisiere-und-Teste".

Beispiel 6.2: Sehen wir jetzt, wie die Inferenzschritte aus dem Beispiel in Bild 6.1.2, den Analyseschritten im Zustandsraum entsprechen. Die letzten sind in Bild 6.1.3 (a) schematisch dargestellt. $H1$ und $H2$ sind *Hypothesisierungsschritte*, die eine implizite Menge von möglichen (vorläufigen) terminellen Zuständen, erreichbar vom aktuellen Startzustand, erzeugen. Ein *Testschritt* für $H1$ im Zustandsraum bedeutet hier eine lokale Analyse zwecks Instantiierung des modifizierten Zielkonzeptes $Q(A)$. Dabei entspricht eine Modifizierung der *Rückwärtsverkettung* zwischen (impliziten) Bereichen im Zustandsraum (Schritte $R1, R2, R3, R4$) und eine Instantiierung der *Vorwärtsverkettung* zwischen Zuständen (Schritte $V1, V2, V3, V4, V5$).

Die A^* -Graphensuche im Entscheidungsbaum

Die Entscheidungen über konkurrierende Ergebnisse, die während des Ablaufes einer Analysestrategie getroffen werden, sind als optimale Suche nach einem Pfad (Weg) im *Entscheidungsbaum* (*Suchbaum*) ausgelegt. Zu jedem Zeitpunkt der Analyse wird der aktuelle Zustand (bzw. der aktuelle Bereich im Problemraum) implizit definiert durch die Datenmenge des aktuellen Suchbaumknotens.

Beispiel 6.3: In Bild 6.1.3 (b) wird dem dargestellten Analyseablauf sein Suchbaum zugeordnet. Die Schritte $H1, R1, R2, V1, V2, R3, V3, V4, V5$ und $H2$ entsprechen einem Pfad im Suchbaum.

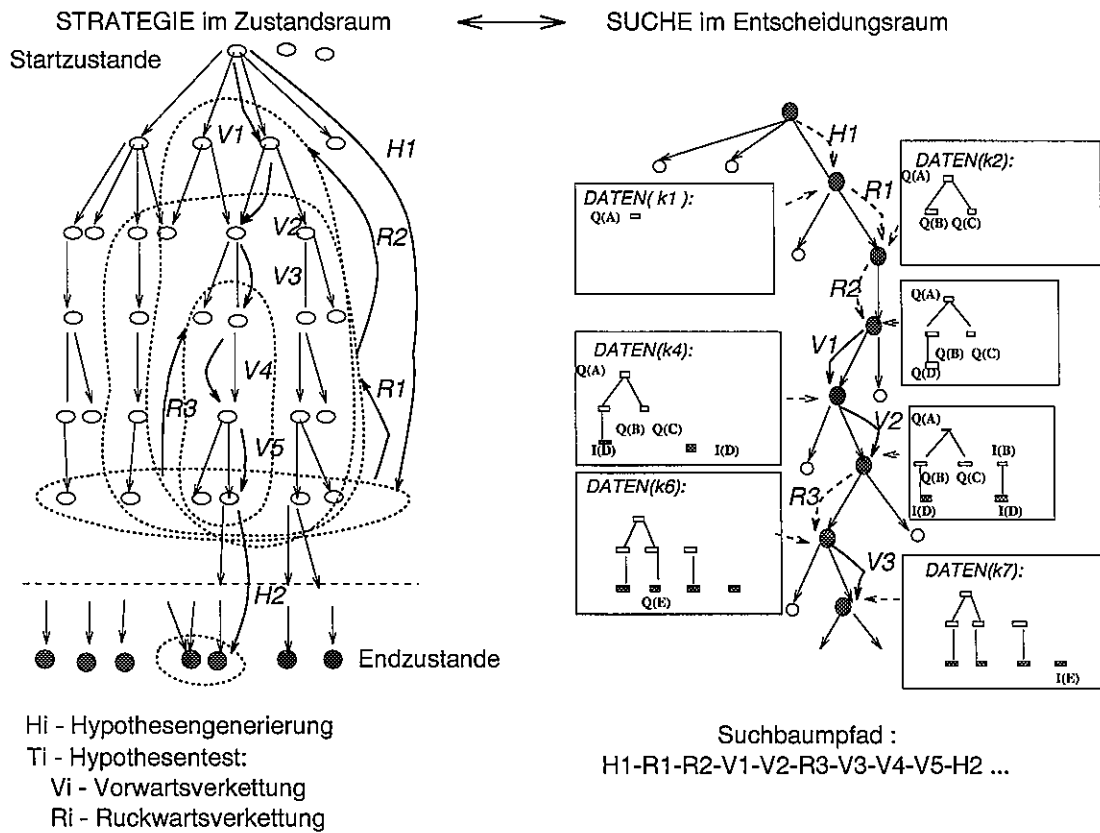


Bild 6.1.3: Die alternierende Analyse als Bewegung im Zustandsraum und die Verwaltung von konkurrierenden Analyseergebnissen als Suche im Entscheidungsbaum

Gemäß Nilsson[Nil71] wird mit A eine *Bestensuche* für endliche Graphen mit additiver Kostenfunktion $f(n) = g(n) + h(n)$ bezeichnet. Dabei bedeuten g die bisherigen Kosten und h (sog. Heuristik) die Restkosten des Lösungspfades im Knoten n .

Folgende *Klassen* der Heuristik $h(n)$ für einen Graphen \mathcal{G} werden gewöhnlich unterschieden:

- Eine Heuristik $h(n)$ ist *zulässig* wenn $h(n) \leq h^*(n), \forall n \in \mathcal{G}$.
- Eine Heuristik h_2 ist *mehr informiert* als h_1 wenn beide zulässig sind und $h_2(n) > h_1(n), \forall n \in \mathcal{G}$ (außer für Zielknoten).
- Eine Heuristik $h(n)$ ist *konsistent* wenn $h(n) \leq c(n, n') + h(n'), \forall \text{Pfade } (n, n') \in \mathcal{G} \times \mathcal{G}$, wobei $c(n, n')$ sind die Kosten des Pfades von n nach n' .
- Eine Heuristik $h(n)$ ist *monoton* wenn $h(n) \leq c(n, n') + h(n'), \forall \text{Kanten } (n, n') \in \mathcal{G} \times \mathcal{G}$.
Monotone und *konsistente* Heuristiken sind äquivalente Begriffe.

In [Nil82] wird der A^* -Algorithmus als ein A -Algorithmus definiert, mit der Bedingung, daß $h(n) \leq h^*(n)$ für jeden Knoten $n \in \mathcal{G}$ erfüllt ist, d.h. h zulässig ist. In der neueren Literatur [BM83], [Pea84], [DP88] wird der Standpunkt vertreten, daß ein Algorithmus vielmehr dadurch charakterisiert wird, WIE er die Daten verarbeitet und nicht durch den TYP der verarbeiteten Daten. Somit gäbe es keinen Unterschied zwischen A und A^* (Bild 6.1.4). Bei unzulässiger Heuristik ist bei der A -Suche folgendes wichtig: bei zwei Pfadem zu demselben Knoten wird der bessere Pfad auf der Basis der Funktion g und nicht f ausgewählt (Schritt 7.b). Die Funktion f wird zur Auswahl des besten Knoten aus OFFEN (Schritt 3) verwendet.

