

Automatische Merkmalsynthese

**Ein Lern- und Klassifikationssystem zur Erkennung komplexer und
deformierter, natürlicher und künstlicher Objekte**

Vom Fachbereich Elektrotechnik und Informatik
der Universität-Gesamthochschule Siegen
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

(Dr.-Ing.)

genehmigte Dissertation

VON

Diplom-Ingenieur Dag Pechtel

1. Gutachter: Prof. Dr. Klaus-Dieter Kuhnert
2. Gutachter: Prof. Dr. Bernd Freisleben

urn:nbn:de:hbz:467-20

Vorwort

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter in der Fachgruppe Prozessdatenverarbeitung im Fachbereich Elektrotechnik und Informatik der Universität-Gesamthochschule Siegen.

Während dieser Zeit durfte ich die weite Welt und die Umgebung von Siegen kennenlernen. Das hat mich toleranter gemacht.

Das Forschen im Bereich der Mustererkennung war genau das Richtige für mich. Es war mir ein Vergnügen.

Ferner habe ich im Dienst mein Faible für die Lehre entdeckt und mich neben meiner rein fachlichen Entwicklung vor allen Dingen auf diesem Gebiet weitergebildet. Das hat mir Spass gemacht.

Mein Dank gilt meinem beruflichen und sozialen Umfeld. Insbesondere sind zu nennen: Prof. Dr.-Ing. K.-D. Kuhnert (Betreuer und 1. Gutachter), Prof. Dr.-Ing. B. Freisleben (2. Gutachter), Prof. Dr.-Ing. H. Bessai (Vorsitzender der Promotionskommission), Prof. Dr.-Ing. O. Loffeld (Leitung des Promotionsverfahrens), Dr.-Ing. Ralf Schaufler (Kollege und Mensch zum Reden nicht nur über Fachliches), Sascha Philipp (studentische Hilfskraft ohne die die experimentelle Umgebung für diese Doktorarbeit nie so schön geworden wäre) und Marlies Wendhut (, die einfach so mit mir nach Siegen gekommen ist).

Bremen, im Juni 2002

Dag Pechtel

*Problems worthy of attack
prove their worth by hitting back.*

*Ein Problem ist einen Angriff wert
wenn es sich dagegen wehrt.*

Piet Hein (1905-1996, dänischer Wissenschaftler und Poet)

Kurzfassung

Diese Arbeit beschäftigt sich mit der Erkennung isolierter visueller Objekte. Diese Objekte entstammen der realen Welt und können komplex und deformiert sein. Die Außenkonturen, der auf eine Ebene projizierten 3D-Objekte jeweils einer bestimmten Klasse, werden hinsichtlich signifikanter lokaler Konturabschnitte untersucht, die ausreichen, um die Konturen dieser Klasse von den Konturen jeweils aller anderen Klassen zu unterscheiden.

Es ist zu betonen, dass in dieser Arbeit das bisher stiefmütterlich behandelte Thema der *automatischen Merkmalsgenerierung* im Vordergrund steht. Die *Merkmalsgenerierung* darf nicht mit der Merkmalsselektion verwechselt werden, in der es um die Auswahl einer möglichst optimalen Untermenge aus einer gegebenen Menge von Merkmalen handelt. Sie darf aber auch nicht mit der **unsystematischen** traditionellen Merkmalsextraktion verwechselt werden, da diese das Ziel verfolgt, vordefinierte Merkmale wie z.B. das Längen-Breiten-Verhältnis eines Objektes im Bild aus dem Bild zu extrahieren. Bei der *automatischen Merkmalsgenerierung* werden abstrakte Merkmale, wie z.B. Flaschenkappen oder Fischflossen, **systematisch** erzeugt, indem gleichartige Basismerkmale, wie z.B. aufeinanderfolgende Konturpunkte, zu komplexeren neuen Merkmalen synthetisiert werden.

Das vorgestellte System besteht dabei aus 2 Hauptmodulen: einem komplexen aus mehreren Einzelmodulen bestehenden autonomen Lernsystem und einem hierarchischen Klassifikator. Das Hauptaugenmerk liegt hier also nicht auf der Konstruktion eines Klassifikators, sondern in einer neuartigen systematischen Merkmalsgenerierung basierend auf einer *automatischen Merkmalsynthese* (siehe Abbildung 1).

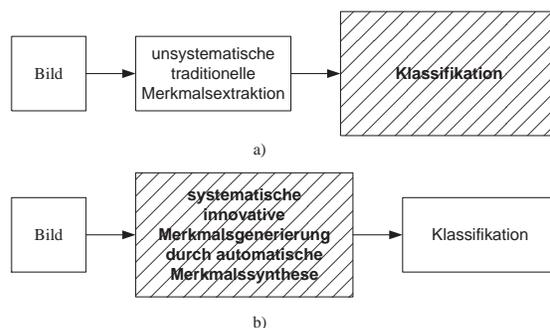


Abbildung 1: Prinzipieller Unterschied zu traditionellen Methoden. a) Traditionelle Erkennungsverfahren extrahieren Merkmale aus Bildern unsystematisch. Das Hauptaugenmerk solcher Verfahren liegt in der Entwicklung von Klassifikatoren, die mit diesen unsystematisch gefundenen Merkmalen möglichst gute Erkennungsraten liefern. b) Das neu entwickelte Verfahren legt seinen Schwerpunkt auf eine systematische Generierung von aussagekräftigen Merkmalen und nicht auf die Entwicklung von Klassifikatoren.

Die Eingangsdaten des Lernsystems sind die Außenkonturen von Objekten aus verschiedenen zu unterscheidenden Klassen. Diese Konturen werden lokal auf signifikante Konturabschnitte untersucht, die die Klassen unterscheiden. Damit besteht die Wissensbasis eines schnellen hierarchischen Klassifikators aus komprimierten Konturen, bei denen die zur Unterscheidung unwichtigen Konturabschnitte weggelassen werden.

Im Zentrum dieses Verfahrens steht die sogenannte *automatische Merkmalsynthese*. Als *Basismerkmale* werden Konturpunkte zunächst zu *wichtigen Konturabschnitten* synthetisiert, die jeweils genau 2 Konturen unterscheiden. Die *wichtigen Konturabschnitte* einer Kontur, die aus dem Vergleich mit allen Konturen aus anderen Klassen entstanden sind, werden anschließend zu *signifikanten Konturabschnitten* synthetisiert, die diese Kontur von den Konturen aus anderen Klassen unterscheidet.

Die experimentellen Ergebnisse zeigen ausführlich die Leistungsfähigkeit der in dieser Arbeit vorgestellten Erkennungsmethode basierend auf der automatischen Merkmalsgenerierung durch eine *automatische Merkmalsynthese*. So können beispielsweise auch komplexe Datensätze nur mit Hilfe eines einzigen verlässlichsten signifikanten Konturabschnitts pro Kontur in der Lernstichprobe mit einer hohen Erkennungsrate klassifiziert werden.

Abstract

This work deals with the recognition of isolated visual objects. These objects stem from the real world and in general they are complex and deformed.

The outer contours of 3d-Objects, that are projected on a plane, are analyzed with regard to local *significant contour sections*. These sections are sufficient in order to distinguish the contours from different classes.

It is emphasized that the subject *automatic feature generation* is to the fore in this work. Up to now there is paid little attention to this subject. The *automatic feature generation* must not be confused with the *feature selection*. *Feature selection* deals with the selection of an as much as possible optimal subset from a given feature set. Also *automatic feature generation* must not be confused with the **unsystematic** conventional *feature extraction* methods. In the field of traditional *feature extraction* predefined features like the length-width-ratio are extracted from an object in the image. In the field of *automatic feature generation* abstract features like the top of a bottle or the fin of a fish are generated systematically. This is done by synthesizing more complex features out of basic features of the same kind like successive contour points.

The introduced system consists of two main modules. The first one is an autonomous learning system that consists of various separate modules and the second one is a hierarchical classifier. The main attention is not directed to the construction of a classifier, but to a systematic innovative feature generation method based on *automatic feature synthesis* (see figure 2).

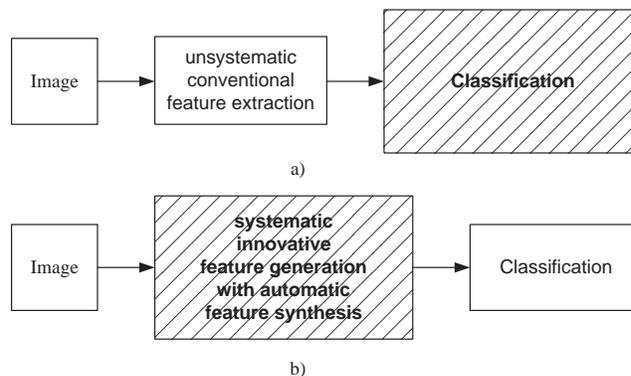


Figure 2: The difference to conventional methods in principal. a) Conventional recognition techniques extract features from images unsystematically. The main attention of such methods is directed to the development of classifiers that provide as much as possible good recognition rates with that unsystematically found features. b) The main attention of the newly developed method is directed to the systematic generation of meaningful features and not to the development of classifiers.

The input data of the learning system are the outer contours of objects stemming from different classes. These contours are analyzed locally in order to get *significant contour sections* distinguishing the objects from different classes. With these *significant contour sections* the knowledge base of a fast hierarchical classifier consists of compressed contours. The contours are compressed because the contour sections that are unimportant for distinguishing objects from different classes are eliminated.

In the centre of this method is the so called *automatic feature synthesis*. First of all the *basic features* contour points are synthesized to *important contour sections* that exactly distinguish 2 contours at a time. The *important contour sections* of one contour being generated out of the comparison with contours from other classes afterwards are synthesized to *significant contour sections* distinguishing that contour from the contours from the other classes.

The experimental results show in detail the potential of the recognition method based on *automatic feature synthesis* introduced in this work. For example, complex data sets are classified with an high recognition rate with the help of only one most reliable *significant contour section* per contour in the learning sample.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Objekterkennung	3
1.1.1	Der Ähnlichkeitsbegriff	4
1.1.2	Problematisierung des Klassifikationsproblems	10
1.2	Aufgabenstellung	20
1.3	Aufbau der Arbeit	26
2	Einordnung von 2D-Konturanalyseverfahren	29
2.1	Klassifizierung nach Pavlidis	30
2.2	Globale und lokale Verfahren	31
2.2.1	Formzahlen (engl. shape numbers)	32
2.2.2	Momente	34
2.2.3	Fourierdeskriptoren	38
2.3	Invariante und variante Verfahren	44
3	Analyse deformierter 2D-Konturen	49
3.1	Teilbasierter Ansatz	51
3.2	Strukturelle Mustererkennung	58
3.3	Syntaktische Anpassung	62
3.4	Elastische Anpassung	67
3.5	Diskussion	71
4	Merkmalsynthese	73
4.1	Überblick über das Verfahren	74
4.1.1	Überblick Merkmalsynthese (Lernphase 1)	77
4.1.2	Überblick Klassifikatorsynthese (Lernphase 2)	79
4.1.3	Überblick Klassifikation	80
4.2	Basisnotationen	80
4.3	Synthese signifikanter Konturabschnitte	82
4.3.1	Kosten für das Ersetzen eines Konturpunktes	83
4.3.2	Korrespondenzliste und globale Ähnlichkeit	88

4.3.3	Einführung von Klassen	101
4.3.4	Wichtige Konturabschnitte	104
4.3.5	Signifikante Konturabschnitte	108
4.4	Ein hierarchischer Klassifikator	116
4.4.1	Erzeugung einer Vergleichstabelle	119
4.4.2	Attribute eines signifikanten Konturabschnittes	119
5	Experimentelle Ergebnisse	123
5.1	Distanz von 2 Konturabschnitten	124
5.2	Lokale (Distanz-)Ähnlichkeitsmatrizen	126
5.3	Korrespondenzlisten mit minimalen Kosten	133
5.3.1	Auswirkung von l_{const} und E auf Korrespondenzlisten . .	133
5.3.2	Auswirkung von Deformation auf Korrespondenzlisten . .	139
5.3.3	Wichtige Konturabschnitte	145
5.4	Globale Distanz- und Ähnlichkeitsmatrizen	147
5.4.1	30 Kunstkonturen	151
5.4.2	Vögel, Flaschen und Fische	154
5.4.3	Datensatz von Sharvit und Kimia	155
5.5	Klassifikation mit signifikanten Abschnitten	162
5.5.1	30 Kunstkonturen	162
5.5.2	Vögel, Flaschen und Fische	169
5.5.3	Datensatz von Kimia	185
5.6	Autonome Modellbildung	196
6	Zusammenfassung und Ausblick	205
	Literaturverzeichnis	209
A	Formelzeichen	221
B	Glossar	233
C	Dynamische Programmierung	245
D	Clusteranalyse	251
E	Algorithmen	261

Abbildungsverzeichnis

1	Prinzipieller Unterschied zu traditionellen Methoden	iii
2	The difference to conventional methods in principal	v
1.1	Aufgabenstellungen im Bereich Industrielle Bildverarbeitung / Machine Vision	2
1.2	Prinzip eines künstlichen Bildverarbeitungssystems	4
1.3	Biegungen an Stellen hoher und niedriger Krümmung	7
1.4	Viele kleine Deformationen versus einer großen Deformation	8
1.5	Der lokale Charakter von Ähnlichkeit.	9
1.6	Schattenrisse verschiedener Objekte mit unterschiedlichem Komplexitätsgrad	13
1.7	Beispiele für die Deformation durch direkte physikalische Einflüsse auf ein Objekt	14
1.8	Beispiele für die Deformation eines Objektes durch Eigenverformung	15
1.9	Beispiele für die perspektivischen Deformationen	16
1.10	Beispiele für die technischen Deformationen	17
1.11	Schattenrisse verschiedener Autos aus dem gleichen Blickwinkel aufgenommen	18
1.12	Schattenrisse verschiedener Fische aus dem gleichen Blickwinkel aufgenommen	19
1.13	Übliches Prinzip der Objektklassifikation	22
1.14	Prinzip des zu entwickelnden Verfahrens	24
1.15	Schattierte Objekte und deren Schattenrisse aus verschiedenen Blickrichtungen	25
1.16	Beispiele für Konturen, die der Untersuchung dieser Arbeit zugrundeliegen	26
2.1	Beispiel für die MAT (Transformation auf mittige Achsen)	30
2.2	Beispiel für das Erzeugen einer <i>shape number</i>	33
2.3	Hauptachsen einer Kontur	37
2.4	Beispiel für die polare Darstellung einer Kontur	40

2.5	Zur Definition der Krümmung	42
2.6	Krümmungswinkel und Φ - s -Funktion eines Polygons	43
2.7	Mögliche Transformationen einer Beispielkontur <i>Hand</i>	45
2.8	Ausrichtung von jeweils 5 nichtidentischen handgeschriebenen Ziffern von 0-9	47
3.1	Glieder und Hälse	51
3.2	Stark konkave Krümmungen	52
3.3	Beispiel zur Kurvenevolution	54
3.4	Zur Kostenfunktion für die diskrete Kurvenevolution	55
3.5	Beispiel zur diskreten Kurvenevolution	56
3.6	Maximale konvexe und konkave Bögen	57
3.7	Konstruktion eines G-Graphen	58
3.8	Segmentierungsgitter und Morphismuskitter	59
3.9	Beispiel für Isomorphismus	61
3.10	Beispiel für die syntaktische Darstellung einer Kontur	63
3.11	Beispiel für die Krümmungsschätzung in einem Punkt	64
3.12	Korrespondenz der Konturpunkte zweier Konturen	68
4.1	Einführungsbeispiel für signifikante Konturabschnitte	75
4.2	Grober Systemüberblick	76
4.3	Überblick Merkmalsynthese (Lernphase 1)	78
4.4	Überblick Klassifikatorsynthese (Lernphase 2)	79
4.5	Diskrete Kontur	82
4.6	Zwei Konturabschnitte der gleichen Bogenlänge	85
4.7	Translation der Konturabschnitte in den Ursprung basierend auf absoluter Orientierung	85
4.8	Rotation der Konturabschnitte um den Ursprung basierend auf absoluter Orientierung	87
4.9	Verschieben der Konturabschnitte längs der x -Achse basierend auf absoluter Orientierung	87
4.10	Translation der Konturabschnitte in den Ursprung basierend auf mittlerer Orientierung	89
4.11	Rotation der Konturabschnitte um den Ursprung basierend auf mittlerer Orientierung	89
4.12	Verschieben der Konturabschnitte längs der x -Achse basierend auf mittlerer Orientierung	89
4.13	Beispiel für eine Teilkorrespondenzliste	93
4.14	Gerichter Graph zur Bestimmung minimaler oder maximaler Kosten	96
4.15	Gerichter Graph - Beispiel	98
4.16	Beispiel für überlappende wichtige Konturabschnitte	107

4.17	Konturpunktindexstrahl mit wichtigen Konturabschnitten	110
4.18	5 Möglichkeiten zur Synthese wichtiger Konturabschnitte zu einem signifikanten Konturabschnitt	113
4.19	Flussdiagramm des hierarchischen Klassifikators	118
5.1	Gestörter Konturabschnitt	124
5.2	Anpassung von Konturabschnitten	125
5.3	Distanzen von Konturabschnitten bei variiertes Störung	126
5.4	Prinzip Ähnlichkeitsmatrix für einfache geometrische Figuren	127
5.5	Die 4 verschiedenen Methoden zur (Distanz-) Ähnlichkeitsmatrixerzeugung im Vergleich	129
5.6	Variationen des Parameters l_{const}	131
5.7	Variationen des Parameters E	131
5.8	4 Methoden zur (Distanz-)Ähnlichkeitsmatrixbestimmung für 2 Fischkonturen	132
5.9	Auswirkungen des Parameters l_{const} bei absoluter Orientierung auf Korrespondenzliste mit minimalen Kosten	134
5.10	Auswirkungen des Parameters l_{const} bei mittlerer Orientierung auf Korrespondenzliste mit minimalen Kosten	135
5.11	Auswirkungen des Parameters E bei absoluter Orientierung auf Korrespondenzliste mit minimalen Kosten	136
5.12	Auswirkungen des Parameters E bei mittlerer Orientierung auf Korrespondenzliste mit minimalen Kosten	137
5.13	Versuchsreihe zur Auswirkung der Eigendeformation auf Korrespondenzliste	140
5.14	Versuchsreihe zur Auswirkung der Perspektive auf Korrespondenzliste	141
5.15	Versuchsreihe zur Auswirkung der physikalischen Deformation auf Korrespondenzliste	142
5.16	Versuchsreihe zur Auswirkung der Variabilität auf Korrespondenzliste	143
5.17	Wichtige Konturabschnitte von 2 Kunstkonturen	146
5.18	Wichtige Konturabschnitte zur Unterscheidung von einer Flasche und einem Fisch	146
5.19	Datensatz mit 30 Kunstkonturen mit und ohne Spitze	148
5.20	Datensatz mit 10 Vögeln, 10 Flaschen und 10 Fischen	149
5.21	Datensatz von Sharvit und Kimia	150
5.22	Dendrogramm der Clusteranalyse mit Ward-Methode für die künstlichen Konturen aus Abbildung 5.19	154
5.23	Dendrogramm der Clusteranalyse mit Ward-Methode für die Konturen von den Vögeln, Flaschen und Fischen aus Abbildung 5.20	155

5.24	Dendrogramm der Clusteranalyse mit Ward-Methode für die Konturen des Sharvit-Kimia-Datensatzes aus Abbildung 5.21	158
5.25	Signifikante Konturabschnitte der 30 künstlichen Konturen mit und ohne Spitzen	163
5.26	Verlässliche Konturabschnitte der 30 künstlichen Konturen mit und ohne Spitze	164
5.27	Teststichprobe mit 30 künstlichen Konturen mit und ohne Spitze .	166
5.28	Fehlklassifikation einer Kontur ohne Spitze	168
5.29	Signifikante Konturabschnitte von 10 Vögeln, 10 Flaschen und 10 Fischen für Importanzschwelle $\tilde{\Theta} = 10$	171
5.30	Verlässliche Konturabschnitte von 10 Vögeln, 10 Flaschen und 10 Fischen für Importanzschwelle $\tilde{\Theta} = 10$ und $L_{min} = 10$	172
5.31	Signifikante Konturabschnitte von 10 Vögeln, 10 Flaschen und 10 Fischen für Importanzschwelle $\tilde{\Theta} = 13$	173
5.32	Verlässliche Konturabschnitte von 10 Vögeln, 10 Flaschen und 10 Fischen für Importanzschwelle $\tilde{\Theta} = 13$ und $L_{min} = 10$	174
5.33	Signifikante Konturabschnitte von 10 Vögeln, 10 Flaschen und 10 Fischen für Importanzschwelle $\tilde{\Theta} = 15$	175
5.34	Verlässliche Konturabschnitte von 10 Vögeln, 10 Flaschen und 10 Fischen für Importanzschwelle $\tilde{\Theta} = 15$ und $L_{min} = 10$	176
5.35	Teststichprobe mit 16 Vögeln, 22 Flaschen und 36 Fischen	177
5.36	Fehlklassifikationen von Flaschen und Fischen für $\tilde{\Theta} = 10$	182
5.37	Fehlklassifikationen von Flaschen und Fischen für $\tilde{\Theta} = 13$	183
5.38	Fehlklassifikationen von Flaschen und Fischen für $\tilde{\Theta} = 15$	184
5.39	Signifikante Konturabschnitte von den Konturen aus dem Datensatz von Kimia [LEM] für den Importanzschwelle $\tilde{\Theta} = 10$	187
5.40	Verlässliche Konturabschnitte von den Konturen der Lernstichprobe aus dem Datensatz von Kimia [LEM] für Importanzschwelle $\tilde{\Theta} = 10$ und $L_{min} = 10$	188
5.41	Signifikante Konturabschnitte von den Konturen aus dem Datensatz von Kimia [LEM] für den Importanzschwelle $\tilde{\Theta} = 13$	189
5.42	Verlässliche Konturabschnitte von den Konturen der Lernstichprobe aus dem Datensatz von Kimia [LEM] für Importanzschwelle $\tilde{\Theta} = 13$ und $L_{min} = 10$	190
5.43	Teststichprobe aus dem Datensatz von Kimia [LEM].	191
5.44	Fehlklassifikation von Fischen, einem Greeble und Maulschlüsseln für $\tilde{\Theta} = 10$	194
5.45	Fehlklassifikation von Greebles und Kaninchen für $\tilde{\Theta} = 13$	195
5.46	Ansichten und Außenkonturen der Objekte Hefter, Kleberrollenhalter und Locher	198

5.47 Geordnete Ansichten und Außenkonturen der Objekte Hefter, Kleberrollenhalter und Locher 200

5.48 Visualisiertes Modell der Hefter-Kleberrollenhalter-Locher-Umwelt 201

5.49 Problematische Ansichten durch Projektion 202

5.50 Ansichten des Objektes *Tisch mit Monitor* 202

C.1 Beispiel Treibstoffproblem 246

D.1 Dendogramm 253

D.2 Beispiel Rohdatenmatrix 256

D.3 Beispieldendogramm Single Linkage 259

D.4 Beispieldendogramm Complete Linkage 260

E.1 Algorithmus Ia zur Berechnung der Distanz von 2 Konturabschnitten basierend auf absoluter Orientierung 262

E.2 Algorithmus Ib zur Berechnung der Distanz von 2 Konturabschnitten basierend auf mittlerer Orientierung 263

E.3 Algorithmus IIa zur Berechnung der lokalen Distanzmatrix für konstante $l_{i,j,k}, l_{r,s,t}$ 264

E.4 Algorithmus IIb zur Berechnung der lokalen Ähnlichkeitsmatrix für variable $l_{i,j,k}, l_{r,s,t}$ 265

E.5 Algorithmus IIIa zur Berechnung eines bestimmten kostengünstigsten Pfades basierend auf $\Lambda^{(i,j),(r,s)}$ 266

E.6 Algorithmus IIIb zur Berechnung eines bestimmten kostengünstigsten Pfades basierend auf $\tilde{\Lambda}^{(i,j),(r,s)}$ 267

E.7 Algorithmus IVa zur Berechnung des günstigsten Pfades in $\Lambda^{(i,j),(r,s)}$ 268

E.8 Algorithmus IVb zur Berechnung des günstigsten Pfades in $\tilde{\Lambda}^{(i,j),(r,s)}$ 269

E.9 Algorithmus V für das Erzeugen der Korrespondenzliste $Q^{(i,j),(r,s)}$ in $\Lambda^{(i,j),(r,s)}$ 270

E.10 Algorithmus VIa zur Berechnung von $w^{(I,J),(R,S)}$ basierend auf der lokalen Distanzmatrix $\Lambda^{(I,J),(R,S)}$ 271

E.11 Algorithmus VIb zur Berechnung von $\tilde{w}^{(I,J),(R,S)}$ basierend auf der lokalen Ähnlichkeitsmatrix $\tilde{\Lambda}^{(I,J),(R,S)}$ 272

E.12 Algorithmus VIIa zur Berechnung von $W^{I,J}$, basierend auf den lokalen Distanzmatrizen $\Lambda^{(I,J),(r,s)}$ 273

E.13 Algorithmus VIIb zur Berechnung von $\tilde{W}^{I,J}$, basierend auf den lokalen Ähnlichkeitsmatrizen $\tilde{\Lambda}^{(I,J),(r,s)}$ 274

E.14 Algorithmus VIII für den hierarchischen Klassifikator 275

Tabellenverzeichnis

5.1	Tabelle mit globalen Ähnlichkeiten zwischen Referenzente 1 und den Konturen 2 - 4 aus Abbildung 5.13 für Eigendeformation. . .	144
5.2	Tabelle mit globalen Ähnlichkeiten zwischen Referenzente 5 und den Konturen 6 - 8 aus Abbildung 5.14 für perspektivische Deformation.	144
5.3	Tabelle mit globalen Ähnlichkeiten zwischen Referenzflasche 1 und den Konturen 2 - 4 aus Abbildung 5.15 für physikalische Deformation.	144
5.4	Tabelle mit globalen Ähnlichkeiten zwischen Referenzkontur 1 und den Konturen 2 - 4 aus Abbildung 5.16 für Variabilität	145
5.5	Globale Distanzmatrix mit den globalen Distanzen der künstlichen Konturen in Abbildung 5.19 mithilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$. . .	152
5.6	Tabelle mit Klassifikationsergebnissen für die künstlichen Konturen in Abbildung 5.19 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$	153
5.7	Tabelle mit Klassifikationsergebnissen für die künstlichen Konturen in Abbildung 5.19 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 10$	153
5.8	Tabelle mit Klassifikationsergebnissen für die künstlichen Konturen in Abbildung 5.19 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 1000.0$	153
5.9	Tabelle mit Klassifikationsergebnissen für die künstlichen Konturen in Abbildung 5.19 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 200.0$	153
5.10	Tabelle mit Klassifikationsergebnissen für die Konturen von den Vögeln, Flaschen und Fischen in Abbildung 5.20 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$	156

5.11	Tabelle mit Klassifikationsergebnissen für die Konturen von den Vögeln, Flaschen und Fischen in Abbildung 5.20 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 10$	156
5.12	Tabelle mit Klassifikationsergebnissen für die Konturen von den Vögeln, Flaschen und Fischen in Abbildung 5.20 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 1000.0$	156
5.13	Tabelle mit Klassifikationsergebnissen für die Konturen von den Vögeln, Flaschen und Fischen in Abbildung 5.20 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 200.0$	156
5.14	Globale Distanzmatrix mit den globalen Distanzen der Konturen von den Vögeln, Flaschen und Fischen in Abbildung 5.20 mithilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$	157
5.15	Tabelle mit Klassifikationsergebnissen für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$.	159
5.16	Tabelle mit Klassifikationsergebnissen für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 10$.	159
5.17	Tabelle mit Klassifikationsergebnissen für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 1000.0$	159
5.18	Tabelle mit Klassifikationsergebnissen für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 200.0$	159
5.19	Tabelle mit Klassifikationsergebnissen von Sharvit und Kimia für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe von <i>Shock-Graphs</i>	159
5.20	Tabelle mit Klassifikationsergebnissen von Gdalyahu und Weins hall für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe der <i>Flexible Syntactic Matching</i> - Methode [GW99]	160
5.21	Globale Distanzmatrix mit den globalen Distanzen der Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mithilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$	161

5.22	Tabelle mit Klassifikationsergebnissen des hierarchischen Klassifikators für die 30 künstlichen Konturen in der Teststichprobe . . .	165
5.23	Tabelle mit den detaillierten Klassifikationsergebnissen des hierarchischen Klassifikators für die 30 künstlichen Konturen in der Teststichprobe	167
5.24	Tabelle mit Trefferquoten des hierarchischen Klassifikators für die Vögel, Flaschen und Fische in der Teststichprobe für verschiedene Parameterkombinationen	169
5.25	Tabelle mit den detaillierten Klassifikationsergebnissen des hierarchischen Klassifikators für die Fische, Flaschen und Vögel in der Teststichprobe für $\tilde{\Theta} = 10$	178
5.26	Tabelle mit Klassifikationsergebnissen des hierarchischen Klassifikators für die Fische, Flaschen und Vögel in der Teststichprobe für $\tilde{\Theta} = 10$	178
5.27	Tabelle mit den detaillierten Klassifikationsergebnissen des hierarchischen Klassifikators für die Fische, Flaschen und Vögel in der Teststichprobe für $\tilde{\Theta} = 13$	179
5.28	Tabelle mit Klassifikationsergebnissen des hierarchischen Klassifikators für die Fische, Flaschen und Vögel in der Teststichprobe für $\tilde{\Theta} = 13$	179
5.29	Tabelle mit den detaillierten Klassifikationsergebnissen des hierarchischen Klassifikators für die Fische, Flaschen und Vögel in der Teststichprobe für $\tilde{\Theta} = 15$	180
5.30	Tabelle mit Klassifikationsergebnissen des hierarchischen Klassifikators für die Fische, Flaschen und Vögel in der Teststichprobe für $\tilde{\Theta} = 15$	180
5.31	Tabelle mit Trefferquoten des hierarchischen Klassifikators für den Datensatz von Kimia in der Teststichprobe (aus [LEM]) für verschiedene Parameterkombinationen	185
5.32	Tabelle mit den detaillierten Klassifikationsergebnissen des hierarchischen Klassifikators für die Konturen aus dem Datensatz von Kimia [LEM] in der Teststichprobe für $\tilde{\Theta} = 10$	192
5.33	Tabelle mit Klassifikationsergebnissen des hierarchischen Klassifikators für den Datensatz von Kimia [LEM] in der Teststichprobe für $\tilde{\Theta} = 10$	192
5.34	Tabelle mit den detaillierten Klassifikationsergebnissen des hierarchischen Klassifikators für die Konturen aus dem Datensatz von Kimia [LEM] in der Teststichprobe	193
5.35	Tabelle mit Klassifikationsergebnissen des hierarchischen Klassifikators für den Datensatz von Kimia [LEM] in der Teststichprobe für $\tilde{\Theta} = 13$	193

5.36	Ergebnis der globalen Distanzberechnung basierend auf absoluter Orientierung für $l_{const}=10$ für die Ansichten des Hefters, des Kleberollenhalters und des Lochers.	199
5.37	Globale Distanzmatrix für die Ansichten des Objektes <i>Tisch mit Monitor</i>	203

Kapitel 1

Einleitung

Eines der leistungsfähigsten Teilsysteme eines natürlichen intelligenten Systems ist das Sehsystem, das für eine Vielzahl alltäglicher (Überlebens-)Aufgaben zuständig ist. Freunde werden von Feinden unterschieden, Objekte unserer Umwelt werden erkannt, Hindernissen kann ausgewichen und Dinge können gezielt angefasst werden. Nahrung wird gesammelt oder gejagt und die Menschheit ist prinzipiell in der Lage Fortbewegungsmaschinen zu kontrollieren.

Dabei kann es eine Vielzahl von (Ab)Bildern seiner Umwelt in kurzer Zeit verarbeiten. Objekte werden im Bild lokalisiert und auf bestimmte Merkmale hin untersucht. Eine nahezu beliebig große Anzahl an Objekten wird erkannt und unterschieden. Darüberhinaus können Längen und Entfernungen abgeschätzt und Bewegungen detektiert werden.

Dies alles sind Fähigkeiten, die die Entwicklung technischer Sehsysteme wirtschaftlich interessant machen. Sind Wissenschaftler, Ingenieure und Techniker in der Lage ähnlich leistungsfähige Systeme zu entwickeln, so ist eine Vielzahl an möglichen Anwendungen denkbar.

Nach einer Marktbefragung des VDMA (Verband Deutscher Maschinen- und Anlagenbauer e.V.) [VDM99a], Fachabteilung Industrielle Bildverarbeitung / Machine Vision sind 1998/1999 folgende Aufgabenstellungen bereits industriell umgesetzt (siehe Abb. 1.1):

- Oberflächeninspektion von Teilen
- Oberflächeninspektion von Endlosmaterial, z.B. Papier
- Lage und Positionskontrolle im Bereich Robot Vision zur Produktionsautomatisierung
- Druckbildkontrolle
- 2D-Vermessung

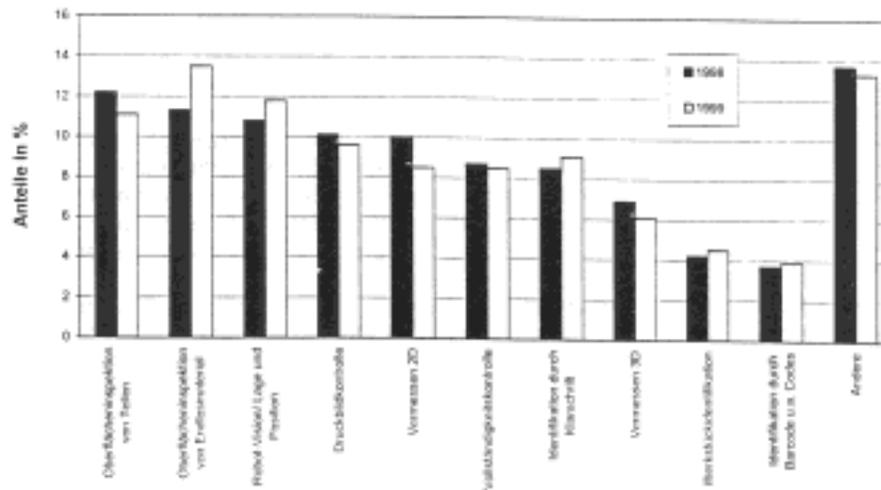


Abbildung 1.1: Aufgabenstellungen im Bereich Industrielle Bildverarbeitung / Machine Vision im Jahre 1998/1999 laut einer Marktbefragung der Fachabteilung Industrielle Bildverarbeitung / Machine Vision in der Fachgemeinschaft Robotik + Automation im VDMA (Abbildung stammt aus [VDM99a]).

- 3D Vermessung
- Vollständigkeitskontrolle
- Identifikation durch Klarschrift
- Werkstückidentifikation
- Identifikation durch Barcodes und andere Codes
- ca. 14 Prozent andere Aufgabenstellungen

Das wirtschaftliche Interesse an technischen Sehsystemen wird durch folgende Zahlen unterstrichen: Nach Umsätzen von 450 Millionen DM, 500 Millionen DM, 570 Millionen DM, 650 Millionen DM und 800 Millionen DM in den Jahren 1995, 1996, 1997, 1998 und 1999 [Sch97], [VIS00] erreicht der deutsche Gesamtmarkt für die Industrielle Bildverarbeitung nach [VDM99b] im Jahr 2000 schätzungsweise ein Umsatzvolumen von rund 1 Mrd. DM und es werden Steigerungsraten von jährlich mehr als 15% prognostiziert. Außerdem wird davon ausgegangen, dass heute erst ca. 15-20% der zur Zeit möglichen Anwendungen erschlossen sind.

Die bereits im industriellen Einsatz befindlichen technischen Sehsysteme zeichnen sich durch eine hohe Schnelligkeit und Robustheit aus. Allerdings sind sie zum einen oft speziell für die zu lösende Aufgabe entwickelt worden und lassen sich damit nicht unbedingt einfach auf andere Aufgabenstellungen übertragen. Zum anderen taucht der Bereich Objekterkennung, abgesehen von der relativ einfachen Identifikation genormter Werkstücke, nicht auf, was den Schluss zulässt, dass eine komplexe Objekterkennung noch nicht wirtschaftlich genug für den industriellen Einsatz ist.

Bei näherer Untersuchung zeigt sich, dass infragekommene Produkte, wie Serviceroboter, autonome Fahrzeuge oder Sortiermaschinen mit komplexem Sortiergut den Weg aus den Forschungsabteilungen der Industrie oder den Labors der Universitäten noch nicht in nennenswertem Umfang gefunden haben.

Serviceroboter beispielsweise müssen sich in fremden und/oder wechselnden Umgebungen orientieren können. Diese Umgebungen können sehr komplex sein. Sollen sie uns Menschen im direkten Kontakt behilflich sein, z.B. bei schwerbehinderten oder älteren Personen, müssen sie in der Lage sein sich auf die Umwelt dieser Personen einzustellen. Um sich in einer solch komplexen Umwelt sicher bewegen und Objekte dieser Umwelt handhaben zu können, ist ein künstliches Sehsystem mit objekterkennenden Fähigkeiten sicher sehr hilfreich.

1.1 Objekterkennung

Ein wichtiges Teilgebiet beim Computersehen (\rightarrow *Computer Vision*) ist die \rightarrow *Objekterkennung*. Damit eine Maschine *Objekte* visuell wahrnehmen und *Objekte* erkennen kann, benötigt sie ein künstliches Sehsystem. Abb. 1.2 zeigt den prinzipiellen Aufbau eines solchen künstlichen Sehsystems. Der intelligente Teil eines solchen Sehsystems untersucht visuelle *Objekte* auf ihre markanten Eigenschaften hin, um diese markanten Eigenschaften bei anderen *Objekten* wiederzufinden.

Die vorliegende Arbeit befasst sich also mit der visuellen Objekterkennung, d.h. mit der Analyse und Klassifikation von *Objekten* in Bildern. Mit *Objekt* ist dabei ein Gegenstand, eine Sache oder Person der realen Welt gemeint, der/die unabhängig vom menschlichen oder einem maschinellen Bewusstsein existiert. Der letzte Relativsatz ist wichtig, da er ausdrückt, dass ein intelligentes System auch mit neuen, ihm noch unbekanntem *Objekten* umgehen können muss.

Ein prinzipiell zu lösendes Problem in der Objekterkennung ist es, eine bestimmte Anzahl von *Objekten* einer \rightarrow *Aufgabendomäne* (Menge aller Objekte, die für eine bestimmte zu lösende Aufgabe relevant sind) einer bestimmten Anzahl unterschiedlicher \rightarrow *Klassen* möglichst sicher zuzuordnen (\rightarrow *Klassifikation*). In diesem Zusammenhang wird unter einer bestimmten *Klasse* diejenige Untermenge aller Objekte der Aufgabendomäne verstanden, die sich *ähnlich* sind.

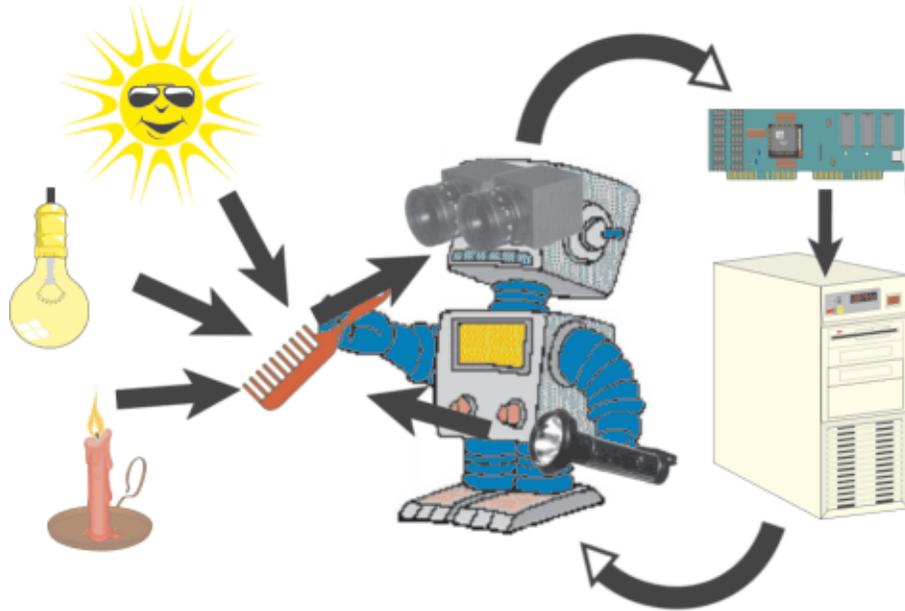


Abbildung 1.2: Das skizzierte Prinzip eines künstlichen Bildverarbeitungssystems besteht aus 2 beweglichen Kameras mit Objektiven, die einer Auswerteeinheit (Rechner) über einen Framegrabber digitalisierte (gerasterte und quantisierte) Bilder eines Objektes *Kamm* liefern. Die Auswertung der Bilder kann dafür sorgen, dass die Kameras einen anderen Blickwinkel zum Objekt einnehmen, dass das Objekt bewegt wird, dass die aktive Beleuchtung (hier dargestellt durch eine Taschenlampe) geändert oder natürlich, dass das Objekt erkannt wird. Ferner ist anzumerken, dass je nach Umwelt auf bestimmte Beleuchtungen und Beleuchtungsänderungen, wie z.B. durch das Sonnenlicht oder künstliche Lichtquellen, kein Einfluß genommen werden kann.

Das bedeutet auch, dass unterschiedliche Klassen Objekte beinhalten, die *unähnlich* sind.

1.1.1 Der Ähnlichkeitsbegriff

Die Begriffe *ähnlich* und *unähnlich* sind schnell dahingeschrieben. Intuitiv hat jeder von uns eine (eigene) Vorstellung davon was gemeint ist, wenn bestimmte Objekte als sich einander ähnlich bezeichnet werden. Eine klare Definition von *Ähnlichkeit* ist aber schwierig. Da die Ähnlichkeit ein zentraler Begriff dieser Arbeit ist und die Beschäftigung mit dieser Begrifflichkeit nicht einfach übergangen werden soll und, um einen Eindruck von der Schwierigkeit einer klaren Definition von Ähnlichkeit zu bekommen, werden an dieser Stelle deshalb kurz einige

wenige Gedanken und Zitate zur Ähnlichkeit formuliert:

Das Wort "ähnlich" stammt vom mittelhochdeutschen "enehil" ab, was "großväterlich" bzw. "dem Ahnen gleich" bedeutet. Etymologisch gesehen gründet das Wort also auf der individuellen Abstammung des Menschen.

Ausgehend von der Definition:

Ähnlichkeit ist die Bezeichnung für eine weitgehende, aber nicht vollständige Übereinstimmung von Dingen.

lassen sich verschiedene Ansätze für die Konkretisierung dieses Begriffes nachlesen.

Zu Beginn sei sinngemäß die sehr harte Definition, auf die man in mathematischen Nachschlagewerken trifft, genannt:

Zwei Figuren werden ähnlich genannt, wenn alle korrespondierenden Winkel gleich sind.

Dies entspricht sicherlich nur einer Teilmenge unseres intuitiven Verständnisses von Ähnlichkeit.

Leibniz schrieb 1687 in seinem Text "Zur Analysis der Lage" [Lei66] folgende Definition nieder:

"Wir werden zwei vorliegende Figuren ähnlich nennen, wenn man nicht imstande ist, in der einen, für sich betrachtet, ein Merkmal anzugeben, das sich nicht auch in der anderen vorfände."

Diese Definition umschreibt schon näherungsweise, die dieser Arbeit zugrundeliegende Auffassung von Ähnlichkeit.

Niemann postuliert in [Nie83] als Mustererkenner konkreter:

*"Zwei Muster sind ähnlich, wenn ihre Merkmale oder ihre einfacheren Bestandteile sich nur **wenig unterscheiden**. ... Bei numerischen Merkmalen lassen sich Unterschiede durch Metriken und andere Abstandsmaße definieren. **Wenig unterscheiden** heißt dann, dass der Wert des Abstandsmaßes unterhalb einer Schwelle bleibt. ... Merkmale oder einfachere Bestandteile müssen, wenn sie nützlich sein sollen, so gewählt werden, dass den Anwender **interessierende Ähnlichkeiten** in ihnen zum Ausdruck kommt."*

Hier wird jedoch vorausgesetzt, dass der Anwender die **interessierenden Ähnlichkeiten** genau definieren kann.

Interessant auch der folgende Definitionsansatz der Musikwissenschaftler Markuse und Schneider [MS00], die sich mit der Ähnlichkeit psychoakustischer Objekte, wie z.B. Klängen auseinandersetzen:

”Streng genommen geht die maximal zu erreichende Ähnlichkeit in Identität dann über, wenn alle Merkmale zweier Objekte (oder einer Mehrzahl von solchen in ... Klassen) übereinstimmen. Bei geringfügigen Unterschieden werden Gebilde als **verwandt** offenbar deshalb empfunden, weil eine Anzahl als konstitutiv oder jedenfalls prägend erachteter Merkmale mit in etwa gleicher Abstufung vorhanden ist. **Verwandte** Gebilde erscheinen daher auf einem subjektiven Ähnlichkeitskontinuum als benachbart, der phänomenalen Ähnlichkeit korrespondiert insoweit die Vorstellung auch räumlicher Nähe.”

Hier wird angesprochen, dass unterschieden werden kann zwischen der Ähnlichkeit von 2 Objekten und der Ähnlichkeit einer Vielzahl von Objekten in einer Klasse. Außerdem wird darauf hingewiesen, dass Ähnlichkeit etwas durchaus subjektives sein kann.

Lin [Lin98] unternimmt eine universelle informations-theoretische Definition, die allerdings ein probabilistisches Modell voraussetzt, indem er folgende Intuitionen über die Ähnlichkeit formuliert (Übersetzung):

- *Die Ähnlichkeit zwischen A und B steht in Bezug zu ihren Gemeinsamkeiten. Je mehr Gemeinsamkeiten sie haben, desto ähnlicher sind sie sich.*
- *Die Ähnlichkeit zwischen A und B steht in Bezug zu den Verschiedenheiten zwischen ihnen. Je mehr Unterschiedlichkeiten sie haben, desto unähnlicher sind sie.*
- *Die maximale Ähnlichkeit zwischen A und B ist erreicht, wenn A und B identisch sind.*

Hier wird der interessante Aspekt angedeutet, dass je mehr Gemeinsamkeiten zwei Objekte haben, sie deshalb nicht unbedingt desto weniger Unterschiedlichkeiten haben müssen.

Genauer betrachtet seien an dieser Stelle die Überlegungen von Basri et al. [BCGJ95], [BCGJ98], die sich detaillierte Gedanken über die Eigenschaften einer \rightarrow *Kostenfunktion* gemacht haben, die Vermutungen über die menschliche Ähnlichkeitswahrnehmung miteinbeziehen. Dies bedeutet vor allem, dass sie für die Eigenschaften einer Kostenfunktion berücksichtigen, dass Objekte oft aus einzelnen Teilen zusammengesetzt dargestellt werden. Ein Mensch besteht beispielsweise aus Kopf, Armen, Beinen, einem Rumpf usw. . Im Einzelnen sollte eine Kostenfunktion $C(O_1, O_2)$ zur Unähnlichkeitsbestimmung von 2 Objekten O_1, O_2 nach Basri et al. folgende 7 Eigenschaften besitzen:

1. ***C* sollte kontinuierlich sein.**

D.h., wenn sich ein Objekt gegenüber einem anderen kontinuierlich (nicht sprunghaft) verändert, so verändert sich auch C kontinuierlich.

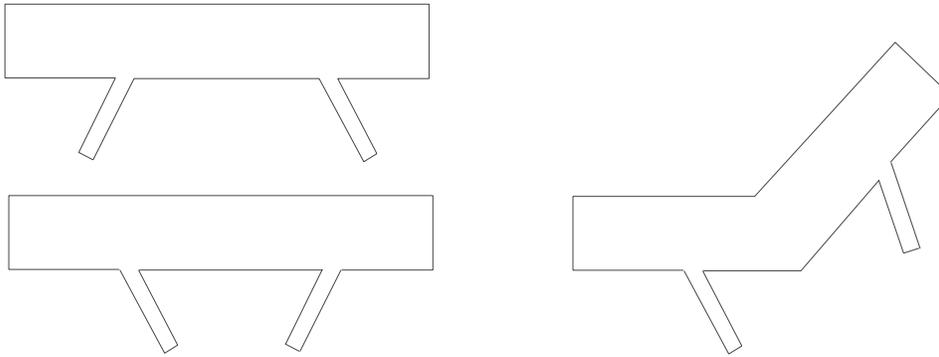


Abbildung 1.3: Die Objekte auf der linken Seite können durch Biegungen an Stellen hoher Krümmung ineinander überführt werden. Die unteren beiden Objekte können durch eine Biegung an einer Stelle niedriger Krümmung ineinander überführt werden. Die Gesamtbiegung (Änderung der Krümmung), um das Objekt links unten in das Objekt oben zu überführen, ist eher größer als die Biegung, um das Objekt links unten in das Objekt rechts zu überführen. Die Objekte links wirken aber trotzdem ähnlicher auf einen menschlichen Betrachter als die Objekte unten.

2. C sollte eine \rightarrow **Metrik** sein.

Dies ist sinnvoll, da eigentlich alle effizienten Methoden zur Detektion des \rightarrow *nächsten Nachbarn* (engl. *nearest neighbour*) auf den metrischen Eigenschaften einer Ähnlichkeitsfunktion bzw. Kostenfunktion basieren. C ist eine Metrik wenn gilt:

$$\begin{aligned}
 C(O_1, O_2) &\geq 0 \\
 C(O_1, O_1) &= 0 \\
 C(O_2, O_2) &= 0 \\
 C(O_1, O_2) &> 0 \quad \forall O_1 \neq O_2 \quad (1.1) \\
 C(O_1, O_2) &= C(O_2, O_1) \\
 C(O_1, O_2) + C(O_2, O_3) &\geq C(O_1, O_3)
 \end{aligned}$$

3. C ist **invariant gegenüber bestimmten Transformationen**.

Die Unähnlichkeit soll möglicherweise unabhängig sein von Rotation und Lage der Objekte oder anderen Transformationen.

4. C sollte sowohl für **kontinuierliche als auch nichtkontinuierliche (diskrete) Objekte** gelten.

D.h., dass das gleiche C sowohl für den Vergleich von kontinuierlichen Ob-

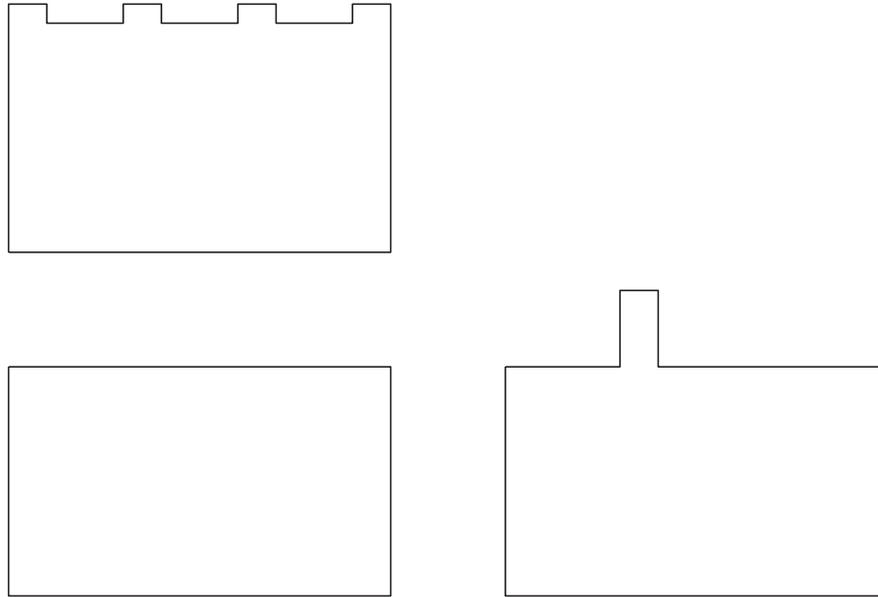


Abbildung 1.4: Das Objekt oben hat viele kleine Deformationen bezogen auf das Objekt links unten. Das Objekt rechts besitzt eine gleichgroße Deformation des Objekts links unten wie das Objekt oben, allerdings nur an einer einzigen Stelle. Die Objekte links wirken aber trotzdem ähnlicher auf einen menschlichen Betrachter als die unteren Objekte.

jekten als auch für den Vergleich von diskreten Objekten herangezogen werden kann.

5. C sollte die Teilstruktur berücksichtigen.

Das Verbiegen eines Objektes an Stellen hoher Krümmung, d.h. an den Stellen, an denen Teile beginnen oder enden, verursacht weniger Kosten als das Biegen an Stellen mit wenig Krümmung. Oder anders ausgedrückt: Objekte erscheinen leichter krümmbar an Stellen hoher Krümmung (siehe Abbildung 1.3).

Ferner sollte die Kostenfunktion berücksichtigen, dass bei der Rotation eines Teiles um die Stelle der Anbindung an den Rest des Objektes sich nur die Stellung des Teiles gegenüber dem Restobjekt ändert, das Teil selbst aber nicht.

6. C sollte monoton sein.

Die Unähnlichkeit zwischen einem Originalobjekt und seiner veränderten Variante sollte wachsen, wenn die Veränderung sich vergrößert.



Abbildung 1.5: Diese Abbildung zeigt ein Beispiel für den lokalen Charakter von Ähnlichkeit. Die beiden abgebildeten Konturen von Flaschen sind sich vor allem wegen ihrer fett markierten Flaschenhälse ähnlich.

7. *C* sollte rauschunempfindlich sein.

Viele kleine Deformationen sollen zu weniger Kosten führen als eine einzige Deformation mit der gleichen Gesamtgröße (Abbildung 1.4).

In [BCGJ95], [BCGJ98] wird aber angedeutet, dass diese wünschenswerten Eigenschaften an ein Ähnlichkeitsmaß, für die bisher bekannten Methoden zur Ähnlichkeitsanalyse, nicht alle gleichzeitig eingehalten werden können. Auch die von ihnen selbst (siehe Abschnitt 3.4) vorgeschlagenen Kostenfunktionen erfüllen nicht alle der 7 genannten Forderungen.

Mit diesem kleinen Ausflug in die Fragestellung, was denn unter dem Ähnlichkeitsbegriff eigentlich zu verstehen sei, wird klar wie schwierig eine exakte Definition von Ähnlichkeit ist. Auch diese Arbeit gibt keine universelle Antwort auf das was Ähnlichkeit wirklich bedeutet. Sie versucht aber den lokalen Charakter von Ähnlichkeit zu unterstreichen, d.h., dass Objekte auch dann als ähnlich erachtet werden, wenn nur bestimmte lokale Bestandteile ähnlich sind, die signifikant für das ganze Objekt sind. Ein Beispiel hierfür illustriert die Abbildung 1.5. Die beiden abgebildeten Außenkonturen von Flaschen sind sich deshalb ähnlich, weil sie beide einen Flaschenhals besitzen (fett markiert). Der Rest der Konturen ist für ihre Ähnlichkeit relativ unwichtig. Diese Arbeit zeigt exemplarisch einen Weg auf, diese *interessierenden Ähnlichkeiten* von einem intelligenten System autonom herauszufinden.

Nach diesem kleinen Exkurs zur Problematisierung des Ähnlichkeitsbegriffes wird im nächsten Abschnitt auf das Klassifikationsproblem eingegangen. D.h. es werden verschiedene Aspekte erörtert, die im Zusammenhang mit der Aufgabe stehen, *ähnliche* Objekte zu einzelnen Klassen zusammenzufassen.

1.1.2 Problematisierung des Klassifikationsproblems

Wie bereits angedeutet, besteht das Klassifikationsproblem darin, die Objekte einer Aufgabendomäne bestimmten Klassen zuzuordnen. Dieses Problem wird in 2 Schritten gelöst.

Im ersten Schritt wird eine bestimmte Menge an Objekten der Aufgabendomäne zum *Einlernen* eines \rightarrow *Klassifikators* herangezogen. *Einlernen* bedeutet, dass die Objekte einer mehr oder weniger aufwendigen Vorverarbeitung unterzogen werden, die die Objekte analysiert, relevante Merkmale der Objekte extrahiert und die Merkmale bestimmten Klassen zuordnet. Diese, den Klassen zugeordneten relevanten Merkmale, dienen dann im zweiten Schritt dem Klassifikator als Wissensbasis. Die Menge dieser *eingelernten* Objekte wird als *Lernstichprobe* bezeichnet.

Im zweiten Schritt werden andere Objekte der Aufgabendomäne mit Hilfe der Wissensbasis des Klassifikators bestimmten Klassen zugeordnet, d.h. erkannt. Die Menge dieser Objekte wird als *Teststichprobe* bezeichnet und ist von der *Lernstichprobe* unabhängig.

Es leuchtet ein, dass die Objekte in der *Lernstichprobe* repräsentativ für ihre Klassen sein sollten. Das bedeutet auch, dass eine bestimmte Mindestanzahl an Objekten in der *Lernstichprobe* enthalten sein sollte. Was aber repräsentative Objekte sind und wieviele von diesen in der *Lernstichprobe* enthalten sein müssen, ist im allgemeinen (noch) nicht beantwortbar. Die Beantwortung dieser Frage ist nicht Bestandteil dieser Arbeit. Die z. Zt. übliche Methode ist die Überprüfung der Repräsentativität mit Hilfe der Klassifikation der Teststichprobe. Sind die Klassifikationsergebnisse gut, so kann davon ausgegangen werden, dass die Lernstichprobe repräsentativ war.

Es gibt prinzipiell 2 unterschiedliche Verfahren Klassen festzulegen. Entweder durch einen Trainer, der dem System zu den jeweiligen Objekten bestimmte Klassen vorgibt oder es gibt keinen Trainer und das System versucht selbständig, ähnliche Objekte in gemeinsame Klassen abzubilden. Im ersten Fall spricht man vom *überwachten Lernen*, im zweiten Fall vom *unüberwachten* oder *selbstorganisierten Lernen*. Beide Lernprinzipien sind in der Objekterkennung wiederzufinden. Auch natürliche Systeme, wie beispielsweise Menschen kennen beide Arten des Lernens. Vor allem im Kindesalter werden viele Objekte *überwacht gelernt*, in dem die Eltern oder andere "Trainer" bestimmte Objekte benennen und sie dem Kind zeigen. Zusätzlich entdecken Kinder ihre Welt selbständig, sie fassen (begreifen) Objekte an und lernen so *unüberwacht* ihre Umwelt kennen.

Zusätzlich gibt es noch eine dritte Variante, das *bestärkende Lernen*, bei dem der Trainer dem System nur angibt, ob es eine Klasse selbständig richtig oder falsch gelernt hat.

Ein anderer wichtiger Punkt beim Festlegen von Klassen in einer Aufgaben-

domäne ist die Tatsache, dass Klassen grob (wenig Klassen) oder fein (viele Klassen) festgelegt werden können. Dies hängt von der jeweiligen Aufgabenstellung ab. Die 2 Grenzfälle sind folgende:

1. Jedes Objekt wird in eine eigene, nur dieses Objekt beinhaltende Klasse eingeordnet. In diesem Fall ist die Unterteilung sehr fein, so dass jedes neu zu klassifizierende Objekt aus der Teststichprobe genau einem Objekt aus der Lernstichprobe zugeordnet wird. Dies ist beispielsweise dann der Fall, wenn die Aufgabendomäne ausschließlich aus uns namentlich bekannten Personen bestünde. Jeder einzelnen Person haben wir in unserem Gedächtnis genau eine Klasse, nämlich deren Namen (z.B. "Marlies", "Klara", "Ruth", ...) zugeordnet.
2. Alle Objekte werden genau einer Klasse zugeordnet. Dieser Fall ist nur dann sinnvoll, wenn ausschließlich entschieden werden soll, ob ein Objekt der Teststichprobe irgendeinem Objekt der Lernstichprobe zugeordnet werden kann oder, falls es den Objekten der Lernstichprobe zu unähnlich ist, nicht.

Im allgemeinen gibt es aber eine bestimmte Anzahl an *Oberklassen*, wie z.B. "Menschen", "Vögel", "Fische", "Schuhe" oder "Flugzeuge". Diese können dann meist noch in mehrere Unterklassen unterteilt werden. Bei den "Vögeln" z.B. in "Raubvögel", "Enten" oder "Papageien". In vielen Aufgabenstellungen reicht diese Unterscheidung aus. Bei der Fahrt mit dem Auto ist es beispielsweise wichtig, die anderen Autos als Autos zu erkennen, aber nicht die Marke jedes Autos, geschweige denn seinen Typ. Ein anderes Beispiel ist das Sortieren von Kunststoffwertstoffen [Kuh98], [SN94]. Die hier notwendigen Oberklassen sind "Hohlkörper", "Becher" und "Getränkekartons". Die Oberklasse "Hohlkörper" beinhaltet dann die Unterklassen "Flaschen", "Tuben" und "Cremedosen", die Oberklasse "Becher" die Unterklassen "Joghurtbecher" und "Margarinebecher". Eine interessante Fragestellung ist also folgende:

Was sind die relevanten signifikanten Merkmale einer Oberklasse, die das Charakteristikum dieser Oberklasse darstellen und gibt es Möglichkeiten diese signifikanten Merkmale zu bestimmen?

Diese Frage ist die zentrale Fragestellung dieser Arbeit, auf die später ausführlich eingegangen wird. Aber zunächst soll noch auf weitere Nuancen des Klassifikationsproblems aufmerksam gemacht werden.

Erfahrungsgemäß wird es um so schwieriger ein technisches System zu entwickeln, das das Klassifikationsproblem zufriedenstellend löst

1. je *komplexer* die Objekte,
2. je *deformierter* die Objekte innerhalb einer Klasse und

3. je *variabler* die Objekte innerhalb einer Klasse sind.

Komplexität

Ein Komplex ist eine gegliederte Gesamtheit, die in ihre Teile bzw. Bereiche zerlegt werden kann. Besitzt ein Objekt Bestandteile, die selbst wieder Objekte sind, so wird auch dieses als komplexes Objekt bezeichnet. Solche Objektbestandteile können wiederum komplexe Bestandteile besitzen usw. bis nichtkomplexe Objekte entstehen, die sich nicht weiter zerlegen lassen. Dabei nimmt der Komplexitätsgrad der resultierenden Objekte nach jeder Zerlegung ab. Der Komplexitätsgrad eines Objekts hängt damit sowohl von der Anzahl der möglichen Weiterzerlegungsebenen als auch von der Gesamtanzahl der nichtkomplexen Objekte ab.

Der Körper eines Menschen (0. Ebene) z.B. besteht aus den einfacheren Bestandteilen Kopf, 2 Armen, 2 Beinen und dem Rumpf. Diese sind wiederum aus noch einfacheren Bestandteilen zusammengesetzt, wie z.B. der Arm (1. Ebene), der aus Oberarm, Unterarm und der Hand (2. Ebene) besteht. Die Hand lässt sich wiederum in den Handteller und die 5 Finger (3. Ebene) zerlegen und jeder Finger schließlich besteht aus 3 Gliedern (4. Ebene). Dies lässt sich für alle Objektbestandteile soweit fortsetzen, bis Objektteile erreicht sind, die sich nicht mehr in weitere (mit bloßem Auge sichtbare) Bestandteile zerlegen lassen. Diese elementaren Objektbestandteile sind dann nicht mehr komplex.

In dieser Arbeit wird unter einem komplexeren Objekt ein solches Objekt verstanden, das sowohl eine *nicht zu kleine* Anzahl elementarer, nichtkomplexer Bestandteile als auch eine *nicht zu kleine* Anzahl Zerlegungsebenen besitzt, ohne hier *nicht zu klein* näher zu definieren. Es soll an dieser Stelle vor allem herausgestellt werden, dass es in dieser Arbeit nicht darum geht Dreiecke, Quadrate, Kreise und ähnlich einfach strukturierte Objekte zum Gegenstand der Untersuchung zu machen, sondern komplexe Objekte zu analysieren, wie sie in der natürlichen Umwelt zuhauf und in künstlichen Umwelten auch nicht selten vorkommen. Exemplarisch sind in Abbildung 1.6 verschiedene Objekte von links nach rechts mit steigendem Komplexitätsgrad dargestellt. Hier werden ausschließlich die Schattenrisse von Objekten dargestellt, da die Arbeit sich, wie in Abschnitt 1.2 noch begründet wird, auf die Außenkonturen solcher Schattenrisse beschränkt. Interessant festzustellen ist, dass je komplexer die Objekte sind, es einem Menschen um so leichter fällt die Objekte zu erkennen. Das liegt sicherlich daran, dass die Objekte mehr Information beinhalten und damit auch mehr Teilinformation, die dieses Objekt charakterisieren. Trotzdem ist es z.Zt. einfacher ein technisches System zu bauen, das Objekte mit geringem Komplexitätsgrad auseinanderhalten kann.

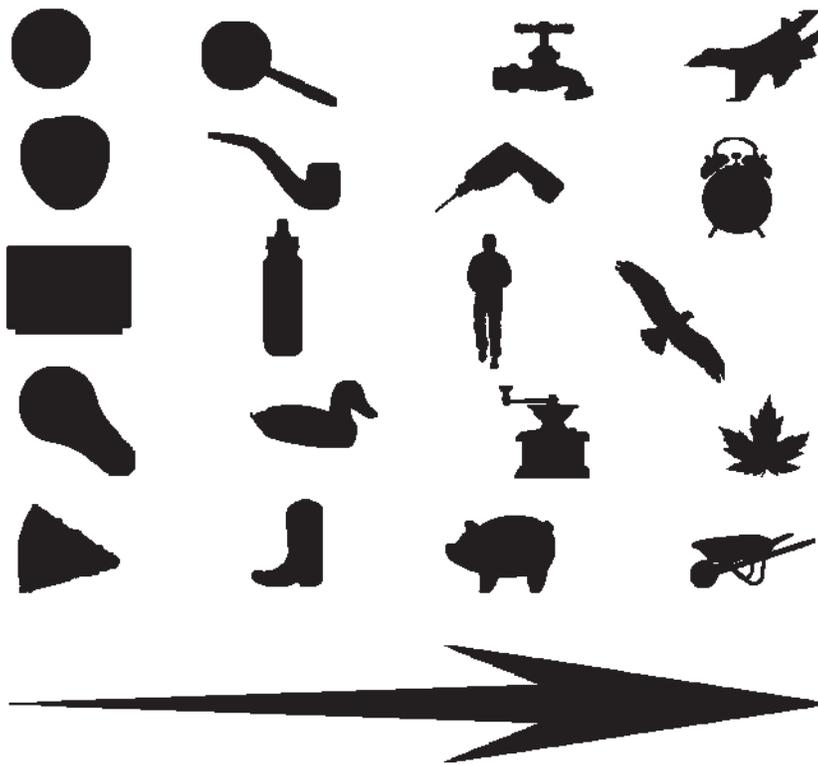


Abbildung 1.6: Schattenrisse verschiedener Objekte mit unterschiedlichem Komplexitätsgrad. Von links nach rechts steigt der Komplexitätsgrad der Objekte.

Deformation

In dieser Arbeit wird der Begriff der *Deformation* anders, d.h. allgemeiner, formuliert als normalerweise üblich. So soll unter dem Begriff *Deformation* hier nicht nur die Änderung der Form eines Objektes unter Druck verstanden werden. Vielmehr werden alle Einflüsse auf ein Objekt, die das Erscheinungsbild eines Objektes, bezogen auf einen Betrachter, verändern als *Deformation* aufgefasst. Im Einzelnen sollen folgende Einflüsse auf ein Objekt als *Deformation* verstanden werden. Zum einen sind dies *direkte physikalische Einflüsse* auf das Objekt:

- *Verformung*, d.h. die Änderung der Objektform unter Druck von außen (nicht vom Objekt selbst), wie z.B. durch Verbiegen, Abknicken, Eindellen, Ausbeulen o.ä. .
- *Zerstörung*, d.h. das Fehlen von Objektbestandteile durch Abreißen, Abschneiden, Durchbohren o.ä. .

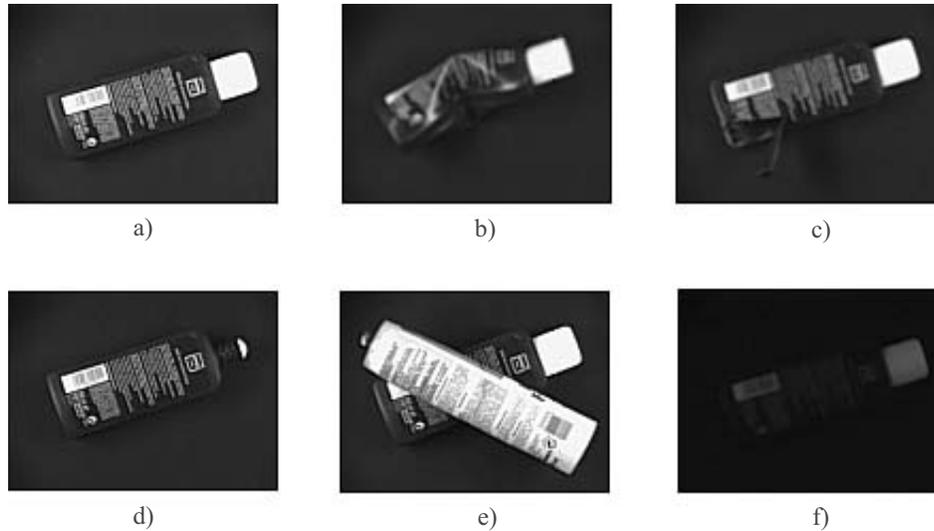


Abbildung 1.7: Beispiele für die Deformation durch direkte physikalische Einflüsse auf ein Objekt: a) Original einer Kunststoffflasche, b) Verformung, c) Zerstörung, d) eine andere Art der Zerstörung ist das Abnehmen bestimmter Objektbestandteile, wie z.B. dem Deckel der Flasche, e) Verdeckung, f) Verdunkelung.

- *Verdeckung*, d.h. das Verdecken bestimmter Objektbestandteile durch andere Objekte vom Beobachter aus gesehen; beispielsweise ein Baum vor einem Reh.
- *Verdunkelung*, d.h. keine ausreichende Beleuchtung bestimmter Objektbestandteile.
- *Eigenverformung*, d.h. die Bewegung bestimmter Bestandteile eines natürlichen oder künstlichen Objektes, die das Objekt selber ausübt. Beispiele sind das Bewegen eines Armes durch einen Menschen oder die Drehung des Rotors eines Hubschraubers.

Die zuvor genannten Einflüsse sind im allgemeinen nicht-*affine* Transformationen eines Objektes. Beispiele zu den direkten physikalischen Deformationen von Objekten, mit Ausnahme der Eigenverformung, sind in Abbildung 1.7 gezeigt. Abbildung 1.8 zeigt Beispiele zur Eigenverformung.

Zur Zeit erhältliche Standardkameras für die Bildverarbeitung bilden ein 3-dimensionales Objekt auf eine 2-dimensionale Projektionsebene, den Sensor, ab (3-dimensionale Bildverarbeitungskameras, die sowohl Tiefen als auch Farbinformationen liefern sind noch in der Entwicklung [SHX95], [Hu00], [HS99]). Dabei laufen von einem Ausgangspunkt, dem *Projektionszentrum*, *Projektionsstrahlen*

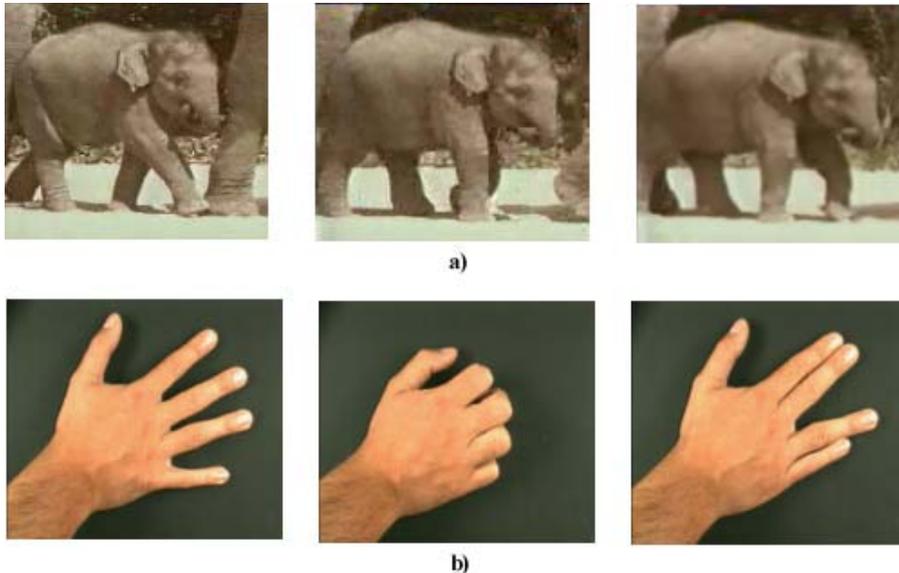


Abbildung 1.8: Beispiele für die Deformation eines Objektes durch Eigenverformung: a) Sequenz von der Laufbewegung eines Elefantenbabys. Hier verformt das Elefantenbaby vor allen Dingen die Stellung seiner Beine zu seinem Rumpf. b) Verschiedene Haltungen einer Hand. Hier ist es die Stellung der Finger zum Handteller, die die Verformung ausmacht.

durch jeden Punkt des Objektes. Die Schnittpunkte jedes *Projektionsstrahles* mit der *Projektionsebene* sind dann die mit den Objektpunkten korrespondierenden Bildpunkte. Dabei werden unterschieden:

- *Zentralprojektion*, bei der das Projektionszentrum einen endlichen Abstand von der Projektionsebene hat.
- *Parallelprojektion*, bei der das Projektionszentrum einen unendlichen Abstand von der Projektionsebene hat.

Das bedeutet, dass die Lage der Projektionsebene zu dem Objekt einen ganz entscheidenden Einfluss auf die Abbildung des Objektes hat. In diesem Zusammenhang spricht man von der Perspektive oder auch dem Blickwinkel auf ein Objekt. Diese, durch den Blickwinkel hervorgerufenen, verschiedenen Abbildungen desselben Objektes werden in dieser Arbeit als Deformation dieses Objektes aufgefasst. Beispiele zu den *perspektivischen Deformationen* sind in Abbildung 1.9 gezeigt.

Hinzu kommen folgende *technischen Einflüsse auf die Deformation*:

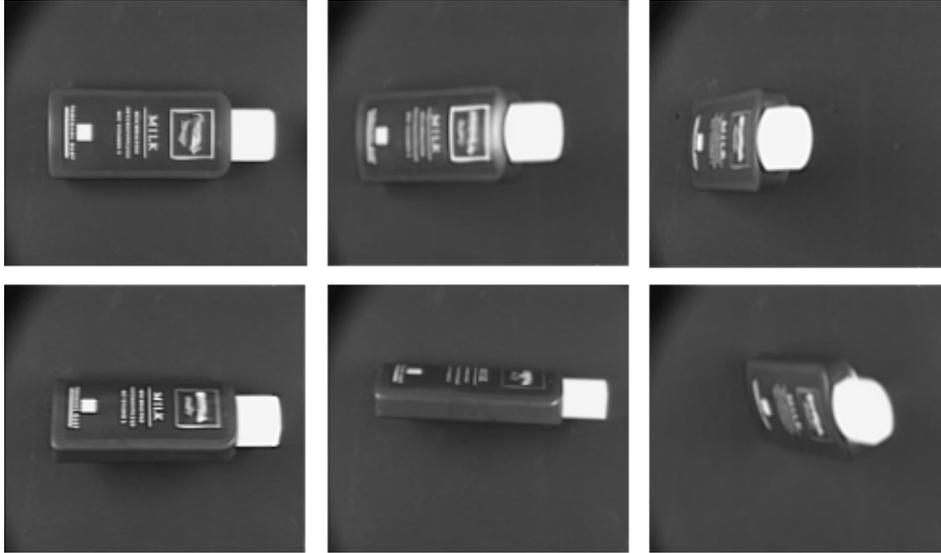


Abbildung 1.9: Beispiele für die perspektivischen Deformationen. Dieselbe Flasche aus verschiedenen Blickwinkeln.

- *Linsenverzeichnung*, d.h. das Objekt wird durch eine i. allg. nicht-ideale Linse optisch verzerrt abgebildet. Typische Verzeichnungsarten sind die kissenförmige oder die tonnenförmige Verzeichnung.
- *Unschärfe*, d.h. das Objekt wird auf die Bildebene unscharf abgebildet.
- *Rasterung* des Bildes, d.h. das Bild ist im Ortsbereich nicht mehr kontinuierlich, sondern diskret. Die Helligkeit wird über die Fläche eines jeden Sensorelementes integriert und beinhaltet somit die mittlere reflektierte Helligkeit der korrespondierenden Objektoberfläche.
- *Quantisierung* der mittleren Helligkeitswerte, die durch ein Sensorelement gemessen wurden, d.h. das Bild wird auch im Wertebereich diskretisiert. Diese Aufgabe übernimmt der A/D-Wandler des Framegrabbers.
- *Rauschen* des Bildsignals, d.h. Störungen, die das Bildsignal beim Transport beeinflussen.
- *Bandbreitenbegrenzung* des Bildsignals durch im allgemeinen nicht ideale elektrische Bauteile, die das Bildsignal beim Transport durchlaufen muss.

Diese technischen Einflüsse auf die Deformation des Objektbildes sind bei hoher Qualität der heutzutage zur Verfügung stehenden Bildaufnahmetechnik im

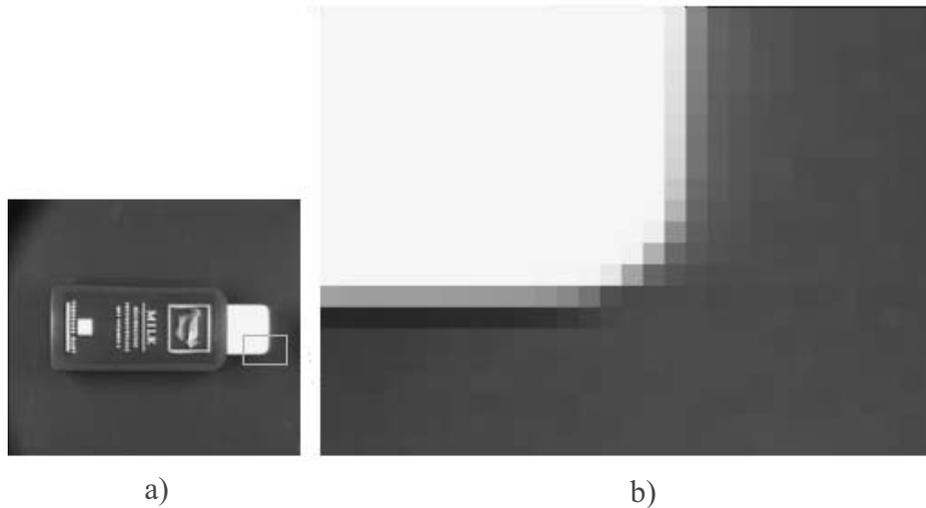


Abbildung 1.10: Beispiele für die technischen Deformationen. a) Bild einer Flasche mit markiertem rechteckigem Ausschnitt, der in Abb. 1.10b vergrößert dargestellt ist. b) Vergrößerung des in Abb. 1.10a markierten rechteckigen Bereiches der Flaschenkappe. Es sind deutlich die Rasterung und der Tiefpasseffekt der Elektronik in horizontaler Richtung beim Sprung von weiß nach schwarz zu sehen.

allgemeinen klein im Verhältnis zu den direkten physikalischen oder den perspektivischen Deformationen und vom ungeschulten menschlichen Auge kaum wahrnehmbar. Daher sind sie in dieser Arbeit weitgehend zu vernachlässigen. Abbildung 1.10 zeigt die beiden Effekte Rasterung und Bandbreitenbegrenzung. In Abbildung 1.10b ist der in Abbildung 1.10a mit einem grauen Rechteck markierte Ausschnitt vergrößert dargestellt. Die Rasterung ist deutlich zu erkennen. Zum anderen erkennt man hier deutlich ein Tiefpassverhalten der vorgeschalteten Elektronik. Der horizontale Übergang von der weißen Kappe zum dunklen Hintergrund wird über 4-5 Pixel verschmiert¹.

Die *technischen Deformationen* haben Einfluss auf die Segmentierung eines Objektes aus einem Bild. Unter Segmentierung versteht man in diesem Zusammenhang das Heraustrennen eines Objektes aus seinem Hintergrund. Dadurch, dass die Kanten des Objektes verwischt wurden und das Bild gerastert ist, kann

¹Der vor der Elektronik vorhandene Sprung fast über den gesamten Bereich des Farbspektrums von weiß nach schwarz, der einer sehr hohen Ortsfrequenz entspricht, wird geglättet. Diesen Effekt sieht man beim vertikalen Übergang von weiß nach schwarz nicht. Das liegt daran, dass das Bildsignal hintereinander aus den einzelnen Zeilen des Bildes zusammengesetzt ist. D.h. auftretende Sprünge in horizontaler Richtung sind bzgl. des eindimensionalen Bildsignals weitaus hochfrequenter als Sprünge in vertikaler Richtung ([Sch95], [Sto95])

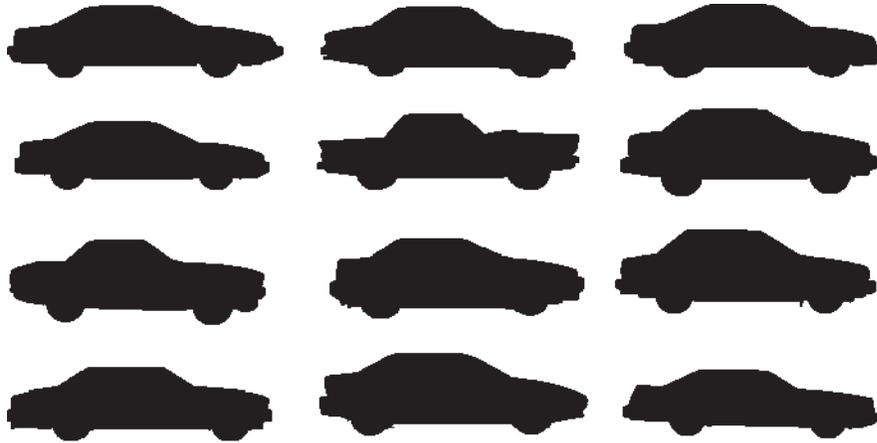


Abbildung 1.11: Schattenrisse verschiedener Autos aus dem gleichen Blickwinkel aufgenommen. Die Variabilität innerhalb dieser Klasse ist geringer als die Variabilität innerhalb der Fischklasse (Abbildung 1.12).

nicht mehr genau festgestellt werden, wo die Originalkante des Objektes verläuft. D.h. wiederum, dass der Originalkantenverlauf nur geschätzt werden kann. In diesem Zusammenhang spielen die Verfahren der Subpixelvermessung eine große Rolle ([DSAW98], [FS93]).

Als eine ganz spezielle Form von Deformation kann die **Variabilität** der Objekte innerhalb einer Klasse aufgefasst werden. Mit dem Begriff **Variabilität** der Objekte innerhalb einer Klasse soll zum Ausdruck gebracht werden, dass die Objekte einer Klasse sich von ihrer Art her mehr oder weniger stark unterscheiden können. So können beispielsweise die Objekte Amsel, Drossel und Fink einer Vogelklasse alle als Deformation des Objektes Star aufgefasst werden. D.h. die Objekte sind merkbar unterschiedlich, allerdings ist die Ähnlichkeit der Objekte untereinander noch so groß, dass diese Unterschiede nicht weiter ins Gewicht fallen. Das wiederum kann so interpretiert werden, dass diejenigen Merkmale, die diese Klasse charakterisieren, noch sowenig gestört sind, dass diese Störung nicht weiter auffällt. Die Art und Stärke der Störung ist dann das, was die Variabilität dieser Klasse ausmacht.

In Abbildung 1.11 und 1.12 sind jeweils eine Klasse mit den Schattenrissen von 12 Autos und 12 Fischen zu sehen, die jeweils aus dem gleichen Blickwinkel aufgenommen wurden. Die Klasse dieser Autos ist in sich weniger variabel, als die Klasse dieser Fische. Wahrscheinlich sind es die Flossen der Fische, die die relevanten Merkmale der Fischklasse sind. Diese sind trotz der relativ großen Variabilität innerhalb der Fischklasse so wenig gestört, dass sie weiterhin das Er-

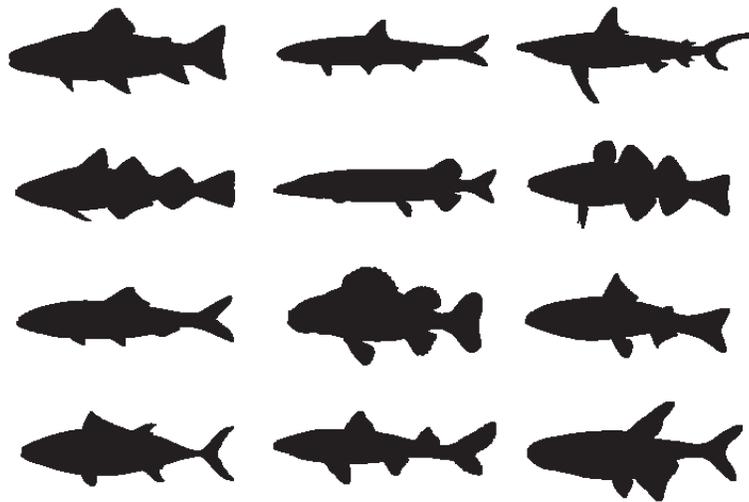


Abbildung 1.12: Schattenrisse verschiedener Fische aus dem gleichen Blickwinkel aufgenommen. Die Variabilität innerhalb dieser Klasse ist größer als die Variabilität innerhalb der Autoklasse (Abbildung 1.11).

scheinungsbild dieser Klasse prägen.

Je gestörter die Objekte einer Klasse im Sinne der Variabilität sind, desto schwieriger wird es erfahrungsgemäß ein Erkennungssystem zu entwickeln, das zufriedenstellend funktioniert. Es ist davon auszugehen, dass Variabilität und Komplexität insofern zusammenhängen, dass je größer die Variabilität einer Klasse ist, desto größer auch die mittlere Komplexität der Objekte in dieser Klasse ist. Oder mit anderen Worten ausgedrückt: komplexe Objekte erlauben mehr Variabilität.

Abschließend zu diesem Abschnitt soll zusammenfassend festgestellt werden, dass der hier geprägte Begriff der *Deformation* lineare als auch nicht-lineare Transformationen nicht weiter unterscheidet. So wird beispielsweise nicht versucht, die Projektionsdeformation zunächst zurückzurechnen, und dann mit der eigentlichen Analyse bzw. Erkennung eines Objektes zu beginnen. Auch die Lebewesen wussten zunächst nichts von Projektionen und derlei Dingen und waren trotzdem in der Lage in ihrer Welt Objekterkennung zu betreiben. So soll das in dieser Arbeit vorgestellte Konzept der *automatischen Merkmalsynthese* am Beispiel der Konturerkennung einen Beitrag zum Verständnis eines weitaus allgemeineren Konzepts zur Mustererkennung leisten:

Der berühmte englische Philosoph David Hume (1711-1776) konstatierte schon vor langer Zeit, dass wir Menschen nur deshalb (über)leben können, da sie sich Wiederholungen anpassen und methodisch Regeln und Gesetze einführen, welche

den zufälligen Wiederholungen eine rationale Grundlage geben sollen. Damit besteht unser Leben aus lauter Wahrscheinlichkeiten und die Realität ergibt sich grob gesagt als Summe dieser Wahrscheinlichkeiten. Beobachten wir, dass ein Ereignis unter ähnlichen Bedingungen immer einen ähnlichen Verlauf nimmt, so gehen wir davon aus, dass es sich in der Zukunft ähnlich zuträgt. Bei der Bestätigung unserer Erfahrung beim wiederholten Auftreten eines ähnlichen Ereignisses werden wir immer überzeugter von der Richtigkeit unserer Assoziationen bzgl. dieses Ereignisses. Wir werden uns sicher, dass ein Ereignis immer wieder einen ähnlichen Verlauf nimmt. Da sich uns ein Ereignis nicht völlig gleich, sondern immer nur ähnlich präsentiert, könnte man auch von deformierten Ereignissen sprechen. Bestimmte Merkmale dieses Ereignisses sind so signifikant, dass wir es als ein uns bekanntes Ereignis einstufen können.

Da sich uns unsere Umwelt, also auch einzelne Ereignisse, durch eine schier unendliche Anzahl an Zuständen präsentiert, ist es notwendig unsere Wahrnehmungen dieser Umwelt in angemessener Form zu komprimieren, so dass aber zukünftige Zustände der Umwelt noch richtig assoziiert werden können. In dieser Arbeit werden beispielhaft dafür lokale Konturabschnitte durch Zeigen von deformierten Varianten von Objekten aus verschiedenen Klassen daraufhin untersucht, inwieweit diese als alleinige Merkmale, also komprimierte Versionen von vollständigen Konturen, zur Vorhersage der Klassenzugehörigkeit von unbekanntem Objekten herangezogen werden können. Eventuell lassen sich auf diese Weise auch allgemeine Rückschlüsse für die Untersuchung anderer Mustererkennungsaufgaben ziehen.

1.2 Aufgabenstellung

Ziel dieser Arbeit ist es, ein automatisches Verfahren zu entwickeln, das als Eingangsgrößen die *deformierten* Objektkonturen (im Sinne des Unterabschnittes 1.1.2) einer Lernstichprobe aus einer Aufgabendomäne hat und als Ausgangsgrößen Konturabschnitte liefert, mit deren Hilfe ein Klassifikator Objektkonturen einer Teststichprobe der Aufgabendomäne ihren Klassen richtig zuordnet.

Um dieses Verfahren von üblichen Verfahren in der Objekterkennung abzugrenzen, sei zunächst kurz die zur Zeit typische Vorgehensweise in der Objekterkennung geschildert: Ein Trainer (Supervisor) gibt dem Klassifikationssystem *Merkmale* (Eigenschaften) der einzelnen Objektklassen vor, die seiner Meinung nach die Objektklassen gut trennen. Dies können beispielsweise das Längen-Breiten-Verhältnis, die Fläche oder der Umfang des Objektes sein. Die Anzahl der Merkmale, die ein Objekt beschreiben, ist dabei typischerweise immer gleich. Diese Merkmale werden dann aus der zugrundeliegenden Objektkontur extra-

hiert (\rightarrow *Merkmalsextraktion*) und sollten im \rightarrow *Merkmalsraum* einen möglichst kompakten Bereich einnehmen. Je kompakter dieser Bereich ist, desto einfacher und fehlerfreier ist die korrekte Klassifikation. Aus einer Vielzahl extrahierter Merkmale kann mit Hilfe von Verfahren der \rightarrow *Merkmalsselektion* (z.B. [BL97], [AB95]) diejenige Untermenge ausgewählt werden, die möglichst optimal für eine korrekte Klassifikation ist. Mit diesen Merkmalen wird ein sogenannter Klassifikator eingelernt, mit dem dann neu zu klassifizierende Objekte den einzelnen Klassen zugeordnet werden. Abbildung 1.13 verdeutlicht dieses Vorgehen.

