

Anhang B

Beispiel und Benutzerschnittstelle des Systems

Um die Implementierung des vorgestellten Systems zur Verkehrszeichenerkennung nicht völlig unberücksichtigt zu lassen, soll an dieser Stelle kurz auf die Implementierung eingegangen werden. Dabei werden diejenigen Komponenten des Systems angesprochen, die in der vorliegenden Arbeit bisher nicht erwähnt wurden. Außerdem soll der Ablauf einer Klassifikation anhand eines Beispiels verdeutlicht werden.

B.1 Die Benutzerschnittstelle

Das System wurde in der Programmiersprache LISP unter dem Betriebssystem Genera implementiert. Die graphische Benutzeroberfläche besteht aus zwei Ansichten, der Klassifikationsansicht und der Netzwerkansicht.

B.1.1 Die Klassifikationsansicht

Abbildung B.1 zeigt die Klassifikationsansicht. Diese besteht aus vier Fenstern. In den drei oberen Fenstern werden die aktuelle Szene, die segmentierte Szene und das Klassifikationsergebnis angezeigt. Im unteren Fenster werden die Aktionen des Benutzers angezeigt, Eingaben angefordert oder Kontrollansichten eingeblendet. Mit Hilfe der oberen Menueleiste können die einzelnen Klassifikationsschritte angestoßen werden. Nach der Klassifikation einer Verkehrsszene kann das Ergebnis durch Auswahl der *inspect* Option genauer betrachtet werden. Nach Wahl des Segments wird