A, A^* :	
(1) bringe s nach OFFEN; setze $g(s) \leftarrow 0, f(s) \leftarrow 0$	
(2) IF	OFFEN ist leer
THEN	STOP mit Fehler
(3) Entferne aus OFFEN den Knoten n mit kleinstem f -Wert und bringe n nach GESCHLOSSEN (Löse Schlingen zugunsten eines Zielknoten auf)	
(4) IF	n ist ein Zielknoten
THEN	STOP mit aktuellem $g(n)$ Wert und Lösungspfad von s nach n
(5) Expandiere n (falls keine Nachfolger dann \rightarrow Schritt 2)	
(6) FOR jeden Nachfolger n_i von n	
	$g_i \leftarrow g(n) + c(n, n_i)$
(7) FOR jeden Nachfolger n_i von n	
(7a) IF	$n_i \notin (\text{OFFEN} \cup \text{GESCHLOSSEN})$
THEN	bringe n_i nach OFFEN, $g(n_i) \leftarrow g_i, f(n_i) \leftarrow g_i + h(n_i)$
(7b) IF	$n_i \in (\text{OFFEN} \cup \text{GESCHLOSSEN}) \wedge (g(n_i) > g_i)$
THEN	$g(n_i) \leftarrow g_i, f(n_i) \leftarrow g_i + h(n_i)$; entferne den früheren Pfad $(s - n_i)$
	IF $n_i \in \text{GESCHLOSSEN}$
	THEN bringe n_i nach OFFEN
(8) \rightarrow Schritt 2	

Bild 6.1.4: Der $A (A^*)$ Algorithmus zur optimalen Graphensuche

Eine Variante von A^* , in der nach Erreichen eines bereits expandierten Knoten n_i (Schritt 7.b), die neue Bewertung auch gleich weiter propagiert wird ohne den Knoten von GESCHLOSSEN nach OFFEN zu bringen, wird in [Nil82] definiert. Bei Baumsuche ist diese Modifikation ohne Bedeutung, da der Knoten n_i immer neu ist und Schritt 7.b dann niemals stattfindet. Bei Graphensuche mit konsistenter Heuristik wird auch der Schritt 7.b niemals ausgeführt, weil die Relation $g(n_i) \leq g_i$ immer gilt.

Die schritthaltende Kontrolle

Die schritthaltende Kontrollkomponente beinhaltet die Basiskontrolle für Einzelbildanalyse, darüber hinaus aber auch weitere Elemente:

1. *schritt haltendes* Verarbeitungsprinzip – die Analysereihenfolge der Einzelbilder entspricht ihrer Reihenfolge in der Bildfolge.
2. *inkrementeller* Beschreibungsaufbau über die Zeit

Ein Teil der Instanzen zum Zeitpunkt k wird übernommen für die Analyse zum Zeitpunkt $k + 1$. Dadurch gelten die Ergebnisse der Verarbeitung von früheren Bildern als "Vorwissen" für die Interpretation des aktuellen Bildes.

3. *rekursive* Beschreibung dank einer zusätzlichen Inferenzart der *Instanzenauffrischung*.

Der zusätzliche Inferenzschritt, angewandt an eine "vorherige" Instanz, heißt *Auffrischung*. Somit können die Attribute dieser Instanz neu berechnet werden, unter Einbeziehen der vorherigen Ergebnisse, was zu einer erwarteten Steigerung der Qualität dieser Ergebnisse führen sollte.

6.2 Die optimale Modell–zur–Kontrolle Relation

Bevor wir zur Beschreibung des Modells und der schritthaltenden Kontrolle kommen, wollen wir die Problematik der dynamischen Objekterkennung ansprechen, unter dem Gesichtspunkt der möglichen Alternativen zum Modell und der schritthaltenden Kontrolle.

6.2.1 Der Analyseaufwand

Als *Datenaufwand* bezeichnen wir die Anzahl der während der Analyse generierten Instanzen (I) und modifizierten Konzepte (Q) (*Datenkomplexität*) multipliziert durch die durchschnittliche Zeit die benötigt wird zur Generierung einer Dateneinheit I oder Q .

Potentielle Gründe für die Datenkomplexität bei Einzelbildanalyse sind:

1. die Syntax des Modells (die Modellstruktur)
2. die syntaktischen Modell- und Analysealternativen (die Anzahl der konkurrierenden Modellpfade und Instanzen)
3. Wiederholung von gleichen oder ähnlichen Inferenzen

Zusätzliche potentielle Gründe der Datenkomplexität für eine Bildfolgenanalyse:

1. keine Fokussierung der Analyse im Laufe der Zeit – keine Lösungsübernahme von t nach $t + 1$, eine wiederholte Analyse, uneingeschränkt durch vorherige Ergebnisse
2. ungenügende Stabilisierung der Ergebnisse – zu frühes Verwerfen von alternativen Lösungen, keine oder nur kurze Verfolgungsphase einer ausgewählten Lösung

Als *Suchbaumaufwand* bezeichnen wir die Anzahl der generierten Suchbaumknoten (SBK) multipliziert durch den durchschnittlichen zeitlichen Aufwand zur Generierung eines Suchbaumknoten, Als *Verwaltungsaufwand* bezeichnen wir die durchschnittliche Zeit zur Auswahl eines Suchbaumknoten auf dem Lösungspfad multipliziert durch die Länge dieses Pfades.

Damit besteht der summarische *Analyseaufwand* aus der Summe von Datenaufwand, Suchbaumaufwand und Verwaltungsaufwand. Das Ziel eines optimalen Aufbaus des Modells und der Kontrolle ist es, diesen Aufwand möglichst gering zu halten.

6.2.2 Anforderungen an das Modell

Syntaktisch–semantische Modelldualität

Die syntaktisch–semantische Dualität der Modellierung bedeutet eine Modellierungsvielfalt zwischen zwei Grenzmodellen:

1. eine einfache Modellsyntax aber komplexe Semantik
2. komplexe Struktur des Modells aber einfache Semantik

Ein Problembereich kann also dual entweder durch mehrere "einfache" Konzepte oder durch wenige komplexe Konzepte modelliert werden. Die einfachen Konzepte enthalten elementare Datenstrukturen, wie Ganzzahl- oder Gleitkommavariablen, als Attributewerte. Im Grenzfall kann einem Konzept eine einzige Variable zugeordnet werden. Die komplexen Konzepte beinhalten viele komplexe Datenstrukturen, wie Matrizen oder Listen als Attributewerte, sowie viele aufwendige Attributberechnungsfunktionen. In diesem Grenzfall der Modellierung kann sich das Modell auf zwei Konzepte reduzieren – das eine Konzept modelliert die Eingabedaten, das

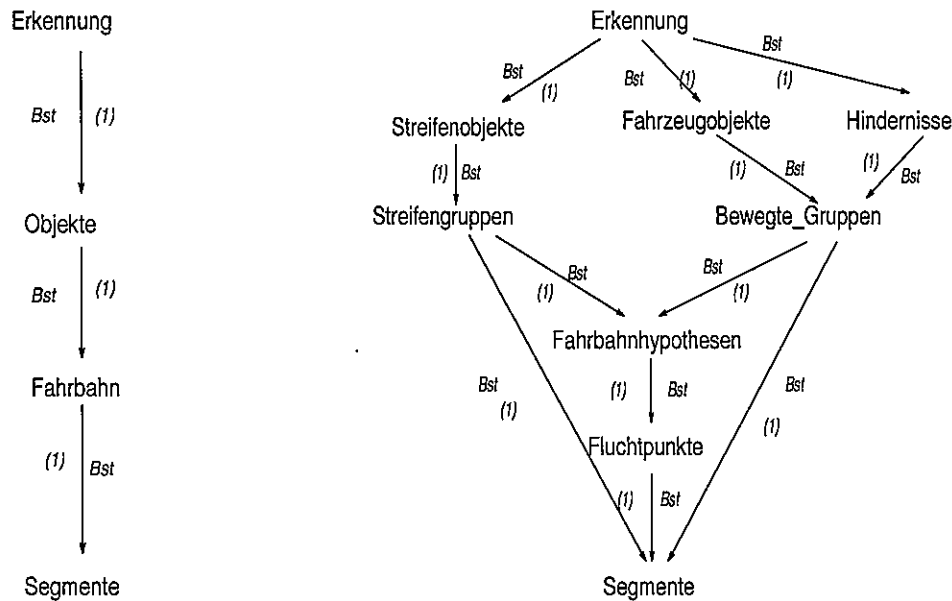


Bild 6.2.1: Zwei syntaktisch unterschiedliche Modellfälle.

andere – die Ergebnisse. Diese Dualität der Modellierung führt im allgemeinen zur unterschiedlichen Höhe des generierten expliziten Suchbaumes. Ein Beispiel in Bild 6.2.1 enthält syntaktisch unterschiedliche Modelle, die mit entsprechend unterschiedlicher Semantik ausgestattet zu demselben Analyseergebnis führen können – der Objekterkennung in Verkehrsszenen.