Der Trainer nimmt also an, dass der Klassifikator die Objekte mit Hilfe der von ihm (dem Trainer) bestimmten Merkmale auseinanderhalten kann. Oft aus der Überzeugung heraus, dass er selbst die Klassen anhand dieser Merkmale unterscheidet. Dies weiß er aber nicht mit Sicherheit - noch weniger weiß er, dass der Klassifikator mit Hilfe dieser Merkmale dazu in der Lage ist. Bestätigt wird seine Annahme erst dann, wenn das Klassifikationsproblem in der Klassifikationsphase zufriedenstellende Ergebnisse liefert. Dieses \rightarrow *deduktive* Vorgehen, d.h. das Folgern eines Klassifikationsergebnisses aus bekannten vom Trainer vorgegebenen Regeln ist so richtig wie die Regeln selbst und führt i.allg. zu Ergebnissen, die in komplexeren Fällen nicht ausreichen. Deshalb wird in der Folge heuristisch und unterstützt durch Verfahren der Merkmalsselektion vorgegangen, d.h. es werden solange verschiedene Merkmalskombinationen und auch verschiedene Klassifikatortypen oder -modifikationen getestet, bis das gewünschte Ergebnis vorliegt oder nicht. Mit der Erfahrung des Trainers nimmt die Zeit für die heuristische Erprobung von Merkmalskombinationen und Klassifikatortypen ab, die Frage aber, warum es denn nun eigentlich genau diese Merkmale sind, die zu befriedigenden Klassifikationsergebnissen führen, bleibt oft unbeantwortet.

In diesem Zusammenhang wird erheblich viel Energie in die Erforschung verschiedener Klassifikatortypen gesteckt. Die Entwicklung von Verfahren zur *automatischen Merkmalsgenerierung* wird dagegen bisher sehr stiefmütterlich behandelt.

Aus diesem Grund wird in dieser Arbeit nach einer Möglichkeit gesucht charakteristische Merkmale automatisch zu generieren (*automatische Merkmalsynthese*², die die Objektklassen voneinander trennen. Im Gegensatz zum *deduktiven* Vorgehen werden hier beim \rightarrow *induktiven* Vorgehen Regeln, d.h. charakteristische Merkmale, aus Beispielen abgeleitet³. Dabei wird in der Objekterkennung

²Der Begriff der *Merkmalsynthese* kann als eine spezielle Form der *Merkmalsfusion* aufgefasst werden. Der Begriff der *Merkmalsfusion* wird aber meistens im Zusammenhang der Fusion von Merkmalen aus verschiedenen Informationsquellen (z.B. verschiedenen Sensoren oder die Informationen, die ein Sensor zu verschiedenen Zeitpunkten liefert) verwendet und sollte deshalb nicht mit der *Merkmalsynthese* verwechselt werden.

³Am Rande sei erwähnt, dass sich damit das zu entwickelnde Verfahren in ein Gebiet des *Maschinellen Lernens* (engl. *machine learning*), die sogenannte *automatische Merkmalsentdeckung*

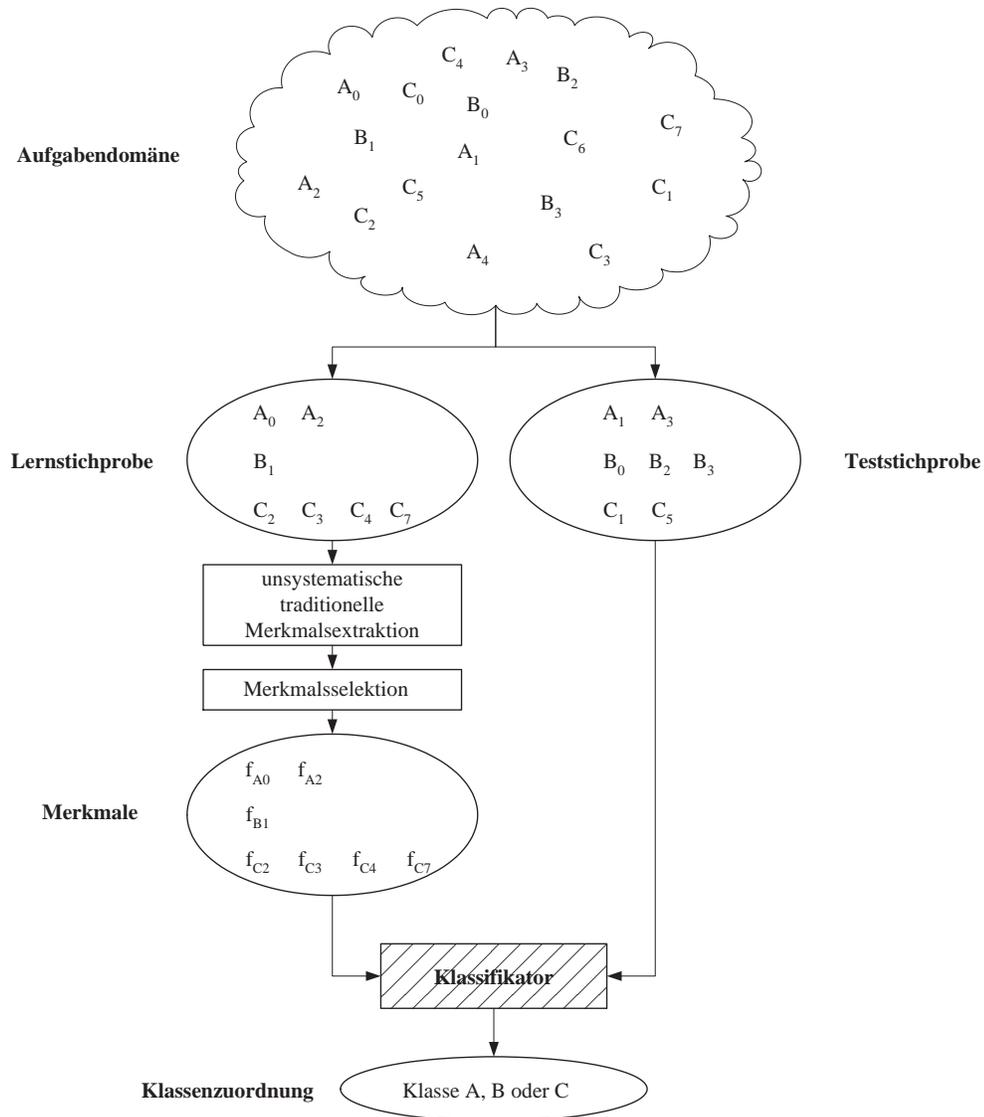


Abbildung 1.13: Übliches Prinzip der Objektklassifikation: In diesem Beispiel besteht die Aufgabendomäne aus 3 Klassen A, B und C. Jede Klasse besteht im allgemeinen aus einer verschieden großen Anzahl von Objekten. Jedes einzelne Objekt sei charakterisiert durch eine Klassenzugehörigkeit und einen Index. Aus der Aufgabendomäne werden bestimmte Objekte der Lernstichprobe und bestimmte Objekte der Teststichprobe zugeordnet. Dabei müssen nicht unbedingt alle Objekte der Aufgabendomäne Verwendung finden, die Lernstichprobe und die Teststichprobe sollten aber verschiedene Objekte beinhalten. Die Objekte in der Lernstichprobe dienen der Extraktion und Selektion einzulernender Merkmale. f_{A_0} steht beispielsweise für eine Menge von Merkmalen für das Objekt 0 aus Klasse A. Diese Merkmale dienen einem Klassifikator als Wissensbasis, der nacheinander die Objekte der Teststichprobe den Klassen A, B oder C zuordnet.

meistens vorausgesetzt, dass die Objekte von einem Trainer schon einzelnen Klassen zugeordnet sind. Das bedeutet, dass bezüglich der Klassen überwacht gelernt wird, bezüglich der Merkmale aber unüberwacht. Dieses Verfahren wird in Abbildung 1.14 kurz skizziert. Es unterscheidet sich dadurch von den herkömmlichen Verfahren, dass es die traditionelle Merkmalsextraktion und -selektion durch eine automatische Merkmalsgenerierung ersetzt.

Desweiteren wird untersucht inwieweit Verfahren entwickelt werden können, die auch unüberwacht bzgl. der Klassen lernen.

Eine Einschränkung soll sein, Objekte lokal zu untersuchen, d.h. es werden charakteristische Merkmale gesucht, die bestimmte lokale Objektbestandteile beschreiben. Der Vorteil besteht darin, dass so eine Toleranz gegenüber Deformationen wie beispielsweise Verdeckung, Zerstörung und andere bereits in Unterabschnitt 1.1.2 genannten Deformationen erreicht wird. Unterstützt wird dieser Ansatz dadurch, dass viele Lebewesen in der Lage sind, verdeckte oder zerstörte, d.h. alle nach Unterabschnitt 1.1.2 deformierten Objekte zu erkennen, solange genug signifikante Merkmale dieser Objekte zu sehen sind. Globale Merkmale, wie z.B. Gesamtoberfläche oder Kompaktheit, alleine reichen für diese Fähigkeit nicht aus, sondern ergänzen die Erkennungsleistung lediglich.

Eine weitere Einschränkung dieser Arbeit ist, dass ausschließlich die Außenkonturen von 2D-Objektabbildern untersucht werden. Der Vorteil besteht darin, dass dadurch eine relativ einfache und schnelle Analyse möglich wird. Das setzt natürlich für die Beurteilung der Ergebnisse voraus, dass die untersuchten Objektkonturen überhaupt erkennbar sind - erkennbar im Sinne eines menschlichen Beobachters, der schließlich die Leistungsfähigkeit des automatischen Systems beurteilt. Grundsätzlich sollten die Untersuchungsergebnisse aber auch prinzipielle Hinweise auf die Herangehensweise bei, nicht auf die Außenkonturen beschränkten, 2D-Objektabbildern liefern. Untersuchungen von u.a. Hayward [Hay98] und Liter et al. ([LTBK97], [LB96]) bestätigen, dass Schattenrisse bzw. deren Umrisse, die aus bestimmten Blickwinkeln aufgenommen wurden, ausreichend für eine Erkennung sind. Das könnte damit erklärt werden, wie Hoffman und Richards [HR84] erstmals veröffentlichten, dass es aus bestimmten Blickwinkeln besser gelingt eine teilbasierte Repräsentation, d.h. das Aufteilen eines komplexen Objektes in seine weniger komplexen Bestandteile durch Detek-

(*engl. automatic feature discovery*) einordnen lässt, in der versucht wird allgemeine Prinzipien zur automatischen Detektion aussagefähiger Merkmale herauszufinden (siehe z.B. [Faw90], [FU92], [Bel95]). Die meisten Arbeiten auf diesem Gebiet versuchen allerdings aus bestimmten Stellungen (Beispielen) bei Spielen (z.B. Schach, Tic-Tac-Toe, Dame, Othello) Regeln für das Gewinnen herzuleiten und sind damit nicht ohne weiteres auf die Objekterkennung anzuwenden. In [Bel95] werden genetische Algorithmen benutzt, um durch Testen verschiedener Kombinationen einer großen Anzahl von Spielstellungen der Spiele Dame und Tic-Tac-Toe Merkmale für das Gewinnen der Spiele herauszufinden.

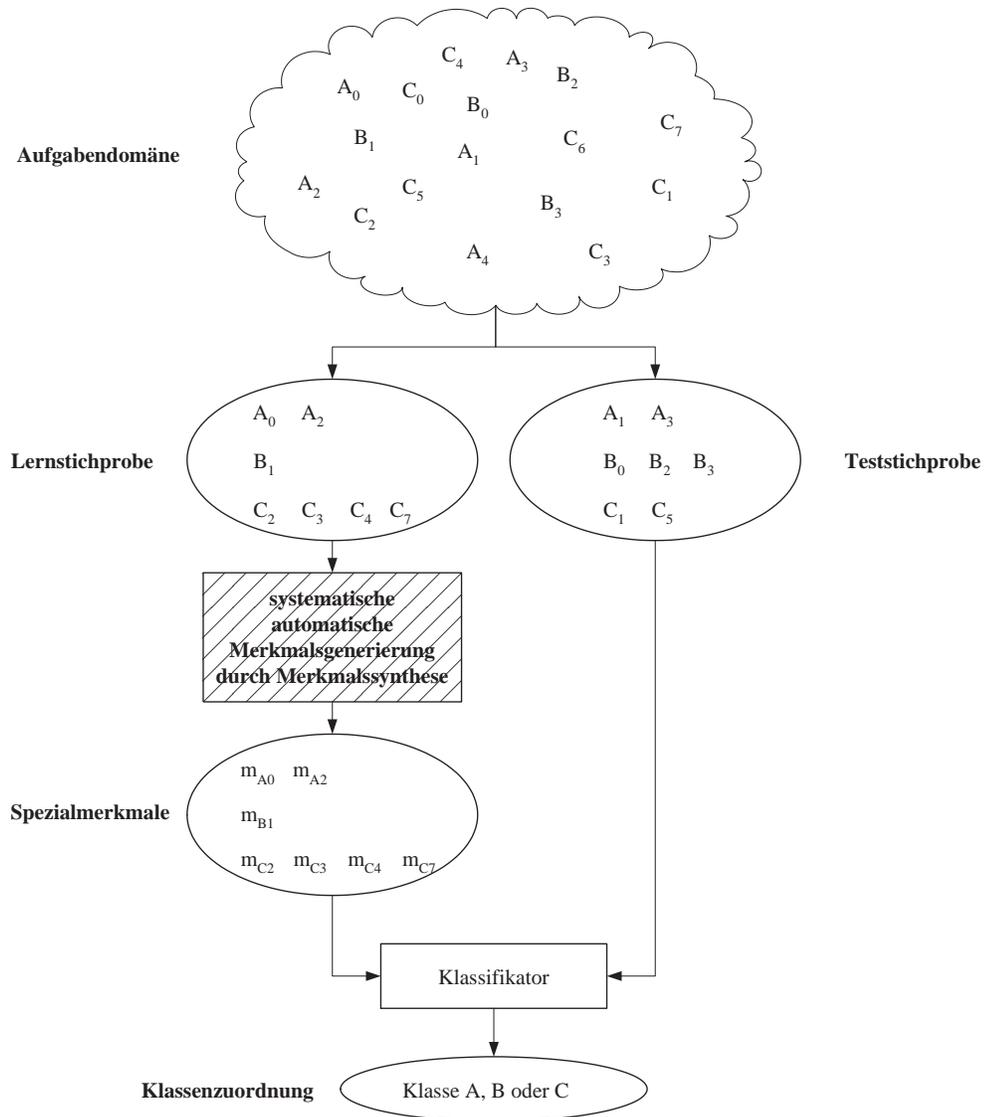


Abbildung 1.14: Prinzip des zu entwickelnden Verfahrens: In diesem Beispiel besteht die Aufgabendomäne aus 3 Klassen A, B und C. Jede Klasse besteht im allgemeinen aus einer verschieden großen Anzahl von Objekten. Jedes einzelne Objekt sei charakterisiert durch eine Klassenzugehörigkeit und einen Index. Aus der Aufgabendomäne werden bestimmte Objekte der Lernstichprobe und bestimmte Objekte der Teststichprobe zugeordnet. Dabei müssen nicht unbedingt alle Objekte der Aufgabendomäne Verwendung finden, die Lernstichprobe und die Teststichprobe sollten aber verschiedene Objekte beinhalten. Die Objekte in der Lernstichprobe dienen der automatischen Merkmalsgenerierung, so dass abstrakte Spezialmerkmale für die einzelnen Objekten erhalten werden. m_{A_0} steht beispielsweise für eine Menge von Spezialmerkmalen für das Objekt 0 aus Klasse A. Diese Spezialmerkmale dienen einem Klassifikator als Wissensbasis, der nacheinander die Objekte der Teststichprobe den Klassen A, B oder C zuordnet.

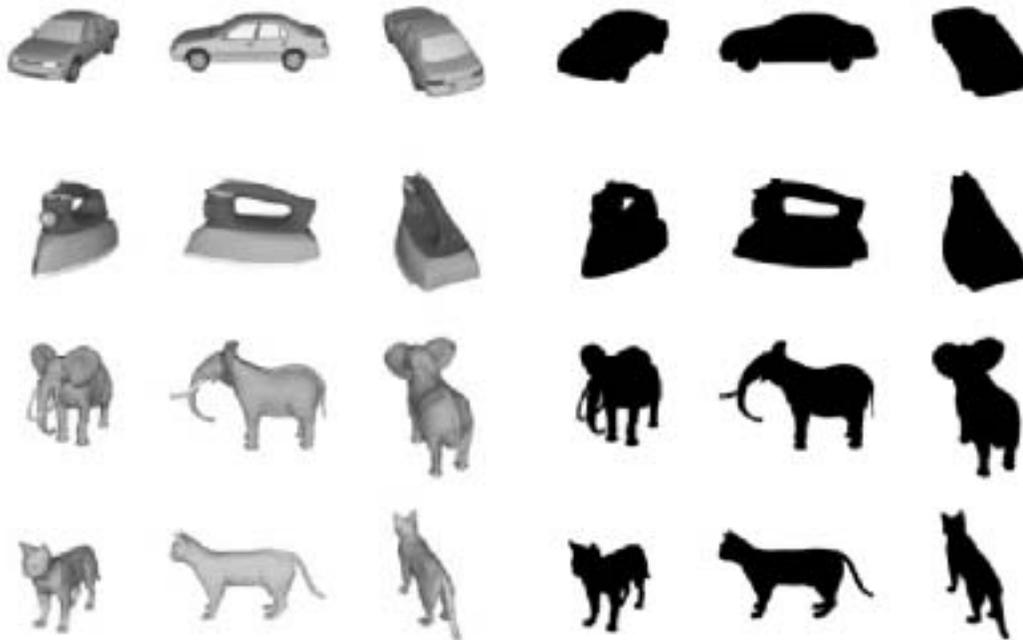


Abbildung 1.15: Schattierte Objekte und deren Schattenrisse aus verschiedenen Blickrichtungen. In der linken Hälfte die schattierten Objekte von schräg vorne, im Profil und von schräg hinten aufgenommen. In der rechten Hälfte sind die Schattenrisse der selben Objekte abgebildet. Aus Liter et al. [LTBK97].

tion von Krümmungsextrema, aus seinem Umriss abzuleiten. Aus anderen Blickwinkeln ist dies nicht mehr möglich, z.B. schräg von hinten. So könnte z.B. eine relativ kurze Auswölbung in der Außenkontur in Wirklichkeit ein viel längeres Objektteil sein, das sich in das Innere der Außenkontur erstreckt. Erst bei Schattierung des Objektes wird dies wieder erkennbar.

Liter et al. [LTBK97] untersuchten u.a. die in Abbildung 1.15 gezeigten Objekte aus einer Datenbank von Viewpoint Data Labs, die jeweils von schräg vorne, im Profil und von schräg hinten aufgenommen wurden. Es zeigte sich, dass nur die im Profil aufgenommenen Objekte sowohl schattiert als auch als Schattenriss gleich schnell von den Probanden erkannt wurden. Aus den anderen beiden Blickrichtungen erkannten die Probanden die Objekte mit Hilfe der Schattierungen wesentlich schneller.

Aus diesen Gründen werden in dieser Arbeit nur Außenkonturen von Objekten untersucht, die so aufgenommen wurden, dass möglichst gut Objektteile sichtbar sind. Eine Kuh soll deshalb beispielsweise von der Seite, eine Spinne allerdings von oben abgebildet sein. Neben den anderen Deformationsarten soll aber eine ge-

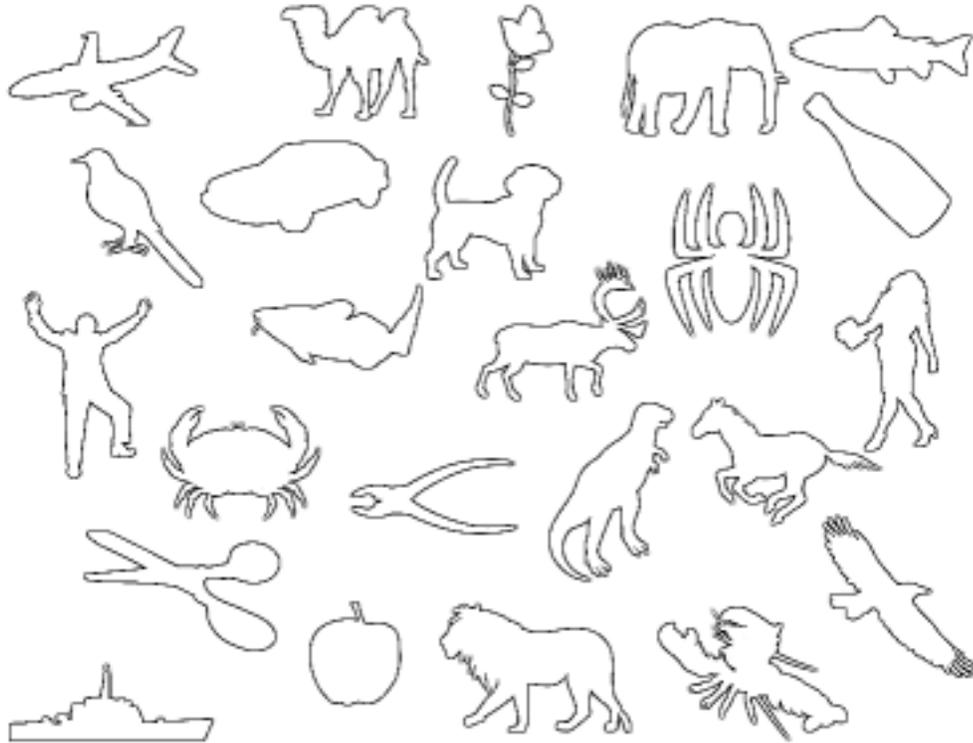


Abbildung 1.16: Beispiele für Konturen, die der Untersuchung dieser Arbeit zugrundeliegen. Die Konturen der Objekte, die aus diesem Blickwinkel aufgenommen wurden, sind für einen Menschen leicht erkennbar.

wisse Projektionsdeformation zulässig sein. Eine weitere Notwendigkeit für das Erkennen eines Objektes nur mit Hilfe der Kontur ist, dass die Objekte eine gewisse Komplexität haben. So ist z.B. die Klasse der Bälle von der Klasse der Apfelsinen sicherlich nur durch einen Experten basierend auf Außenkonturen unterscheidbar. Einige solche Beispielkonturen sind in Abb. 1.16 gezeigt.

1.3 Aufbau der Arbeit

Kap. 2 zeigt verschiedene Möglichkeiten 2D-Konturanalyseverfahren zu klassifizieren. Es werden beispielhaft verschiedene einfache Verfahren vorgestellt, um einen Einblick in die verschiedenen Prinzipien zu bekommen, die Konturanalyseverfahren zugrundeliegen können.

Da das in dieser Arbeit vorgestellte Verfahren auf einer lokalen Analyse de-

formierter Konturen basiert, ist Kap. 3 allein den 2D-Konturanalyseverfahren gewidmet, die mit nicht-linearen Deformationen umgehen und teilweise mit lokalen Anteilen arbeiten. Hier werden verschiedene aktuelle Verfahren vorgestellt. Allen diesen Verfahren liegt zugrunde, dass sie die Ähnlichkeit von 2 Konturen bestimmen. Eine weitergehende Analyse zur Bestimmung von lokalen Merkmalsgruppen, d.h. signifikanten Konturabschnitten, wird in diesen Arbeiten aber nicht durchgeführt und ist Kap. 4 vorbehalten.

Kap. 4 stellt ein neuartiges Verfahren zur automatischen Synthese von benachbarten Konturpunkten zu signifikanten Konturabschnitten vor. Dazu werden lokale Konturabschnitte der Konturen einer Lernstichprobe systematisch miteinander verglichen. Basierend auf dieser Analyse werden benachbarte Konturpunkte zu signifikanten Konturabschnitten synthetisiert, die die Konturen einer bestimmten Klasse von den Konturen aller anderen Klassen möglichst gut unterscheiden. Es wird ein hierarchischer Klassifikator vorgestellt, der mit Hilfe dieser signifikanten Konturabschnitte Objektkonturen einer Teststichprobe den Klassen zuordnet.

Kap. 5 diskutiert ausführlich experimentelle Ergebnisse, die mit dem neuartigen Verfahren erzielt wurden. Es werden verschiedene Aufgabendomänen mit unterschiedlichem Komplexitätsgrad untersucht. Dabei wird der Einfluss verschiedener Methodenparameter, aber auch verschiedener äußerer Parameter wie z.B. dem Blickwinkel untersucht. Hier wird auch eine Anwendung des neuen Ansatzes in der mobilen Robotik zur gleichzeitigen Erkennung und Navigation aufgezeigt. Ein intelligentes System wird mit Hilfe von Teilen der Methode in Kap. 4 in die Lage versetzt, ein implizites Modell seiner Umwelt selbständig aufzubauen.

Kap. 6 diskutiert die Ergebnisse und gibt einen Aus- und Überblick auf bereits begonnene, aber noch nicht abgeschlossene Forschungsarbeiten, die den Ansatz auf 2D-Grauwert- und 2D-Farbbilder erweitern soll.

Kapitel 2

Einordnung von 2D-Konturanalyseverfahren

Da sich diese Arbeit mit der Analyse der Außenkonturen von 2D-Objekten beschäftigt, soll in diesem Abschnitt ein allgemeiner Überblick über dieses Gebiet der Mustererkennung gegeben werden.

Die Analyse der Form bzw. des Umrisses (engl. *shape analysis*) ist in vielen Anwendungsbereichen von Interesse. In industriellen Identifikations- oder Qualitätssicherungsaufgabenstellungen ist der Umriss oft von genügend großem Informationsgehalt. Auch im Bereich des *image retrieval*, d.h. der Suche nach Bildern in einer Datenbank oder im gesamten Internet, die Bildern, die von einem Benutzer vorgegeben werden, ähnlich sind, werden z. Zt. massiv Konturanalyseverfahren untersucht (z.B. [LL], [MAK96], [MRMW98], [RHC99], [WCA99]). So sollen in diesem Zusammenhang Benutzer eine einfache Umrisszeichnung machen und das *image retrieval*-System sucht nach ähnlichen Objektumrissen. Aber auch die Erkennung von Buchstaben (z.B. [TJT96]) spielt in vielen industriellen und Dienstleistungsanwendungen eine große Rolle (→ *OCR*, → *ICR*).

Das Problem die Konturen erst aus realen Farb- bzw. Grauwertbildern auszu-segmentieren ist nicht Gegenstand der *shape analysis* und gehört in den Bereich der → *Segmentierungsverfahren* und wird deshalb auch in dieser Arbeit nicht weiter behandelt (siehe dazu z.B. [ZT98], [KWT87]).

Es existiert eine Vielzahl an Konturanalyseverfahren [Pav78], [Lon98], [VH99]. Eine Klassifizierung der einzelnen Verfahren kann nach verschiedenen Kriterien erfolgen, wobei die Verfahren untereinander aber immer bestimmte Eigenschaften teilen. Dies bestätigen auch verschiedene Klassifizierungen in der Literatur.

2.1 Klassifizierung nach Pavlidis

Zu Beginn sei eine der ersten systematisch durchgeführten Klassifizierungen von Konturanalyseverfahren genannt. So unterscheidet Pavlidis [Pav78] folgende Klassifizierungen von *shape analysis*-Verfahren:

- Verfahren, die auf der Außenkontur (*extern*) oder auf dem Bereich innerhalb der Außenkontur (*intern*) basieren.
- Verfahren, deren Analyseergebnis *numerisch* oder *nicht-numerisch* sind.
- Verfahren, die *informationserhaltend* oder *nicht-informationserhaltend* sind.

Ein bekanntes Beispiel für ein *externes* Verfahren ist die *Fourieranalyse* der Krümmungsfunktion einer Kontur (z.B. [ZR72], [Ott95]). Ein bekanntes *internes* Verfahren ist die von Blum [Blu67] eingeführte sogenannte *medial axis transform* (MAT). Die Außenkontur eines Objektes wird solange "zusammengezogen" bis eine Strichzeichnung, auch Skelett genannt, übrigbleibt, die möglichst in der Mitte zwischen gegenüberliegenden Konturabschnitten liegt (siehe Abb. 2.1).

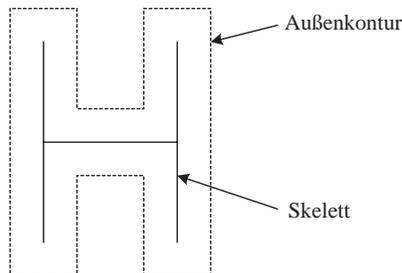


Abbildung 2.1: Beispiel für die MAT (Transformation auf mittige Achsen). Gestrichelt dargestellt ist die Außenkontur des Buchstaben H. Durchgezogen dargestellt sieht man die Strichzeichnung oder auch das sogenannte Skelett des Buchstabens H nach der MAT.

Verfahren werden als *numerisch* bezeichnet, wenn das Ergebnis ihrer Analyse eine oder mehrere Zahlen sind. So ist z.B. die sog. *Kompaktheit* ein Zahlenwert, der sich aus der Länge der Außenkontur (Umfang) und der eingeschlossenen Fläche berechnet:

$$Kompaktheit = \frac{Umfang^2}{Fläche}. \quad (2.1)$$

Ein Beispiel für ein *nicht-numerisches* Verfahren wurde schon vorgestellt. Die MAT ist ein solches, denn sie produziert ein neues Bild - das Skelett - und keine Zahl.

Informationserhaltende Verfahren wie die Fourieranalyse lassen eine Rekonstruktion der Kontur aus ihrer neuen Beschreibung zu. Eine *nicht informationserhaltende* Methode ist die schon genannte Kompaktheit (Gl. 2.1), denn viele sehr unterschiedliche Außenkonturen können die gleiche Kompaktheit haben, d.h. es ist nicht möglich, die Originalkontur nur aus der Kompaktheit wiederherzustellen.

Klassifiziert man nun z.B. die Fourieranalyse, die Kompaktheit und die MAT nach Pavlidis so ergibt sich folgendes:

- Fourieranalyse: extern, numerisch und informationserhaltend
- Kompaktheit: extern/intern, numerisch und nicht-informationserhaltend
- MAT: intern, nicht-numerisch, nicht-informationserhaltend

2.2 Globale und lokale Verfahren

Eine andere Eigenschaft von Konturanalysemethoden ist, dass sie entweder *global*, *lokal* oder beides sind. *Globale* Methoden analysieren die Kontur als Ganzes. Sie unterscheiden dabei nicht, dass einzelne Konturabschnitte einer Kontur aussagekräftiger sein können als andere und "mitteln" den Beitrag dieser Konturabschnitte heraus. *Lokale* Methoden hingegen berücksichtigen, dass Konturen an bestimmten Stellen interessanter für eine zu lösende Aufgabe sein können als an anderen Stellen.

Es gibt einige einfache globale Analyseverfahren zu denen auch die Kompaktheit (Gl. 2.1) gehört, die i. allg. nur mit wenig komplexen Objekten in sich stark unterscheidenden Klassen im Bereich der Objekterkennung vernünftig funktionieren und von denen im Folgenden einige aufgezählt werden:

- *Umfang*, d.h. die Bogenlänge der Kontur
- *Flächeninhalt*, der von der Kontur eingeschlossen ist
- *Kompaktheit*, manchmal auch als *Rundheit* bezeichnet (Gl. 2.1)
- *Exzentrizität* ist das Verhältnis von minimalem zu maximalem \rightarrow *polarem Abstand*
- *Umgebendes Rechteck* (engl. *bounding box*) ist ein Rechteck, das gerade groß genug ist, um die Kontur komplett zu beinhalten

Um zu verdeutlichen, welche unterschiedlichen Prinzipien komplexeren globalen Konturanalyseverfahren zugrunde liegen, werden aus der Vielzahl solcher Verfahren stellvertretend die sogenannten *Formzahlen* (engl. *shape numbers*) (z.B. [BB82], [BG80], [Chu95]), *Momente* (z.B. [Hu62], [PR92]) und die *Fourierdeskriptoren* (z.B. [PF86], [ZR72]) in den folgenden 3 Unterabschnitten vorgestellt.

2.2.1 Formzahlen (engl. shape numbers)

Auf dem Pixelraster besteht eine geschlossene Kontur aus einer endlichen Anzahl n an Konturpunkten. In Abb. 2.2a ist eine solche Kontur abgebildet. Sie besitzt $n = 20$ Konturpunkte. Diese sind markiert mit den zugehörigen Konturpunktindices. Diese Indices finden sich in der ersten Zeile der Abb. 2.2f wieder. Bei der Erzeugung der Konturpunkte wurde eine sog. *4er-Nachbarschaft* zugrundegelegt. Als 4er-Nachbarn eines bestimmten Pixels bezeichnet man die Pixel östlich, nördlich, westlich und südlich dieses bestimmten Pixels. Werden die Pixel, die diagonal zu dem bestimmten Pixel liegen, mit hinzugenommen, so wird von den 8er-Nachbarn gesprochen. Der Übergang von einem Konturpunkt zum benachbarten Konturpunkt kann dann auch diagonal erfolgen.

Eine Kontur kann nicht nur durch die Koordinaten der einzelnen Konturpunkte beschrieben werden, sondern auch durch einen sogenannten *Kettencode* (engl. *chain code*). Dieser gibt ausschließlich die Richtung an, in der sich der nächste Konturpunkt in der Konturpunktsequenz befindet. Die Kontur ist jetzt translationsinvariant dargestellt. Werden alle Konturpunkte um dieselbe Distanz in dieselbe Richtung verschoben, bleibt ihr Kettencode gleich. Die Konturpunkte haben sich im Gegensatz dazu aber geändert, sind also translationsvariant. In Abb. 2.2b ist ein solcher Kettencode für die 4er-Nachbarschaft definiert. 0 steht für östlich, 1 für nördlich, 2 für westlich und 3 für südlich. In Abb. 2.2c ist das Ergebnis einer solchen Kodierung für die Beispielkontur dargestellt. Die Konturpunktindices sind durch den entsprechenden Kettencode ersetzt. Dieser findet sich in der zweiten Zeile der Abb. 2.2f wieder.

Mit Hilfe des Kettencodes lässt sich sehr einfach eine diskrete Krümmungsfunktion der Kontur erzeugen. In Abb. 2.2d sind alle möglichen Kombinationen von 2 aufeinanderfolgenden Richtungen zum nächsten Konturpunkt für eine 4er-Nachbarschaft abgebildet. Dabei sind 3 Krümmungsfälle zu unterscheiden, die folgendermaßen kodiert werden: 0 für Konvexitäten, 1 für Geradigkeiten und 2 für Konkavitäten. In Abb. 2.2e ist das Ergebnis einer solchen diskreten Krümmungsfunktion für die Beispielkontur zu sehen. Der Kettencode ist jetzt durch den entsprechenden Krümmungscode ersetzt. Dieser findet sich in der dritten Zeile der Abb. 2.2f wieder.

Die *shape number* entsteht nun aus dem Krümmungscode, indem dieser als Zahl aufgefasst wird. Die einzelnen Ziffern der Zahl werden solange hinten abgetrennt und vorne wieder hinzugefügt (rotiert) bis die minimalst mögliche Zahl entsteht. Diese entstehende Zahl ist dann die gesuchte *shape number*. Im Beispiel entsteht so folgende *shape number*:

$$0 \cdot 10^{19} + 1 \cdot 10^{18} + 0 \cdot 10^{17} + \dots + 1 \cdot 10^2 + 1 \cdot 10^1 + 2 \cdot 10^0. \quad (2.2)$$

Das ganze hat nun folgenden Sinn. Um 2 Konturen miteinander zu verglei-

chen, werden diese so auf einem Pixelraster skaliert, dass sie die gleiche Anzahl an Konturpunkten n haben. Es wird dann von einer *shape number* der Ordnung n gesprochen. Dies ist allerdings nicht für alle n zu erfüllen und es werden deshalb einfach nur solche Ordnungen n zugelassen, bei denen die zu vergleichenden Konturen die gleiche Anzahl an Konturpunkten haben. Um rotationsinvariant zu werden, d.h. die Ähnlichkeit zweier Objekte soll nicht davon abhängen, wie sie gegeneinander rotiert sind, wird der Krümmungscode wie oben gezeigt rotiert. Mit dem Krümmungscode aus Abb. 2.2d werden die Konturen bzgl. ihrer *shape number* also so gedreht, dass die jeweils konvexesten Konturabschnitte beider Konturen übereinanderliegen.

Die Ähnlichkeit zweier Konturen ist nun die größte Ordnung n bei der ihre beiden *shape numbers* identisch sind. Je größer n desto mehr Feinheiten haben die 2 Konturen gemeinsam.

2.2.2 Momente

Mit *Momenten* werden i. allg. numerische Größen an einer bestimmten Stelle mit einem bestimmten Abstand von einem Referenzpunkt oder einer Referenzachse beschrieben. Sie werden beispielsweise in der Mechanik benutzt, um Körper mit Hilfe ihrer räumlichen Masseverteilung zu charakterisieren. In der Bildverarbeitung lässt sich sehr einfach eine Analogie zur Mechanik herstellen, indem ein Bild als 2D-Körper aufgefasst wird. Jedem Pixel dieses Bildes wird eine Masse, gemäß dem Grauwert an dem Pixelort, zugeordnet.

Das zweidimensionale Carthesische Moment m_{pq}^* der Ordnung $p + q$ ist im Kontinuierlichen für eine Funktion $f(x, y)$ folgendermaßen definiert:

$$m_{pq}^* = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy. \quad (2.3)$$

Aus Gl. (2.3) folgt das Moment für ein diskretisiertes Bild $g(x, y)$, das aus M Pixeln in y -Richtung und N Pixeln in x -Richtung besteht:

$$m_{pq} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x^p y^q g(x, y). \quad (2.4)$$

Ein Moment gewichtet also einen Funktionswert mit einem von p und q abhängenden Produkt seiner zugehörigen Koordinaten, also mit einem gewichteten Ort.

Da es an dieser Stelle um die Analyse von Konturen geht, genügen 2 Werte, um die Kontur zu beschreiben:

$$g(x, y) = \begin{cases} 0 & \text{, dort wo sich kein Konturpunkt befindet} \\ 1 & \text{, dort wo sich ein Konturpunkt befindet.} \end{cases} \quad (2.5)$$

Die Momente 0., 1. und 2. Ordnung repräsentieren fundamentale geometrische Eigenschaften von Körpern und somit auch von Konturen.

Das 0.te Moment

$$m_{00} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} g(x, y) \quad (2.6)$$

repräsentiert nichts anderes als die Anzahl der Konturpunkte.

Die 1.ten Momente ergeben sich aus Gl. (2.4) wie folgt:

$$m_{10} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x g(x, y), \quad (2.7)$$

$$m_{01} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} y g(x, y). \quad (2.8)$$

Da nur an den Konturpunkten gilt: $g(x, y) = 1$, ergibt m_{10} die Gesamtsumme der x -Koordinaten der Konturpunkte der Kontur und m_{01} ergibt die Gesamtsumme der y -Koordinaten der Kontur. Im Zusammenhang mit dem 0.ten Moment m_{00} ergibt sich so der Schwerpunkt der Kontur, der nichts anderes als der Mittelwert der Koordinatensummen in x - und y -Richtung ist:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad (2.9)$$

$$\bar{y} = \frac{m_{01}}{m_{00}}. \quad (2.10)$$

In dem Fall, in dem sich zwei Konturen nur durch einen konstanten Offset auf alle x - und einen konstanten Offset auf alle y -Koordinaten voneinander unterscheiden, sind die beiden Konturen lediglich gegeneinander verschoben. Würde man diese beiden Konturen jetzt ausschließlich basierend auf ihren Konturpunktkoordinaten miteinander vergleichen, so würden beide Konturen, je nach Verschiebung mehr oder weniger ungleich sein. In den allermeisten Fällen ist dieses Ergebnis aber nicht erwünscht. Aus diesem Grunde wird vor einer weitergehenden Analyse zunächst die eine Kontur auf die andere Kontur verschoben. Hierzu bieten sich die Schwerpunkte der beiden Konturen an. Dazu wird zunächst die Differenz der beiden Schwerpunkte festgestellt und dann alle Konturpunkte bzgl. dieser Differenz verschoben, so dass die Schwerpunkte der beiden Konturen übereinanderliegen. Eine andere Möglichkeit ist beide Konturen so zu verschieben, dass ihre Schwerpunkte im Koordinatenursprung liegen. In diesem Fall wird von den *zentrierten Momenten* gesprochen. Diese sind dann folgendermaßen definiert:

$$\mu_{pq} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q g(x, y). \quad (2.11)$$

Im Fall der zentrierten Momente sind die 2. Momente

$$\mu_{20} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} (x - \bar{x})^2 g(x, y). \quad (2.12)$$

$$\mu_{02} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} (y - \bar{y})^2 g(x, y). \quad (2.13)$$

$$\mu_{11} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} (x - \bar{x})(y - \bar{y}) g(x, y). \quad (2.14)$$

einfacher zu interpretieren. μ_{20} ist das *zentrierte axiale Flächenträgheitsmoment* und gibt das Trägheitsmoment der Kontur bzgl. der x -Achse an. μ_{02} ist ebenfalls ein *zentriertes axiales Flächenträgheitsmoment* und gibt das Trägheitsmoment der Kontur bzgl. der y -Achse an. μ_{11} ist das sogenannte *zentrierte gemischte Flächenträgheitsmoment*, das sich sowohl auf die x - als auch auf die y -Achse bezieht.

In der Mechanik ist das Massenträgheitsmoment folgendermaßen definiert:

$$J = \sum_{i=0}^{A-1} r_i^2 \Delta m_i \quad (2.15)$$

Das Massenträgheitsmoment eines Körpers bezüglich einer Drehachse ergibt sich also aus der Gewichtung von A Masselementen Δm_i mit dem jeweils zu diesem Masseilchen gehörenden quadratischen Abstand von der Drehachse. Das bedeutet aber, dass

$$\begin{aligned} J_K &= \mu_{20} + \mu_{02} \\ &= \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} (x - \bar{x})^2 g(x, y) + \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} (y - \bar{y})^2 g(x, y) \\ &= \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} \left((x - \bar{x})^2 + (y - \bar{y})^2 \right) g(x, y) \end{aligned} \quad (2.16)$$

das Trägheitsmoment der Kontur bezüglich der Achse, die die xy -Ebene senkrecht im Schwerpunkt durchstößt, ist. Denn der Term $(x - \bar{x})^2 + (y - \bar{y})^2$ ist ja nichts anderes als der quadratische Abstand eines Konturpunktes vom Schwerpunkt.

Aus den zentrierten Flächenträgheitsmomenten μ_{20} , μ_{02} und μ_{11} lassen sich noch andere nützliche Objektmerkmale bestimmen, die an dieser Stelle nur genannt, aber nicht weiter bewiesen werden.

Die *Hauptachsen* sind diejenigen im Schwerpunkt aufeinander senkrecht stehenden Achsen einer Kontur, durch die minimale bzw. maximale zentrierte axiale Flächenträgheitsmomente μ_{20} , μ_{02} erhalten werden, wenn die Kontur bzgl. dieser Achsen um ihren Schwerpunkt so gedreht wird, dass diese Achsen mit der x -

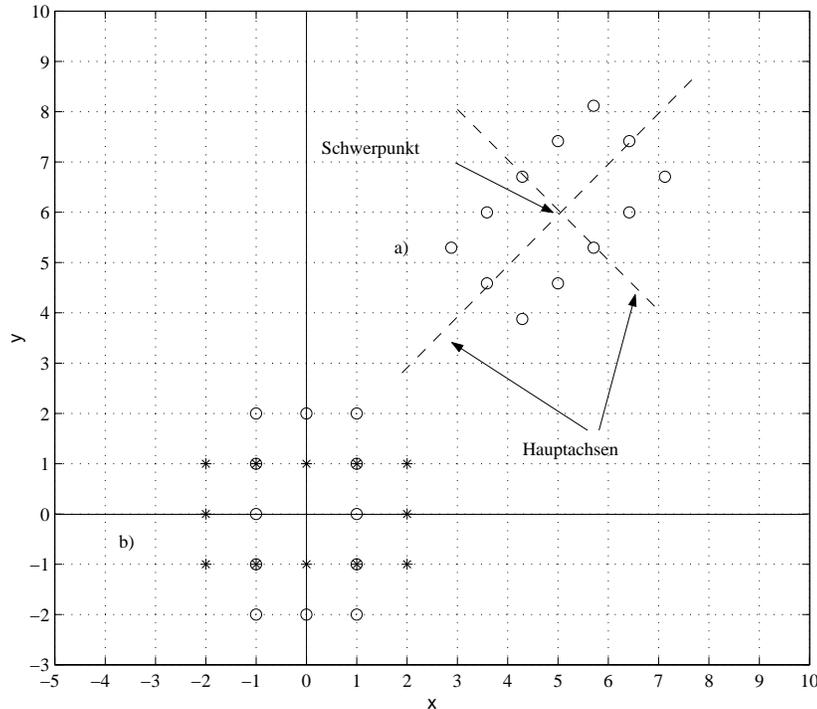


Abbildung 2.3: Hauptachsen einer Kontur. a) Kontur an ihrem Ursprungsort mit Schwerpunkt und Hauptachsen. b) Kontur aus a) in den Schwerpunkt verschoben und um den Winkel der Hauptachsen zur x -Achse rotiert. Hier sind 2 Fälle zu unterscheiden: Die mit \circ gekennzeichnete Kontur wurde so gedreht, dass ihre zentrierten axialen Flächenträgheitsmomente μ_{20} maximal und μ_{02} minimal sind. Bei der mit $*$ gekennzeichneten Kontur ist dies genau andersherum. In beiden Fällen ergibt sich ein zentriertes gemischtes Flächenträgheitsmoment μ_{11} von 0.

und y -Achse übereinstimmen (siehe Abb. 2.3). Ist μ_{20} maximal, so ist μ_{02} minimal und umgekehrt. Das zentrierte gemischte Flächenträgheitsmoment μ_{11} wird in diesen beiden Fällen 0. Diese minimalen und maximalen Momente können als Merkmale der Kontur ebenso benutzt werden, wie J_K aus Gl. 2.16. Zudem kann die Richtung dieser Hauptachsen als Orientierung der Kontur aufgefasst und zum Ausrichten der Kontur herangezogen werden. Der Orientierungswinkel θ ist durch die Beziehung

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (2.17)$$

gegeben. θ gibt denjenigen Winkel im Bereich $-\pi/4 \leq \theta \leq \pi/4$ an, der zu der Hauptachse gehört, die einen kleineren Winkel zur x -Achse hat. Um die Orientie-

zung einer bestimmten Hauptachse zu erhalten, müssen zusätzlich die einzelnen Terme μ_{11} und $(\mu_{20} - \mu_{02})$ ausgewertet werden (siehe dazu z.B. [PR92], [Wer97]).

Es ist möglich eine Ellipse (engl. *image ellipse*) aus den Momenten 1. und 2. Ordnung zu berechnen, so dass die von der Ellipse eingeschlossene Fläche (= Summe der Pixel) und die 2. Momente dieser Ellipse gleich der von der Kontur eingeschlossenen Fläche (= Summe der Pixel) sind. Die große Halbachse a und die kleine Halbachse b errechnet sich wie folgt:

$$a = \sqrt{\frac{2(\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2})}{\mu_{00}}} \quad (2.18)$$

$$b = \sqrt{\frac{2(\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2})}{\mu_{00}}}. \quad (2.19)$$

Auch a und b lassen sich als Merkmale einer Kontur benutzen.

Es wurde weiter vorne schon erwähnt, dass einem Pixel die Masse 1 zugeordnet werden kann, wenn an dieser Koordinate im Bild das Objekt vorhanden ist. Eine Außenkontur schließt eine Fläche ein, die zum Objekt gehört. So entspricht die Anzahl der Konturpixel und der eingeschlossenen Pixel der Masse dieses Objektes. Es lässt sich nun ein sog. Drehradius (engl. *radius of gyration*) bzgl. des Ursprungs berechnen. Dies ist der Radius eines Kreises, dessen Mittelpunkt im Ursprung liegt und auf den die gesamte Masse des Objektes konzentriert ist, ohne dass sich die zentrierten 2. Momente bzgl. der ursprünglichen Kontur ändern. Der Radius dieses Kreises ist ein weiteres auf den Momenten basierendes Merkmal einer Kontur und lässt sich folgendermaßen berechnen:

$$R = \frac{\mu_{20} + \mu_{02}}{\mu_{00}}. \quad (2.20)$$

Mit den höherwertigen Momenten lassen sich noch weitere Merkmale berechnen, die eine globale geometrische Interpretation der Kontur zulassen, wie z.B. Symmetrieeigenschaften. Damit bieten sie einen sehr systematischen und auch anschaulichen Weg, Konturen zu analysieren. Darüberhinaus haben Momente teilweise sehr schöne Invarianzeigenschaften gegenüber \rightarrow *affinen Transformationen* wie in Abschnitt 2.3 noch gezeigt wird.

2.2.3 Fourierdeskriptoren

Konturen von 2D-Objekten lassen sich, wie später noch gezeigt wird, relativ einfach als eindimensionale Funktionen darstellen. Nach *Fourier* [Fou22] lässt sich jede eindeutige periodische Funktion

$$y = f(x) = f(x + kT) \quad \text{für alle } k \in \mathbb{N} \quad (2.21)$$

mit der Periode T , die stückweise monoton und stetig ist, eindeutig als *Fourierreihe*

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos\left(\frac{2\pi kx}{T}\right) + b_k \sin\left(\frac{2\pi kx}{T}\right) \right) \quad \text{für alle } k \in \mathbb{N} \quad (2.22)$$

darstellen. Die Koeffizienten

$$a_k = \frac{2}{T} \int_0^T f(x) \cos\left(\frac{2\pi kx}{T}\right) dx \quad (2.23)$$

$$b_k = \frac{2}{T} \int_0^T f(x) \sin\left(\frac{2\pi kx}{T}\right) dx \quad (2.24)$$

werden im Zusammenhang mit der Konturanalyse als die *Fourierdeskriptoren* bezeichnet und $a_0/2$ ist dabei speziell der Mittelwert der Funktion $f(x)$. Die *Fourierdeskriptoren* sind also die Amplituden eines Spektrums von diskreten Frequenzen $f_k = k/T$. Eine Fourieranalyse darf deshalb durchgeführt werden, da eine *geschlossene* Kontur nichts anderes als eine periodische Funktion darstellt. Nach dem letzten Konturpunkt beginnt die periodische Kontur wieder mit dem ersten Konturpunkt usw..

Es gibt verschiedene Möglichkeiten Konturen als eindimensionale Funktionen darzustellen.

Eine mögliche Variante ist, eine Kontur in äquidistanten Winkelschritten abzutasten (siehe Abb. 2.4). Dabei gehen Strahlen vom Schwerpunkt S der Kontur aus, die äquidistante Winkel einschließen. Die Schnittpunkte dieser Strahlen mit der Kontur sind dann die Abtastpunkte P_i^* der Kontur. Ist die Kontur nicht kontinuierlich, sondern diskret, so muss zwischen den einzelnen Konturpunkten interpoliert werden. Ist die Auflösung des der Kontur zugrunde liegenden Bildes hoch genug gewählt, so kann zwischen den Konturpunkten ohne größeren Fehler linear interpoliert werden.

Die Funktion

$$r(\alpha) = |P_i^*(\alpha) - S| \quad (2.25)$$

kann dann einer *diskreten* Fourieranalyse (nach Carl D. T. Runge (1856-1927)) unterzogen werden. Die Fourierdeskriptoren werden in diesem Fall oft als *polare Fourierdeskriptoren* bezeichnet. Der Nachteil dieser Methode ist, dass es bei komplexeren Konturen vorkommen kann, dass die vom Schwerpunkt ausgehenden Strahlen die Kontur mehrmals schneiden. In diesem Fall muss sich dann für einen der Schnittpunkte als Abtastpunkt entschieden werden (z.B. immer denjenigen mit dem größten Abstand [Neb99]). Würde man dann die Abtastpunkte mit Geradenstücken zu einem Polygon verbinden, so weicht die dadurch erhaltene Konturapproximation aber evtl. erheblich von der Originalkontur ab.

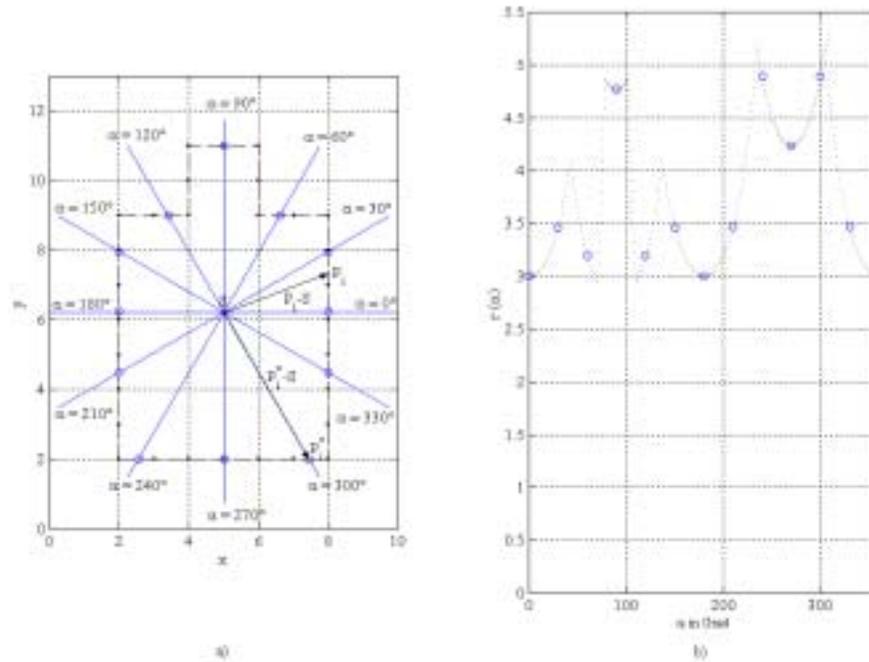


Abbildung 2.4: Beispiel für die polare Darstellung einer Kontur. a) Originalkontur mit Konturpunkten (markiert durch \bullet) zwischen denen linear interpoliert wurde ($\rightarrow P_i$). Die Punkte P_i^* (markiert durch \circ), sind diejenigen Abtastpunkte, an denen die interpolierte Kontur von äquidistanten Winkelstrahlen (hier jeweils 30° auseinander) geschnitten wird. b) Polare Darstellung der Beispielkontur aus a). Die Kreise \circ markieren die Abstände der Abtastpunkte P_i^* vom Schwerpunkt S , d.h. die Beträge der Vektoren $(P_i^* - S)$. Gepunktet dargestellt die Abstände der interpolierten Konturpunkte P_i .

Eine andere Variante, eine Kontur als eindimensionale Funktion darzustellen, ist die sogenannte Φ -s-Funktion. Diese Variante hat den zuvor genannten Nachteil nicht. Bei dieser Methode wird ab einem bestimmten Konturpunkt der Krümmungswinkel der diskreten Kontur aufsummiert und gegen die bis dahin zurückgelegte Bogenlänge s abgetragen.

Zunächst aber einige Worte zum kontinuierlichen Fall. In der Differentialgeometrie wird unter der *Krümmung* einer Kurve folgendes verstanden:

Unter dem *Krümmungskreis* einer Kurve in einem Punkt $P_B = (x_B; y_B)$ wird der Kreis verstanden, der mit der Kurve in diesem Punkt eine Berührung von mindestens 2. Ordnung aufweist. Zwei Kurven $y = f(x)$ und $y = g(x)$ haben in dem Punkt P_B eine Berührung 2. Ordnung, wenn ihre 0., 1. und 2. Ableitung an

diesem Punkt gleich, die 3. Ableitung aber ungleich ist:

$$\begin{aligned} f(x_B) &= g(x_B) \\ f'(x_B) &= g'(x_B) \\ f''(x_B) &= g''(x_B) \\ f'''(x_B) &\neq g'''(x_B). \end{aligned} \quad (2.26)$$

Der Radius eines solchen *Krümmungskreises* heißt *Krümmungsradius* ρ . Sein Mittelpunkt (*Krümmungsmittelpunkt*) liegt auf der Normalen in dem Kurvenpunkt P_B . Gilt für den Krümmungsradius

$$\rho \rightarrow \infty, \quad (2.27)$$

so kann dieser Kreis lokal als eine Gerade angesehen werden. D.h. in einem solchen Berührungspunkt ist die untersuchte Kurve nicht gekrümmt. Deshalb ist die *Krümmung* K in einem solchen Punkt 0. Die *Krümmung* einer Kurve ist definiert als der reziproke Krümmungsradius:

$$K = \frac{1}{\rho} \quad (2.28)$$

Mit den Gleichungen (2.26) und der Funktion

$$g(x) = \pm \sqrt{\rho^2 - x^2} \quad (2.29)$$

für einen Krümmungskreis lässt sich die Krümmung K einer Kurve $f(x)$ in dem Punkt P_B herleiten. Ohne Beweis ergibt sich dann die Krümmung folgendermaßen (siehe [BS87]):

$$K = \frac{f''(x_B)}{(1 + f'(x_B)^2)^{\frac{3}{2}}} \quad (2.30)$$

Eine andere Möglichkeit die *Krümmung* zu beschreiben, wird dadurch erreicht, dass man sich vorstellt, eine Kurve mit einer bestimmten konstanten Geschwindigkeit $\frac{ds}{dt}$ zu durchlaufen. ds sei hier die Entfernung (*Bogenlänge*), die auf der Kurve in der Zeit dt zurückgelegt wird. Die Richtung, in der die Kurve durchlaufen wird, ändert sich um so schneller, je gekrümmter die Kurve ist. Da die Geschwindigkeit des Durchlaufens als konstant angenommen wird, ist ds proportional zu dt und es ist möglich, die *Krümmung* als Änderung der Richtung $\Delta\alpha$ nach dem Durchlaufen einer bestimmten Bogenlänge Δs aufzufassen (Abb. 2.5). Als Maß für die Richtung einer Kurve in einem Punkt wird i. allg. der Winkel α angenommen, den die Tangente durch diesen Punkt mit der x -Achse einschließt. D.h. die *Krümmung* wird gemessen durch die Veränderung der Tangentenrichtung

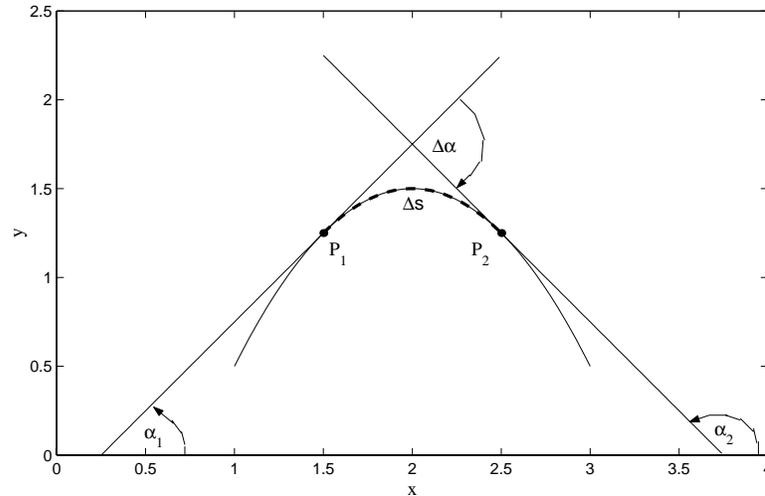


Abbildung 2.5: Zur Definition der Krümmung. Die Krümmung wird gemessen durch die Veränderung der Tangentenrichtung mit der Bogenlänge. Die Tangenten durch die Punkte P_1 und P_2 der Kurve haben einen Richtungsunterschied von $\Delta\alpha = \alpha_1 - \alpha_2$. Die Bogenlänge zwischen diesen beiden Punkten (gestrichelt markiert) beträgt Δs . Die Krümmung zwischen den beiden Punkten P_1 und P_2 beträgt $\Delta\alpha/\Delta s$. Geht der Abstand der beiden Punkten gegen 0, d.h. wird P_2 nach P_1 verschoben, so ergibt sich die Krümmung in dem Punkt $P_1 = (x_1; y_1)$ zu $d\alpha/ds|_{x_1(s), y_1(s)}$.

mit der Bogenlänge. Im kontinuierlichen Fall ist die Krümmung folgendermaßen definiert:

$$K = \lim_{\substack{\Delta\alpha \rightarrow 0 \\ \Delta s \rightarrow 0}} \frac{\Delta\alpha}{\Delta s} = \frac{d\alpha}{ds}. \quad (2.31)$$

D.h. die *Krümmung* K ist die Ableitung des *Krümmungswinkels* α nach der Bogenlänge s . Bei genauerer Betrachtung würde sich herausstellen, dass die Gleichung (2.31) und die Gleichungen (2.28), (2.30) den gleichen Wert für die Krümmung K ergeben.

Im diskreten Fall approximieren die Konturpunkte die kontinuierliche Kontur nur. Die Kontur lässt sich in diesem Fall durch einen Polygonzug bzw. im Falle einer geschlossenen diskreten Kontur als Polygon beschreiben. Interessant sind hier die Krümmungen an den Ecken der Polygone, denn hier ändert sich die Krümmung schlagartig, d.h. die Krümmungsfunktion ist an diesen Stellen nicht stetig. Das bedeutet, in diesen Punkten kann die Krümmung nicht mit Hilfe der Gleichungen (2.28), (2.30) und (2.31) berechnet werden. Dies stellt aber kein allzu großes Problem dar. Man behilft sich in diesem Fall einfach damit, dass man

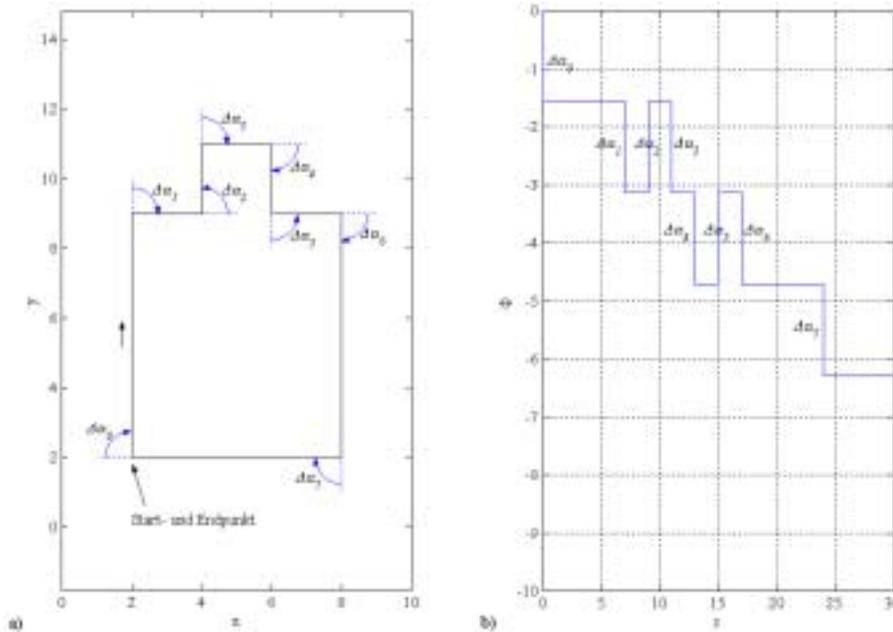


Abbildung 2.6: Krümmungswinkel und Φ - s -Funktion eines Polygons: a) Ein Polygon mit 8 rechtsläufigen Eckpunkten. Ein Krümmungswinkel $\Delta\alpha_i$ an einem Polygoneckpunkt ist derjenige Winkel, um den das Geradenstück links des Polygoneckpunktes um den Eckpunkt rotiert werden muss, so dass das linke Geradenstück und das Geradenstück rechts vom Polygoneckpunkt kollinear sind, d.h. einen Krümmungswinkel von 0 Grad hätten. b) Φ - s -Funktion des Polygons aus Abb. 2.6a. Hier wird die Summe der Krümmungswinkel über der Bogenlänge ab dem Startpunkt abgetragen. An den Polygoneckpunkten kommt es zu Sprüngen. Die Summe der Krümmungswinkel bei einem rechtsherum durchlaufenen Polygon beträgt -360 Grad bzw. -2π .

stellvertretend den Krümmungswinkel $\Delta\alpha_i$ der beiden Geradenstücke, die sich in dem interessierenden Eckpunkt schneiden, zur Krümmungsbeschreibung heranzieht (Abb. 2.6a).

Werden ab einem beliebigen Anfangspunkt die Krümmungswinkel $\Delta\alpha_i$ aufsummiert, so ergibt sich ein funktionaler Zusammenhang zwischen der aktuellen Summe Φ_i und der bis dahin, vom Anfangspunkt, zurückgelegten Bogenlänge s_i . Dieser Zusammenhang wird als Funktion $\Phi(s)$ bzw. Φ - s -Funktion bezeichnet (Beispiel in Abb. 2.6b). Die Summe der Krümmungswinkel ergibt bei geschlossenen Konturen immer 360 Grad, wenn die Kontur linksherum durchlaufen wird

oder -360 Grad, wenn die Kontur rechtsherum durchlaufen wird:

$$\sum_{i=0}^{n-1} \Delta\alpha_i = \pm 360^\circ \quad . \quad (2.32)$$

Wird die Φ -*s-Funktion* äquidistant abgetastet, so lässt sich auch diese mit einer diskreten Fourieranalyse (DFT) analysieren und eine bestimmte Anzahl von den erhaltenen Fourierdeskriptoren als globale Merkmale dieser Objektkontur heranziehen ([KSS86], [Ott95]).

Neben diesen beispielhaft vorgestellten Verfahren zur Darstellung von diskreten Konturen als 1D-Funktionen sei am Schluss dieses Abschnitts noch eine weitere Möglichkeiten genannt. Als Variante zur Abtastung der Kontur mit äquidistanten Winkelstrahlen bietet sich an, die linear interpolierte Kontur mit konstanter Bogenlänge entlang der Kontur abzutasten, und dann die Abstände dieser Abtastpunkte zum Schwerpunkt der Kontur in Abhängigkeit der Gesamtbogenlänge von einem Anfangsabtastpunkt abzutragen.

Beispiele für Verfahren mit lokalen Anteilen werden im Rahmen des Kapitels 3 besprochen.

2.3 Invariante und variante Verfahren

Objekte, die bis auf Lage (Translation), Größe (Skalierung), Drehung (Rotation), Spiegelung, Projektion, Scherung (Abb. 2.7) und anderen in Kapitel 1 genannten Deformationen identisch sind, sollten Objektmerkmale besitzen, die möglichst *invariant* gegenüber solchen Transformationen sind.

Es existieren eine Vielzahl von Arbeiten, die sich mit dieser Thematik beschäftigen. [Woo96] unterscheidet neben anderen Methoden folgende Hauptgruppen: *Integrale Invarianten* sind Invarianten, die auf einer integralen Transformation [FC88] wie z.B. der Fouriertransformation basieren; *algebraische Invarianten* (auch *Momenteninvarianten*) basieren auf Quotienten oder dem Potenzieren von Momenten, wie z.B. der *gewöhnlichen Momente* oder der *Zernike Momente* (z.B. [PL92]); mit Hilfe des sogenannten *weight sharings* (siehe z.B. [Cun89], [RHW86]) werden auch *künstliche neuronale Netze* (z.B. [Bis95], [Zel94]) für die invariante Mustererkennung herangezogen, wie z.B. Fukushima's Neocognitron [Fuk80], [FM82].

Oft sind ausschließlich die Translations-, Skalierungs- und Rotationsinvarianz Gegenstand der Untersuchungen (z.B. [SM95], [Wer97]). Mit Hilfe von nur 3 nicht \rightarrow *kollinearen* Punkten auf einem Objekt lässt sich bei zugelassener Rotation, Skalierung und Translation die beste Anpassung (engl. alignment transformation) an ein vorgegebenes Modell errechnen [HU88], [UB91]. Dabei müssen al-

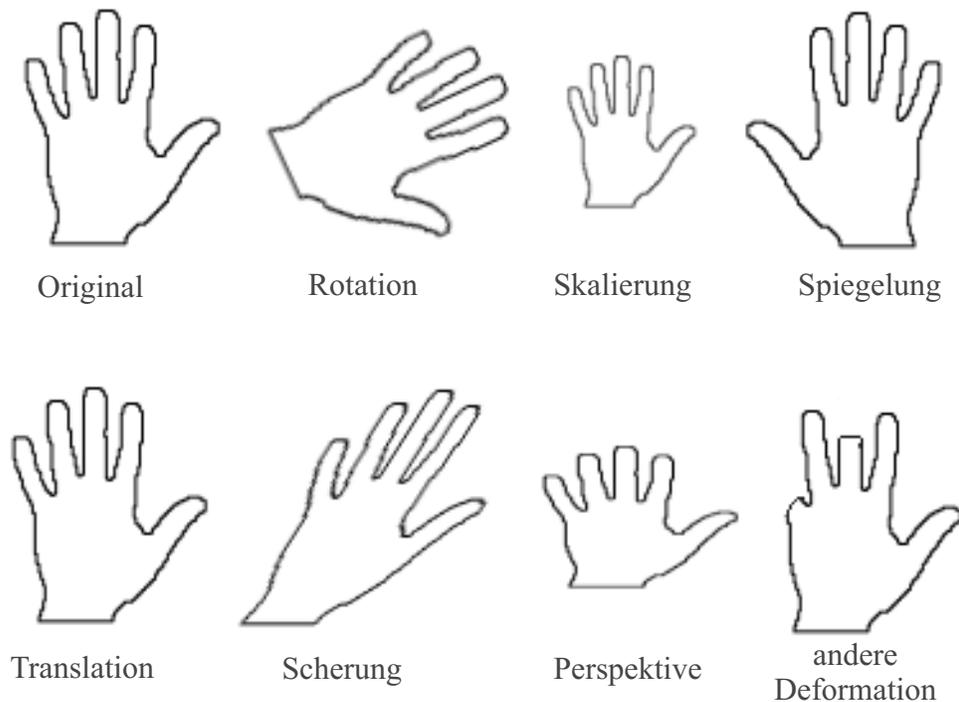


Abbildung 2.7: Mögliche Transformationen einer Beispielkontur *Hand*. Links oben ist die Originalkontur gezeigt, links unten die reine Translation dieser Kontur. Die anderen Konturen sind gegenüber der Originalkontur sowohl translatiert als auch einer anderen Transformation unterzogen worden. Die Kontur rechts unten zeigt keine einfach beschreibbare Transformation. Diese Transformation ist stark nichtlinear.

lerdings die 3 korrespondierenden Punkte auf dem Modell bekannt sein. [BU93], [Sch98a] beschäftigen sich zusätzlich mit der Projektionsinvarianz.

An dieser Stelle soll anhand der aus Abschnitt 2.2.2 bekannten gewöhnlichen Momente beispielhaft auf die Translations-, Skalierungs- und Rotationsinvarianz eingegangen werden.

Unter *Translation* einer Kontur wird die Verschiebung jedes Konturpunktes (x, y) um $(\Delta x, \Delta y)$ in jeweils einen anderen Konturpunkt (x', y') mit Hilfe der folgenden Gleichungen verstanden:

$$x' = x + \Delta x, \quad y' = y + \Delta y \quad (2.33)$$

Die *Translationsinvarianz* einer Kontur lässt sich dadurch erreichen, dass mit Hilfe der 0. und 1. Momente m_{00} , m_{01} und m_{10} die Schwerpunktkoordinaten \bar{x} , \bar{y}

errechnet werden (Gleichungen 2.6-2.10) und alle Konturpunkte so durch Subtraktion der Schwerpunktkoordinaten verschoben werden, dass der neue Schwerpunkt der Kontur im Koordinatenursprung liegt.

$$x' = x - \bar{x}, \quad y' = y - \bar{y} \quad (2.34)$$

Alle Momente von sich nur in der Lage unterscheidenden Konturen werden mittels einer solchen Translation gleich.

Unter *Skalierung* einer Kontur werden Größenänderungen folgender Art verstanden:

$$x' = \sigma x, \quad y' = \sigma y \quad (2.35)$$

Mit Hilfe des Schwerpunktes lassen sich die zentrierten Momente μ_{pq} (Gleichung 2.11) und damit die folgenden *normierten zentrierten Momente der Ordnung $p+q$* einführen (siehe [Hab95], [Wer97]):

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}+1}} \quad (2.36)$$

Die Momente von sich nur in der Größe unterscheidenden Konturen werden durch eine solche Normierung (Gl. (2.36)) *skalierungsinvariant*. Ein Beispiel für ein normiertes zentrales Moment 2. Ordnung ist

$$\eta_{11} = \frac{\mu_{11}}{\mu_{00}^2} = \frac{\sum_{y=0}^{M-1} \sum_{x=0}^{N-1} (x - \bar{x})(y - \bar{y}) g(x, y)}{\left(\sum_{y=0}^{M-1} \sum_{x=0}^{N-1} g(x, y) \right)^2}, \quad (2.37)$$

und mit Gleichung (2.5) steht im Nenner das Quadrat der Anzahl der Konturpunkte. Die mit Gleichung (2.36) eingeführten normierten zentralen Momente η_{pq} garantieren allerdings ausschließlich im kontinuierlichen Fall exakte Skalierungsinvarianz. In diskreten Bildern wird nur eine Annäherung an die Skalierungsinvarianz erzielt.

Unter *Rotation* einer Kontur um einen bestimmten Winkel γ um den Koordinatenursprung versteht man folgenden formellen Zusammenhang für die Konturpunkte:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) \\ -\sin(\gamma) & \cos(\gamma) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.38)$$

Um *Rotationsinvarianz* zu erreichen, wird die Kontur zunächst so verschoben, dass ihr Schwerpunkt im Koordinatenursprung liegt. Dann werden die beiden bereits in Abschnitt 2.2.2, in Gleichung (2.17), berechneten *Hauptachsen* bestimmt. Die beiden Hauptachsen sollen die beiden neuen Koordinatenachsen werden. Um eindeutig festzulegen, welche der Achsen die neue x- bzw. y-Achse werden soll schlägt [Wer97] vor:

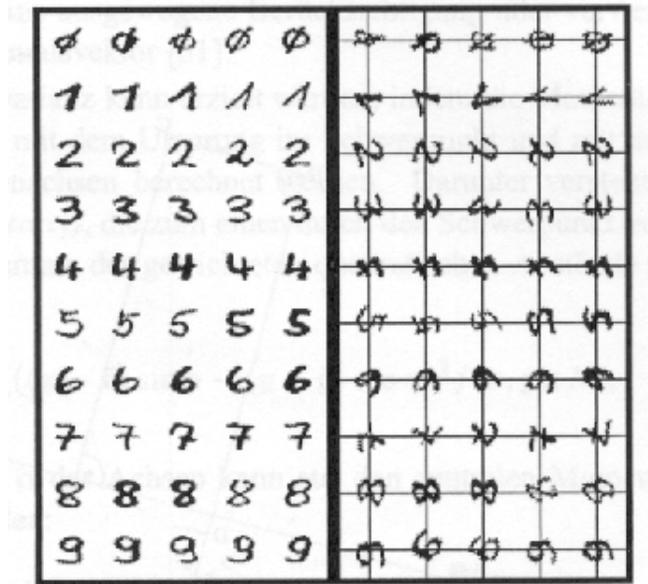


Abbildung 2.8: Ausrichtung von jeweils 5 nichtidentischen handgeschriebenen Ziffern von 0-9. Links die Originalziffern und rechts die ausgerichteten Ziffern mit eingezeichneten normierten Koordinatensystemen (aus [Wer97]).

- Wähle den Winkel der x-Achse so, dass $\mu_{02} > \mu_{20}$ für die im gedrehten Koordinatensystem berechneten Momente erfüllt ist.
- Wähle die Richtung der x-Achse so, dass $\mu_{03} > 0$ gilt. Durch Festlegung des Drehsinns des Winkels zwischen x- und y-Achse, ist auch deren Richtung eindeutig bestimmt.

Die Momente von ausschließlich rotierten Konturen werden durch eine solche Normierung des Koordinatensystems *rotationsinvariant*. Ein Beispiel für ein solch normiertes Koordinatensystem wird in [Wer97] gegeben und ist in Abbildung 2.8 gezeigt. Es wurden jeweils 5 nichtidentische handgeschriebene Ziffern von 0 bis 9 verwendet. Um die einzelnen Ziffern miteinander vergleichen zu können, sollen diese möglichst gleich ausgerichtet werden. Hier bietet sich die Bestimmung des zuvor beschriebenen normierten Koordinatensystems an. Es wurden in diesem Fall allerdings nicht ausschließlich die Konturen, sondern jedes Pixel der mit 256 Graustufen quantisierten Ziffern benutzt. Für jede Ziffer wird ihr Schwerpunkt und ihre Hauptachse berechnet und entsprechend verschoben und gedreht. Es ist leicht zu sehen, dass diese Art der Rotationsinvarianz schon für nur leicht unterschiedliche Objekte nicht sehr robust funktioniert. Die Zuverlässigkeit der

Ausrichtung hängt stark von den zugrunde liegenden Objekten ab. Bei Objekten, die ähnlich große Werte für μ_{02} und μ_{20} bzw. ein μ_{03} bei ungefähr 0.0 besitzen, arbeitet diese Methode unzuverlässig.

Um sowohl translations-, rotations- und skalierungsinvariant zu werden, bieten sich folgende Schritte an:

1. Verschiebung des Koordinatenursprungs in den Schwerpunkt der Kontur (Translationinvarianz).
2. Drehung des Koordinatensystems durch richtige Wahl der Hauptachsen der Kontur als neue Koordinatenachsen (Rotationsinvarianz).
3. Normierung der so erhaltenen zentralen Momente mit Hilfe von Gleichung (2.36) (Skalierungsinvarianz).

[Hu62] definierte aufbauend auf den normierten zentrierten Momenten sieben Merkmale, die invariant gegenüber Translation, Rotation und Skalierung sind (M1-M6 auch invariant gegen Spiegelung) und beispielsweise in [Hem98] zur Klassifikation von Joghurtbechern und Tetrapaks bzw. verschiedener Wendeschneidplatten mit Hilfe von Polynomklassifikatoren ([Sch77]) herangezogen wurden:

$$\begin{aligned}
 M_1 &= \eta_{02} - \eta_{20} \\
 M_2 &= (\eta_{02} - \eta_{20})^2 + 4\eta_{11}^2 \\
 M_3 &= (\eta_{03} - 3\eta_{21})^2 + (3\eta_{12} - \eta_{30})^2 \\
 M_4 &= (\eta_{03} + \eta_{21})^2 + (\eta_{12} + \eta_{30})^2 \\
 M_5 &= (\eta_{03} - 3\eta_{21})(\eta_{03} + \eta_{21}) \left((\eta_{03} + \eta_{21})^2 - 3(\eta_{12} + \eta_{30})^2 \right) \\
 &\quad + (3\eta_{12} - \eta_{30})(\eta_{12} + \eta_{30}) \left(3(\eta_{03} + \eta_{21})^2 - (\eta_{12} + \eta_{30})^2 \right) \\
 M_6 &= (\eta_{02} - \eta_{20}) \left((\eta_{03} + \eta_{21})^2 - (\eta_{12} + \eta_{30})^2 \right) \\
 &\quad + 4\eta_{11}(\eta_{03} + \eta_{21})(\eta_{12} + \eta_{30}) \\
 M_7 &= (3\eta_{12} - \eta_{30})(\eta_{03} + \eta_{21}) \left((\eta_{03} + \eta_{21})^2 - 3(\eta_{12} + \eta_{30})^2 \right) \\
 &\quad + (3\eta_{12} - \eta_{30})(\eta_{12} + \eta_{30}) \left(3(\eta_{03} + \eta_{21})^2 - (\eta_{12} + \eta_{30})^2 \right).
 \end{aligned} \tag{2.39}$$

Bei nichtdeformierten Objekten, die sich ausschließlich durch lineare Transformationen, wie z.B. Translation, Rotation, Skalierung, Spiegelung und bestimmte Perspektiven unterscheiden, lassen sich Invarianzen noch relativ "einfach" bestimmen. Bei nichtlinear deformierten Objekten ist dies allerdings sehr schwierig. So ist es in diesem Fall oft gar nicht möglich zu zeigen, dass diese Verfahren invariant sind. Viele Arbeiten, die auf dem Gebiet der nicht-linear deformierten Objekte arbeiten, bemühen sich aber trotzdem, bestimmte Invarianztypen, allen voran die Translations-, Rotations- und Skalierungsinvarianz, zu gewährleisten.

Kapitel 3

Analyse deformierter 2D-Konturen

Diese Arbeit beschäftigt sich mit der Einordnung von 2D-Objektkonturen in Konturklassen, wie z.B. die Klassen der Vögel, der Autos, der Fische, der Flaschen oder der Flugzeuge. In Abschnitt 1.1.2 wurde in den Vordergrund gestellt, dass Variabilität, Perspektive, physikalische und technische Deformationen als Gesamtdeformation der Objektkonturen einer Klasse untereinander aufgefasst werden können. Deshalb müssen Objekte bestimmte, sie auszeichnende lokale Charakteristiken besitzen. Dies sind beispielsweise die Flossen bei den Fischen, die Krallen bzw. Schnäbel bei den Vögeln und die Flaschenhalse bei den Flaschen. Ohne diese Charakteristiken wäre es nicht möglich, einzelne Objekte einer Objektklasse zuzuordnen, wenn diese aus einer Vielzahl verschiedener (deformierter) Objekte besteht.

Die Deformation zweier Objektkonturen ist i. allg. nichtlinear. D.h. die Deformationsanteile werden sich nicht gleichmäßig über die gesamte Kontur verteilen. Weiter ist anzunehmen, dass die Deformation von Konturen aus verschiedenen Klassen nichtlinearer ist, d.h. einen höheren Deformationsgrad besitzt, als die Deformation von Konturen aus derselben Klasse. Mit anderen Worten heißt das, dass Konturen aus derselben Klasse eine größere Ähnlichkeit besitzen als Konturen aus verschiedenen Klassen, und dass diese größere Ähnlichkeit daher rührt, dass die lokalen Charakteristiken weitestgehend von der Deformation verschont bleiben.

Aus den genannten Gründen ist es also sinnvoll für die Analyse von deformierten Konturen Ähnlichkeitsmaße einzuführen, die lokale Eigenschaften der Konturen berücksichtigen. In diesem Kapitel werden Verfahren vorgestellt, die diese Bedingungen berücksichtigen und die zu einer Gruppe von Verfahren gehören, in die sich, bis auf die in Abschnitt 3.2 vorgestellte Methode zur strukturellen Mustererkennung, auch die neuentwickelte, in Kapitel 4 ausführlich vorgestellte Methode in Anteilen einordnen lässt.

Allen beschriebenen Verfahren liegt mindestens eines der nachfolgenden allgemeinen Konzepte zugrunde:

Ein wichtiges Konzept ist es anzunehmen, dass Objekte Eigenschaften besitzen, die unter jedweder Transformation **in Grenzen** invariant bleiben [PM47] (Transformationstoleranz). Hier finden auch die in Abschnitt 2.3 genannten Verfahren ihren Platz, wobei dort das Augenmerk auf die völlige Invarianz gelegt wurde. Oft werden in diesem Zusammenhang mehrere solcher Eigenschaften bzw. \rightarrow *Merkmale*, die sich in einzelnen Zahlenwerte ausdrücken lassen in einem \rightarrow *Merkmalvektor* zusammengefasst (siehe z.B. [Nie74], [Nie90]). Globale Merkmale sind in diesem Zusammenhang z.B. bestimmte Momente, die Kompaktheit oder die Exzentrizität. Ein lokales Merkmal könnte z.B. ein Zahlenwert sein, der ausdrückt wie gut der detektierte Flaschenhals einer Flasche mit einem Modellflaschenhals übereinstimmt. In diesem Fall werden stellvertretend für die 2 Konturen ihre Merkmalvektoren miteinander verglichen.

Ein anderes wichtiges Konzept ist das der Teilbasiertheit, d.h. die Kontur wird in Konturabschnitte (z.B. [Lak00], [Nis98]) oder das Innere der Kontur in Teilflächen (z.B. [KW96], [SK95]), sogenannte *Primitiven*, zerlegt. Um 2 Konturen miteinander zu vergleichen, werden beispielsweise die einzelnen Konturabschnitte geordnet und miteinander verglichen oder die Struktur (\rightarrow *Strukturelle Mustererkennung* [Pav77]) der Teilflächen wird z.B. durch \rightarrow *attributierte relationale Graphen* dargestellt, die auf \rightarrow *Isomorphie* untersucht werden. Ein anderer Ansatz in diesem Zusammenhang ist die \rightarrow *syntaktische Mustererkennung* ([Fu74], [GW99]). Hier werden Objekte zunächst in einfachste Bestandteile wie z.B. in Geradenstücke zerlegt und diese anschließend mit Hilfe von Methoden aus der Syntaxanalyse formaler Sprachen zu Bestandteilen höherer Ordnung zusammengefasst. Sowohl die strukturelle als auch die syntaktische Mustererkennung gehören zu den hierarchischen Verfahren und unterscheiden sich im Wesentlichen dadurch, dass die strukturelle Bildverarbeitung invariante Eigenschaften bezüglich der Relationen der einzelnen Teile untereinander benutzt. Mit solchen invarianten Relationen sind z.B. gemeint: "über", "rechts von", "beinhaltet" usw..

Ein weiteres wichtiges Konzept ist die Einführung einer Kostenfunktion, die die Transformations- (Deformations-) Kosten, die aufgewendet werden müssen, um eine Kontur in eine andere (nichtlinear) zu transformieren, minimiert. Basri et al. [BCGJ98] basieren auf der sogenannten *elastischen Anpassung* (engl. *elastic matching*). Um lokale Deformationskosten zu berechnen, bedienen sie sich in der Physik und benutzen u.a. ein mechanisches Federmodell. Sederberg et al. [SGWM93] beschäftigten sich im Bereich der Computergrafik mit der \rightarrow *Interpolation* von einer Kontur in eine andere, wobei bestimmte korrespondierende Punkte der beiden Konturen vorgegeben sein müssen. Bei diesem auch als *morphing* bekannten Verfahren ist das Hauptaugenmerk darauf gerichtet, Zwischenkonturen zwischen 2 Konturen so zu berechnen, dass ein zeitlicher Übergang zwischen den beiden Konturen flüssig wahrgenommen wird. Eine Kontur lässt sich auch als *Zeichenkette* (engl. *string*) auffassen, dessen einzelne Zeichen die einzelnen

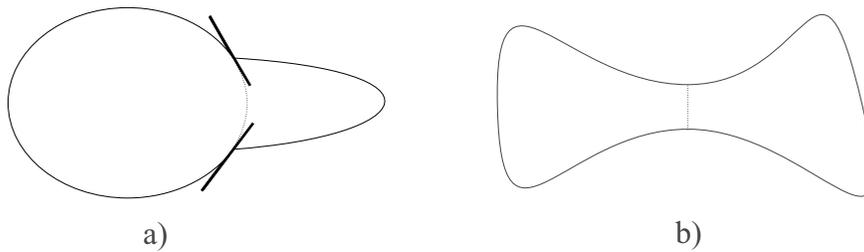


Abbildung 3.1: Glieder (engl. limbs) und Hälse (engl. necks). a) Gliedbasierte Partitionierung. Die Partitionierungslinie (gestrichelt) beginnt und endet an Stellen hoher konkaver Krümmung und verläuft so, dass die Tangenten (fett), würden sie der Partitionierungslinie folgen, ohne Sprung ineinander übergehen. b) Halsbasierte Partitionierung. Die Partitionierungslinie (gestrichelt) ist diejenige Linie, die 2 gegenüberliegende Konturpunkte, deren Abstand lokal minimal ist, durch eine Gerade verbindet. Mit anderen Worten, die Kontur wird an Verengungen aufgeteilt.

Konturpunkte sind ([SK83], [PK99], [PK00b], [Sch98b]). Um eine Kontur in die andere zu transformieren werden hier Konturpunkte ersetzt (verschoben), eingefügt oder entfernt. Diese Operationen sind mit lokalen Kosten verbunden, deren Summe mit Hilfe einer Kostenfunktion minimiert wird.

Die in den folgenden Unterabschnitten vorgestellten Verfahren basieren alle direkt auf der Außenkontur eines Objektes. Es werden beispielhaft verschiedene Prinzipien zur Herangehensweise an die Analyse von deformierten Konturen erläutert. Es werden aber auch Details vorgestellt, die exemplarisch für eine Vielzahl anderer Möglichkeiten stehen, diese Details anzugehen. Im abschließenden Unterabschnitt dieses Kapitels werden die verschiedenen Verfahren diskutiert.

3.1 Teilbasierter Ansatz

Unter teilbasiertem Ansatz versteht man die Unterteilung der Form in einzelne Formteile. So wird entweder das Innere (die eingeschlossene Fläche) einer Kontur unterteilt oder die Kontur selbst. Diese Unterteilung ist dann die Basis für einen Konturvergleich.

Ein Verfahren, das das Innere der Kontur analysiert, wird in Abschnitt 3.2 vorgestellt.



Abbildung 3.2: Auf der rechten Seite ist eine Vergrößerung des Kappenbereiches der Flaschenaußenkontur auf der linken Seite zu sehen. Die Konturpunkte sind 8er-Nachbarn. Mit **A** sind Stellen starker konkaver Krümmung und mit **B** sind Stellen schwacher konkaver Krümmung gekennzeichnet.

Ein anderes Verfahren zur Unterteilung des Konturinneren beschreiben Siddiqi et al. in [SK95], [STK96]. Sie unterteilen das Innere der Kontur in sogenannte *gliedbasierte Teile* (engl. *limb-based parts*) und *halsbasierte Teile* (engl. *neck-based parts*) (Abbildung 3.1). Gliedbasierte Teile entstehen an Partitionierungslinien, deren Endpunkte konkave Stellen der Kontur sind. Zusätzliche Bedingung ist, dass die Tangente an dem einen Endpunkt, entlang der Partitionierungslinie, sanft in die Tangente am anderen Endpunkt übergeht. Beispiele für gliedbasierte Teile sind Schnäbel von Vögeln oder die Finger an einer Hand. Halsbasierte Teile entstehen folgendermaßen: Man sucht die engsten Stellen einer Kontur. Genauer bedeutet dies, dass die Stellen gesucht werden, an denen der Abstand gegenüberliegender Konturpunkte ein lokales Minimum besitzt. Die Geradenstücke, die die Konturpunkte mit minimalem Abstand miteinander verbinden, sind dann die Trennungslinien zwischen einzelnen Teilen. Beispiele für halsbasierte Teile sind die Schwanzflosse eines Fisches oder der Kopf eines Menschen.

Für die Aufteilung von Konturen bedeutet dies, dass die Konturstellen mit starker *konkaver* Krümmung detektiert werden, die die einzelnen Konturen dann in konvexe Polygonzüge einteilen, wie dies auch von Hoffman et al. in [HR84], [HS97] vorgeschlagen wird, die dazu psychologische Experimente durchführten.

Nun bestehen gerasterte Konturen, deren Konturpunkte 4er-Nachbarn oder auch 8er-Nachbarn sind, aus einer Vielzahl von schwachen als auch starken konkaven Krümmungen (Abbildung 3.2), die es zu unterscheiden gilt. Da die schwach konkaven Krümmungen als Störungen durch Rasterungs-, Quantisierungs-, Segmentierungs- und Rauscheffekte gedeutet werden können, muss nach Möglichkei-

ten gesucht werden diese zu unterdrücken, ohne die stark konkaven Krümmungen zu sehr zu beeinflussen.

Ein Ansatz, die zuvor genannten Effekte zu unterdrücken, besteht darin, die Kontur zu glätten. Es wird in diesem Zusammenhang auch oft von *Kontursimplifizierung* gesprochen. Neben Methoden, die versuchen die Punkte der gerasterten Kontur wieder in Punkte zu verschieben, die eine bessere Näherung der realen kontinuierlichen Kontur ergeben (z.B. [BV99]), gibt es eine Vielzahl an Methoden, die eine polygonale Näherung einer Kontur zum Ziel haben. Im letzten Fall werden aufeinanderfolgende Geradenstücke gesucht, die eine optimale Anpassung der Originalkontur ergeben. Dies geschieht nach verschiedensten Kriterien, z.B. dass die Summe der senkrechten Abstände der Originalpunkte auf die Geradenstücke minimal ist. Da eine optimale Anpassung allerdings oft sehr rechenzeitintensiv ist, beschäftigen sich die meisten Arbeiten auf diesem Gebiet damit Verfahren zu finden, die einen guten Kompromiss zwischen der Anpassungsgüte und der Berechnungsgeschwindigkeit einer solchen Anpassung bieten (siehe z.B. [Pav78], [Ros97], [Lon98]).