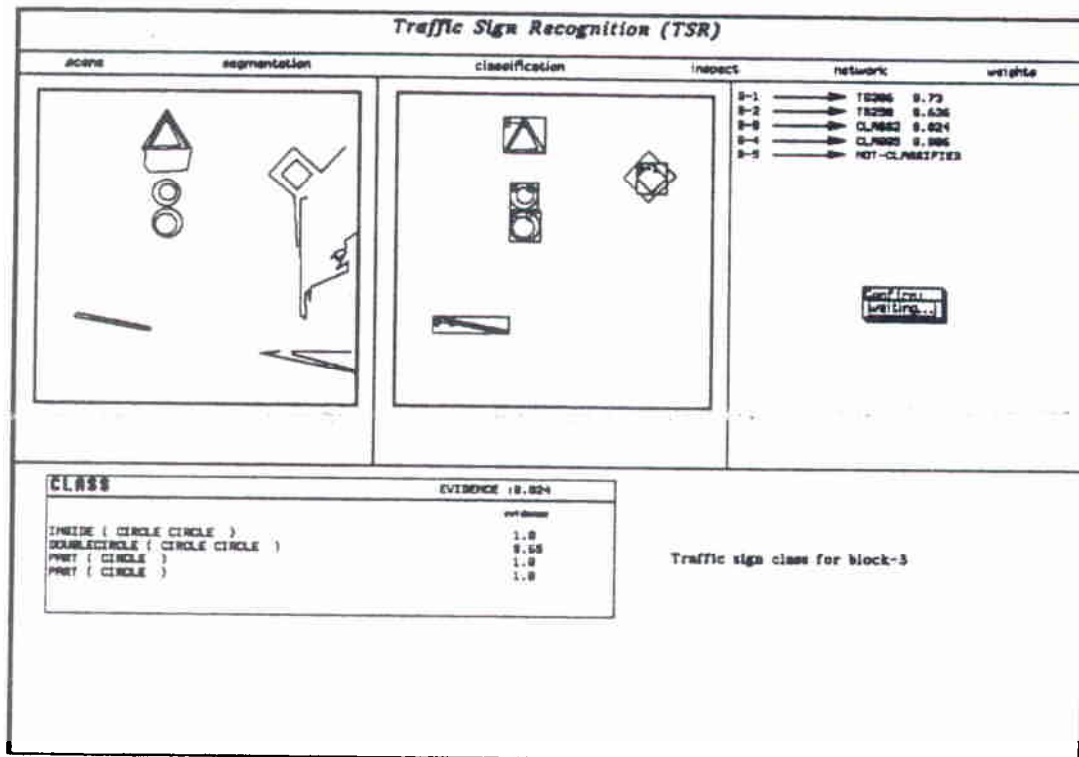


Abbildung B.1: Die Klassifikationsoberfläche der Benutzerschnittstelle

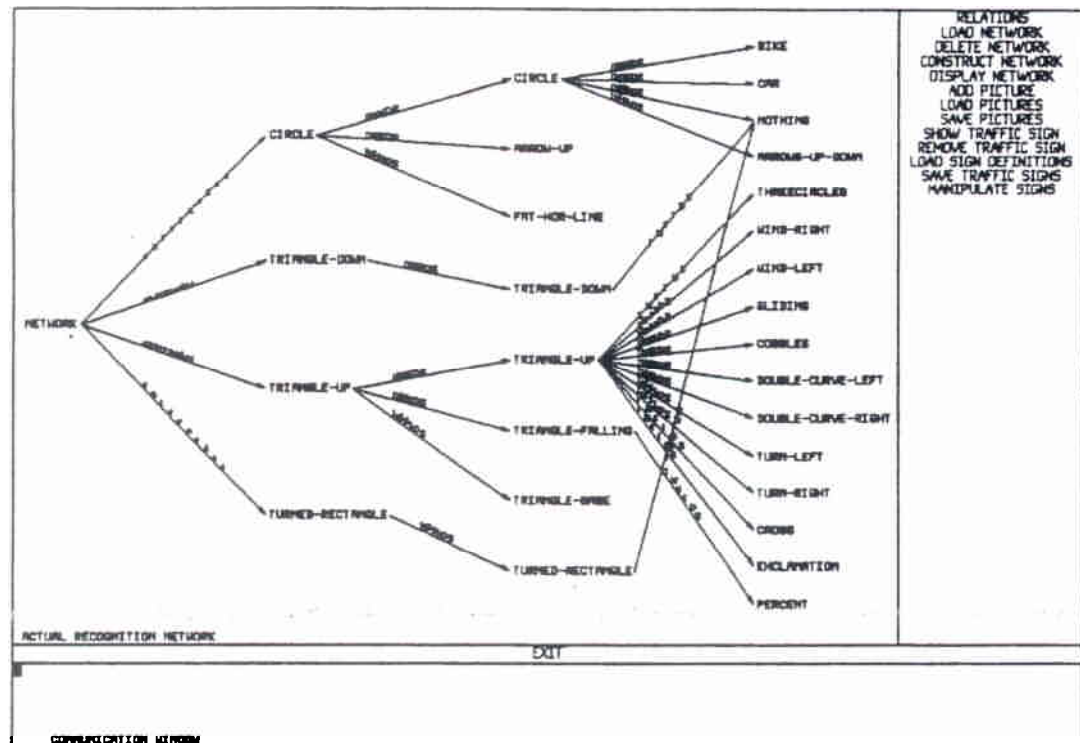


Abbildung B.2: Die Netzwerkansicht der Benutzerschnittstelle

für dieses Segment das Verkehrszeichen oder die Klasse mit allen Komponenten, Gewichten und aktuellen Ähnlichkeitswerten angezeigt. Durch die Option *weights* kann eine Heuristik zur Generierung von Gewichten angestoßen werden, wie sie in 3.4.2 definiert wurde. Eine weitere Option ermöglicht die Einstellung der Schwellwerte für die Klassifikation. Durch die Option *network* wird zu der Netzwerkansicht gewechselt.

B.1.2 Die Netzwerkansicht

Im Hauptfenster der Netzwerkansicht wird das Verkehrszeichennetz angezeigt. Durch die Menuleiste am rechten Rand können Operationen zur Einstellung der Anzeige, zur Erzeugung und Manipulation, sowie zum Laden und Speichern eines Netzwerks aufgerufen werden.