Die Modell- und Analysealternativen

Neben dieser Dualität der Modellierung können auch unterschiedliche Anzahlen von modifizierten Konzepten und Instanzen während der Analyse generiert werden. Die Modellexpansion kann zur unterschiedlicher Komplexität wie folgt führen:

- (A) keine konkurrierenden Konzeptmodifizierungen möglich
- (B) konkurrierende Konzeptmodifizierungen erwartet

Die Instanzengenerierung kann wie folgt zur Komplexität beitragen:

- (A) keine konkurrierenden Instanzen möglich
- (B) konkurrierende Instanzen erwartet

Beschränken wir uns weiterhin auf Instanzen, da für modifizierte Konzepte dieselben folgenden Aussagen zutreffen. Zu Konzepten, die Attributlisten enthalten wird während der Analyse überwiegend nur je eine Instanz generiert (Fall (A)). Die Auflösung der Inkonsistenzen muß durch ein internes Verfahren geschehen. Die möglichen konkurrierenden Analyseergebnisse werden alle in die Liste als Attributwerte solcher Instanz aufgenommen. Im zweiten Fall, anstatt einer Instanz mit einer Liste von Attributen, werden mehrere konkurrierende Instanzen mit je einem Attributelement aus dieser Liste generiert (Fall (B)). Zur Auflösung der Inkonsistenzen wird die anwendungsunabhängige Kontrolle verwendet. Falls die Struktur des Modells in beiden Fällen (A) und (B) gleich ist, ist auch die Höhe des expliziten Suchbaumes, der durch diese Kontrolle aufgebaut und verwaltet wird, gleich. Dafür aber ist der Suchbaum im Fall (B) viel breiter als im anderen Fall.

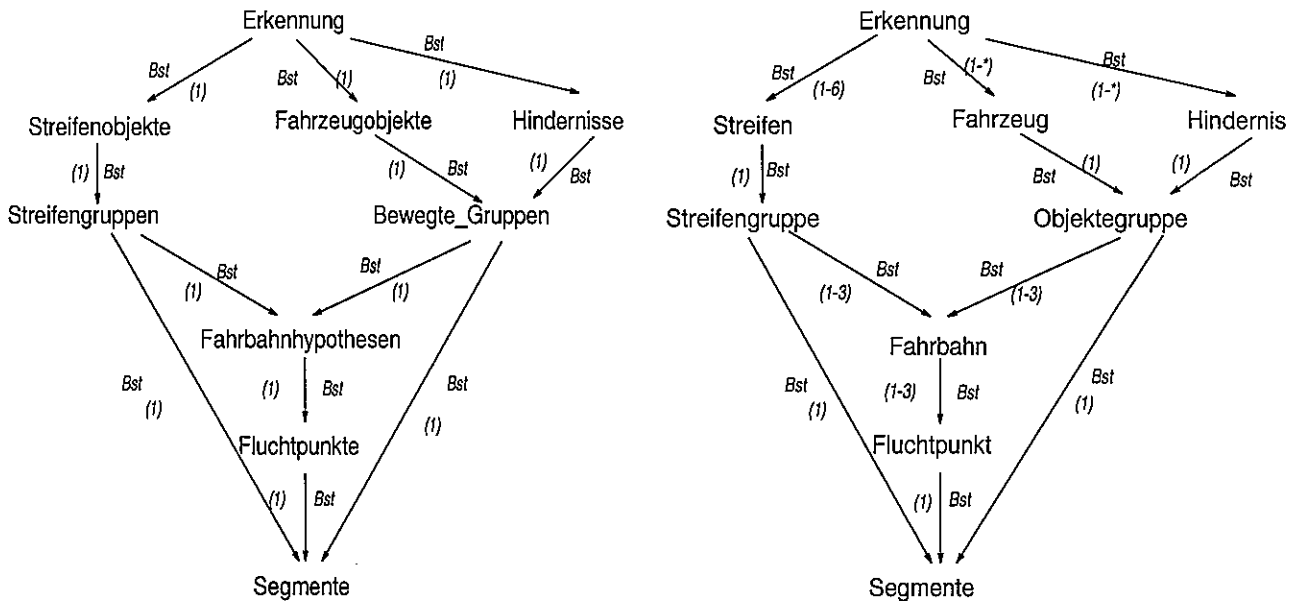


Bild 6.2.2: Zwei syntaktisch gleiche Modelle können zu unterschiedlicher Analysekomplexität führen.

Beispiel 6.4: Als Beispiel betrachten wir ein Konzept *Hindernis*, daß ein Attribut der Datenstruktur *Hindernisbeschreibung* enthält, und das Konzept *Hindernisse*, daß eine Liste dieser Datenstrukturen als Attributtyp enthält. Während der Analyse erwartet wird dann im ersten Fall, daß für jedes Konzept eine gesonderte Instanz generiert wird. Im zweiten Fall kommen wir u.U. nur mit einer einzigen Instanz aus.

Die Bild 6.2.2 enthält syntaktisch gleiche Modelle, die aber zu unterschiedlicher Komplexität der generierten Ergebnisse führen: das linke Modell – keine konkurrierenden Instanzen möglich, das rechte Modell – konkurrierende Instanzen werden erwartet.

Zu einfachen Konzepten werden gewöhnlich mehrere, konkurrierende Instanzen während der Analyse generiert. Auch hier wird zur Auflösung der Inkonsistenzen die anwendungsunabhängige Kontrolle verwendet.

Schlußfolgerung: Modell

Wir haben somit drei Modellfälle unterschieden:

- 2(A) Eine einzige Instanz – keine konkurrierenden Instanzen – keine Aufspaltung des Suchraumes; die Entscheidung über konsistente Untermengen wird innerhalb der Instanz getroffen.
- 2(B) Mehrere Instanzen, wobei jede Instanz konsistente Untermengen von Hindernishypothesen enthält – Suchbaumaufspaltung, gekürzte Suchbaumhöhe
- 1(B) Viele Instanzen für mehrere Konzepte des Modells – Suchbaumaufspaltung, hoher Suchbaumaufwand, da sequentielles Erzeugen von konsistenten Untermengen.

Falls alle Konzepte so aufgebaut sind, wie es der Fall (A) vorschreibt, dann wird während der Einzelbildanalyse nur ein einziger Suchbaumpfad generiert. Wir bezeichnen das im weiteren als *lineare* Einzelbildanalyse. Das Entscheidungsproblem wird hier verlagert vom allgemeinen expliziten Suchbaum in eine Menge von lokalen Entscheidungen innerhalb jedes Konzeptes. Dabei brauchen nicht alle lokalen Entscheidungen gleich getroffen werden. Die konkurrierenden Lösungen (oder ein Teil von ihnen) kann so lange verfolgt werden, bis eine übergeordnete

Entscheidung getroffen wird. Wegen des zu erwarteten Aufwandes wird das aber meistens nicht möglich sein.

Für statische Analyse besitzt eine suchbaumintensive Analyse klare Vorteile gegenüber der linearen Analyse. Die Voraussetzungen einer dynamischen, schritthaltenden Analyse sind aber anders als die der statischen Analyse. Es ist hier eine natürliche Wiederholbarkeit der Analyse gegeben. Deswegen kann ein einziger Lösungspfad komplett durchlaufen werden während einer Einzelbildanalyse.

Wir können uns damit zwei Grenzfälle einer wissensbasierten Realisierung der schritthaltenden Objekterkennung vorstellen: mit *linearer* und *suchbaumintensiver* Einzelbildanalyse. Die Struktur des semantischen Netzes des Modells ist in beiden Fällen gleich. Die Lösungen unterschieden sich nur durch die Komplexität der Konzepte. Der Zusammenhang zwischen Konzeptkomplexität und Suchbaumkomplexität ist ähnlich, wie die syntaktisch-*semantische* Dualität von Grammatiken.

6.2.3 Anforderungen an die Basiskontrolle

Die Basiskontrolle sollte einerseits die Analyse möglichst eingrenzen, andererseits aber eine mehrfache Generierung von inhaltlich gleichen Instanzen vermeiden.

Gleiche Inferenzen

Bei einer suchbaumbasierter Kontrolle kann es vorkommen, daß dieselbe Inferenz auf verschiedenen Suchbaumpfaden generiert wird. Dies ist bedingt entweder durch die Konkurrenz im Modell (sogenannte Modalitäten des Konzeptes) oder durch konkurrierende Instanzen zu früheren Konzepten auf dem Lösungspfad. Als *gleich* bezeichnen wir Inferenzen, die für inhaltlich dasselbe Datenobjekt (Konzept oder modifiziertes Konzept) als Ziel und mit denselben Prämissenobjekten (d.h. Instanzen und modifizierten Konzepten) durchgeführt werden.