Latecki und Lakämper [LL98a], [Lak00] stellen ein mathematisch sehr fundiertes Verfahren zur Polygonapproximation vor, das sie selbst als *diskrete Kurvenevolution* bezeichnen. Der Begriff der *Evolution* einer Kurve entstammt der *Skalenraumtheorie* (engl. *scale space theory*). Eine kontinuierliche Kontur wird dabei immer wieder geglättet, so dass zunächst Störungen entfernt werden. Mokhtarian et al. ([MM92], [MAK96]) repräsentieren eine Kontur als Krümmungsfunktion¹ über der Bogenlänge und falten diese Funktion mit einem \rightarrow *Gaußfilter* (z.B. [Jäh93]), dessen Parameter Halbwertsbreite sich während der Evolution permanent vergrößert. In Abbildung 3.3 ist ein Beispiel einer solchen Evolution dargestellt.

Die diskrete Kurvenevolution basiert auf der Skalenraumtheorie und eliminiert ebenfalls zunächst die Störungen. Im Laufe der Evolution werden nach und nach Konturpunkte p_i aus einer Kontur

$$K = \{p_1, p_2, \dots, p_i, \dots, p_M\} \quad (3.1)$$

mit M Konturpunkten eliminiert. Dies geschieht aufgrund einer Kostenfunktion c_{elem} , die für den zu entfernenden Punkt p_{elem} minimal ist. Es entsteht eine simplifizierte Kontur

$$K' = K \setminus p_{elem}. \quad (3.2)$$

Nun werden mit Hilfe von c_{elem} wiederum die Kosten für das Entfernen der übriggebliebenen Konturpunkte aus K' bestimmt. Der Konturpunkt p'_{elem} mit mini-

¹Diese Herangehensweise ist auch unter dem Namen Krümmungsskalenraum (engl. curvature scale space) bekannt

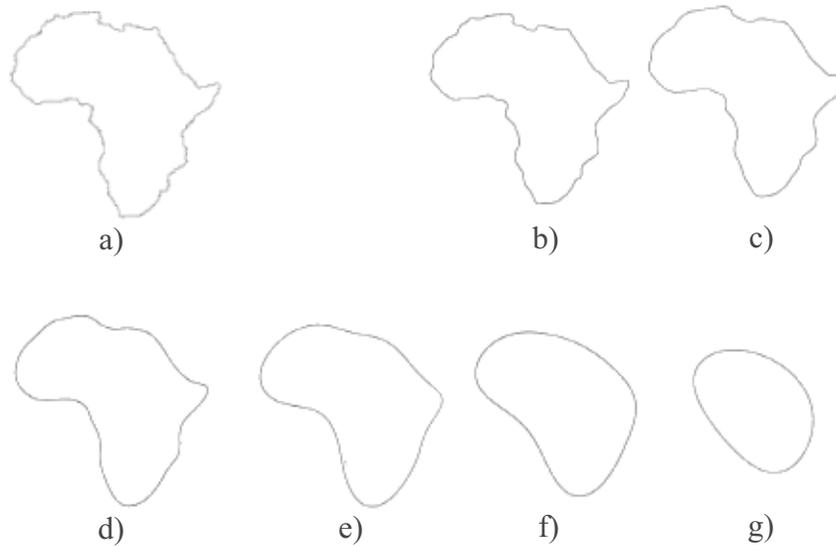


Abbildung 3.3: Beispiel zur Kurvenevolution basierend auf dem Krümmungsskalenraum (engl. curvature scale space). Der Umriss des Kontinents Afrika wird immer wieder mit steigenden Halbwertsbreiten σ gaußgefiltert. a) Original, b) $\sigma=2$, c) $\sigma=4$ d) $\sigma=8$, e) $\sigma=16$ f) $\sigma=32$ g) $\sigma=64$. Alle Abbildungen aus [MM92].

malen Kosten wird entfernt und es entsteht die Kontur

$$K'' = K' \setminus p'_{elem}. \quad (3.3)$$

Mit jedem folgenden diskreten Evolutionsschritt werden so weiter Konturpunkte eliminiert bis eine Kontur mit nur noch 3 Konturpunkten, ein Dreieck, übrigbleibt.

Eine Eigenschaft dieses Verfahrens ist es, dass in der evolutionierten Kontur die übriggebliebenen Konturpunkte eine exakte Teilmenge der Konturpunkte der Ursprungskontur K sind. Latecki und Lakämper leiten folgende Kostenfunktion c_{elem} für das Entfernen eines Konturpunktes her:

$$c_{elem} = |\alpha| \frac{l_1 l_2}{l_1 + l_2}. \quad (3.4)$$

l_1 bzw. l_2 sind die Längen benachbarter Geradenstücke s_1 und s_2 einer diskreten Kontur, die sich in einem Punkt P berühren (siehe Abbildung 3.4). α ist der Wendewinkel im Punkt P , der die Winkellage der Segmente s_1 und s_2 zueinander beschreibt.

c_{elem} ist hoch für stark konvexe bzw. stark konkave Krümmung in P und hat sein Maximum bzgl. l_1, l_2 bei $l_1 = l_2$. Ist $l_1 \gg l_2$ bzw. $l_2 \gg l_1$ bedeutet dies,

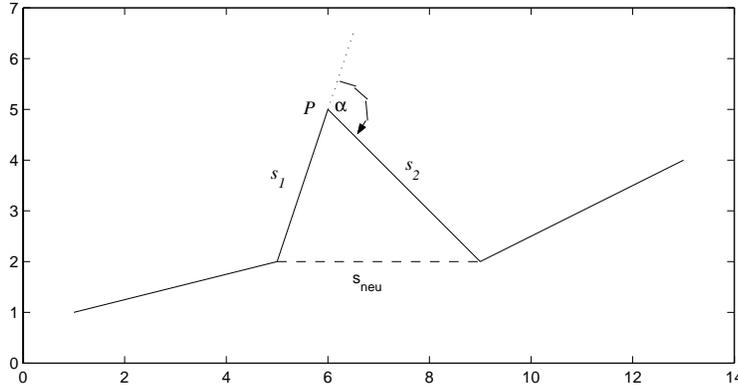


Abbildung 3.4: Zur Kostenfunktion für die diskrete Kurvenevolution. l_1 bzw. l_2 sind die Längen benachbarter Geradenstücke s_1 und s_2 einer diskreten Kontur, die sich in einem Punkt P berühren. α ist der Wendewinkel im Punkt P , der die Winkellage der Segmente s_1 und s_2 zueinander beschreibt. s_{neu} ist das neu entstehende Geradensegment für den Fall, dass der Punkt P eliminiert würde.

dass das durch Eliminierung von P entstehende neue Segment $s_{neu} \approx s_1$ bzw. $s_{neu} \approx s_2$ ist und die Kosten für das Entfernen von P dementsprechend relativ gering ausfallen.

Ein Beispiel für die diskrete Kurvenevolution ist in Abbildung 3.5 dargestellt. Dort wird ein weiteres zu lösendes Problem deutlich, nämlich automatisch zu bestimmen, bei welcher Evolutionsstufe die Evolution abubrechen sei. Bei der in Abbildung 3.5f dargestellten Kontur ist dies sicherlich zu spät, zumindest was die Erkennbarkeit dieses Objektes durch einen Menschen angeht. Dieses Problem wird dadurch gelöst, dass zunächst die Konturen aller Evolutionsstufen berechnet werden. Danach werden rückwärts, d.h. von der stärksten Vereinfachung ausgehend, die zu den jeweiligen Evolutionsstufen gehörenden Konturen mit der Originalkontur verglichen bis ein festzulegender Kostenschwellwert unterschritten wird.

Dieser Vergleich ist derselbe, der von Latecki und Lakämper ([Lak00], [LL00], [LL98b]) auch im allgemeinen für die Ähnlichkeitsanalyse zweier Konturen K und K^* herangezogen wird und wird im Folgenden beschrieben:

Nach Abschluss der diskreten Kurvenevolution der Konturen K und K^* entstehen vereinfachte Konturen \tilde{K} bzw. \tilde{K}^* , die zusätzlich auf gleiche Gesamtlänge skaliert werden, und aus Geradensegmenten \tilde{s}_i bzw. \tilde{s}_j^* bestehen:

$$\tilde{K} = \{\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_i, \dots, \tilde{s}_m\}, \quad (3.5)$$

$$\tilde{K}^* = \{\tilde{s}_1^*, \tilde{s}_2^*, \dots, \tilde{s}_j^*, \dots, \tilde{s}_n^*\}. \quad (3.6)$$

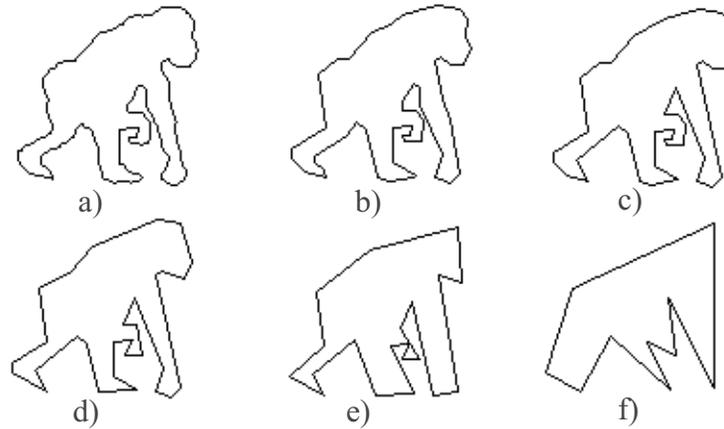


Abbildung 3.5: Beispiel zur diskreten Kurvenevolution. a) Originalkontur eines Menschenaffen mit 317 Konturpunkten. Diskrete Kurvenevolution auf b) 50 Konturpunkte, c) 40 Konturpunkte, d) 30 Konturpunkte, e) 20 Konturpunkte, f) 10 Konturpunkte.

Die Konturen \tilde{K} bzw. \tilde{K}^* werden jetzt in sogenannte *maximale konvexe Bögen* (engl. *maximal convex arcs*) und *maximale konkave Bögen* (engl. *maximal concave arcs*) \tilde{a}_i bzw. \tilde{a}_j^* eingeteilt (siehe Abbildung 3.6). Die maximalen konvexen Bögen sind dort außerhalb der Kontur und die maximalen konkaven Bögen innerhalb der Kontur neben der eigentlichen fett markierten Kontur angedeutet. Maximale konvexe und konkave Bögen wechseln sich ab und aufeinanderfolgende Bögen haben immer genau ein gemeinsames Geradensegment beim Übergang von einem maximalen konkaven zu einem maximalen konvexen Bogen und umgekehrt.

Um 2 maximale Bögen miteinander zu vergleichen, werden diese auf die gleiche Gesamtlänge skaliert. Danach werden sie als kumulierte Winkelkrümmungsfunktionen² (siehe Abschnitt 2.2.3) dargestellt. Diese werden dann basierend auf [ACH⁺91] so gegeneinander verschoben, dass die Fläche A zwischen den beiden Funktionen minimiert wird. A_{min} gibt dann den reinen Formunterschied der beiden maximalen Bögen an.

A_{min} wird zusätzlich gewichtet mit dem Skalierungsfaktor

$$S = \max\left(\frac{l}{l^*}, \frac{l^*}{l}\right) \quad (3.7)$$

und einem Längenfaktor

$$L = \max(l, l^*). \quad (3.8)$$

²Die kumulierte Winkelkrümmungsfunktion wird von Latecki und Lakämper als Stufenfunktion im Tangentenraum bezeichnet.

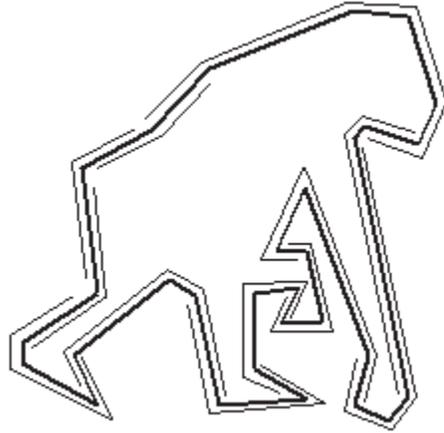


Abbildung 3.6: Maximale konvexe und konkave Bögen. Die maximalen konvexen Bögen sind außerhalb neben der fett markierten Kontur gekennzeichnet. Die maximalen konkaven Bögen sind innerhalb der Kontur gekennzeichnet. Die maximalen konkaven und konvexen Bögen folgen abwechselnd aufeinander und haben immer genau ein gemeinsames Geradenstück.

l bzw. l^* stehen für die Gesamtlängen der maximalen Bögen vor der Normierung auf gleiche Länge. S bewertet die unterschiedlichen relativen Gesamtlängen der maximalen Bögen, da die Längeninformaton in A_{min} durch die Normierung verloren ging. L bewertet die Bedeutung der beiden maximalen Bögen in Relation zu den Gesamtkonturlängen.

Die Gesamtkosten für den Vergleich von 2 maximalen Bögen beträgt also:

$$c_{Bogen} = A_{min} \cdot S \cdot L \quad (3.9)$$

Die Gesamtkosten für den Vergleich zweier Konturen \tilde{K} , \tilde{K}^* ergibt sich aus einer systematischen Suche nach der besten Korrespondenz aller maximalen Bögen von \tilde{K} mit den maximalen Bögen von \tilde{K}^* . Unter bester Korrespondenz ist diejenige Korrespondenz zu verstehen, die die Summe der Kosten c_{Bogen} , die für den Vergleich zweier beliebiger maximaler Bögen anfallen, minimiert. Dies ist ein Optimierungsproblem, das Latecki und Lakämper mit Hilfe der \rightarrow *dynamischen Programmierung* lösen, die im Zusammenhang mit der in Kapitel 4 vorgestellten Methode zur Synthese signifikanter Konturgruppen in Abschnitt 4.3.2 und im Anhang C vorgestellt wird.

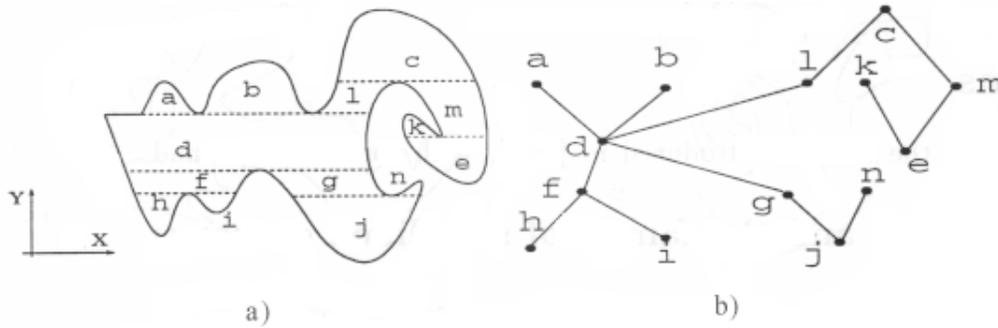


Abbildung 3.7: Konstruktion eines G-Graphen. a) Segmentierung $S(K)$: die Kontur K wird mit Hilfe von Geradenstücken, die sowohl tangential zur Kontur als auch parallel zur x -Achse verlaufen, in sogenannte *Klumpen* (engl. lumps) eingeteilt. b) G-Graph $G(K)$: Die Segmentierung $S(K)$ kann auf einen sogenannten G-Graphen $G(K)$ abgebildet werden, so dass die Klumpen von K mit den Knoten von $G(K)$ korrespondieren, und so dass die Nachbarschaftsbeziehungen zwischen den Klumpen mit den Kanten von $G(K)$ korrespondieren (aus [KB96]).

3.2 Strukturelle Mustererkennung

Der strukturellen Mustererkennung (engl. structural pattern matching) liegt das Prinzip zugrunde, ein komplexes Muster durch ein hierarchisches Zusammenfügen von einfacheren Submustern zu beschreiben. Dies geschieht durch Auswahl von geeigneten Muster-Primitiven und von Regeln, wie die Primitiven zu einem komplexeren Muster zusammengesetzt sind. Die Regeln sind die Grammatik. Das heißt, Objekte werden durch relationale Strukturen repräsentiert, deren Knoten Unterstrukturen und deren Verbindungslinien Relationen zwischen den Knoten, die sie verbinden, darstellen. Damit kann diese Methode als teilbasierte Methode mit invarianten Eigenschaften aufgefasst werden. Die invarianten Eigenschaften sind dann definiert als die Beziehungen der Submuster untereinander. Z.B. sind die Submuster einer Hand der Handteller und die 5 Finger. Die relationale Struktur besteht darin, dass jeder Finger einzeln und nicht in einer Kette aneinandergereiht an den Handteller "angebracht" sind.

Kupeev et al. stellen in [KW94], [KW96], [KB96], [KB98] u.a. ein Verfahren vor, das der strukturellen Mustererkennung zugeordnet werden kann.

Eine Kontur K wird zunächst folgender Segmentierung $S(K)$ unterzogen (Abb. 3.7a): K wird mit Hilfe von Geradenstücken, die sowohl tangential zur Kontur als auch parallel zur x -Achse verlaufen, in sogenannte *Klumpen* (engl. lumps) P_i eingeteilt. Das Beispiel in Abbildung 3.7a enthält die Klumpen $a, b, c, d, e, f, g, h, i, j, k, l, m, n$. Klumpen, die mehr als einen gemeinsamen Punkt besitzen, werden

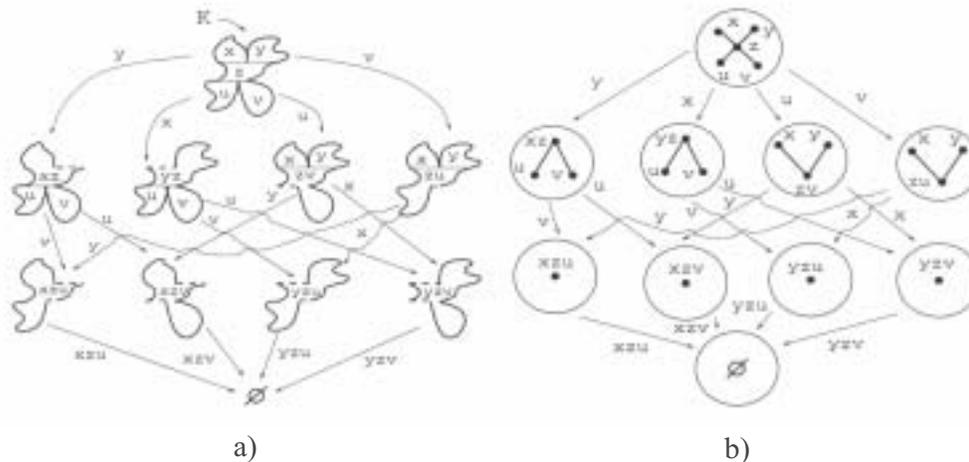


Abbildung 3.8: a) Segmentierungsgitter (engl. segmentation lattice) $M(K)$ einer Kontur K . b) Morphismusgitter (engl. morphism lattice) $L(G(K))$ des G -Graphen $G(K)$ der Kontur K , das zu $M(K)$ aus Abbildung 3.8a gehört (aus [KB96]).

benachbart genannt. Es wird eine *Eltern-Kind-Beziehung* zwischen den Klumpen insofern eingeführt, dass der Elternklumpen, die größeren y -Koordinaten besitzt. In Abbildung 3.7a ist beispielsweise der Klumpen b ein *Elternteil* von Klumpen d und der Klumpen g ist *Kind* von Klumpen d . Als *Geschwister* werden solche Klumpen bezeichnet, die in x -Richtung rechts oder links voneinander liegen und gemeinsame Elternteile besitzen. Z.B. sind in Abb. 3.7a die Klumpen f und g bzw. l und m Geschwister. Besitzt ein Klumpen nur einen einzigen benachbarten Klumpen, so heißt dieser Klumpen *Blatt*. In Abb. 3.7a sind a, b, h, i, k, n Blätter. Zusätzlich wird den Klumpen P_i jeweils ein Gewicht (Attribut) $w(P_i)$ zugeordnet. Ein solches Gewicht könnte z.B. die Summe der Längen der Konturabschnitte, die zu einem Klumpen gehören, oder die Fläche eines solchen Klumpens sein.

Die Segmentierung $S(K)$ kann auf einen sogenannten *G-Graphen* $G(K)$ abgebildet werden, so dass die Klumpen von K mit den Knoten von $G(K)$ korrespondieren, und so dass die Nachbarschaftsbeziehungen zwischen den Klumpen mit den Kanten von $G(K)$ korrespondieren (Abbildung 3.7b). Die Gewichte der Klumpen $w(P)$ werden den Knoten des G -Graphen $G(K)$ zugeordnet.

In Abbildung 3.8a ist nun folgender Prozess dargestellt. Ein beliebiger Blattklumpen P_i wird von der Kontur K entlang des Geradenstücks, das zur Segmentierung verwendet wurde, vom Klumpen P_j abgetrennt. Das Trennungsgerstenstück (gestrichelt in Abbildung 3.8a) wird nun Bestandteil der durch die Abtrennung neu entstandenen Kontur K' . Das Gewicht $w(P_i)$ wird dem Klumpen P_j

zugeordnet.

$$w(P'_j) = w(P_j) + w(P_i) \quad (3.10)$$

Hat der Klumpen P'_j nach der Abtrennung nur noch einen Blattklumpen P'_k , so wird P'_k dem Klumpen P'_j zugeordnet. P'_k wird in diesem Fall nicht abgeschnitten, d.h. P'_j und P'_k werden fusioniert. Die beiden korrespondierenden Gewichte werden addiert und dem neu entstandenen Klumpen zugeordnet.

$$w(P'_j) := w(P'_j) + w(P'_k) \quad (3.11)$$

Wird nun erneut ein willkürlich gewählter Blattklumpen von K' entfernt, so werden wie zuvor beschrieben die Gewichte addiert und evtl. Klumpen zusammengefasst. Es entsteht eine neue Kontur K'' . Dieser Prozess wird solange fortgesetzt, bis nur noch ein Klumpen übrig ist oder in einem nächsten Schritt gar keine Kontur mehr übrig wäre.

$$K \rightarrow K' \rightarrow K'' \rightarrow K''' \rightarrow \dots \rightarrow \emptyset \quad (3.12)$$

Dieser Prozess beschreibt nur genau eine Möglichkeit, von einer Ausgangskontur K zu einer Kontur mit nur einem Klumpen zu gelangen.

Werden alle möglichen Pfade erzeugt, die von K zu Konturen mit nur einem Klumpen führen, so erhält man das in [KB96] sogenannte *Segmentierungsgitter* (engl. *segmentation lattice*) $M(K)$, das in Abbildung 3.8a gezeigt ist. K ist dann die *Wurzel* von $M(K)$. Wird jede Kontur in $M(K)$ durch ihren G-Graphen ersetzt, so erhält man das in Abbildung 3.8b gezeigte sogenannte \rightarrow *Morphismusgitter* (engl. *morphism lattice*) $L(G(K))$.

Zur Ähnlichkeitsbestimmung von 2 Konturen K_1 und K_2 werden ihre Segmentierungs- bzw. Morphismusgitter erzeugt. Die G-Graphen zweier Mitglieder K_1^* , K_2^* von $M(K_1)$ bzw. $M(K_2)$ werden als *isomorph* bezeichnet, wenn sie die gleiche Anzahl an Knoten und Kanten besitzen und eine \rightarrow *bijektive* Abbildung existiert, die die Knoten des einen G-Graphen auf die Knoten des anderen G-Graphen abbildet, so dass auch die korrespondierenden Kanten und Nicht-Kanten aufeinander abgebildet werden (siehe Abbildung 3.9).

Diejenigen Konturen K_1^* , K_2^* aus $M(K_1)$ und $M(K_2)$, deren G-Graphen $G(K_1^*)$, $G(K_2^*)$ isomorph sind, sind strukturell gleich. Wie in Abbildung 3.9 zu sehen, können 2 strukturell gleiche Konturen trotzdem relativ unähnlich sein, d.h. diese Unähnlichkeit muss von einem Ähnlichkeitsmaß berücksichtigt werden. Zusätzlich muss in ein Ähnlichkeitsmaß eingehen, wie viel Strukturvereinfachung nötig war, um von den Wurzeln K_1 , K_2 zu den strukturgleichen Konturen K_1^* , K_2^* zu gelangen. Aus diesem Grunde werden all die Konturen, deren G-Graphen isomorph sind, einer weiteren Analyse unterzogen. Zu diesem Zweck werden die den Knoten der G-Graphen zugeordneten Gewichte ausgewertet. Diese beinhalten

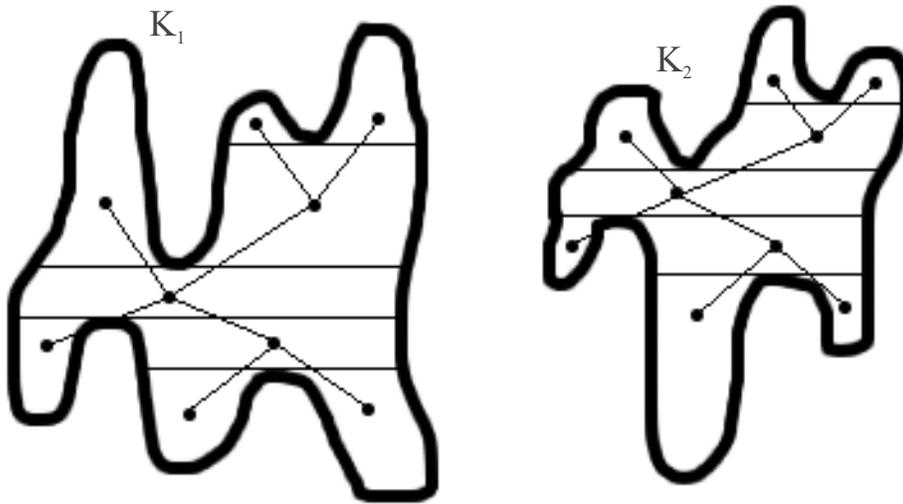


Abbildung 3.9: Beispiel für Isomorphismus. Die beiden Konturen K_1' und K_2' besitzen in die Konturen eingezeichnete G-Graphen, die $\rightarrow isomorph$ sind. Obwohl die G-Graphen isomorph sind, können die Konturen doch sehr unterschiedlich aussehen.

lokales "Wissen" über die einzelnen Klumpen. In [KB96] wird folgendes Grundähnlichkeitsmaß für 2 Konturen in y -Richtung, d.h. für 2 Konturen in einer ganz bestimmten Rotationslage, vorgeschlagen:

$$sim_y(K_1, K_2) = \min F(K_1^*, K_2^*), \quad (3.13)$$

mit

$$F(K_1^*, K_2^*) = c(D(K_1^*) + D(K_2^*)) + d(K_1^*, K_2^*), \quad (3.14)$$

und

$$d(K_1^*, K_2^*) = \sum_{v \in V(G(K_1^*))} |w(v) - w(I(v))|. \quad (3.15)$$

Darin ist $V(G(K_1^*))$ die Menge aller Knoten von $G(K_1^*)$ und I ist der Isomorphismus zwischen den Graphen $G(K_1^*)$ und $G(K_2^*)$, d.h. I beschreibt, welche Knoten der beiden G-Graphen miteinander korrespondieren.

$d(K_1^*, K_2^*)$ beschreibt also die Unterschiedlichkeit von Konturen K_1^* und K_2^* gleicher Struktur mit Hilfe der Gewichte der korrespondierenden Klumpen.

$D(K_1^*)$ gibt die Summe der Gewichte aller Blattklumpen an, die auf dem Pfad durch das Segmentierungsgitter von der Wurzel K_1 bis zu K_1^* abgeschnitten wurden. Diese Summe ist unabhängig von der Wahl des Pfades. Analog gilt dies für $D(K_2^*)$. Damit gibt der Teil $D(K_1^*) + D(K_2^*)$ in Gleichung (3.14) an, um wieviel

die Strukturen von K_1 und K_2 , gewichtet mit den Gewichten der weggeschnittenen Klumpen, vereinfacht werden mussten, so dass $G(K_1^*)$ und $G(K_2^*)$ isomorph werden.

Die Konstante c in Gleichung (3.14) stellt einen Parameter dar, mit dem der strukturelle Ähnlichkeitsanteil $D(K_1^*) + D(K_2^*)$ gegenüber $d(K_1^*, K_2^*)$ gewichtet werden kann.

Mit Gleichung (3.13) wird dann dasjenige Paar K_1^*, K_2^* im Segmentierungsgitter $M(K)$ detektiert, für das F minimal ist.

Damit ist das in [KW96], [KB96] beschriebene Prinzip eines Ähnlichkeitsmaßes basierend auf G-Graphen erklärt. In [KB96] wird dieses Ähnlichkeitsmaß noch weiter verfeinert und ein Verfahren zur Rotationsinvarianz beschrieben. Skalierungsinvarianz wurde dadurch erreicht, dass alle Konturen auf die gleiche Länge skaliert wurden.

3.3 Syntaktische Anpassung

Im Bereich von 1-dimensionalen Funktionen und als solche lässt sich eine Außenkontur auffassen (siehe Abschnitt 2.2.3), wenn ihre Konturpunkte in einer geordneten Liste vorliegen, versteht man unter einer syntaktischen Repräsentation einer Kontur eine **geordnete** Liste von Konturabschnitten, denen Attribute wie z.B. Länge, Krümmungswinkel usw. zugeordnet sind.

Viele syntaktische Anpassungsmethoden machen Gebrauch von Methoden aus dem Zeichenkettenvergleich (engl. string matching), in der durch sogenannte Editieroperationen Zeichen ersetzt, gelöscht oder eingefügt werden (z.B. [SK83], [CL96]). Beim Konturvergleich sind die einzelnen Konturabschnitte ein Synonym für die einzelnen Zeichen. Um eine Kontur K in eine andere Kontur K^* zu überführen (zu deformieren) ist es zum einen notwendig, bestimmte Konturabschnitte in K^* einzufügen oder aus K zu entfernen, da 2 Konturen i. allg. aus einer unterschiedlichen Anzahl an Konturabschnitten bestehen, und zum anderen Konturabschnitte von K^* durch Konturabschnitte von K zu ersetzen.

Den Editieroperationen Einfügen, Entfernen und Ersetzen von Konturabschnitten werden Kosten zugeordnet. Die Summe dieser Kosten wird dann minimiert.

Ein Beispiel für eine syntaktische Repräsentation ist folgende:

Die syntaktische Repräsentation einer Kontur K wird dadurch erreicht, dass Punkte P_i , die sich an Stellen maximaler Krümmung befinden, durch Geraden miteinander verbunden werden (Abbildung 3.10). Die Kontur K besteht dann aus M geordneten Geradensegmenten, die entweder der Kontur linksherum oder rechtsherum folgen:

$$K = \{k_1, k_2, \dots, k_i, \dots, k_M\}. \quad (3.16)$$

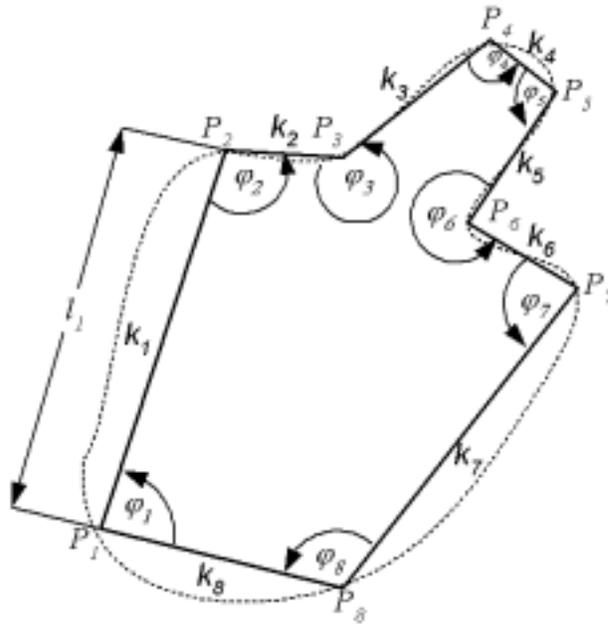


Abbildung 3.10: Beispiel für die syntaktische Darstellung einer Kontur. Die syntaktische Darstellung der gestrichelten Kontur wird dadurch erreicht, dass Punkte P_i , die sich an Stellen maximaler Krümmung befinden, detektiert und durch Geradenstücke k_i miteinander verbunden werden. Wichtig ist, dass die Geradenstücke in einer geordneten Reihenfolge vorliegen. Den Geradensegmenten werden zusätzlich Attribute zugeordnet. Hier sind dies die Länge l_i der Geradensegmente und die beiden Winkel, die das Geradensegment mit den beiden benachbarten Geradenstücken einschließt. Für das Geradensegment k_1 sind dies beispielsweise die Länge l_1 und die Winkel φ_1 und φ_2 .

Jedem Segment k_i werden als Attribute die Länge l_i des Geradensegmentes und die Krümmungswinkel φ_i und φ_{i+1} , die das Geradensegment k_i mit dem vorhergehenden und dem nachfolgenden Segment k_{i-1} bzw. k_{i+1} einschließt, zugeordnet:

$$k_i := \{l_i, \varphi_i, \varphi_{i+1}\} \quad (3.17)$$

Um die Punkte P_i an den Stellen maximaler Krümmung zu bestimmen, ist es zunächst notwendig, die Krümmungen der Kontur an jedem einzelnen der m Punkte p_i der Kontur

$$K = \{p_1, p_2, \dots, p_i, \dots, p_m\} \quad (3.18)$$

zu bestimmen. Dazu wird zu jedem Punkt p_i ein Winkel ρ_i berechnet, den die Vektoren \vec{v}_1 und \vec{v}_2 einschließen (Abbildung 3.11). Der Vektor \vec{v}_1 ergibt sich aus

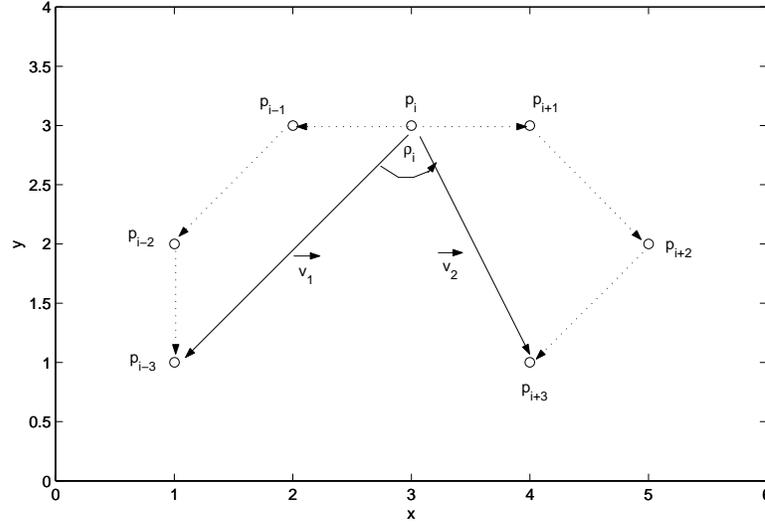


Abbildung 3.11: Beispiel für die Krümmungsschätzung in einem Punkt. Um den Krümmungswinkel ρ_i zu einem Punkt p_i zu bestimmen, werden zunächst die Vektoren \vec{v}_1 und \vec{v}_2 berechnet. \vec{v}_1 ergibt sich aus der vektoriellen Summe der $q = 3$ Vektoren zur Linken von p_i und der Vektor \vec{v}_2 ergibt sich entsprechend aus der vektoriellen Summe der $q = 3$ Vektoren zur Rechten von p_i . Der von \vec{v}_1 und \vec{v}_2 eingeschlossene Winkel ist dann der geschätzte Krümmungswinkel ρ_i .

der vektoriellen Summe der q Vektoren $p_{i-1} - p_i, \dots, p_{i-q} - p_{i-q+1}$ zur Linken von p_i und der Vektor \vec{v}_2 ergibt sich entsprechend aus der vektoriellen Summe der q Vektoren $p_{i+1} - p_i, \dots, p_{i+q} - p_{i+q-1}$ zur Rechten von p_i :

$$\begin{aligned}\vec{v}_1 &= \sum_{r=1}^q (p_{i-r} - p_{i-r+1}) \\ \vec{v}_2 &= \sum_{r=1}^q (p_{i+r} - p_{i+r-1})\end{aligned}\quad (3.19)$$

Der auf diese Weise gefundene Winkel ρ_i wird als Schätzung für die Krümmung am Punkt p_i herangezogen. Aus den Maxima und Minima der durch die ρ_i bestimmten diskreten Krümmungsfunktion ergeben sich die Punkte P , die zur Segmentierung der Kontur K herangezogen werden.

Eine Kontur K^*

$$K^* = \{k_1^*, k_2^*, \dots, k_j^*, \dots, k_N^*\}, \quad (3.20)$$

die mit Kontur K verglichen werden soll, besteht aus N Segmenten k_j^* , die entsprechend den Segmenten der Kontur K links oder rechts herum geordnet sind.

Jedem Segment werden äquivalent zu den Segmenten der Kontur K die Attribute zugeordnet:

$$k_j^* := \{l_j^*, \varphi_j^*, \varphi_{j+1}^*\}. \quad (3.21)$$

Um die Kosten für das **Ersetzen** des Geradensegmentes k_j^* durch das Geradensegment k_i zu bestimmen, ist eine Analyse der Unähnlichkeit der beiden Segmente notwendig. Die Geradensegmente unterscheiden sich zum einen durch ihre Länge und zum anderen durch die beiden Winkel, die sie mit ihren beiden benachbarten Geradensegmenten einschließen. D.h. es bietet sich an, zunächst 2 getrennte Kostenfunktionen für die Längen und für die Winkel festzulegen.

Längenkosten

Eine Möglichkeit l_i mit l_j^* zu vergleichen ist es ihr Verhältnis festzustellen. Sind die beiden Längen gleich, beträgt ihr Verhältnis 1. Eine potentielle Kostenfunktion ist die folgende:

$$c_{ers}^l = \begin{cases} 1 - \frac{l_i}{l_j^*} & , \text{ falls } l_i \leq l_j^* \\ 1 - \frac{l_j^*}{l_i} & , \text{ falls } l_j^* < l_i \end{cases} \quad (3.22)$$

Durch diese Form der Kostenfunktion c_{ers}^l ist sichergestellt, dass

$$c_{ers}^l(l_i, l_j^*) = c_{ers}^l(l_j^*, l_i) \quad (3.23)$$

ist, d.h. c_{ers}^l ist symmetrisch. Ferner gilt für $l_i = l_j^*$, dass $c_{ers}^l = 0.0$ ist, und sich für $l_i \gg l_j^*$ bzw. umgekehrt $c_{ers}^l = 1.0$ ergibt. Es gilt:

$$0.0 \leq c_{ers}^l \leq 1.0. \quad (3.24)$$

Diese Kostenfunktion ist rotations- und translationsinvariant. Skaliert man die Gesamtlängen der zu vergleichenden Konturen oder genauer der Polygone zunächst auf gleiche Längen, so ist die Kostenfunktion auch skalierungsinvariant.

Winkelkosten

Die Winkel φ_i, φ_{i+1} und $\varphi_j^*, \varphi_{j+1}^*$ sind ein Synonym für die Krümmungen an den Verbindungsstellen der Segmente k_i und k_j^* . Eine einfache Möglichkeit die Krümmungsunterschiede zu detektieren, ist den Betrag der korrespondierenden Winkel-differenzen zu bestimmen und aufzusummieren. Die Kostenfunktion

$$c_{ers}^\varphi = \frac{|\varphi_i - \varphi_j^*| + |\varphi_{i+1} - \varphi_{j+1}^*|}{4\pi} \quad (3.25)$$

gewährleistet Symmetrie

$$c_{ers}^\varphi(\{\varphi_i, \varphi_{i+1}\}, \{\varphi_j^*, \varphi_{j+1}^*\}) = c_{ers}^\varphi(\{\varphi_j^*, \varphi_{j+1}^*\}, \{\varphi_i, \varphi_{i+1}\}) \quad (3.26)$$

und die Kosten bewegen sich zwischen 0.0 und 1.0 für 2 exakt gleiche bzw. 2 maximal unterschiedliche Winkelpaare. Es gilt also:

$$0.0 \leq c_{ers}^{\varphi} \leq 1.0. \quad (3.27)$$

c_{ers}^{φ} ist translations-, rotations- und skalierungsinvariant. c_{ers}^{φ} erfüllt allerdings nicht die Forderung 5 aus Abschnitt 1.1.1, die besagt, dass das Verbiegen eines Objektes an Stellen hoher Krümmung weniger Kosten verursachen sollte, als das Biegen an Stellen mit wenig Krümmung, da dies vermutlich in der menschlichen Ähnlichkeitsbeurteilung eine Rolle spielt.

Die Teilkosten für den Längenunterschied und die Winkelunterschiede können dann mit folgender Kostenfunktion für das Ersetzen zusammengefasst werden:

$$c_{ers} = w_1 c_{ers}^l + (1 - w_1) c_{ers}^{\varphi} \quad \text{mit} \quad 0.0 \leq w_1 \leq 1.0. \quad (3.28)$$

Mit w_1 können die einzelnen Anteile der Kostenfunktion gegeneinander gewichtet werden. Durch die Form von Gleichung (3.28) und die Gleichungen (3.22), (3.25) ist gewährleistet, dass gilt:

$$0.0 \leq c_{ers} \leq 1.0. \quad (3.29)$$

Das **Einfügen** von Abschnitten in eine Kontur K^* kann aufgefasst werden als das Ersetzen eines einzelnen Segmentes der einen Kontur K^* durch mehrere benachbarte Segmente der anderen Kontur K . Wird das Segment k_j^* durch r benachbarte Geradensegmente $k_i, k_{i+1}, \dots, k_{i+r-1}$ ersetzt, so ergeben sich die Kosten für das Einfügen dieser Segmente als Summe der Einzelersetzungen:

$$c_{ein} = \sum_{k=0}^{r-1} c_{ers}(k_{i+k}, k_j^*). \quad (3.30)$$

Das **Entfernen** von Abschnitten aus einer Kontur K^* kann diametral zum Einfügen aufgefasst werden als das Ersetzen mehrerer benachbarter Segmente der Kontur K^* durch ein einzelnes Segment der anderen Kontur K . Werden r benachbarte Segmente $k_j^*, k_{j+1}^*, \dots, k_{j+r-1}^*$ ersetzt durch ein Geradensegment k_i , so ergeben sich die Kosten für das Entfernen dieser Segmente als Summe der Einzelersetzungen:

$$c_{ent} = \sum_{k=0}^{r-1} c_{ers}(k_i, k_{j+k}^*). \quad (3.31)$$

Da die Gleichungen (3.30) und (3.31) auch für $r = 1$ gelten, behandeln sie das Ersetzen von Segmenten gleich mit, d.h. Einfügen, Entfernen und reines Ersetzen werden in diesen Kostenfunktionen miteinander vermischt und es gilt

$$c_{ers} = c_{ein} = c_{ent} \quad \text{für} \quad r = 1. \quad (3.32)$$

Um die Kontur K in die Kontur K^* umzueditieren, müssen der Kontur K^* sowohl Segmente entnommen und hinzugefügt werden als auch Segmente ersetzt werden. D.h. um die minimalen Kosten C für das Umeditieren zu bestimmen, muss eine Kombination der Summen in den Gleichungen (3.30), (3.31) gefunden werden, die in ihrer Summe minimal wird.

Dieses komplexe Optimierungsproblem stellt sich in ganz ähnlicher Form auch im Kapitel 4 und wird dort als vorbereitender Schritt für die Synthese signifikanter Konturabschnitte benötigt. Dort wird das Problem mit Hilfe der bereits erwähnten *dynamische Programmierung* gelöst. Aus diesem Grund wird diese Optimierungsmethode dort und in Anhang C näher erläutert.

3.4 Elastische Anpassung

Die Methoden, die auf der elastischen Anpassung basieren (engl. elastic matching), bestimmen die Ähnlichkeit zwischen 2 Konturen aufbauend auf einer spezifischen Korrespondenz zwischen ihren Konturpunkten (siehe dazu Abbildung 3.12). Da Konturen i. allg. nicht die gleiche Anzahl an Konturpunkten besitzen, sind sowohl 1-1-Korrespondenzen, d.h. genau ein Konturpunkt der einen Kontur korrespondiert mit genau einem Konturpunkt der anderen Kontur, als auch 1-n-Korrespondenzen, d.h. genau ein Konturpunkt der einen Kontur korrespondiert mit mehreren Konturpunkten der anderen Kontur und umgekehrt, zugelassen. D.h. aber auch, dass elastische Anpassungsmethoden das *Dehnen* (engl. *stretching*) zulassen, so dass längere Konturabschnitte der einen Kontur auf kürzere Konturabschnitte der anderen Kontur abgebildet werden können und umgekehrt. Zusätzlich wird bei elastischen Anpassungsmethoden das *Biegen* (engl. *bending*) von Konturabschnitten zugelassen. Biegen und Dehnen zusammen ergeben dann die Deformation einer Kontur.

Die lokale Ähnlichkeit korrespondierender Konturpunkte wird mit Hilfe lokaler Ähnlichkeitsmaße bestimmt, die auf lokalen Eigenschaften dieser Punkte fußen, wie z.B. der Krümmung der Kontur an diesen Punkten. Die Gesamtähnlichkeit (globale Ähnlichkeit) der beiden Konturen ergibt sich dann aus der Summe der lokalen Ähnlichkeiten. Typischerweise wird diese Summe der lokalen Ähnlichkeiten selbst dafür benutzt, um die "beste" Korrespondenz zwischen den beiden Konturen zu finden. D.h., dass diejenige Deformation von einer Kontur in eine andere gesucht wird, die ein globales Ähnlichkeitsmaß maximiert.

Elastische Anpassungsmethoden erzeugen sowohl ein Maß (also eine Zahl) für die globale Ähnlichkeit als auch eine Korrespondenz, d.h. eine Abbildungsfunktion der Konturpunkte der Konturen aufeinander, zwischen den Konturen. Elastische Anpassungsmethoden besitzen also 2 Hauptkomponenten. Zum einen besitzen sie eine Ähnlichkeitsfunktion basierend auf einem lokalen Ähnlichkeits-

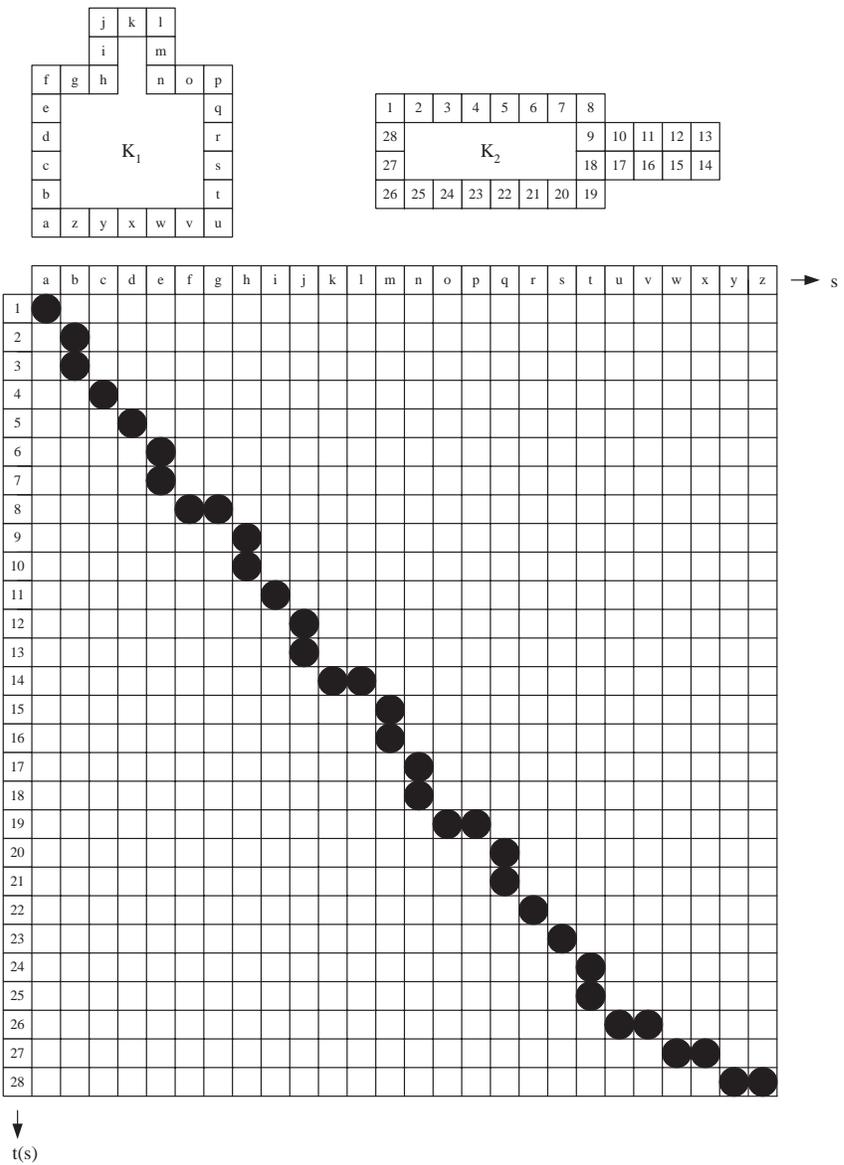


Abbildung 3.12: Die Konturpunkte der Kontur K_1 und K_2 sind fortlaufend indiziert mit a-z bzw. 1-28. a-z indizieren wiederum die Spalten und 1-28 die Zeilen des unter den Konturen abgebildeten Arrays. Da benachbarte Konturpunkte 4er-Nachbarn sind, sind alle Konturpunkte gleichweit voneinander entfernt. D.h. die Konturpunktindizes sind ein Synonym für die Bogenlänge s . Die schwarzen Kreise stellen die Korrespondenz einzelner Konturpunkte her. Damit ist die Gesamtheit der Kreise eine diskrete Funktion $t(s)$, die die Konturpunkte von K_1 auf die Konturpunkte von K_2 abbildet.

maß. Zum anderen benötigen sie einen Suchmechanismus, der nach der, im Sinne von maximaler Ähnlichkeit, besten Anpassung sucht.

Basri et al. ([BCGJ95], [BCGJ98]) schlagen folgende elastische Anpassungsmethode vor, die sie unter Berücksichtigung von vermuteten wichtigen Eigenschaften menschlicher Ähnlichkeitseinschätzung formulieren. Dies ist der Grund, warum sie in ihrer Methode darauf achten, dass Abbildungsfunktionen (Deformationen) gefunden werden, die die Teilestruktur der Konturen möglichst nicht verändert. Dies schlägt sich u.a. in den in Abschnitt 1.1.1 bereits vorgestellten Anforderungen an ein Ähnlichkeitsmaß nieder. Der Unterschied zu den teilbasierten Methoden besteht darin, dass hier die Stellen an denen Teile aneinandergrenzen berücksichtigt werden, ohne explizit die Teilstruktur der Kontur zu berechnen.

Eine Kontur K_1 wird parametrisiert mit ihrer Bogenlänge s . Daraus folgt, dass $K_1(s)$ einen Konturpunkt von K_1 darstellt. Die Korrespondenz zwischen den Konturen wird hergestellt durch eine Abbildungsfunktion $t(s)$ zwischen den Bogenlängen von K_1 und K_2 . Daraus folgt, dass der Konturpunkt $K_1(s)$ auf einen Konturpunkt $K_2(t(s))$ der Kontur K_2 abgebildet wird.

Bei gegebener Abbildungsfunktion $t(s)$ wird ein lokales Kosten³maß F aufintegriert. Das Minimum dieser Integrale über alle möglichen Abbildungsfunktionen $t(s)$ sind dann die minimalsten Kosten D für die Deformation einer Kontur in eine andere:

$$D(K_1, K_2) = \min_{K_1} \int F \left(c_1, c_2, \frac{dt}{ds} \right) ds \quad , \quad (3.33)$$

wobei es sich bei c_1 um die Krümmung der Kontur K_1 im Punkt $K_1(s)$ und bei c_2 um die Krümmung der Kontur K_2 im Punkt $K_2(t(s))$ handelt. $\frac{dt}{ds}$ (siehe Abbildung 3.12) kann aufgefasst werden als die relative Geschwindigkeit des Fortschritts auf beiden Konturen und damit als lokale Dehnungen von K_1 gegenüber K_2 . Die Kosten D sind also der minimale Aufwand an lokaler Biegung, d.h. der Änderungen der Krümmungen einer Kontur K_1 , und an Dehnung von K_1 , um diese mit Kontur K_2 identisch zu machen.

Eine mögliche lokale Kostenfunktion F für Gleichung (3.33) basiert auf dem Prinzip der *kontinuierlichen Deformation*. Dieser Methode liegt folgende Vorstellung zugrunde:

Die gesamten Kosten des Biegens und Dehnens für das Überführen (Deformieren) einer Kontur K_1 in eine andere Kontur K_2 ergeben sich aus der Summe einer unendlichen Anzahl \rightarrow *infinitesimal* kleiner Deformationen. Das bedeutet, dass K_1 und K_2 durch eine kontinuierliche Folge von Konturen miteinander ver-

³Da die Maximierung einer Ähnlichkeitsfunktion gleichzusetzen ist mit der Minimierung einer Kostenfunktion (Unähnlichkeitsfunktion), ist in diesem Abschnitt ab jetzt nur noch von Kosten die Rede.

bunden sind und die Gesamtkosten der Deformation die Summe der Deformationen von einer Kontur zur nächsten in dieser Folge ist.

Vorgeschlagen wird folgende lokale Kostenfunktion:

$$F\left(c_1, c_2, \frac{dt}{ds}\right) = \left(\frac{dt}{ds} + 1\right) \left| \int_{c_1}^{c_2} e^{-\alpha c} dk \right| + \lambda \left| \frac{dt}{ds} - 1 \right|. \quad (3.34)$$

Für den Fall des reinen Dehnens, d.h. $c_1 = c_2$, folgt aus Gleichung (3.34)

$$F\left(c_1, c_2, \frac{dt}{ds}\right) = \lambda \left| \frac{dt}{ds} - 1 \right| \quad (3.35)$$

und für den Fall des reinen Biegens, d.h. $\frac{dt}{ds} = 1$, folgt aus Gleichung (3.34)

$$F\left(c_1, c_2, \frac{dt}{ds}\right) = 2 \left| \int_{c_1}^{c_2} e^{-\alpha c} dk \right|. \quad (3.36)$$

$e^{-\alpha c}$ gibt die Kosten an, die notwendig sind, um einen Konturpunkt infinitesimal zu deformieren. Durch die Exponentialfunktion ist gewährleistet, dass je größer die Krümmung c in diesem Punkt ist, desto kleiner die Kosten für die Deformation dieses Punktes sind. Damit wird Bedingung 5 aus Abschnitt 1.1.1 erfüllt. Mit dem Parameter α kann das Abklingverhalten der Exponentialfunktion beeinflusst werden.

Integriert man von der Krümmung c_1 in einem Punkt auf K_1 bis zur Krümmung c_2 im korrespondierenden Punkt auf K_2 , so repräsentiert das Integral die lokalen Gesamtkosten des Verbiegens.

Bei λ handelt es sich um einen positiven Parameter, der den Dehnanteil an den Kosten gegenüber den Biegekosten gewichtet.

Es kann gezeigt werden, dass eine Kostenfunktion D (Gleichung (3.33)) basierend auf einer lokalen Kostenfunktion F (Gleichung (3.34)) für die kontinuierliche Deformation alle Forderungen an eine Kostenfunktion aus Abschnitt 1.1.1, bis auf Bedingung 7, erfüllt.

Basri et al. schlagen noch 2 weitere lokale Kostenfunktionen in D vor. Zum einen

$$F\left(c_1, c_2, \frac{dt}{ds}\right) = \frac{\left|c_2 \frac{dt}{ds} - c_1\right|^2}{\left|c_2 \frac{dt}{ds}\right| + |c_1|} + \lambda \frac{\left(\frac{dt}{ds} - 1\right)^2}{\frac{dt}{ds} + 1}, \quad (3.37)$$

die auf einem physikalischen Federmodell (\rightarrow *Hooke'sches Federgesetz*) beruht und zum anderen eine lineare lokale Kostenfunktion

$$F\left(c_1, c_2, \frac{dt}{ds}\right) = \left|c_2 \frac{dt}{ds} - c_1\right| + \lambda \left| \frac{dt}{ds} - 1 \right|, \quad (3.38)$$

die aus einer verallgemeinerten Form des physikalischen Federmodells hervorgeht. Wird die lokale Kostenfunktion F aus Gleichung 3.37 in D (Gleichung (3.33)) eingesetzt, erfüllt D alle Forderungen aus Abschnitt 1.1.1, bis auf Forderung 2. Wird die lokale Kostenfunktion F aus Gleichung 3.38 in D (Gleichung (3.33)) eingesetzt, erfüllt D alle Forderungen aus Abschnitt 1.1.1 mit Ausnahme von Forderung 5.

Zur Suche nach den minimalen Kosten wurde die \rightarrow *dynamische Programmierung* (siehe auch Anhang C) benutzt. Im diskreten Fall werden die Krümmungen der Punkte mit Hilfe der Tangentenrichtungsänderungen aufeinanderfolgender Punkte geschätzt. Die Tangentenrichtung selbst wird mit Hilfe der Richtungen zwischen 7 aufeinanderfolgenden Punkten, mit dem interessierenden Punkt in deren Mitte, ebenfalls geschätzt.

3.5 Diskussion

In diesem Kapitel wurden beispielhaft verschiedene Ansätze zur Bestimmung der Ähnlichkeit deformierter Konturen vorgestellt. Alle Verfahren ziehen lokale Information zur Analyse heran. Die Verfahren, die psychophysikalisches Vorwissen in die Analyse mit einfließen lassen, wie z.B. das Herausheben von Stellen, an denen Konturen hohe Krümmungen aufweisen, sind leistungsstark bezogen auf die Erkennungsleistung.

Ein sehr schönes, schnelles und mathematisch fundiertes Verfahren ist das Ähnlichkeitsbestimmungsverfahren basierend auf der diskreten Kurvenevolution von Lakämper und Latecki. Der Nachteil bei diesem Verfahren liegt darin, dass noch nicht abschließend geklärt ist, bei welcher Stufe der Evolution einer Kontur jeweils abgebrochen werden sollte. So kann es passieren, dass eventuell zur Erkennung lokal wichtige Konturabschnitte „wegevolutioniert“ werden.

Interessant auch das Federmodell von Basri et al., da es die physikalische Verformung von Objekten modelliert. Der Hauptnachteil dieser Methode liegt allerdings in der störungsanfälligen Berechnung von Ableitungen.

Die Präsentation einer Kontur durch Graphen wie z.B. die vorgestellte Methode von Kupeev und Wolfson hat vor allen Dingen Nachteile, da diese zum einen sehr anfällig gegen die Verdeckung von Teilen der Objekte ist, d.h. dass dadurch komplett andere Graphen entstehen können. Ferner ist bei Graphenproblemen das \rightarrow *NP-vollständige* und damit aufwendige \rightarrow *Graph-Isomorphismus-Problem* zu lösen. Bei Kupeev und Wolfson kommt zusätzlich das Problem hinzu, dass das Graph-Isomorphismus-Problem mehrfach gelöst werden muss, um rotationsinvariant zu sein.

Die syntaktischen Methoden setzen als Vorwissen die natürliche Ordnung der Konturpunkte voraus. Dadurch lassen sich gutverstandene Methoden aus dem Zei-

chenkettenvergleich heranziehen, die Editieroperationen (Ersetzen, Löschen, Einfügen) verwenden.

Allen Methoden gemeinsam ist, dass sie die globale Ähnlichkeit von Konturen bestimmen. Sie benutzen dazu bei der Klassifikation nur implizit lokale Konturabschnitte. D.h. es geht keine Analyse dahingehend voraus, welche expliziten charakteristischen lokalen Konturabschnitte ausreichen würden, um diese Konturen genauso gut oder vielleicht sogar besser korrekt ihren Klassen zuzuordnen zu können. Das letztgenannte Problem wird bisher im allgemeinen recht stiefmütterlich behandelt. Dem Autor sind bis auf [PK00a], [PK00d], [PK02] und [KP02] aus der Literatur jedenfalls keine Verfahren bekannt, die systematisch lokale Konturabschnitte aus Beispielkonturen automatisch generieren, die ausreichen, um Konturen ausreichend zu beschreiben und mit diesen zu klassifizieren.

Kapitel 4

Merkmalsynthese

In den vorangegangenen Abschnitten wurden verschiedene Verfahren zur Ähnlichkeitsanalyse von Außenkonturen vorgestellt. Diese hatten alle zum Ziel, die globale Ähnlichkeit der Konturen mit Hilfe der gesamten Kontur zu bestimmen.

Die Verfahren aus Kapitel 3 beziehen zwar alle lokale Informationen zur Ähnlichkeitsbestimmung mit ein, letztendlich wurde aber die Gesamtkontur zum Vergleich herangezogen.

Wie zu Beginn des Kapitels 3 schon angedeutet wurde, sind für eine hohe Ähnlichkeit von Konturen aber in erster Linie charakteristische lokale Konturabschnitte verantwortlich, die durch eine Deformation nur schwach verändert werden, d.h. die in Grenzen invariant sind¹. D.h. aber, dass nur diese Konturabschnitte notwendig sein müssten, um die Ähnlichkeit von Konturen zu bestimmen. Sind diese lokalen Konturabschnitte in einer zu klassifizierenden Kontur nur leicht deformiert vorhanden, so müsste dies genauso zu hoher Ähnlichkeit führen. Diese Aussage wird dadurch gestützt, dass Lebewesen in der Lage sind Objekte, die teilweise verdeckt sind, zu erkennen, falls nur bestimmte Objektabschnitte zu sehen sind. Diese Vermutung wird beispielsweise durch eine kleine Versuchsreihe in [Yin98] bestärkt, in der die Versuchspersonen nur kleine Ausschnitte von Objekten zu sehen bekamen und diese klassifizieren sollten. In diesem Kapitel wird ein Verfahren vorgestellt, das solche charakteristischen lokalen Konturabschnitte **automatisch** generiert (siehe auch [PK99], [PK00a], [PK00b], [PK00d], [PK02], [KP02]).

Viele der in Kapitel 3 vorgestellten Verfahren und Prinzipien basieren in irgendeiner Form darauf, psychophysikalische Erkenntnisse oder Vermutungen in ihre Analyseverfahren zu integrieren. So bauen die teilbasierten Verfahren sehr stark darauf auf, aber auch die anderen Verfahren basieren darauf, dass Kon-

¹In [GW99] wird in diesem Zusammenhang von *schwach ähnlichen Konturen* (engl. *weakly similar curves*) gesprochen, falls die Konturen nur wenig deformiert sind. In [GW99] wird dieser Ausdruck aber auch nur auf die Gesamtkontur bezogen.

turpunkte, an denen hohe Krümmungen auftreten, wichtige Konturpunkte sind. Zweifelsohne sind mit psychophysikalischen Ansätzen große Fortschritte im Bereich der Formanalyse erzielt worden, trotzdem basiert die hier vorgestellte Methode nicht auf einem solchen Ansatz.

Der Grund hierfür liegt darin, dass diese Arbeit einen Beitrag für ein allgemeineres Verständnis von Mustererkennung liefern möchte. Dies soll bedeuten, dass der hier vorgestellte Ansatz für Außenkonturen, d.h. im Prinzip für eindimensionale Funktionen, Erkenntnisse auch für das Erlernen von anderen 1- oder n-dimensionalen Funktionen, wie z.B. Sprachsignalen oder Farbbildern liefern soll. Wie sich herausstellen wird, ergeben sich so Einblicke in das Erlernen von Objektcharakteristiken, wie z.B. auch (aber nicht ausschließlich) charakteristischen Teilen, die durch Stellen hoher konkaver Krümmung begrenzt sind, wie die teilbasierten Verfahren voraussetzen. Aus diesem Grunde will das hier vorgestellte Verfahren bzgl. der Erkennungsrate nicht in erster Linie mit anderen spezialisierten Konturvergleichsverfahren in Konkurrenz treten, wird sich dem Vergleich mit den Ergebnissen dieser Verfahren aber trotzdem stellen.

4.1 Überblick über das Verfahren

Diese Arbeit stellt ein erstes vollständiges Lern- und Klassifikationssystem vor, das *selbständig* und mit nur *minimalem Vorwissen* lokale charakteristische Merkmale, sogenannte *signifikante Konturabschnitte*, einer Objektklasse basierend auf deren Außenkonturen bestimmt (siehe Beispiel in Abb. 4.1). Aufeinanderfolgende Konturpunkte (Basismerkmale) werden aufgrund zu bestimmender Eigenschaften automatisch zu Konturpunktgruppen (Merkmalsgruppen) zusammengefasst (Synthese). Aus diesem Grund wird auch von *automatischer Merkmalsynthese* gesprochen. Mit Hilfe solch synthetisierter signifikanter Konturabschnitte werden Objekte verschiedenen Klassen zugeordnet.

Einen groben Überblick über das System gibt Abbildung 4.2. Eine Aufgabendomäne D besteht aus einer Menge interessierender Objekte in verschiedenen Klassen. D wird aufgeteilt in eine Lernstichprobe L , die zum Erlernen signifikanter Konturabschnitte und zum Einlernen eines Klassifikators notwendig ist, in eine Teststichprobe T , die unabhängig von L ist und zum Testen des Klassifikators herangezogen wird, und den Rest R :

$$D = \{L, T, R \mid L \cap T = \emptyset, L \cap R = \emptyset, T \cap R = \emptyset\}. \quad (4.1)$$

Die Lernstichprobe besteht aus Konturen in unterschiedlichen Klassen. In der Lernphase 1 werden jeweils für die Konturen aus einer bestimmten Klasse signifikante Konturabschnitte erlernt, die die Konturen dieser Klasse von den Konturen jeweils aller anderen Klassen der Lernstichprobe unterscheiden.

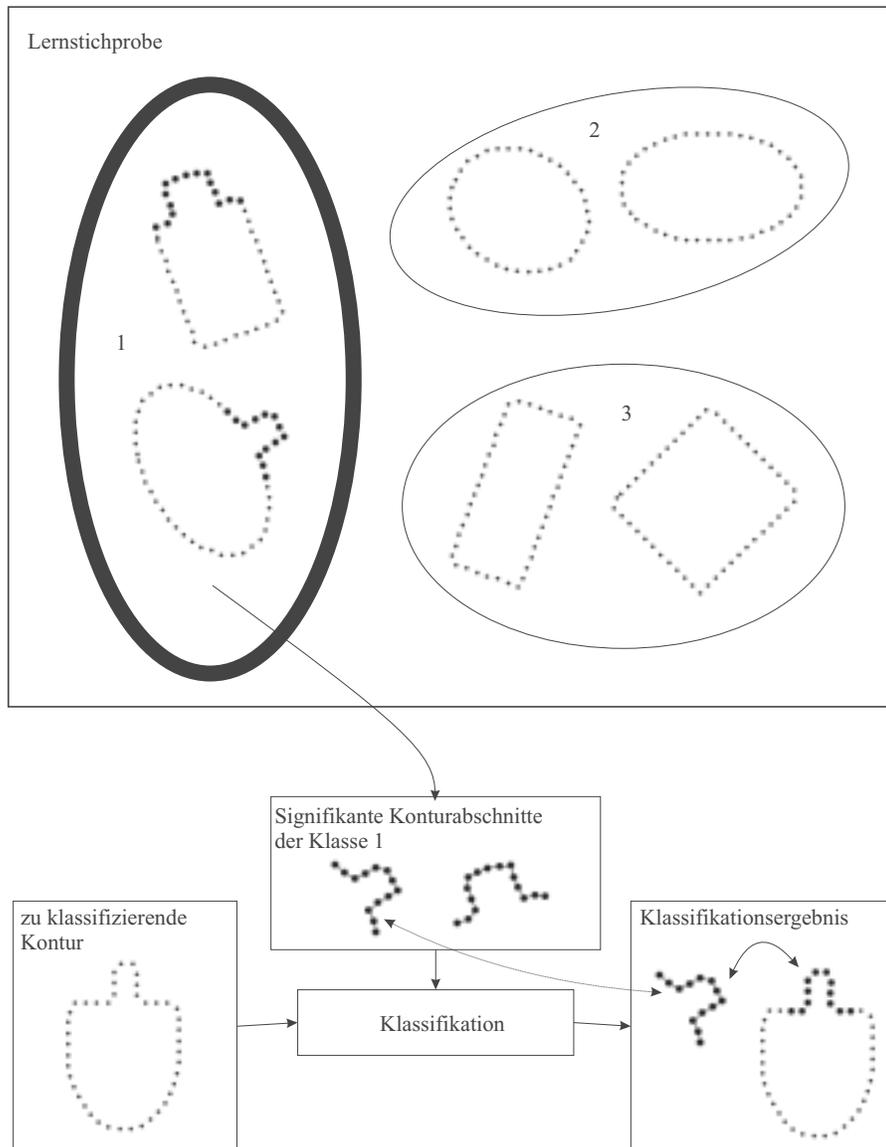


Abbildung 4.1: Einführungsbeispiel für signifikante Konturabschnitte. In diesem einfachen Beispiel besteht eine Lernstichprobe aus den Klassen 1,2, und 3. Diese Klassen bestehen aus je 2 einfachen geometrischen Objekten. Fett markiert sind diejenigen Konturpunkte der Konturen der Klasse 1, deren lokale Umgebungen sich stark von lokalen Umgebungen der Konturen in den anderen beiden Klassen unterscheiden. Sind solche Konturpunkte aufeinanderfolgend, so werden sie zu *signifikanten Konturabschnitte* zusammengefasst. Diese werden bei der Klassifikation in anderen Konturen gesucht und bei genügender Übereinstimmung der entsprechenden Klasse zugeordnet.

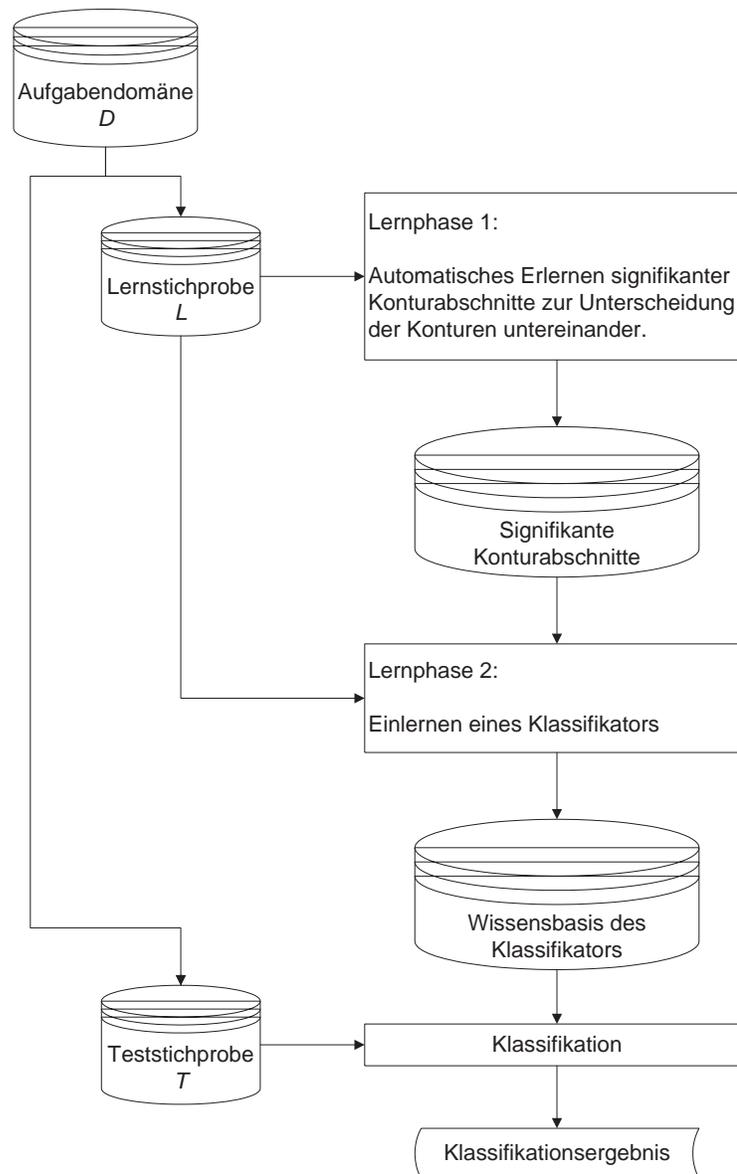


Abbildung 4.2: Grober Systemüberblick. Eine Aufgabendomäne D wird aufgeteilt in eine Lernstichprobe L , eine Teststichprobe T und einen Rest R . In 2 Lernphasen werden mit der Lernstichprobe L signifikante Konturabschnitte autonom erlernt und ein Klassifikator trainiert. Mit Hilfe der sich aus den Lernphasen ergebenden Wissensbasis werden die Objekte der Teststichprobe T klassifiziert. So lässt sich die Güte und Repräsentativität des gesamten Systems bzgl. des zu erwartenden Klassifikationsergebnisses für den Rest R abschätzen.

Die anschließende Lernphase 2 lernt einen Klassifikator auf Basis der signifikanten Konturabschnitte ein. Im einfachsten Fall besteht die Wissensbasis des Klassifikators danach ausschließlich aus den signifikanten Konturabschnitten und den korrespondierenden Klassen. Es können aber auch zusätzliche Attribute eingelesen werden, wie z.B. Rückweisungsschwellen, die dazu dienen eine Klassifikation zurückzuweisen, falls sich der Klassifikator in seiner Entscheidung unsicher ist. Die Konturen der Teststichprobe T werden basierend auf der Wissensbasis mit Hilfe eines Klassifikators klassifiziert. Es ergibt sich ein Klassifikationsergebnis, das die Güte und Repräsentativität des gesamten Systems bzgl. einer Klassifikation des Restes R abschätzt. Der Rest R ist i. allg. die Menge der Objekte, auf die man in der Lern- und Testphase keinen Zugriff hat, die aber beim Einsatz des Klassifikators trotzdem klassifiziert werden müssen. Beispielsweise hat man in der Lern- und Testphase eines Systems zur Klassifikation von Verpackungsmüll nur einen kleinen Ausschnitt dieser Aufgabendomäne zur Verfügung. Hier ist der Rest R der Müll, der vom Verbraucher in Zukunft entsorgt wird und die Mächtigkeit der Menge R ist nahezu unendlich viel größer als die Mächtigkeiten von L und T zusammengenommen.

4.1.1 Überblick Merkmalsynthese (Lernphase 1)

In Abbildung 4.3 ist ein Überblick über den Ablauf der automatischen Merkmalsynthese dargestellt, die in Abschnitt 4.3 detailliert beschrieben wird.

Alle Konturen der Lernstichprobe L werden paarweise miteinander verglichen. Dazu wird jedes Konturenpaar systematisch lokal untersucht. Jedem Konturpunktpaar werden Kosten für das Ersetzen des einen Konturpunktes durch den anderen zugeordnet (Abschnitt 4.3.1). Darauf basierend wird die kostengünstigste Deformation der einen Kontur in die andere berechnet (Abschnitt 4.3.2). Aus diesem Schritt ergibt sich für jedes Konturpaar eine kostengünstigste Abbildungsfunktion (Deformation) der Konturpunkte aufeinander und es stehen zusätzlich die lokalen Kosten für jedes Konturpunktpaar dieser Abbildung zur Verfügung.

Jetzt werden die Konturen der Lernstichprobe L in Klassen eingeordnet, da dies für die nachfolgenden Schritte von Bedeutung ist. Dies kann entweder überwacht durch einen Trainer oder unüberwacht basierend auf einer Analyse der Gesamtkosten zur Deformation, d.h. der globalen Ähnlichkeit, der einzelnen Konturen ineinander geschehen (Abschnitt 4.3.3).

Im nächsten Schritt werden die Abbildungsfunktionen jedes Konturpaares, deren Konturen aus verschiedenen Klassen stammen, auf Konturpunktpaare untersucht, die hohe Kosten zur Deformation beigetragen haben. Diese Konturpunkte sind wichtige Konturpunkte zur Unterscheidung von jeweils 2 Konturen. Aufeinanderfolgende wichtige Konturpunkte werden zu wichtigen Konturabschnitten synthetisiert (Abschnitt 4.3.4). Nach Abschluss dieses Schrittes stehen für jede

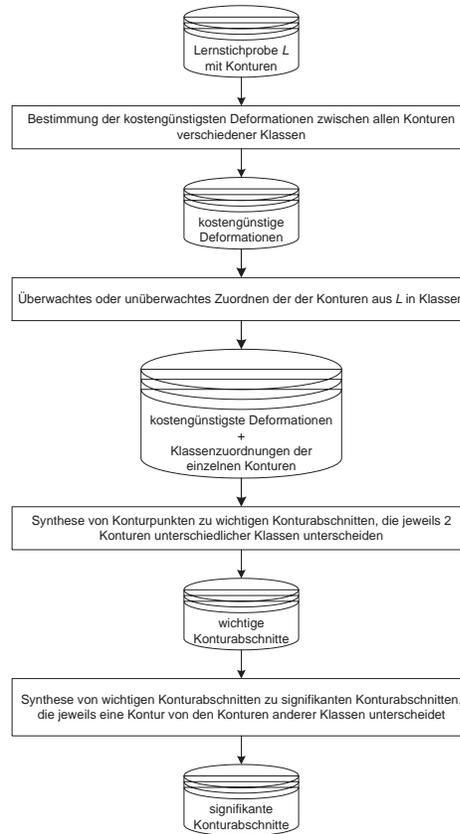


Abbildung 4.3: Überblick Merkmalsynthese (Lernphase 1): Von den Konturen der Lernstichprobe L zu deren signifikanten Konturabschnitten, die einzelne Konturen von Konturen der anderen Klassen unterscheiden.

Kontur wichtige Konturabschnitte bereit, die aus einem Vergleich mit allen Konturen aus den anderen Klassen herrühren und diese Konturen unterscheiden.

Der letzte Schritt der Lernphase 1 besteht darin, die wichtigen Konturabschnitte einer Kontur zusammenzufassen. Dabei werden sich überlappende wichtige Konturabschnitte, d.h. Konturabschnitte, die gemeinsame Konturpunkte besitzen, zu signifikanten Konturabschnitten synthetisiert (Abschnitt 4.3.5). Überlappende wichtige Konturabschnitte kommen dadurch zustande, dass eine Kontur mit allen Konturen der anderen Klassen verglichen wurde und sich ungefähr der gleiche Konturbereich von mindestens 2 Konturen unterscheidet. Am Ende der Lernphase 1 ergeben sich ein oder mehrere sich nicht überlappende signifikante Konturabschnitte für jede Kontur, die diese signifikant von den Konturen anderer Klassen unterscheiden.

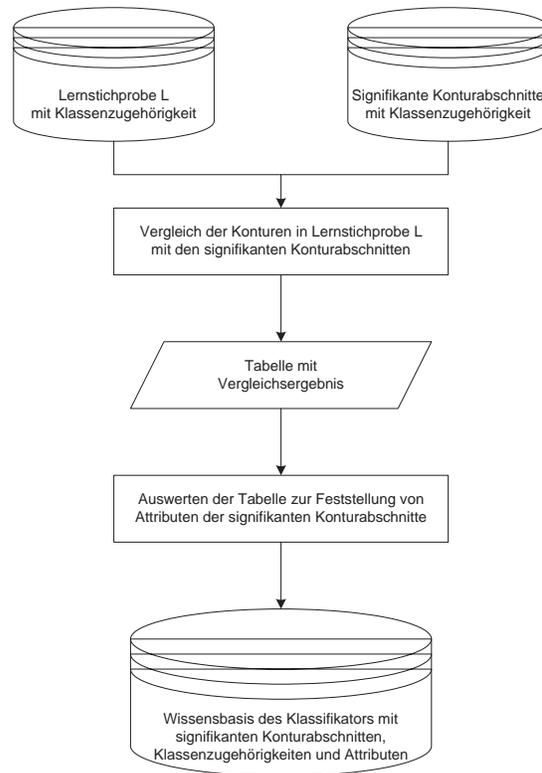


Abbildung 4.4: Überblick Klassifikatorsynthese (Lernphase 2): Den signifikanten Konturabschnitten aus Lernphase 1 werden, basierend auf einer systematischen Analyse, Attribute für die Klassifikation zugeordnet.

4.1.2 Überblick Klassifikatorsynthese (Lernphase 2)

Die Klassifikatorsynthese, die in Abschnitt 4.4 genau beschrieben wird, dient dem Einlernen des Klassifikators. Da der Klassifikator auf den automatisch erlernten signifikanten Konturabschnitten und deren Klassenzugehörigkeit aus Lernphase 1 basieren soll, werden diese mit Hilfe der Lernstichprobe L weiter analysiert. Einen Überblick über diese Analyse gibt Abbildung 4.4.

In einem ersten Schritt werden, basierend auf einer Kostenfunktion, die Konturabschnitte jeder Kontur der Lernstichprobe L mit jedem signifikanten Konturabschnitt verglichen (Abschnitt 4.4.1). Es gibt jeweils einen Konturabschnitt einer Kontur der Lernstichprobe L , der die geringsten Kosten beim Vergleich mit einem bestimmten signifikanten Konturabschnitt besitzt. Diese minimalsten Kosten werden in eine Tabelle eingetragen, die die Vergleichsergebnisse aller (Kontur)-(Signifikanter Konturabschnitt)-Paare beinhaltet.

Im zweiten und letzten Schritt wird diese Tabelle ausgewertet. Da durch die

Lernphase 1 auch die Klassenzugehörigkeiten der Konturen aus der Lernstichprobe L feststehen, kann die Vergleichstabelle durch einen Soll-Ist-Klassenvergleich ausgewertet werden. Diese Auswertung ergibt Attribute der signifikanten Konturabschnitte, wie z.B. die Verlässlichkeit dieser Attribute für ein korrektes Klassifikationsergebnis, oder es lassen sich Zurückweisungsschwellen abschätzen, die die Zuverlässigkeit der Klassifikation verbessern.

4.1.3 Überblick Klassifikation

Der in diesem System eingesetzte Klassifikator ist ein hierarchischer Klassifikator, der ebenfalls in Abschnitt 4.4 exakter beschrieben wird.

Die Hierarchie des Klassifikators ergibt sich aus dem Ordnen der signifikanten Konturabschnitte bzgl. ihrer in Lernphase 2 ermittelten Verlässlichkeiten für eine korrekte Klassifikation. Eine zu klassifizierende Kontur der Teststichprobe T wird solange nacheinander mit den geordneten signifikanten Konturabschnitten verglichen, bis der dem jeweiligen signifikanten Konturabschnitt zugeordnete Schwellwert unterschritten wird. Die Kontur wird der Klasse dieses signifikanten Konturabschnitts zugeordnet. Statt die Kontur mit allen signifikanten Konturabschnitten zu vergleichen, wird die Klassifikation in den allermeisten Fällen vorher abgebrochen, was den Klassifikator schnell macht. Wird kein einziger Schwellwert unterschritten, so wird die Kontur keiner Klasse zugeordnet, d.h. die Kontur gehört zu einer unbekanntem Klasse.

Sind alle Konturen der Teststichprobe klassifiziert, erhält man ein Klassifikationsergebnis, das eine Abschätzung für die Klassifikation anderer Konturen zulässt.

Zum Abschluss des Überblicks ist anzumerken, dass dem relativ einfachen Klassifikator eine komplexe Lernphase zur Erzeugung charakteristischer Merkmale gegenübersteht.

4.2 Basisnotationen

Bevor die detaillierte Beschreibung der einzelnen Schritte der Lernphase beginnt, werden an dieser Stelle grundlegende Notationen vor allem für die Beschreibung einer diskreten Kontur definiert.

Die Aufgabendomäne D setzt sich aus einer Lernstichprobe L , einer Teststichprobe T und aus einer Menge R restlicher Objekte, die nicht der Lernstichprobe L oder der Teststichprobe T zugeordnet wurden, zusammen. L , T und R besitzen paarweise kein einziges Objekt gemeinsam:

$$D = \{L, T, R \mid L \cap T = \emptyset, L \cap R = \emptyset, T \cap R = \emptyset\}. \quad (4.2)$$

Die Lernstichprobe L besteht aus N Klassen G_i :

$$L = \{G_1, G_2, \dots, G_i, \dots, G_N\}. \quad (4.3)$$

Jede Klasse G_i besteht aus m_i geschlossenen diskreten 2D-Objektkonturen $K_{i,j}$:

$$G_i = \{K_{i,1}, K_{i,2}, \dots, K_{i,j}, \dots, K_{i,m_i}\}. \quad (4.4)$$

Jede Kontur $K_{i,j}$ besteht aus $n_{i,j}$ aufeinanderfolgenden Punkten $P_{i,j,k}$

$$K_{i,j} = \{P_{i,j,1}, P_{i,j,2}, \dots, P_{i,j,k}, \dots, P_{i,j,n_{i,j}}\} \quad (4.5)$$

mit den gleichen Euklidischen Abständen

$$d = d(P_{i,j,k}, P_{i,j,k+1}) = \text{const}. \quad (4.6)$$

Jede Kontur $K_{i,j}$ besteht aus $n_{i,j}$ sich überlappenden Abschnitten $A_{i,j,k}^{l_{i,j,k}}$:

$$K_{i,j} = \{A_{i,j,1}^{l_{i,j,1}}, A_{i,j,2}^{l_{i,j,2}}, \dots, A_{i,j,k}^{l_{i,j,k}}, \dots, A_{i,j,n_{i,j}}^{l_{i,j,n_{i,j}}}\}. \quad (4.7)$$

Jeder Konturabschnitt $A_{i,j,k}^{l_{i,j,k}}$ besteht aus $(2l_{i,j,k} + 1)$ Punkten $P_{i,j,k}$:

$$A_{i,j,k}^{l_{i,j,k}} = \{P_{i,j,k-l_{i,j,k}}, P_{i,j,k-l_{i,j,k}+1}, \dots, P_{i,j,k}, \dots, P_{i,j,k+l_{i,j,k}}\} \quad (4.8)$$

mit

$$l_{i,j,k} \in \mathbb{N}.$$

Jeder Punkt $P_{i,j,k}$ besteht schließlich aus einer x - und y -Koordinate:

$$P_{i,j,k} = \{x_{i,j,k}, y_{i,j,k}\}. \quad (4.9)$$

Zusätzlich wird jedem Konturabschnitt $A_{i,j,k}^{l_{i,j,k}}$ entweder ein *absoluter Orientierungsvektor*

$$o_{i,j,k} = P_{i,j,k+l_{i,j,k}} - P_{i,j,k-l_{i,j,k}} \quad (4.10)$$

oder ein *mittlerer Orientierungsvektor*

$$\bar{o}_{i,j,k} = \frac{\sum_{z=0}^{l_{i,j,k}} (P_{i,j,k+z} - P_{i,j,k-z})}{l_{i,j,k} + 1} \quad (4.11)$$

zugeordnet.

Ein Beispiel für eine solche Kontur ist in Abbildung 4.5 gezeigt.

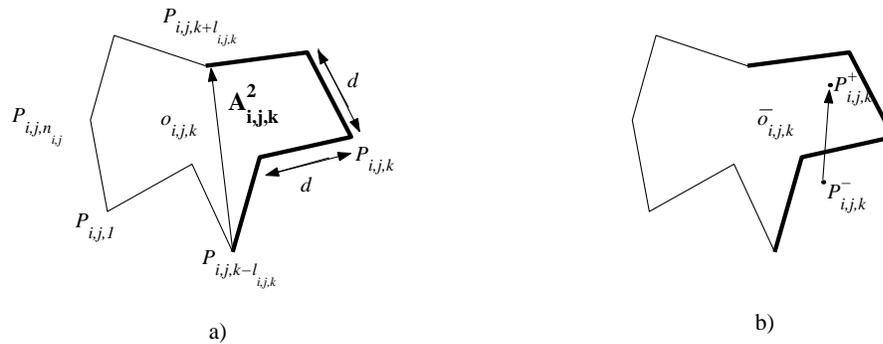


Abbildung 4.5: Diskrete Kontur: a) Beispiel für eine diskrete Kontur $K_{i,j}$ mit einem der $n_{i,j}$ Abschnitte $A_{i,j,k}^{l_{i,j,k}} = A_{i,j,k}^2$ (fett markiert). Die Anzahl der Konturpunkte beträgt 5 also ist $l_{i,j,k} = 2$. $o_{i,j,k}$ ist der absolute Orientierungsvektor von $A_{i,j,k}^2$. Aufeinanderfolgende Konturpunkte haben alle den gleichen Euklidischen Abstand d . b) Dies ist eine Kopie der diskreten Kontur aus Abbildung 4.5a. Hier ist aber der mittlere Orientierungsvektor $\bar{o}_{i,j,k}$ für den Konturabschnitt $A_{i,j,k}^2$ eingezeichnet.

Zwei Konturen werden durch folgende Indizes unterschieden:

$$K_{i,j} \quad \text{und} \quad K_{r,s}.$$

Zwei Punkte oder zwei Abschnitte verschiedener Konturen werden folgendermaßen auseinandergehalten:

$$P_{i,j,k} \quad \text{und} \quad P_{r,s,t}$$

sowie

$$A_{i,j,k}^{l_{i,j,k}} \quad \text{und} \quad A_{r,s,t}^{l_{r,s,t}}.$$

Die Indizes der Konturpunkte müssen modulo berechnet werden, da die Konturen geschlossen sind.

4.3 Synthese signifikanter Konturabschnitte

Dieser Abschnitt erklärt detailliert wie signifikante Konturabschnitte einer Kontur autonom gefunden werden, die diese Kontur von den Konturen der anderen Klassen unterscheiden. Im Mittelpunkt steht hier die **automatische Merkmalsynthese**, d.h. die Synthese aufeinanderfolgender Konturpunkte zu wichtigen Konturabschnitten zur Unterscheidung von 2 Konturen und die Synthese sich überlappender

wichtiger Konturabschnitte einer Kontur bzgl. aller Konturen aus anderen Klassen zu signifikanten Konturabschnitten.

4.3.1 Kosten für das Ersetzen eines Konturpunktes

Die hier vorgestellte Methode basiert auf dem Ersetzen, Einfügen bzw. Entfernen von Konturpunkten, um eine Kontur $K_{i,j}$ in eine andere $K_{r,s}$ zu deformieren. Im Abschnitt 3.3 wurde angedeutet, dass das Einfügen bzw. Entfernen von Geradensegmenten nichts anderes ist, als das Ersetzen eines Geradensegmentes der einen Kontur durch mehrere aufeinanderfolgende Geradensegmente der anderen Kontur. Die Kosten für das Ersetzen ergaben sich dort aus der Summe der Einzelersetzungen des Geradensegmentes der einen Kontur mit jedem einzelnen Geradensegment der anderen Kontur. Diese Herangehensweise gilt hier analog für das Einfügen bzw. Entfernen von Konturpunkten und wird in Abschnitt 4.3.2 näher erläutert. Aus diesem Grund werden in diesem Abschnitt zunächst verschiedene Kostenfunktionen für das Ersetzen einzelner Konturpunkte definiert.