B.2 Die Lagerrelationen

Bisher wurde nur die Lagerrelation *double.circle* genauer beschrieben. Hier sollen jetzt die übrigen Lagerrelationen, die im Netzwerk implementiert sind, kurz dargestellt werden.

double.up.triangle Beschreibt die Lagerrelation zwischen zwei Dreiecken, deren Spitze nach oben zeigt. Die Berechnung wurde sehr einfach gehalten. Die umschreibenden Rechtecke der beiden Dreiecke werden verglichen und deren Abstände voneinander an allen vier Kanten gemessen. Für die Kantenabstände sind Richtwerte festgelegt worden, mit denen die Werte der Kandidaten verglichen werden.

double.down.triangle Beschreibt die Lagerrelation zwischen zwei Dreiecken, deren Spitze nach unten zeigt. Die Berechnung erfolgt ebenfalls basierend auf den umschreibenden Rechtecken. Es werden lediglich andere Richtwerte festgelegt.

double.turned.rectangle Beschreibt die Lagerrelation zwischen zwei um 45 Grad gedrehten Quadraten, wie sie bei der Beschreibung eines Vorfahrtsschildes auftreten. Auch hier erfolgt die Berechnung mit Hilfe der umschreibenden Rechtecke.

Die Lagerrelationen sind zwar sehr grob beschrieben, aber wenn man sich die Darstellung einer Szene durch Formprimitive betrachtet, so fällt auf, daß die Konturen stark vereinfacht und dadurch gegenüber der Originalszene stark verzerrt sind. Bei einer exakten Messung müßte eine hohe Toleranzgrenze gesetzt werden, um diese Verzerrungen zu akzeptieren.

B.3 Ergebnisse der Klassifikation

Anhand einer Verkehrsszene sollen ausgehend von einem Grauwertbild die einzelnen Stadien angezeigt werden, die ein Bild durchläuft.