Beispiel 6.5: In Bild 6.2.3 sind $Q1(C)$, $Q2(C)$ und $Q1(E)$, $Q2(E)$ Paare von gleichen Objekten. Dementsprechend sind sich gleich die Instantiierungen von $Q1(E)$ und $Q2(E)$, wie auch drei Paare von Instantiierungen ($Inst_i(Q1(C))$, $Inst_i(Q2(C))$) ($i=1, \dots, 3$).

Um wiederholte Inferenzen zu vermeiden, müßte ein *globales* Inferenznetz aufgebaut werden. Solch ein Netz könnte aber selbst zu komplex werden und der Aufwand zur Feststellung einer ähnlichen Instanz könnte größer sein als die wiederholte Generierung dieser Instanz.

Lokale contra globale Inferenzverwaltung

Der Unterschied zwischen einem Suchbaum mit konsistenten lokalen Inferenzmengen und einer globalen Verwaltung von (teilweise inkonsistenten) Inferenzen (wie das in den *Begründungsverwaltungssystemen* der Fall ist) ist graphisch dargestellt in Bild 6.2.4.

Schlußfolgerung: Basiskontrolle

Für Basiskontrolle der Einzelbildanalyse muß ein Optimum zwischen zwei Polen gefunden werden: der modellgesteuerten Kontrolle und der datengesteuerten Kontrolle: Die Vorteile von beiden Ansätzen liegen auf der Hand:

1. für ein syntaktisch komplexes Modell ist eine Strategie eines selektiven Modell-zu-Daten Abgleiches gefragt \Rightarrow hier ergeben sich Vorteile einer modellgesteuerten Kontrolle mit konsistenten lokalen Inferenznetzen

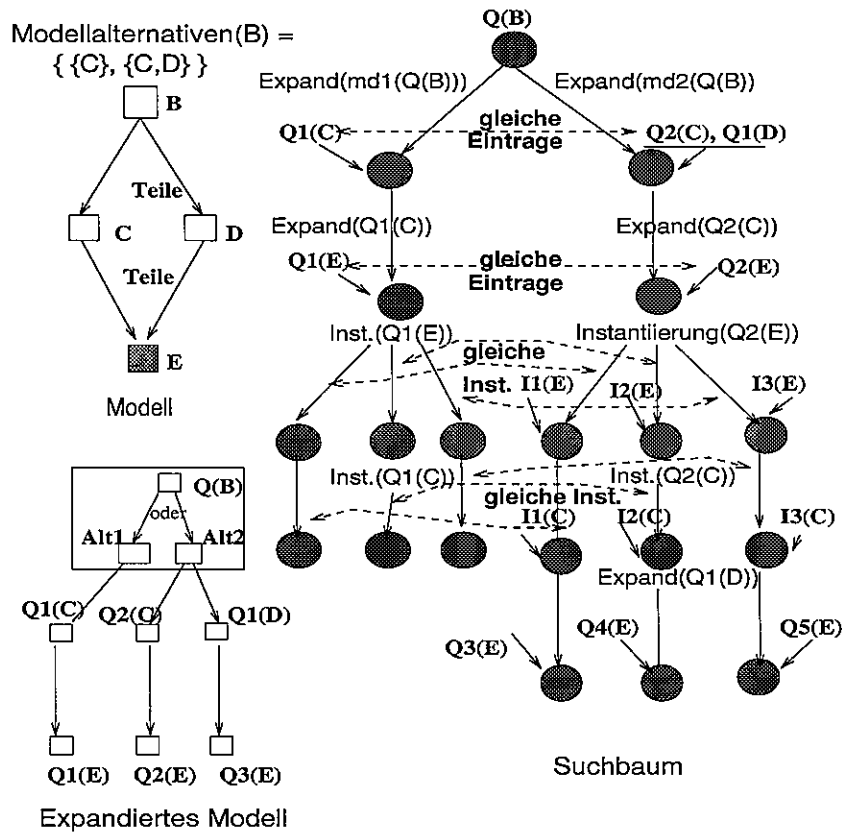


Bild 6.2.3: Gleiche Instantiierungen von gleichen Knoteneinträgen

2. falls semantisch komplexe Konzepte im Modell überwiegen, dann ist das Vermeiden von wiederholten Inferenzen gefragt \Rightarrow Vorteile einer datengesteuerten Kontrolle mit einem globalen Inferenznetz

6.2.4 Anforderungen an die Bildfolgenkontrolle

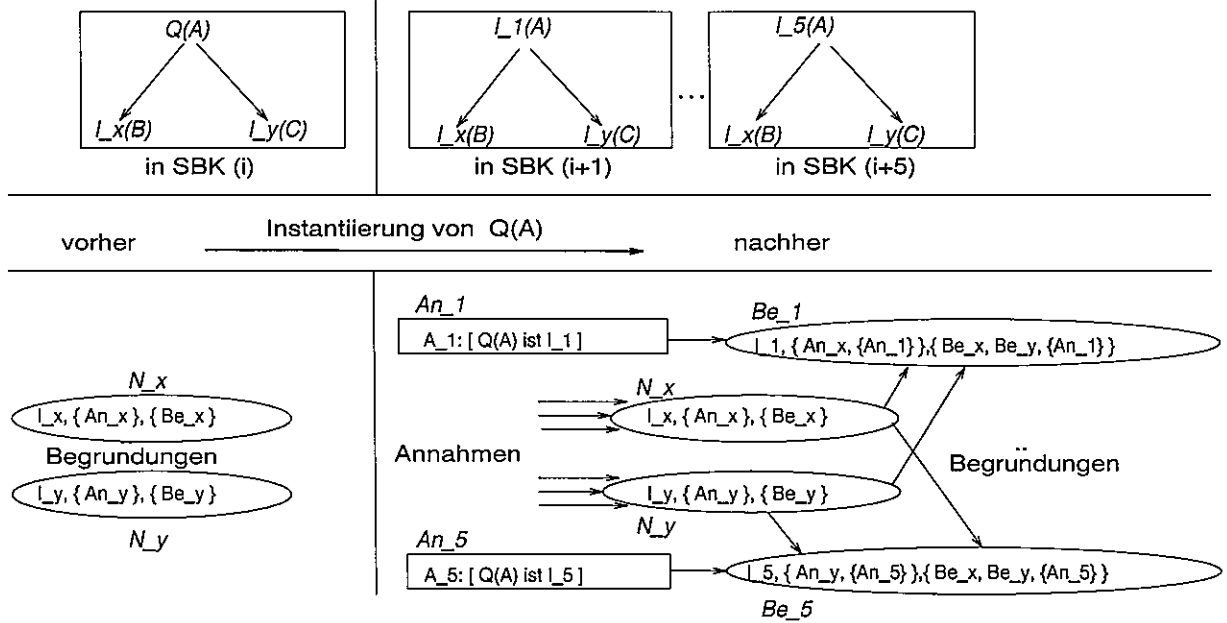
Eine schritthaltende Kontrolle soll

1. frühere Ergebnisse übernehmen und das Auffrischen mit neuen Daten steuern,
2. stabile Ergebnisse einer Bildfolgenanalyse liefern. Unter Umständen bedeutet dies eine Erstreckung der Initialisierungsphase über mehrere Bilder – eine Verfolgung von konkurrierenden Instanzen solange bis eine sichere Entscheidung möglich wird.

Das Auffrischen

Die Basiskontrolle ([Kum92]) enthält keine Möglichkeit zur Auffrischung von Instanzen. Sie wurde für Einzelbildanalyse angewendet (z.B. im Projekt PAP [NBSS90]). Die schritthaltende Kontrolle, die im Rahmen des Projektes MOVIE ([Wei92]) entwickelt wurde, konzentriert sich auf die Verfolgung von ausgewählten Suchbaumknoten (SBK). Die Auffrischung erfolgt global für alle übernommenen Instanzen am Anfang der Analyse jedes neuen Bildes. Der Nachteil dieser Lösung ist, daß beim Abweichen auf nicht übernommene Pfade, die ganze Analyse unter einem neuen Modellgleichzeitpunkt von Anfang an beginnen muß. Diese Version der schritthaltenden Kontrolle wurde zwar verwendet im Projekt MOSAIK ([Wet95]) zur Bildfolgenanalyse, aber in einer nur begrenzten Wirkungsform. Durch die Verlagerung der Modellalternativen

Lokale Inferenzverweise in SBKs:



Globale Inferenzverwaltung

Bild 6.2.4: Globale Inferenzverwaltung

und der Analysealternativen in das innere der Konzepte und Instanzen bleibt die syntaktische Analysekomplexität gering.