Einzelne Konturpunkte isoliert, ohne ihre Nachbarschaft zu betrachten, macht keinen Sinn. Aus diesem Grund ist allen im Folgenden vorgestellten Kostenfunktionen gemeinsam, dass lokale Abschnitte $A_{i,j,k}^{l_{i,j,k}}$ und $A_{r,s,t}^{l_{r,s,t}}$ miteinander verglichen werden. Die Kosten für den Vergleich werden dem Konturpunktpaar

$$\{P_{i,j,k}, P_{r,s,t}\},$$

d.h. den Konturpunkten in der Mitte der Abschnitte $A_{i,j,k}^{l_{i,j,k}}$ und $A_{r,s,t}^{l_{r,s,t}}$, zugeordnet. Diese Vergleichskosten werden als Kosten für das Ersetzen von $P_{i,j,k}$ durch $P_{r,s,t}$ bzw. umgekehrt interpretiert. Weiterhin sollen nur Konturabschnitte miteinander verglichen werden, die die gleiche Konturpunktanzahl haben:

$$l_{i,j,k} = l_{r,s,t}. \quad (4.12)$$

Zusätzlich wird gefordert, dass aufeinanderfolgende Konturpunkte aller Konturen der Aufgabendomäne den gleichen Euklidischen Abstand d haben sollen. Dabei wird vorausgesetzt, dass d so klein ist, dass die diskreten Konturen gute Näherungen der zugrunde liegenden stetigen Konturen sind. Damit werden im Zusammenhang mit Gleichung (4.12) ausschließlich Konturabschnitte gleicher Bogenlänge miteinander verglichen. Außerdem wird jedem Konturpunkt des Abschnitts $A_{i,j,k}^{l_{i,j,k}}$ genau ein Konturpunkt des anderen Konturabschnitts $A_{r,s,t}^{l_{r,s,t}}$ zugeordnet (1-1-Abbildung). Dies geschieht vernünftigerweise gemäß ihrer Reihenfolgen in der Kontur, d.h. es ergeben sich folgende Korrespondenzpaare:

$$\begin{aligned} & \{P_{i,j,k-l_{i,j,k}}, P_{r,s,t-l_{r,s,t}}\} \\ & \{P_{i,j,k-l_{i,j,k+1}}, P_{r,s,t-l_{r,s,t+1}}\} \end{aligned}$$

$$\begin{array}{c}
\vdots \\
\{P_{i,j,k} \quad , \quad P_{r,s,t}\} \\
\vdots \\
\{P_{i,j,k+l_{i,j,k}} \quad , \quad P_{r,s,t+l_{r,s,t}}\}.
\end{array}$$

Außerdem sollen alle Konturpunkte der verglichenen Konturabschnitte gleichberechtigt zur lokalen Ähnlichkeit beitragen. Punkte an Stellen hoher Krümmung werden also nicht anders gewichtet, als Punkte an Stellen geringer Krümmung, wie z.B. in [BCGJ98] (vergleiche Abschnitt 3.4).

In Abbildung 4.6 sind 2 Konturabschnitte mit gleicher Anzahl an Konturpunkten gezeigt, die für die Kostenberechnung für das Ersetzen von $P_{i,j,k}$ durch $P_{r,s,t}$ herangezogen werden. Dazu muss die Distanz (Unähnlichkeit) der beiden Konturabschnitte festgestellt werden.

Eine Forderung an das Distanzmaß ist, dass es zwischen konkaven und konvexen Konturabschnitten unterscheidet, d.h. dass die Abschnitte in den Kontext mit den Konturen aus denen sie stammen gesetzt werden. Eine Möglichkeit diese Forderung zu erfüllen ist es, den Konturabschnitten Orientierungen zuzuordnen.

Absolute Orientierung

Unter *absoluter Orientierung* eines Konturabschnitts $A_{i,j,k}^{l_{i,j,k}}$ sei die Richtung des Vektors

$$o_{i,j,k} = P_{i,j,k+l_{i,j,k}} - P_{i,j,k-l_{i,j,k}} \quad (4.13)$$

zu verstehen, der den Anfangs- und Endpunkt des Abschnitts verbindet. Die Distanz der Abschnitte $A_{i,j,k}^{l_{i,j,k}}$, $A_{r,s,t}^{l_{r,s,t}}$ berechnet sich als minimale Summe der quadratischen Abstände zwischen korrespondierenden Punkten unter der Berücksichtigung, dass die Orientierungsvektoren $o_{i,j,k}$ und $o_{r,s,t}$ zunächst übereinandergelegt werden und in die gleiche Richtung zeigen.

Die Berechnung der Distanz lässt sich in 3 Schritte zerlegen:

1. Translation von $A_{i,j,k}^{l_{i,j,k}}$ und $A_{r,s,t}^{l_{r,s,t}}$, so dass die zugehörigen Anfangspunkte $P_{i,j,k-l_{i,j,k}}$ und $P_{r,s,t-l_{r,s,t}}$ in den Ursprung des Koordinatensystems verschoben werden. Es ergeben sich die translatierten Konturabschnitte $\check{A}_{i,j,k}^{l_{i,j,k}}$ und $\check{A}_{r,s,t}^{l_{r,s,t}}$ (Abbildung 4.7).
2. Rotation von $\check{A}_{i,j,k}^{l_{i,j,k}}$ und $\check{A}_{r,s,t}^{l_{r,s,t}}$ um die Winkel $\alpha_{i,j,k}$ bzw. $\alpha_{r,s,t}$ um den Koordinatenursprung, so dass die Richtungen der zugehörigen absoluten Orientierungsvektoren $o_{i,j,k}$ und $o_{r,s,t}$ mit der positiven x-Achse übereinstimmen. Es ergeben sich die translatierten und rotierten Konturabschnitte $\hat{A}_{i,j,k}^{l_{i,j,k}}$ und $\hat{A}_{r,s,t}^{l_{r,s,t}}$ (Abbildung 4.8).

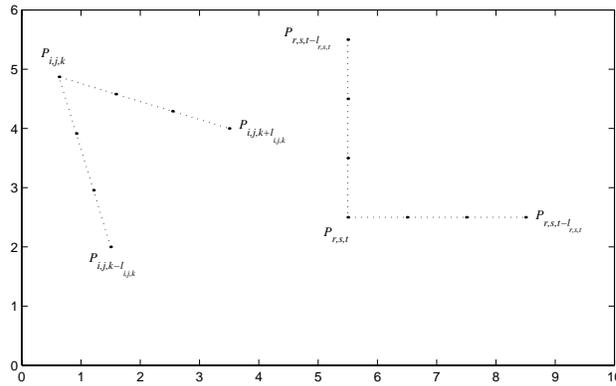


Abbildung 4.6: Zwei Konturabschnitte $A_{i,j,k}^{l_{i,j,k}}$ und $A_{r,s,t}^{l_{r,s,t}}$ der gleichen Bogenlänge.

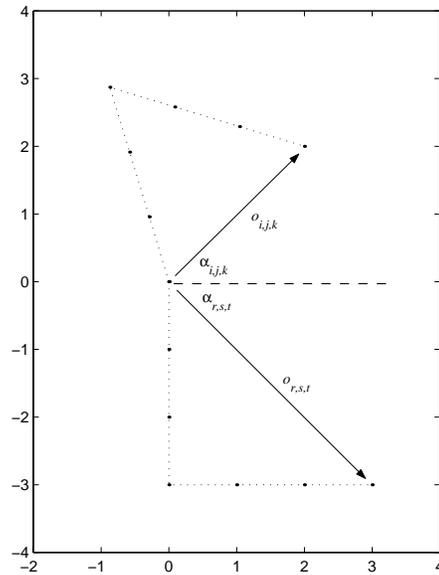


Abbildung 4.7: Translation der Konturabschnitte $A_{i,j,k}^{l_{i,j,k}}$ und $A_{r,s,t}^{l_{r,s,t}}$ aus Abbildung 4.6, so dass die zugehörigen Anfangspunkte $P_{i,j,k+l_{i,j,k}}$ und $P_{r,s,t+l_{r,s,t}}$ in den Ursprung des Koordinatensystems verschoben werden. Es ergeben sich die translatierten Konturabschnitte $\check{A}_{i,j,k}^{l_{i,j,k}}$ und $\check{A}_{r,s,t}^{l_{r,s,t}}$.

3. Verschiebung des Konturabschnittes $\hat{A}_{r,s,t}^{l_{r,s,t}}$ entlang der x-Achse, so dass die Summe der quadrierten Euklidischen Distanzen zwischen korrespondierenden Konturpunkten von $\hat{A}_{i,j,k}^{l_{i,j,k}}$ und $\hat{A}_{r,s,t}^{l_{r,s,t}}$ minimal wird (Abbildung 4.9).

Zur Herleitung wird die Verschiebungsvariable ξ eingeführt:

$$\begin{aligned}
& \sum_{a=-l_{i,j,k}}^{+l_{i,j,k}} (\hat{x}_{i,j,k+a} - (\hat{x}_{r,s,t+a} + \xi))^2 + (\hat{y}_{i,j,k+a} - \hat{y}_{r,s,t+a})^2 \rightarrow \min \\
\Rightarrow \frac{\delta}{\delta \xi} \sum_{a=-l_{i,j,k}}^{+l_{i,j,k}} (\hat{x}_{i,j,k+a} - (\hat{x}_{r,s,t+a} + \xi))^2 + (\hat{y}_{i,j,k+a} - \hat{y}_{r,s,t+a})^2 &= 0 \\
\Rightarrow -2 \sum_{a=-l_{i,j,k}}^{+l_{i,j,k}} (\hat{x}_{i,j,k+a} - (\hat{x}_{r,s,t+a} + \xi)) &= 0 \\
\Rightarrow \sum_{a=-l_{i,j,k}}^{+l_{i,j,k}} (\hat{x}_{i,j,k+a} - \hat{x}_{r,s,t+a}) - \sum_{a=-l_{i,j,k}}^{+l_{i,j,k}} \xi &= 0 \quad (4.14) \\
\Rightarrow \sum_{a=-l_{i,j,k}}^{+l_{i,j,k}} (\hat{x}_{i,j,k+a} - \hat{x}_{r,s,t+a}) &= (2l_{i,j,k} + 1)\xi \\
\Rightarrow \frac{\sum_{a=-l_{i,j,k}}^{+l_{i,j,k}} (\hat{x}_{i,j,k+a} - \hat{x}_{r,s,t+a})}{2l_{i,j,k} + 1} &= \xi \\
\Rightarrow \xi &= \bar{x}_{i,j,k} - \bar{x}_{r,s,t}
\end{aligned}$$

Die Distanz $\Delta_{(i,j,k),(r,s,t)}$ zwischen $A_{i,j,k}^{l_{i,j,k}}$ und $A_{r,s,t}^{l_{r,s,t}}$ ergibt sich damit zu

$$\begin{aligned}
\Delta_{(i,j,k),(r,s,t)} &= \Delta(A_{i,j,k}^{l_{i,j,k}}, A_{r,s,t}^{l_{r,s,t}}) \\
&= \sum_{a=-l_{i,j,k}}^{+l_{i,j,k}} \left((x_{i,j,k} - \bar{x}_{i,j,k}) - (x_{r,s,t} - \bar{x}_{r,s,t}) \right)^2 + (y_{i,j,k} - y_{r,s,t})^2 \quad (4.15)
\end{aligned}$$

Bei $\Delta_{(i,j,k),(r,s,t)}$ handelt es sich um ein Minimum bzgl. ξ , da es kein definiertes Maximum gibt. Die Distanz der beiden Konturabschnitte strebt beim Verschieben des einen Konturabschnitts gegen den anderen für $\xi \rightarrow \infty$ ebenfalls gegen unendlich.

Der Algorithmus Ia zur Berechnung der Distanz zwischen zwei Konturabschnitten basierend auf der absoluten Orientierung ist in Abbildung E.1 im Anhang E zusammengefasst.

Mittlere Orientierung

Die absolute Orientierung eines Konturabschnitts hängt ausschließlich von seinem Anfangs- und Endpunkt ab. Ist nur einer der beiden gestört, so ändert sich

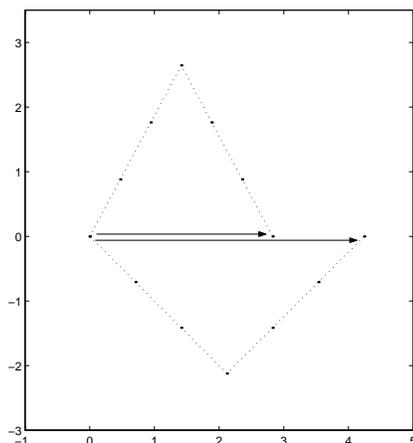


Abbildung 4.8: Rotation der Konturabschnitte $\tilde{A}_{i,j,k}^{l_{i,j,k}}$ und $\tilde{A}_{r,s,t}^{l_{r,s,t}}$ aus Abbildung 4.7, so dass die zugehörigen Orientierungsvektoren $o_{i,j,k}$ und $o_{r,s,t}$ mit der positiven x -Achse übereinstimmen. Es ergeben sich die translatierten und rotierten Konturabschnitte $\hat{A}_{i,j,k}^{l_{i,j,k}}$ und $\hat{A}_{r,s,t}^{l_{r,s,t}}$.

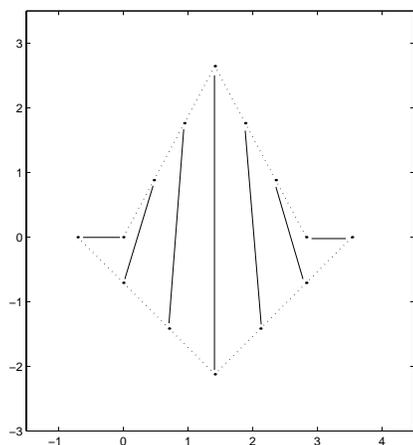


Abbildung 4.9: Verschiebe die Konturabschnitte $\hat{A}_{i,j,k}^{l_{i,j,k}}$ und $\hat{A}_{r,s,t}^{l_{r,s,t}}$ aus Abbildung 4.8 in x -Richtung so gegeneinander, dass die Summe der quadrierten Euklidischen Distanzen korrespondierender Konturpunkte (verbunden durch Linien) minimal wird.

die Orientierung und damit die Distanz $\Delta_{(i,j,k),(r,s,t)}$ zu einem anderen Konturabschnitt.

Eine Alternative besteht darin, eine sogenannte *mittlere Orientierung* zu berechnen, die alle Punkte eines Konturabschnitts in die Berechnung mit einbezieht, so dass diese Orientierung auf eine Störung eines Konturpunktes nicht so stark reagiert. Unter mittlerer Orientierung eines Konturabschnittes $A_{i,j,k}^{l_{i,j,k}}$ wird die Richtung des Vektors

$$\bar{o}_{i,j,k} = \frac{\sum_{z=0}^{l_{i,j,k}} (P_{i,j,k+z} - P_{i,j,k-z})}{l_{i,j,k+1}} \quad (4.16)$$

verstanden, der den Schwerpunkt $P_{i,j,k}^-$ der ersten $l_{i,j,k}$ Konturpunkte mit dem Schwerpunkt $P_{i,j,k}^+$ der letzten $l_{i,j,k}$ Konturpunkte verbindet (siehe Abbildung 4.5b).

Die Berechnung der Distanz $\Delta_{(i,j,k),(r,s,t)}$ der Abschnitte $A_{i,j,k}^{l_{i,j,k}}$ und $A_{r,s,t}^{l_{r,s,t}}$ gestaltet sich analog zur Berechnung der Distanz basierend auf absoluter Orientierung. Der einzige Unterschied besteht darin, dass im ersten Schritt die Schwerpunkte $P_{i,j,k}^-$ und $P_{r,s,t}^-$ in den Ursprung des Koordinatensystems verschoben werden.

Die einzelnen Schritte zur Distanzberechnung, d.h. die Translation in und die Rotation der Konturabschnitte aus Abbildung 4.6 um den Schwerpunkt des Koordinatensystems sowie das Verschieben eines Konturabschnittes gegenüber dem anderen ist in den Abbildungen 4.10-4.12 gezeigt. Der Algorithmus zur Berechnung der Distanz zwischen 2 Konturabschnitten basierend auf der mittleren Orientierung ist in Abbildung E.2 in Anhang E zusammengefasst.

Die Distanz zwischen 2 Konturabschnitten $\Delta_{(i,j,k),(r,s,t)} = \Delta(A_{i,j,k}^{l_{i,j,k}}, A_{r,s,t}^{l_{r,s,t}})$ ist sowohl basierend auf absoluter Orientierung als auch mittlerer Orientierung eine Metrik. Das bedeutet im einzelnen:

$$\begin{aligned} \Delta_{(i,j,k),(r,s,t)} &\geq 0 \\ \Delta_{(i,j,k),(i,j,k)} &= 0 \\ \Delta_{(r,s,t),(r,s,t)} &= 0 \\ \Delta_{(i,j,k),(r,s,t)} &> 0 \quad \forall (i,j,k) \neq (r,s,t) \\ \Delta_{(i,j,k),(r,s,t)} &= \Delta_{(r,s,t),(i,j,k)} \\ \Delta_{(i,j,k),(r,s,t)} + \Delta_{(r,s,t),(x,y,z)} &\geq \Delta_{(i,j,k),(x,y,z)}. \end{aligned} \quad (4.17)$$

4.3.2 Korrespondenzliste und globale Ähnlichkeit

Ein erstes Ziel auf dem Weg zu signifikanten Konturabschnitten ist es, eine Korrespondenzliste (Abbildungsfunktion) der Konturpunkte von 2 Konturen zu be-

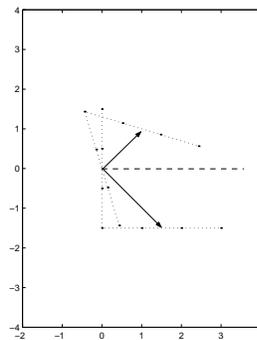


Abbildung 4.10: Translation der Konturabschnitte aus Abbildung 4.6 in den Ursprung basierend auf mittlerer Orientierung.

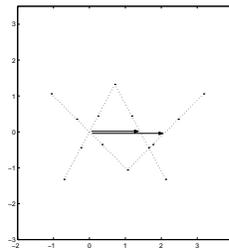


Abbildung 4.11: Rotation der Konturabschnitte aus Abbildung 4.10 um den Ursprung basierend auf mittlerer Orientierung.

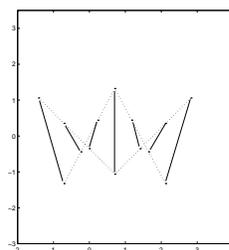


Abbildung 4.12: Verschiebe die Konturabschnitte aus Abbildung 4.11 längs der x -Achse so gegeneinander, dass die Summe der quadrierten Euklidischen Distanzen korrespondierender Konturpunkte (verbunden durch Linien) minimal wird.

stimmen, so dass die Kosten für die Deformation der einen in die andere Kontur minimal werden.

Um Rotationsinvarianz der Methode sicherzustellen, ist es notwendig die Ersetzungskosten aller möglichen Konturpunktpaare zweier Konturen $K_{i,j}$, $K_{r,s}$ zu berechnen. Diese werden geordnet in eine *lokale Distanzmatrix* eingetragen.

Darauf folgend wird in einer solchen lokalen Distanzmatrix nach einer Korrespondenzliste zwischen den beiden Konturen gesucht, die die Summe der Ersetzungskosten minimiert. Diese Korrespondenzliste kann als Pfad durch die lokale Distanzmatrix interpretiert werden, die die einzelnen Elemente mit den zugehörigen Ersetzungskosten durchläuft. Die Summe der durchlaufenen Ersetzungskosten wird als Maß für die globale (Un)ähnlichkeit der beiden Konturen aufgefasst.

Lokale (Distanz-)Ähnlichkeitsmatrizen

Zur Erstellung einer lokalen (Distanz-)Ähnlichkeitsmatrix werden die Konturen beginnend bei ihren Anfangspunkten nacheinander bis zu ihren Endpunkten durchlaufen. In der ersten Zeile der lokalen (Distanz-)Ähnlichkeitsmatrix werden die Ersetzungskosten für den Punkt $P_{i,j,1}$ durch die Punkte $P_{r,s,1}$ bis $P_{r,s,n_{r,s}}$ eingetragen. In der zweiten Zeile der lokalen (Distanz-)Ähnlichkeitsmatrix werden die Ersetzungskosten für den Punkt $P_{i,j,2}$ durch die Punkte $P_{r,s,1}$ bis $P_{r,s,n_{r,s}}$ eingetragen usw..

Lokale Distanzmatrix für konstante $l_{i,j,k}$, $l_{r,s,t}$. In diesem Fall werden **allen** Konturpunkten Konturabschnitte **konstanter** Länge zugeordnet:

$$l_{i,j,k} = l_{r,s,t} = l_{const} \quad (4.18)$$

In die lokale Distanzmatrix werden direkt die Distanzen $\Delta_{(i,j,k),(r,s,t)}$ (Gleichung (4.15)) zwischen den Konturabschnitten $A_{i,j,k}^{l_{const}}$ und $A_{r,s,t}^{l_{const}}$ eingetragen. Es entsteht die lokale Distanzmatrix

$$\Lambda^{(i,j),(r,s)} = \begin{pmatrix} \Delta_{(i,j,1),(r,s,1)} & \cdots & \Delta_{(i,j,1),(r,s,t)} & \cdots & \Delta_{(i,j,1),(r,s,n_{r,s})} \\ \Delta_{(i,j,2),(r,s,1)} & \cdots & \Delta_{(i,j,2),(r,s,t)} & \cdots & \Delta_{(i,j,2),(r,s,n_{r,s})} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta_{(i,j,k),(r,s,1)} & \cdots & \Delta_{(i,j,k),(r,s,t)} & \cdots & \Delta_{(i,j,k),(r,s,n_{r,s})} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta_{(i,j,n_{i,j}),(r,s,1)} & \cdots & \Delta_{(i,j,n_{i,j}),(r,s,t)} & \cdots & \Delta_{(i,j,n_{i,j}),(r,s,n_{r,s})} \end{pmatrix} \quad (4.19)$$

für die Konturen $K_{i,j}$ und $K_{r,s}$. Der Algorithmus IIa zur Berechnung von $\Lambda^{(i,j),(r,s)}$ ist zusammengefasst in Abbildung E.3 in Anhang E.

Lokale Ähnlichkeitsmatrix für variable $l_{i,j,k}$, $l_{r,s,t}$. In diesem Fall dürfen die den Konturpunktpaaren zugeordneten Konturabschnitte verschieden lang sein.

Für jedes Konturpunktpaar wird nach maximalem

$$l_{(i,j,k),(r,s,t)}^{max} = l_{i,j,k}^{max} = l_{r,s,t}^{max} \quad (4.20)$$

für die Abschnitte $A_{i,j,k}^{l_{i,j,k}}$ und $A_{r,s,t}^{l_{r,s,t}}$ um die Punkte $P_{i,j,k}$ und $P_{r,s,t}$ herum gesucht, so dass für die Distanz der beiden Abschnitte gerade noch gilt:

$$\Delta_{(i,j,k),(r,s,t)} \leq E. \quad (4.21)$$

E ist die vorgegebene maximal erlaubte konstante Distanz zwischen je zwei Abschnitten $A_{i,j,k}^{l_{i,j,k}}$ und $A_{r,s,t}^{l_{r,s,t}}$. $l_{i,j,k}$ und $l_{r,s,t}$ werden, beginnend bei $l_{i,j,k} = l_{r,s,t} = 1$, solange jeweils um 1 erhöht, bis E überschritten wird. Das letzte $l_{i,j,k}$ bzw. $l_{r,s,t}$ bei dem E noch nicht überschritten wurde, ist dann das gesuchte $l_{(i,j,k),(r,s,t)}^{max}$. $(2l_{(i,j,k),(r,s,t)}^{max} + 1)$ gibt dann die maximale Länge der Abschnitte $A_{i,j,k}^{l_{i,j,k}}$ und $A_{r,s,t}^{l_{r,s,t}}$ um die Punkte $P_{i,j,k}$ und $P_{r,s,t}$ herum an, die bzgl. einer zugelassenen maximalen Distanz E als gleich angesehen werden können. D.h. den einzelnen Konturpunkten werden Konturabschnitte variabler Länge (siehe Gleichung 4.20) zugeordnet.

Je größer $l_{(i,j,k),(r,s,t)}^{max}$, desto kleiner die Kosten für das Ersetzen des Punktes $P_{i,j,k}$ durch $P_{r,s,t}$. Das lässt sich nachvollziehen, wenn man annimmt, dass sich die einzelnen Distanzbeiträge gleichmäßig über die korrespondierenden Konturpunkte der Abschnitte verteilen. Das hieße, würden alle Abschnitte konstante Längen haben, dass die Summe der einzelnen Distanzbeiträge umso geringer ist, je länger die Konturabschnitte sind.

Der Wert $l_{(i,j,k),(r,s,t)}^{max}$ wird in die Matrix $\tilde{\Lambda}^{(i,j),(r,s)}$ in Zeile k und in Spalte t eingetragen:

$$\tilde{\Lambda}^{(i,j),(r,s)} = \begin{pmatrix} l_{(i,j,1),(r,s,1)}^{max} & \cdots & l_{(i,j,1),(r,s,t)}^{max} & \cdots & l_{(i,j,1),(r,s,n_{r,s})}^{max} \\ l_{(i,j,2),(r,s,1)}^{max} & \cdots & l_{(i,j,2),(r,s,t)}^{max} & \cdots & l_{(i,j,2),(r,s,n_{r,s})}^{max} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{(i,j,k),(r,s,1)}^{max} & \cdots & l_{(i,j,k),(r,s,t)}^{max} & \cdots & l_{(i,j,k),(r,s,n_{r,s})}^{max} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{(i,j,n_{i,j}),(r,s,1)}^{max} & \cdots & l_{(i,j,n_{i,j}),(r,s,t)}^{max} & \cdots & l_{(i,j,n_{i,j}),(r,s,n_{r,s})}^{max} \end{pmatrix} \quad (4.22)$$

Da $l_{(i,j,k),(r,s,t)}^{max}$ im Gegensatz zu $\Delta_{(i,j,k),(r,s,t)}$ bei kleinerer Distanz größere Werte angibt, ist es unmissverständlicher hier von *lokaler Ähnlichkeitsmatrix* anstelle von *lokaler Distanzmatrix* zu sprechen. Der Algorithmus IIB zur Berechnung der lokalen Ähnlichkeitsmatrix $\tilde{\Lambda}^{(i,j),(r,s)}$ ist in Abbildung E.4 in Anhang E dargestellt.

Ermittlung einer Korrespondenzliste mit minimalen Kosten

Die Deformation einer Kontur $K_{i,j}$ in eine andere Kontur $K_{r,s}$ besteht aus vielen kleinen Elementardeformationen. Die Kosten für das Ersetzen von Konturpunkten werden als diese Elementardeformationen interpretiert. In den lokalen Distanz- bzw. Ähnlichkeitsmatrizen $\Lambda^{(i,j),(r,s)}$ bzw. $\tilde{\Lambda}^{(i,j),(r,s)}$ sind die Kosten für das Ersetzen aller möglichen Konturpunktombinationen enthalten, so dass als nächstes eine Liste all der Konturpunktpaare gefunden werden muss, die die Gesamtkosten, d.h. die Summe der Ersetzungskosten möglicher Konturpunktombinationen, minimiert. Mit anderen Worten heißt das, dass eine diskrete Abbildungsfunktion der Kontur $K_{i,j}$ auf die Kontur $K_{r,s}$ gefunden werden muss, die den Aufwand der Gesamtdeformation minimiert (siehe dazu auch Abbildung 3.12).

Die Konturen $K_{i,j}$ und $K_{r,s}$ haben im allgemeinen eine unterschiedliche Anzahl $n_{i,j} \neq n_{r,s}$ an Konturpunkten. Dies hat zur Folge, dass Konturpunkte nicht nur ersetzt, sondern Konturpunkte auch eingefügt oder entfernt werden müssen, um $K_{i,j}$ in $K_{r,s}$ zu deformieren. Dazu wird von der Vorstellung ausgegangen, dass sich ein Konturpunkt der einen Kontur in 2 oder mehrere benachbarte Konturpunkte der anderen Kontur aufteilt. D.h. der eine Konturpunkt wird durch 2 oder mehrere benachbarte Konturpunkte der anderen Kontur ersetzt. Damit addieren sich die Kosten der Einzelersetzungen.

Die Matrizen $\Lambda^{(i,j),(r,s)}$ bzw. $\tilde{\Lambda}^{(i,j),(r,s)}$ wurden so konstruiert, dass benachbarte Elemente benachbarte Punkte in $K_{i,j}$ und $K_{r,s}$ aufeinander abbilden. So bilden beispielsweise die benachbarten Elemente $\Lambda_{k,t}^{(i,j),(r,s)}$ und $\Lambda_{k,t+1}^{(i,j),(r,s)}$ den Konturpunkt $P_{i,j,k}$ auf die benachbarten Konturpunkte $P_{r,s,t}$ und $P_{r,s,t+1}$ ab. Die Kosten für eine solche Korrespondenz belaufen sich dann auf $\Delta_{(i,j,k),(r,s,t)} + \Delta_{(i,j,k),(r,s,t+1)}$.

Ein Beispiel für 5 benachbarte Elemente zeigt Abbildung 4.13. Dort werden beispielhaft die Konturpunkte $P_{i,j,k}$, $P_{i,j,k+1}$, $P_{i,j,k+2}$ auf die Punkte $P_{r,s,t}$, $P_{r,s,t+1}$, $P_{r,s,t+2}$, $P_{r,s,t+3}$, $P_{r,s,t+4}$ abgebildet. D.h. es werden 2 Konturpunkte aus $K_{r,s}$ entfernt bzw. in $K_{i,j}$ eingefügt. Im Einzelnen werden die Punkte $P_{i,j,k}$ bzw. $P_{i,j,k+2}$ durch jeweils 2 Punkte $\{P_{r,s,t}, P_{r,s,t+1}\}$ bzw. $\{P_{r,s,t+3}, P_{r,s,t+4}\}$ ersetzt. Da in diesen Fällen je ein Konturpunkt aus $K_{i,j}$ mit je 2 Konturpunkten aus $K_{r,s}$ korrespondieren, sind die zugehörigen Elemente der Matrix horizontal benachbart. Im umgekehrten Fall wären die Elemente vertikal benachbart. Der Punkt $P_{i,j,k+1}$ wird nur durch einen einzigen Punkt $P_{r,s,t+2}$ ersetzt. Damit korrespondiert $P_{i,j,k}$ nur mit einem einzigen Punkt und das zugehörige Element der Matrix ist mit dem vorhergehenden diagonal benachbart. Da $P_{i,j,k+1}$ nicht mit $P_{r,s,t+3}$ bzw. $P_{i,j,k+2}$ nicht mit $P_{r,s,t+2}$ korrespondiert, ist in diesem Fall auch das nachfolgende Element mit dem Element in der (k+1)ten Zeile und (t+2)ten Spalte diagonal benachbart. Die Kosten für das Ersetzen der dargestellten Konturabschnitte $\{P_{i,j,k}, P_{i,j,k+1}, P_{i,j,k+2}\}$ und $\{P_{r,s,t}, P_{r,s,t+1}, P_{r,s,t+2}, P_{r,s,t+3}, P_{r,s,t+4}\}$ belaufen sich auf $\Delta_{(i,j,k),(r,s,t)} + \Delta_{(i,j,k),(r,s,t+1)} + \Delta_{(i,j,k+1),(r,s,t+2)} + \Delta_{(i,j,k+2),(r,s,t+3)}$

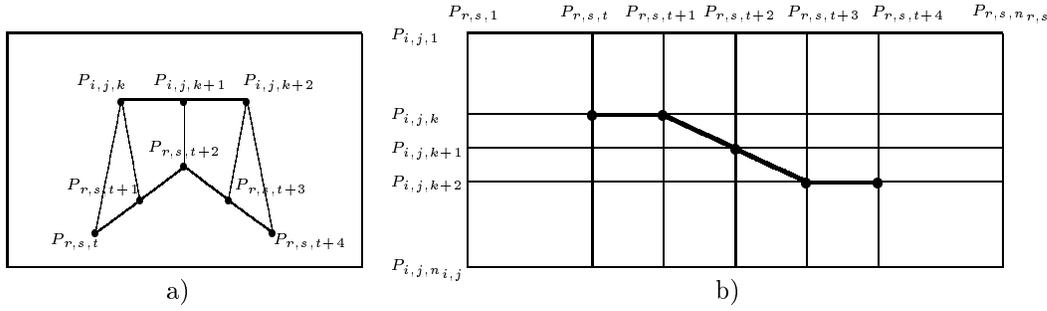


Abbildung 4.13: Beispiel für eine Teilkorrespondenzliste. a) Die Konturabschnitte $\{P_{i,j,k}, P_{i,j,k+1}, P_{i,j,k+2}\}$ aus einer Kontur $K_{i,j}$ und $\{P_{r,s,t}, P_{r,s,t+1}, P_{r,s,t+2}, P_{r,s,t+3}, P_{r,s,t+4}\}$ aus einer Kontur $K_{r,s}$ werden beispielsweise so aufeinander abgebildet, dass $P_{i,j,k}$ mit $\{P_{r,s,t}, P_{r,s,t+1}\}$, $P_{i,j,k+1}$ mit $P_{r,s,t+2}$ und $P_{i,j,k+2}$ mit $\{P_{r,s,t+3}, P_{r,s,t+4}\}$ korrespondieren (angedeutet durch die Linien zwischen den Punkten). b) Darstellung der Teilkorrespondenzlisten für die Abschnitte in a).

+ $\Delta_{(i,j,k+2),(r,s,t+4)}$.

Es wird nun nach einer Liste $Q^{(i,j),(r,s)}$ bzw. $\tilde{Q}^{(i,j),(r,s)}$ benachbarter Elemente in $\Lambda^{(i,j),(r,s)}$ bzw. $\tilde{\Lambda}^{(i,j),(r,s)}$ gesucht, für die die Summe der zugehörigen Ersetzungskosten in $\Lambda_{k,t}^{(i,j),(r,s)}$ minimal bzw. in $\tilde{\Lambda}_{k,t}^{(i,j),(r,s)}$ maximal ist.

Eine solche Liste $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) besteht aus $Z^{(i,j),(r,s)}$ ($\tilde{Z}^{(i,j),(r,s)}$) Elementen q_z (\tilde{q}_z):

$$Q^{(i,j),(r,s)} = \{q_1, q_2, \dots, q_z, \dots, q_{Z^{(i,j),(r,s)}}\} \quad (4.23)$$

bzw.

$$\tilde{Q}^{(i,j),(r,s)} = \{\tilde{q}_1, \tilde{q}_2, \dots, \tilde{q}_z, \dots, \tilde{q}_{\tilde{Z}^{(i,j),(r,s)}}\}. \quad (4.24)$$

Jedes Element wiederum hat die Koordinaten

$$q_z = (k(z), t(z)) \quad (4.25)$$

bzw.

$$\tilde{q}_z = (\tilde{k}(z), \tilde{t}(z)). \quad (4.26)$$

Damit ergibt sich als Summe der Ersetzungskosten für eine solche Liste

$$\sum_{z=1}^{Z^{(i,j),(r,s)}} \Lambda_{k(z),t(z)}^{(i,j),(r,s)}$$

bzw.

$$\sum_{z=1}^{\tilde{Z}^{(i,j),(r,s)}} \tilde{\Lambda}_{\tilde{k}(z),\tilde{t}(z)}^{(i,j)(r,s)}.$$

Diese Summen müssen im Fall der lokalen Distanzmatrix $\Lambda^{(i,j)(r,s)}$ minimiert

$$g^{(i,j),(r,s)} = \sum_{z=1}^{Z^{(i,j),(r,s)}} \Lambda_{k(z),t(z)}^{(i,j)(r,s)} \rightarrow \min! \quad (4.27)$$

und im Fall der lokalen Ähnlichkeitsmatrix $\tilde{\Lambda}^{(i,j)(r,s)}$ maximiert werden:

$$\tilde{g}^{(i,j),(r,s)} = \sum_{z=1}^{\tilde{Z}^{(i,j),(r,s)}} \tilde{\Lambda}_{\tilde{k}(z),\tilde{t}(z)}^{(i,j)(r,s)} \rightarrow \max!. \quad (4.28)$$

Man erhält also sowohl die globale Distanz $g^{(i,j),(r,s)}$ bzw. die globale Ähnlichkeit $\tilde{g}^{(i,j),(r,s)}$ zwischen den Konturen $K_{i,j}$ und $K_{r,s}$ als auch eine Korrespondenzliste $Q^{(i,j),(r,s)}$ bzw. $\tilde{Q}^{(i,j),(r,s)}$, die die Korrespondenz der Konturpunkte beinhaltet. Eine Korrespondenzliste

$$t = f(k) \quad (4.29)$$

kann nicht eindeutig bestimmt werden, da mehrere Konturpunkte auf ein- und denselben Konturpunkt abgebildet werden können.

Für die Korrespondenzliste gelten noch folgende zusätzlichen Randbedingungen:

- $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) *muss geschlossen sein*, da $K_{i,j}$ und $K_{r,s}$ geschlossen sind. Da dieser Untersuchung geschlossene Konturen zugrunde liegen, kann man sich die Matrizen $\Lambda^{(i,j)(r,s)}$ bzw. $\tilde{\Lambda}^{(i,j)(r,s)}$ in alle Richtungen beliebig lang erweitert vorstellen. Das soll heißen, dass sich an die letzte Spalte die erste Spalte und an die letzte Zeile die erste Zeile anschließt usw.. Damit kann man sich die Matrix auch zu einem \rightarrow *Torus* gebogen denken. Das bedeutet, dass das letzte Element $q_{Z^{(i,j),(r,s)}}$ ($\tilde{q}_{\tilde{Z}^{(i,j),(r,s)}}$) der Korrespondenzliste $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) dem ersten Element q_1 (\tilde{q}_1) benachbart ist. Das hat zur Folge:
 1. $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) muss mindestens ein Element aus jeder Spalte und Zeile von $\Lambda^{(i,j)(r,s)}$ ($\tilde{\Lambda}^{(i,j)(r,s)}$) beinhalten.
 2. $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) muss mindestens $\max(n_{i,j}, n_{r,s})$ Elemente beinhalten.

- $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) muss monoton sein,
d.h. dass die Koordinaten

$$k(z) \geq k(z-1) \quad \text{bzw.} \quad \tilde{k}(z) \geq \tilde{k}(z-1) \quad (4.30)$$

oder

$$t(z) \geq t(z-1) \quad \text{bzw.} \quad \tilde{t}(z) \geq \tilde{t}(z-1) \quad (4.31)$$

oder die Gleichungen (4.30) und (4.31) gleichzeitig erfüllt sein müssen. Dies bedeutet, dass die Reihenfolge der Konturpunkte in die Korrespondenzlistensuche miteinbezogen wird. D.h., dass Konturpunkte ausschließlich durch benachbarte Konturpunkte ersetzt werden dürfen. Das hat zur Folge, dass $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) maximal $(n_{i,j} + n_{r,s} - 1)$ Elemente besitzt, da die Bedingung sonst nicht erfüllt werden kann.

Eine Möglichkeit eine Korrespondenzliste $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) zu finden, die die obigen Bedingungen sowie die Gleichungen (4.27) bzw. (4.28) erfüllt, ist die **Dynamische Programmierung** (siehe Anhang C). Mit dieser Optimierungsmethode wird die lokale Distanzmatrix $\Lambda_{k,t}^{(i,j),(r,s)}$ bzw. die lokale Ähnlichkeitsmatrix $\tilde{\Lambda}_{k,t}^{(i,j),(r,s)}$ untersucht.

Ausführlich wird jetzt zunächst der Algorithmus für die Minimumsuche in $\Lambda_{k,t}^{(i,j),(r,s)}$ mit Hilfe der dynamischen Programmierung erklärt: Wird aus Vereinfachungsgründen angenommen, dass

$$M = \Lambda^{(i,j),(r,s)}, \quad (4.32)$$

so lässt sich der in Abbildung 4.14 gezeigte gerichtete, gewichtete Graph konstruieren. Dieser Graph ist sowohl vertikal als auch horizontal geschlossen, lässt sich also wie die Matrix $\Lambda^{(i,j),(r,s)}$ als Torus vorstellen.

Jedem Punktpaar $(P_{i,j,k}, P_{r,s,t})$ wird ein Knoten $\kappa_{(i,j,k),(r,s,t)}$ zugeordnet (Kreis in Abbildung 4.14).

Die Knoten sind durch gewichtete Kanten gerichtet miteinander verbunden. Die Gewichte der Kanten sind die Ersetzungskosten zugehörig zu den Knoten, d.h. bestimmten Punktpaaren, von denen sie ausgehen. Da eine der geforderten Bedingungen für die Korrespondenzliste $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) Monotonie (siehe Gleichungen (4.30), (4.31)) ist, gehen die Kanten ausschließlich in die Richtungen Süd (S), Ost (O) und Südost (SO).

Die gesuchte Korrespondenzliste $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) entspricht in einem solchen Graphen einem Pfad $\Psi^{(i,j),(r,s)}$, beginnend und endend bei einem der Knoten in der ersten Zeile. Der Anfangs- und Endknoten ist der gleiche, da die Konturen geschlossen sind. Alle geforderten Bedingungen für $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) gelten genauso auch für den Pfad $\Psi^{(i,j),(r,s)}$. Wird dem Pfad von diesem Knoten aus entlang den noch zu findenden Kanten gefolgt und die Gewichte (Ersetzungskosten)

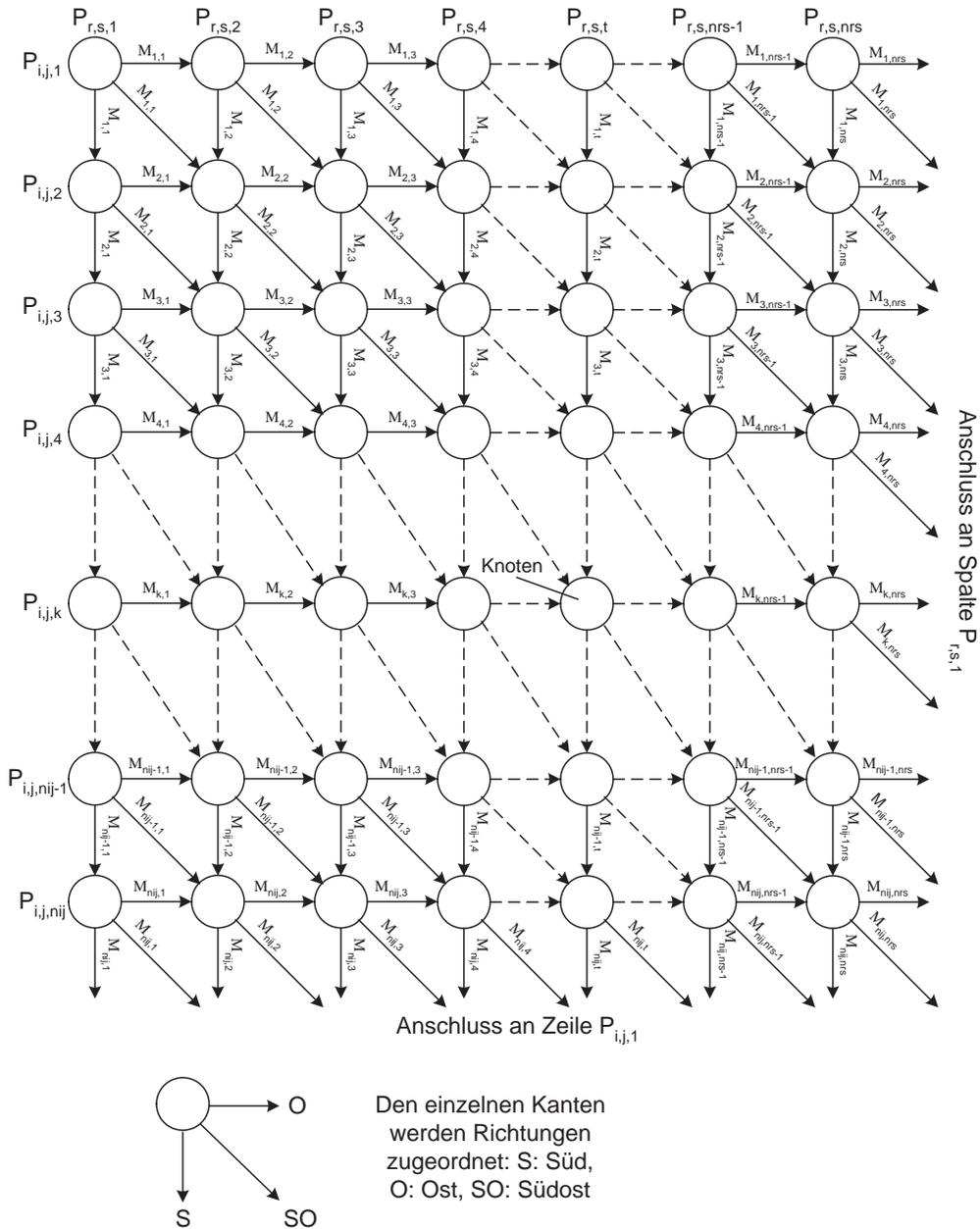


Abbildung 4.14: Gerichteter Graph zur Bestimmung minimaler oder maximaler Kosten mit Hilfe eines Pfades durch diesen Graphen. Die Knoten $K_{(i,j,k),(r,s,t)}$ (Kreise) sind durch gewichtete Kanten gerichtet miteinander verbunden. Die Gewichte der Kanten entsprechen den Ersetzungskosten in $\Lambda^{(i,j),(r,s)}$ bzw. $\tilde{\Lambda}^{(i,j),(r,s)}$. Der Graph ist analog zu dem in Anhang C besprochenen Beispiel aufgebaut. Daraus folgt in diesem Fall, dass jeder Kante, die von einem Knoten ausgeht, die gleichen Kosten zugeordnet werden, um der allgemeinen Vorgehensweise in Anhang C zu entsprechen.

entlang des Pfades aufsummiert, so werden die minimalen Gesamtkosten für die Deformation der Kontur $K_{i,j}$ in die Kontur $K_{r,s}$ gefunden.

Der Pfad muss nicht unbedingt im Knoten $\kappa_{(i,j,1),(r,s,1)}$ beginnen, da es zwar sein kann, aber nicht sein muss, dass Punkt $P_{i,j,1}$ durch $P_{r,s,1}$ ersetzt werden muss, um die Summe der Ersetzungen minimal zu bekommen. Aus diesem Grunde müssen zunächst alle Pfade

$$\Psi_{\psi}^{(i,j),(r,s)}, \quad 1 \leq \psi \leq n_{r,s} \quad (4.33)$$

mit minimalen Kosten für alle $n_{r,s}$ Knoten der ersten Zeile des Graphen in Abbildung 4.14 berechnet werden. Danach wird dann derjenige Pfad $\Psi_{\psi^{min}}^{(i,j),(r,s)}$ genommen, der über alle $n_{r,s}$ Pfade die geringsten Gesamtkosten verursacht.

Der Algorithmus soll nun anhand eines kleinen Beispiels für den Pfad $\Psi_1^{(i,j),(r,s)}$ erläutert werden (siehe dazu Abbildung 4.15).

Um die Bedingungen für den Pfad einzuhalten und den Algorithmus so einfach wie möglich zu halten, wird die erste Spalte des Graphen zusätzlich als neue letzte Spalte angehängt. Genauso wird die erste Zeile des Graphen zusätzlich als neue letzte Zeile angehängt. Außerdem wird der Knoten $\kappa_{(i,j,1),(r,s,1)}$ in die rechts unten entstandene Lücke aufgenommen, so dass wieder ein rechteckiger Graph entsteht. Dieser neu entstandene Graph gilt nur für die Suche nach dem Pfad $\Psi_1^{(i,j),(r,s)}$, der in Knoten $\kappa_{(i,j,1),(r,s,1)}$ beginnt und in Knoten $\kappa_{(i,j,5),(r,s,4)}$ endet. D.h. es wird ein geschlossener Pfad gesucht, der im Konturpunkt paar $(P_{i,j,1}, P_{r,s,1})$ beginnt und endet.

Dem Knoten $\kappa_{(i,j,5),(r,s,4)}$ in Ebene 0 werden die Kosten 0 zugeordnet:

$$sum_{(i,j,5),(r,s,4)} := 0. \quad (4.34)$$

Nun werden die Kosten für die Knoten $\kappa_{(i,j,4),(r,s,4)}$ und $\kappa_{(i,j,5),(r,s,3)}$ in der Ebene 1 bzgl. des Knotens $\kappa_{(i,j,5),(r,s,4)}$ in der Ebene 0 berechnet:

$$\begin{aligned} sum_{(i,j,4),(r,s,4)} &:= sum_{(i,j,5),(r,s,4)} + 4 \\ &:= 4 \\ sum_{(i,j,5),(r,s,3)} &:= sum_{(i,j,5),(r,s,4)} + 8 \\ &:= 8 \end{aligned} \quad (4.35)$$

Als nächstes werden die minimalen Kosten für die Knoten $\kappa_{(i,j,3),(r,s,4)}$, $\kappa_{(i,j,4),(r,s,3)}$ und $\kappa_{(i,j,5),(r,s,2)}$ in der Ebene 2 bzgl. der Knoten in Ebene 1 und damit bzgl. des Knotens $\kappa_{(i,j,5),(r,s,4)}$ in der Ebene 0 berechnet:

$$\begin{aligned} sum_{(i,j,3),(r,s,4)} &:= sum_{(i,j,4),(r,s,4)} + 2 \\ &:= 6 \end{aligned}$$

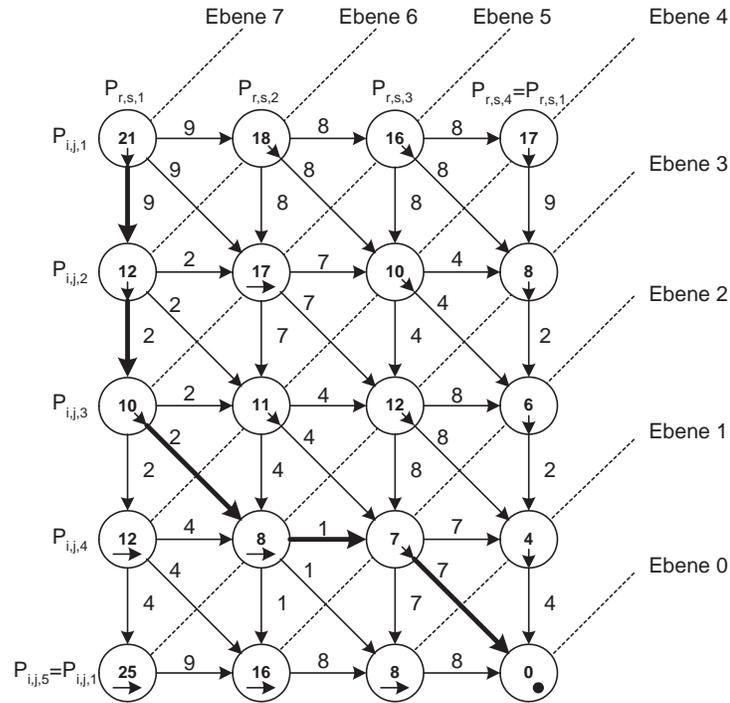


Abbildung 4.15: Einfaches Beispiel der Realisierung eines Graphen für die Algorithmen IIIa (IIIb). Die erste Spalte des Graphen wurde als neue letzte Spalte angehängt. Die erste Zeile des Graphen wurde als neue letzte Zeile angehängt. Der Knoten $\kappa_{(i,j,1),(r,s,1)}$ wurde in die nun rechts unten entstandene Lücke kopiert. Dieser Graph gilt ausschließlich für die Suche eines Pfades vom Knoten $\kappa_{(i,j,1),(r,s,1)}$ zum Knoten $\kappa_{(i,j,5),(r,s,4)}$.

$$\begin{aligned} \text{sum}_{(i,j,4),(r,s,3)} &:= \min(\text{sum}_{(i,j,4),(r,s,4)}, \text{sum}_{(i,j,5),(r,s,3)}, \text{sum}_{(i,j,5),(r,s,4)}) + 7 \\ &:= 7 \end{aligned} \quad (4.36)$$

$$\begin{aligned} \text{sum}_{(i,j,5),(r,s,2)} &:= \text{sum}_{(i,j,5),(r,s,3)} + 8 \\ &:= 16. \end{aligned}$$

So wird bis Ebene 7 weiter fortgefahren. Am Ende sind dann dem Knoten $\kappa_{(i,j,1),(r,s,1)}$ in der Ebene 7 die minimalen Gesamtkosten bzgl. Knoten $\kappa_{(i,j,5),(r,s,4)}$ in Ebene 0 zugeordnet.

Die Minimumsuche ist hier das Entscheidende. Durch sie werden rekursiv die Kosten der vorigen Ebenen in die Kostenberechnung miteinbezogen und so eine Menge Aufwand eingespart, der notwendig gewesen wäre, wenn alle möglichen Pfade berechnet worden wären (siehe dazu Anhang C).

Zusätzlich zu den Kosten wird sich in jedem Knoten die Richtung $\text{dir}_{(i,j,k),(r,s,t)}$

gemerkt, durch die die minimalen Kosten dieses Knotens, bzgl. eines Knotens aus der davor liegenden Ebene, entstanden sind.

Ebene 1:

$$\begin{aligned} dir_{(i,j,4),(r,s,4)} &:= S \\ dir_{(i,j,5),(r,s,3)} &:= O \end{aligned} \quad (4.37)$$

Ebene 2:

$$\begin{aligned} dir_{(i,j,3),(r,s,4)} &:= S \\ dir_{(i,j,4),(r,s,3)} &:= SO \\ dir_{(i,j,5),(r,s,2)} &:= O \end{aligned} \quad (4.38)$$

Dem Knoten in der Ebene 0 wird statt einer Richtung eine Endmarkierung *end* (schwarzer Kreis in Abbildung 4.15) zugeordnet, da dieser Knoten keine Vorgängerknoten besitzt.

Die einzelnen Richtungen $dir_{(i,j,k),(r,s,t)}$ sind in den Kreisen in Abbildung 4.15 durch Pfeile angedeutet und werden in eine Richtungsmatrix $Dir^{(i,j),(r,s)}$ eingetragen:

$$Dir_{k,t}^{(i,j),(r,s)} = dir_{(i,j,k),(r,s,t)}. \quad (4.39)$$

Werden vom Knoten $\kappa_{(i,j,1),(r,s,1)}$ in Ebene 7 diese Richtungen bis zum Knoten $\kappa_{(i,j,5),(r,s,4)}$ in Ebene 0 zurückverfolgt, erhält man den Pfad $\Psi_1^{(i,j),(r,s)}$ mit minimalen Kosten für das Anfangspunktpaar $(P_{i,j,1}, P_{r,s,1})$. Der Algorithmus IIIa für das Berechnen eines Pfades mit minimaler Gesamtsumme der Ersetzungskosten ist zusammengefasst in Abbildung E.5 in Anhang E. Der Algorithmus gilt nur für den Fall, dass

- die 1. Zeile und 1. Spalte des zu untersuchenden Graphen jeweils als $(n_{i,j} + 1)$. Zeile bzw. $(n_{r,s} + 1)$. Spalte nochmals vorkommen,
- dass der Knoten $\kappa_{(i,j,1),(r,s,1)}$ als Knoten $\kappa_{(i,j,n_{i,j}+1),(r,s,n_{r,s}+1)}$ nochmals vorkommt,
- der Pfad vom Knoten $\kappa_{(i,j,1),(r,s,1)}$ zum Knoten $\kappa_{(i,j,n_{i,j}+1),(r,s,n_{r,s}+1)}$ laufen soll.

Da ausschließlich dafür gesorgt werden muss, dass für die Minimumbestimmung eines Knotens $\kappa_{(i,j,k),(r,s,t)}$ die Vorgängerknoten $\kappa_{(i,j,k+1),(r,s,t)}$, $\kappa_{(i,j,k),(r,s,t+1)}$ und $\kappa_{(i,j,k+1),(r,s,t+1)}$ berechnet sein müssen, können die Zeilen des Graphen beginnend bei Knoten $\kappa_{(i,j,n_{i,j}+1),(r,s,n_{r,s}+1)}$ zeilenweise von rechts nach links bis zum Knoten $\kappa_{(i,j,1),(r,s,1)}$ abgearbeitet werden.

Um den Pfad $\Psi_{\psi^{min}}^{(i,j),(r,s)}$ unter allen $n_{r,s}$ Pfaden zu detektieren, wird die lokale Distanzmatrix $\Lambda^{(i,j),(r,s)}$ zunächst so rotiert, dass die Matrizen

$$M^t = \left(\Lambda_{1\dots n_{i,j}, t\dots n_{r,s}}^{(i,j),(r,s)} \mid \Lambda_{1\dots n_{i,j}, 1\dots t-1}^{(i,j),(r,s)} \right) \quad (4.40)$$

entstehen, so dass das Element

$$M_{1,1}^t = \Lambda_{1,t}^{(i,j),(r,s)} \quad (4.41)$$

ist. Nach jeder Rotation wird die Matrix M^{t+} gebildet:

$$M^{t+} = \begin{pmatrix} M^t & M_{1\dots n_{i,j}, 1}^t \\ M_{1, 1\dots n_{r,s}}^t & M_{1,1}^t \end{pmatrix}. \quad (4.42)$$

Die Matrix M^{t+} besitzt als letzte Spalte die erste Spalte von M^t und als letzte Zeile die erste Zeile von M^t . Zusätzlich beinhaltet sie rechts unten das Element $M_{1,1}^t$ der Matrix M^t .

Es werden nacheinander alle $n_{r,s}$ Matrizen M^{t+} erzeugt. In diesen Matrizen werden jeweils mit Hilfe des Algorithmus IIIa (Abbildung E.5) die kostengünstigsten Pfade von Element $M_{1,1}^{t+}$ bis Element $M_{n_{i,j}, n_{r,s}}^{t+}$ berechnet. Der Pfad mit den minimalsten Kosten aller kostengünstigsten Pfade ist dann der gesuchte Pfad $\Psi_{\psi^{min}}^{(i,j),(r,s)}$. Der Algorithmus IVa ist zusammengefasst in Abbildung E.7 in Anhang E gezeigt.

Nach Abschluss des Algorithmus IVa sind zwar die minimalsten Kosten $g^{(i,j),(r,s)}$ der Deformation von $K_{i,j}$ in $K_{r,s}$ bekannt, die Korrespondenzliste $Q^{(i,j),(r,s)}$ allerdings noch nicht. Es wird ein weiterer Algorithmus V (Abbildung E.9 in Anhang E) benötigt, der das Zurückverfolgen (engl. backtracking) der einzelnen Richtungsangaben in der Richtungsmatrix Dir^* , d.h. in der Richtungsmatrix, in der der Pfad mit den minimalsten Kosten gefunden wurde, übernimmt.

In ψ^{min} steht der Index desjenigen Konturpunktes in $K_{i,j}$, der mit dem ersten Konturpunkt der Kontur $K_{r,s}$ korrespondiert. D.h. mit ψ^{min} müssen die Spaltenindices der Richtungsmatrix korrigiert werden, um die korrekten Konturpunktindices der Kontur $K_{i,j}$ zu erhalten.

Der Ablauf des Algorithmus ist folgendermaßen: Zunächst wird das erste Element von $Q^{(i,j),(r,s)}$ (siehe Gleichung 4.23) initialisiert:

$$q_1 := (1, \psi^{min}). \quad (4.43)$$

Im nächsten Schritt wird die Richtung in Element $Dir_{1,1}^*$ ausgelesen und entsprechend der Auswertung, ob der nächste Knoten in südlicher (S), östlicher (O) oder südöstlicher (SO) Richtung liegt, wird das Element q_2 bezogen auf q_1 berechnet. Dabei ist wichtig, dass die Indizes modulo berechnet werden, da der Pfad rechts

aus der Matrix herauslaufen kann und links dann wieder hineinlaufen muss.

Für den Fall $Dir_{1,1}^* = S$ gilt:

$$\begin{aligned} q_2 &:= (1 + 1, (\psi^{min} + 0) \bmod n_{rs}) \\ &:= (2, \psi^{min}). \end{aligned} \quad (4.44)$$

Für den Fall $Dir_{1,1}^* = O$ gilt:

$$\begin{aligned} q_2 &:= (1 + 0, (\psi^{min} + 1) \bmod n_{r,s}) \\ &:= (1, (\psi^{min} + 1) \bmod n_{r,s}). \end{aligned} \quad (4.45)$$

Für den Fall $Dir_{1,1}^* = SO$ gilt:

$$\begin{aligned} q_2 &:= (1 + 1, (\psi^{min} + 1) \bmod n_{r,s}) \\ &:= (2, (\psi^{min} + 1) \bmod n_{r,s}). \end{aligned} \quad (4.46)$$

Dieses Verfahren wird solange fortgeführt, bis das Matrixelement $Dir_{n_{i,j}+1, n_{r,s}+1}^*$ mit der Endmarkierung *end* erreicht ist. Danach sind alle Elemente der Korrespondenzliste $Q^{(i,j),(r,s)}$ bestimmt.

Für die Ermittlung einer Korrespondenzliste $\tilde{Q}^{(i,j),(r,s)}$ für eine lokale Ähnlichkeitsmatrix $\tilde{\Lambda}^{(i,j),(r,s)}$ gelten analog die Algorithmen für die lokale Distanzmatrix $\Lambda^{(i,j),(r,s)}$, bis auf die mit der Minimumsuche zusammenhängenden Algorithmenelemente. Da in $\tilde{\Lambda}^{(i,j),(r,s)}$ größere Werte für größere Ähnlichkeit, also geringere Ersetzungskosten, stehen, muss hier nach einem Pfad mit maximaler Summe gesucht werden. Der Algorithmus IIIb (Abbildung E.6 in Anhang E) und Algorithmus IVb (Abbildung E.8 in Anhang E) sind dementsprechend angepasst. Der Algorithmus V zur letztendlichen Bestimmung von $\tilde{Q}^{(i,j),(r,s)}$ besitzt keine minimumspezifischen Anteile und kann dementsprechend auch für das Rückverfolgen für $\tilde{\Lambda}^{(i,j),(r,s)}$ herangezogen werden. Dort müssen der formalen Richtigkeit halber nur $Q^{(i,j),(r,s)}$ durch $\tilde{Q}^{(i,j),(r,s)}$, $Z^{(i,j),(r,s)}$ durch $\tilde{Z}^{(i,j),(r,s)}$ und die q_z durch \tilde{q}_z ersetzt werden.

4.3.3 Einführung von Klassen

Bisher wurde ausschließlich besprochen, wie die globale Distanz (Ähnlichkeit) $g^{(i,j),(r,s)}$ ($\tilde{g}^{(i,j),(r,s)}$) und eine Abbildungsfunktion $Q^{(i,j),(r,s)}$ ($\tilde{Q}^{(i,j),(r,s)}$) von 2 Konturen unabhängig von ihrer Klassenzugehörigkeit bestimmt werden. Zur Bestimmung signifikanter Konturabschnitte ist die Klassenzugehörigkeit der Konturen von entscheidender Bedeutung, da die signifikanten Konturabschnitte einer Kontur sich ja gerade signifikant von allen Konturabschnitten der Konturen aus anderen Klassen unterscheiden sollen. Aus diesem Grund ist es jetzt an der Zeit Klassen zu bilden.

Die Klassenbildung kann überwacht oder unüberwacht geschehen.

Bei der überwachten Klassenbildung ordnet ein Trainer die Konturen in bestimmte Klassen ein. Dies geschieht gemeinhin bezogen auf eine bestimmte Aufgabenstellung. Sollen z.B. Flaschen, Getränkekartons und Joghurtbecher auseinander gehalten werden, so werden alle Flaschen der Klasse G_1 , alle Getränkekartons der Klasse G_2 und alle Joghurtbecher der Klasse G_3 zugeordnet. Damit werden die bisher unbestimmten Klassenindizes i bzw. r in $K_{i,j}$ bzw. $K_{r,s}$ bestimmt.

Der Trainer kann weiter auf die Klasseneinteilung Einfluss nehmen, falls er der Meinung ist, dass beispielsweise die Getränkekartons noch in 2 Unterklassen eingeteilt werden sollten, da es sowohl lange, schmale und kurze, breite Getränkekartons gibt. Dies hat seinen Grund darin, dass das Lernsystem nicht unnötig gezwungen werden sollte, Objekte in einer Klasse zu verarbeiten, die visuell unterschiedlich sind. In diesem Fall gäbe es in dem Beispiel jetzt 4 Klassen: G_1 (Flaschen), G_2 (Getränk kartons lang u. schmal), G_3 (Getränk kartons kurz u. breit) und G_4 (Joghurtbecher). Nach der Klassifikation können die Klassen G_2 und G_3 wieder zu einer Klasse zusammengefasst werden.

Die überwachte Klassifikation hätte natürlich auch schon am Beginn der Lernphase 1 stehen können. Der Grund dafür, dass dies erst an dieser Stelle geschieht, ist die unüberwachte Klasseneinteilung. Diese lässt sich erst zu diesem Zeitpunkt der Lernphase 1 durchführen. Für sie wird nämlich ein Maß benötigt, das die einzelnen Objekte der Lernstichprobe zueinander in Beziehung setzt. Genau solch ein Maß ist die globale Distanz $g^{(i,j),(r,s)}$ bzw. die globale Ähnlichkeit $\tilde{g}^{(i,j),(r,s)}$.

Vor der unüberwachten Klasseneinteilung kann man von 2 verschiedenen Startpartitionen ausgehen. Entweder wird davon ausgegangen, dass die Lernstichprobe L aus m^* Konturen in einer einzigen Klasse G_1 besteht:

$$G_1 = \{K_{1,1}, K_{1,2}, \dots, K_{1,j}, \dots, K_{1,m^*}\} \quad (4.47)$$

oder es wird angenommen, dass die Lernstichprobe L aus m^* Konturen in m^* Klassen G_i besteht, d.h. es wird davon ausgegangen, dass jede Kontur einer eigenen Klasse zugeordnet ist:

$$\begin{aligned} G_1 &= \{K_{1,1}\} \\ G_2 &= \{K_{2,1}\} \\ &\vdots \\ G_i &= \{K_{i,1}\} \\ &\vdots \\ G_{m^*} &= \{K_{m^*,1}\}. \end{aligned} \quad (4.48)$$

Im ersten Fall wird die erste Klasseneinteilung (Partition) schrittweise verfeinert, so dass nach und nach feinere Partitionierungen entstehen. Im zweiten Fall ist dies genau umgekehrt. Hier wird die feinste Partition sukzessiv zu größeren Partitionen zusammengefasst.

Hier wird nur der zweite Fall betrachtet: Werden für alle möglichen Konturpaare in L die globalen Distanzen $g^{(i,1),(r,1)}$ bestimmt, so erhält man die *globale Distanzmatrix*

$$\Gamma = \begin{pmatrix} g^{(1,1),(1,1)} & g^{(1,1),(2,1)} & \dots & g^{(1,1),(r,1)} & \dots & g^{(1,1),(m^*,1)} \\ g^{(2,1),(1,1)} & g^{(2,1),(2,1)} & \dots & g^{(2,1),(r,1)} & \dots & g^{(2,1),(m^*,1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g^{(i,1),(1,1)} & g^{(i,1),(2,1)} & \dots & g^{(i,1),(r,1)} & \dots & g^{(i,1),(m^*,1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g^{(m^*,1),(1,1)} & g^{(m^*,1),(2,1)} & \dots & g^{(m^*,1),(r,1)} & \dots & g^{(m^*,1),(m^*,1)} \end{pmatrix}. \quad (4.49)$$

Diese Matrix ist symmetrisch, da der Vergleich von $K_{i,1}$ mit $K_{r,1}$ das gleiche Ergebnis hat, wie der Vergleich von $K_{r,1}$ mit $K_{i,1}$:

$$g^{(i,1),(r,1)} = g^{(r,1),(i,1)}. \quad (4.50)$$

Die Elemente in der Hauptdiagonalen von Γ sind alle 0.0, da hier dieselben Konturen miteinander verglichen wurden.

Ähnliche Konturen haben ähnlich niedrige Werte in der Matrix Γ , die zu Gruppen zusammengefasst werden können. Ein Werkzeug zur automatischen Einteilung von Klassen basierend auf der globalen Distanzmatrix Γ ist die \rightarrow *Clusteranalyse*² (siehe Anhang D, [Boc74], [SL77]). Mit Hilfe eines \rightarrow *agglomerativen Clusteranalyseverfahrens* lassen sich die einzelnen Konturen nach und nach zu Klassen mit schwach ähnlichen Konturen zusammenfassen.

Genauso kann die *globale Ähnlichkeitsmatrix* $\tilde{\Gamma}$ basierend auf den berechneten globalen Ähnlichkeiten $\tilde{g}^{(i,1)(r,1)}$ aller möglichen Konturpaare bestimmt werden.

$$\tilde{\Gamma} = \begin{pmatrix} \tilde{g}^{(1,1),(1,1)} & \tilde{g}^{(1,1),(2,1)} & \dots & \tilde{g}^{(1,1),(r,1)} & \dots & \tilde{g}^{(1,1),(m^*,1)} \\ \tilde{g}^{(2,1),(1,1)} & \tilde{g}^{(2,1),(2,1)} & \dots & \tilde{g}^{(2,1),(r,1)} & \dots & \tilde{g}^{(2,1),(m^*,1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{g}^{(i,1),(1,1)} & \tilde{g}^{(i,1),(2,1)} & \dots & \tilde{g}^{(i,1),(r,1)} & \dots & \tilde{g}^{(i,1),(m^*,1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{g}^{(m^*,1),(1,1)} & \tilde{g}^{(m^*,1),(2,1)} & \dots & \tilde{g}^{(m^*,1),(r,1)} & \dots & \tilde{g}^{(m^*,1),(m^*,1)} \end{pmatrix}. \quad (4.51)$$

Auch $\tilde{\Gamma}$ ist symmetrisch. Die Hauptdiagonalelemente enthalten jeweils die Anzahl der Hälfte der Konturpunkte der Kontur, die mit sich selbst verglichen wurde.

²Im deutschen Sprachraum wird die Clusteranalyse oft auch *automatische Klassifikation* genannt.

Ähnliche Konturen haben hier ähnlich hohe Einträge in der Matrix $\tilde{\Gamma}$, die mit Hilfe der Clusteranalyse zu Klassen zusammengefasst werden können.

Nach der Klassenbildung, ob überwacht oder unüberwacht, sind die Klassenindices i bzw. r der einzelnen Konturen $K_{i,j}$ bzw. $K_{r,s}$ definiert.

4.3.4 Wichtige Konturabschnitte

Mit Hilfe der Klasseneinteilung in Abschnitt 4.3.3 lassen sich jetzt all diejenigen Konturpunkte einer bestimmten Kontur $K_{I,J}$ bestimmen, die sich möglichst stark von den Konturpunkten anderer Klassen G_r , $r \neq I$ unterscheiden, da sie hohe Ersetzungskosten besitzen.

Dazu werden zunächst alle lokalen Distanz- bzw. Ähnlichkeitsmatrizen $\Lambda^{(I,J),(r,s)}$ bzw. $\tilde{\Lambda}^{(I,J),(r,s)}$ und alle Abbildungslisten $Q^{(I,J),(r,s)}$ bzw. $\tilde{Q}^{(I,J),(r,s)}$ mit $r \neq I$ berechnet. All diejenigen Konturpunkte $P_{I,J,k}$ für die gilt

$$\Lambda_{Q_{z,1}^{(I,J),(r,s)}, Q_{z,2}^{(I,J),(r,s)}}^{(I,J),(r,s)} > \Theta \quad , \quad r \neq I \quad (4.52)$$

bzw.

$$\tilde{\Lambda}_{\tilde{Q}_{z,1}^{(I,J),(r,s)}, \tilde{Q}_{z,2}^{(I,J),(r,s)}}^{(I,J),(r,s)} < \tilde{\Theta} \quad , \quad r \neq I \quad (4.53)$$

heißten *wichtige Konturpunkte* zur Unterscheidung von $K_{I,J}$ und jeweils einer anderen Kontur $K_{r,s}$ der anderen Klassen. Θ bzw. $\tilde{\Theta}$ ist ein Schwellwert, der die minimale Unterschiedlichkeit festlegt. Dieser Schwellwert heißt *Importanzschwelle* und ist für alle Konturpunkte in der Lernstichprobe L konstant. Je größer Θ bzw. je kleiner $\tilde{\Theta}$, desto unterschiedlicher müssen die Konturpunkte sein, um den Punkt $P_{I,J,k}$ als wichtigen Konturpunkt zu deklarieren.

Die nachfolgenden Notationen und Prinzipien gelten basierend auf der lokalen Distanzmatrix $\Lambda^{(I,J),(r,s)}$ und gelten analog für die lokalen Ähnlichkeitsmatrizen $\tilde{\Lambda}^{(I,J),(r,s)}$.

Bei dem Vergleich einer bestimmten Kontur $K_{I,J}$ mit einer anderen bestimmten Kontur $K_{R,S}$ mit $I \neq R$ entstehen *wichtige Konturabschnitte*, in denen benachbarte wichtige Konturpunkte zu einer Einheit synthetisiert werden. Ein einzelner Konturpunkt wird ebenfalls als ein eigener wichtiger Konturabschnitt angesehen. Die Kontur $K_{I,J}$ besteht aus einer Anzahl $a^{(I,J),(R,S)}$ wichtiger Konturabschnitte

$$w_p^{(I,J),(R,S)} = \{P_{I,J,sta_p^{I,J}}, P_{I,J,sta_p^{I,J}+1}, \dots, P_{I,J,end_p^{I,J}}\} \quad (4.54)$$

bzgl. der Kontur $K_{R,S}$ mit $sta_p^{I,J}$ bzw. $end_p^{I,J}$ als Start- bzw. Endpunktindex des p -ten wichtigen Konturabschnitts. $w_p^{(I,J),(R,S)}$ besteht aus $F_p^{(I,J),(R,S)}$ Konturpunkten, so dass

$$end_p^{I,J} = (sta_p^{I,J} + F_p^{(I,J),(R,S)} - 1) \bmod n_{I,J} \quad (4.55)$$

ist. Die Teilkorrespondenzlisten $\Delta Q_p^{(I,J),(R,S)}$, zugehörig zu den wichtigen Konturabschnitten $w_p^{(I,J),(R,S)}$, enthalten die folgenden Elemente:

$$\Delta Q_p^{(I,J),(R,S)} = \{q_{z_p^{sta}}, q_{z_p^{sta}+1}, \dots, q_z, \dots, q_{z_p^{end}}\} \quad (4.56)$$

mit

$$z_p^{end} = z_p^{sta} + \Delta Z^{(I,J),(R,S)} - 1, \quad (4.57)$$

wobei $\Delta Z^{(I,J),(R,S)}$ die Anzahl der Elemente in $\Delta Q_p^{(I,J),(R,S)}$ ist.

Die Wichtigkeit eines solchen Konturabschnitts wird durch die Summe der Ersetzungskosten der Punkte $P_{I,J,sta_p} \dots P_{I,J,end_p}$ durch die korrespondierenden Punkte auf der Kontur $K_{R,S}$ bestimmt:

$$\theta_p^{(I,J),(R,S)} = \sum_{z=z_p^{sta}}^{z_p^{end}} \Lambda_{q_{z,1},q_{z,2}}^{(I,J),(R,S)}. \quad (4.58)$$

$\theta_p^{(I,J),(R,S)}$ heißt *Importanz* des wichtigen Konturabschnitts $w_p^{(I,J),(R,S)}$.

Wichtige und unwichtige Konturabschnitte wechseln sich entlang der Kontur $K_{I,J}$ ab.

$w^{(I,J),(R,S)}$ fasst alle wichtigen Konturabschnitte $w_p^{(I,J),(R,S)}$ der Kontur $K_{I,J}$ bezogen auf die Kontur $K_{R,S}$ zusammen:

$$w^{(I,J),(R,S)} = \left\{ \begin{array}{c} w_1^{(I,J),(R,S)}, \\ w_2^{(I,J),(R,S)}, \\ \vdots \\ w_p^{(I,J),(R,S)}, \\ \vdots \\ w_a^{(I,J),(R,S)} \end{array} \right\}. \quad (4.59)$$

Mit $w_{p,q}^{(I,J),(R,S)}$ wird der q -te Punkt des wichtigen Konturabschnitts $w_p^{(I,J),(R,S)}$ indiziert. $w_{p,q_x}^{(I,J),(R,S)}$ bzw. $w_{p,q_y}^{(I,J),(R,S)}$ bezeichnen die x - bzw. y -Koordinate des Punktes $w_{p,q}^{(I,J),(R,S)}$.

Der Algorithmus VIa (siehe Abbildung E.10 in Anhang E) berechnet $w^{(I,J),(R,S)}$ für die lokale Distanzmatrix $\Lambda^{(I,J),(R,S)}$. Es werden dort nach und nach benachbarte wichtige Konturpunkte zu wichtigen Konturabschnitten zusammengefasst. Zum Abschluss des Algorithmus wird die Benachbarung von wichtigen Konturabschnitten an der Nahtstelle zwischen End- und Anfangspunkt der Kontur $K_{I,J}$ überprüft. Da die Konturen geschlossen sind, werden die beiden benachbarten wichtigen Konturabschnitte an dieser Stelle zu einem einzigen wichtigen Konturabschnitt zusammengefasst.

Für die Liste $\tilde{w}^{(I,J),(R,S)}$, basierend auf der lokalen Ähnlichkeitsmatrix $\tilde{\Lambda}^{(I,J),(R,S)}$, gilt der modifizierte Algorithmus VIb in Abbildung E.11 in Anhang E.

Da die Kontur $K_{I,J}$ nicht nur mit einer einzigen, sondern mit allen Konturen $K_{r,s}$ aus anderen Klassen verglichen wird, ergibt sich für die Kontur $K_{I,J}$ eine Liste $W^{I,J}$ mit allen wichtigen Konturabschnitten $w^{(I,J),(r,s)}$:

$$W^{I,J} = \left\{ \begin{array}{l} \{w^{(I,J),(1,1)}, w^{(I,J),(1,2)}, \dots, w^{(I,J),(1,m_1)}\}, \\ \{w^{(I,J),(2,1)}, w^{(I,J),(2,2)}, \dots, w^{(I,J),(2,m_2)}\}, \\ \vdots \\ \{w^{(I,J),(r,1)}, w^{(I,J),(r,2)}, \dots, w^{(I,J),(r,m_r)}\}, \\ \vdots \\ \{w^{(I,J),(N,1)}, w^{(I,J),(N,2)}, \dots, w^{(I,J),(N,m_N)}\} \end{array} \right\}, \quad r \neq I. \quad (4.60)$$

Auf Basis der Liste $W^{I,J}$ werden im nächsten Abschnitt die signifikanten Konturstücke der Kontur $K_{I,J}$ bestimmt. Da dort die Information, aus welchen Klassen und welchen Konturen die einzelnen wichtigen Konturabschnitte stammen, nicht mehr benötigt wird, lässt sich $W^{I,J}$ vereinfacht schreiben:

$$W^{I,J} = \left\{ \begin{array}{l} \omega_1^{I,J} \\ \omega_2^{I,J} \\ \vdots \\ \omega_c^{I,J} \\ \vdots \\ \omega_{C^{I,J}}^{I,J} \end{array} \right\} \quad (4.61)$$

mit

$$\omega_c^{I,J} = w_p^{(I,J),(r,s)}. \quad (4.62)$$

D.h. $W^{I,J}$ besitzt $C^{I,J}$ durchnummerierte wichtige Konturabschnitte $\omega_c^{I,J}$. Es ist anzumerken, dass sich die einzelnen Abschnitte $\omega_c^{I,J}$ überlappen können.

$W_c^{I,J}$ indiziert den wichtigen Konturabschnitt $\omega_c^{I,J}$. $W_{c,d}^{I,J}$ indiziert den d -ten Konturpunkt von $\omega_c^{I,J}$ und $W_{c,d_x}^{I,J}$ bzw. $W_{c,d_y}^{I,J}$ bezeichnet die x - bzw. y -Koordinate von $W_{c,d}^{I,J}$. $sta_c^{I,J}$ und $end_c^{I,J}$ indizieren den Start- bzw. Endpunkt von $W_c^{I,J}$.

In Algorithmus VIIa (siehe Abbildung E.12 in Anhang E) ist die Erzeugung der Gesamtliste $W^{I,J}$ mit allen wichtigen Konturabschnitten der Kontur $K_{I,J}$ basierend auf den lokalen Distanzmatrizen $\Lambda^{(I,J),(r,s)}$ zusammengefasst. Dort wird vorausgesetzt, dass die Klassen überwacht festgelegt wurden.

Algorithmus VIIb berechnet $\tilde{W}^{I,J}$ basierend auf den lokalen Ähnlichkeitsmatrizen $\tilde{\Lambda}^{(I,J),(r,s)}$ und ist in Abbildung E.13 in Anhang E zusammengefasst.

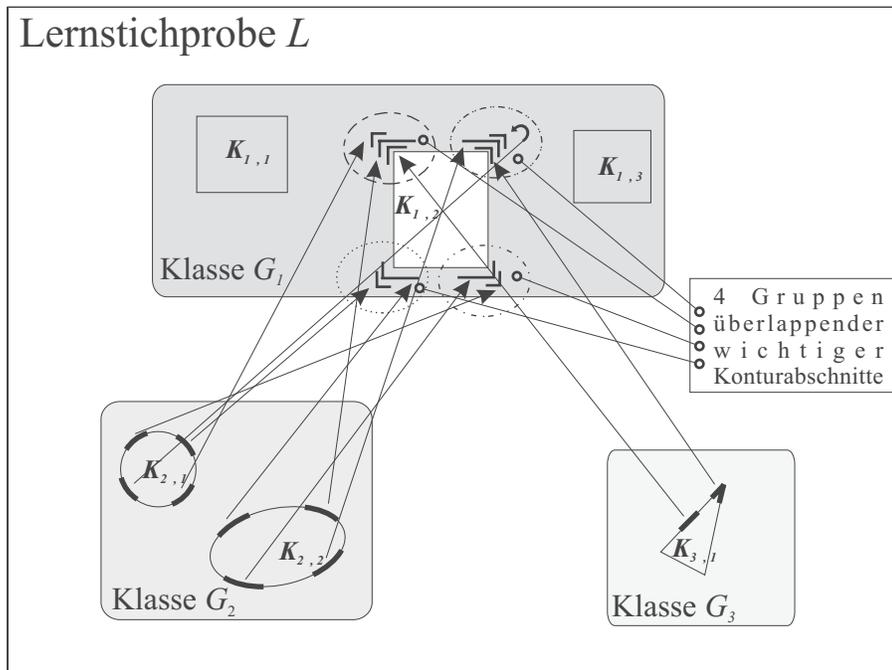


Abbildung 4.16: Beispiel für überlappende wichtige Konturabschnitte. Die Lernstichprobe besteht aus 3 Klassen mit 3 Rechtecken (G_1), 2 Ellipsen (G_2) und 1 Dreieck (G_3). Die Kontur, für die die wichtigen Konturabschnitte bestimmt worden sind, ist $K_{I,J} = K_{1,2}$. Alle wichtigen Konturabschnitte von $K_{1,2}$ und alle korrespondierenden Konturabschnitte der Konturen aus den anderen beiden Klassen G_1 und G_2 sind fett markiert. Es ergeben sich 4 verschiedene Gruppen sich überlappender wichtiger Konturabschnitte für $K_{1,2}$, die durch verschiedene gepunktete Ellipsen eingerahmt sind.

Abbildung 4.16 zeigt ein einfaches Beispiel für eine Lernstichprobe L , bestehend aus 3 Klassen mit 3 Rechtecken (G_1), 2 Ellipsen (G_2) und 1 Dreieck (G_3). Die Kontur, für die die wichtigen Konturabschnitte bestimmt worden sind, ist $K_{I,J} = K_{1,2}$. Alle wichtigen Konturabschnitte von $K_{1,2}$ und alle korrespondierenden Konturabschnitte der Konturen aus den anderen beiden Klassen G_1 und G_2 sind fett markiert. Es ergeben sich 4 verschiedene Gruppen sich überlappender wichtiger Konturabschnitte, die durch verschiedene gepunktete Ellipsen eingerahmt sind. Die Überlappungen kommen zustande, weil die Ecken des Rechtecks $K_{1,2}$ sich von mehreren Objekten aus den anderen beiden Klassen unterscheiden. Sie sind nicht exakt gleich, da die Objekte, mit denen die Kontur $K_{1,2}$ verglichen wurde, auch nicht exakt gleich sind.

4.3.5 Signifikante Konturabschnitte

Wie in Abschnitt 4.3.4 gezeigt, lässt sich für eine bestimmte Kontur $K_{I,J}$ eine Liste $W^{I,J}$ bzw. $\tilde{W}^{I,J}$ bestimmen, die alle wichtigen Konturabschnitte $\omega_c^{I,J}$ bzw. $\tilde{\omega}_c^{I,J}$ enthält, die die Kontur $K_{I,J}$ von allen Konturen $K_{r,s}$ mit $r \neq I$ unterscheidet.

Im allgemeinen enthält diese Liste sowohl $\omega_c^{I,J}$, die sich überlappen, d.h. die gemeinsame Konturpunkte besitzen, als auch $\omega_c^{I,J}$, die sich nicht überlappen, d.h. die keinen einzigen Konturpunkt gemeinsam besitzen³.

Um diese Überlappung auch formal zu beschreiben, werden die sich überlappenden wichtigen Konturabschnitte zu $U_{I,J}$ Gruppen

$$\Omega_u^{I,J} = \{\nu_{u,1}^{I,J}, \nu_{u,2}^{I,J}, \dots, \nu_{u,v}^{I,J}, \dots, \nu_{u,V_u^{I,J}}^{I,J}\} \quad (4.63)$$

zusammengefasst. Darin bedeuten mit den Gleichungen (4.62) und (4.54):

$$\nu_{u,v}^{I,J} = \omega_c^{I,J} = w_p^{(I,J),(r,s)}. \quad (4.64)$$

Eine Gruppe $\Omega_u^{I,J}$ besitzt $V_u^{I,J}$ wichtige Konturabschnitte $\nu_{u,v}^{I,J}$. Alle Gruppen werden in

$$\Omega^{I,J} = \left\{ \begin{array}{c} \Omega_1^{I,J} \\ \Omega_2^{I,J} \\ \vdots \\ \Omega_u^{I,J} \\ \vdots \\ \Omega_{U_{I,J}}^{I,J} \end{array} \right\} \quad (4.65)$$

zusammengefasst. Die wichtigen Konturabschnitte $\nu_{u,v}^{I,J}$ bestehen aus $F_{u,v}^{I,J}$ Konturpunkten $P_{I,J,k}$:

$$\nu_{u,v}^{I,J} = \{P_{I,J,sta_{u,v}^{I,J}}, P_{I,J,sta_{u,v}^{I,J}+1}, \dots, P_{I,J,end_{u,v}^{I,J}}\}. \quad (4.66)$$

Darin gilt für den Startpunktindex $sta_{u,v}^{I,J}$ mit Gleichung (4.54):

$$sta_{u,v}^{I,J} = sta_p^{I,J}. \quad (4.67)$$

Es gilt

$$F_{u,v}^{I,J} = F_p^{(I,J),(r,s)} \quad (4.68)$$

und damit ergibt sich der Endpunktindex $end_{u,v}^{I,J}$ äquivalent zu Gleichung (4.55) zu:

$$end_{u,v}^{I,J} = (sta_{u,v}^{I,J} + F_{u,v}^{I,J} - 1) \bmod n_{I,J}. \quad (4.69)$$

³Auch in diesem Abschnitt werden die Erklärungen basierend auf den lokalen Distanzmatrizen $\Lambda^{(I,J),(r,s)}$ durchgeführt. Die Erklärungen gelten analog für das Verfahren basierend auf den lokalen Ähnlichkeitsmatrizen $\tilde{\Lambda}^{(I,J),(r,s)}$.

Ausgehend von der Voraussetzung, dass Konturen aus der gleichen Klasse sich ähnlicher sind als Konturen aus verschiedenen Klassen, ist es vernünftig anzunehmen, dass Konturen aus der gleichen Klasse sich deshalb ähnlich sind, weil sie ähnliche *signifikante Konturabschnitte* besitzen, die in den Konturen der anderen Klassen nicht vorkommen.

Die wichtigen Konturabschnitte in $\Omega^{I,J}$ sind die Basis für die Bestimmung von signifikanten Konturabschnitten. Werden in $\Omega^{I,J}$ sich überlappende wichtige Konturabschnitte detektiert, die eine hohe Anzahl, bezogen auf die Anzahl der Konturen mit denen $K_{I,J}$ verglichen wurde, ausmachen und die nur wenig in ihrer Position variieren, gibt dies einen Hinweis darauf, dass es sich bei diesen wichtigen Konturabschnitten $\nu_{u,v}^{I,J}$ um höchst signifikante Konturabschnitte handeln könnte.

Synthese von wichtigen Konturabschnitten zu signifikanten Konturabschnitten

Es stellt sich die Frage, wie sich aus den einzelnen wichtigen überlappenden Konturabschnitten $\nu_{u,v}^{I,J}$ einer Gruppe $\Omega_u^{I,J}$ ein einziger signifikanter Konturabschnitt

$$s_u^{I,J} = \{P_{I,J,\sigma_u^{I,J}}, P_{I,J,\sigma_u^{I,J}+1}, \dots, P_{I,J,\epsilon_u^{I,J}}\} \quad (4.70)$$

gewinnen lässt. $\sigma_u^{I,J}$ bzw. $\epsilon_u^{I,J}$ sind der Anfangs- bzw. Endpunktindex von $s_u^{I,J}$.

Eine Möglichkeit ist es, einen bestimmten wichtigen Konturabschnitt $\nu_{u,v}^{I,J}$ auszuwählen, der bestimmte Eigenschaften besitzt. Aber was für Eigenschaften sollen das sein:

- der Längste?
- der Unterschiedlichste?
- der Mittigste?

Diese Frage muss aber nicht beantwortet werden, da es $V_u^{I,J}$ wichtige Konturabschnitte in $\Omega_u^{I,J}$ gibt, die alle ihren Beitrag zu dem signifikanten Konturabschnitt $s_u^{I,J}$ leisten sollten.

D.h. die zentrale Frage dieses Abschnitts lautet:

Wie lassen sich $V_u^{I,J}$ überlappende wichtige Konturabschnitte $\nu_{u,v}^{I,J}$ zu einem einzigen signifikanten Konturabschnitt $s_u^{I,J}$ synthetisieren?

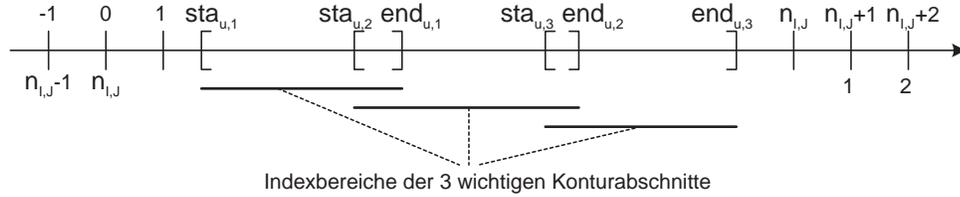


Abbildung 4.17: Dargestellt ist ein Zahlenstrahl für die Konturpunktindices. Bei den mit [markierten Indices beginnen wichtige Konturabschnitte und bei den mit] markierten Indices enden wichtige Konturabschnitte. In dem dargestellten Fall überlappen sich die wichtigen Konturabschnitte $\nu_{u,1}^{I,J}$, $\nu_{u,2}^{I,J}$ und $\nu_{u,3}^{I,J}$ nur wenig. Im Verhältnis zu dem Überlappungsbereich sind die einzelnen Abschnitte aber relativ lang, so dass insgesamt ein sehr langer signifikanter Konturabschnitt entsteht, wenn man den minimalen Startpunktindex und den maximalen Endpunktindex als Start- bzw. Endpunktindex von $s_u^{I,J}$ heranzieht.