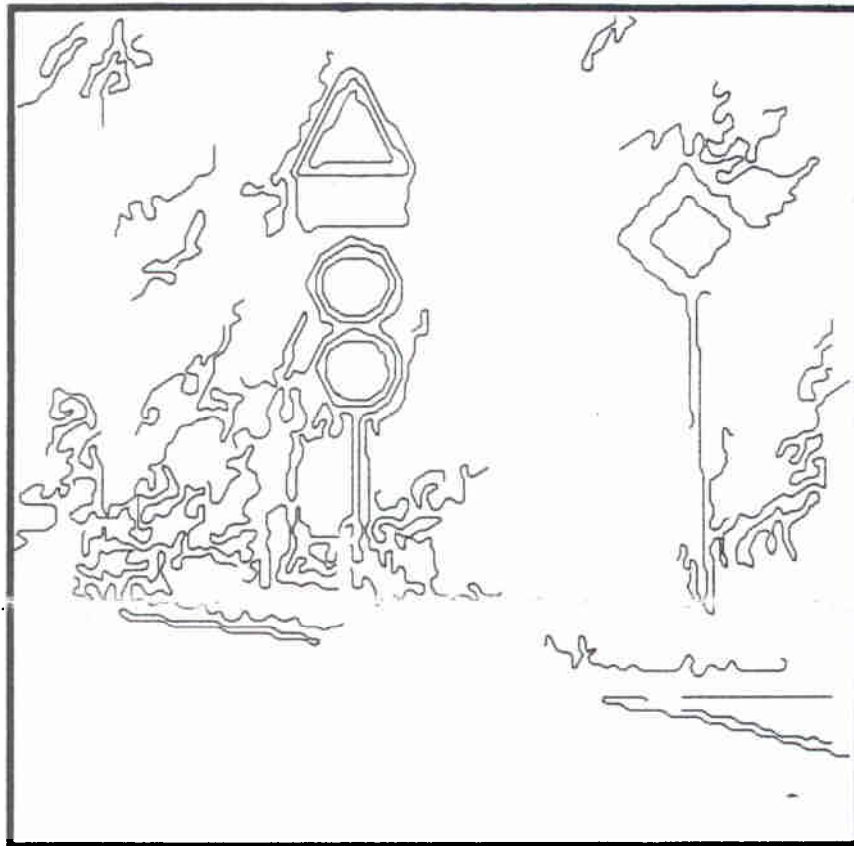


Abbildung B.3: Das Grauwertbild einer Verkehrsszene



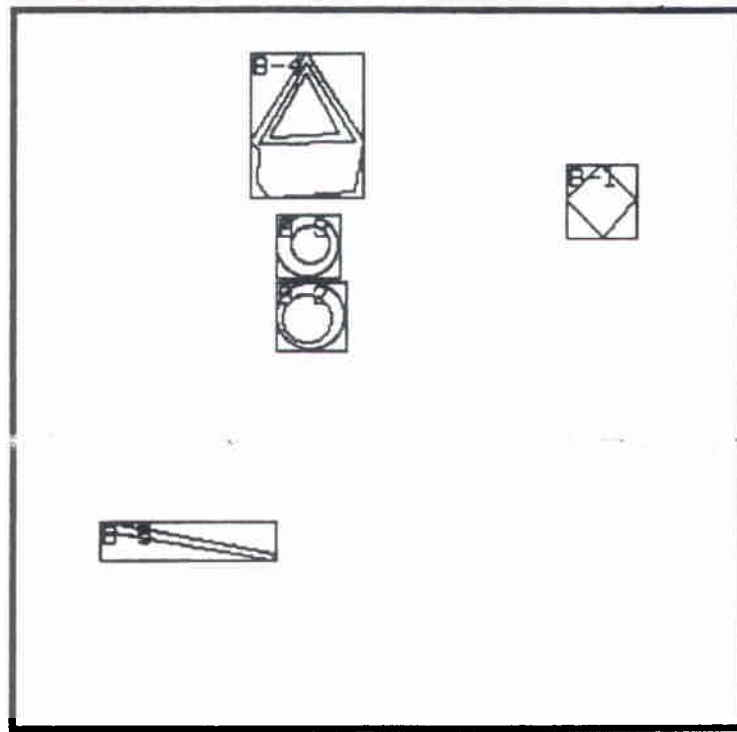
Als zweites wird ein Bild gezeigt, welches sich durch eine Kantenextraktion aus dem Grauwertbild ergibt (Abbildung B.3). Betrachtet man das Bild, so wird man zunächst selbst Probleme haben, in diesem Bild Verkehrszeichen zu erkennen. Bei näherem Hinsehen und durch die Kenntnis der möglichen Verkehrszeichen, kann man diese dann doch lokalisieren und erkennen.

Abbildung B.4: Ein durch Kantenextraktion gewonnenes Bild



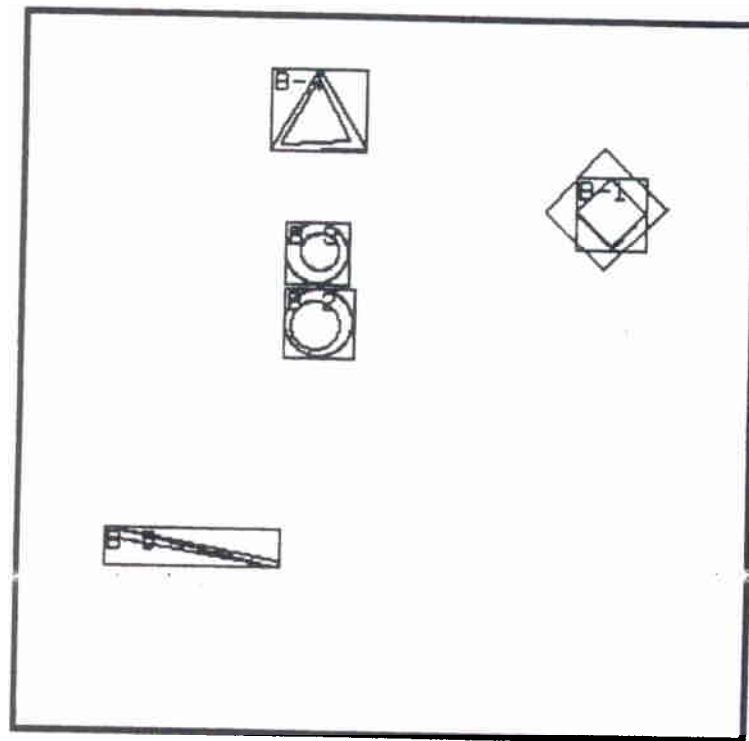
Die in diesem Bild dargestellten Kanten stellen die HSC-Objekte dar, deren Wurzel auf einer hohen Ebene liegt, so daß die Objekte zur Weiterverarbeitung ausgewählt werden. Dabei fallen die Konturen von den Piktogrammen bereits alle heraus, da sie zu klein sind und somit als Rauschen bzw. Hintergrund betrachtet werden. Aus diesen Daten sind bereits nur noch die Verkehrszeichenklassen zu erkennen, dafür ist ein Großteil der störenden Linien des Hintergrundes verschwunden.