Stabile Lösungen

Am Beispiel von drei Bildern mit je 3 Objekten und je 3 Hypothesen zu einem Objekt (Bild 6.2.5, [Wet95]) werden die Unterschiede zwischen der bisherigen potentiell unstabilen Lösung (Bild 6.2.6, [Wei92]) und einer benötigten stabilen Lösung (Bild 6.2.7) deutlich.

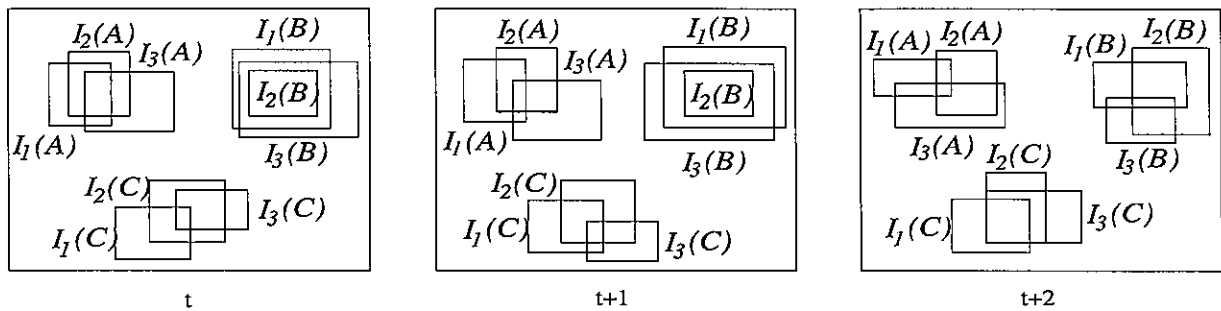


Bild 6.2.5: Beispiel von drei Objekten mit je 3 konkurrierenden Instanzen

Beispiel 6.6: Im Bild t wurde ein Suchbaumpfad für drei Instantiierungen erzeugt – jeweils eine Verzweigung für die Instantiierung von Konzepten A, B, C . Während jeder Instantiierung wurden drei konkurrierende Instanzen erzeugt. Zwecks Vereinfachung wird angenommen, daß sich diese Situation in den nächsten Bildern wiederholt. Das gibt einen Suchbaumknoten mit drei Instanzen $\{I_3(A), I_3(B), I_3(C)\}$ in Bild (t) , $\{I_1(A), I_3(B), I_2(C)\}$ in Bild $(t + 1)$ und $\{I_3(A), I_2(B), I_3(C)\}$ in Bild $(t + 2)$. Falls während der Analyse für

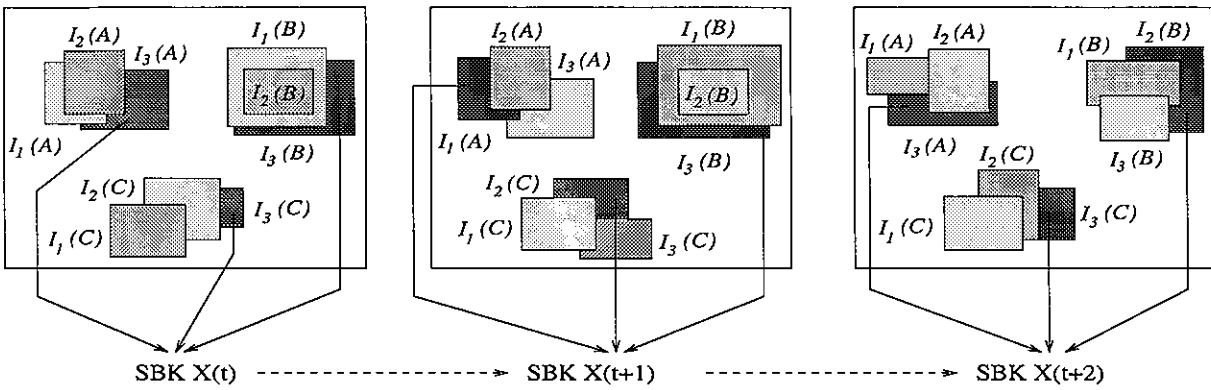


Bild 6.2.6: Beispiel einer potentiell instabilen Lösung

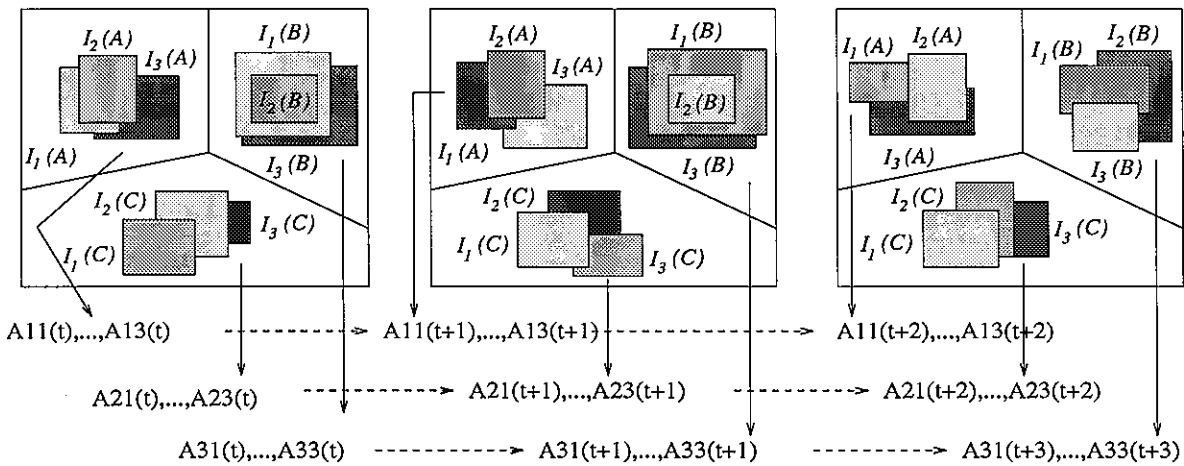


Bild 6.2.7: Beispiel einer benötigten stabilen Lösung.

das Bild $(t + 2)$ festgestellt wird, daß die Lösung im Knoten X_{ijk} falsch ist, gibt es keine Möglichkeit die Geschichte vom ersten Bild an zu wiederholen. Es sei denn alle konkurrierende Lösungen wurden die ganze Zeit parallel verfolgt und aufgefrischt, was aner entgegen dem Sinn der suchbaumbasierten Kontrolle ist. Es ist sehr wahrscheinlich, daß eine sichere Entscheidung am Analyseanfang nicht getroffen werden kann und daß 9 Lösungspfade mit insgesamt $3 \times 3 \times 3$ Instanzen (27 Suchbaumknoten) generiert werden.

Im unteren Bild sind Ergebnisse einer benötigten schritthaltenden Analyse dargestellt. Die Idee dieses Ansatzes basiert auf paralleler Verfolgung von alternativen Instanzen. In diesem Fall wird lediglich ein Pfad der Länge 3 mit 3×3 konkurrierenden Instanzen (9 SBKs) generiert. Nach einigen Bildern wird sich entscheiden welche Instanzen als sicher gelten und zur weiteren Verfolgung ausgewählt werden können. Die Anzahl der Konkurrenzen wird schrittweise abgebaut mit der Zuversicht, daß eine verworfene Hypothese später nicht wieder generiert werden muß. Zum Schluß bleiben nur die drei besten Instanzen übrig.

Schlußfolgerung: schritthaltende Kontrolle

Es wird eine schritthaltende Kontrolle benötigt, die folgende zum Teil sich ausschließende Bedingungen erfüllt:

1. gezielte Auswahl von Modell- und Datenmengen zur Analyse

2. ist optimal bezüglich des Analyseaufwandes sowohl für ein Einzelbild wie auch für eine Bildfolge
3. liefert zeitlich stabile Lösungen

Eine optimale suchbaumbasierte Kontrolle erfüllt am besten die erste Bedingung. Die Minimierung des Aufwandes der Einzelbildanalyse haben wir bereits angesprochen im Abschnitt 6.2.3. Im Kontext der dynamischen Analyse besitzt der Ansatz mit dem globalen Inferenznetz eine Überlegenheit gegenüber der suchbaumbasierten Kontrolle. Hier kann sehr flexibel der Zeitpunkt bestimmt werden, zu dem die Konkurrenzen aufgelöst werden, so daß eine ausreichende Stabilität der Lösungen gewährleistet ist.