Dazu gibt es mehrere Möglichkeiten⁴, von denen an dieser Stelle 5 intuitiv vernünftige vorgestellt werden:

1. Man nehme den minimalen Startpunktindex

$$\sigma_u^{I,J} = \min_v(sta_{u,v}^{I,J}) \quad (4.71)$$

und den maximalen Endpunktindex

$$\epsilon_u^{I,J} = \max_v(end_{u,v}^{I,J}) \quad (4.72)$$

als Start- und Endpunktindex des signifikanten Konturabschnitts $s_u^{I,J}$. Dies führt unter Umständen zu sehr langen signifikanten Konturabschnitten, da die Konturabschnitte sich evtl. nur wenig überlappen und so lange Ketten bilden können. Dazu ist in Abbildung 4.17 ein Beispiel gezeigt. Dargestellt ist ein Zahlenstrahl für die Konturpunktindices. Bei den mit [markierten Indices beginnen wichtige Konturabschnitte und bei den mit] markierten Indices enden wichtige Konturabschnitte. In dem dargestellten Fall überlappen sich die wichtigen Konturabschnitte $\nu_{u,1}^{I,J}$, $\nu_{u,2}^{I,J}$ und $\nu_{u,3}^{I,J}$ nur wenig. Im Verhältnis zu dem Überlappungsbereich sind die einzelnen Abschnitte aber relativ lang, so dass insgesamt ein sehr langer signifikanter Konturabschnitt entsteht.

⁴Bei den im Folgenden vorgeschlagenen Möglichkeiten wird von der Vorstellung ausgegangen, dass sich die Kontur $K_{i,j}$ am Konturschluss mit den Punkten $P_{i,j,n_{i,j}+1}$, $P_{i,j,n_{i,j}+2}$ usw. bzw. in die andere Richtung mit den Punkten $P_{i,j,0}$, $P_{i,j,-1}$, $P_{i,j,-2}$ usw. fortsetzt, um Begriffe wie z.B. Minimum und Maximum von Konturpunktindices unmissverständlicher benutzen zu können.

2. Man nehme den maximalen Startpunktindex

$$\sigma_u^{I,J} = \max_v(sta_{u,v}^{I,J}) \quad (4.73)$$

und den minimalen Endpunktindex

$$\epsilon_u^{I,J} = \min_v(end_{u,v}^{I,J}) \quad (4.74)$$

als Start- und Endpunktindex des signifikanten Konturabschnitts $s_u^{I,J}$. In bestimmten Fällen führt dies zu sehr kurzen oder gar nicht mehr vorhandenen signifikanten Konturabschnitten (siehe für den letzten Fall Abbildung 4.17).

3. Ein Kompromiss aus den beiden ersten Möglichkeiten ist es, den Mittelwert der Startpunktindices

$$\sigma_u^{I,J} = round(\overline{sta}_{u,v}^{I,J}) \quad (4.75)$$

mit

$$\overline{sta}_{u,v}^{I,J} = \frac{\sum_{v=1}^{V_u^{I,J}} sta_{u,v}^{I,J}}{V_u^{I,J}} \quad (4.76)$$

und der Endpunktindices

$$\epsilon_u^{I,J} = round(\overline{end}_{u,v}^{I,J}) \quad (4.77)$$

mit

$$\overline{end}_{u,v}^{I,J} = \frac{\sum_{v=1}^{V_u^{I,J}} end_{u,v}^{I,J}}{V_u^{I,J}} \quad (4.78)$$

als Start- und Endpunktindex des signifikanten Konturabschnittes $s_u^{I,J}$ heranzuziehen. *round* rundet den Mittelwert auf die nächste ganze Zahl, um wieder einen definierten Konturpunktindex zu erhalten.

4. Da es sich bei der Mittelwertbildung um einen statistischen Verarbeitungsschritt handelt, kann es sinnvoll sein, auch die Standardabweichung der Start- und Endpunktindices bzgl. ihrer Mittelwerte zu berücksichtigen. Die Start- und Endpunktindices des signifikanten Konturabschnitts $s_u^{I,J}$ ergeben sich dann zu:

$$\sigma_u^{I,J} = round\left(\overline{sta}_{u,v}^{I,J} - \sqrt{\overline{sta}_{u,v}^{I,J}}\right) \quad (4.79)$$

mit

$$\overline{sta}_{u,v}^{I,J} = \sum_{v=1}^{V_u^{I,J}} (sta_{u,v}^{I,J} - \overline{sta}_{u,v}^{I,J})^2 \quad (4.80)$$

und

$$\epsilon_u^{I,J} = \text{round}\left(\overline{end}_{u,v}^{I,J} + \sqrt{\overline{\overline{end}}_{u,v}^{I,J}}\right) \quad (4.81)$$

mit

$$\overline{\overline{end}}_{u,v}^{I,J} = \sum_{v=1}^{V_u^{I,J}} (end_{u,v}^{I,J} - \overline{end}_{u,v}^{I,J})^2. \quad (4.82)$$

5. Da sich auch mehrere einzelne Konturpunkte unter den wichtigen Konturabschnitten befinden können und diese nicht besonders ins Gewicht fallen sollen, kann es sinnvoll sein die einzelnen Start- bzw. Endpunktindices bei der Mittelwert- bzw. Varianzberechnung mit der Anzahl $F_{u,v}^{I,J}$ der Konturpunkte der einzelnen wichtigen Konturabschnitte zu gewichten. Dies führt zu dem folgenden Start- und Endpunktindex des signifikanten Konturabschnitts $s_u^{I,J}$:

$$\sigma_u^{I,J} = \text{round}\left(\overline{sta}_{u,v}^{I,J} - \sqrt{\overline{\overline{sta}}_{u,v}^{I,J}}\right) \quad (4.83)$$

mit

$$\overline{sta}_{u,v}^{I,J} = \frac{\sum_{v=1}^{V_u^{I,J}} F_{u,v}^{I,J} \cdot sta_{u,v}^{I,J}}{\sum_{v=1}^{V_u^{I,J}} F_{u,v}^{I,J}} \quad (4.84)$$

und

$$\overline{\overline{sta}}_{u,v}^{I,J} = \frac{\sum_{v=1}^{V_u^{I,J}} F_{u,v}^{I,J} (sta_{u,v}^{I,J} - \overline{sta}_{u,v}^{I,J})^2}{\sum_{v=1}^{V_u^{I,J}} F_{u,v}^{I,J}} \quad (4.85)$$

sowie

$$\epsilon_u^{I,J} = \text{round}\left(\overline{end}_{u,v}^{I,J} + \sqrt{\overline{\overline{end}}_{u,v}^{I,J}}\right) \quad (4.86)$$

mit

$$\overline{end}_{u,v}^{I,J} = \frac{\sum_{v=1}^{V_u^{I,J}} F_{u,v}^{I,J} \cdot end_{u,v}^{I,J}}{\sum_{v=1}^{V_u^{I,J}} F_{u,v}^{I,J}} \quad (4.87)$$

und

$$\overline{\overline{end}}_{u,v}^{I,J} = \frac{\sum_{v=1}^{V_u^{I,J}} F_{u,v}^{I,J} (end_{u,v}^{I,J} - \overline{end}_{u,v}^{I,J})^2}{\sum_{v=1}^{V_u^{I,J}} F_{u,v}^{I,J}}. \quad (4.88)$$

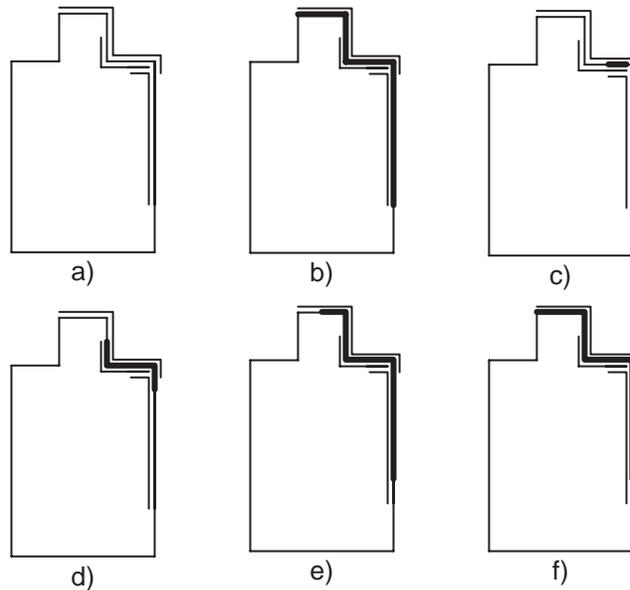


Abbildung 4.18: 5 Möglichkeiten zur Synthese wichtiger Konturabschnitte zu einem signifikanten Konturabschnitt. Die wichtigen Konturabschnitte sind neben der eigentlichen Kontur markiert. Der signifikante Konturabschnitt ist auf den Konturen fett markiert. a) Skizzierte Flasche mit 3 wichtigen Konturabschnitten. b) Minimaler Startpunkt und maximaler Endpunkt. c) Maximaler Startpunkt und minimaler Endpunkt. d) Mittelwert der Start- und Endpunkte. e) (Mittelwert-Standardabweichung) der Startpunkte und (Mittelwert+Standardabweichung) der Endpunkte. f) (gewichteter Mittelwert - gewichtete Standardabweichung) der Startpunkte und (gewichteter Mittelwert + gewichtete Standardabweichung) der Endpunkte.

Ein Beispiel für die Synthese von 3 sich überlappenden wichtigen Konturabschnitten einer skizzierten Flasche zu einem signifikanten Konturabschnitt, basierend auf den zuvor vorgeschlagenen Synthesemöglichkeiten, ist in Abbildung 4.18 qualitativ gezeigt.

Signifikanzmaß

Im allgemeinen hat die Kontur $K_{I,J}$ mehrere signifikante Konturabschnitte $s_u^{I,J}$. Da ein signifikanter Konturabschnitt auch aus einem einzelnen Konturpunkt bestehen kann (siehe dazu auch Gleichungen (4.52),(4.53)) oder die signifikanten Konturabschnitte ganz einfach verschieden signifikant sind, ist es sinnvoll, die signifikanten Konturabschnitte bzgl. ihrer Signifikanz zu ordnen und von wenig

signifikanten Konturabschnitten bei der weiteren Verwendung abzusehen.

Dazu wird ein *Signifikanzmaß* $\zeta_u^{I,J}$ für jeden signifikanten Konturabschnitt $s_u^{I,J}$ benötigt, für das im Folgenden ein Vorschlag gemacht wird. Die nachfolgend genannten 3 Abhängigkeiten dieses Signifikanzmaßes sind weitestgehend intuitiver Natur.

So ist es sicherlich vernünftig anzunehmen, dass ein Konturabschnittsbereich einer Kontur sehr signifikant ist, wenn er sich von vielen Konturen aus anderen Klassen unterscheidet. Da die Konturen alle unterschiedlich sind, werden die wichtigen Konturabschnitte dieser Kontur nicht alle den selben Konturpunktindexbereich abdecken, aber einen ähnlichen. Mit anderen Worten heißt das, dass die Konturabschnitte sich überlappen. D.h. die erste Abhängigkeit der Signifikanz kann folgendermaßen formuliert werden:

- Je größer die Anzahl $V_u^{I,J}$ sich überlappender wichtiger Konturabschnitte $\nu_{u,v}^{I,J}$, d.h. je mehr Konturen aus anderen Klassen sich von diesem Konturbereich unterscheiden, desto größer ist die Signifikanz von $s_u^{I,J}$.

Nun können sich überlappende Konturabschnitte mehr oder weniger stark überlappen. So kann es sein, dass wichtige Konturabschnitte einer Kontur nur wenige Konturpunkte am Ende des einen bzw. am Anfang eines anderen wichtigen Konturabschnitts gemeinsam haben und sich so zu sehr langen signifikanten Konturabschnitten verketteten können. Solche signifikanten Konturabschnitte entstehen aus einer wenig kompakten Gruppe sich überlappender wichtiger Konturabschnitte. Je kompakter also die Gruppe sich überlappender Konturabschnitte, desto signifikanter ist der daraus resultierende signifikante Konturabschnitt. Die Kompaktheit lässt sich ausdrücken durch die Summe der Varianzen der Start- und Endpunktindices der sich überlappenden wichtigen Konturabschnitte. Formal heißt das für die zweite Abhängigkeit der Signifikanz:

- Je kleiner die Variation

$$v_u^{I,J} = \overline{\overline{sta}}_{u,v}^{I,J} + \overline{\overline{end}}_{u,v}^{I,J} \quad (4.89)$$

der Start- und Endpunktindices der $V_u^{I,J}$ sich überlappenden $\nu_{u,v}^{I,J}$, d.h. je kompakter die Gruppe $\Omega_u^{I,J}$, desto größer die Signifikanz von $s_u^{I,J}$.

In den beiden bisher genannten Abhängigkeiten für ein Signifikanzmaß geht die Unterschiedlichkeit der sich überlappenden wichtigen Konturabschnitte einer Kontur zu den korrespondierenden wichtigen Konturabschnitten der anderen Konturen in den anderen Klassen noch nicht ein. D.h. es fehlt noch eine dritte Abhängigkeit der Signifikanz, die formal die mehr oder weniger großen Importanzen (Gl. (4.58)) der wichtigen Konturabschnitte berücksichtigt.

- Je größer die mittlere Summe der Unterschiedlichkeiten

$$d_u^{I,J} = \frac{\sum_{v=1}^{V_u^{I,J}} \theta_{u,v}^{I,J}}{V_u^{I,J}} \quad (4.90)$$

bzw.

$$\tilde{d}_u^{I,J} = \frac{\sum_{v=1}^{V_u^{I,J}} (\tilde{\theta}_{u,v}^{I,J})^{-1}}{V_u^{I,J}} \quad (4.91)$$

zwischen den $V_u^{I,J}$ einzelnen überlappenden $\nu_{u,v}^{I,J}$ und den $V_u^{I,J}$ korrespondierenden Konturabschnitten der Konturen $K_{r,s}$, desto größer ist die Signifikanz von $s_u^{I,J}$.

$\theta_{u,v}^{I,J}$ ist die umgeordnete Importanz $\theta_p^{(I,J),(r,s)}$ zwischen 2 einzelnen korrespondierenden Konturabschnitten der Konturen $K_{I,J}$ und einer Kontur $K_{r,s}$ (siehe dazu Gleichung (4.58)).

Eine Möglichkeit die 3 genannten Abhängigkeiten in einem Signifikanzmaß zu kombinieren ist die folgende:

$$\zeta_u^{I,J} = \frac{V_u^{I,J}{}^\alpha + d_u^{I,J}{}^\beta}{v_u^{I,J}{}^\gamma + 1} \quad , \quad \alpha, \beta, \gamma \in \mathbb{R}. \quad (4.92)$$

Die additive 1 im Nenner sorgt dafür, dass $\zeta_u^{I,J}$ einen endlichen Wert für $v_u^{I,J} = 0$ annimmt. α , β und γ sind Parameter, mit denen $V_u^{I,J}$, $d_u^{I,J}$ und $v_u^{I,J}$ exponentiell gewichtet werden können. Eine andere Möglichkeit gewichtet nur linear:

$$\zeta_u^{I,J} = \frac{\alpha V_u^{I,J} + \beta d_u^{I,J}}{\gamma v_u^{I,J} + 1} \quad , \quad \alpha, \beta, \gamma \in \mathbb{R}. \quad (4.93)$$

$\zeta_u^{I,J}$ wird im Folgendem abgekürzt *Signifikanz* des signifikanten Konturabschnitts $s_u^{I,J}$ genannt.

Zusätzlich sollen signifikante Konturabschnitte eine bestimmte Mindestkonturpunktanzahl L_{min} besitzen, da es beispielsweise sein kann, dass signifikante Konturabschnitte nur aus einem einzigen Konturpunkt bestehen können, wenn ein isolierter Konturpunkt über der Importanzschwelle liegt und sich nicht mit anderen wichtigen Konturabschnitten überlappt.

Mit L_{min} und der Signifikanz $\zeta_u^{I,J}$ ist es möglich, bestimmte zu kurze und weniger signifikante Konturabschnitte zu eliminieren. Mit

$$\zeta_u^{I,J} \geq \Upsilon \max_u(\zeta_u^{I,J}) \quad \wedge \quad L_{min} \geq \lambda_u^{I,J} \quad , \quad 0.0 \leq \Upsilon \leq 1.0 \quad (4.94)$$

werden nur diejenigen signifikanten Konturabschnitte in die Liste mit signifikanteren Konturabschnitten

$$S^{I,J} = \{s_1^{I,J}, s_2^{I,J}, \dots, s_u^{I,J}, \dots, s_{U_{I,J}^\Upsilon}^{I,J}\} \quad (4.95)$$

übernommen, die die Bedingung (4.94) erfüllen. Υ heißt *Signifikanzschwelle*,

$$\lambda_u^{I,J} = \epsilon_u^{I,J} - \sigma_u^{I,J} + 1 \quad (4.96)$$

ist die Anzahl der Konturpunkte in $s_u^{I,J}$ und $U_{I,J}^\Upsilon$ ist die Anzahl der signifikanteren Konturabschnitte in $S^{I,J}$. Außerdem gilt:

$$\begin{aligned} s_u^{I,J} &= \{P_{u,1}^{I,J}, P_{u,2}^{I,J}, \dots, P_{u,w}^{I,J}, \dots, P_{u,\lambda_u^{I,J}}^{I,J}\} \\ &= \{P_{I,J,\sigma_u^{I,J}}^{I,J}, P_{I,J,\sigma_u^{I,J}+1}^{I,J}, \dots, P_{I,J,k}^{I,J}, \dots, P_{I,J,\epsilon_u^{I,J}}^{I,J}\}. \end{aligned} \quad (4.97)$$

Der Punkt $P_{u,w}^{I,J}$ besitzt wiederum x - und y -Koordinaten:

$$P_{u,w}^{I,J} = \{x_{u,w}^{I,J}, y_{u,w}^{I,J}\}. \quad (4.98)$$

4.4 Ein hierarchischer Klassifikator

In Abschnitt 4.3 wurden signifikante Konturabschnitte $S^{I,J}$ für eine bestimmte Kontur $K_{I,J}$ bestimmt, die diese Kontur von den Konturen in anderen Klassen unterscheidet. Für den Aufbau einer Wissensbasis \mathcal{W} eines Klassifikators \mathcal{K} wird eine Liste mit allen signifikanten Konturabschnitten $S^{i,j}$ aller Konturen $K_{i,j}$ in der Lernstichprobe L bestimmt:

$$S = \left\{ \begin{array}{l} \{S^{1,1}, S^{1,2}, \dots, S^{1,j}, \dots, S^{1,m_1}\}, \\ \{S^{2,1}, S^{2,2}, \dots, S^{2,j}, \dots, S^{2,m_2}\}, \\ \vdots \\ \{S^{i,1}, S^{i,2}, \dots, S^{i,j}, \dots, S^{i,m_i}\}, \\ \vdots \\ \{S^{N,1}, S^{N,2}, \dots, S^{N,j}, \dots, S^{N,m_N}\} \end{array} \right\}. \quad (4.99)$$

Der Kern des Klassifikators besteht darin, die minimale Distanz zu dem ähnlichsten Abschnitt in einer zu klassifizierenden Kontur K zu bestimmen. Dazu wird der Konturabschnitt $s_u^{i,j}$ mit jedem Konturabschnitt der Kontur

$$K = \{P_1, P_2, \dots, P_k, \dots, P_{n_K}\} \quad (4.100)$$

mit gleicher Punktanzahl $\lambda_u^{i,j}$ verglichen. K besitzt n_K Konturpunkte P_k , die wiederum aus einer x - bzw. y -Koordinate bestehen:

$$P_k = \{x_k, y_k\}. \quad (4.101)$$

Die zu vergleichenden Konturabschnitte werden dazu, basierend auf dem Distanzmaß mit absoluter Orientierung bzw. mittlerer Orientierung (siehe Abschnitt 4.3.1), so gedreht und verschoben, dass das Distanzmaß minimal wird (vergleiche Gleichung (4.15)). Ob das Distanzmaß basierend auf absoluter Orientierung oder mittlerer Orientierung benutzt werden sollte, hängt davon ab, welches Distanzmaß zu Beginn der Lernphase 1 zur Bestimmung der Distanz zwischen lokalen Konturabschnitten herangezogen wurde; es sollte das gleiche benutzt werden. Die resultierende Distanz wird durch die Anzahl der verglichenen Konturpunkte $\lambda_u^{i,j}$ geteilt:

$$\Delta_{(i,j)_u,k} = \frac{\sum_{w=0}^{\lambda_u^{i,j}-1} \left((x_{u,w+1}^{i,j} - \overline{x_{u,w+1}^{i,j}}) - (x_{k+w} - \overline{x_{k+w}}) \right)^2 + \left(y_{u,w+1}^{i,j} - y_{k+w} \right)^2}{\lambda_u^{i,j}} \quad (4.102)$$

mit den Mittelwerten

$$\overline{x_{u,w+1}^{i,j}} = \frac{\sum_{w=0}^{\lambda_u^{i,j}-1} x_{u,w+1}^{i,j}}{\lambda_u^{i,j}} \quad (4.103)$$

und

$$\overline{x_{k+w}} = \frac{\sum_{w=0}^{\lambda_u^{i,j}-1} x_{k+w}}{\lambda_u^{i,j}}. \quad (4.104)$$

Diese Berechnung wird für alle n_k Konturabschnitte der Kontur K bzgl. $s_u^{i,j}$ durchgeführt und es ergibt sich eine Liste von Distanzen

$$\{ \Delta_{(i,j)_u,1}, \Delta_{(i,j)_u,2}, \dots, \Delta_{(i,j)_u,k}, \dots, \Delta_{(i,j)_u,n_k} \}. \quad (4.105)$$

Die Distanz des zu $s_u^{i,j}$ ähnlichsten Konturabschnitts in K ergibt sich dann zu

$$\Delta_{(i,j)_u,K}^{min} = \min_k (\Delta_{(i,j)_u,k}). \quad (4.106)$$

Die einfachste Möglichkeit für einen Klassifikator \mathcal{K} ist ein *Nächster-Nachbar-Klassifikator* (engl. *nearest neighbour classifier*), dessen Wissensbasis \mathcal{W} ausschließlich aus den signifikanten Konturabschnitten in S besteht:

$$\mathcal{W} = S \quad (4.107)$$

Für jeden Konturabschnitt $s_u^{i,j}$ in \mathcal{W} wird die Distanz $\Delta_{(i,j)_u,K}^{min}$ bestimmt und die Kontur K wird derjenigen Klasse \hat{g} zugeordnet

$$\hat{g} = \mathcal{K}(\mathcal{W}, K), \quad (4.108)$$

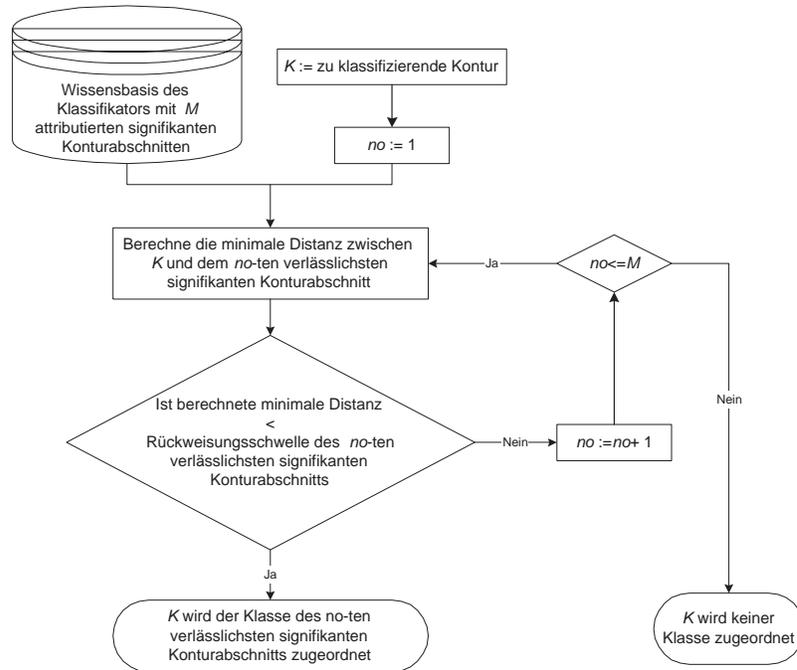


Abbildung 4.19: Flussdiagramm des hierarchischen Klassifikators.

aus der der signifikante Konturabschnitt $s_u^{i,j}$ stammt, der den minimalsten Wert Δ_K^{MIN} über alle $\Delta_{(i,j)u,K}^{min}$ hat:

$$\Delta_K^{MIN} = \min_{i,j,u} \Delta_{(i,j)u,K}^{min} \quad (4.109)$$

Der Nächster-Nachbar-Klassifikator hat den Nachteil, dass er nicht effizient ist, da er alle signifikanten Konturabschnitte in die Berechnung seines Klassifikationsergebnisses mit einbezieht. Der hier vorgeschlagene Klassifikator ist ein *hierarchischer Klassifikator*. In Abbildung 4.19 ist das Prinzip des hierarchischen Klassifikators in Form eines Flussdiagrammes dargestellt. Den einzelnen signifikanten Konturabschnitten werden eine Verlässlichkeit, die sogenannte *Reliabilität* $\rho_u^{i,j}$, für ein richtiges Klassifikationsergebnis und eine Rückweisungsschwelle, die sogenannte *Konfidenz* $\tau_u^{i,j}$, zugeordnet. Die signifikanten Konturabschnitte werden gemäß ihrer Reliabilität geordnet.

Bei der Klassifikation wird nach und nach, angefangen beim verlässlichsten signifikanten Konturabschnitt, die Distanz $\Delta_{(i,j)u,K}^{min}$ zu dem ähnlichsten Konturabschnitt in der zu klassifizierenden Kontur K festgestellt. Ist die Distanz zuverlässig genug, d.h. kleiner der Konfidenz $\tau_u^{i,j}$, wird die Kontur K derjenigen Klasse i zugeordnet, aus der $s_u^{i,j}$ stammt. Wird die Konfidenz überschritten, wird mit dem

zweitverlässlichsten signifikanten Konturabschnitt fortgefahren. Wird die zugehörige Konfidenz wieder überschritten, wird mit dem drittverlässlichsten signifikanten Konturabschnitt fortgefahren usw., bis die berechnete Distanz kleiner der zu $s_u^{i,j}$ gehörenden Konfidenz $\tau_u^{i,j}$ wird. Ist keine Distanz $\Delta_{(i,j)_u,K}^{min}$ kleiner irgendeiner Konfidenz, so wird die Kontur einer unbekanntes Klasse, der sogenannten *Restklasse*, zugeordnet.

Damit arbeitet der hier vorgeschlagene hierarchische Klassifikator wesentlich effizienter als der Nächster-Nachbar-Klassifikator, da er, wie noch in den experimentellen Ergebnissen (siehe Kapitel 5) gezeigt wird, im Mittel nur einen Bruchteil der signifikanten Konturabschnitte in seiner Wissensbasis für die Klassifikation benötigt.

4.4.1 Erzeugung einer Vergleichstabelle

Den signifikanten Konturabschnitten $s_u^{i,j}$ werden je 2 Attribute zugeordnet. Dies ist zum einen die Reliabilität $\rho_u^{i,j}$ und die Konfidenz $\tau_u^{i,j}$. Um diese zu bestimmen, wird der signifikante Konturabschnitt $s_u^{i,j}$ mit allen Konturen aus der Lernstichprobe L verglichen. Dies führt zu der Vergleichstabelle

$$\mathcal{V}_u^{i,j} = \left\{ \begin{array}{l} \{ \Delta_{(i,j)_u,(1,1)}^{min}, \Delta_{(i,j)_u,(1,2)}^{min}, \dots, \Delta_{(i,j)_u,(1,s)}^{min}, \dots, \Delta_{(i,j)_u,(1,m_1)}^{min} \} \\ \{ \Delta_{(i,j)_u,(2,1)}^{min}, \Delta_{(i,j)_u,(2,2)}^{min}, \dots, \Delta_{(i,j)_u,(2,s)}^{min}, \dots, \Delta_{(i,j)_u,(2,m_2)}^{min} \} \\ \vdots \\ \{ \Delta_{(i,j)_u,(r,1)}^{min}, \Delta_{(i,j)_u,(r,2)}^{min}, \dots, \Delta_{(i,j)_u,(r,s)}^{min}, \dots, \Delta_{(i,j)_u,(r,m_r)}^{min} \} \\ \vdots \\ \{ \Delta_{(i,j)_u,(N,1)}^{min}, \Delta_{(i,j)_u,(N,2)}^{min}, \dots, \Delta_{(i,j)_u,(N,s)}^{min}, \dots, \Delta_{(i,j)_u,(N,m_N)}^{min} \} \end{array} \right\} \quad (4.110)$$

mit $\Delta_{(i,j)_u,(r,s)}^{min}$ als der minimalen Distanz zwischen dem signifikanten Konturabschnitt $s_u^{i,j}$ und dem dazu ähnlichsten Konturabschnitt der Kontur $K_{r,s}$.

Für exakt ein Element in $\mathcal{V}_u^{i,j}$ gilt

$$(i = r) \wedge (j = s) \quad (4.111)$$

mit

$$\Delta_{(i,j)_u,(i,j)}^{min} = 0, \quad (4.112)$$

da in diesem Fall der Konturabschnitt $s_u^{i,j}$ mit sich selbst verglichen wurde.

4.4.2 Attribute eines signifikanten Konturabschnittes

Zur Bestimmung der Reliabilität und der Konfidenz eines bestimmten signifikanten Konturabschnittes $s_u^{i,j}$ werden die Distanzen in $\mathcal{V}_u^{i,j}$ zunächst geordnet, und zwar beginnend mit dem niedrigsten Eintrag.

Reliabilität

Ein Höchstmaß an Verlässlichkeit eines Konturabschnitts $s_u^{i,j}$ aus der Klasse i für ein korrektes Klassifikationsergebnis ist dann gegeben, wenn die ersten m_i Distanzen $\Delta_{(i,j)_u,(r,s)}^{min}$ zu Konturen gehören, die aus der gleichen Klasse i stammen, wie der Konturabschnitt $s_u^{i,j}$. Sind unter den ersten m_i Distanzen, die mit Konturen aus anderen Klassen $r \neq i$ korrespondieren, so sinkt die Verlässlichkeit von $s_u^{i,j}$. D.h. es gibt nur noch $m_i^* \leq m_i$ Distanzen, die mit der gleichen Klasse korrespondieren wie $s_u^{i,j}$, unter den ersten m_i Distanzen. Formal lässt sich dieser Zusammenhang für die Reliabilität folgendermaßen ausdrücken:

$$\rho_u^{i,j} = \frac{m_i^*}{m_i} \quad , \quad m_i > 1 \quad (4.113)$$

$\rho_u^{i,j}$ bewegt sich in einem Wertebereich zwischen 0.0 (keine Verlässlichkeit) und 1.0 (höchste Verlässlichkeit).

Konfidenz

Die erste der geordneten Distanzen $\Delta_{(i,j)_u,(r,s)}^{min}$ für die gilt

$$r \neq i, \quad (4.114)$$

d.h. die mit einer anderen Klasse korrespondiert als der signifikante Konturabschnitt $s_u^{i,j}$, wird als Konfidenz $\tau_u^{i,j}$ herangezogen:

$$\tau_u^{i,j} = \min_r (\Delta_{(i,j)_u,(r,s)}^{min} | i \neq r) \quad (4.115)$$

Der mit $\tau_u^{i,j}$ korrespondierende Konturabschnitt der Kontur $K_{r,s}$ ist der ähnlichste Konturabschnitt in der Lernstichprobe L , der zu einer Kontur gehört, die nicht aus der Klasse stammt, aus der $s_u^{i,j}$ kommt. Deshalb ist es vernünftig diese Distanz als Konfidenz $\tau_u^{i,j}$ heranzuziehen.

Wissensbasis des hierarchischen Klassifikators

Werden die Reliabilitäten $\rho_u^{i,j}$ und die Konfidenzen $\tau_u^{i,j}$ für jeden signifikanten Konturabschnitt $s_u^{i,j}$ berechnet und werden die $s_u^{i,j}$ und die zugeordneten $\tau_u^{i,j}$ gemäß ihrer zugeordneten Reliabilität geordnet, so erhält man die Wissensbasis

$$\mathcal{W} = \begin{bmatrix} s^1 & \hat{g}^1 & \tau^1 & \rho^1 \\ s^2 & \hat{g}^2 & \tau^2 & \rho^2 \\ \vdots & \vdots & \vdots & \vdots \\ s^{no} & \hat{g}^{no} & \tau^{no} & \rho^{no} \\ \vdots & \vdots & \vdots & \vdots \\ s^M & \hat{g}^M & \tau^M & \rho^M \end{bmatrix} \quad (4.116)$$

mit

$$\rho^{no} > \rho^{no+1} \quad (4.117)$$

und

$$s^{no} = s_u^{i,j} \quad (4.118)$$

$$= \{P_{i,j,\sigma^{no}}, \dots, P_{i,j,k}, \dots, P_{i,j,\epsilon^{no}}\} \quad (4.119)$$

$$\tau^{no} = \tau_u^{i,j} \quad (4.120)$$

$$\rho^{no} = \rho_u^{i,j}. \quad (4.121)$$

für den hierarchischen Klassifikator. \hat{g}^{no} ist der Klassenindex der Klasse aus der s^{no} stammt. σ^{no} und ϵ^{no} bezeichnen den Start- bzw. Endpunktindex des signifikanten Konturabschnitts s^{no} . Der Algorithmus VIII des hierarchischen Klassifikators ist in Abbildung E.14 in Anhang E zusammengefasst.

Kapitel 5

Experimentelle Ergebnisse

In diesem Kapitel werden Experimente zu allen Teilbereichen der in Kapitel 4 beschriebenen *automatischen Merkmalsynthese* vorgestellt.

In Abschnitt 5.1 wird der Unterschied zwischen dem lokalen Distanzmaß basierend auf absoluter bzw. mittlerer Orientierung vorgestellt.

In Abschnitt 5.2 werden beispielhaft lokale Distanz- bzw. Ähnlichkeitsmatrizen für verschiedene Objektkonturen gezeigt und besprochen. Dabei wird gezeigt, wie sich die Variation der Parameter Abschnittslänge l_{const} bzw. maximal erlaubter Distanz E auf die Matrizen auswirken.

Abschnitt 5.3 zeigt Beispiele für Korrespondenzlisten mit minimalen Kosten zwischen verschiedenen diskreten Konturen. Es werden dort auch die Ergebnisse der Auswirkungen bestimmter Deformationen (z.B. Perspektive, Zerstörung, Eigendeformation) ein- und derselben Konturen auf die Korrespondenzliste und die globale Ähnlichkeit vorgestellt.

Globale Distanz- bzw. Ähnlichkeitsmatrizen von größeren Datensätzen werden in Abschnitt 5.4 präsentiert. In einem Fall werden hier die Ergebnisse mit einem Datensatz aus der Literatur verglichen. Ferner werden in diesem Abschnitt die Ergebnisse der Clusteranalyse, d.h. der unüberwachten Klasseneinteilung vorgestellt.

Im Abschnitt 5.5 werden die Ergebnisse des hierarchischen Klassifikators basierend auf signifikanten Konturabschnitten anhand verschiedener Datensätzen präsentiert.

Im abschließenden Abschnitt 5.6 wird dann noch gezeigt, wie ein System, basierend auf den globalen Ähnlichkeiten von Objekten, autonom ein Modell seiner Umwelt erlernen kann.

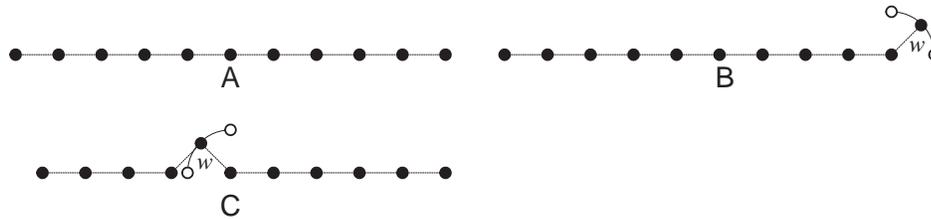


Abbildung 5.1: Konturabschnitt A besteht aus 11 gleichabständigen Punkten auf einer angedeuteten Geraden. Konturabschnitt B ist die am Endpunkt gestörte Version des Konturabschnitts A . Der Endpunkt wird durch Variation des Winkels w zwischen 0 und 90 Grad gestört. Bei Konturabschnitt C ist ein Punkt innerhalb des Konturabschnitts gegenüber Konturabschnitt A gestört. Der Störungswinkel w variiert auch hier zwischen 0 und 90 Grad. Die links vom gestörten Punkt befindlichen Konturpunkte werden so entlang der gedachten Geraden verschoben, dass alle Punktabstände des Konturabschnitts wieder gleich sind.

5.1 Distanz von 2 Konturabschnitten

Im Abschnitt 4.3.1 wurden 2 Distanzmaße für Konturabschnitte gleicher Länge vorgestellt. Das eine basiert auf der sogenannten absoluten Orientierung, das andere auf der sogenannten mittleren Orientierung.

Um den Unterschied zu beleuchten, wurde folgendes Experiment durchgeführt: Eine gerade Linie wird gleichabständig mit 11 Punkten abgetastet. Danach wird diese Linie entweder am Endpunkt oder an einem dem mittleren Punkt benachbarten Punkt gestört (siehe dazu Abbildung 5.1). Konturabschnitt A ist der ungestörte Konturabschnitt. Konturabschnitt B ist der am Endpunkt gestörte Konturabschnitt und Konturabschnitt C ist in einem dem mittleren Punkt benachbarten Punkt gestört.

Die Abbildung 5.2 zeigt die rotierten und gegeneinander verschobenen Konturabschnitte A und B bzw. A und C zum einen basierend auf absoluter Orientierung und zum anderen basierend auf mittlerer Orientierung für eine Störung von $|w| = \pi/4$.

In Abbildung 5.3 sind die Distanzkurven (Summe der quadrierten euklidischen Distanzen SQD) für das Variieren des Störungswinkels von $|w| = 0$ bis $|w| = \pi/2$ aufgetragen. In Abbildung 5.3a) und c) sieht man die Distanzkurven für die Störung des Endpunktes. Man erkennt deutlich, dass der Fehler, basierend auf der mittleren Orientierung, fast ausschließlich durch die Distanz der Endpunkte bestimmt wird. Bei der Methode der absoluten Orientierung bewirkt die Störung des Endpunktes ein 'Abheben' des restlichen Konturabschnitts gegenüber dem

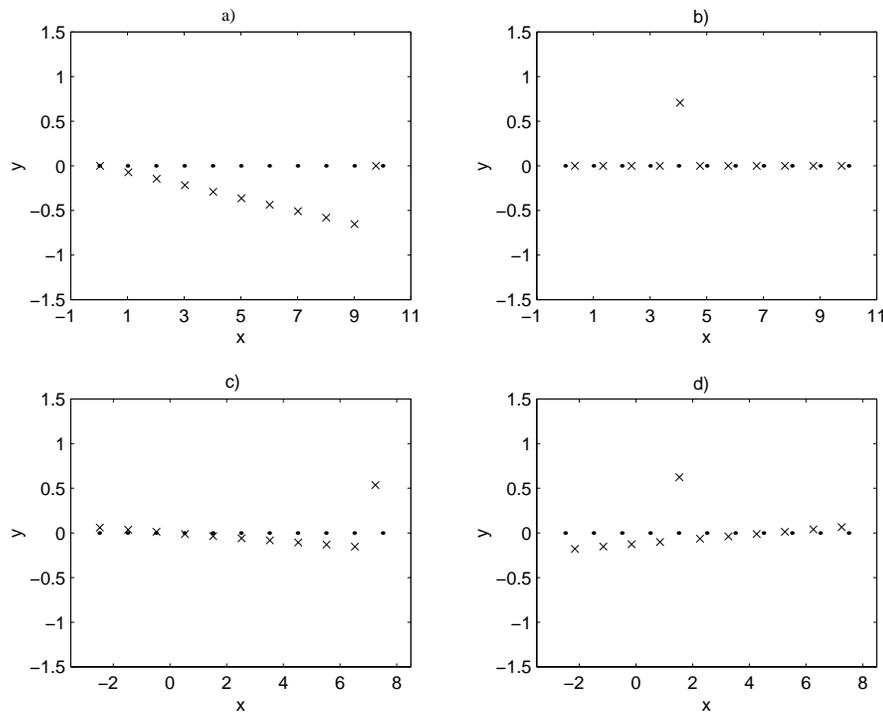


Abbildung 5.2: Konturabschnitte A ($\cdot\cdot\cdot$) und B (\times) bzw. A ($\cdot\cdot\cdot$) und C (\times) bei einem Störungswinkel von jeweils 45 Grad so gegeneinander verschoben und rotiert, dass die Summe ihrer quadratischen euklidischen Abstände minimal wird. Achtung: unterschiedliche Maßstäbe in x - und y -Richtung zur besseren Erkennbarkeit der Störungseffekte. a) Endpunkt gestört, absolute Orientierung. b) Innerhalb des Konturabschnitts liegender Punkt gestört, absolute Orientierung. c) Endpunkt gestört, mittlere Orientierung. d) Innerhalb des Konturabschnitts liegender Punkt gestört, mittlere Orientierung.

ungestörten Konturabschnitt. Dies führt in der Summe zu einer weit größeren Distanz als bei der Methode der mittleren Orientierung. Eine Störung des Endpunktes wirkt sich also wie zu erwarten war bei mittlerer Orientierung des Konturabschnitts weit weniger auf die Summe der quadratischen euklidischen Distanzen aus als bei der absoluten Orientierung.

Die Abbildungen 5.3b) und d) zeigen die Distanzkurven für die Störung eines innerhalb des Konturabschnitts liegenden Konturpunktes. D.h. die mittlere Orientierung ist gestört, die absolute Orientierung allerdings nicht. Trotzdem ergeben sich nahezu identische Distanzkurven, d.h. die Störung der mittleren Orientierung wirkt sich im Verhältnis zur absoluten Orientierung kaum aus.

Damit ist die Distanzberechnung basierend auf mittlerer Orientierung aber

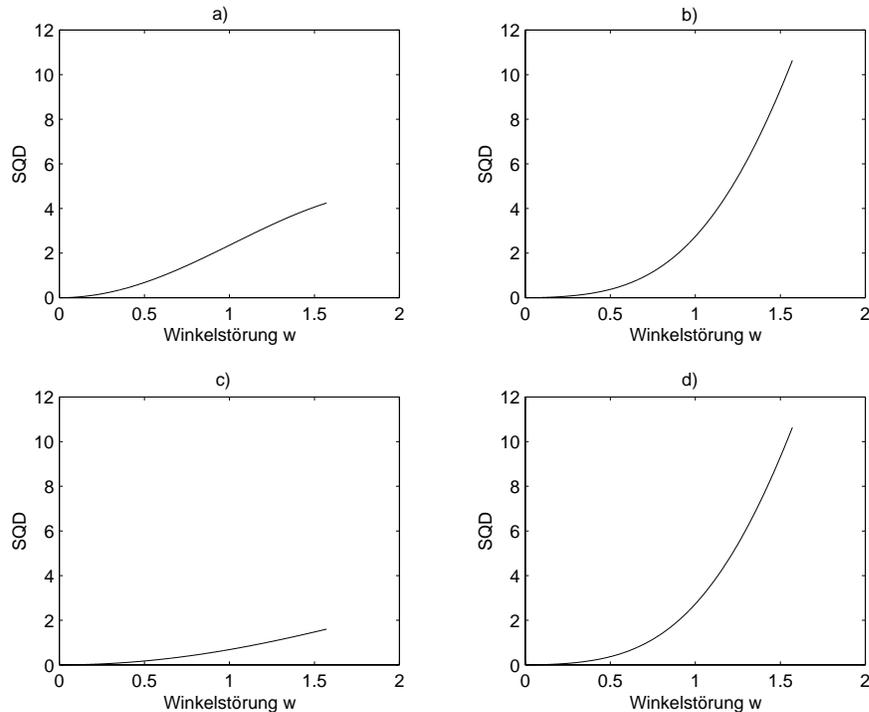


Abbildung 5.3: Summe der quadratischen euklidischen Distanzen SQD aufgetragen gegen die Winkelstörung w im Bereich von $0 \dots \pi/2$. a) Endpunkt gestört, absolute Orientierung. b) Innerhalb des Konturabschnitts liegender Punkt gestört, absolute Orientierung. c) Endpunkt gestört, mittlere Orientierung. d) Innerhalb des Konturabschnitts liegender Punkt gestört, mittlere Orientierung.

insgesamt robuster, da sich bei dieser Methode die Störung von Anfangs- bzw. Endpunkten nicht so stark auswirkt wie bei der Distanzberechnung basierend auf absoluter Orientierung.

5.2 Lokale (Distanz-)Ähnlichkeitsmatrizen

Im Abschnitt 4.3.2 wurde die Erzeugung lokaler Distanz- und Ähnlichkeitsmatrizen für den Vergleich von 2 Konturen vorgestellt.

Zunächst werden an dieser Stelle die experimentellen Ergebnisse für 2 einfache geometrische Figuren - ein Dreieck und ein Viereck - vorgestellt, um die prinzipiellen Eigenschaften der unterschiedlichen Verfahren darzustellen.

Um Skalierungsinvarianz zu erreichen, werden die beiden Konturen zunächst auf dieselbe Bogenlänge von 500 Pixeleinheiten skaliert. Danach werden die bei-

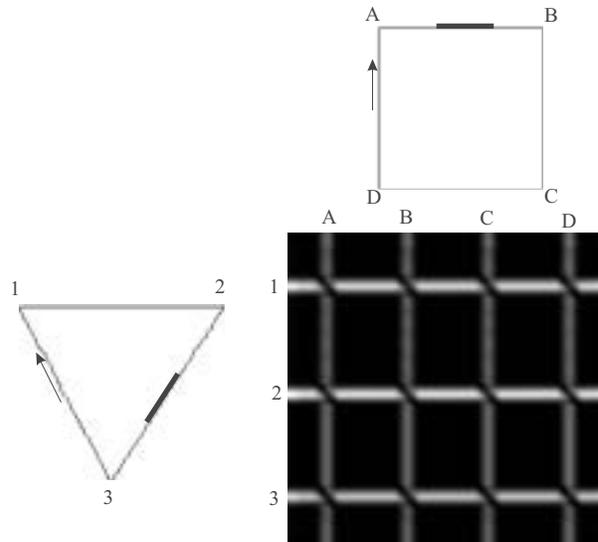


Abbildung 5.4: Abgebildet sind eine visualisierte lokale Distanzmatrix, basierend auf mittlerer Orientierung für $l_{const} = 10$, und die beiden Konturen, ein Dreieck und ein Viereck, deren Vergleich zu dieser Matrix geführt hat. Die Konturen wurden zunächst auf gleiche Bogenlänge von 500 Pixeleinheiten skaliert und dann mit 2.0 Pixeleinheiten gleichabständig abgetastet, so dass das abgebildete Dreieck mit 242 und das abgebildete Viereck mit 248 Konturpunkten entstand. Jeder Eintrag der 242×248 -Matrix wurde dazu in einen diskreten Wertebereich zwischen 0 und 255 skaliert und kann so einfach als Grauwert eines Pixels in einem Bild dargestellt werden. Das Pixel links oben im Bild visualisiert das Element (1,1) der Matrix und das Pixel rechts unten visualisiert das Element (242,248) der Matrix. Hohe Werte in der Matrix werden hell und niedrige Werte in der Matrix werden dunkel dargestellt.

den Konturen mit einem Abstand von 2.0 Pixeleinheiten abgetastet, um für die lokalen Distanzmaße zu gewährleisten, dass alle Konturpunkte denselben Abstand haben. Dazu wird zwischen den Originalkonturpunkten linear interpoliert. So entstehen die in Abbildung 5.4 dargestellten Konturen des Dreiecks mit 242 und des Vierecks mit 248 Konturpunkten. In Abbildung 5.4 ist zusätzlich die lokale Distanzmatrix, basierend auf mittlerer Orientierung des Dreiecks und des Vierecks für $l_{const} = 10$ (beispielhaft fett markiert auf den Konturen), visualisiert. Jeder Eintrag der 242×248 -Matrix wurde dazu in einen diskreten Wertebereich zwischen 0 und 255 skaliert und kann so einfach als Grauwert eines Pixels in einem Bild dargestellt werden. Das Pixel links oben im Bild visualisiert das Element (1,1) der Matrix und das Pixel rechts unten visualisiert das Element (242,248) der

Matrix. Hohe Werte in der Matrix werden hell und niedrige Werte in der Matrix werden dunkel dargestellt. Da bei den lokalen Distanzmatrizen niedrige Werte eine hohe Ähnlichkeit bedeuten, stellen dunkle Flächen in der Matrix die Korrespondenz zwischen Konturpunkten her, deren zugehörige lokale Konturabschnitte sehr ähnlich sind. Die kleineren helleren Flächen wiederum zeigen Korrespondenzen zwischen Konturpunkten an, die sich unähnlich sind.

Die beiden Konturen beginnen dort, wo die Pfeile beginnen. Die Konturpunkte folgen dann aufeinander in Pfeilrichtung, also in diesem Fall rechtsherum. Das Voranschreiten auf den Konturen korrespondiert beim Dreieck mit immer weiter unten angesiedelten Zeilen und beim Viereck mit immer weiter rechts angesiedelten Spalten im Matrixbild. Entlang den Konturen sind ausgezeichnete Stellen markiert. Bei dem Dreieck sind dies die Ecken 1-3 und beim Viereck die Ecken A-D. Diese Markierungen finden sich auch an den mit den Konturpunktindices korrespondierenden Zeilen bzw. Spalten der Matrix wieder.

Es ergeben sich 3 relativ schmale helle Streifen in Zeilenrichtung, die mit den 3 Ecken des Dreiecks korrespondieren und 4 relativ schmale helle Streifen in Spaltenrichtung, die mit den 4 Ecken des Vierecks korrespondieren. Die relativ breiten dunklen Flächen zwischen den hellen Streifen korrespondieren mit Konturpunkten, deren zugehörige lokale Konturabschnitte Geradenstücke der beiden Konturen und sich damit sehr ähnlich sind. Die hellen Streifen wiederum sind das Ergebnis der Vergleiche von Ecken mit Geradenstücken. Die schmalen dunklen Kanäle im Kreuzungsbereich der hellen Streifen resultieren aus dem Vergleich von Ecken des Dreiecks mit Ecken des Quadrats.

In diesem einfachen Beispiel lässt sich die Struktur einer solchen lokalen Distanzmatrix gut erkennen.

Abbildung 5.5 zeigt die Unterschiede der einzelnen Methoden zur Erzeugung lokaler (Distanz-) Ähnlichkeitsmatrizen für das Dreieck und das Viereck aus Abbildung 5.4. Abbildung 5.5a zeigt die lokale Distanzmatrix basierend auf mittlerer Orientierung für $l_{const} = 10$. Abbildung 5.5b visualisiert die lokale Distanzmatrix basierend auf absoluter Orientierung für $l_{const} = 10$. Abbildung 5.5c stellt die lokale Ähnlichkeitsmatrix basierend auf mittlerer Orientierung für eine zulässige maximale Distanz von $E = 200.0$ dar und Abbildung 5.5d schließlich zeigt die lokale Ähnlichkeitsmatrix basierend auf absoluter Orientierung für $E = 200.0$.

In Abbildung 5.5b sieht man deutlich die Diskretisierungseffekte der 2 diagonalen Seiten des Dreiecks. Diese Seiten sind nicht so glatt wie die obere Seite des Dreiecks. Diese Effekte sieht man sehr schön in allen vertikalen hellen Streifen, in jeweils 2 Bereichen. Die Streifen erscheinen in diesen Bereichen nicht homogen. Das liegt an der Empfindlichkeit der Methode basierend auf absoluter Orientierung, die sich nur auf die Anfangs- und Endpunkte der Konturabschnitte zur Berechnung der Orientierung stützt. Ansonsten unterscheiden sich die beiden lokalen Distanzmatrizen von ihrer Struktur her kaum.

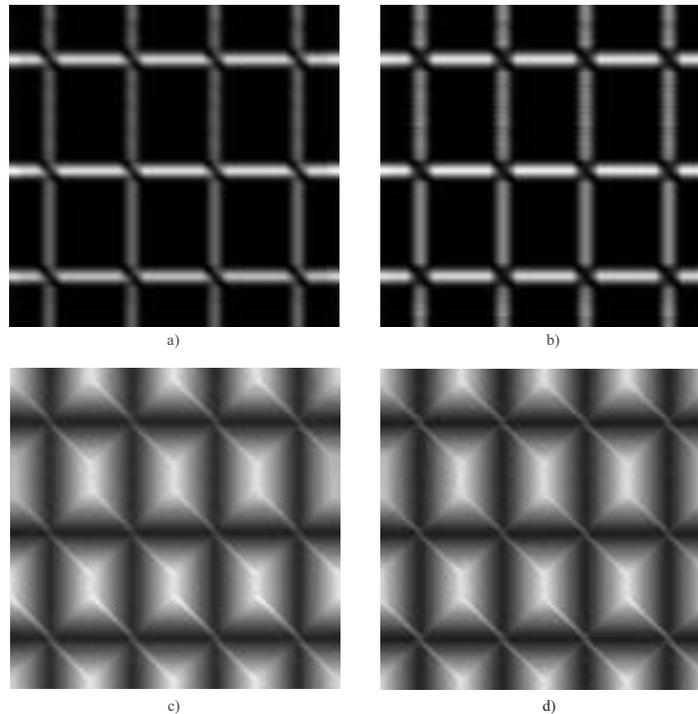


Abbildung 5.5: Die 4 verschiedenen Methoden zur (Distanz-) Ähnlichkeitsmatrixerzeugung im Vergleich. Die lokalen (Distanz-)Ähnlichkeitsmatrizen sind für das Dreieck und das Viereck aus Abbildung 5.4 erzeugt worden: a) lokale Distanzmatrix, mittlere Orientierung, $l_{const} = 10$; b) lokale Distanzmatrix, absolute Orientierung, $l_{const} = 10$; c) lokale Ähnlichkeitsmatrix, mittlere Orientierung, $E = 200.0$; d) lokale Ähnlichkeitsmatrix, absolute Orientierung, $E = 200.0$.

Die Abbildungen 5.5c und 5.5d zeigen visualisierte lokale Ähnlichkeitsmatrizen. D.h. den einzelnen Elementen der Matrix werden maximal mögliche l^{max} zugeordnet, die bei einer vorgegebenen maximalen Distanz E um einen Konturpunkt herum erreicht werden können. D.h. je größer l^{max} , desto größer der Matrixeintrag und, im Unterschied zur Visualisierung der lokalen Distanzmatrix, desto heller der Grauwert des zugehörigen Pixels. Grob gesagt verhalten sich die lokalen Distanz- und Ähnlichkeitsmatrizen also invers zueinander: Dort, wo die lokale Distanzmatrix helle Bereiche aufweist, hat die lokale Ähnlichkeitsmatrix dunkle Bereiche und umgekehrt. In den lokalen Ähnlichkeitsmatrizen sind es jetzt die schmalen dunklen horizontalen und vertikalen Streifen, die aus einem Vergleich von Ecken mit Geradenstücken herrühren.

Zum anderen sind die mit den dunklen Bereichen in den lokalen Distanzmatrizen korrespondierenden hellen Bereiche der lokalen Ähnlichkeitsmatrizen we-

sentlich feiner quantisiert. Dies liegt daran, dass bei den lokalen Distanzmatrizen nicht unterschieden wird, ob 2 vergleichende Geradenstück mehr am Anfang oder mehr am Ende einer Dreiecks- bzw. Vierecksseite liegen. Dies wird im Gegensatz dazu bei den lokalen Ähnlichkeitsmatrizen getan, da diese eine maximal mögliche Länge l^{max} bzgl. E um einen Konturpunkt herum suchen. Wird E klein genug gewählt, so wird l^{max} um so kürzer, je näher sich der Mittelpunkt des Konturabschnitts an einer Ecke befindet.

Auch die beiden lokalen Ähnlichkeitsmatrizen basierend auf mittlerer und absoluter Orientierung unterscheiden sich von ihrer Struktur her kaum. Auch hier lässt sich bei genauer Betrachtung der Diskretisierungseffekt der diagonal verlaufenden Dreiecksseiten in den damit korrespondierenden hellen Flächen der visualisierten Matrix basierend auf absoluter Orientierung beobachten. Dieser Effekt tritt hier allerdings lange nicht so deutlich auf wie bei den lokalen Distanzmatrizen.

In der Abbildung 5.6 ist die Abhängigkeit der lokalen Distanzmatrix basierend auf mittlerer Orientierung für das Dreieck und das Viereck aus Abbildung 5.4 in Abhängigkeit vom Parameter l_{const} dargestellt. Für $l_{const} = 2, 5, 10, 25$ in den Abbildungen 5.6a-d erkennt man eine zunehmende Verbreiterung der hellen Streifen, die mit den Ecken korrespondieren. Das lässt sich dadurch erklären, dass die Konturabschnitte in diesem Bereich über immer mehr Konturabschnittsmittelpunktindexe eine Ecke beinhalten. Die Struktur dieser Matrizen bleibt aber ähnlich.

Für $l_{const} = 50, 100$ (Abbildung 5.6e-f) ändert sich dies stark. Zwar ist immer noch die Anzahl der Ecken erkennbar, doch sind die klaren horizontalen und vertikalen hellen Streifen aus den Abbildungen 5.6a-d verschwunden. Dies gründet darauf, dass die lokalen Konturabschnitte jetzt so lang sind, dass keine reinen Geradenstücke in den Vergleich mit einer Ecke mehr eingehen. Ein lokaler Konturabschnitt besitzt in diesen Fällen mindestens eine Ecke. Im Fall von $l_{const} = 50$ können die Konturabschnitte 2 Ecken, im Fall von $l_{const} = 100$ bis zu 3 Ecken besitzen.

Abbildung 5.7 zeigt eine Reihe von lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung, die durch einen Vergleich des Dreiecks und dem Viereck aus Abbildung 5.4 für $E = 50.0, 100.0, 200.0, 1000.0, 2500.0, 10000.0$ entstanden sind. Hier sind bei steigendem E die gleichen Effekte wie bei steigendem l_{const} in Abbildung 5.6 zu beobachten. Für $E = 50.0, 100.0, 200.0, 1000.0$ bleibt die Struktur der Matrizen ähnlich. Durch größeres E steigen die erlaubten Längen der Konturabschnitte und haben im Bereich von Ecken über größere Konturabschnittsmittelpunktindexbereiche eine Ecke in sich. Daraus folgt, dass die Matrizenbilder 'unschärfer' werden. In den Abbildungen 5.7e-f verschwimmt die Struktur aus den Abbildungen 5.7a-d völlig. Das liegt wiederum daran, dass E jetzt so groß ist, dass alle verglichenen Konturabschnitte dadurch mindestens 1

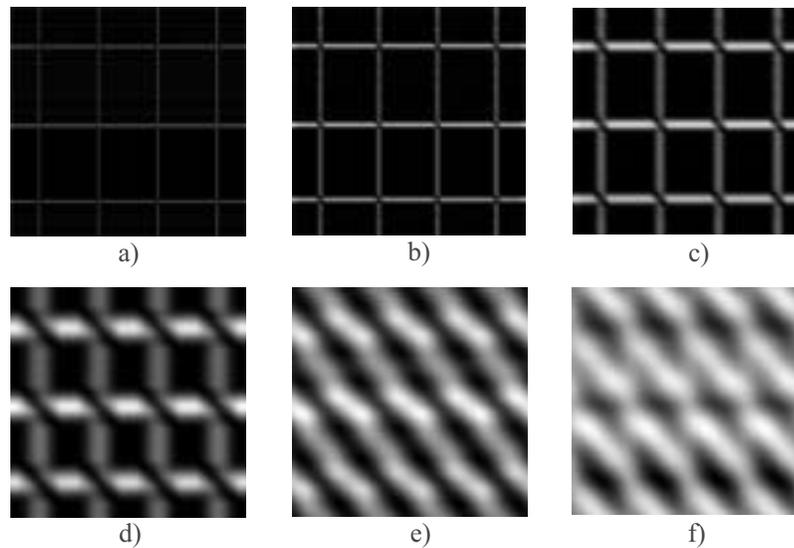


Abbildung 5.6: Variation des Parameters l_{const} bei Erzeugung der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für das Dreieck und das Viereck aus Abbildung 5.4: a) $l_{const} = 2$, b) $l_{const} = 5$, c) $l_{const} = 10$, d) $l_{const} = 25$, e) $l_{const} = 50$, f) $l_{const} = 100$.

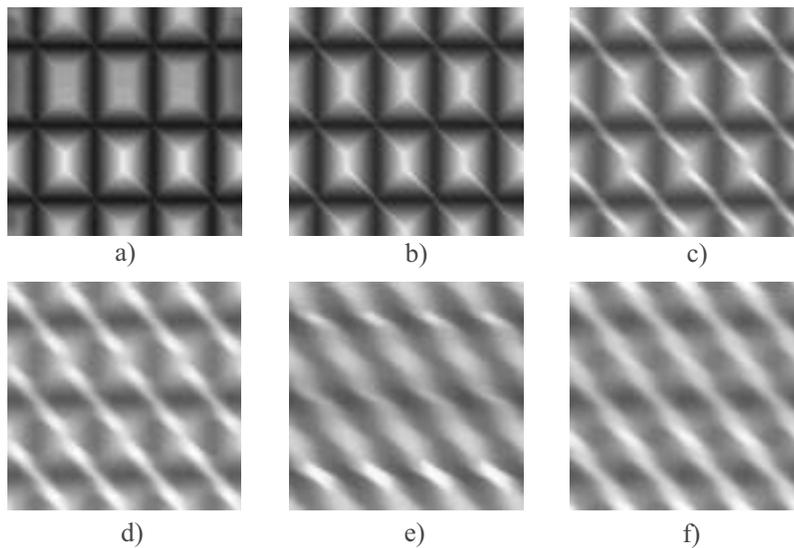


Abbildung 5.7: Variation des Parameters E bei Erzeugung der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für das Dreieck und das Viereck aus Abbildung 5.4: a) $E = 50.0$, b) $E = 100.0$, c) $E = 200.0$, d) $E = 1000.0$, e) $E = 2500.0$, f) $E = 10000.0$.

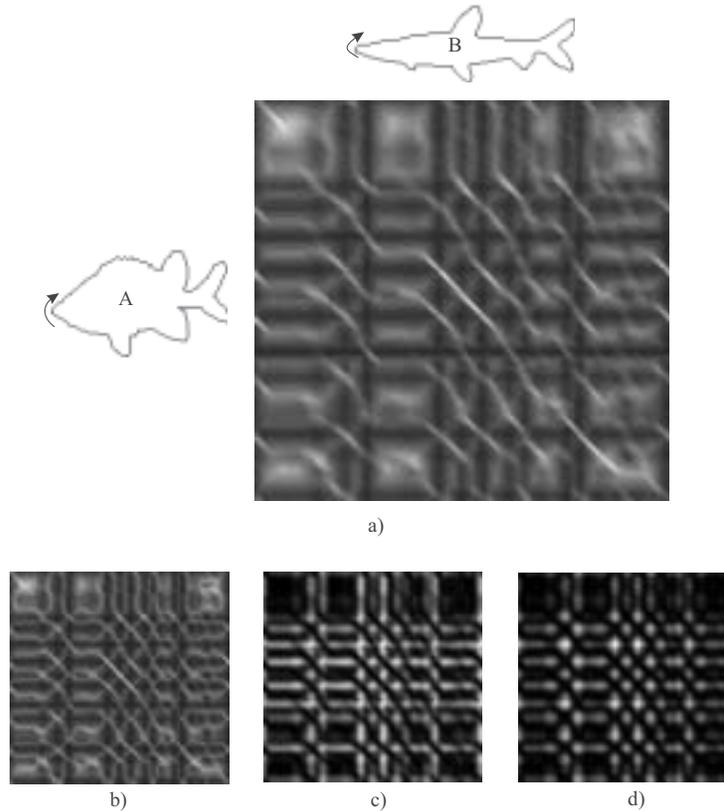


Abbildung 5.8: Die lokalen (Distanz-) Ähnlichkeitsmatrizen sind für die 2 abgebildeten Fischkonturen erzeugt worden: a) lokale Ähnlichkeitsmatrix, mittlere Orientierung, $E = 200.0$; b) lokale Ähnlichkeitsmatrix, absolute Orientierung, $E = 200.0$; c) lokale Distanzmatrix, mittlere Orientierung, $l_{const} = 10$; d) lokale Distanzmatrix, absolute Orientierung, $l_{const} = 10$.

Ecke besitzen.

Zum Abschluss dieses Abschnitts sind in Abbildung 5.8 die lokalen (Distanz-) Ähnlichkeitsmatrizen für 2 komplexere Konturen - 2 Fische - abgebildet. Die beiden Konturen wurden, wie auch das Dreieck und das Quadrat, zunächst auf eine Bogenlänge von 500 Pixeleinheiten skaliert und danach gleichabständig mit 2.0 Pixeleinheiten abgetastet. Das ergibt bei Fisch A 232 Konturpunkte und bei Fisch B 240 Konturpunkte. Abbildung 5.8a-b zeigen die lokalen Ähnlichkeitsmatrizen basierend auf mittlerer und absoluter Orientierung für $E = 200.0$. Abbildung 5.8c-d zeigen die lokalen Distanzmatrizen basierend auf mittlerer und absoluter Orientierung für $l_{const}=10$. Durch die vielen Krümmungen auf den Fischkonturen werden die Matrizen jetzt 'filigraner'. Eine Struktur lässt sich trotzdem

noch deutlich wahrnehmen. In den Matrizen sind in den Bereichen, die mit den starken Krümmungen (Flossen) der Fische korrespondieren, wieder dunkle bzw. helle vertikale und horizontale Streifen zu sehen.

Außerdem ist ein deutlicher Unterschied zwischen den Abbildungen 5.8c und 5.8d zu erkennen. Abbildung 5.8d wirkt wie eine zu dunkle Abbildung 5.8c. Das rührt daher, dass Abbildung 5.8d auf der absoluten Orientierung basiert, die wie in Abschnitt 5.1 gezeigt eine insgesamt höhere Distanz berechnet als die Methode basierend auf mittlerer Orientierung. Damit ist auch der Wertebereich der Matrix basierend auf absoluter Orientierung höher, was bei einer Quantisierung auf 256 Graustufen zu dieser insgesamt dunkleren Abbildung führt.

5.3 Korrespondenzlisten mit minimalen Kosten

In diesem Abschnitt werden Beispiele für das Erzeugen von Korrespondenzlisten mit minimalen Kosten zwischen den Konturpunkten von 2 verschiedenen Konturen gezeigt. Auch hier werden die Parameter l_{const} und E variiert und die Auswirkungen auf die Korrespondenzlisten vorgestellt (Abschnitt 5.3.1).

Im Abschnitt 5.3.2 werden die Auswirkungen von verschiedenen Deformationen (Eigendeformation, perspektivische Deformation, physikalische Deformation und Variabilität) auf die ansonsten selben Objekte gezeigt.

Abschnitt 5.3.3 zeigt Beispiele für wichtige Konturabschnitte zur Unterscheidung von 2 verschiedenen Konturen.

5.3.1 Auswirkung von l_{const} und E auf Korrespondenzlisten

In den Abbildungen 5.9 - 5.12 sind jeweils 2 Fischkonturen mit den zugehörigen lokalen (Distanz-)Ähnlichkeitsmatrizen und den Korrespondenzlisten für verschiedene Parametereinstellungen von l_{const} und E abgebildet.

Die Konturen der Fische wurden auf eine Bogenlänge von 500 Pixeleinheiten (PE) skaliert und danach gleichabständig mit 2.0 PE ab dem Markierungspunkt auf den Konturen rechtsläufig abgetastet. Fisch A besitzt danach 240 und Fisch B 234 Konturpunkte.

Die Kontur A wurde zur besseren Visualisierung der Korrespondenz verkleinert innerhalb der Kontur B dargestellt. Jeweils links sind die lokalen (Distanz-) Ähnlichkeitsmatrizen für verschiedene Parametereinstellungen gezeigt. In den Matrizen ist ein zusammenhängender Pfad markiert, der der Korrespondenzliste mit minimalen Kosten der beiden Konturen entspricht. Der Zeilen- und Spaltenindex von Punkten auf diesem Pfad entspricht den an dieser Stelle aufeinander abgebildeten Konturpunktindices der beiden Konturen. Jeweils rechts ist

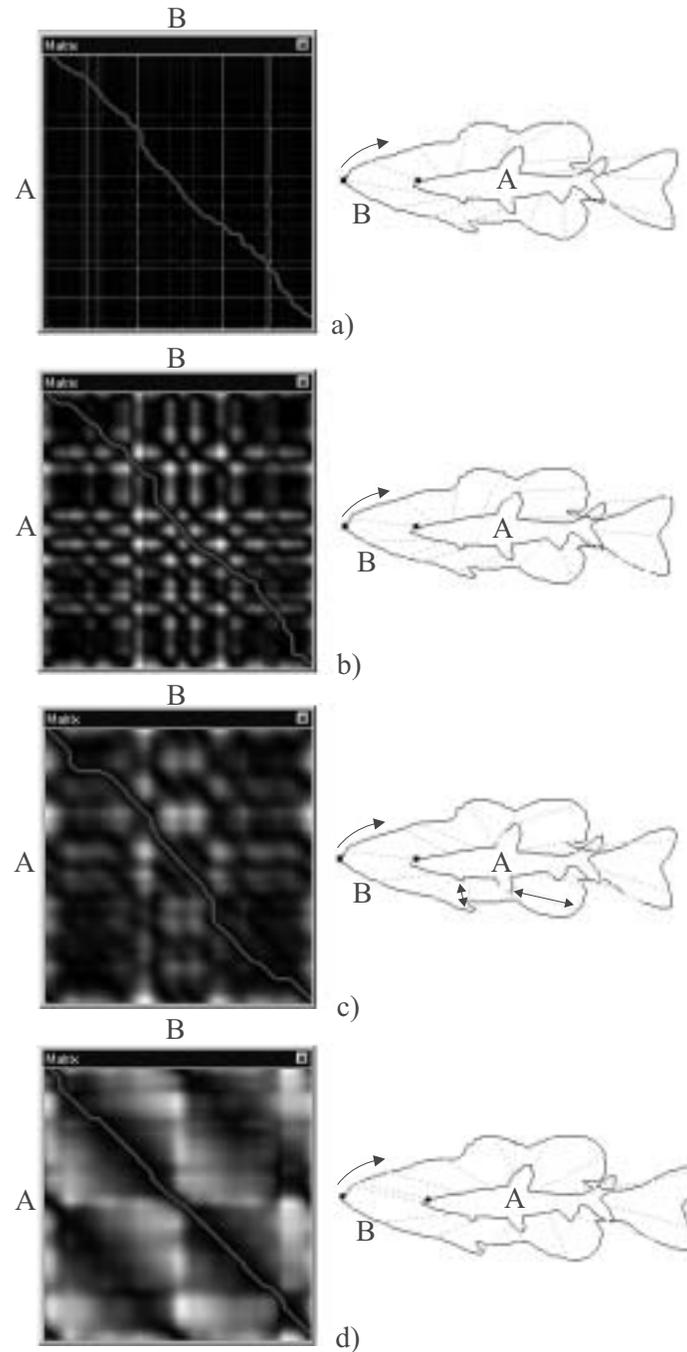


Abbildung 5.9: Auswirkungen des Parameters l_{const} bei absoluter Orientierung auf die Korrespondenzliste mit minimalen Kosten: a) $l_{const} = 1$, b) $l_{const} = 10$, c) $l_{const} = 20$, d) $l_{const} = 100$.

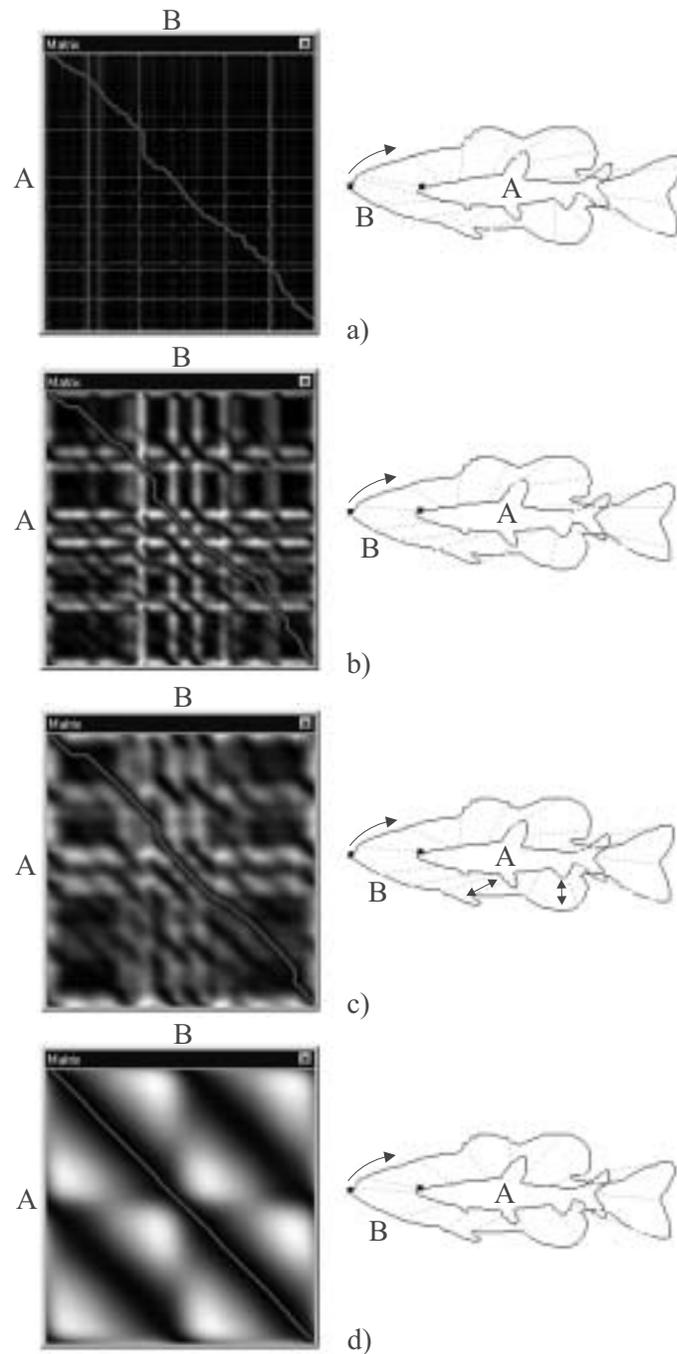


Abbildung 5.10: Auswirkungen des Parameters l_{const} bei mittlerer Orientierung auf die Korrespondenzliste mit minimalen Kosten: a) $l_{const} = 1$, b) $l_{const} = 10$, c) $l_{const} = 20$, d) $l_{const} = 100$.

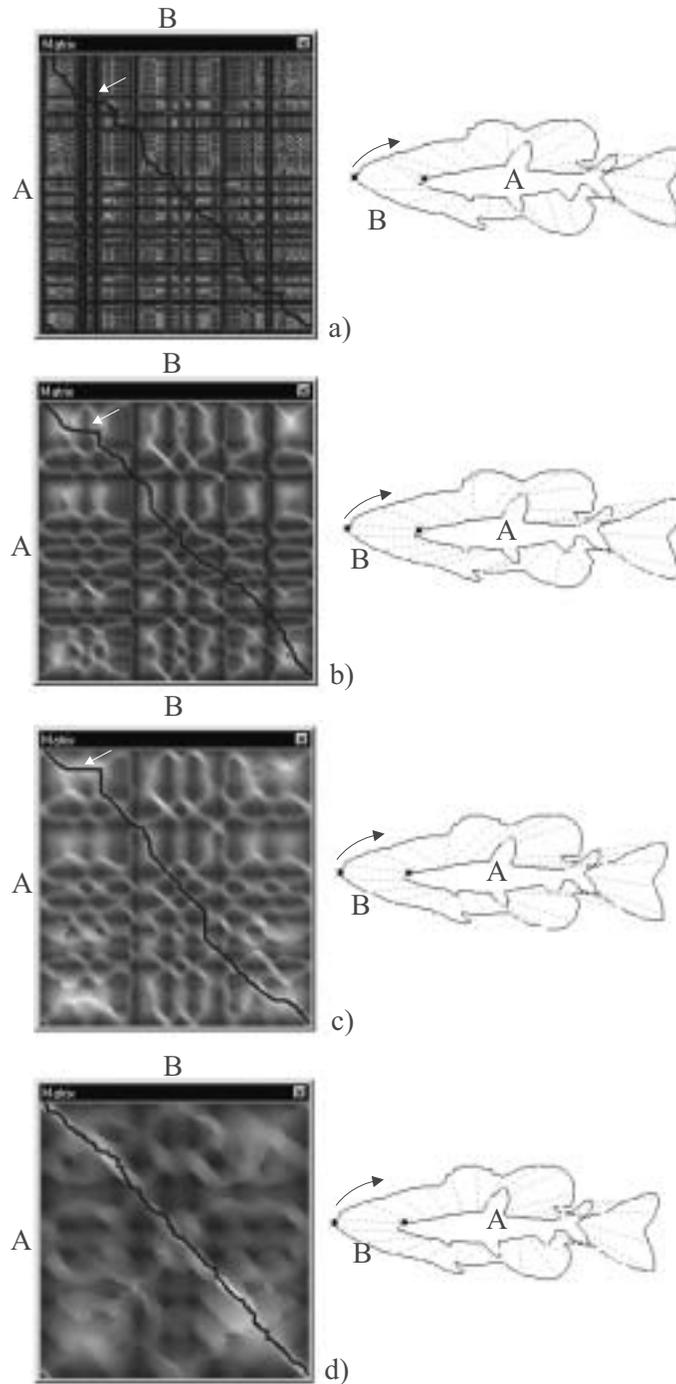


Abbildung 5.11: Auswirkungen des Parameters E bei absoluter Orientierung auf die Korrespondenzliste mit minimalen Kosten: a) $E = 1.0$, b) $E = 200.0$, c) $E = 1000.0$, d) $E = 10000.0$.

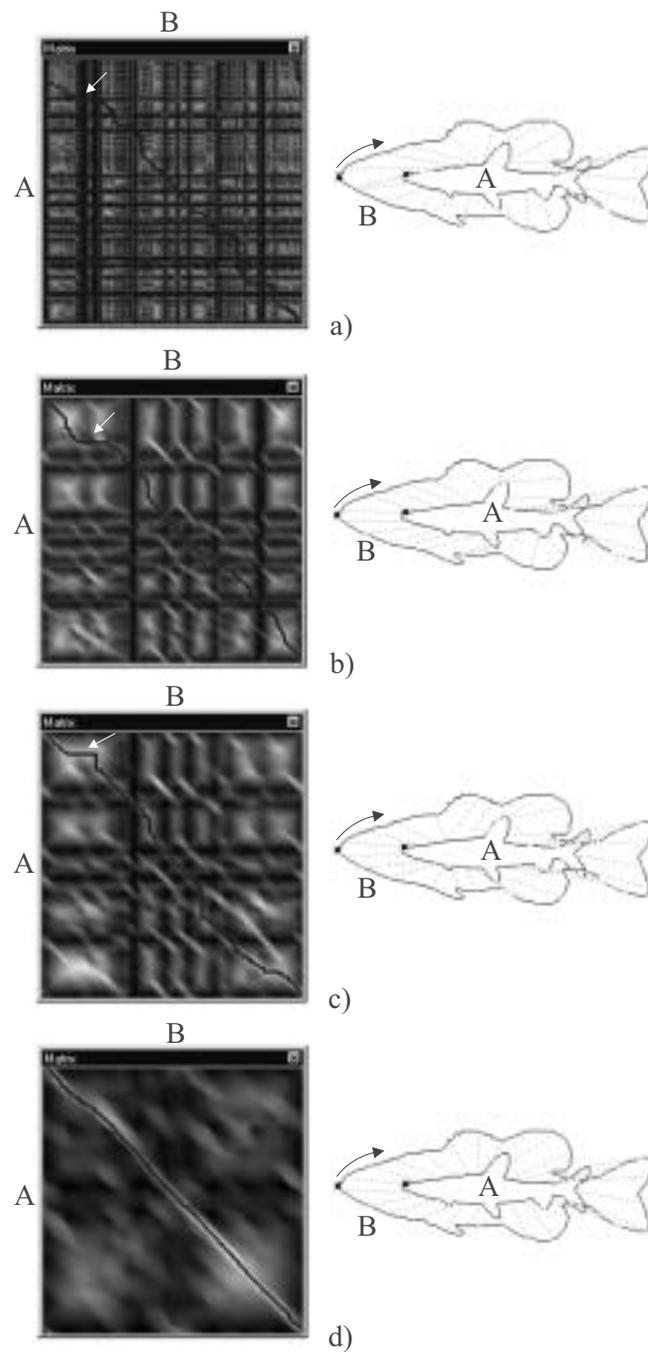


Abbildung 5.12: Auswirkungen des Parameters E bei mittlerer Orientierung auf die Korrespondenzliste mit minimalen Kosten: a) $E = 1.0$, b) $E = 200.0$, c) $E = 1000.0$, d) $E = 10000.0$.

diese Korrespondenz durch die gestrichelten Linien zwischen den Konturen für einige aufeinander abgebildete Konturpunktpaare visualisiert.

Auffällig ist, dass selbst bei sehr kleinen l_{const} bzw. sehr kleinem E die globale Abbildung der Konturen aufeinander im Großen und Ganzen funktioniert. In den Abbildungen 5.11a und 5.12a lassen sich im Konturschlussbereich schlechte lokale Anpassungen feststellen. Dies liegt aber vor allem daran, dass der Anfangs- und Endpunkt weniger als 2.0 PE voneinander entfernt liegen und damit an dieser Stelle im Verhältnis zu anderen Konturabschnitten schnell ein großer Fehler entsteht.

Der Anfangs- und Endpunkt der Konturen sind weniger als 2.0 PE voneinander entfernt, da dies durch die Art der Abtastung nicht gewährleistet werden kann: Die Abtastung der Konturen beginnt am Startpunkt. Von dort aus wird ein Kreisbogen von 2.0 PE Radius gezogen. Dort, wo der Kreisbogen das Geradenstück zwischen 2 benachbarten Konturpunkten schneidet, liegt der nächste Abtastpunkt der Kontur. Von diesem Abtastpunkt wird wieder ein Kreisbogen mit 2.0 PE Radius geschlagen usw. bis wieder kurz vor dem Endpunkt angelangt wird, der i. allg. durch diese Methode nicht wieder genau getroffen wird. Bis auf diesen Randeffekt sind die lokalen Anpassungen für die in den Abbildungen 5.9a - 5.12a dargestellten Korrespondenzlisten aber schon erstaunlich gut.

Die gefundenen Pfade für die in den Abbildungen 5.9b - 5.12b abgebildeten Fälle sind sich sehr ähnlich. Die Effekte am Konturschluss sind aufgrund genügend großer l_{const} bzw. E behoben. Die Rückenflossen, die Schwanzflossen und die hinteren und vorderen Brustflossen der beiden Fische werden in allen Fällen aufeinander abgebildet.

Interessant ist der Pfad in Abbildung 5.10c. Die zugehörige Methode behält die Zuordnung der Brustflossen aus Abbildung 5.10b bei, wohingegen der Pfad in Abbildung 5.9c dies bzgl. dem Pfad in Abbildung 5.9b nicht tut. In diesem Fall wird die kleine Ausbuchtung des Fisches A auf die vordere Brustflosse des Fisches B und die vordere Brustflosse des Fisches A auf die hintere Brustflosse des Fisches B abgebildet. Diese Entscheidung wird auch von den Pfaden in Abbildung 5.11c und 5.12c geteilt. Die Konturabschnitte für $l_{const} = 20$ beinhalten im Brustbereich entweder die kleine Ausbuchtung und die vordere Brustflosse oder die vordere und hintere Brustflosse des Fisches A bzw. die beiden Brustflossen des Fisches B, was in diesem Fall zu den Unterschieden der lokalen Distanzberechnung, basierend auf mittlerer Orientierung und absoluter Orientierung, führt.

Welche Anpassung im Brustbereich der Fische die bessere ist, ist sicherlich Geschmackssache, der Autor schließt sich aber intuitiv der Mehrheit der Verfahren, dargestellt in den Abbildungen 5.9c, 5.11c und 5.12c, an.

Die Abbildungen 5.9d - 5.12d zeigen den Grenzfall, dass l_{const} bzw. E jetzt so groß sind, dass fast die gesamten Konturen in den dann nicht mehr ganz so lokalen Konturabschnittsvergleich eingehen. Daraus folgt, dass wirklich lokale

Eigenschaften der Konturen kaum noch herauskommen. Das sieht man an den jetzt ziemlich geraden Pfaden. Gut sieht man diesen Effekt im Bereich der Brustflossen, die nicht mehr anderen Brustflossen zugeordnet werden.

Allgemein lässt sich feststellen, dass je größer l_{const} und E , desto gerader werden die Pfade und desto weniger werden lokale Konturabschnitte in der Anpassung beider Konturen aufeinander berücksichtigt.

Unterschiede zwischen den Pfaden in den lokalen Ähnlichkeitsmatrizen und den lokalen Distanzmatrizen treten in den Abbildungen 5.11a-c und 5.12a-c in den mit weißen Pfeilen markierten Pfadbereichen auf. Dieser Bereich korrespondiert mit dem Konturbereich vom Maul bis zum Anfang der Rückenflosse des Fisches A sowie dem dem Konturbereich vom Maul bis zum Anfang der ersten Rückenflosse bei den lokalen Distanzmatrizen bzw. dem Maul bis zum Anfang der zweiten Rückenflosse bei den lokalen Ähnlichkeitsmatrizen des Fisches B. Dies hat zur Folge, dass die Methode basierend auf lokalen Distanzmatrizen die Rückenflosse von Fisch A auf die gesamte geteilte Rückenflosse des Fisches B abbildet und die Methode basierend auf den lokalen Ähnlichkeitsmatrizen die Rückenflosse von Fisch A ausschließlich auf den hinteren Teil der geteilten Rückenflosse von Fisch B abbildet. Auch hier lässt sich nicht sagen, welche der beiden Anpassungen die bessere ist. Für beide Anpassungen lassen sich sicherlich Argumente finden.

Zusammenfassend zu diesem Unterabschnitt lässt sich sagen, dass die Anpassungen über einen weiten Parameterbereich vernünftige Ergebnisse liefern. Die Parameter dürfen allerdings nicht zu groß werden, da ansonsten die lokalen Charakteristika der Konturen verlorengehen.

Für die Konturen skaliert auf 500 PE und einer gleichabstandigen Abtastung von 2.0 PE werden folgende Parameterbereiche vorgeschlagen:

$$\begin{aligned} l_{const} &= 10 \dots 20 \\ E &= 200.0 \dots 1000.0. \end{aligned}$$

5.3.2 Auswirkung von Deformation auf Korrespondenzlisten

Es ist wünschenswert, dass lokale ähnliche Konturabschnitte von 2 identischen Objekten, die in Grenzen deformiert werden, d.h. deren charakteristische Konturabschnitte erkennbar bleiben, auch nach der Deformation aufeinander abgebildet werden.

Aus diesem Grund wurden u.a. die folgenden Versuchsreihen für Eigendeformation (Abbildung 5.13), Perspektive (Abbildung 5.14), physikalische Deformation (Abbildung 5.15) und Variabilität (Abbildung 5.16) durchgeführt.

Für die Eigendeformation wurde eine Referenzplastikente 1 in drei andere Positionen (Konturen 2 - 4 in Abbildung 5.13) gebracht. Der Blickwinkel blieb dabei konstant und die Ente wurde nach und nach aufgerichtet.

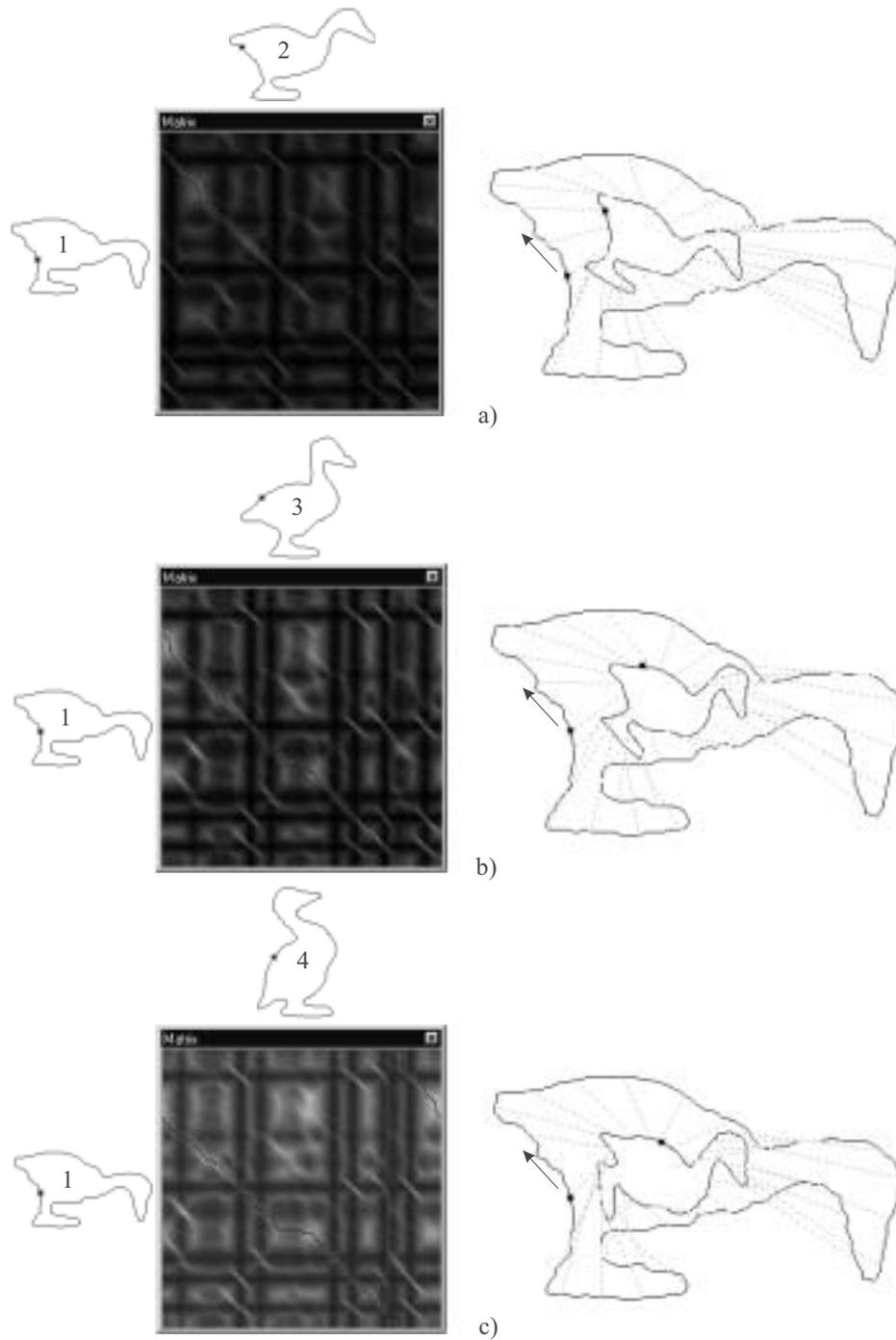


Abbildung 5.13: Versuchsreihe zur Auswirkung der Eigendeformation auf die Korrespondenzliste. Ente 1 ist die Referenzkontur mit der dieselben Enten 2 - 4 in anderer Haltung verglichen werden.