Abbildung B.5: HSC Vorfilterung



Das Bild zeigt die Konturen, die in Segmente zusammengefaßt werden konnten, die anderen werden nicht angezeigt. Jedes der angezeigten Segmente wird an die Klassifikationskomponente weitergegeben.

Abbildung B.7: Segmentierung des Bildes



B-1	→	TS306	0.73
B-2	→	TS250	0.636
B-3	→	CLASS2	0.824
B-4	→	CLASS5	0.806
B-5	→	NOT-CLASSIFIED	

Das letzte Bild zeigt die Segmente, die von der Klassifikationskomponente untersucht wurden, sowie die Rekonstruktionen, die durchgeführt wurden.

Abbildung B.8: Klassifikation mit rekonstruierten Konturen

Abbildungsverzeichnis

2.1	Gewichtung der umgebenden Pixel zur Glättung	13
2.2	Differenzierung einer Intensitätsfunktion	14
2.3	Laplace Filter	15
2.4	Ein mexikanischer Hut als Filter	16
2.5	Iterative Bestimmung der Eckpunkte	18
2.6	Beispiel einer chain code-Beschreibung	18
2.7	Die Repräsentationsform <i>quad tree</i>	19
3.1	Konzeption des Bilderkennungssystems	22
3.2	Schema des Klassifikationsnetzwerks	24
3.3	Semantischer Hierarchiebaum der Verkehrszeichen	27
3.4	Aufbau des Zeichens <i>Geschwindigkeitsbeschränkung</i>	28
3.5	Strukturierung der Menge der Verkehrszeichen	29
3.6	Inselbildung des HSC	31
3.7	Wurzeln des HSC auf verschiedenen Ebenen	32
3.8	Die Menge der definierten Formprimitivtypen	34
3.9	Repräsentation des Verkehrszeichens 101	40
3.10	Der Aufbau eines Verkehrszeichennetzwerkes	42
3.11	Beispiel für die Winkel/Längen-Konturbeschreibung	44
3.12	Die Lagerrelation <i>double.circle</i>	48
3.13	Übersetzung einer Verkehrszeichendefinition	51
3.14	Einfügen eines neuen Verkehrszeichens	54
3.15	Komplexität des geometrischen in Prädikats	58
3.16	Beispiel für die Segmentierung	60
3.17	Abbildung eines Segmentbaumes auf das Netzwerk	62
3.18	Algorithmus für den ersten Abbildungsschritt	64
3.19	Algorithmus zur Umrißklassifikation	67
3.20	Konflikt durch ähnliche Verkehrszeichen	70
3.21	Bestimmung der Gewichte der Verkehrszeichenknoten	72

3.22	Algorithmus zur Klassifikation eines Segments	73
3.23	Algorithmus der Verkehrszeichenklassifikation	74
3.24	Mögliche Winkel/Längen-Beschreibungen einer Kontur	76
3.25	Vergleich von Formprimitiv und Konturmuster	78
3.26	Konvertierung von Formprimitiven	80
3.27	Veränderte Konturen	81
3.28	Fehlerhafte Rekonstruktion	82
3.29	Methoden zur Rekonstruktion	84
3.30	Toleranzeinstellung durch die Ähnlichkeitskurve	89
4.1	Wiedererkennung eines Objekts	94
4.2	Schema einer Objektverfolgungskomponente	95
4.3	Integration der Objektverfolgungskomponente	97
A.1	Architektur des Systems	100
A.2	Beispiel einer Regel des KEE-Systems	103
B.1	Die Klassifikationsoberfläche der Benutzerschnittstelle	108
B.2	Die Netzwerkansicht der Benutzerschnittstelle	109
B.3	Das Grauwertbild einer Verkehrsszene	111
B.4	Ein durch Kantenextraktion gewonnenes Bild	112
B.5	HSC Vorfilterung	113
B.6	Graphische Darstellung der Formprimitve	114
B.7	Segmentierung des Bildes	115
B.8	Klassifikation mit rekonstruierten Konturen	116