In dieser Arbeit wird eine suchbaumbasierte Kontrolle entworfen. Die Gründe dafür sind zum Teil von pragmatischer Natur – es existiert bereits eine Basiskontrolle, die auf dieser Basis arbeitet. Aber auch die klare Einteilung der Analyseergebnisse in konsistente Untermengen ist von Vorteil, insbesondere dann, wenn durch entsprechende Struktur des Modells die beschriebenen Probleme der dynamischen Analyse vermieden werden können. Andererseits werden wir versuchen solche Elemente in die neue Kontrolle einzubauen, die uns auf dem Weg zur Stabilisierung der Analyse unterstützen.

6.3 Das Modell

Das in Bild 6.3.1 angegebene Schema des semantischen Netzes vereinigt das Wissen zur Segmentierung–Bewegungsschätzung, zur Fahrbahnerkennung und zur Objekterkennung.

Eine Reduzierung des Aufwandes während der Einzelbildanalyse, wie auch die Stabilität der gefundenen Lösungspfade wird durch entsprechende Syntax und Semantik des Modells gewährleistet. Durch Verringern der Modellvariabilität wird das Wiederholen von gleichen Inferenzen möglichst gering gehalten. Ein paralleles Verfolgen von mehreren Hypothesen zum Fluchpunkt, Fahrbahn und zu Objekten wird durch das Einfügen von entsprechenden Konzepten mit der Endung *xxx-hypo* möglich. Die Auswahl zwischen den konkurrierenden Hypothesen in diesen Fällen findet nicht innerhalb der Kontrolle statt, sondern innerhalb der Instantiierung von folgenden Konzepten: *FP* bzw. *FOE*, *Fahrbahn* und *Erkennung*. Die Reduktion der parallelen Verfolgung von konkurrierenden Lösungen erfolgt schrittweise, indem ein Übergang auf der horizontalen Modellebene der Spezialisierungen stattfindet. Dadurch können wir eine möglichst einfache (generelle) schritthaltende Kontrolle entwerfen, die suchbaumbasiert arbeitet und trotzdem die bisherigen Analyseprobleme vermeidet.

6.3.1 Die Struktur des Modellnetzes

Das Modell wird auf zwei Hierarchien des semantischen Netzwerkes abgebildet: die *Bestandteil-* und *Konkretisierungs-*Ebenen, sowie die *Spezialisierungs-*Hierarchie. Es können drei Konkretisierungsebenen und drei Spezialisierungsgrade unterschieden werden. Die Konkretisierungsebenen werden bezeichnet als: 3 – *D* (Objekterkennung), 2.5 – *D* (Fahrbahnerkennung) und 2 – *D* (Segmentierung und Verschiebungsdetektion). Die Spezialisierungsgrade entsprechen den drei Konzepten der höchsten Konkretisierungsebene 3 – *D* : *Erkennung*, *Erkennung_Form* und *Erkennung_Fein*. Diese Konzepte beschreiben das allgemeine Analyseziel zerlegt in drei zeitlich sequentielle Phasen.

Die 3 – *D* Ebene enthält auch im generellen Teil die Konzeptfamilie *Objekte* und das Konzept *Eigenzustand*. Die im Bereich der Fahrbahn sich befindenden Objekte werden in drei Klassen eingeteilt: *Streifen_Hypo* (Fahrbahnmarkierung), *Fahrzeug_Hypo* und *Hindernis_Hypo* (Objekte

die auf der Straße stehen bzw. sich bewegen). Eine Zunahme von *Umgebungs*-Objekten (Objekte, die neben der Straße stehen) sowie *Horizont*-Objekten (Objekte über der Fahrbahn) wäre prinzipiell auch möglich. Die Fahrzeugobjekte werden weiter spezialisiert in *PKW_Form* und *LKW_Form*. In der dritten Spezialisierungsstufe sind detaillierte Fahrzeugmodelle in Form von solchen Konzepten, wie *PKW_Fein* und *LKW_Fein* vorgesehen. Das 3 – D Modell besteht weiterhin aus der Konzeptfamilie *Gruppen* sowie den Konzepten *Kamera* (Kameramodell) und *Eigenzustand* (Eigenfahrzeugparameter).

Das 2,5 – D Modell beschreibt die Orientierung und Lage der Kamera bezüglich der Fahrbahn und besteht aus den Konzepten *Fahrbahn*, *Fahrbahn_Hypo*, *Kontur_Klasse* und vier Konzepten für die Fluchtpunkte *FP*, *FOE*.

Das 2 – D Modell umfaßt Konzepte für ikonische Verarbeitung (*Bild*, *Normal_Bild*, *Kantenbild* und *Dyn_Maske*), Konzepte zur Konturdetektion und Bewegungsschätzung (*Konturen*, *L_Segmente*, *Regionen* und *Bewegte_Konturen*). Hier sind auch enthalten: das aus der Projektion der vorherigen Objekte resultierende Konzept *Objektsegmente* sowie *2-D_Messung* und *Verif_L_Segmente* als Ergebnis des Abgleiches der erkannten Objekte mit Liniensegmenten im Bild.

6.3.2 Die Zyklensfreiheit des Modells

Eine wichtige Voraussetzung für das Terminieren einer Einzebildanalyse ist die Zyklensfreiheit des Modells. Dies widerspricht der Tatsache, daß für Konzepte der Klasse *Erkennung_Form* und *Erkennung_Fein* ein geschlossener Zyklus zwischen Objekten und Segmenten erfolgen muß. Andererseits wollen wir ein Objekt über die Zeit auch immer nur durch dieselbe Instanz darstellen. Um diesen beiden sich gegenseitig widersprechenden Forderungen Rechnung zu tragen, sollte der Modellierungszyklus geöffnet werden. Dies wird durch das explizite Einfügen von minimalen Konzepten der Familie *Vorherige_xxx* erreicht, z.B. *Vorheriges_Objekt*. Als Eingabedaten zum nächsten Zeitpunkt dienen dann neben dem aktuellen Bild auch minimale Instanzen dieser neuen Konzepte. Sie werden während der Analyseinitialisierung als Kopien von entsprechenden Instanzen zu "ähnlichen" Konzepten erzeugt.