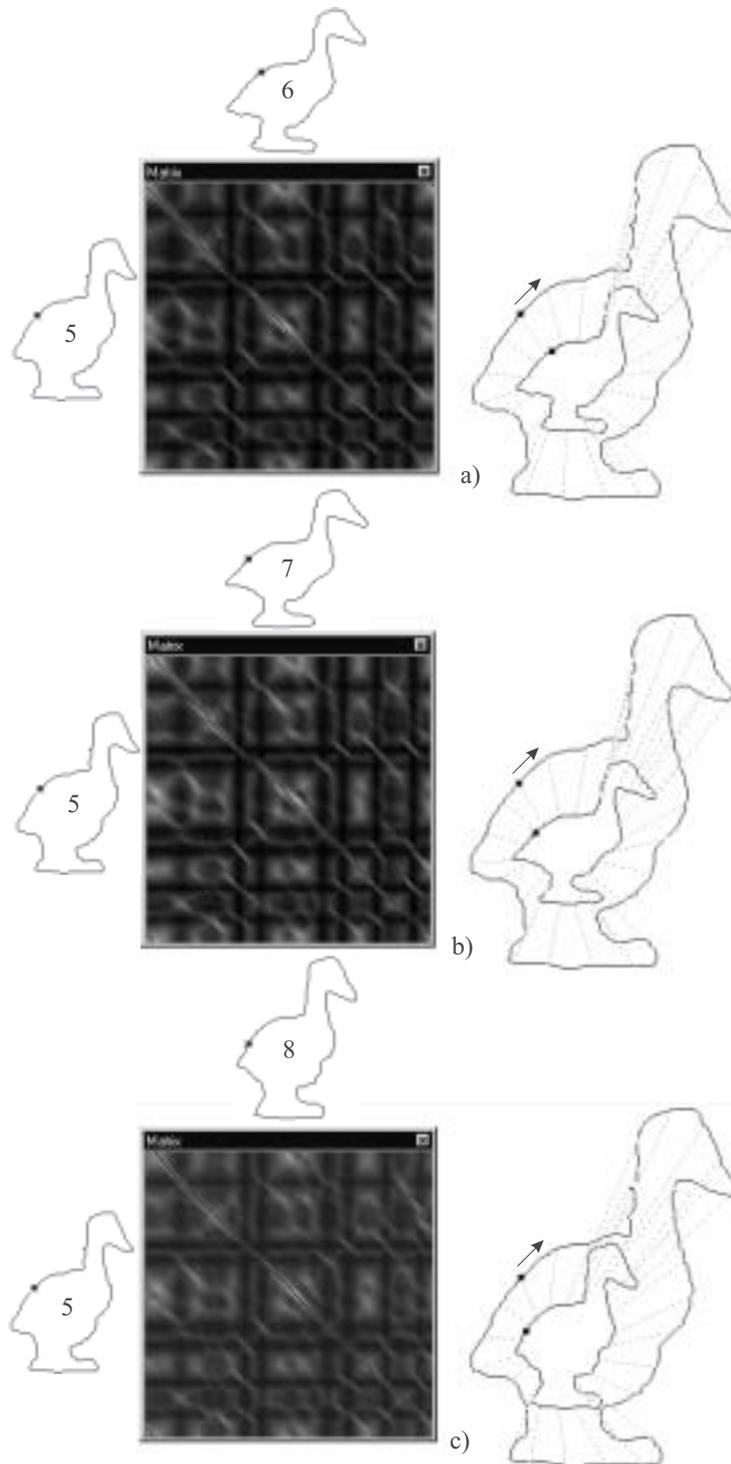


Abbildung 5.14: Versuchsreihe zur Auswirkung der Perspektive auf die Korrespondenzliste. Ente 5 ist die Referenzkontur mit der dieselben Enten 6 - 8, die aus einem anderen Blickwinkel aufgenommen wurden, verglichen werden.

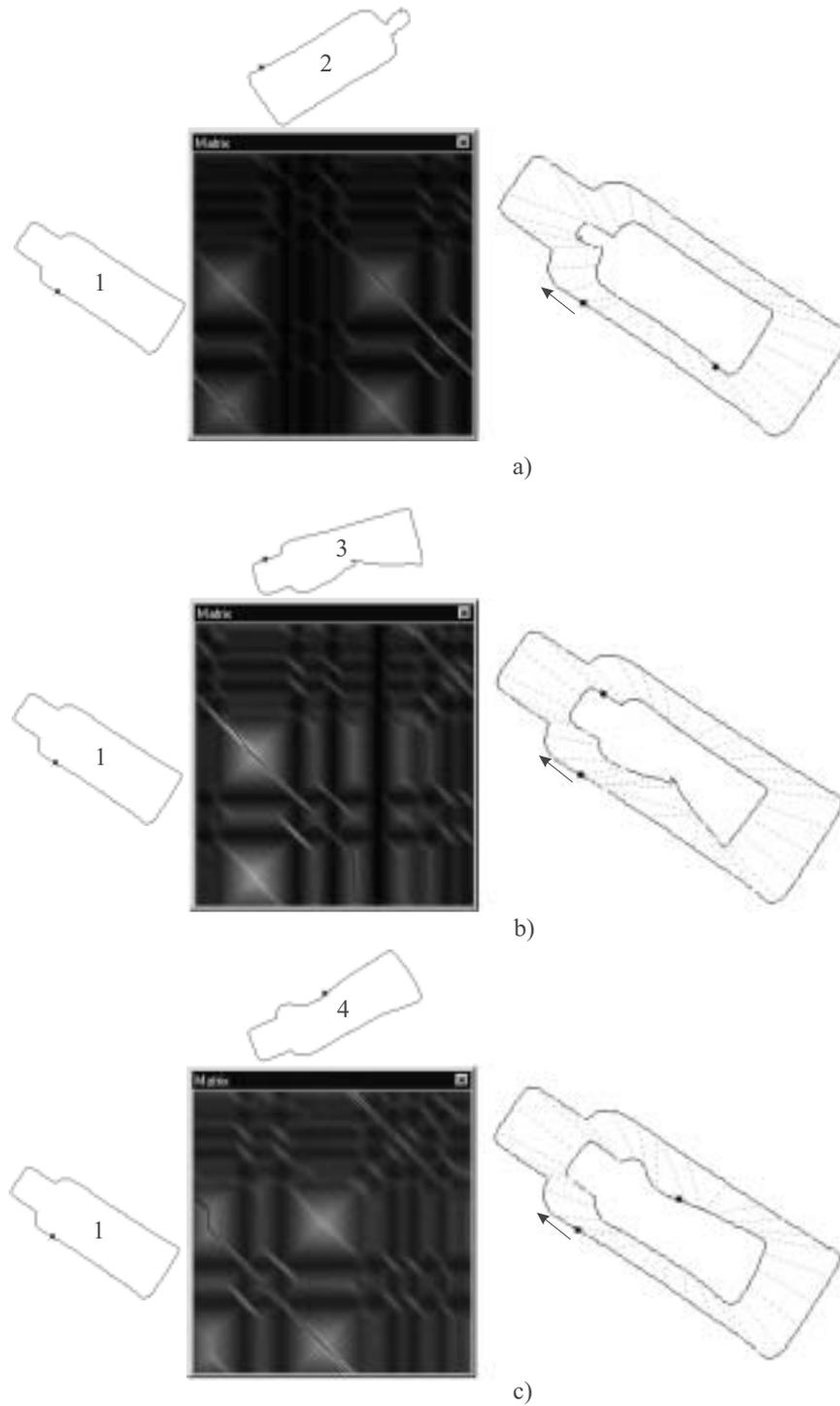


Abbildung 5.15: Versuchsreihe zur Auswirkung der physikalischen Deformation auf die Korrespondenzliste. Plastikflasche 1 ist die Referenzkontur mit der dieselben Flaschen 2 - 4 verglichen wurden. Bei Flasche 2 wurde der Deckel abgenommen. Die Flaschen 3 und 4 wurden seitlich eingedrückt.

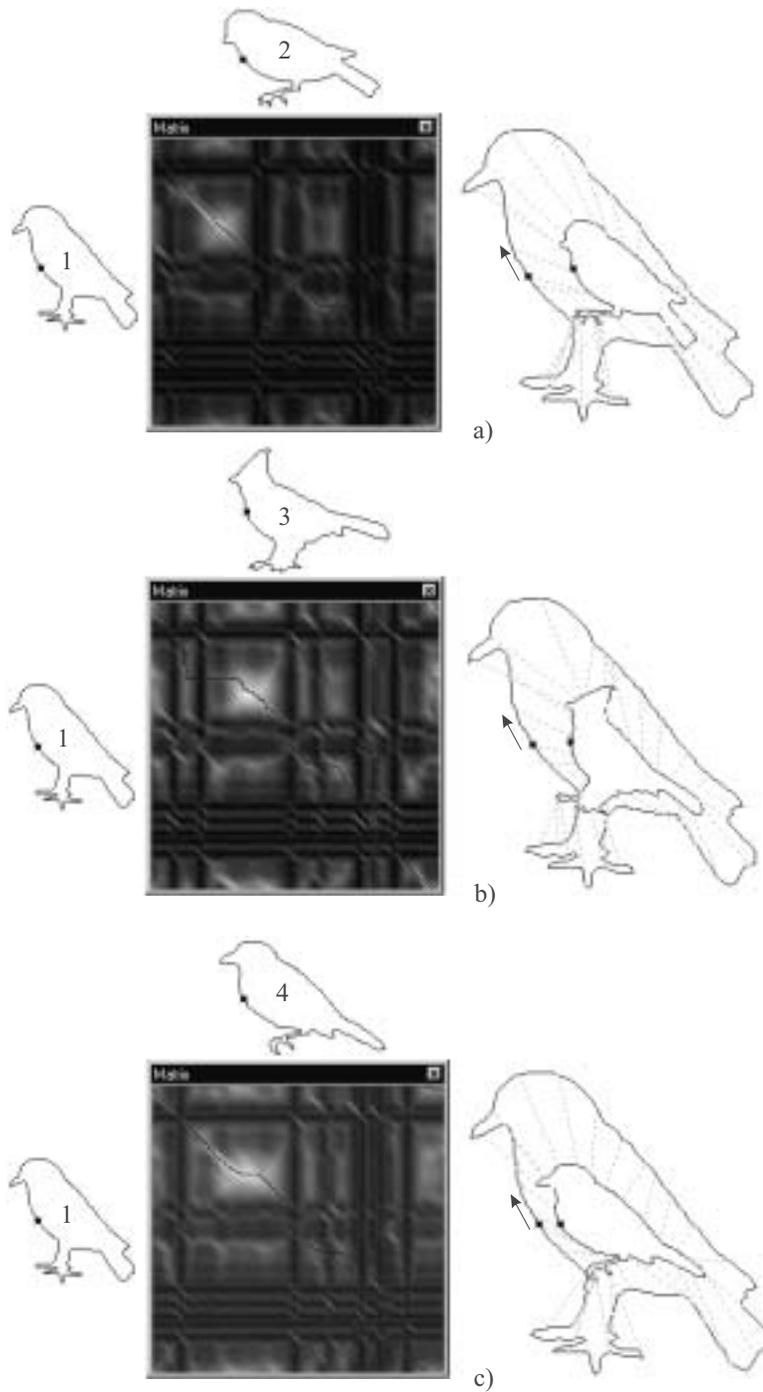


Abbildung 5.16: Versuchsreihe zur Auswirkung der Variabilität auf die Korrespondenzliste. Vogel 1 ist die willkürlich gewählte Referenzkontur mit der andere Vögel 2 - 4 verglichen wurden.

Die Referenzente 5 zur perspektivischen Deformation wurde seitlich von hinten aufgenommen. Auf gleicher Höhe wurde die Kamera rechts um die Ente herumgefahren. Beim Herumfahren wurden nacheinander die Aufnahmen gemacht, die zu den Konturen 6, 7 und 8 in Abbildung 5.14 führten. Die Kontur 8 der Ente wurde dabei von schräg vorne aufgenommen. Die Haltung der Ente war für alle 4 Aufnahmen konstant.

Die Referenzkontur 1 zur physikalischen Deformation (Abbildung 5.15) ist eine undeformierte Plastikflasche. Flasche 2 ist dieselbe Flasche mit abgenommener Kappe. Bei den Flaschen 3 und 4 handelt es sich um eingedrückte Versionen der Flasche 1.

Zur Untersuchung der Variabilität wurden 4 verschiedene Konturen von Vögeln aus der gleichen Blickrichtung und ungefähr der gleichen Haltung erzeugt. Vogel 1 ist hier die Referenzkontur.

Die Ergebnisse basieren allesamt auf einer Skalierung von 500 PE Bogenlänge, gleichabstandiger Abtastung mit 2.0 PE, lokalen Ähnlichkeitsmatrizen, mittlerer Orientierung und maximal zulässigem $E = 200.0$.

In allen Deformationsfällen lässt sich in den Abbildungen jeweils rechts nachvollziehen, dass die Korrespondenzen aller Konturabschnitte, bezogen auf die jeweilige Referenzkontur, erhalten bleiben. Die Ergebnisse der globalen Ähnlichkeiten sind in den Tabellen 5.1 - 5.4 dokumentiert. Hinter den Konturen stehen jeweils in Klammern die Konturpunktanzahlen der Konturen. Größere Werte

	Ente 2 (236)	Ente 3 (237)	Ente 4 (236)
Ente 1 (236)	14607	9577	8517

Tabelle 5.1: Tabelle mit globalen Ähnlichkeiten zwischen Referenzente 1 und den Konturen 2 - 4 aus Abbildung 5.13 für Eigendeformation.

	Ente 6 (238)	Ente 7 (235)	Ente 8 (238)
Ente 5 (238)	10841	10213	14558

Tabelle 5.2: Tabelle mit globalen Ähnlichkeiten zwischen Referenzente 5 und den Konturen 6 - 8 aus Abbildung 5.14 für perspektivische Deformation.

	Flasche 2 (234)	Flasche 3 (233)	Flasche 4 (233)
Flasche 1 (237)	19079	15781	18357

Tabelle 5.3: Tabelle mit globalen Ähnlichkeiten zwischen Referenzflasche 1 und den Konturen 2 - 4 aus Abbildung 5.15 für physikalische Deformation.

	Vogel 2 (226)	Vogel 3 (229)	Vogel 4 (232)
Vogel 1 (229)	7226	6707	7779

Tabelle 5.4: Tabelle mit globalen Ähnlichkeiten zwischen Referenzkontur 1 und den Konturen 2 - 4 aus Abbildung 5.16 für Variabilität

in den Tabellen bedeuten eine höhere Ähnlichkeit. Die Verhältnisse der globalen Ähnlichkeiten bezogen auf die jeweiligen Referenzkonturen in den einzelnen Deformationsfällen lassen sich in den dazugehörigen Abbildungen qualitativ nachvollziehen.

Zusammengefasst lässt sich in diesem Unterabschnitt feststellen, dass sich das vorgestellte Verfahren zur Ermittlung einer Korrespondenzliste mit minimalen Deformationskosten robust gegen verschiedene Deformationsarten verhält. Vor allen Dingen bleiben korrespondierende Konturabschnitte auch nach der Deformation weiterhin korrespondierend. Ähnlich robuste Ergebnisse werden basierend auf den lokalen Distanzmatrizen erreicht.

5.3.3 Wichtige Konturabschnitte

Mit Hilfe der Korrespondenzlisten lassen sich wichtige Konturabschnitte von 2 verschiedenen Konturen feststellen, die diese Konturen unterscheiden (siehe Abschnitt 4.3.4).

Abbildung 5.17 zeigt ein einfaches Beispiel für 2 künstliche Konturen. Kontur 2 ist eine Kopie von Kontur 1, bei der nur im Bereich einer Ecke eine Spitze angefügt wurde. Die Konturen wurden danach auf einen Umfang von 500 PE skaliert und gleichabständig mit 2.0 PE abgetastet. Für die Kontur 1 ergeben sich so 243 und für die Kontur 2 244 Konturpunkte. Links ist die lokale Ähnlichkeitsmatrix, basierend auf mittlerer Orientierung für $E = 200.0$, mit der markierten Korrespondenzliste abgebildet. Rechts ist die Korrespondenz der beiden Konturen mit ihren Anfangspunkten visualisiert. Die Ähnlichkeitsmatrix enthält Werte von $l^{max} = 7$ im unähnlichsten und $l^{max} = 61$ im ähnlichsten Fall. Für eine Importanzschwelle von $\tilde{\Theta} = 17$ ergibt sich dann der auf dem Pfad schraffiert markierte Bereich, der einen sehr dunklen Bereich der Matrix passiert. Die korrespondierenden Konturabschnitte sind auf den beiden Konturen fett markiert und gehören richtigerweise zur Spitze der Kontur 2 und dem korrespondierenden Abschnitt auf Kontur 1.

In Abbildung 5.18 wurde das Experiment zur Detektion von wichtigen Konturabschnitten für 2 komplexe natürliche Objekte, einer Flasche (1) und einem Fisch (2), durchgeführt. Die Konturen wurden auch in diesem Fall auf einen Umfang von 500 PE skaliert und gleichabständig mit 2.0 PE abgetastet. Es ergeben

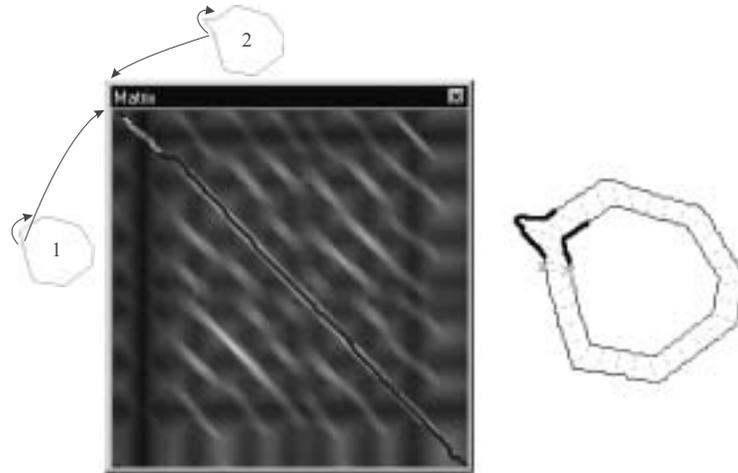


Abbildung 5.17: Wichtige Konturabschnitte von 2 Kunstkonturen. Kontur 2 ist eine Kopie der Kontur 1 mit einer Spitze an einer Ecke der Kontur 1. Links ist der wichtige Pfadabschnitt schraffiert dargestellt. Die korrespondierenden wichtigen Konturabschnitte sind auf den Konturen rechts in der Abbildung fett markiert. Links in der Abbildung sollen die Pfeile, die von den Startpunkten der Konturen auf die erste Zeile bzw. erste Spalte zeigen, andeuten, dass hier das Vergleichsergebnis der Konturabschnitte um die Startpunkte herum eingetragen ist.

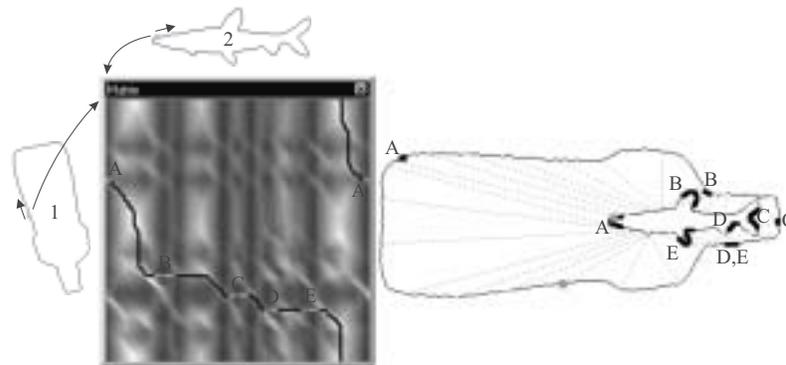


Abbildung 5.18: Wichtige Konturabschnitte zur Unterscheidung von einer Flasche und einem Fisch. Links sind die 5 wichtigen Pfadabschnitte A - E schraffiert dargestellt. Die korrespondierenden wichtigen Konturabschnitte sind auf den Konturen rechts in der Abbildung fett markiert. Links in der Abbildung sollen die Pfeile, die von den Startpunkten der Konturen auf die erste Zeile bzw. erste Spalte zeigen, andeuten, dass hier das Vergleichsergebnis der Konturabschnitte um die Startpunkte herum eingetragen ist.

sich für Kontur 1 239 und für Kontur 2 240 Konturpunkte. Die links abgebildete lokale Ähnlichkeitsmatrix, basierend auf mittlerer Orientierung für $E = 1000.0$, umfasst einen Wertebereich von $l^{max} = 8 \dots 32$. Für eine Importanzschwelle von $\tilde{\Theta} = 20$ ergeben sich die auf dem Pfad schraffiert markierten Bereiche. Diese korrespondieren wiederum mit den rechts fett markierten wichtigen Konturabschnitten der Konturen. Die Buchstaben A - E zeigen an, welche Kontur- und Pfadabschnitte zusammengehören. Die Flossen und das Maul des Fisches sind also die richtig erkannten wichtigen Konturabschnitte zur Unterscheidung dieser Kontur von der Flasche.

Bei diesen Konturen ist interessant, dass die wichtigen Konturabschnitte des Fisches durch nur jeweils genau einen Konturpunkt der Flasche ersetzt werden. Damit werden diese Abschnitte des Fisches quasi gelöscht, um den Fisch mit minimalen Kosten in die Flasche zu deformieren.

5.4 Globale Distanz- und Ähnlichkeitsmatrizen

Es ist zwar nicht das Hauptziel dieser Arbeit die Ähnlichkeit von Konturen basierend auf vollständigen Konturen festzustellen, denn dazu sollen später ausschließlich signifikante Konturabschnitte benutzt werden. Trotzdem ist es sicherlich interessant, wie sich die verschiedenen vorgestellten Verfahren bei der globalen Ähnlichkeitsanalyse verhalten. Außerdem können die globalen Ähnlichkeiten Basis einer Clusteranalyse sein, um unüberwacht die Konturen bestimmten Klassen zuzuordnen.

Für diese Experimente wurden 3 Datensätze herangezogen. Der erste Datensatz besteht aus künstlich erstellten Konturen, von denen eine Gruppe Spitzen besitzt und die andere nicht (Abbildung 5.19). Der zweite Datensatz ist komplexer und besteht aus je 10 Vögeln, 10 Flaschen und 10 Fischen (Abb. 5.20). Der dritte Datensatz stammt von Sharvit und Kimia [SCTK98], [LEM]. Er besteht aus 25 Objekten in den 6 Klassen: Fische, \rightarrow *Greebles*¹, Hände, Flugzeuge, Kaninchen und Maulschlüssel (Abbildung 5.21). Er ist damit der komplexeste der 3 Datensätze.

Die Konturen aller Datensätze wurden auf die gleiche Bogenlänge von 500 PE skaliert und danach gleichabständig mit 2.0 PE abgetastet. Es ergeben sich Konturen mit einer Konturpunktanzahl zwischen 220 und 246. Für die Konturen in allen Datensätzen wurden die globalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 10$ und $l_{const} = 20$ sowie die globalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 200.0$ und $E = 1000.0$ berechnet.

¹Greebles sind von Michael Tarr geschaffene künstliche Objekte, die sich bzgl. der Erkennung ähnlich verhalten wie Gesichter.

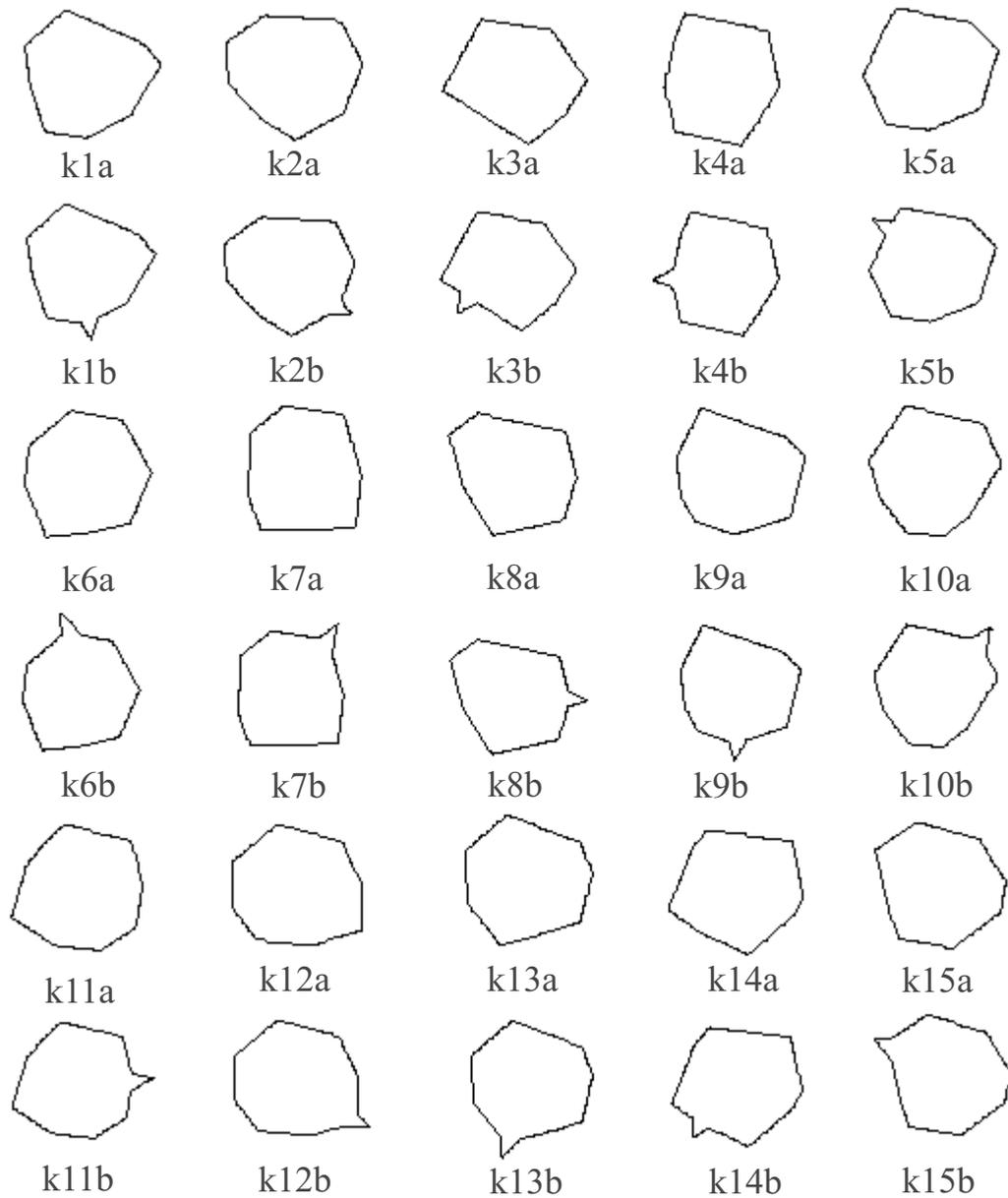


Abbildung 5.19: Datensatz mit 30 Kunstkonturen mit und ohne Spitze. Die mit b am Ende bezeichneten Konturen sind Kopien der mit a bezeichneten Konturen. Der einzige Unterschied ist eine Spitze an einer zufällig gewählten Ecke.

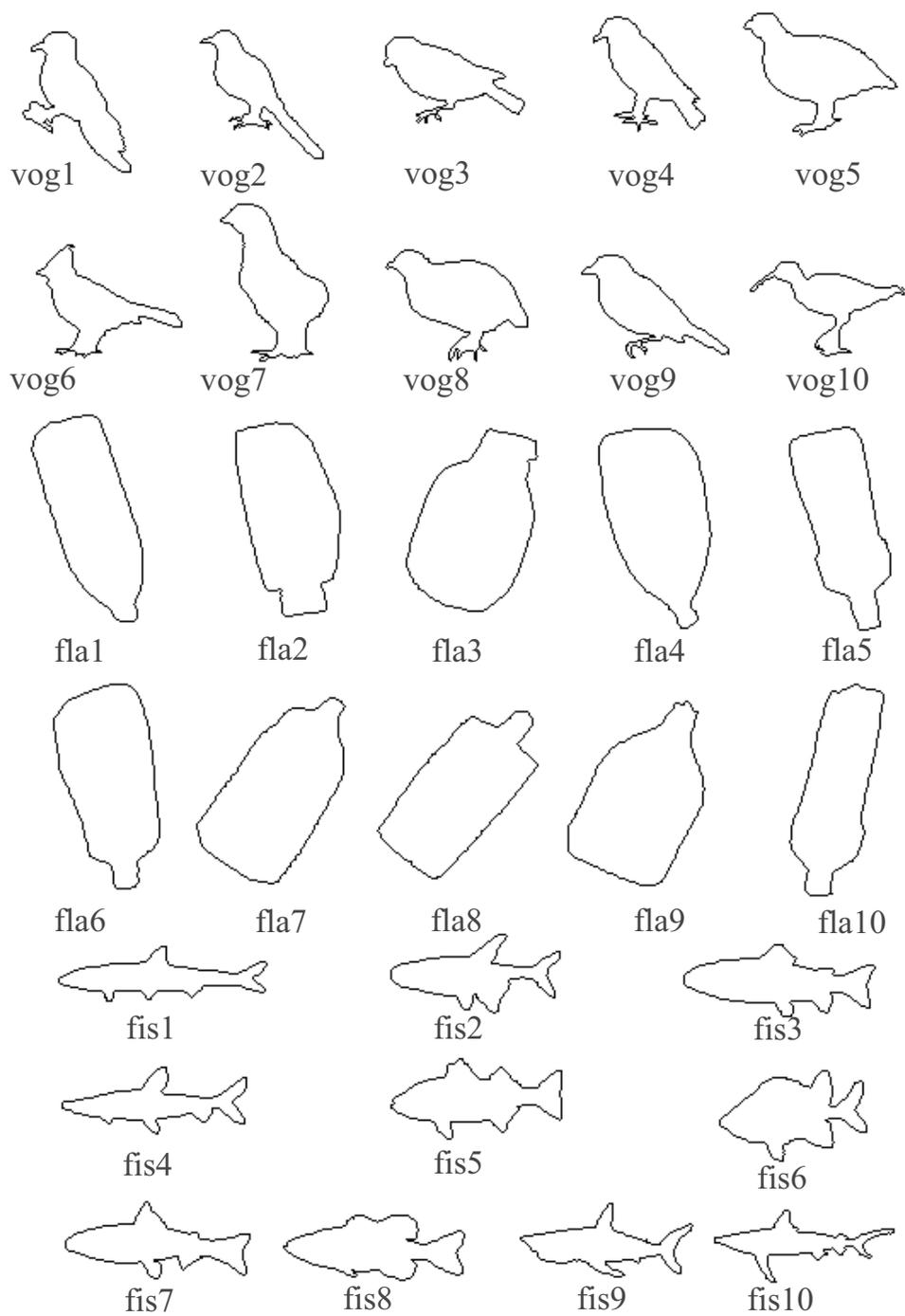


Abbildung 5.20: Datensatz mit 10 Vögeln (vog1-vog10), 10 Flaschen (fla1-fla10) und 10 Fischen (fis1-fis10).

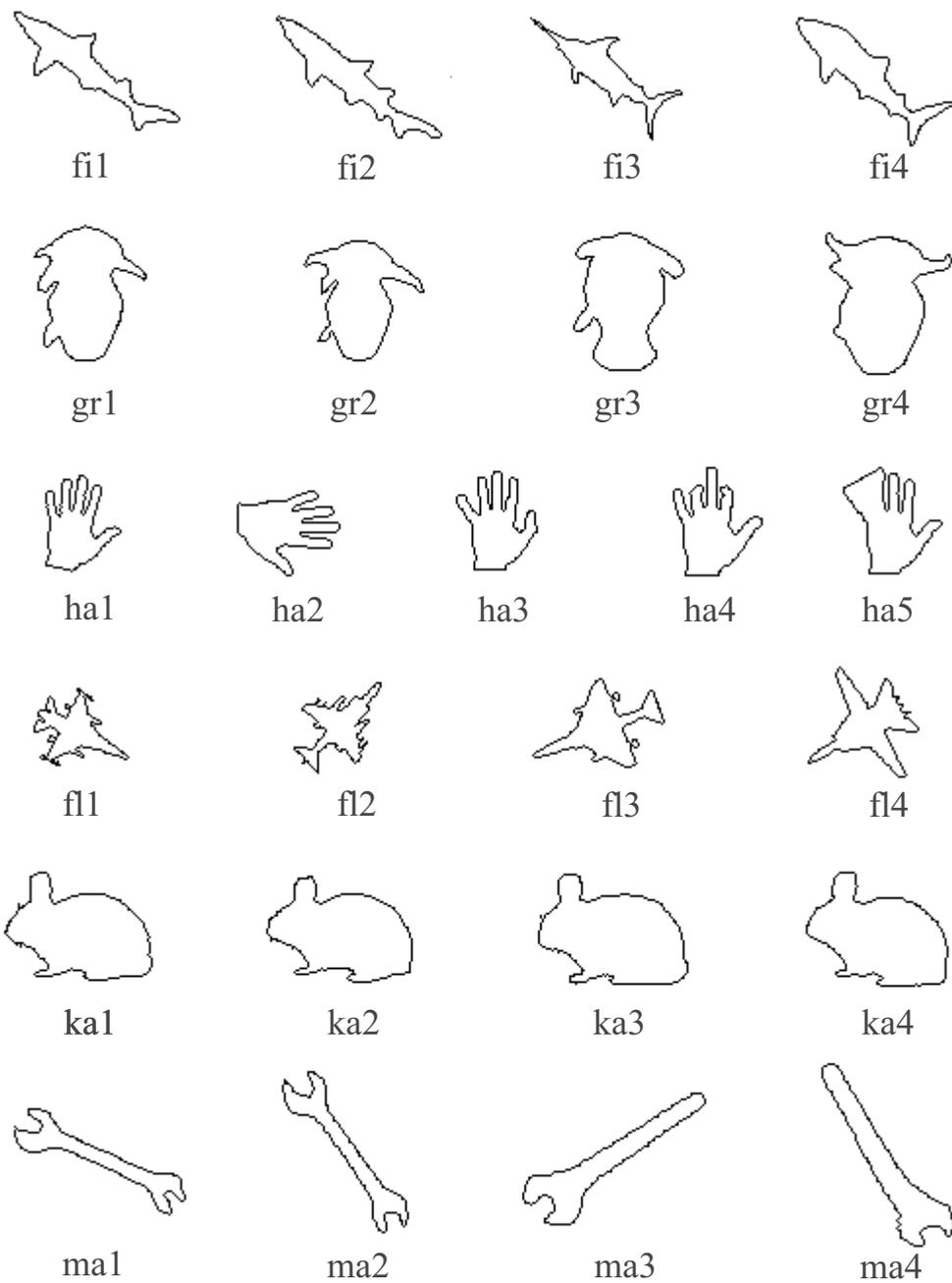


Abbildung 5.21: Datensatz von Sharvit und Kimia [SCTK98], [LEM]. Er besteht aus 6 Klassen mit Fischen (fi1-fi4), → *Greebles* (gr1-gr4), Händen (ha1-ha5), Flugzeugen (fl1-fl4), Kaninchen (ka1-ka4) und Maulschlüssel (ma1-ma4).

5.4.1 30 Kunstkonturen

Der Datensatz besteht aus 15 Konturen ohne markante Spitze und 15 Konturen mit markanter Spitze. Die Konturen ohne Spitze sind zufällig erzeugte Polygone. Die Konturen mit Spitze sind Kopien der 15 Konturen ohne Spitze, wobei jeweils willkürlich eine Ecke durch jeweils eine zufällig erzeugte Spitze ersetzt wurde. D.h. es gibt 15, bis auf die Spitzen, identische Konturpaare (k1a, k1b) bis (k15a, k15b). Ein Mensch würde die Konturen in 2 Klassen aufteilen: die Konturen mit Spitze und die Konturen ohne Spitze. Ein rein globales Verfahren zur Bestimmung von Konturähnlichkeiten würde die auseinander entstandenen Konturpaare zueinander für am ähnlichsten halten. Es ist also zu untersuchen, wie sich die in dieser Arbeit vorgestellten Verfahren zur Bestimmung der globalen Ähnlichkeit, basierend auf einer lokalen Untersuchung der Konturen, verhalten.

Da jede Klasse aus 15 Konturen (k1a - k15a und k1b - k15b) besteht, wurden die jeweils 14 nächsten Nachbarn zu jeder Kontur bestimmt. Gehört der nächste Nachbar zur Klasse der zu klassifizierenden Kontur, so wurde die Anzahl der korrekt zugeordneten nächsten Nachbarn um 1 erhöht. Im besten Fall ergibt sich für 30 zu klassifizierende Konturen für jeden der 14 nächsten Nachbarn ein Zuordnungsverhältnis von $V = \frac{30}{30} = 1.00$. Bei Falschklassifikation nimmt dieses Verhältnis entsprechend ab².

Das beste Ergebnis wurde so mit Hilfe der lokalen Distanzmatrizen, basierend auf mittlerer Orientierung für $l_{const} = 20$, erzielt. Die Tabelle 5.6 zeigt das Gesamtergebnis.

Die dazugehörige globale Distanzmatrix zeigt Tabelle 5.5. Das Ergebnis zeigt eindeutig, dass diese Methode sicher die Konturen in die 2 Klassen mit und ohne Spitze klassifiziert.

Auch das Clusteranalyseergebnis, basierend auf der Ward-Methode (siehe Anhang D), zeigt klar die Möglichkeit, die 2 Klassen mit und ohne Spitzen unüberwacht zu erkennen (siehe Dendogramm in Abbildung 5.22).

Die anderen Ergebnisse zeigen die Tabellen 5.7 - 5.9. Man sieht an den Ergebnissen, dass die Methoden basierend auf lokalen Distanzmatrizen zu ähnlich guten Klassifikationsergebnissen führen. Die Methode basierend auf lokalen Ähnlichkeitsmatrizen hat Schwierigkeiten die Konturen mit Spitze und ohne Spitze auseinanderzuhalten. Bei einem maximal zulässigen Fehler von $E = 200.0$ ergibt sich sogar, dass keine einzige Kontur als 1. nächster Nachbar klassifiziert wurde. Bei näherer Untersuchung zeigt sich, dass die ersten nächsten Nachbarn der Konturen jeweils die Kopien mit bzw. ohne Spitze sind. Da diese Konturen bis auf die Spitzen identisch sind, erzeugen sie natürlich große l^{max} bis zur Überschreitung

²Da die später vorgestellten Vergleichsergebnisse zum Datensatz von Sharvit und Kimia in der Literatur in dieser Art dargestellt werden, wurde aus Einheitlichkeitsgründen diese Darstellung der Ergebnisse für alle Datensätze übernommen.

	k1a	k2a	k3a	k4a	k5a	k6a	k7a	k8a	k9a	k10a	k11a	k12a	k13a	k14a	k15a	k1b	k2b	k3b	k4b	k5b	k6b	k7b	k8b	k9b	k10b	k11b	k12b	k13b	k14b	k15b
k1a	0	308	345	309	170	193	299	321	330	272	307	236	217	287	265	1107	952	1364	1178	1150	1335	1378	1382	1087	962	1605	760	861	1004	722
k2a	308	0	461	396	234	248	358	319	320	295	271	226	258	360	278	1075	781	1383	1413	1148	1430	1494	1374	1129	1166	1649	817	1058	907	883
k3a	345	461	0	154	357	415	298	346	380	471	429	453	391	217	391	798	727	1002	1029	1104	1089	1053	1051	969	735	1348	713	747	836	615
k4a	309	396	154	0	323	380	287	268	390	437	381	334	292	207	377	890	806	1186	1142	1132	1095	1185	1121	958	844	1465	701	877	762	693
k5a	170	234	357	323	0	197	285	306	208	262	195	223	242	267	248	1033	831	1385	1176	994	1315	1254	1391	1131	1042	1565	816	919	919	787
k6a	193	248	415	380	197	0	281	332	297	338	246	198	183	324	175	1057	762	1241	1197	1044	1147	1205	1266	1011	899	1400	612	700	847	570
k7a	299	358	298	287	285	281	0	268	353	389	258	265	294	296	311	989	682	1309	1077	1121	1209	1028	1302	1128	980	1585	679	979	918	734
k8a	321	319	346	268	306	332	268	0	347	307	288	382	301	252	293	1019	805	1135	1176	1138	1235	1159	1278	1072	1014	1539	830	871	868	815
k9a	330	320	380	390	208	297	353	347	0	255	163	312	342	275	280	1067	736	1395	999	872	1156	1021	1255	1068	992	1379	600	755	897	707
k10a	272	295	471	437	262	338	389	307	255	0	278	277	268	390	210	1187	822	1342	1325	1092	1382	1348	1250	1129	975	1538	819	895	925	766
k11a	307	271	429	381	195	246	258	288	163	278	0	272	268	314	312	1035	815	1427	1173	1012	1183	1100	1292	1110	1053	1440	678	806	956	773
k12a	236	226	453	334	223	198	265	382	312	277	272	0	234	331	287	1161	856	1461	1331	1117	1336	1397	1378	1191	1045	1614	510	919	1000	753
k13a	217	258	391	292	242	183	294	301	342	268	268	234	0	243	224	1070	796	1269	1262	1177	1285	1337	1309	1043	986	1523	796	697	838	703
k14a	287	360	217	207	267	324	296	252	275	390	314	331	243	0	359	883	825	1212	1092	1168	1166	1156	1071	884	954	1387	759	817	763	778
k15a	265	278	391	377	248	175	311	293	280	210	312	287	224	359	0	983	753	1095	1199	1088	1249	1389	1247	1122	771	1587	759	829	823	465
k1b	1107	1075	798	890	1033	1057	989	1019	1067	1187	1035	1161	1070	883	983	0	367	549	393	392	259	670	495	330	343	494	599	447	572	422
k2b	952	781	727	806	831	762	682	805	736	822	815	856	796	825	753	367	0	678	408	447	422	529	540	523	394	533	434	488	601	390
k3b	1364	1383	1002	1186	1385	1241	1309	1135	1395	1342	1427	1461	1269	1212	1095	549	678	0	585	486	680	1150	444	638	694	714	1056	838	319	730
k4b	1178	1413	1029	1142	1176	1197	1077	1176	999	1325	1173	1331	1262	1092	1199	393	408	585	0	569	460	466	357	526	592	476	720	666	700	615
k5b	1150	1148	1104	1132	994	1044	1121	1138	872	1092	1012	1117	1177	1168	1088	392	447	486	569	0	573	704	554	476	531	518	555	632	446	600
k6b	1335	1430	1089	1095	1315	1147	1209	1235	1156	1382	1183	1336	1285	1166	1249	259	422	680	460	573	0	423	593	517	369	460	479	399	823	480
k7b	1378	1494	1053	1185	1254	1205	1028	1159	1021	1348	1100	1397	1337	1156	1389	670	529	1150	466	704	423	0	615	481	648	529	559	440	1059	615
k8b	1382	1374	1051	1121	1391	1266	1302	1278	1255	1250	1292	1378	1309	1071	1247	495	540	444	357	554	593	615	0	345	529	347	904	624	454	648
k9b	1087	1129	969	958	1131	1011	1128	1072	1068	1129	1110	1191	1043	884	1122	330	523	638	526	476	517	481	345	0	512	255	635	369	608	519
k10b	962	1166	735	844	1042	899	980	1014	992	975	1053	1045	986	954	771	343	394	694	592	531	369	648	529	512	0	544	419	328	696	188
k11b	1605	1649	1348	1465	1565	1400	1585	1539	1379	1538	1440	1614	1523	1387	1587	494	533	714	476	518	460	529	347	255	544	0	711	563	774	727
k12b	760	817	713	701	816	612	679	830	600	819	678	510	796	759	759	599	434	1056	720	555	479	559	904	635	419	711	0	339	987	333
k13b	861	1058	747	877	919	700	979	871	755	895	806	919	697	817	829	447	488	838	666	632	399	440	624	369	328	563	339	0	769	223
k14b	1004	907	836	762	919	847	918	868	897	925	956	1000	838	763	823	572	601	319	700	446	823	1059	454	608	696	774	987	769	0	726
k15b	722	883	615	693	787	570	734	815	707	766	773	753	703	778	465	422	390	730	615	600	480	615	648	519	188	727	333	223	726	0

Tabelle 5.5: Globale Distanzmatrix mit den globalen Distanzen der künstlichen Konturen in Abbildung 5.19 mithilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$. In jeder Spalte sind die 14 kleinsten globalen Distanzen fett markiert. Jeder Eintrag in der Matrix muss mit 100 multipliziert werden, um die Originaldistanzen zu erhalten.

i. NN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
V	1.00	1.00	1.00	1.00	1.00	0.93	1.00	1.00	0.97	0.93	0.93	1.00	0.87	0.87

Tabelle 5.6: Tabelle mit Klassifikationsergebnissen für die künstlichen Konturen in Abbildung 5.19 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

i. NN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
V	1.00	1.00	0.97	1.00	1.00	1.00	1.00	1.00	0.97	1.00	0.97	0.97	0.93	0.90

Tabelle 5.7: Tabelle mit Klassifikationsergebnissen für die künstlichen Konturen in Abbildung 5.19 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 10$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

i. NN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
V	0.53	0.67	0.70	0.67	0.53	0.47	0.57	0.63	0.57	0.70	0.60	0.57	0.60	0.63

Tabelle 5.8: Tabelle mit Klassifikationsergebnissen für die künstlichen Konturen in Abbildung 5.19 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 1000.0$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

i. NN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
V	0.00	0.77	0.60	0.67	0.60	0.67	0.67	0.53	0.67	0.53	0.60	0.50	0.30	0.63

Tabelle 5.9: Tabelle mit Klassifikationsergebnissen für die künstlichen Konturen in Abbildung 5.19 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 200.0$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

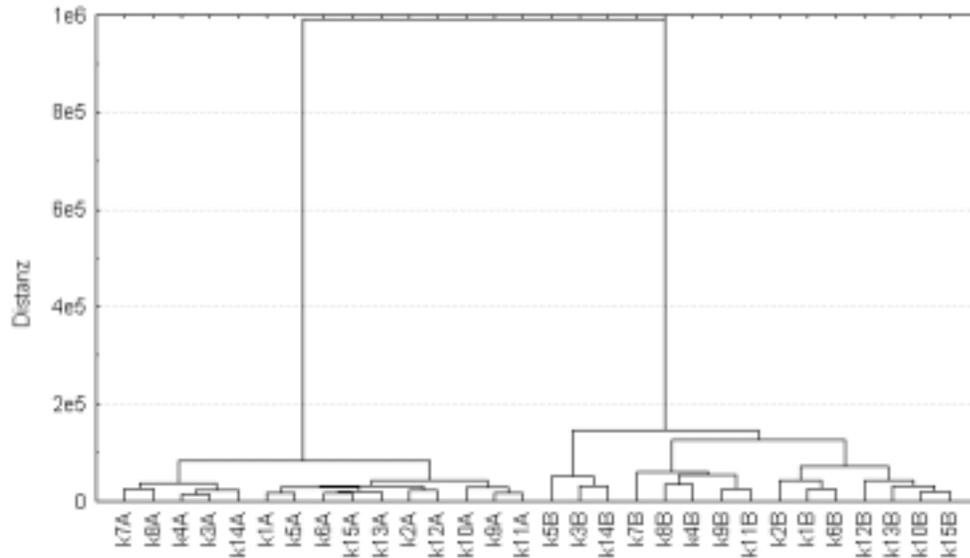


Abbildung 5.22: Dendrogramm der Clusteranalyse mit Ward-Methode für die künstlichen Konturen aus Abbildung 5.19 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$.

der maximal zulässigen Distanz E . Nur im Bereich der Spitze sind die l^{max} klein. Der relativ kurze Bereich der Spitze reicht dann nicht aus, die Summe der l^{max} so niedrig zu halten, dass die Kopien unähnlicher zueinander sind als die Konturen mit Spitzen. Damit unterscheidet sich diese Methode bei fast identischen Konturen nicht großartig von einer rein globalen Vergleichsmethode.

Dies ist bei den Methoden basierend auf lokalen Distanzmatrizen anders. Hier werden immer lokale Konturabschnitte konstanter Länge l_{const} miteinander verglichen. Zum größten Teil bestehen die Konturen aus geraden Stücken, so dass hier bei dem Vergleich von 2 Konturabschnitten in den meisten Fällen näherungsweise eine Distanz von 0.0 berechnet wird. Hier wird die globale Distanz zwischen den Kopien fast ausschließlich durch die Spitze bestimmt. Und diese ist ausreichend, um die Konturen mit und ohne Spitze zu unterscheiden.

5.4.2 Vögel, Flaschen und Fische

Die in Abbildung 5.20 abgebildeten Konturen sind die Konturen von je 10 Vögeln, Flaschen und Fischen. Da jede Klasse aus 10 Konturen besteht, wurden die jeweils 9 nächsten Nachbarn zu jeder Kontur bestimmt. Gehört der nächste Nachbar zur Klasse der zu klassifizierenden Kontur, so wird die Anzahl der korrekt

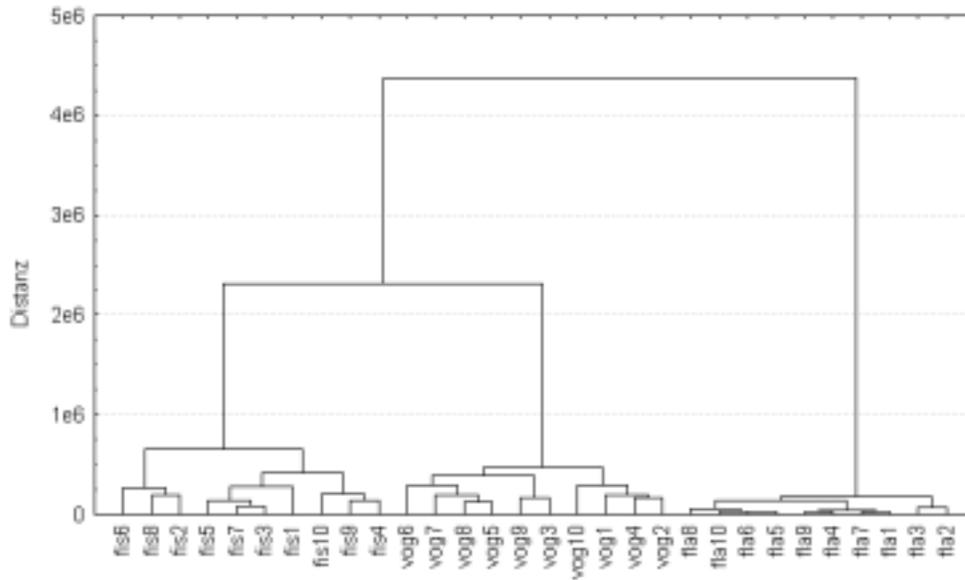


Abbildung 5.23: Dendrogramm der Clusteranalyse mit Ward-Methode für die Konturen von den Vögeln, Flaschen und Fischen aus Abbildung 5.20 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$.

zugeordneten nächsten Nachbarn um 1 erhöht. Im besten Fall ergibt sich für 30 zu klassifizierende Konturen für jeden der 9 nächsten Nachbarn ein Zuordnungsverhältnis von $V = \frac{30}{30} = 1.00$. Die Ergebnisse der einzelnen Methoden sind in den Tabellen 5.10 - 5.13 dokumentiert. Das beste Ergebnis liefert auch hier die Methode mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$. Die Methoden basierend auf den lokalen Ähnlichkeitsmatrizen bringen jetzt zwar stark verbesserte Ergebnisse im Vergleich zu den Kunstkonturen aus Abschnitt 5.4.1, doch kommen sie an die Ergebnisse basierend auf lokalen Distanzmatrizen nicht heran.

Die globale Distanzmatrix für $l_{const} = 20$ zeigt Tabelle 5.14. Das Ergebnis der Clusteranalyse mit Ward-Methode dieser Matrix ist in Form eines Dendrogramms in Abbildung 5.23 visualisiert. Auch in diesem Fall lassen sich also die 3 Klassen Vögel, Flaschen und Fische exakt voneinander trennen und unüberwacht in Klassen einteilen.

5.4.3 Datensatz von Sharvit und Kimia

Der Datensatz von Sharvit und Kimia besteht aus 25 Konturen in 6 Klassen (Abbildung 5.21). Es gibt jeweils 4 Konturen in den Klassen Fische, Greebles, Flug-

i. NN	1	2	3	4	5	6	7	8	9
V	1.00	1.00	1.00	1.00	1.00	0.93	0.90	0.83	0.67

Tabelle 5.10: Tabelle mit Klassifikationsergebnissen für die Konturen von den Vögeln, Flaschen und Fischen in Abbildung 5.20 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

i. NN	1	2	3	4	5	6	7	8	9
V	1.00	1.00	1.00	0.97	0.93	0.97	0.93	0.77	0.70

Tabelle 5.11: Tabelle mit Klassifikationsergebnissen für die Konturen von den Vögeln, Flaschen und Fischen in Abbildung 5.20 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 10$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

i. NN	1	2	3	4	5	6	7	8	9
V	1.00	0.83	0.83	0.80	0.77	0.70	0.57	0.53	0.43

Tabelle 5.12: Tabelle mit Klassifikationsergebnissen für die Konturen von den Vögeln, Flaschen und Fischen in Abbildung 5.20 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 1000.0$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

i. NN	1	2	3	4	5	6	7	8	9
V	0.83	0.73	0.63	0.47	0.57	0.43	0.43	0.53	0.57

Tabelle 5.13: Tabelle mit Klassifikationsergebnissen für die Konturen von den Vögeln, Flaschen und Fischen in Abbildung 5.20 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 200.0$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

	vog1	vog2	vog3	vog4	vog5	vog6	vog7	vog8	vog9	vog10	fla1	fla2	fla3	fla4	fla5	fla6	fla7	fla8	fla9	fla10	fis1	fis2	fis3	fis4	fis5	fis6	fis7	fis8	fis9	fis10
vog1	0	185	255	202	215	252	374	256	220	209	599	466	632	533	488	504	568	418	524	494	410	527	326	428	421	496	436	445	532	547
vog2	185	0	347	174	204	309	343	240	220	268	764	638	726	702	672	693	690	609	665	662	515	537	461	530	531	514	498	532	592	544
vog3	255	347	0	308	255	277	340	270	171	402	711	615	651	646	590	615	670	503	641	612	505	513	528	526	562	487	483	484	540	596
vog4	202	174	308	0	265	292	302	194	283	299	704	657	699	612	570	621	656	525	657	576	467	501	451	519	515	494	439	491	543	599
vog5	215	204	255	265	0	165	163	122	204	219	542	470	464	478	421	448	475	358	431	456	387	393	355	399	370	495	412	363	513	517
vog6	252	309	277	292	165	0	314	253	240	303	588	563	571	572	480	522	545	427	528	472	351	360	355	331	368	535	384	384	447	421
vog7	374	343	340	302	163	314	0	195	316	349	425	367	361	376	328	346	382	330	349	339	365	482	357	423	358	545	352	401	551	518
vog8	256	240	270	194	122	253	195	0	203	279	553	505	506	496	443	460	500	387	477	466	396	404	419	402	443	461	397	352	510	574
vog9	220	220	171	283	204	240	316	203	0	265	715	665	675	650	592	613	675	503	639	617	524	510	491	527	550	613	531	539	611	547
vog10	209	268	402	299	219	303	349	279	265	0	706	650	741	667	587	653	675	528	611	629	392	464	388	347	398	521	393	450	430	419
fla1	599	764	711	704	542	588	425	553	715	706	0	91	76	33	62	41	23	120	41	44	349	752	409	616	480	785	490	474	812	772
fla2	466	638	615	657	470	563	367	505	665	650	91	0	74	107	102	81	92	116	109	106	359	579	256	500	366	610	368	342	634	685
fla3	632	726	651	699	464	571	361	506	675	741	76	74	0	93	117	88	64	146	70	122	420	684	363	591	442	721	453	440	724	841
fla4	533	702	646	612	478	572	376	496	650	667	33	107	93	0	56	43	26	107	27	59	329	681	379	591	480	713	477	469	769	760
fla5	488	672	590	570	421	480	328	443	592	587	62	102	117	56	0	27	51	39	41	30	291	548	302	478	348	628	377	355	594	636
fla6	504	693	615	621	448	522	346	460	613	653	41	81	88	43	27	0	31	47	43	34	329	589	287	493	404	677	383	380	629	717
fla7	568	690	670	656	475	545	382	500	675	675	23	92	64	26	51	31	0	85	31	57	338	695	381	586	431	734	472	444	764	748
fla8	418	609	503	525	358	427	330	387	503	528	120	116	146	107	39	47	85	0	96	67	307	459	254	410	304	526	317	296	482	571
fla9	524	665	641	657	431	528	349	477	639	611	41	109	70	27	41	43	31	96	0	65	321	664	379	566	398	714	451	429	721	706
fla10	494	662	612	576	456	472	339	466	617	629	44	106	122	59	30	34	57	67	65	0	287	592	313	490	393	670	410	380	633	697
fis1	410	515	505	467	387	351	365	396	524	392	349	359	420	329	291	329	338	307	321	287	0	325	237	191	223	608	205	285	339	298
fis2	527	537	513	501	393	360	482	404	510	464	752	579	684	681	548	589	695	459	664	592	325	0	214	173	265	273	249	199	241	407
fis3	326	461	528	451	355	355	357	419	491	388	409	256	363	379	302	287	381	254	379	313	237	214	0	159	113	311	80	183	252	305
fis4	428	530	526	519	399	331	423	402	527	347	616	500	591	591	478	493	586	410	566	490	191	173	159	0	204	384	184	214	146	198
fis5	421	531	562	515	370	368	358	443	550	398	480	366	442	480	348	404	431	304	398	393	223	265	113	204	0	235	143	200	298	278
fis6	496	514	487	494	495	535	545	461	613	521	785	610	721	713	628	677	734	526	714	670	608	273	311	384	235	0	450	216	454	492
fis7	436	498	483	439	412	384	352	397	531	393	490	368	453	477	377	383	472	317	451	410	205	249	80	184	143	450	0	302	176	247
fis8	445	532	484	491	363	384	401	352	539	450	474	342	440	469	355	380	444	296	429	380	285	199	183	214	200	216	302	0	354	460
fis9	532	592	540	543	513	447	551	510	611	430	812	634	724	769	594	629	764	482	721	633	339	241	252	146	298	454	176	354	0	188
fis10	547	544	596	599	517	421	518	574	547	419	772	685	841	760	636	717	748	571	706	697	298	407	305	198	278	492	247	460	188	0

Tabelle 5.14: Globale Distanzmatrix mit den globalen Distanzen der Konturen von den Vögeln, Flaschen und Fischen in Abbildung 5.20 mithilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$. In jeder Spalte sind die 9 kleinsten globalen Distanzen markiert. Jeder Eintrag in der Matrix muss mit 1000 multipliziert werden, um die Originaldistanzen zu erhalten.

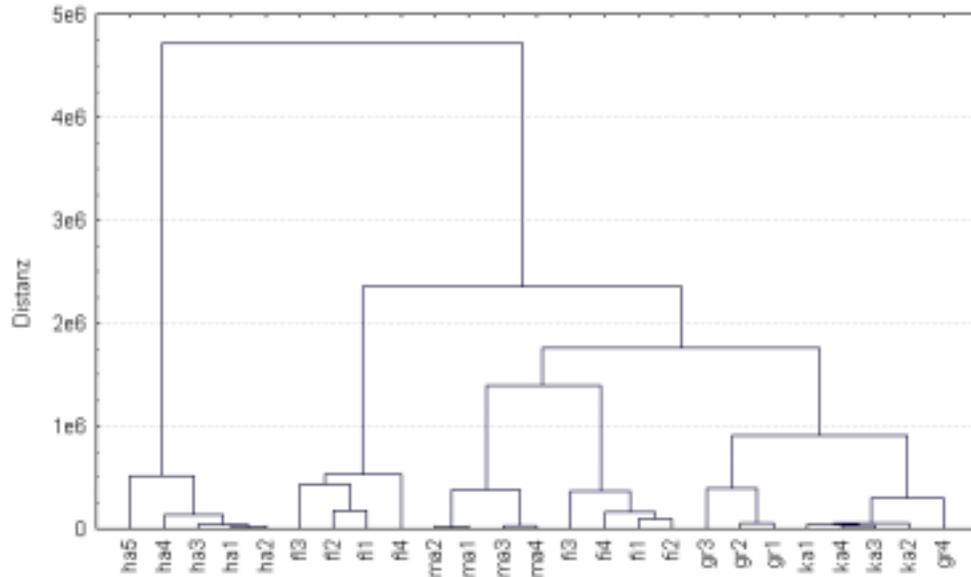


Abbildung 5.24: Dendrogramm der Clusteranalyse mit Ward-Methode für die Konturen des Sharvit-Kimia-Datensatzes aus Abbildung 5.21 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$.

zeuge, Kaninchen und Maulschlüssel sowie 5 Konturen in der Klasse Hände. Dieser Datensatz ist damit der anspruchsvollste der hier vorgestellten Datensätze.

Da jede Klasse aus mindestens 4 Konturen besteht, wurden die jeweils 3 nächsten Nachbarn zu jeder Kontur bestimmt. Bei 25 Konturen ist das bestmögliche Ergebnis für jeden der 3 nächsten Nachbarn ein Zuordnungsverhältnis von $V = \frac{25}{25} = 1.00$.

Die Ergebnisse der einzelnen Methoden sind in den Tabellen 5.15 - 5.18 zu finden. Sharvit und Kimia verglichen den Datensatz in [SCTK98] basierend auf sogenannten *Shock-Graphs* [SSDZ98], einem strukturellen Verfahren. Ihr Ergebnis ist in Tabelle 5.19 eingetragen. Und Gdalyahu und Weinshall erhalten für denselben Datensatz das in Tabelle 5.20 gezeigte Ergebnis basierend auf ihrer *Flexible Syntactic Matching* - Methode [GW99]. D.h. die Ergebnisse, die die Methode mit Hilfe globaler Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$ erzielt, ist mit den Methoden von Sharvit und Kimia bzw. Gdalyahu und Weinshall durchaus vergleichbar.

Die globale Distanzmatrix für $l_{const} = 20$ zeigt Tabelle 5.21. Das dazugehörige Ergebnis der Clusteranalyse mit Ward-Methode zeigt das Dendrogramm in Abbildung 5.24. Bis auf den Greeble gr4, der bei der Clusteranalyse der Klasse der Kaninchen zugeordnet wird, könnten alle anderen Objekte unüberwacht den

i. NN	1	2	3
V	0.92	0.96	0.92

Tabelle 5.15: Tabelle mit Klassifikationsergebnissen für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

i. NN	1	2	3
V	0.88	0.88	0.76

Tabelle 5.16: Tabelle mit Klassifikationsergebnissen für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 10$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

i. NN	1	2	3
V	0.88	0.88	0.76

Tabelle 5.17: Tabelle mit Klassifikationsergebnissen für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 1000.0$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

i. NN	1	2	3
V	0.80	0.88	0.72

Tabelle 5.18: Tabelle mit Klassifikationsergebnissen für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe der lokalen Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für $E = 200.0$. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

i. NN	1	2	3
V	0.92	0.84	0.80

Tabelle 5.19: Tabelle mit Klassifikationsergebnissen von Sharvit und Kimia [SCTK98] für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe von *Shock-Graphs* [SSDZ98]. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

i. NN	1	2	3
V	1.00	0.84	0.76

Tabelle 5.20: Tabelle mit Klassifikationsergebnissen von Gdalyahu und Weinshall [GW99] für die Konturen des Sharvit-Kimia-Datensatzes in Abbildung 5.21 mit Hilfe der *Flexible Syntactic Matching* - Methode [GW99]. NN: Nächster Nachbar, V: Zuordnungsverhältnis.

richtigen Klassen zugeordnet werden.

Die Ergebnisse bzgl. der globalen Ähnlichkeit lassen sich weiter verbessern, wenn l_{const} und E weiter erhöht werden. Für $l_{const} = 100$, also für lokale Konturabschnitte, die fast die gesamte Kontur ausmachen, ergibt sich das bestmögliche Ergebnis von $V = \frac{25}{25} = 1.00$ für die ersten, zweiten und dritten nächsten Nachbarn. Auch für $E = 20000.0$ ergeben sich bessere Werte als mit niedrigerem E . Das Ergebnis mit $V = \frac{24}{25} = 0.96$ für den ersten nächsten Nachbarn, $V = \frac{23}{25} = 0.92$ für den zweiten nächsten Nachbarn und $V = \frac{22}{25} = 0.88$ für den dritten nächsten Nachbarn kommt aber nicht an das Ergebnis für $l_{const} = 100$ heran.

Damit kann aber wohl kaum noch von einer lokalen Analyse der Konturen gesprochen werden. D.h. dem lokalen Ansatz dieser Arbeit würde widersprochen. Außerdem steigt die Zeit zur Berechnung der lokalen Distanz- bzw. Ähnlichkeitsmatrizen bei größerem l_{const} bzw. E .

Für $l_{const} = 20$ und $E = 1000.0$ kann aber noch von einer lokalen Analyse gesprochen werden. Insbesondere für $l_{const} = 20$ konnte bei allen Datensätzen ein sehr gutes globales Konturvergleichsergebnis erzielt werden.

	fi1	fi2	fi3	fi4	gr1	gr2	gr3	gr4	ha1	ha2	ha3	ha4	ha5	fl1	fl2	fl3	fl4	ka1	ka2	ka3	ka4	ma1	ma2	ma3	ma4
fi1	0	98	291	152	519	541	444	393	1030	1038	1072	898	866	794	731	519	470	415	405	410	399	434	405	359	354
fi2	98	0	312	145	560	591	486	429	1101	1110	1123	962	875	804	724	523	542	388	383	404	415	557	533	411	352
fi3	291	312	0	251	612	657	658	576	880	884	875	810	893	722	707	564	393	560	521	594	554	680	611	626	584
fi4	152	145	251	0	501	520	503	402	1006	1019	1021	911	785	798	808	547	489	381	367	387	398	509	464	425	370
gr1	519	560	612	501	0	51	281	272	1045	1022	1044	902	823	787	800	549	687	339	337	357	346	623	596	553	531
gr2	541	591	657	520	51	0	327	294	981	953	969	845	761	722	745	540	610	410	366	438	403	589	573	539	529
gr3	444	486	658	503	281	327	0	394	969	995	983	834	599	863	865	521	594	356	333	350	337	416	419	416	436
gr4	393	429	576	402	272	294	394	0	1036	1043	1077	1003	801	972	1010	641	711	200	226	194	187	546	556	427	395
ha1	1030	1101	880	1006	1045	981	969	1036	0	12	33	91	343	770	842	817	727	936	965	997	999	931	938	946	969
ha2	1038	1110	884	1019	1022	953	995	1043	12	0	35	102	340	808	885	821	701	943	970	1005	1001	942	951	979	995
ha3	1072	1123	875	1021	1044	969	983	1077	33	35	0	117	332	819	861	840	711	954	976	1027	1002	930	955	989	1027
ha4	898	962	810	911	902	845	834	1003	91	102	117	0	357	749	761	732	658	820	865	896	873	870	857	824	844
ha5	866	875	893	785	823	761	599	801	343	340	332	357	0	771	775	670	611	786	776	811	835	816	788	741	771
fl1	794	804	722	798	787	722	863	972	770	808	819	749	771	0	178	379	413	760	801	810	826	894	897	842	855
fl2	731	724	707	808	800	745	865	1010	842	885	861	761	775	178	0	355	524	729	734	811	779	897	868	1064	1075
fl3	519	523	564	547	549	540	521	641	817	821	840	732	670	379	355	0	433	463	439	516	488	699	686	536	542
fl4	470	542	393	489	687	610	594	711	727	701	711	658	611	413	524	433	0	555	624	602	633	691	660	545	531
ka1	415	388	560	381	339	410	356	200	936	943	954	820	786	760	729	463	555	0	50	34	41	564	549	411	364
ka2	405	383	521	367	337	366	333	226	965	970	976	865	776	801	734	439	624	50	0	48	42	530	514	378	350
ka3	410	404	594	387	357	438	350	194	997	1005	1027	896	811	810	811	516	602	34	48	0	29	530	541	409	365
ka4	399	415	554	398	346	403	337	187	999	1001	1002	873	835	826	779	488	633	41	42	29	0	518	525	422	372
we1	434	557	680	509	623	589	416	546	931	942	930	870	816	894	897	699	691	564	530	530	518	0	18	193	201
we2	405	533	611	464	596	573	419	556	938	951	955	857	788	897	868	686	660	549	514	541	525	18	0	202	211
we3	359	411	626	425	553	539	416	427	946	979	989	824	741	842	1064	536	545	411	378	409	422	193	202	0	28
we4	354	352	584	370	531	529	436	395	969	995	1027	844	771	855	1075	542	531	364	350	365	372	201	211	28	0

Tabelle 5.21: Globale Distanzmatrix mit den globalen Distanzen der Konturen des Shavit-Kimia-Datensatzes in Abbildung 5.21 mithilfe der lokalen Distanzmatrizen basierend auf mittlerer Orientierung für $l_{const} = 20$. In jeder Spalte sind die 3 kleinsten globalen Distanzen markiert. Jeder Eintrag in der Matrix muss mit 1000 multipliziert werden, um die Originaldistanzen zu erhalten.

5.5 Klassifikation mit signifikanten Abschnitten

In diesem Abschnitt sind Beispiele für signifikante Konturabschnitte und die Klassifikationsergebnisse basierend auf diesen signifikanten Konturabschnitten dokumentiert.

Als Lernstichproben werden die Datensätze in den Abbildungen 5.19, 5.20 und 5.21 herangezogen, die auch schon für die Experimente in Abschnitt 5.4 verwendet worden sind. Allerdings wurde bei dem Datensatz von Sharvit und Kimia die Hand ha5 aus der Lernstichprobe in die Teststichprobe verschoben, um dort eine größere Anzahl an Händen zu erhalten.

Alle Datensätze der Lern- und Teststichproben wurden auf 500 PE Umfang skaliert und danach mit 2.0 PE abgetastet. Die Datensätze wurden mit den Methoden basierend auf mittlerer Orientierung analysiert. Ferner wurden die Anfangs- und Endpunktindices der signifikanten Konturabschnitte mit Hilfe der Mittelwerte abzüglich bzw. hinzugezählter einfacher Standardabweichung der Start- bzw. Endpunktindices sich überlappender wichtiger Konturabschnitte bestimmt (siehe Gleichungen 4.79 - 4.82). Die Konturen wurden überwacht in Klassen eingeteilt.

5.5.1 30 Kunstkonturen

In Abbildung 5.25 sind die signifikanten Konturabschnitte der Lernstichprobe fett markiert. Diese signifikanten Konturabschnitte wurden mit Hilfe lokaler Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für eine maximal zulässige Distanz von $E = 200.0$ erzeugt. Als Importanzschwelle wurde $\tilde{\Theta} = 15$ benutzt³.

Die signifikanten Konturabschnitte wurden auf diejenigen beschränkt, die eine Konturpunktanzahl $L_{min} \geq 30$ besitzen. Diese signifikanteren Konturabschnitte wurden auf ihre Verlässlichkeit für eine Klassifikation überprüft. Das Ergebnis zeigt Abbildung 5.26. Die Konturen sind dort bzgl. der Reliabilität ρ ihrer signifikanten Konturabschnitte geordnet. An den Abschnitten ist jeweils ihre Platzierung und die zugehörige Reliabilität notiert. Beispiel: Der signifikante Abschnitt der Kontur k10b ist der 9. verlässlichste signifikante Konturabschnitt der Lernstichprobe mit einer Reliabilität von 0.87.

Nach einer Einschränkung der Konturpunktanzahl der Konturabschnitte auf $L_{min} \geq 30$ bleiben alle Spitzen und einige wenige Konturabschnitte der Konturen ohne Spitzen übrig. 11 Konturen ohne Spitzen besitzen so gar keine signifikanteren Konturabschnitte mehr. Die Spitzen sind wesentlich verlässlicher für die

³Die Parameter Importanzschwelle $\tilde{\Theta} = 15$ und minimale Konturpunktanzahl $L_{min} \geq 30$ wurden mit Hilfe eines Programmes, das die entstehenden signifikanten Konturabschnitte der Lernstichprobe in Abhängigkeit von diesen beiden Parametern visualisiert, durch Ausprobieren gefunden. Die sich so ergebenden Konturabschnitte sind nicht zu lang und nicht zu kurz (siehe dazu auch die kritischen Anmerkungen im Ausblick in Kapitel 6).

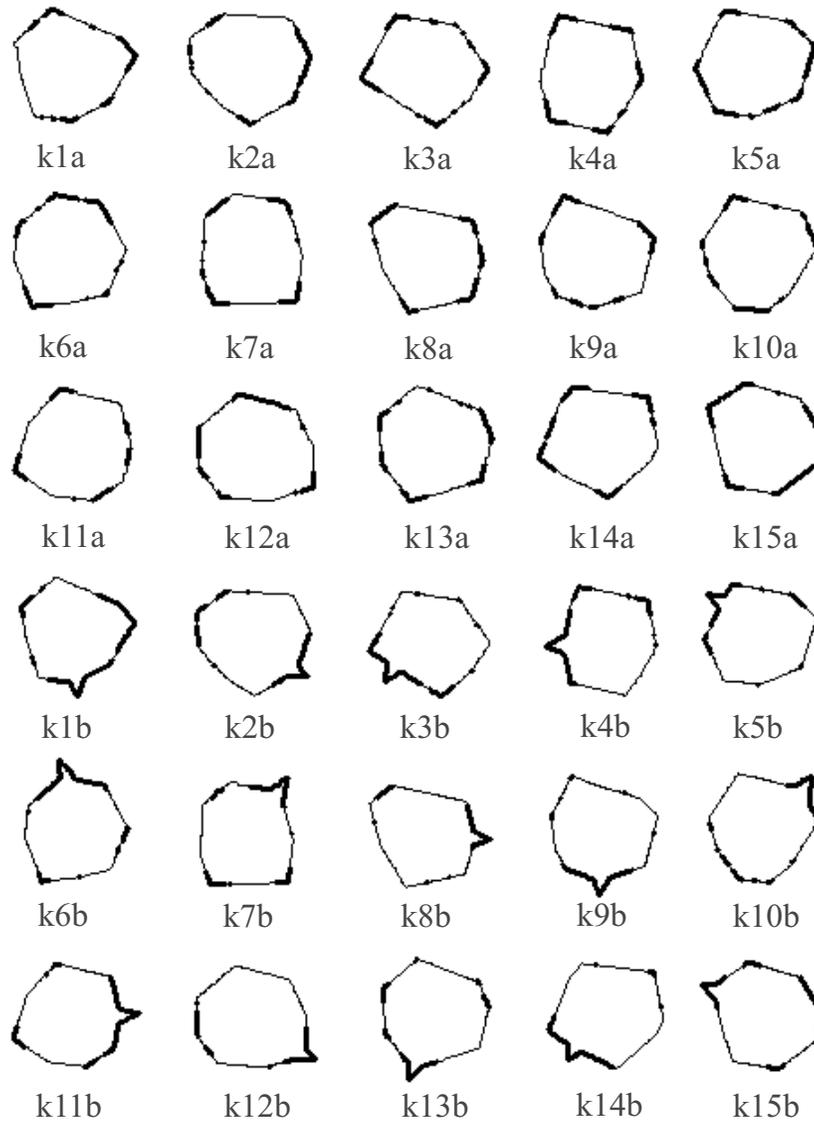


Abbildung 5.25: Signifikante Konturabschnitte (fett markiert) der Lernstichprobe bestehend aus 30 künstlichen Konturen mit und ohne Spitzen. Methode: lokale Ähnlichkeitsmatrizen, mittlere Orientierung, $E = 200.0$, Importanzschwelle $\tilde{\Theta} = 15$.

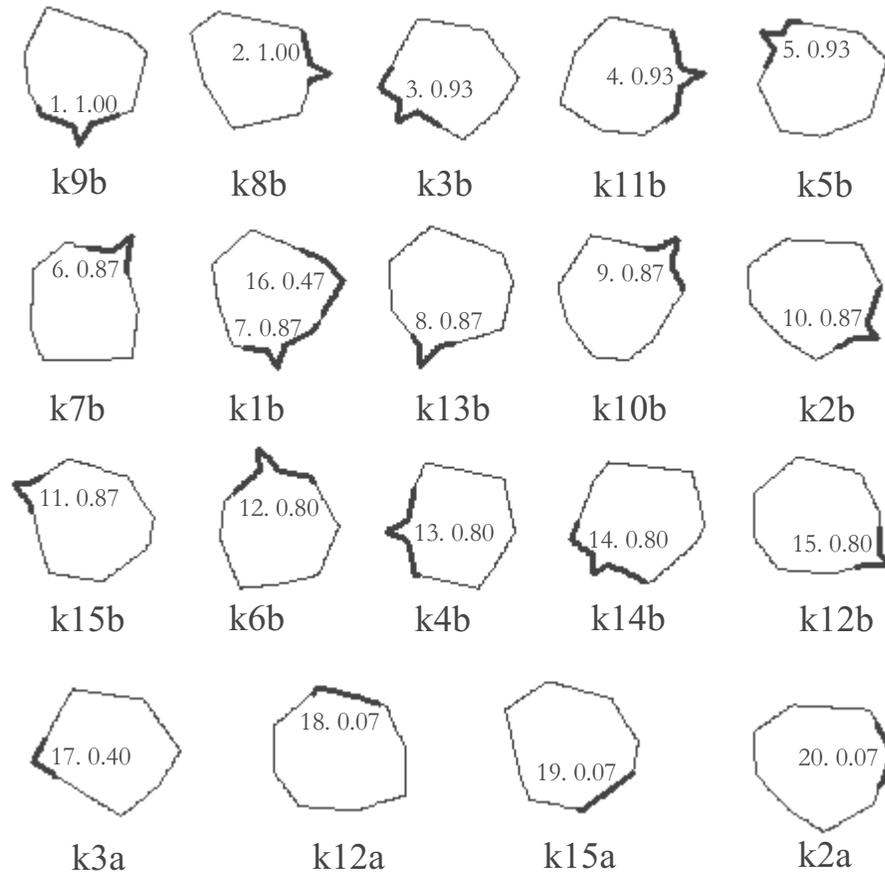


Abbildung 5.26: Abgebildet sind die signifikanteren Konturabschnitte der Lernstichprobe bestehend aus 30 künstlichen Konturen mit und ohne Spitzen mit einer Mindestkonturpunktanzahl von 30. 11 Konturen ohne Spitze haben keine signifikanteren Konturabschnitte und sind deshalb nicht abgebildet. Die Konturen sind gemäß der Reliabilität ihrer fett markierten signifikanten Konturabschnitte geordnet. Neben den Konturabschnitten ist ihre Platzierung innerhalb der Lernstichprobe und ihre Reliabilität notiert.

Istklasse → Sollklasse ↓	ohne Spitze	mit Spitze	unbekannt	Ausbeute ⁴
ohne Spitze	-	1	14	$\frac{1}{15} = 6.7\%$
mit Spitze	-	15	-	$\frac{15}{15} = 100.0\%$
Reinheit ⁵	-	$\frac{15}{16} = 93.8\%$	X	TQ ⁶ : $\frac{15}{30} = 50\%$

Tabelle 5.22: Tabelle mit Klassifikationsergebnissen für die künstlichen Konturen der Teststichprobe in Abbildung 5.27.

Klassifikation als die anderen signifikanten Konturabschnitte. Es liegt eine große Lücke zwischen der Reliabilität 0.8 des 15. verlässlichsten Konturabschnitts mit Spitze und der Reliabilität 0.4 des verlässlichsten Konturabschnitts ohne Spitze.

Die Teststichprobe zeigt Abbildung 5.27. Es handelt sich auch hier um je 15 künstliche Konturen mit und ohne Spitzen. Das Klassifikationsergebnis ist in Tabelle 5.22 zusammengefasst.

Das detaillierte Klassifikationsergebnis mit zu klassifizierender Kontur (Testkontur), ähnlichstem Konturabschnitt (Istabschnitt), Distanz des Konturabschnitts aus der Wissensbasis des hierarchischen Klassifikators zum ähnlichsten Konturabschnitt in zu klassifizierender Kontur (Distanz) und Konfidenz τ des Konturabschnitts aus der Wissensbasis zeigt Tabelle 5.23.

Es sind bei diesem Datensatz zwei wesentliche Ergebnisse festzuhalten.

Zum einen werden alle Konturen mit Spitze richtig klassifiziert. Und das im wesentlichen mit Hilfe des 1. verlässlichsten Konturabschnitts. Nur einmal musste der Klassifikator auf den 2. verlässlichsten Konturabschnitt zurückgreifen. Damit wird die Klassifikation bzgl. der Konturen mit Spitze sehr schnell.

Zum anderen wird keine einzige Kontur ohne Spitze richtig klassifiziert. 14 Konturen ohne Spitze werden als unbekannt und einer sogar falsch klassifiziert. Dies Ergebnis sieht auf den ersten Blick schlecht aus. Bei näherer Betrachtung ist dieses Ergebnis für die 14 als unbekannt klassifizierten Konturen ohne Spitze aber

⁴Unter Ausbeute wird das Verhältnis von richtig in eine bestimmte Klasse eingeordnete Testkonturen und Gesamtanzahl der Testkonturen in dieser Klasse verstanden.

⁵Unter Reinheit wird das Verhältnis von richtig in eine bestimmte Klasse eingeordnete Testkonturen und Gesamtanzahl der Testkonturen, die in diese Klasse eingeordnet wurden, verstanden. Die Reinheit ist bei bestimmten Sortieraufgaben ein wichtiges Gütemaß. So wird beispielsweise in [Kuh98] eine Sortierung von Verpackungsmaterialien aus Kunststoff vorgestellt, deren sortierte Fraktionen einem Recyclingprozess zugeführt werden. Hier ist ganz entscheidend, dass die Reinheit der einzelnen Fraktionen hoch, d.h. die Anzahl der Fehlklassifikationen klein ist, um die einzelnen Fraktionen überhaupt recyceln zu können.

⁶TQ ist die Abkürzung von Trefferquote. Die Trefferquote gibt das Verhältnis von allen richtig eingeordneten Konturen der Teststichprobe und der Gesamtanzahl der Konturen in der Teststichprobe an. Sie ist damit ein Gütemaß für den gesamten Klassifikationsprozess.

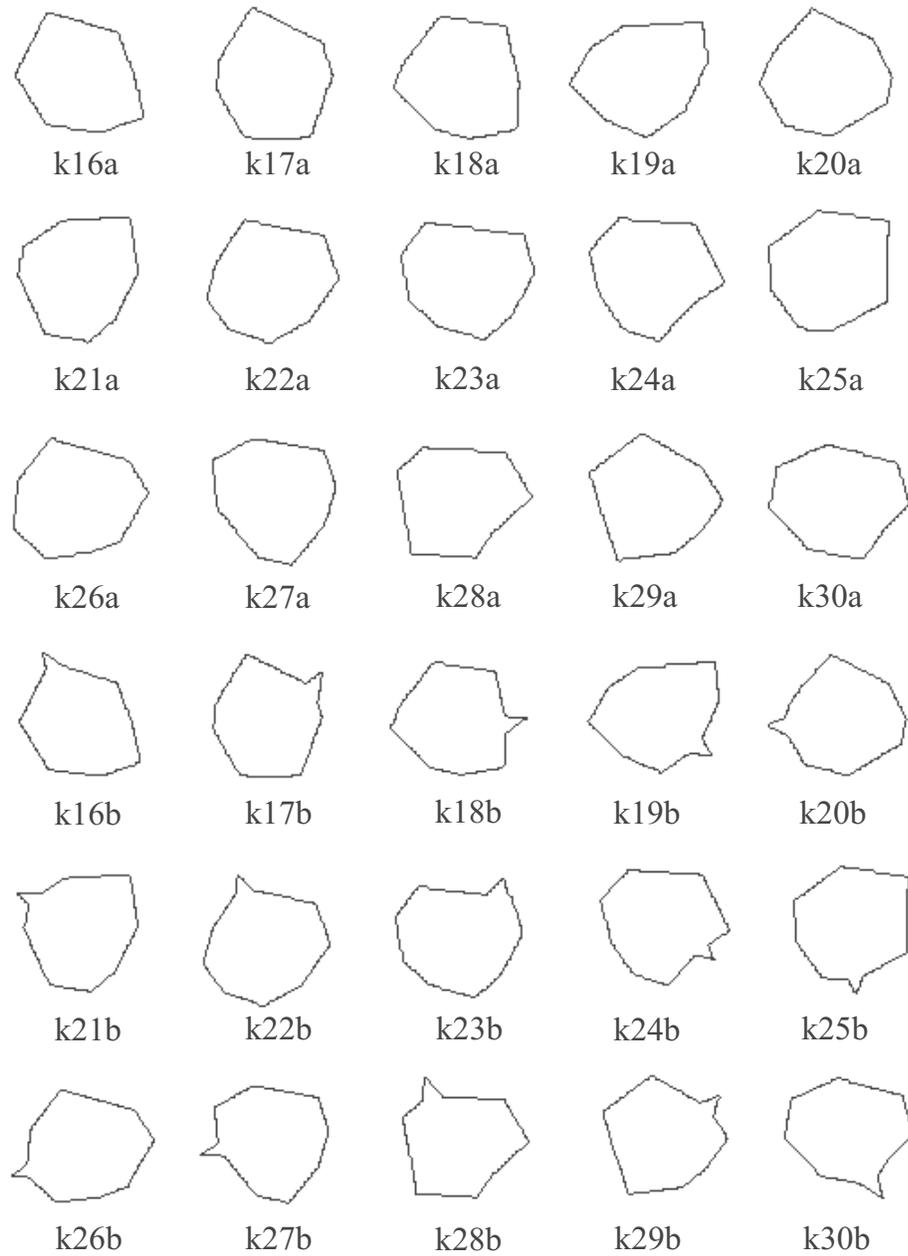


Abbildung 5.27: Teststichprobe mit 15 künstlichen Konturen k16a-k30a ohne Spitze und 15 künstlichen Konturen k16b-k30b mit Spitze.

Testkontur	Istabschnitt	Distanz	Konfidenz
k16a	k11b 4. 0.93	75.48	< 84
k17a	unbekannt	-	-
k18a	unbekannt	-	-
k19a	unbekannt	-	-
k20a	unbekannt	-	-
k21a	unbekannt	-	-
k22a	unbekannt	-	-
k23a	unbekannt	-	-
k24a	unbekannt	-	-
k25a	unbekannt	-	-
k26a	unbekannt	-	-
k27a	unbekannt	-	-
k28a	unbekannt	-	-
k29a	unbekannt	-	-
k30a	unbekannt	-	-
k16b	k8b 2. 1.00	8.56	< 50
k17b	k9b 1. 1.00	26.26	< 56
k18b	k9b 1. 1.00	34.07	< 56
k19b	k9b 1. 1.00	13.25	< 56
k20b	k9b 1. 1.00	13.32	< 56
k21b	k9b 1. 1.00	18.64	< 56
k22b	k9b 1. 1.00	23.51	< 56
k23b	k9b 1. 1.00	34.40	< 56
k24b	k9b 1. 1.00	50.63	< 56
k25b	k9b 1. 1.00	14.84	< 56
k26b	k9b 1. 1.00	9.36	< 56
k27b	k9b 1. 1.00	11.79	< 56
k28b	k9b 1. 1.00	7.41	< 56
k29b	k9b 1. 1.00	4.44	< 56
k30b	k9b 1. 1.00	51.01	< 56

Tabelle 5.23: Tabelle mit den detaillierten Klassifikationsergebnissen für die künstlichen Konturen der Teststichprobe in Abbildung 5.27. Fett markiert ist die Zeile mit der Falschklassifikation.

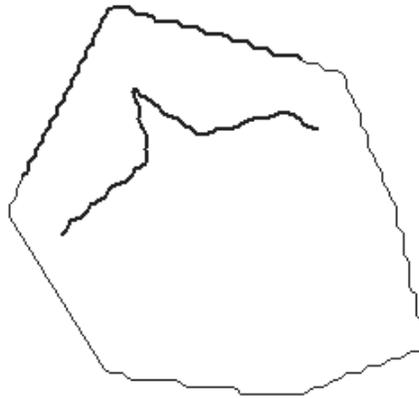


Abbildung 5.28: Fehlklassifikation einer Kontur ohne Spitze. Der fett markierte Konturabschnitt auf der Kontur k16a wird für den 4. verlässlichsten Konturabschnitt (Kontur k11b) der Lernstichprobe gehalten.

durchaus verständlich, da diese Konturen wie auch die Konturen der Lernstichprobe ohne Spitze im Grunde keine verlässlichen Konturabschnitte besitzen. Die nach der Einschränkung auf mindestens 30 Konturpunkte übrigbleibenden signifikanten Konturabschnitte sind nur deshalb signifikante Konturabschnitte der Konturen ohne Spitzen geworden, da es in der verwendeten Lernmethode zwingend signifikante Konturabschnitte geben muss, die mit den signifikanten Konturabschnitten *Spitzen* korrespondieren. Tatsächlich haben die Konturen ohne Spitze keine markanten, für eine Klassifikation verlässlichen Konturabschnitte, was das hier vorgestellte Klassifikationsergebnis bestätigt. Das einzige was diese Konturklasse ausmacht ist, dass sie **keine** Spitzen hat. Die *automatische Merkmalsynthese* basiert aber darauf, dass Konturen bestimmte sie auszeichnende Konturabschnitte besitzen, und **nicht** darauf, dass sie solche Abschnitte **nicht** besitzen.

Die einzige Falschklassifikation zeigt Abbildung 5.28. Hier wird der fett markierte Konturabschnitt der Kontur k16a für den 4. verlässlichsten Konturabschnitt der Lernstichprobe in Kontur k11b gehalten. Der Grund dafür liegt in diesem Fall in der zu hohen Konfidenz. Die Distanz zwischen den beiden Konturabschnitten beträgt 75.48 und liegt damit nur knapp unter der Konfidenz von 84 für den signifikanten Konturabschnitt. Damit wird zumindest eine gewisse Unsicherheit dieser Klassifikation festgestellt.

E bzw. l_{const}	$\tilde{\Theta}$ bzw. Θ	L_{min}	Trefferquote
$E = 200.0$	≤ 10	10	$\frac{58}{74} = 78.4\%$
$E = 200.0$	≤ 13	10	$\frac{66}{74} = 89.2\%$
$E = 200.0$	≤ 15	10	$\frac{70}{74} = 94.6\%$
$E = 1000.0$	≤ 15	20	$\frac{37}{74} = 50.0\%$
$E = 1000.0$	≤ 18	20	$\frac{61}{74} = 82.4\%$
$E = 1000.0$	≤ 20	20	$\frac{71}{74} = 95.9\%$
$l_{const} = 10$	≥ 750.0	10	$\frac{32}{74} = 43.2\%$
$l_{const} = 10$	≥ 450.0	10	$\frac{49}{74} = 66.2\%$
$l_{const} = 10$	≥ 150.0	10	$\frac{63}{74} = 85.1\%$
$l_{const} = 20$	≥ 5000.0	20	$\frac{45}{74} = 60.8\%$
$l_{const} = 20$	≥ 3500.0	20	$\frac{61}{74} = 82.4\%$
$l_{const} = 20$	≥ 2000.0	20	$\frac{58}{74} = 78.4\%$

Tabelle 5.24: Tabelle mit Trefferquoten des hierarchischen Klassifikators für die Vögel, Flaschen und Fische der Teststichprobe in Abbildung 5.35 für verschiedene Parameterkombinationen.

5.5.2 Vögel, Flaschen und Fische

Der Datensatz der Lernstichprobe der Vögel, Flaschen und Fische in Abbildung 5.20 wurde mit Hilfe von lokalen Distanz- und Ähnlichkeitsmatrizen basierend auf mittlerer Orientierung für verschiedene Parameterkombinationen analysiert. Die jeweiligen \rightarrow Trefferquoten der hierarchischen Klassifikation sind in Tabelle 5.24 zusammengefasst. E und l_{const} wurden so gewählt wie es die Voruntersuchungen im Abschnitt 5.3 empfehlen. Die Importanzschwellen $\tilde{\Theta}$ bzw. Θ sind so gewählt worden, dass nicht ausschließlich extrem kurze bzw. die ganze Kontur umfassende signifikante Konturabschnitte entstanden.