Index

CL <i>component link</i>	Verbindungskante zwischen Verkehrszeichenknoten und den Komponenten des Verkehrszeichens	37
CN <i>contour node</i>	Menge der Netzwerkknoten, die Konturinformation tragen	37
E <i>entrynode</i>	Menge der Eintrittskanten	37
enthält	Prädikat, welches bestimmt, ob zwei Formprimitive ineinanderliegen	57
Formprimitive	Menge von symbolischen Objekten, die Konturen von Objekten aus einem Bild beschreiben.	34
Initialgewichtung	Setzen der Kantengewichte	72
Konflikt	Ein Konflikt tritt auf, wenn einem Segment zwei Verkehrszeichen zugeordnet werden.	69
Konturerkennungsfunktion	Eine Funktion, die eine Ähnlichkeit zwischen einem Formprimitiv und einer Konturbeschreibung mißt	78
L <i>links</i>	Menge der Netzwerkkanten	38
l(x)	Liste von Kantenlängen einer Winkel/Längen-Konturbeschreibung	42
N <i>nodes</i>	Menge der Netzwerkknoten	38
PN <i>picture node</i>	Menge der Netzwerkknoten, die Piktogramme repräsentieren	37

Regelsprache	Syntaxfestlegung für die Eingabe von Verkehrszeichen	50
Rekonstruktions- funktion	Funktion zur Erzeugung eines vervollständigten Formprimitivs aus einem Formprimitiv und einer Kontur- beschreibung.	85
Relationenfunktion	Zweistellige Funktion über Formprimitive, die eine Lagerrelation zwischen diesen bestimmt.	44
RL <i>relation links</i>	Menge der Kanten, die Lagerrelationen repräsentieren	37
SN <i>sign node</i>	Menge der Knoten, die Piktogramme repräsentieren	37
Strukturnetz	Subnetz des Verkehrszeichennetzes, welches nur aus CN- und PN-Knoten, sowie RL-Kanten besteht.	38
Umschreibendes Rechteck	Rechteck, dessen Ecken sich aus den Minima und Maxima der x und y Koordinaten eines Objekts bestimmen	57
Verkehrszeichennetzwerk	Definition der Wissensbasis	35
Verkehrszeichensubnetz	Subnetz des Verkehrszeichennetzes, welches nur Information über ein einziges Verkehrszeichen speichert	39
VZ	Menge der Verkehrszeichen (-kodes)	36
Winkel/Längen- Konturbeschreibung	Konturbeschreibung bestehend aus einer Liste von Winkeln und einer Liste von Kantenlängen	42
$\alpha(x)$	Liste der Winkel einer Winkel/Längen - Konturbeschreibung	42
Γ	Menge der Formprimitive	33
Γ_x	Menge der Formprimitive vom Typ x	33
Δ	Menge der Lagerrelationen	36
Θ	Menge der Konturbeschreibungen	36
Ω	Menge der Piktogramme	36

Literaturverzeichnis

- [Au91] E. Auld, Recognition of Traffic Signs using the KEE System, Report for Summer 1991 at Daimler-Benz, 1991
- [Bar87] N. Bartneck, Ein Verfahren zur Umwandlung der ikonischen Bildinformation digitalisierter Bilder in Datenstrukturen zur Bildauswertung, Dissertation 1987
- [BB82] D. H. Ballard, C. M. Brown, Computer Vision, Prentice Hall 1982
- [BEE85] B. Borge, A. Ebert, D. Ernst, Symbolische Bildfolgenbeschreibung zur Objekterkennung, in Informatik Fachberichte Mustererkennung 1985
- [BF81] A. Barr, E. A. Feigenbaum, The Handbook of Artificial Intelligence, Volume 1, Pitman 1981
- [CD85] E. Charniak, D. McDermot, Introduction to Artificial Intelligence, Addison-Wesley 1985
- [CF82] P. R. Cohen, E. A. Feigenbaum, The Handbook of Artificial Intelligence, Volume 3, Pitman 1982
- [Cha82] C.F. Chabris, Artificial Intelligence and Turbo C, Dow Jones - Irwin, 1982
- [DH86] S. Drüe, G. Hartmann, Modellgestützte Erkennung hierarchisch codierter Objekte, in Mustererkennung 1986, 8.DAGM-Symposium 1986
- [Fe90] W. Feith, Bilderfänger - Funktionsweise und Entwurf einer Frame-Grabber-Karte für PCs, in ct - magazin für computer und technik 3/90, Heise Verlag
- [Fe92a] W. Feith, Ladungskübel und Eimerketten - Aufbau und Funktionsprinzip von CCD-Bildsensoren, in ct - magazin für computer und technik 1/92, Heise Verlag