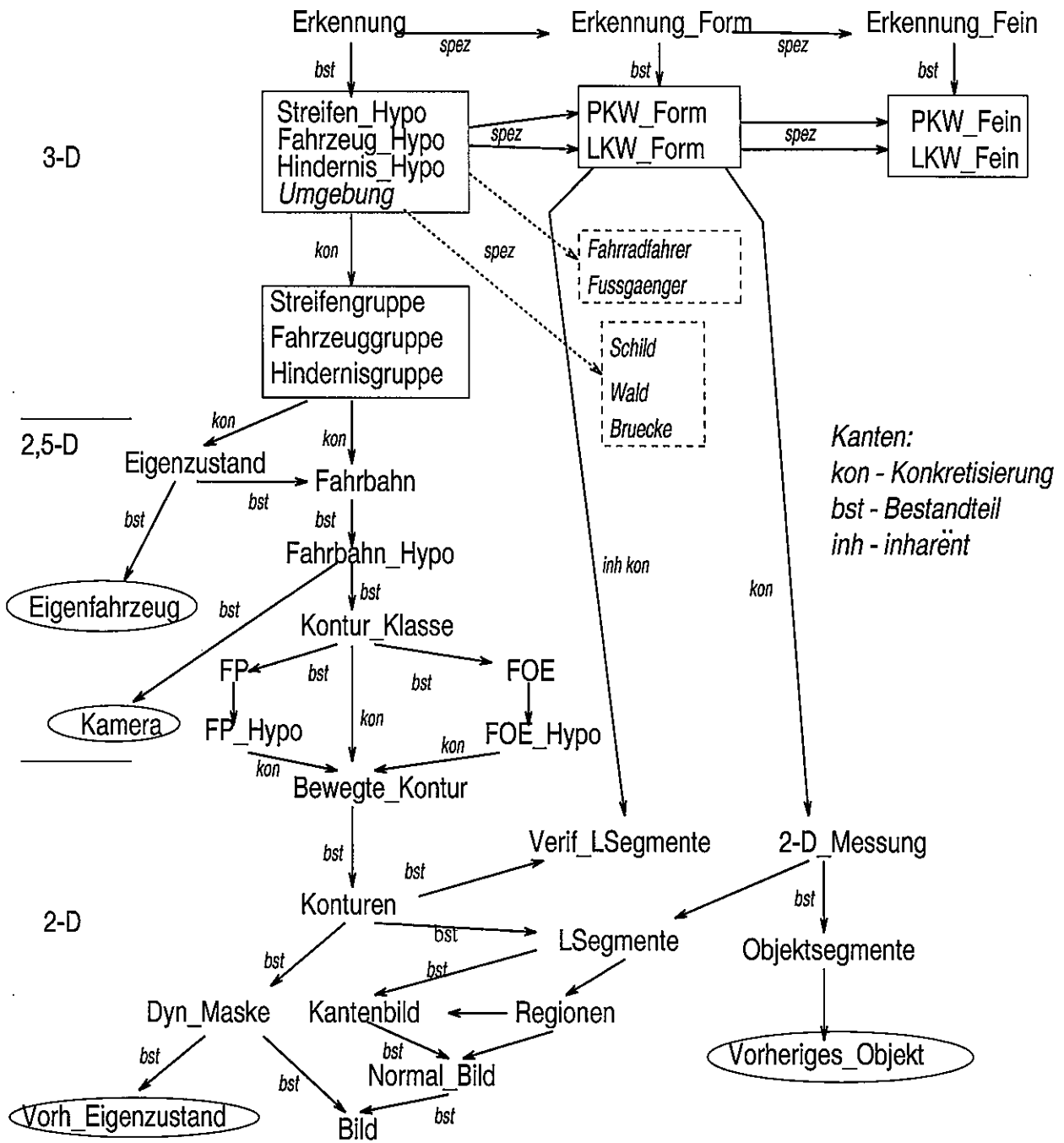


Bild 6.3.1: Das semantische Netzwerk des Modells

6.4 Die schritthaltende Graphensuche

6.4.1 Das Verfahren

Die schritthaltende Kontrolle ist in Bild 6.4.1 dargestellt.

ANW: <i>Parameter_Init</i>	
<i>Suchbaum_Init</i> ;	
WHILE ANW: kein <i>Ende</i>	
WHILE SBKs in OFFEN	
<i>v = Selekt_SBK_aus_OFFEN</i> ;	
IF ANW: <i>Einzelende</i>	
THEN GOTO NEXT	
ANW: <i>S = Ziele(v)</i> ;	
IF <i>S</i> ist nicht leer	
THEN <i>Init_Untersuchraum(S)</i> ;	
ELSE IF $o_i \in \text{DATEN}(v)$ kann aufgefrischt werden	
THEN	<i>Auffrischen(o_i, v)</i> ;
ELSE IF $o_i \in \text{DATEN}(v)$ kann instantiiert werden	
THEN	<i>Instantiieren(o_i, v)</i> ;
ELSE IF $o_i \in \text{DATA}(v)$ kann modifiziert werden	
THEN	<i>Modifizieren(o_i, v)</i> ;
NEXT: <i>Neu_Init()</i>	

Bild 6.4.1: Das Verfahren der schritthaltenden Graphensuche

Dabei wurde die Basiskontrolle um zwei wichtige Schritte erweitert: um das *Init_Untersuchraum*, was eine Auswahl von Untersuchräumen und Initialisierung eines neuen Suchbaumes bedeutet, und das *Auffrischen* – eine wiederholte Instantiierung von früheren Instanzen mit neuen Daten.

6.4.2 Die Suchbaumübernahme und Auffrischung

Es folgt ein Vergleich der beiden Schritte (Übernahme von Suchbaumpfaden und Instanzenauffrischung) wie sie im Projekt MOVIE (Bild 6.4.2) ([Wei92]) oder in dieser Arbeit vorgeschlagen werden (Bild 6.4.3).

In dem vorherigen Verfahren (in MOVIE) wird von einer direkten Zuordnung einer vorherigen Instanz zum neuen Datenelement ausgegangen auf der Basis der Indexgleichheit in der Menge der konkurrierenden Instanzen. Diese Prozedur entspricht einer unmittelbaren Auffrischung von allen vorherigen Instanzen zum Anfang der neuen Analyse. Der Schritt *Auffrischen* führt in MOVIE zur Neuberechnung der ganzen Instanzenmenge eines aufgefrischten Suchbaumknoten in einem Inferenzschritt. Auf diese Weise kann die Struktur der Inferenzen in einem übernommenen Lösungspfad zum späteren Zeitpunkt nicht mehr geändert werden.

Dagegen wird in dieser Arbeit das Auffrischen als ein neuer Inferenzoperator behandelt der homogen ist mit der Instantiierung oder Modifizierung.

Es werden nicht unmittelbar die besten Suchbaumknoten sondern ihre Vorgängerknoten übernommen, in denen die Analyse für das aktuelle Zielkonzept begonnen hat. Dies neuen SBKs erhalten aber das lokale Inferenznetz ihres ausgewählten Blattes. Auf diese Weise wird das

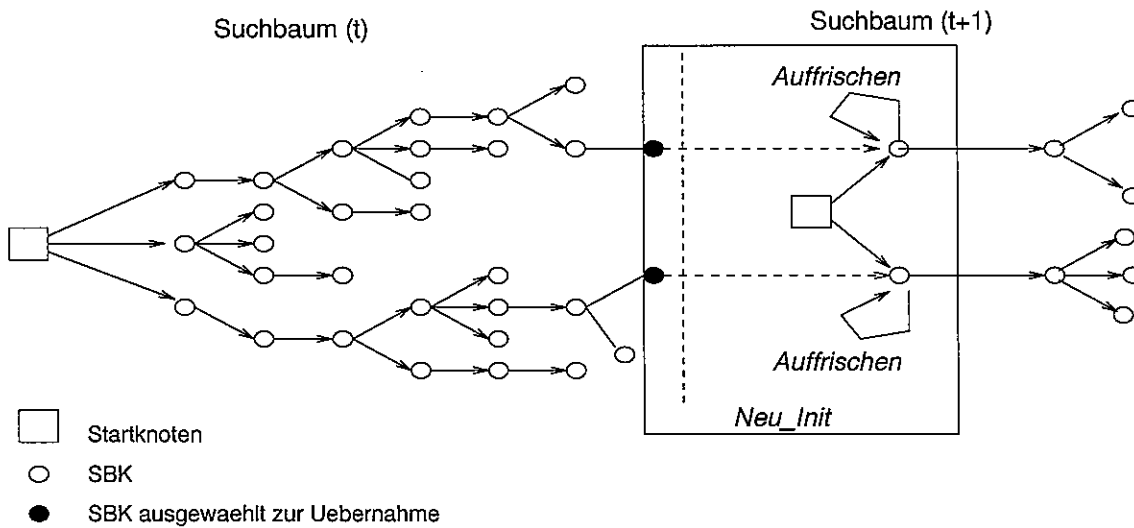


Bild 6.4.2: Übernahme von SBKs mit gleichzeitigem *Auffrischen*

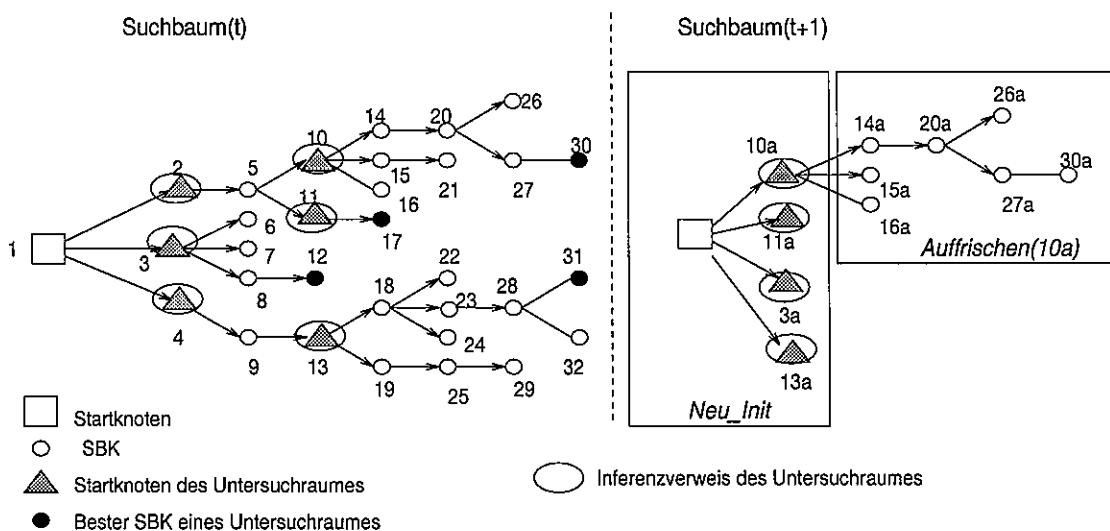


Bild 6.4.3: Übernahme von Untersuchräumen mit bester lokaler Inferenzmenge und späteres *Auffrischen* (Wiederaufbau) der besten Suchbaumpfade (in dieser Arbeit)