Die Beschränkung der signifikanten auf signifikantere Konturabschnitte wurde ausschließlich durch eine Mindestkonturpunktanzahl L_{min} vorgenommen. Es stellt sich nämlich als schwierig heraus die anderen zur Signifikanz beitragenden Werte nachvollziehbar gegeneinander zu gewichten. Gute Ergebnisse konnten zwar dadurch erzielt werden, die Anzahl der überlappenden wichtigen Konturabschnitte hoch zu gewichten, doch lassen sich ähnliche Ergebnisse allein mit einer Mindestkonturpunktanzahl L_{min} erreichen. L_{min} wurde den Parametern E und l_{const} angepasst. Das soll heißen, dass je größer E , desto größer L_{min} , da bei größeren E längere wichtige Konturabschnitte und damit auch längere signifikante Konturabschnitte produziert werden. Bei den l_{const} wurden pragmatisch die L_{min} auf jeweils die Hälfte der Konturpunktanzahl der miteinander verglichenen lokalen Konturabschnitte beschränkt.

Zwei Hauptergebnisse sind in Tabelle 5.24 abzulesen: zum einen ergeben sich, bis auf einen Fall, umso bessere \rightarrow *Trefferquoten*, je länger die signifikanten Konturabschnitte sind. So wird das beste Ergebnis für die Parametereinstellungen $E = 1000.0$, $\tilde{\Theta} = 20$ und $L_{min} = 20$ mit einer Trefferquote von 95.9% und das zweitbeste Ergebnis für die Parametereinstellungen $E = 200.0$, $\tilde{\Theta} = 15$ und $L_{min} = 20$ mit einer Trefferquote von 94.6% erreicht. **Dies heißt aber nicht, dass die längstmöglichen Konturabschnitte (also die gesamten Konturen) zu den höchsten Trefferquoten führen. Auch hierzu wurde ein Experiment durchgeführt, mit dem nur eine Trefferquote von 81.1% erzielt wurde.** Zum anderen werden im Gegensatz zu den globalen Vergleichen im Abschnitt 5.4 die besseren Ergebnisse mit den Methoden basierend auf lokalen Ähnlichkeitsmatrizen erzielt.

Bei kürzeren (also lokaleren) signifikanten Konturabschnitten schneidet die Parameterkombination $E = 200.0$, $\tilde{\Theta} = 10$ und $L_{min} = 20$ mit einer Trefferquote von 78.4% am besten ab. Dies wird zum Anlass genommen alle Ergebnisse für $E = 200.0$ hier näher vorzustellen.

In den Abbildungen 5.29, 5.30, 5.31, 5.32, 5.33 und 5.34 sind aus diesem Grunde die signifikanten und die nach Verlässlichkeit geordneten signifikanteren Konturabschnitte für alle untersuchten Importanzschwellen bei $E = 200.0$ abgebildet.

Man erkennt deutlich, dass bei steigender Importanzschwelle $\tilde{\Theta}$, d.h. bei steigender Anzahl von wichtigen Konturpunkten, auch die signifikanten Konturabschnitte länger werden. Sind bei $\tilde{\Theta} = 10$ die signifikanten Konturabschnitte noch sehr kurz (Krallen, Schnäbel, Flossen, Flaschenhälse), so umfassen sie bei $\tilde{\Theta} = 15$ teilweise mehr als die halbe Kontur.

Ferner ist auffällig, dass bei kurzen signifikanten Konturabschnitten gilt: je mehr Krümmungsänderungen in den signifikanten Konturabschnitten vorhanden sind, desto verlässlicher sind diese Konturabschnitte. Dadurch sind die Krallen der Vögel gefolgt von den Flossen der Fische am verlässlichsten für eine korrekte Klassifikation. Die Flaschenhälse folgen in der Mehrzahl am Ende der Reliabilitätskala.

Je länger die signifikanten Konturabschnitte werden, desto mehr vermischen sich die signifikanten Konturabschnitte der verschiedenen Klassen bzgl. ihrer Reliabilität. Hier werden globalere Merkmale wichtig. Z.B. hat die Flasche *fla2* in Abbildung 5.34 den 2. verlässlichsten signifikanten Konturabschnitt der Lernstichprobe, da sich an den Flaschenhals ein längerer gerader Abschnitt anschließt, der in der Lernstichprobe so nur bei den Flaschen wiederzufinden ist. Die Flaschenhälse alleine konnten noch zu leicht mit den Flossen der Fische, den Schnäbeln oder bestimmten Schwänzen der Vögel verwechselt werden.

Die Klassifikationsergebnisse für die Teststichprobe in Abbildung 5.35 sind in den Tabellen 5.25 - 5.30 zusammengetragen. Sie zeigen die zusammengefassten und die detaillierten Klassifikationsergebnisse des jeweils eingelernten hierarchi-

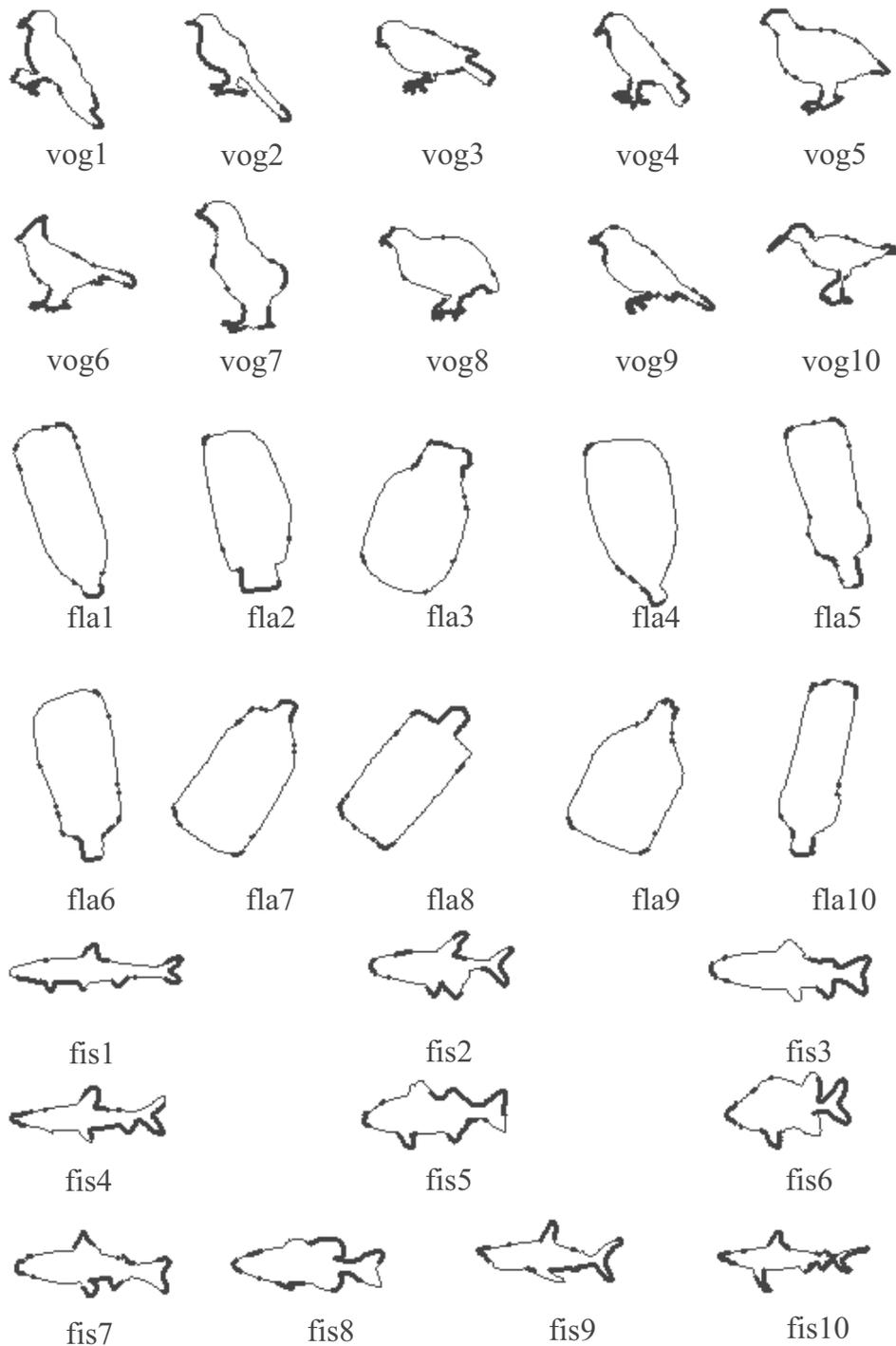


Abbildung 5.29: Signifikante Konturabschnitte (fett markiert) der Lernstichprobe bestehend aus 10 Vögeln, 10 Flaschen und 10 Fischen. Methode: lokale Ähnlichkeitsmatrizen, $E = 200.0$, Importanzschwelle $\tilde{\Theta} = 10$.

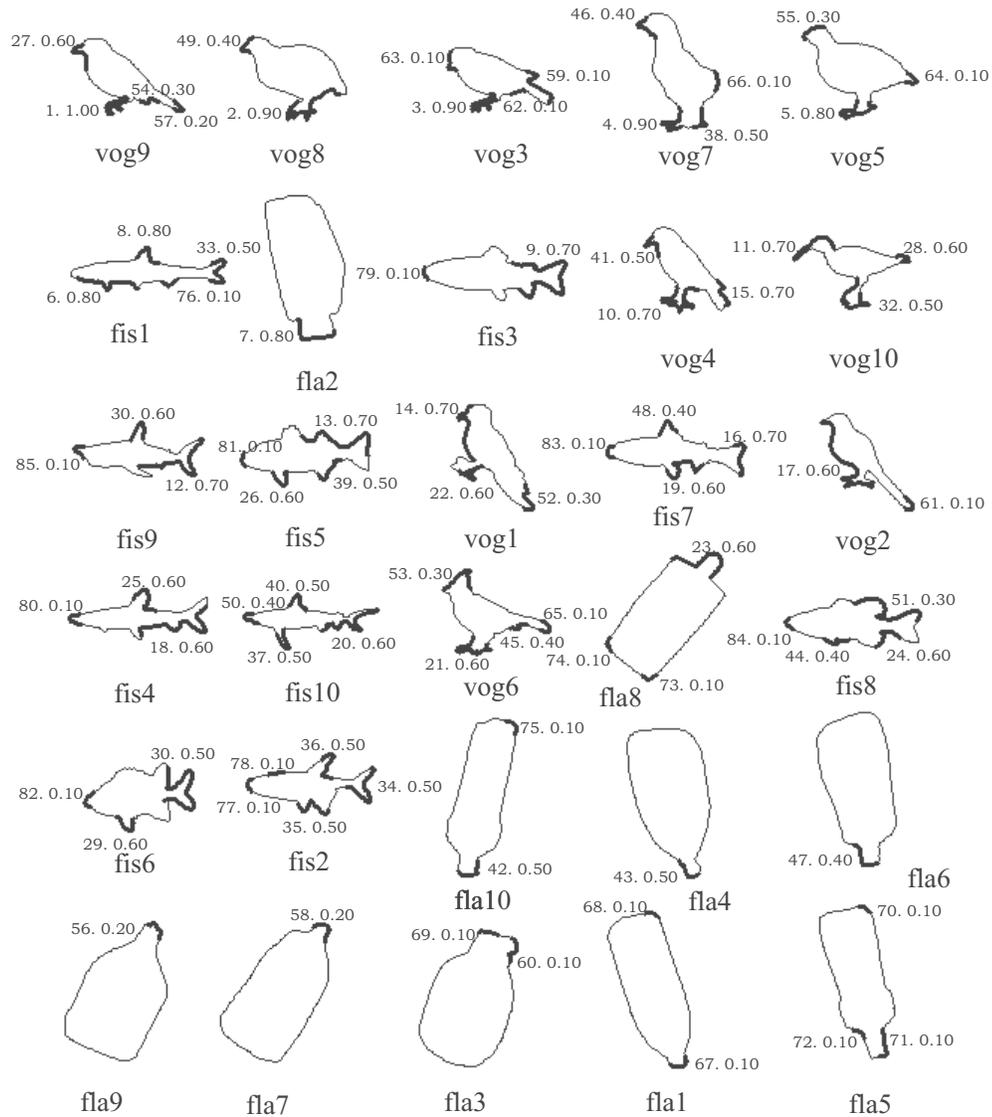


Abbildung 5.30: Abgebildet sind die signifikanteren Konturabschnitte der Lernstichprobe bestehend aus 10 Vögeln, 10 Flaschen und 10 Fischen für eine Wichtigkeitsschwelle von $\tilde{\Theta} = 10$ und eine Mindestkonturpunktanzahl von $L_{min} = 10$. Die Konturen sind gemäß der Reliabilität ihrer fett markierten signifikanten Konturabschnitte geordnet. Neben den Konturabschnitten ist ihre Plazierung bzgl. ihrer Verlässlichkeit innerhalb der Lernstichprobe und ihre Reliabilität notiert.

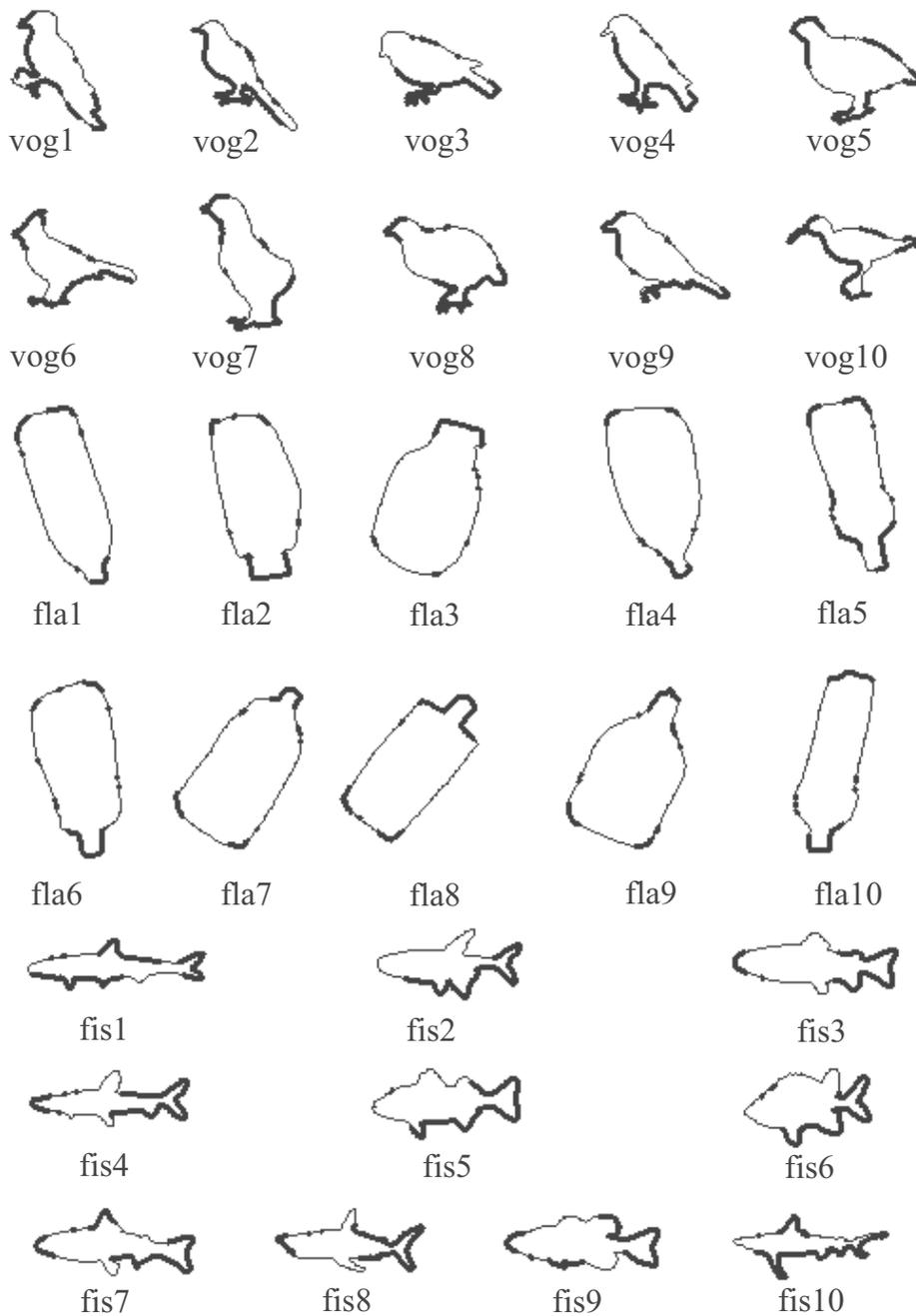


Abbildung 5.31: Signifikante Konturabschnitte (fett markiert) der Lernstichprobe bestehend aus 10 Vögeln, 10 Flaschen und 10 Fischen. Methode: lokale Ähnlichkeitsmatrizen, $E = 200.0$, Importanzschwelle $\hat{\Theta} = 13$.

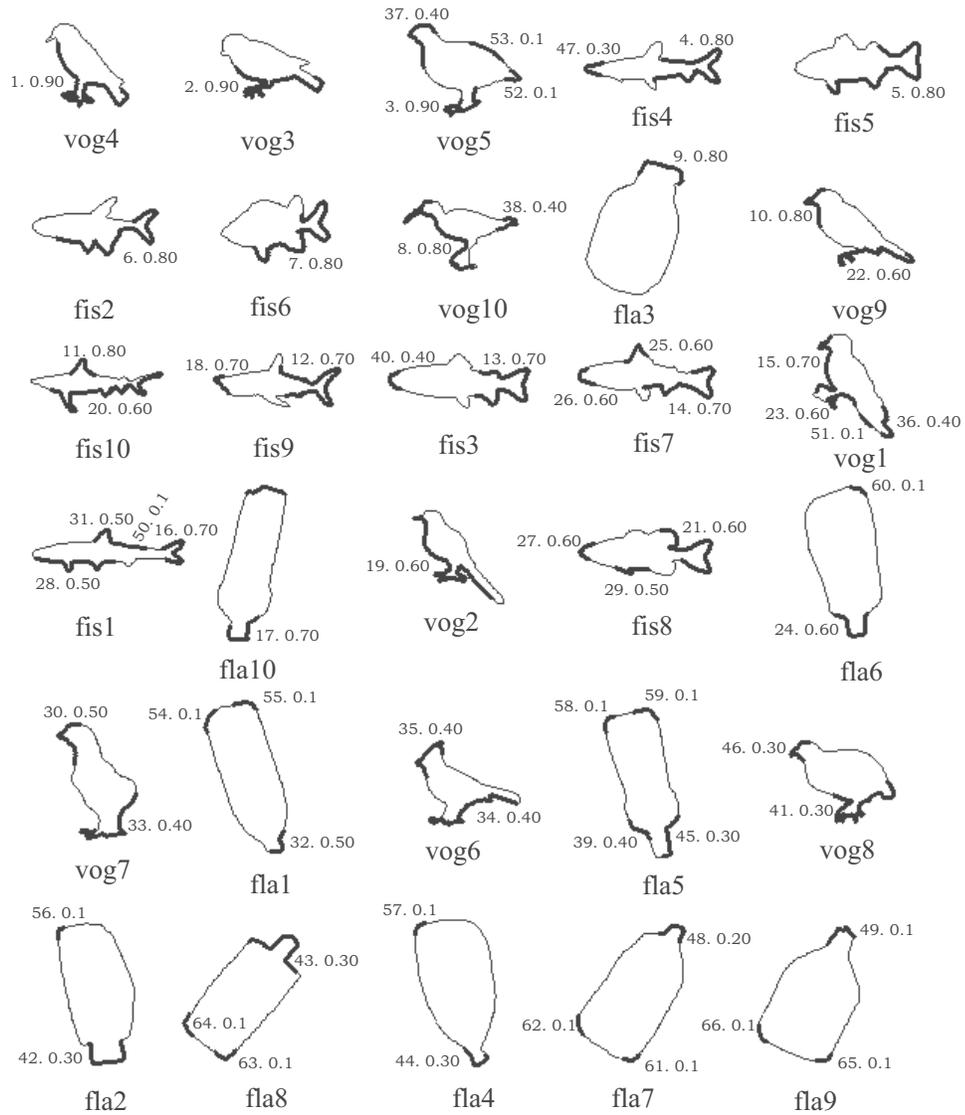


Abbildung 5.32: Abgebildet sind die signifikanteren Konturabschnitte der Lernstichprobe bestehend aus 10 Vögeln, 10 Flaschen und 10 Fischen für eine Importanzschwelle von $\tilde{\Theta} = 13$ und eine Mindestkonturpunktzahl von $L_{min} = 10$. Die Konturen sind gemäß der Reliabilität ihrer fett markierten signifikanten Konturabschnitte geordnet. Neben den Konturabschnitten ist ihre Plazierung bzgl. ihrer Verlässlichkeit innerhalb der Lernstichprobe und ihre Reliabilität notiert.

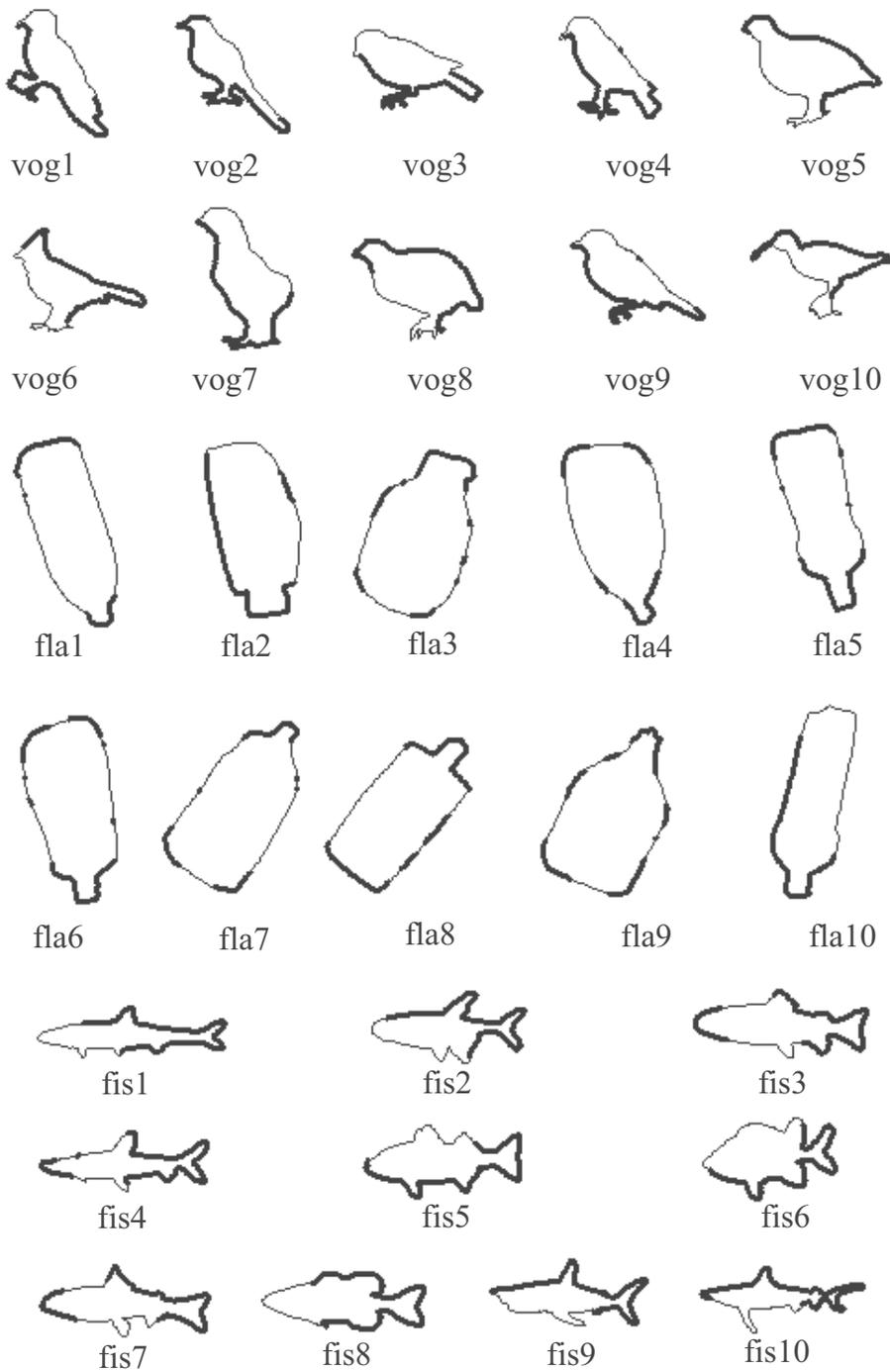


Abbildung 5.33: Signifikante Konturabschnitte (fett markiert) der Lernstichprobe bestehend aus 10 Vögeln, 10 Flaschen und 10 Fischen. Methode: lokale Ähnlichkeitsmatrizen, $E = 200.0$, Importanzschwelle $\tilde{\Theta} = 15$.

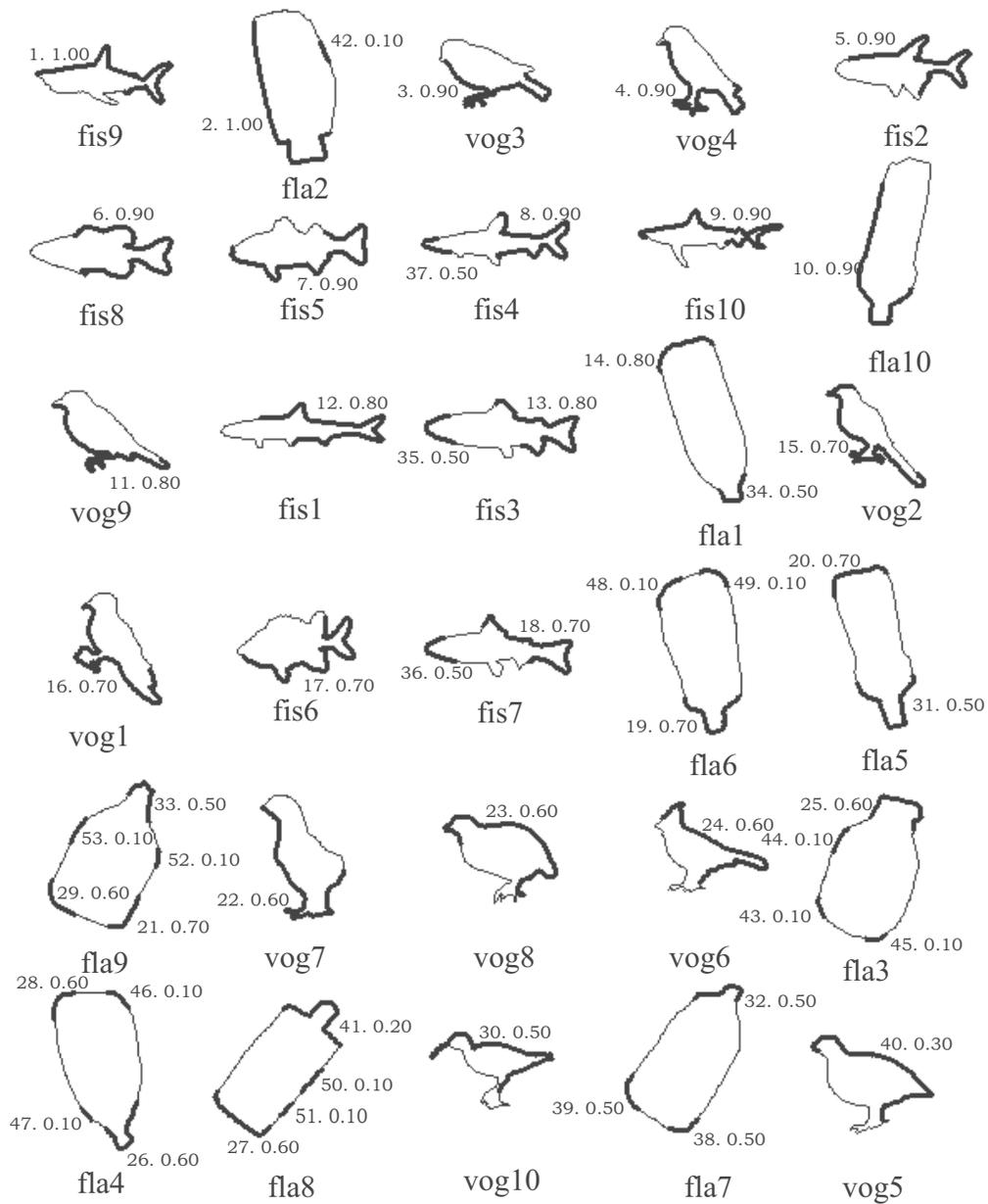


Abbildung 5.34: Abgebildet sind die signifikanteren Konturabschnitte der Lernstichprobe bestehend aus 10 Vögeln, 10 Flaschen und 10 Fischen für eine Importanzschwelle von $\hat{\Theta} = 15$ und eine Mindestkonturpunktanzahl von $L_{min} = 10$. Die Konturen sind gemäß der Reliabilität ihrer fett markierten signifikanten Konturabschnitte geordnet. Neben den Konturabschnitten ist ihre Platzierung bzgl. ihrer Verlässlichkeit innerhalb der Lernstichprobe und ihre Reliabilität notiert.

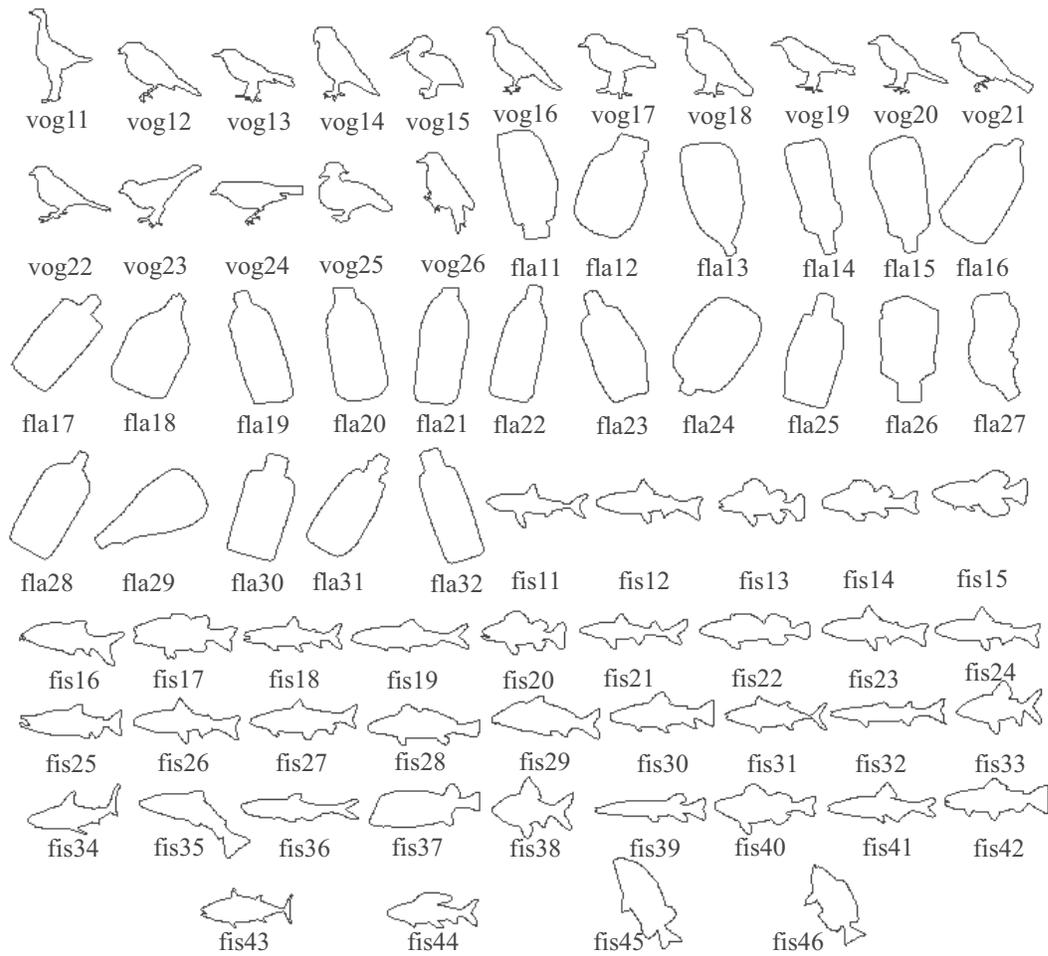


Abbildung 5.35: Teststichprobe mit 16 Vögeln vog11-vog26, 22 Flaschen fla11-fla32 und 36 Fischen fis11-fis46.

Testkontur	Istabschnitt	Distanz	Konfidenz	Testkontur	Istabschnitt	Distanz	Konfidenz
vog11	vog9 1. 1.00	86.6	< 90	fis11	fis1 6. 0.80	5.4	< 19
vog12	vog9 1. 1.00	5.4	< 90	fis12	fis3 9. 0.70	31.8	< 111
vog13	vog9 1. 1.00	88.1	< 90	fis13	fis3 9. 0.70	109.2	< 111
vog14	vog9 1. 1.00	48.9	< 90	fis14	fis1 6. 0.80	16.0	< 19
vog15	vog9 1. 1.00	72.6	< 90	fis15	vog4 15. 0.70	6.1	< 8
vog16	vog9 1. 1.00	41.4	< 90	fis16	fis9 30. 0.60	0.5	< 1
vog17	vog1 14. 0.70	6.5	< 12	fis17	fis1 6. 0.80	13.5	< 19
vog18	vog8 2. 0.90	58.2	< 76	fis18	fis1 6. 0.80	2.3	< 19
vog19	vog9 1. 1.00	89.1	< 90	fis19	fis1 6. 0.80	15.8	< 19
vog20	vog8 2. 0.90	49.4	< 76	fis20	fis3 9. 0.70	86.4	< 111
vog21	vog9 1. 1.00	2.8	< 90	fis21	fis1 6. 0.80	5.3	< 19
vog22	vog9 1. 1.00	53.4	< 90	fis22	fis3 9. 0.70	78.2	< 111
vog23	vog9 1. 1.00	52.2	< 90	fis23	fis3 9. 0.70	32.7	< 111
vog24	vog9 1. 1.00	34.2	< 90	fis24	fis3 9. 0.70	27.1	< 111
vog25	vog1 14. 0.70	7.1	< 12	fis25	fis3 9. 0.70	12.3	< 111
vog26	vog9 1. 1.00	87.0	< 90	fis26	fis3 9. 0.70	35.2	< 111
fla11	fla2 7. 0.80	0.0	< 5	fis27	fis1 6. 0.80	7.3	< 19
fla12	fla3 60. 0.10	0.0	< 2	fis28	fis3 9. 0.80	47.4	< 111
fla13	-	-	-	fis29	fis1 6. 0.80	15.3	< 19
fla14	fla8 23. 0.60	3.6	< 10	fis30	fis1 6. 0.80	1.5	< 19
fla15	fla6 47. 0.40	0.0	< 1	fis31	fis1 6. 0.80	6.1	< 19
fla16	-	-	-	fis32	fis1 6. 0.80	5.4	< 19
fla17	fla8 23. 0.60	3.6	< 10	fis33	vog7 4. 0.90	43.4	< 45
fla18	fla9 56. 0.20	0.0	< 1	fis34	fis1 8. 0.80	1.9	< 2
fla19	fla6 47. 0.40	0.0	< 1	fis35	fis3 9. 0.70	63.4	< 111
fla20	-	-	-	fis36	vog7 4. 0.90	41.3	< 45
fla21	-	-	-	fis37	fis7 48. 0.40	0.5	< 1
fla22	-	-	-	fis38	fis7 19. 0.60	29.8	< 40
fla23	fla8 23. 0.60	5.2	< 10	fis39	fis1 6. 0.80	2.1	< 19
fla24	-	-	-	fis40	fis1 8. 0.80	1.9	< 2
fla25	fla8 23. 0.60	9.1	< 10	fis41	fis1 6. 0.80	17.1	19
fla26	vog4 15. 0.70	7.0	< 8	fis42	fis1 6. 0.80	4.5	< 19
fla27	fis1 6. 0.80	16.3	< 19	fis43	fis1 6. 0.80	3.4	< 19
fla28	fla8 23. 0.60	7.3	< 10	fis44	vog7 4. 0.90	43.7	< 45
fla29	fla10 42. 0.50	0.7	< 1	fis45	vog4 15. 0.70	6.6	< 8
fla30	-	-	-	fis46	fis7 16. 0.70	3.2	< 5
fla 31	fis1 6. 0.80	16.3	19				
fla 32	-	-	-				

Tabelle 5.25: Tabelle mit den detaillierten Klassifikationsergebnissen für die Vögel, Flaschen und Fische in der Teststichprobe in Abbildung 5.35 für $\tilde{\Theta} = 10$. Fett markiert sind die Zeilen mit den Fehlklassifikationen.

Istklasse → Sollklasse ↓	Vögel	Flaschen	Fische	unbekannt	Ausbeute
Vögel	16	-	-	-	$\frac{16}{16} = 100.0\%$
Flaschen	1	11	2	8	$\frac{11}{22} = 50.0\%$
Fische	5	-	31	-	$\frac{31}{36} = 86.1\%$
Reinheit	$\frac{16}{22} = 72.7\%$	$\frac{11}{11} = 100.0\%$	$\frac{31}{33} = 93.9\%$	X	TQ: $\frac{58}{74} = 78.4\%$

Tabelle 5.26: Tabelle mit Klassifikationsergebnissen für die Fische, Flaschen und Vögel der Teststichprobe in Abbildung 5.35 für $\tilde{\Theta} = 10$.

Testkontur	Istabschnitt	Distanz	Konfidenz	Testkontur	Istabschnitt	Distanz	Konfidenz
vog11	vog5 3. 0.90	25.6	< 111	fis11	fis4 4. 0.80	19.5	< 136
vog12	vog4 1. 0.90	190.0	< 334	fis12	fis4 4. 0.80	91.3	< 136
vog13	vog4 1. 0.90	49.1	< 334	fis13	fis5 5. 0.80	120.6	< 121
vog14	vog4 1. 0.90	75.0	< 334	fis14	fis4 4. 0.80	121.9	< 136
vog15	vog5 3. 0.90	77.1	< 111	fis15	fis2 6. 0.80	97.9	< 115
vog16	vog4 1. 0.90	88.4	< 334	fis16	fis1 28. 0.50	25.3	< 27
vog17	vog4 1. 0.90	201.1	< 334	fis17	fis5 5. 0.80	45.0	< 121
vog18	vog4 1. 0.90	130.9	< 334	fis18	fis4 4. 0.80	94.5	< 136
vog19	vog4 1. 0.90	64.4	< 334	fis19	fis4 4. 0.80	29.0	< 136
vog20	vog4 1. 0.90	47.4	< 334	fis20	fis5 5. 0.80	94.5	< 121
vog21	vog4 1. 0.90	222.5	< 334	fis21	fis4 4. 0.80	96.0	< 136
vog22	vog4 1. 0.90	187.7	< 334	fis22	fis5 5. 0.80	78.4	< 121
vog23	vog4 1. 0.90	289.8	< 334	fis23	fis4 4. 0.80	113.1	< 136
vog24	vog4 1. 0.90	185.3	< 334	fis24	fis4 4. 0.80	92.8	< 136
vog25	vog5 3. 0.90	88.7	< 111	fis25	fis4 4. 0.80	50.7	< 136
vog26	vog5 3. 0.90	82.9	< 111	fis26	fis4 4. 0.80	102.9	< 136
fla11	fla3 9. 0.80	9.9	< 13	fis27	fis4 4. 0.80	88.8	< 136
fla12	fla3 9. 0.80	0.0	< 13	fis28	fis5 5. 0.80	45.8	< 121
fla13	fla4 44. 0.3	0.0	< 2	fis29	fis4 4. 0.80	64.4	< 136
fla14	fla6 24. 0.60	8.1	< 10	fis30	fis4 4. 0.80	77.1	< 136
fla15	fla10 17. 0.70	1.1	< 3	fis31	vog5 3. 0.90	99.9	< 111
fla16	fla5 39 0.40	0.7	< 1	fis32	fis4 4. 0.80	63.2	< 136
fla17	fla6 24. 0.60	4.5	< 10	fis33	fis6 7. 0.80	38.0	< 101
fla18	fla6 24. 0.60	8.9	< 10	fis34	fis4 4. 0.80	84.1	< 136
fla19	fla10 17. 0.70	1.4	< 3	fis35	fis4 4. 0.80	48.0	< 136
fla20	fla3 9. 0.8	11.7	< 13	fis36	fis4 4. 0.80	33.2	< 136
fla21	-	-	-	fis37	fis5 5. 0.80	119.8	< 121
fla22	fla6 24. 0.6	8.7	< 10	fis38	fis6 7. 0.80	88.5	< 101
fla23	fla6 24. 0.60	6.1	< 10	fis39	fis5 5. 0.80	106.9	< 121
fla24	-	-	-	fis40	fis4 4. 0.80	109.3	< 136
fla25	fla5 39. 0.40	0.8	< 1	fis41	fis4 4. 0.80	98.9	< 136
fla26	fla3 9. 0.80	6.3	< 13	fis42	fis4 4. 0.80	49.1	< 136
fla27	fla3 9. 0.80	6.1	< 13	fis43	vog5 3. 0.90	92.0	< 111
fla28	fla6 24. 0.60	3.7	< 10	fis44	vog4 1. 0.90	333.0	< 334
fla29	fla3 9. 0.80	12.7	< 13	fis45	vog5 3. 0.90	108.7	< 111
fla30	fla5 39 0.4	0.9	< 1	fis46	vog5 3. 0.90	109.1	< 111
fla 31	fis1 28. 0.50	19.8	< 27				
fla 32	fla5 39 0.4	0.6	< 1				

Tabelle 5.27: Tabelle mit den detaillierten Klassifikationsergebnissen für die Vögel, Flaschen und Fische in der Teststichprobe in Abbildung 5.35 für $\hat{\Theta} = 13$. Fett markiert sind die Zeilen mit den Fehlklassifikationen.

Istklasse → Sollklasse ↓	Vögel	Flaschen	Fische	unbekannt	Ausbeute
Vögel	16	-	-	-	$\frac{16}{16} = 100.0\%$
Flaschen	-	19	1	2	$\frac{19}{22} = 86.4\%$
Fische	5	-	31	-	$\frac{31}{36} = 86.1\%$
Reinheit	$\frac{16}{21} = 76.2\%$	$\frac{19}{19} = 100.0\%$	$\frac{31}{32} = 96.9\%$	X	TQ: $\frac{66}{74} = 89.2\%$

Tabelle 5.28: Tabelle mit Klassifikationsergebnissen für die Fische, Flaschen und Vögel der Teststichprobe in Abbildung 5.35 für $\hat{\Theta} = 13$.

Testkontur	Istabschnitt	Distanz	Konfidenz	Testkontur	Istabschnitt	Distanz	Konfidenz
vog11	vog3 3. 0.90	250.7	< 417	fis11	fis9 1. 1.00	50.4	< 271
vog12	vog3 3. 0.90	39.5	< 417	fis12	fis9 1. 1.00	86.4	< 271
vog13	vog3 3. 0.90	163.6	< 417	fis13	fis9 1. 1.00	176.3	< 271
vog14	vog3 3. 0.90	184.7	< 417	fis14	fis9 1. 1.00	88.6	< 271
vog15	vog3 3. 0.90	262.5	< 417	fis15	fis9 1. 1.00	183.6	< 271
vog16	vog3 3. 0.90	86.9	< 417	fis16	fis9 1. 1.00	138.0	< 271
vog17	vog3 3. 0.90	300.8	< 417	fis17	fis9 1. 1.00	167.1	< 271
vog18	fla2 2. 1.00	160.3	< 178	fis18	fis9 1. 1.00	90.9	< 271
vog19	vog3 3. 0.90	166.4	< 417	fis19	fis9 1. 1.00	155.3	< 271
vog20	vog3 3. 0.90	174.4	< 417	fis20	fis9 1. 1.00	181.1	< 271
vog21	vog3 3. 0.90	38.0	< 417	fis21	fis9 1. 1.00	133.9	< 271
vog22	vog3 3. 0.90	104.0	< 417	fis22	fis9 1. 1.00	188.6	< 271
vog23	vog3 3. 0.90	335.1	< 417	fis23	fis9 1. 1.00	83.1	< 271
vog24	vog3 3. 0.90	99.7	< 417	fis24	fis9 1. 1.00	83.7	< 271
vog25	vog3 3. 0.90	317.9	< 417	fis25	fis9 1. 1.00	132.9	< 271
vog26	vog3 3. 0.90	240.3	< 418	fis26	fis9 1. 1.00	96.4	< 271
fla11	fla2 2. 1.00	0.0	< 178	fis27	fis9 1. 1.00	142.6	< 271
fla12	fla2 2. 1.00	65.5	< 178	fis28	fis9 1. 1.00	210.0	< 271
fla13	fla2 2. 1.00	49.4	< 178	fis29	fis9 1. 1.00	164.1	< 271
fla14	fla2 2. 1.00	42.1	< 178	fis30	fis9 1. 1.00	107.5	< 271
fla15	fla2 2. 1.00	45.0	< 178	fis31	fis9 1. 1.00	85.3	< 271
fla16	fla2 2. 1.00	70.9	< 178	fis32	fis9 1. 1.00	200.2	< 271
fla17	fla2 2. 1.00	31.4	< 178	fis33	fis9 1. 1.00	124.3	< 271
fla18	fla2 2. 1.00	76.8	< 178	fis34	fis9 1. 1.00	161.1	< 271
fla19	fla2 2. 1.00	24.2	< 178	fis35	fla2 2. 1.00	172.7	< 178
fla20	fla2 2. 1.00	40.5	< 178	fis36	fis9 1. 1.00	162.8	< 271
fla21	fla2 1. 1.00	39.9	< 178	fis37	fis9 1. 1.00	241.0	< 271
fla22	fla2 2. 1.00	36.2	< 178	fis38	fis9 1. 1.00	140.0	< 271
fla23	fla2 2. 1.00	41.1	< 178	fis39	fis9 1. 1.00	192.5	< 271
fla24	fla1 14. 0.8	2.6	6	fis40	fis9 1. 1.00	134.8	< 271
fla25	fla2 2. 1.00	25.2	< 178	fis41	fis9 1. 1.00	53.8	< 271
fla26	fla2 2. 1.00	37.0	< 178	fis42	fis9 1. 1.00	104.0	< 271
fla27	fla2 2. 1.00	110.1	< 178	fis43	fis9 1. 1.00	71.4	< 271
fla28	fla2 2. 1.00	38.3	< 178	fis44	fis9 1. 1.00	122.8	< 271
fla29	fla2 2. 1.00	99.7	< 178	fis45	fla2 2. 1.00	172.7	< 178
fla30	fla2 2. 1.00	16.4	< 178	fis46	vog3 3. 0.90	377.8	< 417
fla 31	fla2 2. 1.00	64.3	< 178				
fla 32	fla2 2. 1.00	15.8	< 178				

Tabelle 5.29: Tabelle mit den detaillierten Klassifikationsergebnissen für die Vögel, Flaschen und Fische in der Teststichprobe in Abbildung 5.35 für $\tilde{\Theta} = 15$. Fett markiert sind die Zeilen mit den Fehlklassifikationen.

Istklasse → Sollklasse ↓	Vögel	Flaschen	Fische	unbekannt	Ausbeute
Vögel	15	1	-	-	$\frac{15}{16} = 93.8\%$
Flaschen	-	22	-	-	$\frac{22}{22} = 100.0\%$
Fische	1	2	33	-	$\frac{33}{36} = 91.7\%$
Reinheit	$\frac{15}{16} = 93.8\%$	$\frac{22}{25} = 88.0\%$	$\frac{33}{33} = 100.0\%$	X	TQ: $\frac{70}{74} = 94.6\%$

Tabelle 5.30: Tabelle mit Klassifikationsergebnissen für die Fische, Flaschen und Vögel der Teststichprobe in Abbildung 5.35 für $\tilde{\Theta} = 15$.

schen Klassifikators.

Das Klassifikationsergebnis wird mit steigendem $\tilde{\Theta}$ hauptsächlich dadurch verbessert, dass die Flaschen statt in die unbekannte Klasse mehr und mehr richtig klassifiziert werden. Der Gegeneffekt ist allerdings, dass jetzt bestimmte Vögel und Fische für eine Flasche gehalten werden, da die lokalen Feinheiten der Konturen jetzt immer unwichtiger werden. Die Fehlklassifikationen sind für die einzelnen Importanzschwellen $\tilde{\Theta} = 10$, $\tilde{\Theta} = 13$ und $\tilde{\Theta} = 15$ in den Abbildungen 5.36, 5.37 und 5.38 gezeigt.

Die Verbesserung der Gesamttrefferquote bei steigendem $\tilde{\Theta}$ ist außerdem ein Indiz für folgende Annahme: zieht ein Klassifikator gleichzeitig mehrere kurze signifikante Konturabschnitte für die Klassifikation heran, die bei größeren $\tilde{\Theta}$ in nur einem einzigen signifikanten Konturabschnitt zusammengefasst sind, so wird ein ähnlich gutes Klassifikationsergebnis erzielt, ohne bestimmte Fehlklassifikationen zu haben.

Insgesamt lassen sich aber auch schon erstaunlich gute Klassifikationsergebnisse mit jeweils nur einem einzigen signifikanten Konturabschnitt pro Hierarchieebene des Klassifikators erzielen.

Weiterhin ist interessant, dass sich bei näherer Untersuchung der Tabellen 5.25, 5.27 und 5.29 herausstellt, dass nur eine Teilmenge der signifikanten Konturabschnitte auf das gleiche Klassifikationsergebnis geführt hätte. Dazu werden all die Abschnitte aus den Istabschnitt-Spalten der Tabellen herausgesucht, die zu richtigen Klassifikationsergebnissen geführt haben. Für $\tilde{\Theta} = 10$ sind dies die 16 Abschnitte: vog1 (14), vog8 (2), vog9 (1), fla2 (7), fla3 (60), fla6 (47), fla8 (23), fla9 (56), fla10 (42), fis1 (6), fis1 (8), fis3 (9), fis7 (16), fis7 (19), fis7 (48), fis9 (30).

Für $\tilde{\Theta} = 13$ sind dies die 12 Abschnitte: vog4 (1), vog5 (3), fla3 (9), fla4 (44), fla5 (39), fla6 (24), fla10 (17), fis1 (28), fis2 (6), fis4 (4), fis5 (5), fis6 (7).

Und für $\tilde{\Theta} = 15$ sind dies nur noch die 4 Abschnitte: vog3 (3), fla1 (14), fla2 (2), fis9 (1).

Dies ist zum einen eine Möglichkeit den hierarchischen Klassifikator noch schneller zu machen. Und zum anderen sind solche Konturabschnitte interessant bzgl. einer weitergehenden Untersuchung auf einige wenige abstrakte Konturabschnitte, die eine Konturklasse ausmachen.

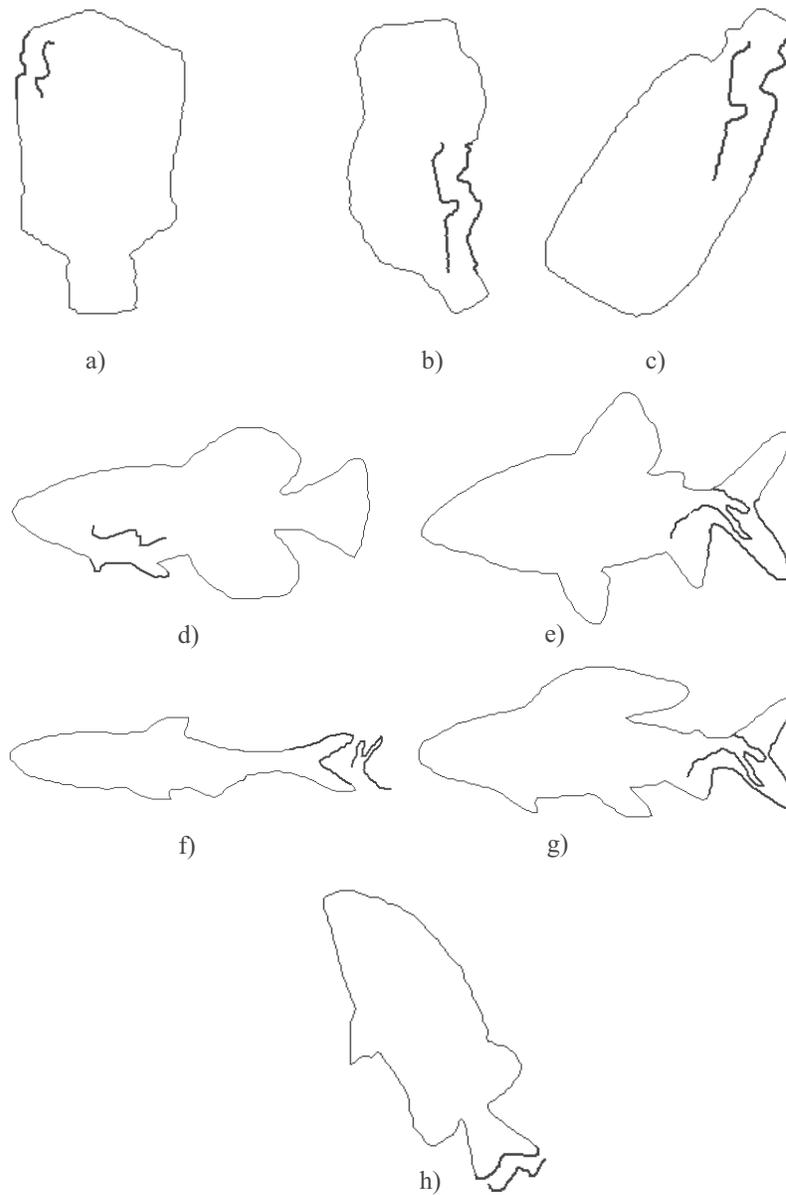


Abbildung 5.36: Fehlklassifikationen von Flaschen und Fischen für $\tilde{\Theta} = 10$. Siehe auch Tabelle 5.25. a) Der fett markierte Konturabschnitt auf Flasche fla26 wird für 15. verlässlichsten Konturabschnitt der Kontur vog4 in der Lernstichprobe gehalten: fla26 \rightarrow vog4 (15), b) fla27 \rightarrow fis1 (6), c) fla31 \rightarrow fis1 (6), d) fis15 \rightarrow vog4 (15), e) fis33 \rightarrow vog7 (4), f) fis36 \rightarrow vog7 (4), g) fis44 \rightarrow vog7 (4), h) fis45 \rightarrow vog4 (15).

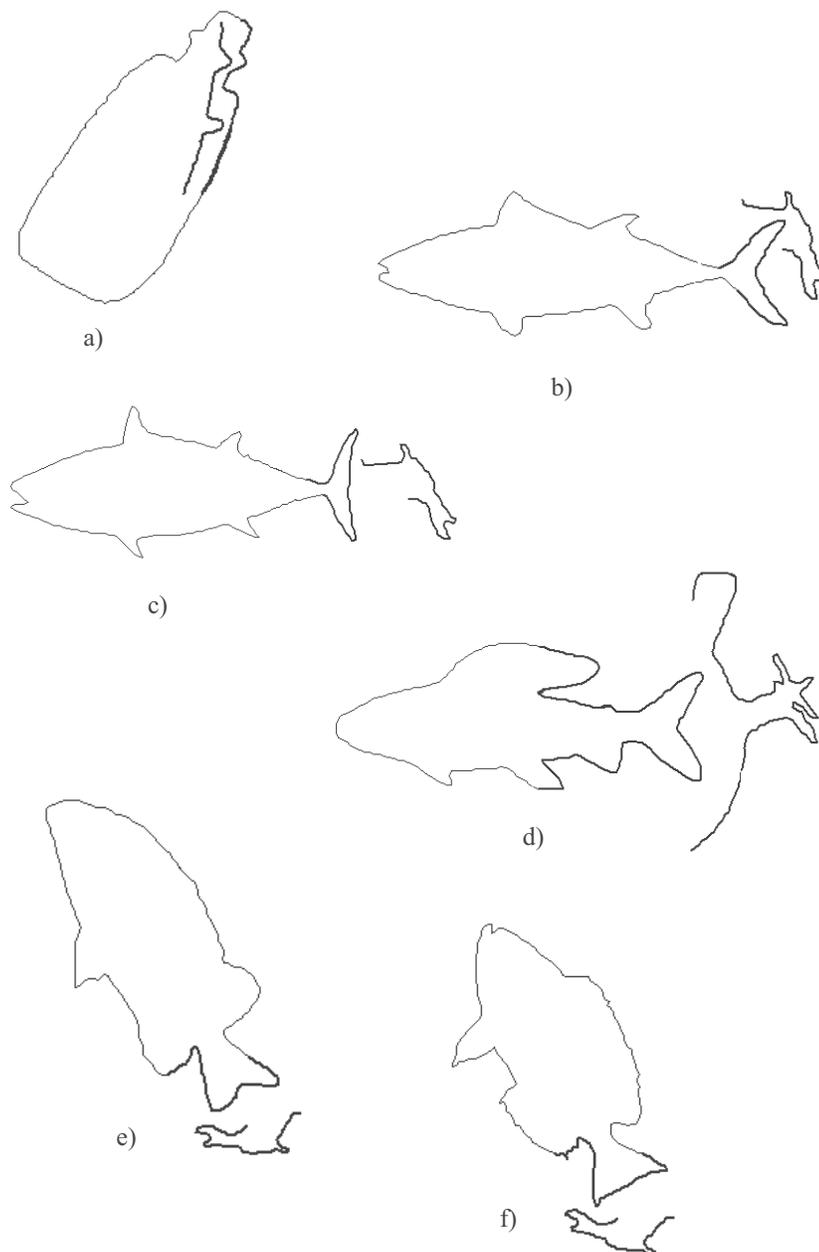


Abbildung 5.37: Fehlklassifikationen von Flaschen und Fischen für $\tilde{\Theta} = 13$. Siehe auch Tabelle 5.27. a) Der fett markierte Konturabschnitt auf Flasche fla31 wird für 28. verlässlichsten Konturabschnitt der Kontur fis1 in der Lernstichprobe gehalten: fla31 \rightarrow fis1 (28), b) fis31 \rightarrow vog5 (3), c) fis43 \rightarrow vog5 (3), d) fis44 \rightarrow vog4 (1), e) fis45 \rightarrow vog5 (3), f) fis46 \rightarrow vog5 (3).

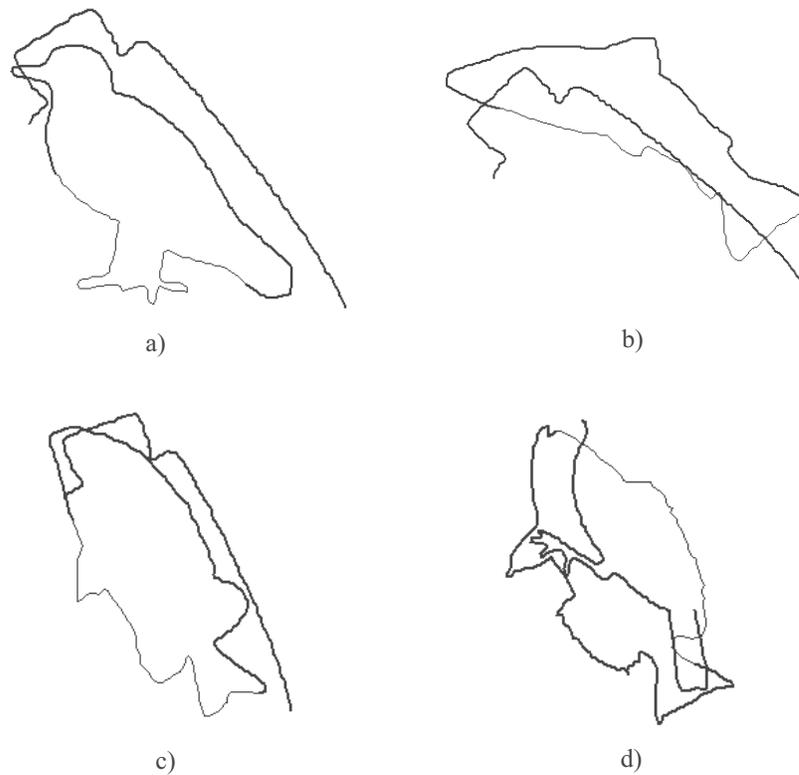


Abbildung 5.38: Fehlklassifikationen von Flaschen und Fischen für $\tilde{\Theta} = 15$. Siehe auch Tabelle 5.29. a) Der fett markierte Konturabschnitt auf Vogel vog18 wird für 2. verlässlichsten Konturabschnitt der Kontur fla2 in der Lernstichprobe gehalten: vog18 → fla2 (2), b) fis35 → fla2 (2), c) fis45 → fla2 (2), d) fis46 → vog3 (3).

E / l_{const}	$\tilde{\Theta} / \Theta$	L_{min}	Trefferquote
$E = 200.0$	≤ 10	10	$\frac{17}{23} = 73.9\%$
$E = 200.0$	≤ 13	10	$\frac{19}{23} = 82.6\%$
$E = 200.0$	≤ 15	10	$\frac{18}{23} = 78.3\%$
$E = 1000.0$	≤ 15	20	$\frac{14}{23} = 60.9\%$
$E = 1000.0$	≤ 18	20	$\frac{17}{23} = 73.9\%$
$E = 1000.0$	≤ 20	20	$\frac{17}{23} = 73.9\%$
$l_{const} = 10$	≥ 750.0	10	$\frac{8}{23} = 34.8\%$
$l_{const} = 10$	≥ 450.0	10	$\frac{13}{23} = 56.5\%$
$l_{const} = 10$	≥ 150.0	10	$\frac{18}{23} = 78.3\%$
$l_{const} = 20$	≥ 5000.0	20	$\frac{16}{23} = 69.6\%$
$l_{const} = 20$	≥ 3500.0	20	$\frac{17}{23} = 73.9\%$
$l_{const} = 20$	≥ 2000.0	20	$\frac{17}{23} = 73.9\%$

Tabelle 5.31: Tabelle mit Trefferquoten des hierarchischen Klassifikators für den Datensatz von Kimia der Teststichprobe in Abbildung 5.43 für verschiedene Parameterkombinationen.

5.5.3 Datensatz von Kimia

Der Datensatz der in diesem Abschnitt verwendeten Lernstichprobe von Kimia ist in Abbildung 5.21 gezeigt. Nur die Hand ha5 wurde wie schon erwähnt aus der Lernstichprobe in die Teststichprobe verschoben, um dort eine größere Anzahl an Händen zu erhalten. Die Lernstichprobe enthält jetzt also jeweils 4 Konturen in 6 Klassen.

Für diesen Datensatz wurden für die gleichen Parameterkombinationen wie für den Datensatz *Vögel, Flaschen und Fische* die Trefferquoten bestimmt. Diese sind in Tabelle 5.31 zusammengefasst.

Es stellt sich heraus, dass auch bei diesem Datensatz das beste Ergebnis mit einer Trefferquote von $\frac{19}{23} = 82.6\%$ für die Parametereinstellung $E = 200.0$ erzielt wird. Dies Ergebnis ist quantitativ schlechter als das beste Ergebnis bei den *Vögel, Flaschen und Fische*, doch ist es sicherlich auch schwieriger 6 Klassen mit Hilfe nur eines signifikanten Konturabschnittes auseinanderzuhalten als nur 3 Klassen. Zusätzlich erschwert wird die Klassifikationsaufgabe dadurch, dass in der Lernstichprobe des Datensatzes von Kimia in jeder Klasse lediglich 4 Konturen sind, was sicherlich eine extrem niedrige Zahl für die Repräsentativität einer Klasse darstellt, wohingegen sich im Datensatz *Vögel, Flaschen und Fische* jeweils 10 Konturen pro Klasse befinden. Aus diesem Grund ist das Klassifikationsergebnis ähnlich gut einzuschätzen wie das Ergebnis für die *Vögel, Flaschen und Fische*.

Im einzelnen seien hier die Ergebnisse für die Parameterkombinationen

$E = 200.0$ und $\tilde{\Theta} = 10$ bzw. $\tilde{\Theta} = 13$ näher vorgestellt.

Die Abbildungen 5.39, 5.40 und 5.41, 5.42 zeigen die signifikanten bzw. die nach Reliabilität geordneten signifikanteren Konturabschnitte der Konturen in der Lernstichprobe.

Die Teststichprobe zeigt Abbildung 5.43.

Die detaillierten Klassifikationsergebnisse und die Vertauschungsmatrizen für die Klassen basierend auf dieser Teststichprobe sind in den Tabellen 5.32, 5.33 bzw. 5.34, 5.35 zusammengefasst.

Die Fehlklassifikationen für die beiden verschiedenen Importanzschwellen $\tilde{\Theta}$ sind in den Abbildungen 5.44 und 5.45 dargestellt.

Die Fehlklassifikationen sind auch hier nachvollziehbar und die Distanzwerte befinden sich schon sehr nah an den Konfidenzen der einzelnen signifikanten Abschnitte der Wissensbasis, so dass darin die Unsicherheit dieser Klassifikationen zum Ausdruck kommt. Interessantes Ergebnis am Rande ist die verblüffende Ähnlichkeit eines Flugzeugabschnittes mit einem Kaninchenabschnitt in Abbildung 5.45c.

Auch bei diesem Datensatz lässt sich wie beim Datensatz *Vögel, Flaschen und Fische* feststellen, dass nur eine Untermenge der signifikanten Konturabschnitte notwendig ist, um die gleiche Trefferquote zu erzielen und gleichzeitig auch noch Fehlklassifikationen zu vermeiden. Dies sind für den Importanzschwellwert $\tilde{\Theta} = 10$ die 9 signifikanten Konturabschnitte fi1 (7), gr2 (14), gr3 (15), ha2 (2), fi2 (1), ka3 (5), ka4 (8), ma1 (26) und ma3 (16), sowie für den Importanzschwellwert $\tilde{\Theta} = 13$ die 7 signifikanten Konturabschnitte fi1 (15), gr4 (14), ha3 (3), ha4 (31), fi1 (1), ka1 (9) und ma1 (13). D.h. die Erkenntnisse bzgl. der Beschleunigung des Klassifikators und der Möglichkeit der Erlangung einiger weniger abstrakter Konturabschnitte für die einzelnen Klassen werden an dieser Stelle bestätigt.

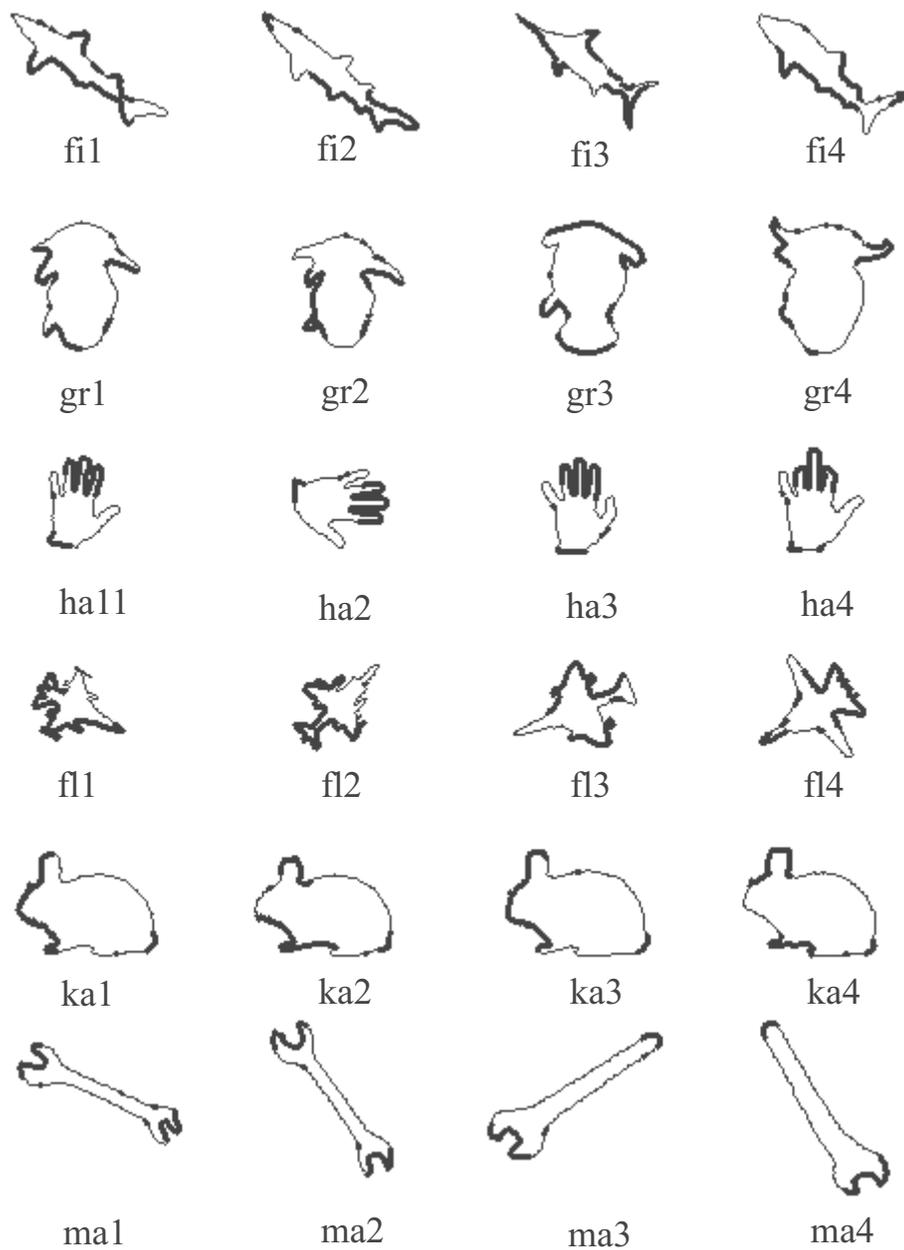
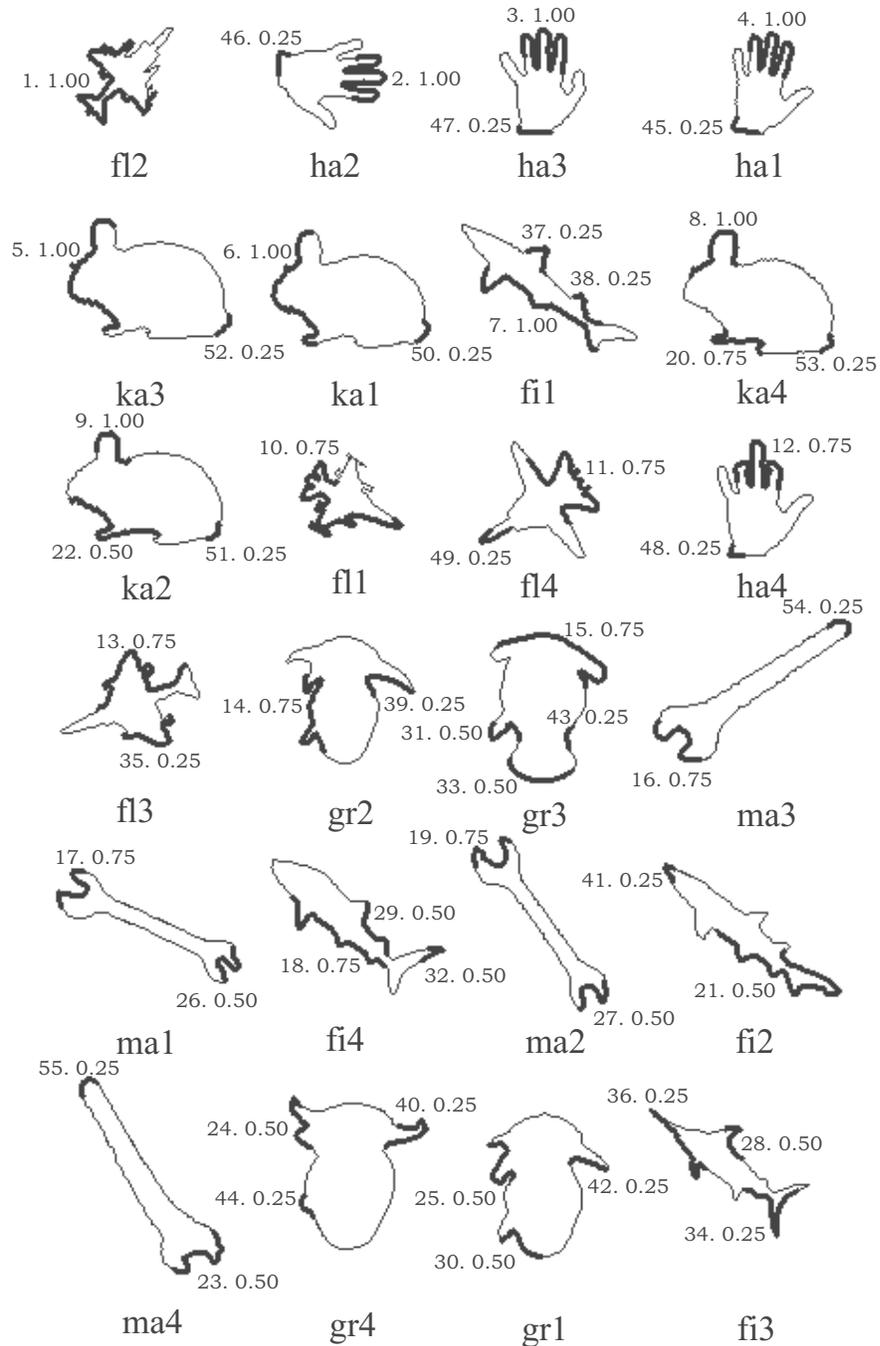


Abbildung 5.39: Signifikante Konturabschnitte von den Konturen der Lernstichprobe aus dem Datensatz von Kimia [LEM]. Methode: lokale Ähnlichkeitsmatrizen, $E = 200.0$, Importanzschwelle $\tilde{\Theta} = 10$.



Abbildungung 5.40: Abgebildet sind die signifikanteren Konturabschnitte der Lernstichprobe aus dem Datensatz von Kimia [LEM] für eine Importanzschwelle von $\tilde{\Theta} = 10$ und eine Mindestkonturpunktanzahl von $L_{min} = 10$. Die Konturen sind gemäß der Reliabilität ihrer fett markierten signifikanteren Konturabschnitte geordnet. Neben den Konturabschnitten ist ihre Reliabilitätsplatzierung innerhalb der Lernstichprobe und ihre Reliabilität notiert.

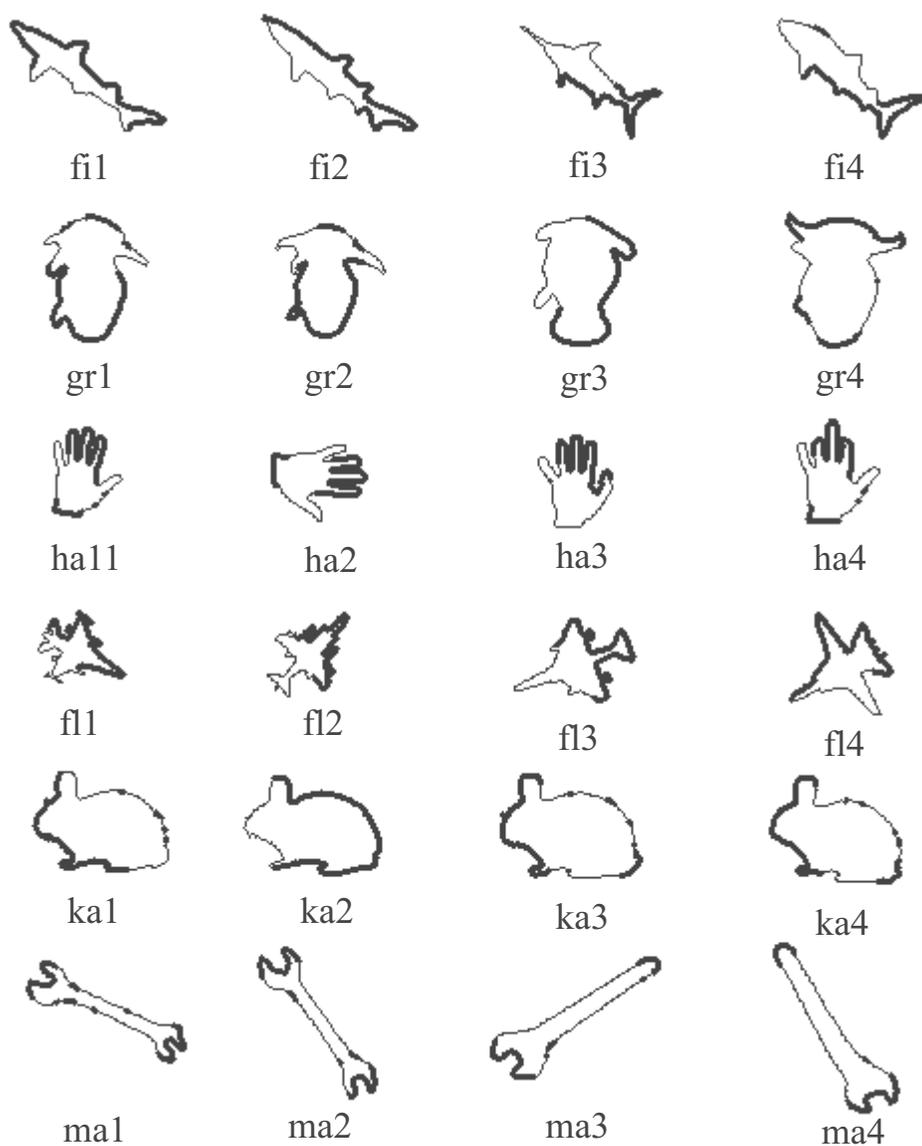


Abbildung 5.41: Signifikante Konturabschnitte von den Konturen der Lernstichprobe aus dem Datensatz von Kimia [LEM]. Methode: lokale Ähnlichkeitsmatrizen, $E = 200.0$, Importanzschwelle $\tilde{\Theta} = 13$.

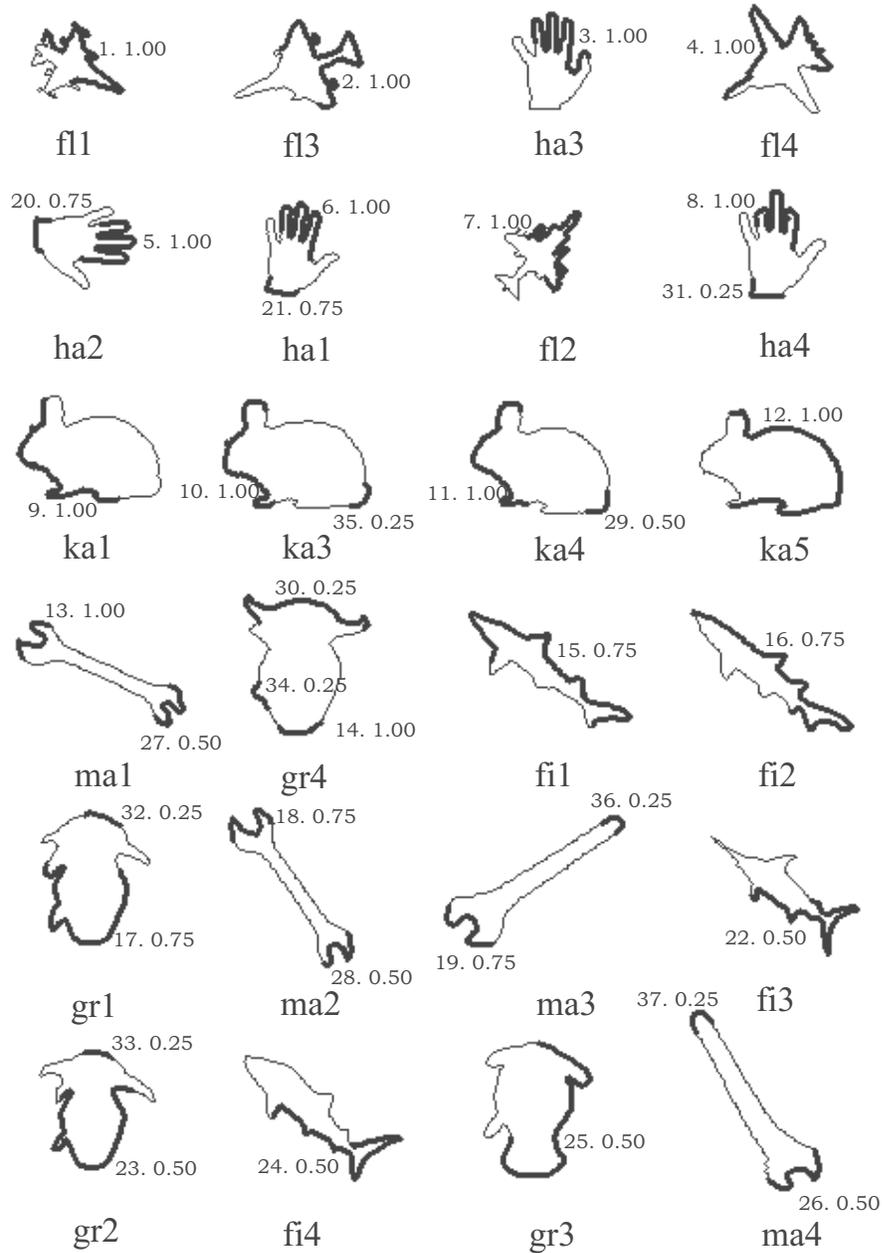


Abbildung 5.42: Abgebildet sind die signifikanteren Konturabschnitte der Lernstichprobe aus dem Datensatz von Kimia [LEM] für eine Importanzschwelle von $\tilde{\Theta} = 13$ und eine Mindestkonturpunktanzahl von $L_{min} = 10$. Die Konturen sind gemäß der Reliabilität ihrer fett markierten signifikanteren Konturabschnitte geordnet. Neben den Konturabschnitten ist ihre Reliabilitätsplatzierung innerhalb der Lernstichprobe und ihre Reliabilität notiert.

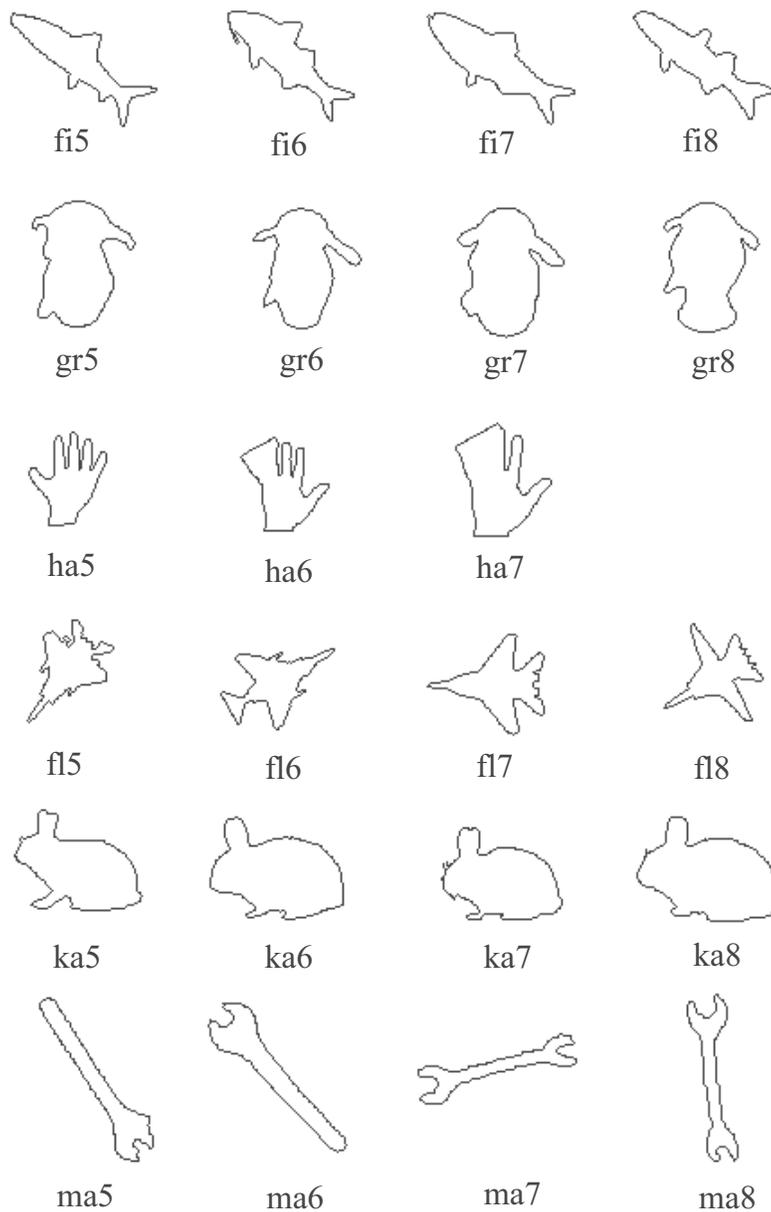


Abbildung 5.43: Teststichprobe aus dem Datensatz von Kimia [LEM] bestehend aus 4 Fischen fi5-fi8, 4 Greebles gr5-gr8, 3 Händen ha5-ha7, 4 Flugzeugen fl5-fl8, 4 Kaninchen ka5-ka8 und 4 Maulschlüssel ma5-ma8.

Testkontur	Istabschnitt	Distanz	Konfidenz
fi5	ma4 23. 0.50	21.4	< 22
fi6	fi2 1. 1.00	280.1	< 297
fi7	fi1 7. 1.00	46.1	< 66
fi8	fi1 7. 1.00	29.2	< 66
gr5	gr3 15. 0.75	24.9	< 56
gr6	ka3 5. 1.00	74.1	< 75
gr7	gr3 15. 0.75	47.9	< 56
gr8	gr2 14. 0.75	51.3	< 65
ha5	ha2 2. 1.00	18.4	< 205
ha6	ha2 2. 1.00	109.5	< 205
ha7	-	-	-
fl5	fi2 1. 1.00	188.4	< 297
fl6	fi2 1. 1.00	97.5	< 297
fl7	fi2 1. 1.00	265.2	< 297
fl8	fi2 1. 1.00	211.5	< 297
ka5	ka3 5. 1.00	22.4	< 75
ka6	ka3 5. 1.00	10.7	< 75
ka7	ka3 5. 1.00	62.1	< 75
ka8	ka4 8. 1.00	4.7	< 13
ma5	ma1 26. 0.50	5.0	< 7
ma6	ma3 16. 0.75	9.9	< 25
ma7	fi1 7. 1.00	61.7	< 66
ma8	fi1 7. 1.00	57.3	< 66

Tabelle 5.32: Tabelle mit den detaillierten Klassifikationsergebnissen für die Konturen aus dem Datensatz von Kimia [LEM] der Teststichprobe in Abbildung 5.43 für $\tilde{\Theta} = 10$. Fett markiert sind die Zeilen mit den Fehlklassifikationen.

Istklasse → Sollklasse ↓	Fische	Greebles	Hände	Flugzeuge	Kaninchen	Maulschlüssel	unbekannt	Ausbeute
Fische	2	-	-	1	-	1	-	50.0%
Greebles	-	3	-	-	1	-	-	75.0%
Hände	-	-	2	-	-	-	1	66.7%
Flugzeuge	-	-	-	4	-	-	-	100.0%
Kaninchen	-	-	-	-	4	-	-	100.0%
Maulschlüssel	2	-	-	-	-	2	-	50.0%
Reinheit	50.0%	100.0%	100.0%	80.0%	80.0%	66.7%	X	TQ: 73.9%

Tabelle 5.33: Tabelle mit Klassifikationsergebnissen für den Datensatz von Kimia [LEM] in der Teststichprobe in Abbildung 5.43 für $\tilde{\Theta} = 10$.

Testkontur	Istabschnitt	Distanz	Konfidenz
fi5	fi1 15. 0.75	150.1	< 199
fi6	fi1 15. 0.75	166.9	< 199
fi7	fi1 15. 0.75	135.5	< 199
fi8	fi1 15. 0.75	103.4	< 199
gr5	gr4 14. 1.00	0.35	< 1
gr6	gr4 14. 1.00	0.13	< 1
gr7	ka3 10. 1.00	109.4	< 115
gr8	ma1 13. 1.00	30.5	< 35
ha5	ha3 3. 1.00	101.9	< 230
ha6	ha3 3. 1.00	137.8	< 230
ha7	ha4 31. 0.25	0.1	< 1
fl5	fl1 1. 1.00	174.6	< 261
fl6	fl1 1. 1.00	165.9	< 261
fl7	fl1 1. 1.00	189.9	< 261
fl8	fl1 1. 1.00	83.6	< 261
ka5	fl4 4. 1.00	199.7	< 212
ka6	ka1 9. 1.00	28.2	< 122
ka7	fl1 1. 1.00	178.6	< 261
ka8	ka1 9. 1.00	102.8	< 122
ma5	ma1 13. 1.00	19.9	< 35
ma6	ma1 13. 1.00	4.1	< 35
ma7	ma1 13. 1.00	5.1	< 35
ma8	ma1 13. 1.00	11.1	< 35

Tabelle 5.34: Tabelle mit den detaillierten Klassifikationsergebnissen für die Konturen aus dem Datensatz von Kimia [LEM] der Teststichprobe in Abbildung 5.43 für $\tilde{\Theta} = 13$. Fett markiert sind die Zeilen mit den Fehlklassifikationen.

Istklasse → Sollklasse ↓	Fische	Greebles	Hände	Flugzeuge	Kaninchen	Maulschlüssel	unbekannt	Ausbeute
Fische	4	-	-	-	-	-	-	100.0%
Greebles	-	2	-	-	1	1	-	50.0%
Hände	-	-	3	-	-	-	-	100.0%
Flugzeuge	-	-	-	4	-	-	-	100.0%
Kaninchen	-	-	-	2	2	-	-	50.0%
Maulschlüssel	-	-	-	-	-	4	-	100.0%
Reinheit	100.0%	100.0%	100.0%	66.7%	66.7%	80.0%	X	TQ: 82.6%

Tabelle 5.35: Tabelle mit Klassifikationsergebnissen für den Datensatz von Kimia [LEM] in der Teststichprobe in Abbildung 5.43 für $\tilde{\Theta} = 13$.