- [Fe92b] W. Feith, Immer im Bilde - Einfache CCD-Kamera mit Rechner-Schnittstelle, in *ct - magazin für computer und technik* 1/92, Heise Verlag
- [Fra89] U. Franke, Regionenorientierte Bildbeschreibung - Algorithmen und Möglichkeiten, Fortschritt-Berichte VDI Reihe 10: Informatik/Kommunikationstechnik 1989
- [GR90] Reinhard E. Gämlich, Werner Ritter, A Knowledge Based System for Traffic Sign Recognition, Daimler-Benz AG, in *Informatik Fachberichte Mustererkennung* 1990
- [Har83] G. Hartmann, Erzeugung und Verarbeitung hierarchisch codierter Konturinformation, VDE-Fachberichte 35, VDE-Verlag (1983)
- [HAV87] Hinweise für das Anbringen von Verkehrszeichen und Verkehrseinrichtungen, Kirchbaum, 7. Auflage 1987
- [KZ85] K.-D. Kuhnert, A. Zapp, Wissensgesteuerte Bildfolgeauswertung zur automatischen Führung von Straßenfahrzeugen in Echtzeit, in *Informatik Fachberichte Mustererkennung* 1985
- [La90] T. Labrousse, Classification d'objets symboliques dans un environnement LISP, Diplomarbeit, Juni 1990
- [MBH90] B. Mertsching, I. Böhmer, G. Hartmann, Kontrollalgorithmen für auf hierarchischen Datenstrukturen arbeitende wissensbasierte Bildanalysesysteme, in *Mustererkennung 1990, 12.DAGM-Symposium 1990*
- [MH88] B. Mertsching, G. Hartmann, Modulare Modellierung von hierarchisch-strukturcodierten Objekten und Szenen durch ein semantisches Netzwerk, in *Mustererkennung 1988, 10.DAGM-Symposium 1988*
- [Mo87] R. Mohr, A General Purpose Line Drawing Analysis System, in *Syntactic and Structural Pattern Recognition Theory and Applications*, H. Bunke, Sanfeliu
- [Mö89] O. Möller, Neuronale Netze in der Bildverarbeitung, in *Autbild '89*
- [MOT91] Mit besten Empfehlungen, *mot-Technik* 2/1991
- [Pog89] T. Poggio, Wie Computer und Menschen sehen, in *Spektrum der Wissenschaft, Computer-Systeme* 1989
- [Ric89] M. M. Richter, Prinzipien der Künstlichen Intelligenz, Teubner, 1989

- [San90] A. Sanfeliu, Matching Tree Structures, in Syntactic and Structural Pattern Recognition Theory and Applications, H.Bunke,A.Sanfeliu, 1990
- [Som89] G.Sommer, Wissensverarbeitung in der Bildanalyse, in Autbild '89
- [St84] G. Steele, Common LISP - The Language, Digital Press 1984
- [SZ86] C.-K. Sung, G. Zimmerman, Detektion und Verfolgung mehrerer Objekte in Bildfolgen, in Mustererkennung 1986, 8.DAGM-Symposium 1986
- [Val91] J.J.Valdes, Differential Treatment of Premisses in Production Rules by Evidence Ranking Vectors, Universität Kaiserslautern 1991
- [We91] S.Weß, PATDEX/2 - ein System zum adaptiven, fallfokussierten Lernen in technischen Diagnosesituationen, SEKI Working Paper 91-01 (SFB), Universität Kaiserslautern