Analyseergebnis zwar übernommen, aber der Pfad muß wiederaufgebaut werden. Dieser Wiederaufbau findet statt, falls der Knoten zur Auffrischung ausgewählt wurde. Im Unterschied zur vorherigen Kontrolle entstehen neben dem früheren optimalen Pfad jetzt auch alternative SBKs. Diese Alternativen enthalten aber zum Teil die (vermutlich) lange verfolgten Instanzen des aufgefrischten Pfades. Wenn später der bisherige Pfad verlassen wird, bedeutet das nicht den wiederholten Anfang der Analyse. Dadurch trägt die schritthaltende Kontrolle zur Stabilität der Analyse bei.

6.4.3 Das Bewertungsschema im Suchbaum

Falls eine *monotone* Bewertungsfunktion definiert wird, läßt sich mit dem *A**-Verfahren zur Baumsuche immer ein optimaler Pfad finden (falls er existiert). Solch eine Bewertungsfunktion enthält üblicherweise zwei additive Komponenten: die bisherigen Kosten der Generierung des Suchbaumpfades und eine optimistische Schätzung der Restkosten bis zum Erreichen einer

Lösung.

Beispiel 6.7: In [Sag90] werden zwei verschiedene Bewertungsparadigmen voneinander getrennt – statistische Maße und heuristische Maße. Als einzelne Bewertungsmaße werden betrachtet:

- Die Zulässigkeit $Z(H)$ einer Hypothese in Bezug auf die sprachliche und inhaltliche Kompetenz des Systems.
- Die akustische Qualität $q(H)$ einer Hypothese.
- Die Sicherheit $s(H)$ einer Hypothese.
- Die pragmatische Priorität $pP(h)$ einer Hypothese.
- Die semantische Priorität $pS(A_i^M)$ und $pS(A_i^I)$ für spezielle modifizierte Konzepte bzw. Instanzen.

Bis auf die semantische Priorität genügen all diese Maße zumindest näherungsweise der Forderung des A^* -Algorithmus nach optimistischer Schätzung. Jede Bewertung $f(v_i)$ eines Suchraumknotens v_i ist durch die Bewertung $\phi(\text{ZIEL}(v_i))$ gegeben:

$$\phi(O_i) = (Z(O_i), q(O_i), s(O_i), pP(O_i), sP(O_i)) \quad (6.4.1)$$

Der Vergleich zwischen zwei Bewertungsvektoren erfolgt gemäß ihrer lexikographischen Ordnung.

Beispiel 6.8: In [NBSS90] wird zwischen der *Kompatibilität* und der *Sicherheit* einer Zuordnung unterschieden. Die Kompatibilität gibt die Qualität der Korrespondenz wieder. Die Sicherheit basiert auf dem Bildsegment, z.B. auf der Linienstärke eines Liniensegments. In erster Komponente f_1 des Bewertungsvektors wird die Kompatibilität und in zweiter f_2 die Sicherheit abgespeichert. Beide Maße basieren auf den Attributwerten von Objekten in Bezug auf die Attribute eines Modellobjektes. Die dritte Komponente f_3 des Bewertungsvektors beinhaltet das Ergebnis des Relationtests. Sie gibt wieder, wie gut die referierten Attribute die Strukturrelation eines Konzepts erfüllen. Die Kombination der Bewertungen f_1 , f_2 und f_3 erfolgt durch die Berechnung des Minimum von allen Werten.

Die ganzzahligen Komponenten i_i des Bewertungsvektors werden benutzt um den strukturellen Unterschied zwischen dem Modellobjekt und den Segmentierungsdaten darzustellen. In der Komponente i_1 wird die gesamte Anzahl der Bestandteile und Konkretisierungen der Instanz gespeichert. Komponente i_2 beinhaltet die Anzahl der Instanzen für Bestandteile und Konkretisierungen, die in den Segmentierungsdaten enthalten sind. i_3 ist die Anzahl der Default Instanzen.

Ein Pfad S_i ist besser als Pfad S_j , wenn die Kompatibilität f_1 von S_i größer ist als die von S_j , wenn die Anzahl der zugeordneten Teile i_2 von S_i größer ist als die von S_j , und wenn die Anzahl der substituierten Instanzen i_3 von S_i kleiner ist als die von S_j .

Während einer dynamischen Analyse sind nicht nur die Qualität der generierten Hypothese, sondern auch der Zeitaufwand für ihre Erstellung von Bedeutung. Beide Kriterien müssen miteinander abgewogen werden. Dann haben wir die Möglichkeit, eine Hypothese mit einer schlechteren Qualität einer anderen vorzuziehen, die aber einen viel größeren zu erwartenden Zeitaufwand besitzt als die von uns gewählte. Damit werden wir dem Wunsch nach zeitlich

definiertem Verhalten der Analyse bei gleichzeitiger Wahrung der Robustheit von Analyseergebnissen gerecht. Für jeden Suchbaumknoten O definieren wir folgende Struktur des Bewertungsvektors $\psi(O)$:

$$\psi = [f, g, R, g_R, 1 - S] \quad (6.4.2)$$

wobei die Komponenten folgende Bedeutung besitzen:

- f – die additive *Gesamtkostenfunktion* des Suchbaumknoten:

$$f = g + h = R + A = \max(g_R, h_R) + (g_A + h_A)$$

- g – die bisherigen *Kosten* des Suchbaumknoten:

$$g = g_R + g_A$$

- R – die *Risikokosten* der globalen Zielinstanz (sie ist dual zur *Qualität* Q) :

$$R = 1 - Q = \max(g_R, h_R)$$

- g_R – die bisherigen *Risikokosten* des Suchbaumknoten

- S – die zu erwartende *Sicherheit* der Instanz des aktuellen Zielkonzepts

Der Term A bedeutet den zeitlichen *Aufwand* der Analyse und er besteht aus dem bisherigen Aufwand und dem zu erwartenden Restaufwand bis zum Erreichen einer globalen Zielinstanz: $A = g_A + h_A$.

Die *Qualität* Q ist eine Funktion der Sicherheit der Daten im Suchbaumknoten O und der optimistischen Schätzung der zu erwartenden Sicherheit der Interpretation von restlichen minimalen Instanzen (genauer gesagt einem ausreichend großen Teil von ihnen) :

$$Q = c * \min[S(O), S(\text{RestDaten}(O))] = c * \min[Q_g, Q_h] \quad (6.4.3)$$

$$g_R = 1 - Q_g; \quad h_R = 1 - Q_h \quad (6.4.4)$$

wobei gilt $0 \leq S \leq 1$ und c ist eine Konstante, die Aufwandkosten und Risikokosten gewichtet.

$$S(\text{RestDaten}(O)) = \max_{i \in \text{SEGM-Daten}(O)} [S(i)] \quad (6.4.5)$$

SEGM bezeichnet alle minimalen Instanzen einer Einzelanalyse und *Daten*(O) den Teil aus *SEGM*, der im Knoten O bereits interpretiert ist. $S(i)$ ist die Sicherheit der minimalen Instanz i . Die *Sicherheit* S eines Analyseobjektes O (Instanz, modifiziertes Konzept) wird wie folgt ermittelt:

$$S(O) = \left[\min_{i \in \{Kon(O) \cup Bst(O)\}} S(i) \right] * S[(Attr \cup Rel)(O)] \quad (6.4.6)$$

d.h. die Sicherheit einer Instanz kann nicht größer sein, als die Sicherheit seiner Teile und die Sicherheit, mit der seine Attribute und Relationen durch vorhandene Daten erfüllt sind.