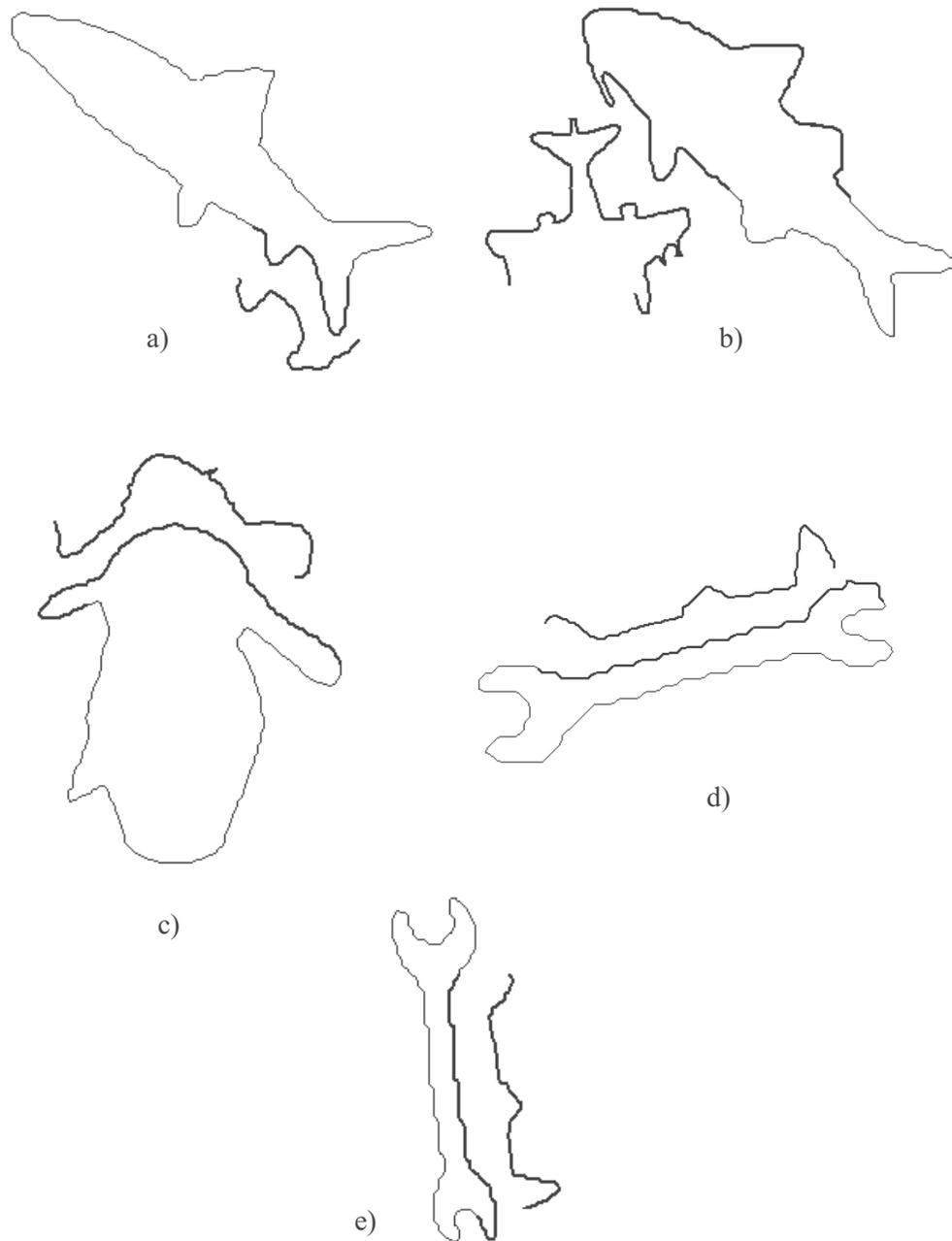


Abbildung 5.44: Fehlklassifikation von Fischen, einem Greeble und Mausschlüsseln für $\tilde{\Theta} = 10$ (siehe auch Tabelle 5.32). a) Der fett markierte Konturabschnitt auf Fisch fi5 wird für den 23. verlässlichsten Konturabschnitt des Mausschlüssels ma4 gehalten: fi5 \rightarrow ma4 (23); b) fi6 \rightarrow fi2 (1); c) gr6 \rightarrow ka3 (5); d) ma7 \rightarrow fi1 (7); e) ma8 \rightarrow fi1 (7).

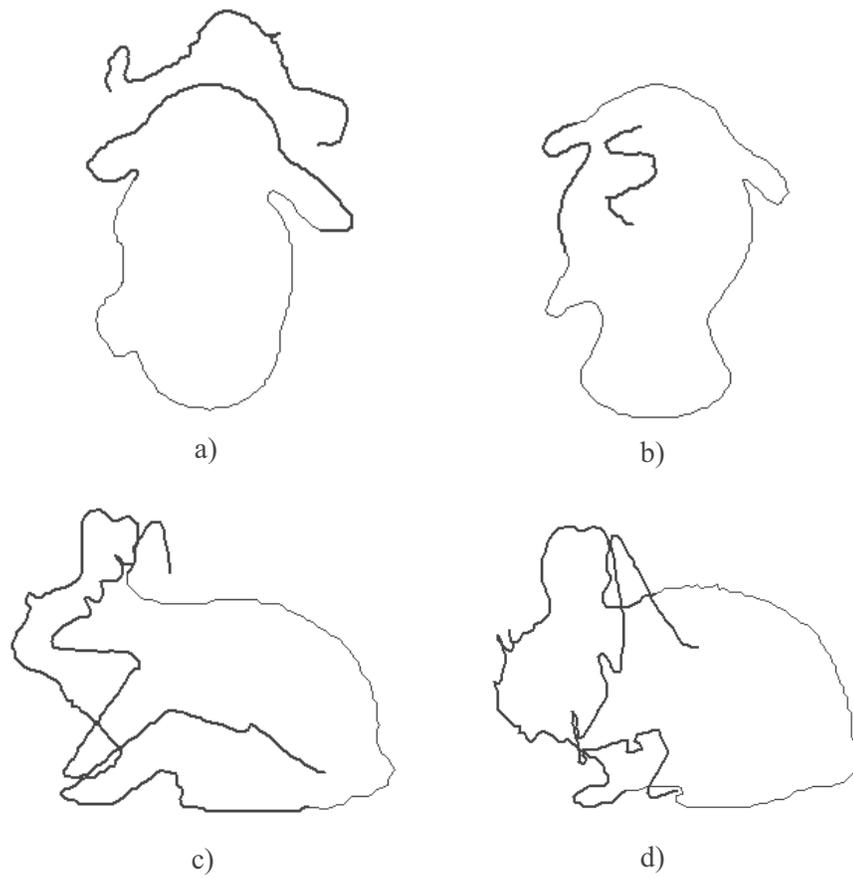


Abbildung 5.45: Fehlklassifikation von Greebles und Kaninchen für $\tilde{\Theta} = 13$ (siehe auch Tabelle 5.34). a) Der fett markierte Konturabschnitt auf Fisch gr7 wird für den 10. verlässlichsten Konturabschnitt des Kaninchens ka3 gehalten: gr7 → ka3 (10); b) gr8 → ma1 (13); c) ka5 → fl4 (4); d) ka7 → fl1 (1).

5.6 Autonome Modellbildung

In [PK00c] stellen Pechtel und Kuhnert einen Ansatz vor, wie ein autonomes (mobiles) System seine Umwelt erkundet und mit Hilfe der globalen Distanzen aus Abschnitt 4.3.2 selbständig ein Modell seiner Umwelt erstellen kann. Das intelligente System bewegt sich dazu in seiner Umwelt und bildet die darin befindlichen Objekte mit einer Kamera ab. Dabei werden auf der einen Seite unbekannte Objekte erlernt und auf der anderen Seite bereits bekannte Objekte wiedererkannt, so dass das intelligente System daraus für sich Handlungen ableiten kann (\rightarrow *Active Vision*).

Dieser Abschnitt fasst die experimentellen Ergebnisse zusammen, wie ein intelligentes System autonom die \rightarrow *Topologie* bzw. Geometrie von Objekten erlernt, indem es Bilder dieser Objekte aus verschiedenen Blickwinkeln aufnimmt, ohne die absoluten oder relativen Positionen seines Sehsystems zu kennen. Mit diesen Bildern schafft sich das System selbständig ein Modell seiner Umwelt. Zur Vereinfachung dieser Aufgabenstellung werden ausschließlich die Außenkonturen von einzelnen isolierten Objekten betrachtet.

Als Eingangsinformation erhält das System eine Reihe von Bildern, die willkürlich von verschiedenen Objekten, aus verschiedenen Blickwinkeln, gemacht wurden. Vor dem Erlernen des Modells schaut das System in seiner Umwelt herum und nimmt unsystematisch Bilder auf. Dabei können auch zunächst Abbilder eines Objektes A, danach Abbilder anderer Objekte und dann wieder Abbilder des Objektes A gemacht werden. Die Reihenfolge der Objektaufnahmen und Blickwinkel spielt also keine Rolle. Ferner ist dem System unbekannt, um was für Objekte es sich handelt.

Zwischen allen Außenkonturen der aus verschiedenen Blickwinkeln aufgenommenen Objekte werden die globalen Distanzen mit Hilfe der Algorithmen Ia , IIa , $IIIa$, IVa berechnet. Die ähnlichsten Ansichten eines Objektes, d.h. die Objekte mit kleinster globaler Distanz, sind die topologisch am meisten benachbarten Ansichten und das intelligente System kann die Ansichten mit Hilfe der globalen Distanzen ordnen. Ferner sind dann auch die korrespondierenden Positionen, aus denen die Aufnahmen gemacht wurden, topologisch benachbart.

Mit Hilfe des so selbständig gelernten topologischen Modells der Ansichten der Objekte kann das intelligente System seine eigene Position bestimmen und Aktionen ableiten.

Die drei wichtigsten Ansätzen in der Literatur zum Erlernen eines Umweltmodells sind:

1. In der Umwelt sind einfach erkennbare visuelle Hinweise, sogenannte Landmarken, enthalten (z.B. [VRH⁺93]).
2. Explizite 3D-Modelle der Umwelt werden für das System von einem Trai-

ner konstruiert (z.B. [CRW97]).

3. Bildfolgen der Umwelt werden zunächst gespeichert und dann global und linear transformiert. Es entstehen sogenannte *Panoramic Eigenimages* [JL99].

Der hier vorgestellte Ansatz ist dem letzten Ansatz am ähnlichsten. Er unterscheidet sich aber dadurch, dass er nichtlinear ist, und dass er das Erlernen eines Modells der Umwelt und das Erkennen von Objekten in sich vereinigt.

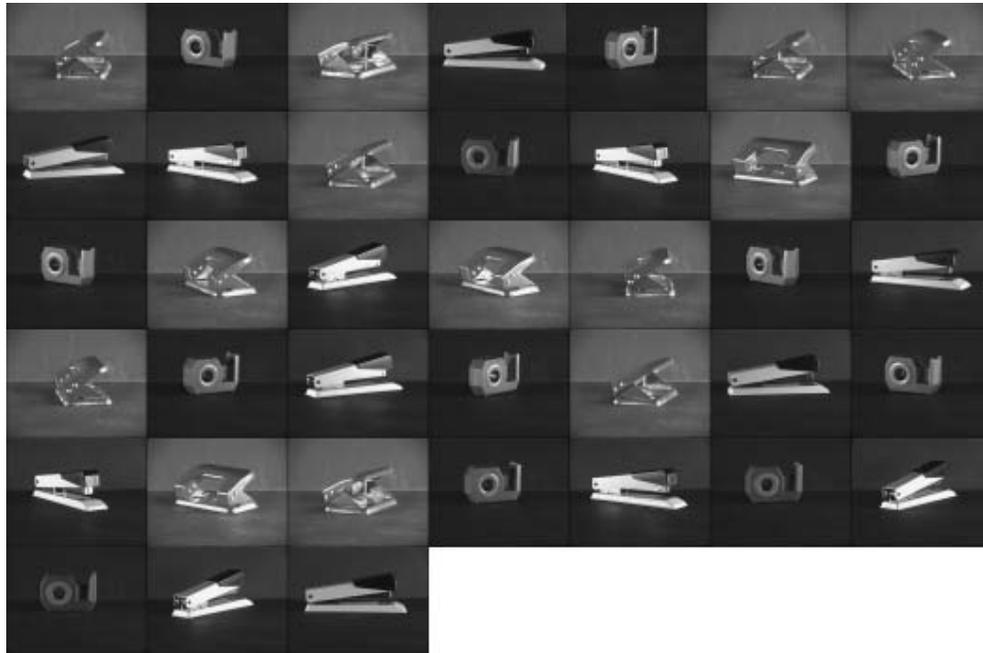
Ein erstes Experiment wurde mit 3 verschiedenen Objekten aus der Büroumgebung gemacht: Einem Hefter H, einem Kleberrollenhalter K und einem Locher L (siehe Abbildung 5.46a). In Abbildung 5.46b sind beispielhaft die Außenkonturen für die Ansichten von 3 Objekten aus verschiedenen Klassen gezeigt. 13 Bilder wurden vom Hefter bzw. dem Locher und 12 Bilder wurden vom Kleberrollenhalter aufgenommen. Die Objekte lagen alle ungefähr in der Mitte auf einem Tisch. Der Kameraabstand zur Mitte des Tisches und zum Fußboden war während der Objektaufnahmen rund um die Objekte herum nur ungefähr konstant. Die Rotationswinkel, bezogen auf die Anfangsposition der Kamera, wurden auch nicht exakt vermessen, sondern nur ungefähr gleichabständig eingestellt. So ergeben sich die Ansichten H1 - H13, K1 - K12 und L1 - L13 in den Abbildungen 5.47, 5.46a.

Aus den Aufnahmebedingungen ergibt sich, dass die Distanz der Objekte von der Kamera und der vertikale Blickwinkel ebenfalls in Grenzen variierten. Diese Ungenauigkeiten wurden absichtlich eingeführt, da komplexe Roboter wie z.B. Gehmaschinen (engl. walking machines) aufgrund von Positionssensor-Ungenauigkeiten auch nicht in der Lage sind, Bilder aus exakten Positionen aufzunehmen.

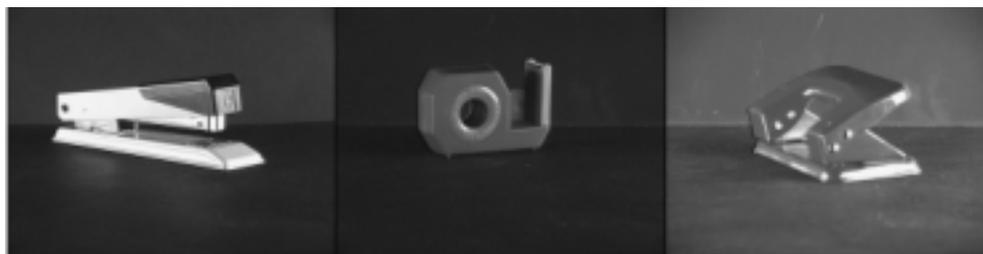
In Abbildung 5.47 sieht man, dass die Methode die Bildsequenzen bzgl. der Topologie der Ansichten ordnen kann. H2 ist topologisch benachbart zu H1 und H3. H3 ist topologisch benachbart zu H2 und H4 usw.. Es soll an dieser Stelle noch einmal herausgestellt werden, dass der Durchführende dieses Experimentes wusste, in welcher Reihenfolge die Aufnahmen der Objektansichten gemacht wurden. **Das intelligente System weiß dies zum Zeitpunkt der Datenaufnahme aber nicht.** Man hätte dem System die Bilder auch in einer komplett anderen Reihenfolge anbieten können und es wäre zum gleichen Ergebnis gekommen.

In Tabelle 5.36 sind in den Spalten 2 - 4 die jeweils 3 ähnlichsten Ansichten mit ihren globalen Distanzwerten, bezogen auf eine bestimmte Ansicht (Spalte 1), eingetragen. In den Spalten 5 und 6 sind die ähnlichsten Ansichten und zugehörigen globalen Distanzwerte von Objekten aus den 2 anderen Objektklassen aufgeführt.

Ein Ergebnis ist, dass es bezogen auf ein bestimmtes Objekt keine Ansicht der beiden anderen Objekte gibt, die ähnlicher ist, als eine Ansicht des Bezugsobjektes selbst. D.h. das intelligente System kann die 3 Objekte auseinanderhalten, also korrekt klassifizieren.



a)



b)

Abbildung 5.46: In Abbildung 5.46a sind die Ansichten der Objekte Hefter, Kleberollenhalter und Locher abgebildet. Die Ansichten der einzelnen Objekte sind durcheinander abgebildet, da hiermit verdeutlicht werden soll, dass das lernende System auch mit einer solchen Situation umgehen kann. Abbildung 5.46b zeigt beispielhaft eine Außenkontur für ein Objekt aus jeder Klasse.

H1	H2	0.06	H3	0.09	H4	0.14	K5	4.76	L9	2.35
H2	H3	0.04	H1	0.06	H4	0.10	K5	5.02	L9	2.48
H3	H4	0.02	H2	0.04	H1	0.09	K5	4.76	L9	2.13
H4	H3	0.02	H2	0.10	H7	0.12	K5	4.77	L9	2.06
H5	H6	0.03	H7	0.12	H4	0.13	K5	4.94	L12	2.16
H6	H5	0.03	H7	0.12	H4	0.14	K5	4.72	L12	1.99
H7	H6	0.06	H8	0.10	H3	0.11	K5	4.88	L11	1.82
H8	H9	0.02	H7	0.10	H10	0.12	K5	4.49	L10	1.83
H9	H8	0.02	H10	0.12	H4	0.22	K5	4.58	L9	2.11
H10	H11	0.07	H12	0.09	H9	0.12	K5	4.36	L12	1.89
H11	H12	0.03	H13	0.06	H10	0.07	K5	4.89	L11	2.21
H12	H13	0.00	H11	0.03	H10	0.09	K5	4.28	L11	2.02
H13	H12	0.00	H11	0.06	H10	0.19	K5	4.19	L12	2.12
K1	K2	0.35	K3	0.93	K12	1.18	H12	4.58	L12	2.41
K2	K3	0.23	K1	0.35	K11	0.70	H13	4.83	L8	2.39
K3	K4	0.18	K2	0.23	K10	0.67	H13	4.73	L8	1.96
K4	K3	0.18	K5	0.48	K8	0.66	H13	5.48	L13	1.68
K5	K7	0.35	K6	0.40	K4	0.48	H13	4.19	L12	1.18
K6	K7	0.32	K5	0.40	K8	0.88	H13	4.58	L13	1.18
K7	K6	0.32	K8	0.34	K5	0.35	H13	5.22	L13	1.09
K8	K9	0.14	K7	0.34	K5	0.57	H13	5.00	L13	1.41
K9	K8	0.14	K10	0.31	K3	0.79	H13	4.62	L13	1.59
K10	K11	0.29	K9	0.31	K3	0.67	H13	4.67	L7	2.29
K11	K10	0.29	K12	0.57	K3	0.83	H13	5.54	L7	2.88
K12	K11	0.57	K2	1.06	K1	1.18	H10	5.44	L13	2.16
L1	L2	0.07	L3	0.08	L4	0.63	H7	3.32	K1	3.79
L2	L1	0.07	L3	0.15	L4	0.32	H7	2.95	K6	3.66
L3	L1	0.08	L2	0.15	L4	0.50	H13	3.54	K1	3.88
L4	L5	0.15	L2	0.32	L3	0.50	H7	2.39	K5	3.27
L5	L4	0.15	L6	0.46	L11	0.50	H6	2.20	K5	2.57
L6	L5	0.46	L12	1.07	L4	1.34	H6	2.88	K9	2.29
L7	L12	0.71	L11	0.80	L10	0.86	H7	2.01	K5	1.78
L8	L9	0.26	L10	0.56	L11	0.83	H8	2.00	K5	1.52
L9	L10	0.04	L8	0.26	L11	0.30	H8	1.96	K5	1.56
L10	L9	0.04	L11	0.09	L12	0.12	H8	1.83	K5	1.58
L11	L12	0.07	L10	0.09	L13	0.43	H7	1.82	K5	1.66
L12	L11	0.07	L13	0.12	L10	0.12	H7	1.82	K5	1.18
L13	L12	0.12	L11	0.43	L10	0.67	H7	2.34	K7	1.09

Tabelle 5.36: Ergebnis der globalen Distanzberechnung, basierend auf absoluter Orientierung für $l_{const}=10$, für die Ansichten des Hefters, des Kleberrollenhalters und des Lochers aus Abbildung 5.46a bzw. 5.47. Die Konturen wurden mit 1.2 PE abgetastet und vorher nicht skaliert. Kleine Werte in der Tabelle korrespondieren mit hoher Ähnlichkeit. 1. Spalte: Eine bestimmte Ansicht. 2 - 4. Spalte: Die 3 ähnlichsten Ansichten mit Distanzwert zu der bestimmten Ansicht in Spalte 1. 5. - 6. Spalte: Die ähnlichsten Ansichten und zugehörigen Distanzwerte von Objekten aus den anderen beiden Klassen bezogen auf die Ansicht in Spalte 1.

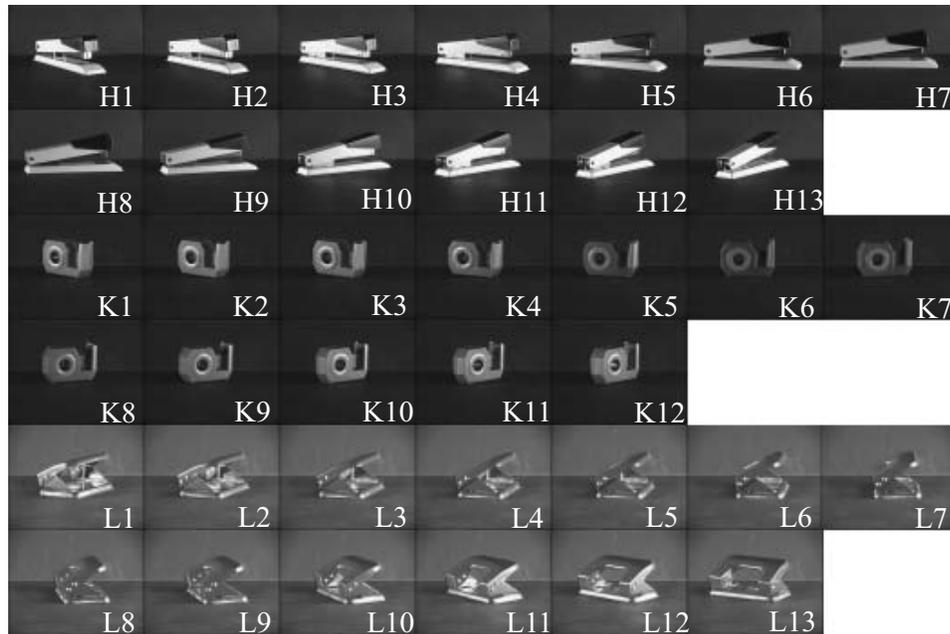


Abbildung 5.47: In dieser Abbildung sind die geordneten Ansichten der Objekte Hefter, Klebrollenhalter und Locher aus Abbildung 5.46a abgebildet.

Das andere Ergebnis ist, dass das System die Topologie der Ansichten der Objekte lernt. Dies geschieht zwar nicht exakt, aber wie Abbildung 5.48 zeigt, doch sehr gut.

Das intelligente System kennt also nach der Lernphase die topologisch benachbarten Ansichten einer bestimmten Ansicht und darüberhinaus hat das System eine Metrik gelernt, da es die globale Distanz zwischen 2 benachbarten Ansichten festgestellt hat. Damit erfährt das System implizit etwas über die Geometrie der Objekte und seine eigene relative Position zu den Objekten. Fehler im Modell werden vor allen Dingen dadurch hervorgerufen, dass 2 nicht benachbarte Ansichten sehr ähnliche 2D-Projektionen besitzen und diese Ansichten deshalb als benachbart eingelernt werden (siehe dazu ein Beispiel in Abbildung 5.49).

Eine weiteres Experiment wurde mit einem *Tisch mit Monitor* benannten Objekt durchgeführt. Die 16 aufgenommen Ansichten M1-M16 sind in Abbildung 5.50 gezeigt. Wiederum sei darauf hingewiesen, dass die Reihenfolge der aufgenommenen Bilder bzgl. der Topologie der Ansichten geordnet ist. **Das System weiß aber vor dem Lernen nichts von dieser Ordnung.** Tabelle 5.37 zeigt die komplette globale Distanzmatrix der Ansichten. Dass das System die Topologie der Ansichten erlernt hat, lässt sich daran erkennen, dass die Distanzwerte der ähnlichsten Ansichten neben der Hauptdiagonalen der Matrix liegen.

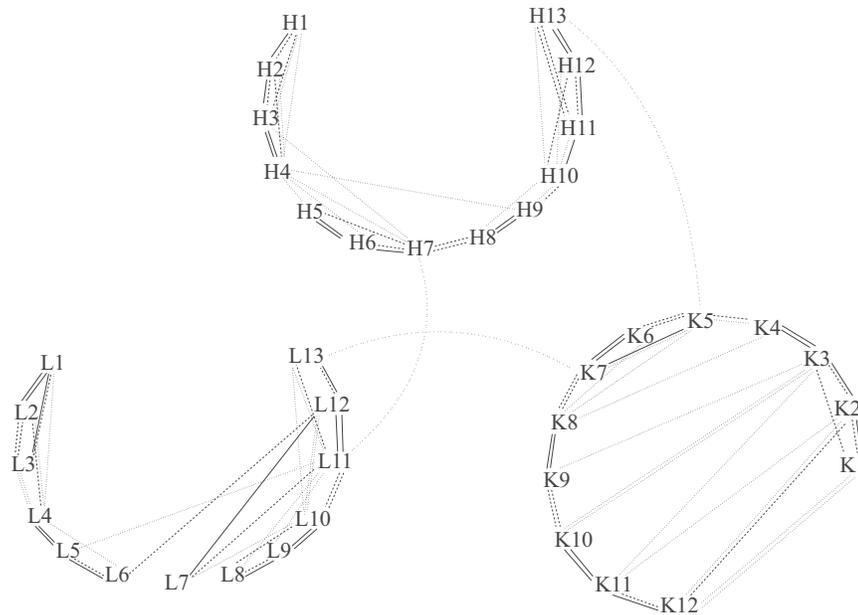


Abbildung 5.48: Visualisiertes Modell der Hefter-Kleberrollenhalter-Locher-Umwelt. Durchgezogene Linien korrespondieren mit den nächsten Nachbarn in Spalte 2, gestrichelte Linien mit den zweiten nächsten Nachbarn in Spalte 3 und gepunktete Linien mit den dritten nächsten Nachbarn in Spalte 4 der Tabelle 5.36. Die 3 gepunkteten Linien zwischen den einzelnen Objektmodellen deuten die jeweils ähnlichsten Ansichten zwischen 2 verschiedenen Objekten an.

Zusammengefasst lässt sich sagen, dass die Methode nur mit den Außenkonturen der 2D-Projektionen von 3D-Objekten in der Lage ist, gute Modelle seiner Umwelt zu erstellen und Objekte seiner Umwelt auseinanderzuhalten - beides innerhalb eines Rahmenwerkes.

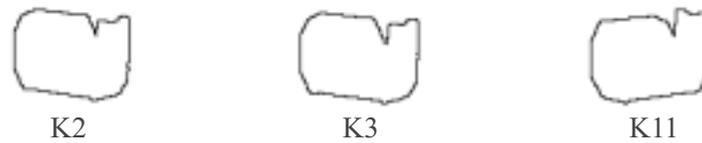


Abbildung 5.49: Problematische Ansichten durch Projektion. Die Ansichten des Kleberrollenhalters K2 und K3 sind am ähnlichsten, aber die Ansichten von K2 und K11 sind sich auch sehr ähnlich, da sich die 2D-Projektionen der Ansichten fast gleich sind.

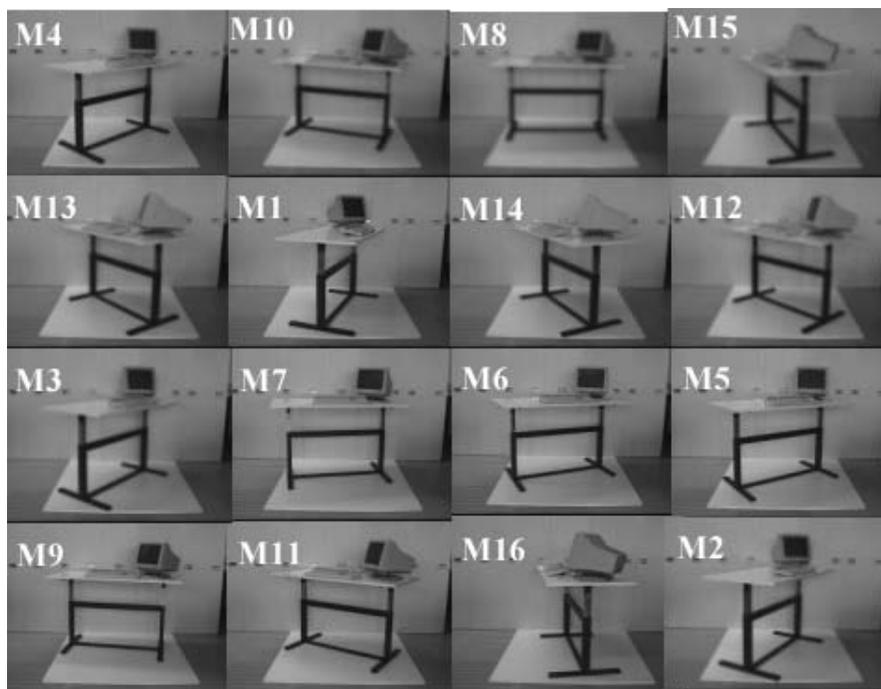


Abbildung 5.50: Die 16 ungeordneten Ansichten M1-M16 des Objektes *Tisch mit Monitor*.

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16
M1	0.0	0.2	0.6	1.1	2.3	2.8	5.0	5.2	5.9	5.5	4.9	4.5	4.7	4.0	3.8	3.5
M2	0.2	0.0	0.4	0.8	1.8	2.5	5.2	5.3	6.1	5.6	4.9	4.4	4.7	3.8	3.8	3.6
M3	0.6	0.4	0.0	0.5	1.2	2.3	4.4	4.9	6.0	5.7	4.6	4.5	4.5	3.9	3.7	3.8
M4	1.1	0.8	0.5	0.0	0.5	1.7	4.2	4.3	5.3	5.1	4.8	4.6	4.6	3.8	4.2	4.7
M5	2.3	1.8	1.2	0.5	0.0	1.0	3.0	3.1	4.3	4.3	4.8	4.8	5.0	4.2	4.6	5.6
M6	2.8	2.5	2.3	1.7	1.0	0.0	1.2	1.4	3.1	3.7	4.9	5.4	5.2	5.2	5.0	5.3
M7	5.0	5.2	4.4	4.2	3.0	1.2	0.0	0.6	2.3	3.0	4.5	5.8	5.6	5.5	5.0	5.4
M8	5.2	5.3	4.9	4.3	3.1	1.4	0.6	0.0	1.2	1.5	3.5	4.9	5.0	4.8	4.7	5.3
M9	5.9	6.1	6.0	5.3	4.3	3.1	2.3	1.2	0.0	0.4	2.7	4.2	4.0	4.2	4.8	5.0
M10	5.5	5.6	5.7	5.1	4.3	3.7	3.0	1.5	0.4	0.0	1.5	2.6	2.7	3.1	3.6	4.0
M11	4.9	4.9	4.6	4.8	4.8	4.9	4.5	3.5	2.7	1.5	0.0	0.8	1.9	1.9	2.3	3.2
M12	4.5	4.4	4.5	4.6	4.8	5.4	5.8	4.9	4.2	2.6	0.8	0.0	1.2	1.4	2.0	3.0
M13	4.7	4.7	4.5	4.6	5.0	5.2	5.6	5.0	4.0	2.7	1.9	1.2	0.0	0.8	1.3	1.7
M14	4.0	3.8	3.9	3.8	4.2	5.2	5.5	4.8	4.2	3.1	1.9	1.4	0.8	0.0	0.3	0.9
M15	3.8	3.8	3.7	4.2	4.6	5.0	5.0	4.7	4.8	3.6	2.3	2.0	1.3	0.3	0.0	0.6
M16	3.5	3.6	3.8	4.7	5.6	5.3	5.4	5.3	5.0	4.0	3.2	3.0	1.7	0.9	0.6	0.0

Tabelle 5.37: Globale Distanzmatrix für die Ansichten des Objektes *Tisch mit Monitor*. Die 3 ähnlichsten Ansichten bezogen auf die Ansichten in den Überschriften der Spalten sind in jeder Spalte fett markiert.

Kapitel 6

Zusammenfassung und Ausblick

In dieser Arbeit wurde unter dem Begriff **automatische Merkmalsynthese** ein allgemeines Konzept zur Bearbeitung von Mustererkennungsaufgaben neu eingeführt. Unter **automatischer Merkmalsynthese** wird die autonome Synthese von Basismerkmalen zu abstrakten Merkmalen verstanden, die ein künstliches intelligentes System in die Lage versetzt, seine Umwelt einzuordnen, d.h. bestimmte Situationen oder auch nur Objekte wiederzuerkennen.

Dazu erfährt das System ähnliche Situationen wiederholt und lernt daraus selbständig Schlüsse für die Wiedererkennung solcher Situationen zu ziehen. Aufgrund der Fülle von Informationen, die eine Situation oder ein Objekt ausmachen, ist es notwendig, diese zu komprimieren und anhand dieser komprimierten Informationen trotzdem ähnliche Situationen oder Objekte richtig wiederzuerkennen.

Als erster Schritt auf dem Weg zur Realisierung eines solch allgemeinen Mustererkennungskonzeptes wurden die Außenkonturen von 2D-Objekten untersucht. Konturpunkte als Basismerkmale wurden zunächst zu wichtigen Konturabschnitten synthetisiert, die 2 verschiedene Konturen unterscheiden. Im nächsten Schritt wurden die wichtigen Konturabschnitte einer Kontur, die diese Kontur von den Konturen anderer Klassen unterscheiden, zu signifikanten Konturabschnitten synthetisiert. Diese signifikanten Konturabschnitte wurden bzgl. ihrer Verlässlichkeiten für eine korrekte Klassifikation geordnet und waren so Basis eines schnellen hierarchischen Klassifikators.

All dies geschah basierend auf einer lokalen Analyse, auch deformierter, Konturen in einer Lernstichprobe.

Es wurde ein Gesamtsystem entwickelt, das neben einem komplexen, aus vielen Einzelmodulen bestehenden, Lernsystem einen relativ einfachen Klassifikator beinhaltet, dessen Entscheidungen im Gegensatz zu z.B. künstlichen neuronalen Netzen (z.B. [Bis95], [Zel94]) von einem Menschen intuitiv nachvollzogen werden können. Dabei wird die Objekterkennung translations-, rotations- und skalierungsinvariant sowie deformationstolerant durchgeführt.

Da es sich bei den Außenkonturen von 2D-Objekten im Prinzip um eindimensionale Funktionen handelt, lässt sich das vorgestellte Prinzip leicht auf andere 1D-Mustererkennungsprobleme portieren.

Ein weiterer Vorteil der gewonnenen synthetisierten Merkmale, d.h. der signifikanten Konturabschnitte ist, dass sie keine gleiche Anzahl an Merkmalen, d.h. Konturpunkten, besitzen müssen, da der vorgestellte hierarchische Klassifikator darauf eingestellt ist.

Die experimentellen Ergebnisse zeigen die erstaunliche Leistungsfähigkeit des Gesamtsystems basierend auf verschiedenen komplexen Datensätzen.

So werden basierend auf nur wenigen Objekten in der Lernstichprobe und auch nur jeweils einem einzigen signifikanten Konturabschnitt pro Hierarchieebene des Klassifikators schon gute bis sehr gute Erkennungsleistungen erzielt. Die **automatische Merkmalsynthese** benötigt also im Gegensatz zu anderen Lernverfahren wie z.B. der beliebten \rightarrow *Backpropagation*-Methode ([RHW86], [RHW88]) nur eine sehr kleine Lernstichprobe. *Backpropagation* würde beispielsweise für ein ähnlich gutes Klassifikationsergebnis der sehr variablen Teststichprobe *Vögel, Flaschen und Fische* weitaus mehr Objekte in der Lernstichprobe benötigen (Hinweise darauf finden sich in [Wer97]).

Die experimentellen Ergebnisse zu den einzelnen Modulen stellen heuristisch nach und nach die ohnehin wenigen Parameter ein und so lässt sich von einem nahezu automatischen Lern- und Klassifikationssystem sprechen.

Das vorgestellte Verfahren wurde im globalen Fall basierend auf einem bekannten Datensatz aus dem World Wide Web mit 2 anderen aktuellen globalen Konturanalyseverfahren verglichen und es stellte sich heraus, dass die Erkennungsleistungen durchaus vergleichbar, teils sogar besser sind.

Außerdem konnte gezeigt werden, dass der globale Teil des Lernsystems dazu verwendet werden kann, einen (mobilen) Roboter in die Lage zu versetzen, ein implizites Modell seiner Umwelt selbständig zu erlernen, mit dem er navigieren oder auch einzelne Objekte erkennen könnte. Und das Ganze in einem Rahmenwerk.

Folgende Verbesserungen und Erweiterungen des hier vorgestellten Lern- und Klassifikationssystems sollten in Angriff genommen werden:

Allem voran bietet sich für eine robustere Erkennung und hohe Trefferquoten an, zur Klassifikation nicht nur jeweils einen signifikanten Konturabschnitt zu benutzen, sondern mehrere. Dazu sollte die Reihenfolge und die ungefähre Position solcher signifikanter Konturabschnitte in der Kontur genutzt werden.

Ferner muss darüber nachgedacht werden, wie man die Anzahl der Fehlklassifikationen einschränkt, ohne dabei die Trefferquote allzusehr zu vermindern. Hier bietet sich zunächst eine nähere Untersuchung der Bestimmung der, den einzelnen

signifikanten Konturabschnitten zugeordneten, Zurückweisungsschwellen an.

Das Einstellen der Parameter Importanzschwelle Θ bzw. $\tilde{\Theta}$ und der minimalen Konturpunktanzahl L_{min} eines signifikanten Konturabschnittes sowie der Parameter zur Bestimmung der lokalen Ähnlichkeiten von Konturabschnitten l_{const} bzw. E durch Ausprobieren ist sicherlich noch keine endgültige Lösung. Es lassen sich zwar schnell enge Intervalle angeben, die für verschiedene Datensätze zu ähnlich guten Klassifikationsergebnissen führen wenn die Konturen auf gleiche Länge normiert sind, doch ist es notwendig darüber nachzudenken wie auch die Einstellung dieser Parameter automatisiert werden kann.

Zu untersuchen ist weiterhin die Skalierungsinvarianz. Das populäre Skalieren der Außenkonturen auf gleichen Umfang ist sicherlich nur ein Behelf. Man stelle sich nur ein- und denselben Menschen vor, der einmal seine Arme anlegt und einmal ausstreckt. Die Außenkontur des Menschen mit ausgestreckten Armen ist länger als die des Menschen mit angelegten Armen. Daraus folgt, dass bei einer Skalierung auf gleichen Umfang gleich große lokale Konturabschnitte plötzlich unterschiedlich groß sind. Es sollte also auch bei der Skalierung untersucht werden, inwieweit eine globale Skalierung durch lokale Skalierungen ersetzt werden können. Es ist in diesem Zusammenhang auch prinzipiell die Frage zu stellen, ob es nicht ausreicht Erkennungsverfahren skalierungstolerant und nicht skalierungsinvariant auszulegen. In [Zin01] findet man erste Hinweise darauf, dass Verfahren entwickelt werden können, die die Skalierungstoleranz eines Objektes in einem Bereich von ca. halb so klein bis doppelt so groß gewährleisten.

Ein weiterer Punkt ist die Einschränkung der signifikanten Konturabschnitte einer Klasse zu einigen wenigen abstrakten Konturabschnitten dieser Klasse. Dies dient der weiteren Komprimierung der Wissensbasis des Klassifikators, d.h. diese Maßnahme würde zu einer Verkleinerung des Speicherbedarfs und der Rechenzeit führen. Hier wurde in den experimentellen Ergebnissen schon gezeigt, dass nur eine Untermenge der signifikanten Konturabschnitte in der Wissensbasis des Klassifikators überhaupt zur Klassifikation herangezogen werden. Dies könnte ein Ansatzpunkt für weitere Untersuchungen in diese Richtung sein.

Eine andere Herangehensweise könnten sogenannte α -shapes ([EKS83], [AEF⁺95]) bieten. α -shapes sind eine Verallgemeinerung der *konvexen Hüllen* und stellen eine bessere Approximation des Umrisses einer Punktverteilung in der Ebene oder im Raum dar. Der Parameter α sorgt dabei, je nach Einstellung, für eine feinere oder gröbere Anpassung. Ein Ansatz wäre also, die signifikanten Konturabschnitte einer Klasse, beispielsweise die Flaschenhalse in der Flaschenklasse, bestmöglich übereinanderzulegen und um diese Konturpunktverteilung herum ein α -shape zu berechnen. Fällt dann ein Konturabschnitt einer zu klassifizierenden Kontur in ein solches α -shape, dann wird diese Kontur in die Klasse des α -shapes klassifiziert.

Im allgemeinen ist es notwendig, die hier vorgestellten Konzepte auf 2D-Grauw- bzw. 2D-Farbbilder, aber auch in einem nächsten Schritt auf den 3D-Fall, zu erweitern. In diesem Zusammenhang wurden in [Sta00] und [Zin01] erste Untersuchungen zur Ähnlichkeit lokaler Bereiche in Grauwert- und Farbbildern unternommen. Speziell in [Sta00] wurde ein Algorithmus, die sogenannte *elastische Sprite-Korrelation*, realisiert, der in einem Bild einen lokalen Bildbereich (Sprite) sucht, der auch deformiert sein darf. Zur Erhöhung der Verarbeitungsgeschwindigkeit wird dort in einzelnen Pyramidenstufen des Bildes gesucht.

Abschließend zu dieser Arbeit sei gesagt, dass die in diesem ersten Ansatz zur Realisierung eines allgemeinen Mustererkennungskonzeptes, basierend auf der **automatischen Merkmalsynthese**, erhaltenen Ergebnisse so ermutigend sind, dass weiterhin in diese Richtung nachgedacht werden sollte.

Literaturverzeichnis

- [AB95] D. W. Aha and R. L. Bankert. A comparative evaluation of sequential feature selection algorithms. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pages 1–7, Fort Lauderdale, FL, 1995.
- [ACH⁺91] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 13(3):209–216, März 1991.
- [AEF⁺95] N. Akkiraju, H. Edelsbrunner, M. Facello, P. Fu, E. P. Mücke, and C. Varela. Alpha shapes: Definition and software. In *Proceedings of the 1st International Computational Geometry Software Workshop*, pages 63–66, 1995.
- [BB82] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [BCGJ95] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. In *IEEE Workshop on Physics-Based Modeling in Computer Vision*, pages 135–143, 1995.
- [BCGJ98] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1998.
- [Bel57] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.
- [Bel95] T. B. Van Belle. A new approach to genetic-based automatic feature discovery. Master’s thesis, University of Alberta, Department of Computing Science, Edmonton, Alberta, 1995.
- [BG80] E. Bribiesca and A. Guzman. How to describe pure form and how to measure differences in shapes using shape numbers. *Pattern Recognition*, 12:101–112, 1980.

- [Bis95] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.
- [BL97] A. L. Blum and P. L. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, pages 245–271, 1997.
- [Blu67] H. Blum. A transformation for extracting new descriptors of shape. In W. Whaten-Dunn, editor, *Models for the Perception of Speech and Visual Forms*, pages 362–380. MIT Press, Cambridge, Massachusetts, 1967.
- [Boc74] H. H. Bock. *Automatische Klassifikation*. Vandenhoeck & Ruprecht, Göttingen, 1974.
- [BS87] I. N. Bronstein and K. A. Semendjajew. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Thun und Frankfurt/Main, 1987.
- [BU93] R. Basri and S. Ullman. The alignment of objects with smooth surfaces. *CVGIP: Image Understanding*, 57(3):331–345, May 1993.
- [BV99] J. P. Braquelaire and A. Vialard. Euclidean paths: A new representation of boundary of discrete regions. *Graphical Models and Image Processing*, 61:16–43, 1999.
- [Chu95] K.-L. Chung. Finding shape numbers in parallel. *Pattern Recognition Letters*, 16:699–702, 1995.
- [CL96] C. Charras and T. Lecroq. Exact string matching algorithms. Technical report, Laboratoire d’Informatique de Rouen, Université de Rouen, 1996. URL: <http://www-igm.univ-mlv.fr/le-croq/string/string.ps>.
- [CRW97] P. I. Corke, J. M. Roberts, and G.J. Winstanley. 3d perception for mining robots. In *Proceedings of International Conference on Field and Service Robotic*, pages 41–47, Canberra, 1997.
- [Cun89] Y. Le Cun. Generalization and network design strategies. In R. Pfeifer, Z. Schreter, F. Fogelmansoulie, and L. Steels, editors, *Connectionism in Perspective*, chapter 40, pages 143–155. Elsevier Science, Amsterdam, 1989.
- [DSAW98] C. Demant, B. Streicher-Abel, and P. Waszkewitz. *Industrielle Bildverarbeitung*. Springer, Berlin, Heidelberg, New York, 1998.

- [EKS83] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983.
- [Faw90] T. E. Fawcett. Feature discovery for inductive concept learning. Technical Report 90-15, University of Massachusetts, Department of Computer and Information Science, Amherst, MA, 1990.
- [FC88] M. Ferraro and T. M. Caelli. Relationship between integral transform invariances and lie group theory. *Journal Optical Society America*, 5:738–742, Mai 1988.
- [FM82] K. Fukushima and S. Miyake. Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 15(6):445–469, 1982.
- [Fou22] J. B. Fourier. *Theorie analytique de la chaleur*, 1822.
- [FS93] R.W. Frischholz and K.P. Spinnler. Class of algorithms for real-time subpixel registration. In D.W. Braggins, editor, *Computer Vision for Industry*, SPIE Proceedings Vol. 1989, pages 50–59, 1993.
- [Fu74] K. S. Fu. *Syntactic Methods in Pattern Recognition*. Academic Press, New York, 1974.
- [FU92] T. E. Fawcett and P. E. Utgoff. Automatic feature generation for problem solving systems. In D. Sleeman and P. Edwards, editors, *Machine Learning: Proceedings of the Ninth International Workshop (ML92)*, pages 144–153, San Mateo, California, 1992.
- [Fuk80] K. Fukushima. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by a shift in position. *Biological Cybernetics*, 36:193–202, 1980.
- [GW99] Y. Gdalyahu and Daphna Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *Transactions on Pattern Analysis and Machine Intelligence*, 21:1312–1328, 1999.
- [Hab95] P. Haberäcker. *Praxis der Digitalen Bildverarbeitung und Mustererkennung*. Hanser, München, Wien, 1995.
- [Hay98] W. G. Hayward. Effects of outline shape in recognition. *Journal of Experimental Psychology: Human Perceptions and Performance*, 24:427–440, 1998.

- [Hem98] Th. Hemrich. Implementierung eines linearen- und Polynomklassifikators und Einsatz in einer industriellen Qualitätsprüfungsaufgabe, November 1998. Diplomarbeit an der Universität Siegen, Fachbereich Elektrotechnik und Informatik, Fachgruppe Prozessdatenverarbeitung.
- [HR84] D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, 18:65–96, 1984.
- [HS97] D. D. Hoffman and M. Singh. Saliency of visual parts. *Cognition*, 63:29–78, 1997.
- [HS99] H. Heinol and R. Schwarte. Photomischdetektor erfaßt 3D-Bilder. *Elektronik*, pages 80–90, Dezember 1999.
- [Hu62] M .K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:179–187, 1962.
- [HU88] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment. In *Proceedings of the DARPA Image Understanding Workshop*, pages 1114–1124, Cambridge, April 1988.
- [Huß00] S. Hußmann. *Schnelle 3D-Objektvermessung mittels PMD/CMOS-Kombizeilensensor und Signalkompressions-Hardware*. PhD thesis, Universität Siegen, 2000.
- [Jäh93] B. Jähne. *Digitale Bildverarbeitung*. Springer-Verlag, Berlin, Heidelberg, 1993.
- [JL99] M. Jogan and A. Leonardis. Panoramic eigenimages for spatial localisation. In *8th International Conference on Computer Analysis of Images and Patterns CAIP 99*, pages 558–567, 1999.
- [Jun94] D. Jungnickel. *Graphen, Netzwerke und Algorithmen*. BI-Wissenschafts-Verlag, Mannheim; Leipzig; Wien; Zürich, 1994.
- [KB96] K. Y. Kupeev and V. L. Brailovsky. A slowest-random algorithm for a partial contour similarity problem. Technical Report 310/96, Computer Science Department, Tel Aviv University, 1996. www.math.tau.ac.il.
- [KB98] K. Y. Kupeev and V. L. Brailovsky. A reinforced random algorithm for a partial contour perceptual similarity problem. *Pattern Recognition Letters*, 19:287–297, 1998.

- [KP02] K.-D. Kuhnert and D. Pechtel. Towards creating abstract features of complex objects - the fusion of contour points in significant contour sections for object recognition. *Pattern Recognition Letters*, 23(6):713–718, April 2002.
- [KSS86] A. Kalvin, E. Schonberg, J. T. Schwartz, and M. Sharir. Two-dimensional, model-based, boundary matching using footprints. *International Journal of Robotics Research*, 5:38–55, 1986.
- [Kuh98] K.-D. Kuhnert. Recognition of complex objects for sorting post consumer goods by using image processing and neural networks. In E.H.M. Cheung, S.Y.L. Lang, P.W. Tse, W.L. Xu, and R.W.H. Yeung, editors, *Proceedings of Fifth International Conference on Mechatronics And Machine Vision in Practice*, pages 265–270, 1998.
- [KW94] K. Y. Kupeev and H. J. Wolfson. On shape similarity. In *Proceedings of International Conference on Computer Vision & Image Processing*, pages 227–231, 1994.
- [KW96] K. Y. Kupeev and H. J. Wolfson. A new method of estimating shape similarity. *Pattern Recognition Letters*, pages 873–887, 1996.
- [KWT87] M. Kass, A. Witkin, and D. Terzopolous. Snakes: Active contour models. *International Journal Computer Vision*, 1(4):321–331, 1987.
- [Lak00] R. Lakämper. *Formbasierte Identifikation zweidimensionaler Objekte*. PhD thesis, Fachbereich Mathematik, Universität Hamburg, 2000.
- [LB96] J. C. Litter and H. H. Bülthoff. An introduction to object recognition. Technical Report 43, Max-Planck-Institut für biologische Kybernetik, Spemannstraße 38, 72076 Tübingen, Germany, November 1996. [HTTP://www.mpik-tueb.mpg.de/bu.html](http://www.mpik-tueb.mpg.de/bu.html).
- [Lei66] G. W. Leibniz. *Zur Analysis der Lage*. In *Hauptschriften zur Grundlegung der Philosophie*. Meiner, Hamburg, 1687/1966.
- [LEM] Silhouette Database of the Vision Group of the LEMS at Brown University, Providence, Rhode Island, USA: www.lems.brown.edu/vision/researchAreas/SIID/silhouette-database.tar.gz.

- [Lin98] D. Lin. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning 1998*, pages 296–304, 1998.
- [LL] L. J. Latecki and R. Lakämper. Application of planar shape comparison to object retrieval in image databases. Erscheint in *Pattern Recognition*.
- [LL98a] L. J. Latecki and R. Lakämper. Discrete approach to curve evolution. In P. Levi, R. J. Ahlers, F. May, and M. Schanz, editors, *Mustererkennung 1998*, pages 85–92, Berlin, Heidelberg, New York [...], 1998. DAGM, Springer-Verlag.
- [LL98b] L.J. Latecki and R. Lakämper. Shape decomposition and shape similarity measure. In P. Levi, M. Schanz, R.-J. Ahlers, and F. May, editors, *Mustererkennung 1998*, pages 367–376, Berlin, Heidelberg, New York [...], 1998. DAGM, Springer-Verlag.
- [LL00] L.J. Latecki and R. Lakämper. Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10), Oktober 2000.
- [Lon98] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.
- [LTBK97] J. C. Liter, B. S. Tjan, H. H. Bülthoff, and N. Köhnen. Viewpoint effects in naming silhouette and shaded images of familiar objects. Technical Report 54, Max-Planck-Institut für biologische Kybernetik, Spemannstraße 38, 72076 Tübingen, Germany, 1997. [HTTP://www.mpik-tueb.mpg.de/bu.html](http://www.mpik-tueb.mpg.de/bu.html).
- [MAK96] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In *Image Databases and Multi-Media Search. Proceedings of the First International Workshop IDB-MMS'96*, pages 35–42, Amsterdam, 1996.
- [MM92] F. Mokhtarian and A. K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:789–805, August 1992.
- [MRMW98] S. Müller, G. Rigoll, D. Mazurenok, and D. Willet. Invariante Erkennung handskizzierter Piktogramme mit Anwendungsmöglichkeiten in der inhaltsorientierten Bilddatenbankabfrage. In Peter

- Levi, R.-J. Ahlers, F. May, and M. Schanz, editors, *Mustererkennung 1998*, pages 271–278, Stuttgart, 1998. Deutsche Arbeitsgemeinschaft für Mustererkennung, Springer.
- [MS00] B. Markuse and A. Schneider. Ähnlichkeit, Nähe, Distanz: Zur Anwendung multidimensionaler Skalierung in musikwissenschaftlichen Untersuchungen. In W. Auhagen, B. Gätjen, and K. W. Niemöller, editors, *Systemische Musikwissenschaft. Festschrift Jobst Peter Fricke zum 65. Geburtstag*. Peter Lang, Frankfurt, 2000.
- [Neb99] S. Nebeling. Analyse der Konturen deformierter Objekte (Leichtverpackungen) aus DSD (Duales System Deutschland) - Sammlungen mit Hilfe clusteranalytischer Methoden zur Objekterkennung, 1999. Studienarbeit, FG Prozessdatenverarbeitung, FB Elektrotechnik und Informatik, Universität GHS Siegen.
- [Nie74] H. Niemann. *Methoden der Mustererkennung*. Akademische Verlagsgesellschaft, Frankfurt am Main, 1974.
- [Nie83] H. Niemann. *Klassifikation von Mustern*. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1983.
- [Nie90] H. Niemann. *Pattern Analysis and Understanding*. Springer-Verlag, Berlin; Heidelberg; New York [u.a.], 1990.
- [Nis98] H. Nishida. Matching and recognition of deformed closed contours based on structural transformation models. *Pattern Recognition*, 31(10):1557–1571, 1998.
- [Ott95] R. Otterbach. *Robuste 3D-Objekterkennung und Lagebestimmung durch Auswertung von 2D-Bildfolgen*. PhD thesis, Universität Siegen, 1995.
- [Pav77] T. Pavlidis. *Structural Pattern Recognition*. Springer-Verlag, Berlin [u.a.], 1977.
- [Pav78] T. Pavlidis. A review of algorithms for shape analysis. *Computer Graphics Image Processing*, 7:243–258, 1978.
- [PF86] E. Persoon and K. S. Fu. Shape discrimination using fourier descriptors. *PAMI*, 8(3):388–397, May 1986.

- [PK99] D. Pechtel and K.-D. Kuhnert. Automatic generation of significant and local feature groups of complex and deformed objects. In B. Werner, editor, *Proceedings 10th International Conference on Image Analysis and Processing (ICIAP99)*, pages 340–345. IAPR, IEEE Computer Society, 1999.
- [PK00a] D. Pechtel and K.-D. Kuhnert. Ein Lern- und Klassifikationssystem zur Erkennung komplexer und/oder deformierter 2D-Objektkonturen mit Merkmalsfusion. In *Mustererkennung 2000*, pages 229–236. Deutsche Arbeitsgemeinschaft für Mustererkennung e.V., Springer-Verlag, 2000.
- [PK00b] D. Pechtel and K.-D. Kuhnert. Generating automatically local feature groups of complex and deformed objects. In W. Gaul and R. Decker, editors, *Classification and Information Processing at the Turn of the Millenium*, pages 237–244. Gesellschaft für Klassifikation e.V., Springer-Verlag, 2000.
- [PK00c] D. Pechtel and K.-D. Kuhnert. Towards automatically learning an implicit model from 2d-images based on a local similarity analysis of contours. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.
- [PK00d] D. Pechtel and K.-D. Kuhnert. Towards feature fusion - the synthesis of contour sections distinguishing contours from different classes. In G. Borgefors, I. Nyström, and G. Sanniti di Baja, editors, *9th Discrete Geometry for Computer Imagery Conference*, LNCS 1953, pages 518–529. Springer-Verlag, 2000.
- [PK02] D. Pechtel and K.-D. Kuhnert. Towards feature fusion - a classifier on the basis of automatically generated significant contour sections. In W. Gaul and G. Ritter, editors, *Classification, Automation, and New Media*, pages 151–158. Gesellschaft für Klassifikation e.V., Springer-Verlag, 2002.
- [PL92] S. Perantonis and P. Lisboa. Translation, rotation and scale invariant pattern recognition by high-order neural networks and moment classifiers. *IEEE Transactions Neural Networks*, 3:241–251, März 1992.
- [PM47] W. Pitts and W. McCulloch. How we know universals: the perception of auditory and visual forms. *Bulleting of Mathematical Biophysics*, 9:127–147, 1947.

- [PR92] R. J. Prokop and A. P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphical Models and Image Processing*, 54(5):438–460, 1992.
- [RHC99] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Past, present and future. *Journal of Visual Communication and Image Representation*, 10:1–23, 1999.
- [RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, pages 318–362. MIT Press, Cambridge, 1986.
- [RHW88] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. In J. A. Anderson and E. Rosenfeld, editors, *Neurocomputing: Foundations of Research*, chapter 42, pages 696–700. MIT Press, 1988.
- [Rit74] J. Ritter, editor. *Historisches Wörterbuch der Philosophie, Bd. 3: G-H*, pages 1115–1120. Wissenschaftliche Buchgesellschaft, Darmstadt, 1974.
- [Ros97] P. L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Transactions on Pattern Analysis and Machine Vision*, 19:659–666, Juni 1997.
- [Sch77] J. Schürmann. *Polynomklassifikatoren für die Zeichenerkennung*. R. Oldenbourg Verlag, München, Wien, 1977.
- [Sch95] R. Schmid. *Industrielle Bildverarbeitung*. Vieweg, Braunschweig/Wiesbaden, 1995.
- [Sch97] U. Schnell. Gipfelstürmer - Industrielle Bildverarbeitung bleibt weiterhin Vorzeigebbranche. *Maschinenmarkt*, 103:12–14, 1997.
- [Sch98a] R. Schiller. *Konturbasierte Verfahren in der lageinvarianten Mustereerkennung*. PhD thesis, Technische Universität Hamburg-Harburg, 1998. Fortschr.-Ber. VDI-Reihe 10 Nr. 556. Düsseldorf: VDI-Verlag.
- [Sch98b] T. Schüttler. Untersuchung der Ähnlichkeit von Konturen mittels verschiedener Optimierungsverfahren, September 1998. Diplomarbeit an der Universität Siegen, Fachbereich Elektrotechnik und Informatik, Fachgruppe Prozessdatenverarbeitung.

- [SCTK98] D. Sharvit, J. Chan, H. Tek, and B. Kimia. Symmetry based indexing of image databases. *Journal Visual Communication and Image Representation*, 1998.
- [SGWM93] T. Sederberg, P. Gao, G. Wang, and H. Mu. 2d shape blending: An intrinsic solution to the vertex path problem. *Computer Graphics*, 38:15–18, 1993.
- [SHX95] R. Schwarte, H. Heinol, and Z. Xu. A new fast, precise and flexible 3d-camera concept using rf-modulated and incoherent illumination. In *Proceedings of SENSOR'95*, Nürnberg, 1995.
- [SK83] D. Sankoff and B.K. Kruskal, editors. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, Reading, 1983.
- [SK95] K. Siddiqi and B.B. Kimia. Parts of visual form: Computational aspects. *IEEE Transactions On Pattern Analysis and Machine Intelligence (PAMI)*, 17(3):239–251, 1995.
- [SL77] D. Steinhausen and Klaus Langer. *Clusteranalyse*. Walter de Gruyter, Berlin, New York, 1977.
- [SM95] H. Schulz-Mirbach. *Anwendung von Invarianzprinzipien zur Merkmalgewinnung in der Mustererkennung*. PhD thesis, Technische Universität Hamburg-Harburg, 1995. Fortschr.-Ber. VDI Reihe 10 Nr. 372. Düsseldorf: VDI-Verlag.
- [SN94] D. Schoder and H. Nücke. Intelligente Abfallsortierung. *Design & Elektronik*, November 1994.
- [SSDZ98] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. In *Proceedings of International Conference on Computer Vision ICCV 98*, pages 222–229, 1998.
- [Sta00] A. Staudt. Elastic sprite correlation, Dezember 2000. Studienarbeit an der Universität Siegen, Fachbereich Elektrotechnik und Informatik, Fachgruppe Prozessdatenverarbeitung.
- [STK96] K. Siddiqi, K. J. Tresness, and B. B. Kimia. Parts of visual form: Psychophysical aspects. *Perception*, 25:399–424, 1996.
- [Sto95] D. Stotz. *Computergestützte Audio- und Videotechnik*. Springer, Berlin, Heidelberg, 1995.

- [TJT96] O.D. Trier, A.K. Jain, and T. Taxt. Feature-extraction methods for character-recognition: A survey. *PR*, 29(4):641–662, April 1996.
- [UB91] S. Ullman and R. Basri. Recognition by linear combination of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992–1006, 1991.
- [VDM99a] Automatisierung mit Industrieller Bildverarbeitung 2000/01. Fachabteilung Industrielle Bildverarbeitung / Machine Vision der Fachgemeinschaft Robotik + Automation im Verband Deutscher Maschinen- und Anlagenbau e.V. (VDMA), 1999.
- [VDM99b] Marktbefragung Industrielle Bildverarbeitung/Machine Vision 1998/99. Fachabteilung Industrielle Bildverarbeitung / Machine Vision der Fachgemeinschaft Robotik + Automation im Verband Deutscher Maschinen- und Anlagenbau e.V. (VDMA), April 1999.
- [VH99] R.C. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. Technical Report 1999-27, Department of Computer Science, Utrecht University, 1999.
- [VIS00] Information der MESSE STUTTGART zur VISION 2000, der Internationalen Fachmesse für industrielle Bildverarbeitung und Identifikationstechnologien vom 18.-20. Okt. 2000, 2000.
- [VRH⁺93] T. Viéville, F. Romann, B. Hotz, H. Mathieu, M. Buffa, L. Robert, P. E. D. S. Facao, O. Faugeras, and J. T. Audren. Autonomous navigation of a mobile robot using inertial and visual cues. In *Proceedings of 1993 International Conference on Intelligent Robots and Systems IROS 93*, Yokohama, 1993.
- [WCA99] J. Wang, W. Chang, and R. Acharya. Efficient and effective similar shape retrieval. In *Proceedings of 6th IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)*, 1999.
- [Wer97] R. Werner. *Invariante Mustererkennung zweidimensionaler Objekte*. PhD thesis, Universität Koblenz-Landau, Dezember 1997.
- [Woo96] J. Wood. Invariant pattern recognition: A review. *Pattern Recognition*, 29(1):1–17, 1996.
- [Yin98] Y. Yin. Implementierung einer Bildausschnittsanzeige zur Untersuchung der Erkennbarkeit eines Objektes im Bild, 1998. Studienarbeit, FG Prozessdatenverarbeitung, FB Elektrotechnik und Informatik, Universität GH Siegen.

- [Zel94] A. Zell. *Simulation Neuronaler Netze*. Addison-Wesley, Bonn; Paris; Reading; Mass. [u.a.], 1994.
- [Zin01] H. Zinke. Lokale Rotationsinvarianz mittels Hauptkomponenten, Januar 2001. Diplomarbeit an der Universität Siegen, Fachbereich Elektrotechnik und Informatik, Fachgruppe Prozessdatenverarbeitung.
- [ZR72] C. Zahn and R. Roskies. Fourier descriptors for plane closed curves. *Computer Graphics Image Processing*, 21:269–281, 1972.
- [ZT98] D. Ziou and S. Tabbone. Edge detection techniques - an overview. *International Journal of Pattern Recognition and Image Analysis*, 8(4):537–559, 1998.

Anhang A

Formelzeichen

Kapitel 2

α	Krümmungswinkel
γ	Rotationswinkel
$\Delta\alpha$	Differential bzw. Zuwachs eines Winkels
$\Delta\alpha_i$	Krümmungswinkel bei Polygonen
Δm_i	Diskretes Masselement eines Objektes
Δs	Differential bzw. Zuwachs der Bogenlänge
Δt	Differential bzw. Zuwachs der Zeit
Δx	Differential bzw. Zuwachs der x-Koordinate
Δy	Differential bzw. Zuwachs der y-Koordinate
η_{pq}	zweidimensionales normiertes zentriertes Carthesisches Moment der Ordnung $p+q$ im diskreten Fall
μ_{pq}	zweidimensionales zentriertes Carthesisches Moment der Ordnung $p+q$ im diskreten Fall
$\Phi, \Phi - s$	Funktion zur Akkumulierung von Krümmungswinkeln bei Polygonen
ρ	Krümmungsradius eines Krümmungskreises
σ	Skalierungsfaktor
θ	Orientierungswinkel einer Hauptachse zur x-Achse
a	große Halbachse einer Ellipse
a_k	Koeffizienten der Cosinusanteile einer Fourierreihenentwicklung
A	Anzahl diskreter Masselemente Δm_i eines Objektes

b	kleine Halbachse einer Ellipse
b_k	Koeffizienten der Sinusanteile einer Fourierreihenentwicklung
$\frac{d\alpha}{ds}$	Differentialquotient, der die Änderung des Winkels α bzgl. der Bogenlänge s in einem Punkt beschreibt
$f(x), g(x)$	Funktion mit einer Veränderlichen
$f'(x), f''(x), f'''(x)$	1., 2. und 3. Ableitung einer Funktion $f(x)$
$f(x, y)$	Kontinuierliche Funktion von 2 Veränderlichen
f_k	diskrete Frequenzen bei einer Fourierreihenentwicklung
<i>Fläche</i>	Fläche eines 2D-Objektes
$g'(x), g''(x), g'''(x)$	1., 2. und 3. Ableitung einer Funktion $g(x)$
$g(x, y)$	Diskrete Funktion von 2 Veränderlichen auf dem Pixelraster
J	Massenträgheitsmoment eines physikalischen Körpers
J_K	Massenträgheitsmoment eines 2D-Objektes oder einer Kontur bzgl. ihres Schwerpunktes
K	Krümmung einer Kurve in einem Punkt
<i>Kompaktheit</i>	Kompaktheit einer Außenkontur oder eines 2D-Objektes
m_{pq}	zweidimensionales Carthesisches Moment der Ordnung $p+q$ im diskreten Fall
m_{pq}^*	zweidimensionales Carthesisches Moment der Ordnung $p+q$ im kontinuierlichen Fall
M	Anzahl der Pixel einer Spalte eines diskreten Bildes
$M_1 - M_7$	Invariante Merkmale basierend auf regulären Momenten
N	Anzahl der Pixel einer Zeile eines diskreten Bildes; Menge der natürlichen Zahlen
P_B	Punkt einer Kurve
P_i	Konturpunkte
$r(\alpha)$	Eindimensionale Funktion, die den polaren Abstand von Konturpunkten zum Schwerpunkt beschreibt
r_i	Abstand eines diskreten Masselementes Δm_i eines Objektes von seinem Schwerpunkt
r_α	Abstand eines Konturpunktes vom Schwerpunkt der Kontur (polarer Abstand)

R	Drehradius (engl. radius of gyration)
S	Schwerpunkt einer Kontur
T	Periode einer periodischen Funktion
<i>Umfang</i>	Umfang eines 2D-Objektes
x	x-Koordinate eines 2D-Objektes oder einer Kontur
x'	Transformierte x-Koordinate
\bar{x}	Mittelwert der x-Koordinaten eines 2D-Objektes oder einer Kontur
y	y-Koordinate eines 2D-Objektes oder einer Kontur
y'	Transformierte y-Koordinate
\bar{y}	Mittelwert der y-Koordinaten eines 2D-Objektes oder einer Kontur

Kapitel 3

Teilbasierter Ansatz

α	Wendewinkel an einem Eckpunkt eines Polygons
σ	Halbwertsbreite eines Gaußfilters
$\tilde{a}_i, \tilde{a}_j^*$	maximale konvexe bzw. konkave Bögen
A	Fläche zwischen 2 Winkelkrümmungsfunktionen
A_{min}	minimierte Fläche zwischen 2 Winkelkrümmungsfunktionen
c_{elem}	Kosten für das Eliminieren von Konturpunkten
c_{Bogen}	Ähnlichkeitsmaß für den Vergleich von 2 maximalen Bögen
K, K^*	Diskrete Konturen
K', K''	Konturen auf diskreten Evolutionsstufen
\tilde{K}, \tilde{K}^*	Konturen nach Abschluss der diskreten Kurvenevolution auf bestimmte Evolutionsstufen
l, l^*	Gesamtlängen von maximalen konvexen bzw. konkaven Bögen

l_1, l_2	Längen von Geradensegmenten zwischen den Punkten einer Kontur
L	Längenfaktor für das Ähnlichkeitsmaß zweier maximaler Bögen
m, n	Anzahl der Konturpunkte von \tilde{K}, \tilde{K}^*
M	Anzahl der Konturpunkte von K
p_{elem}, p'_{elem}	Eliminierte Punkte einer Kontur
p_i	Punkt der Kontur K
P	Punkt einer Kontur
s_1, s_2	Geradensegmente zwischen Punkten einer Kontur
s_{neu}	Neu entstehendes Geradensegment nach Elimination eines Konturpunktes
S	Skalierungsfaktor für das Ähnlichkeitsmaß zweier maximaler Bögen

Strukturelle Mustererkennung

c	Konstanter Faktor zur Gewichtung von Ähnlichkeitsmaßanteilen
d	Summe der Differenzen zwischen den Gewichten korrespondierender Klumpen zweier Konturen
D	Summe der Gewichte der Blattklumpen auf einem Pfad durch $M(K)$
F	Ähnlichkeitsmaß von 2 Konturen
$G(K)$	G-Graph
I	Isomorphismus zwischen 2 Graphen
K	Kontur
K', K'', \dots	Konturen nach Abtrennung von Klumpen
K^*	Beliebige Kontur des Segmentierungsgitters $M(K)$
$L(G(K))$	Morphismusgitter (engl. morphism lattice)
$M(K)$	Segmentierungsgitter (engl. segmentation lattice)
$S(K)$	Segmentierung einer Kontur
P_i, P_j, P_k	Klumpen (engl. lump) einer Kontur nach Segmentierung
P'_j, P'_k	Klumpen (engl. lump) einer Kontur K'

sim_y	Ähnlichkeitsmaß von 2 Konturen in y-Richtung
v	Ein Element aus $V(G(K^*))$
$V(G(K^*))$	Die Menge aller Knoten von $G(K^*)$
w	Gewicht

Syntaktische Anpassung

φ_i, φ_j^*	(Krümmungs)winkel, der von 2 Geradensegmenten eingeschlossen wird
ρ_i	Schätzung des Krümmungswinkels am Punkt p_i
c_{ein}	Kosten für das Einfügen eines Geradensegmentes
c_{ent}	Kosten für das Entfernen eines Geradensegmentes
c_{ers}	Gesamtkosten für das Ersetzen eines Geradensegmentes
c_{ers}^l	Längenanteil an Ersetzungskosten
c_{ers}^φ	Krümmungsanteil an Ersetzungskosten
k_i, k_j^*	Geradensegmente der Konturen K bzw. K^*
l_i, l_j^*	Länge eines Geradensegmentes
K, K^*	Konturen
m	Anzahl der Konturpunkte p_i der Kontur K
M	Anzahl der Geradensegmente der Kontur K
N	Anzahl der Geradensegmente der Kontur K^*
p_i	Punkte einer Kontur
P_i	Punkte an Stellen maximaler Konturkrümmung
\vec{v}_1, \vec{v}_2	Vektoren zur Berechnung von ρ_i
w_1	Parameter zur Gewichtung der Teilkosten in c_{ers}

Elastische Anpassung

α	Abklingparameter einer Exponentialfunktion
λ	Wichtungsfaktor
c, c_1, c_2	Krümmungen in einem Konturpunkt
$\frac{dt}{ds}$	Differentialquotient zur Beschreibung der lokalen Dehnungsänderung
D	globale Kostenfunktion

F	lokale Kostenfunktion
K_1, K_2	Konturen
$K_1(s)$	Konturpunkt
$K_2(t(s))$	Konturpunkt
s	Bogenlänge
$t(s)$	Abbildungsfunktion der Bogenlängen zweier Konturen aufeinander

Kapitel 4

α	Exponentielles bzw. lineares Gewicht von $V_u^{I,J}$
$\alpha_{i,j,k}, \alpha_{r,s,t}$	Winkel, die die Orientierungsvektoren $o_{i,j,k}$ bzw. $o_{r,s,t}$ oder $\bar{o}_{i,j,k}$ bzw. $\bar{o}_{r,s,t}$ mit der x -Achse einschließen
β	Exponentielles bzw. lineares Gewicht von $d_u^{I,J}$
γ	Exponentielles bzw. lineares Gewicht von $v_u^{I,J}$
$\Delta_{(i,j,k),(r,s,t)}$	Distanz zwischen den Konturabschnitten $A_{i,j,k}^{l_{i,j,k}}$ und $A_{r,s,t}^{l_{r,s,t}}$
$\Delta_{(i,j)_u,k}$	Mittlere Distanz des signifikanten Konturabschnitts $s_u^{i,j}$ zum k -ten Konturabschnitt der Kontur K
$\Delta_{(i,j)_u,K}^{min}$	Distanz eines signifikanten Konturabschnitts $s_u^{i,j}$ zum dazu ähnlichsten Konturabschnitt der Kontur K
$\Delta_{(i,j)_u,(r,s)}^{min}$	Distanz eines signifikanten Konturabschnitts $s_u^{i,j}$ zum dazu ähnlichsten Konturabschnitt der Kontur $K_{r,s}$
Δ_K^{MIN}	Ähnlichster signifikanter Konturabschnitt in S zu einem Konturabschnitt der Kontur K
$\Delta Q_p^{(I,J),(R,S)}$	Teilkorrespondenzliste zugehörig zu $w_p^{(I,J),(R,S)}$
$\Delta Z_p^{(I,J),(R,S)}$	Anzahl der Elemente in $\Delta Q_p^{(I,J),(R,S)}$
$e_u^{I,J}$	Endpunktindex von $s_u^{I,J}$
Γ	Globale Distanzmatrix
$\tilde{\Gamma}$	Globale Ähnlichkeitsmatrix
$\mathcal{K}_{(i,j,k),(r,s,t)}$	Knoten im mit $\Lambda^{(i,j)(r,s)}$ bzw. $\tilde{\Lambda}^{(i,j)(r,s)}$ korrespondierenden Graph zur dynamischen Programmierung

$\lambda_u^{I,J}$	Anzahl der Konturpunkte in $s_u^{I,J}$
$\Lambda^{(i,j),(r,s)}$	Lokale Distanzmatrix für die Konturen $K_{i,j}$ und $K_{r,s}$
$\Lambda_{k,t}^{(i,j),(r,s)}$	Element der k. Zeile und t. Spalte der lokalen Distanzmatrix $\Lambda^{(i,j),(r,s)}$
$\Lambda_{k(z),t(z)}^{(i,j),(r,s)}$	Die Elemente der Matrix $\Lambda^{(i,j),(r,s)}$, die in der Korrespondenzliste $Q^{(i,j),(r,s)}$ stehen
$\tilde{\Lambda}^{(i,j),(r,s)}$	Lokale Ähnlichkeitsmatrix für die Konturen $K_{i,j}$ und $K_{r,s}$
$\tilde{\Lambda}_{k,t}^{(i,j),(r,s)}$	Element der k. Zeile und t. Spalte der lokalen Ähnlichkeitsmatrix $\tilde{\Lambda}^{(i,j),(r,s)}$
$\tilde{\Lambda}_{k(z),t(z)}^{(i,j),(r,s)}$	Die Elemente der Matrix $\tilde{\Lambda}^{(i,j),(r,s)}$, die in der Korrespondenzliste $\tilde{Q}^{(i,j),(r,s)}$ stehen
ψ	Index der $n_{r,s}$ möglichen Pfade im Graphen für dynamische Programmierung
ψ^{min}	Index des Pfades im Graphen für dynamische Programmierung mit minimaler oder maximaler Gewichtssumme entlang des Pfades
$\Psi^{(i,j),(r,s)}$	Pfad durch mit $\Lambda^{(i,j),(r,s)}$ bzw. $\tilde{\Lambda}^{(i,j),(r,s)}$ korrespondierendem Graphen mit minimaler bzw. maximaler Summe der Gewichte entlang des Pfades
$\rho_u^{i,j}$	Reliabilität von $s_u^{I,J}$
ρ^{no}	Reliabilität zugehörig zum no -ten verlässlichsten Konturabschnitt s^{no}
$\sigma_u^{I,J}$	Startpunktindex von $s_u^{I,J}$
$\varsigma_u^{I,J}$	Signifikanz von $s_u^{I,J}$
$\tau_u^{i,j}$	Konfidenz von $s_u^{I,J}$
τ^{no}	Konfidenz zugehörig zum no -ten verlässlichsten Konturabschnitt s^{no}
$\nu_u^{I,J}$	Variation, d.h. die Summe der Varianzen der Start- und Endpunktindizes der $\nu_{u,v}^{I,J}$ in Gruppe $\Omega_u^{I,J}$
Υ	Signifikanzschwelle
$\theta_p^{(I,J),(R,S)}$	Importanz von $w_p^{(I,J),(R,S)}$
$\Theta, \tilde{\Theta}$	Importanzschwelle
$\nu_{u,v}^{I,J}$	v -ter wichtiger Konturabschnitt aus Gruppe $\Omega_u^{I,J}$

ξ	Distanz, um die der Konturabschnitt $\hat{A}_{r,s,t}^{l_{r,s,t}}$ gegenüber $\hat{A}_{i,j,k}^{l_{i,j,k}}$ verschoben werden muss, damit die Summe der quadratischen Abstände korrespondierender Konturpunkte minimal wird
$\omega_c^{I,J}$	Wichtiger Konturabschnitt aus $W^{I,J}$
$\Omega_u^{I,J}$	Alle Gruppen $\Omega_u^{I,J}$ überlappender wichtiger Konturabschnitte $\nu_{u,v}^{I,J}$ der Kontur $K_{I,J}$
$\Omega_u^{I,J}$	Eine Gruppe überlappender wichtiger Konturabschnitte $\nu_{u,v}^{I,J}$ der Kontur $K_{I,J}$ in $\Omega^{I,J}$
$\mathcal{V}_u^{i,j}$	Liste mit allen $\Delta_{(i,j)_u,(r,s)}^{min}$ der Lernstichprobe L
\mathcal{W}	Wissensbasis eines Klassifikators
\mathcal{K}	Klassifikator
$a^{(I,J),(R,S)}$	Anzahl der $w_p^{(I,J),(R,S)}$ in $w^{(I,J),(R,S)}$
$A_{i,j,k}^{l_{i,j,k}}, A_{r,s,t}^{l_{r,s,t}}$	Konturabschnitt der Länge $2l_{i,j,k} + 1$ bzw. $2l_{r,s,t} + 1$ mit Konturpunkt $P_{i,j,k}$ bzw. $P_{r,s,t}$ in der Mitte der Kontur j bzw. s in Klasse i bzw. r
$\check{A}_{i,j,k}^{l_{i,j,k}}, \check{A}_{r,s,t}^{l_{r,s,t}}$	Translation des Konturabschnitts $A_{i,j,k}^{l_{i,j,k}}$ bzw. $A_{r,s,t}^{l_{r,s,t}}$
$\hat{A}_{i,j,k}^{l_{i,j,k}}, \hat{A}_{i,j,k}^{l_{i,j,k}}$	Rotation des Konturabschnitts $\check{A}_{i,j,k}^{l_{i,j,k}}$ bzw. $\check{A}_{r,s,t}^{l_{r,s,t}}$
$C^{I,J}$	Anzahl der wichtigen Konturabschnitte in $W^{I,J}$
d	Euklidischer Abstand aufeinanderfolgender Konturpunkte aller Konturen der Aufgabendomäne D
$d_u^{I,J}$	Mittlere Summe der Importanzen der $\nu_{u,v}^{I,J}$ in Gruppe $\Omega_u^{I,J}$
D	Aufgabendomäne bestehend aus Lernstichprobe L , Teststichprobe T und Rest R
$dir_{(i,j,k),(r,s,t)}$	Element der Richtungsmatrix $Dir^{(i,j),(r,s)}$
$Dir_{(i,j),(r,s)}$	Richtungsmatrix, die die den Knoten im Graphen zugeordneten Richtungen auf andere Knoten enthält
$Dir_{k,t}^{(i,j),(r,s)}$	Element der Richtungsmatrix: $Dir_{k,t}^{(i,j),(r,s)} = dir_{(i,j,k),(r,s,t)}$
Dir^*	Richtungsmatrix zugehörig zu der Matrix M^{t+} , in der der Pfad $\Psi_{\psi^{min}}^{(i,j),(r,s)}$ gefunden wurde.
E	Maximal zulässige Distanz für Konturabschnitte gleicher Länge
$end_c^{I,J}$	Endpunktindex von $W_c^{I,J}$ bzw. $\omega_c^{I,J}$
$end_p^{I,J}$	Endpunktindex von $w_p^{(I,J),(R,S)}$

$\frac{end_{u,v}^{I,J}}{end_{u,v}^{I,J}}$	Endpunktindex von $\nu_{u,v}^{I,J}$
$\overline{end_{u,v}^{I,J}}$	(Gewichteter) Mittelwert der Endpunktindices der $\nu_{u,v}^{I,J}$ in $\Omega_u^{I,J}$
$\overline{end_{u,v}^{I,J}}$	Varianz der Endpunktindices der $\nu_{u,v}^{I,J}$ in $\Omega_u^{I,J}$
$F_p^{(I,J),(R,S)}$	Anzahl der Konturpunkte von $w_p^{(I,J),(R,S)}$
$F_{u,v}^{I,J}$	Anzahl der Konturpunkte von $\nu_{u,v}^{I,J}$
$g^{(i,j),(r,s)}$	Globale Distanz zwischen den Konturen $K_{i,j}$ und $K_{r,s}$ basierend auf l_{const}
$\tilde{g}^{(i,j),(r,s)}$	Globale Ähnlichkeit zwischen den Konturen $K_{i,j}$ und $K_{r,s}$ basierend auf $l_{(i,j,k),(r,s,t)}^{max}$
\hat{g}	Klassifikationsergebnis
\hat{g}^{no}	Klasse, zu der s^{no} gehört
G_i, G_r	Klasse i bzw. Klasse r
$k(z), \tilde{k}(z)$	Mit z parametrisierter Konturpunktindex k bzw. \tilde{k} der Kontur $K_{i,j}$
K	Eine zu klassifizierende Kontur aus T
$K_{i,j}, K_{r,s}$	Kontur j in Klasse i , Kontur s in Klasse r
l_{const}	Für alle Konturen in D werden allen Konturpunkten Konturabschnitte der gleichen Länge $2l_{const} + 1$ zugeordnet
$l_{i,j,k}, l_{r,s,t}$	Ein Konturabschnitt besitzt $2l_{i,j,k} + 1$ bzw. $2l_{r,s,t} + 1$ Konturpunkte mit Punkt $P_{i,j,k}$ bzw. $P_{r,s,t}$ in der Mitte
$l_{(i,j,k),(r,s,t)}^{max}$, $l_{r,s,t}^{max}$	Maximales $l_{i,j,k} = l_{r,s,t}$, so dass Distanz $\Delta_{(i,j,k),(r,s,t)}$ kleiner oder gleich einer maximal zulässigen Distanz E bleibt
L	Lernstichprobe
L_{min}	Konturpunktanzahlschwellwert für signifikantere Konturabschnitte
m_i, m_r	Anzahl der Konturen in Klasse i bzw. r
m_i^*	Anzahl der $\Delta_{(i,j)_u,(r,s)}^{min}$ unter den ersten m_i minimalsten $\Delta_{(i,j)_u,(r,s)}^{min}$, die mit der gleichen Klasse i wie $s_u^{i,j}$ korrespondieren.
m^*	Anzahl der Konturen in Lernstichprobe L
M	Hilfsmatrix zur einfacheren Darstellung von $\Lambda_{(i,j),(r,s)}$
$M_{k,t}$	Element der k . Zeile und t . Spalte der Matrix M
M^t	In Spaltenrichtung rotierte Matrix M
M^{t+}	Um eine Spalte und eine Zeile erweiterte Matrix M^t für die dynamischen Programmierung

n_K	Anzahl der Konturpunkte der Kontur K
$n_{i,j}, n_{r,s}$	Anzahl der Konturpunkte bzw. Konturabschnitte von Kontur j bzw. s in Klasse i bzw. r
N	Anzahl der Klassen in L
$O_{i,j,k}, O_{r,s,t}$	Sogenannter absoluter Orientierungsvektor, der vom Anfangspunkt zum Endpunkt des Konturabschnitts $A_{i,j,k}^{l_{i,j,k}}$ bzw. $A_{r,s,t}^{l_{r,s,t}}$ zeigt
$\bar{O}_{i,j,k}, \bar{O}_{r,s,t}$	Sogenannter mittlerer Orientierungsvektor, der vom Schwerpunkt der ersten $l_{i,j,k}$ bzw. $l_{r,s,t}$ Konturpunkte zum Schwerpunkt der letzten $l_{i,j,k}$ bzw. $l_{r,s,t}$ Konturpunkte des Konturabschnitts $A_{i,j,k}^{l_{i,j,k}}$ bzw. $A_{r,s,t}^{l_{r,s,t}}$ zeigt
$P_{i,j,k}, P_{r,s,t}$	Konturpunkt k der Kontur j in Klasse i , Konturpunkt t der Kontur s in Klasse r
$P_{i,j,k}^-, P_{r,s,t}^-$	Schwerpunkt der ersten $l_{i,j,k}$ bzw. $l_{r,s,t}$ Konturpunkte der Konturabschnitte $A_{i,j,k}^{l_{i,j,k}}$ bzw. $A_{r,s,t}^{l_{r,s,t}}$
$P_{i,j,k}^+, P_{r,s,t}^+$	Schwerpunkt der letzten $l_{i,j,k}$ bzw. $l_{r,s,t}$ Konturpunkte der Konturabschnitte $A_{i,j,k}^{l_{i,j,k}}$ bzw. $A_{r,s,t}^{l_{r,s,t}}$
$\check{P}_{i,j,k}, \check{P}_{r,s,t}$	Versobener Konturpunkt $P_{i,j,k}$ bzw. $P_{r,s,t}$
$\hat{P}_{i,j,k}, \hat{P}_{r,s,t}$	Rotierter Konturpunkt $\check{P}_{i,j,k}$ bzw. $\check{P}_{r,s,t}$
$P_{u,w}^{I,J}$	w -ter Konturpunkt des signifikanten Konturabschnitts $s_u^{I,J}$
P_k	k -ter Konturpunkt der Kontur K
q_z	Ein Element der Korrespondenzliste $Q^{(i,j),(r,s)}$ mit einem Indexpaar von 2 korrespondierenden Konturpunkten
\tilde{q}_z	Ein Element der Korrespondenzliste $\tilde{Q}^{(i,j),(r,s)}$ mit einem Indexpaar von 2 korrespondierenden Konturpunkten
$Q^{(i,j),(r,s)}$	Korrespondenzliste mit den Indizes der korrespondierenden Punkte der Konturen $K_{i,j}$ und $K_{r,s}$ mit minimalen Kosten für die Deformation basierend auf $\Lambda^{(i,j),(r,s)}$
$\tilde{Q}^{(i,j),(r,s)}$	Korrespondenzliste mit den Indizes der korrespondierenden Punkte der Konturen $K_{i,j}$ und $K_{r,s}$ mit minimalen Kosten für die Deformation basierend auf $\tilde{\Lambda}^{(i,j),(r,s)}$
R	Objekte der Aufgabendomäne ohne L und T
$s_u^{I,J}$	Signifikanter Konturabschnitt der Kontur $K_{I,J}$

s^{no}	no -ter verlässlichster Konturabschnitt der Konturen in L bzw. \mathcal{W}
S	Liste aller signifikanten Konturabschnitte $s_u^{i,j}$ aller Konturen $K_{i,j}$ in L
$S^{I,J}$	Alle signifikanten Konturabschnitte $s_u^{I,J}$ der Kontur $K_{I,J}$
$sta_c^{I,J}$	Startpunktindex von $W_c^{I,J}$ bzw. $\omega_c^{I,J}$
$sta_p^{I,J}$	Startpunktindex von $w_p^{(I,J),(R,S)}$
$sta_{u,v}^{I,J}$	Startpunktindex von $\nu_{u,v}^{I,J}$
$\overline{sta}_{u,v}^{I,J}$	(Gewichteter) Mittelwert der Startpunktindices der $\nu_{u,v}^{I,J}$ in $\Omega_u^{I,J}$
$\overline{\overline{sta}}_{u,v}^{I,J}$	Varianz der Startpunktindices der $\nu_{u,v}^{I,J}$ in $\Omega_u^{I,J}$
$sum_{(i,j,k),(r,s,t)}$	Zwischensumme, die einem Knoten im Graphen für die dynamische Programmierung zugeordnet wird
$t(z), \tilde{t}(z)$	Mit z parametrisierter Konturpunktindex t bzw. \tilde{t} der Kontur $K_{r,s}$
T	Teststichprobe
$U_{I,J}$	Anzahl der Gruppen $\Omega_u^{I,J}$ in der Kontur $K_{I,J}$
$U_{I,J}^Y$	Anzahl der signifikanteren Konturabschnitte der Kontur $K_{I,J}$
$V_u^{I,J}$	Anzahl wichtiger Konturabschnitte in $\Omega_u^{I,J}$
$w_{(I,J),(R,S)}$	Alle wichtigen Konturabschnitte $w_p^{(I,J),(R,S)}$ der Kontur $K_{I,J}$ bzgl. Kontur $K_{R,S}$ für $I \neq R$
$\tilde{w}_{(I,J),(R,S)}$	Alle wichtigen Konturabschnitte $\tilde{w}_p^{(I,J),(R,S)}$ der Kontur $K_{I,J}$ bzgl. Kontur $K_{R,S}$ für $I \neq R$
$w_p^{(I,J),(R,S)}$	Wichtiger Konturabschnitt der Kontur $K_{I,J}$ bzgl. Kontur $K_{R,S}$ für $I \neq R$
$w_{p,q}^{(I,J),(R,S)}$	q -ter Konturpunkt von $w_p^{(I,J),(R,S)}$
$w_{p,q_x}^{(I,J),(R,S)}$	x -Koordinate des q -ten Konturpunktes von $w_p^{(I,J),(R,S)}$
$w_{p,q_y}^{(I,J),(R,S)}$	y -Koordinate des q -ten Konturpunktes von $w_p^{(I,J),(R,S)}$
$W^{I,J}$	Alle wichtigen Konturabschnitte $w_p^{(I,J),(r,s)}$ bzw. $\omega_c^{I,J}$ der Kontur $K_{I,J}$ bzgl. aller Konturen $K_{r,s}$ aus anderen Klassen $I \neq r$
$W_c^{I,J}$	c -ter wichtiger Konturabschnitt in $W^{I,J}$
$W_{c,d}^{I,J}$	d -ter Konturpunkt von $W_c^{I,J}$
$W_{c,d_x}^{I,J}$	x -Koordinate von $W_{c,d}^{I,J}$
$W_{c,d_y}^{I,J}$	y -Koordinate von $W_{c,d}^{I,J}$

$x_{i,j,k}, x_{r,s,t}$	x -Koordinate des Punktes $P_{i,j,k}$ bzw. $P_{r,s,t}$
$\tilde{x}_{i,j,k}, \tilde{x}_{r,s,t}$	x -Koordinate des Punktes $\tilde{P}_{i,j,k}$ bzw. $\tilde{P}_{r,s,t}$
$\hat{x}_{i,j,k}, \hat{x}_{r,s,t}$	x -Koordinate des Punktes $\hat{P}_{i,j,k}$ bzw. $\hat{P}_{r,s,t}$
$\bar{x}_{i,j,k}, \bar{x}_{r,s,t}$	Mittelwert der x -Koordinaten der Punkte des Konturabschnitts $\hat{A}_{i,j,k}^{l_{i,j,k}}$ bzw. $\hat{A}_{r,s,t}^{l_{r,s,t}}$
$x_{i,j,k}^+, x_{r,s,t}^+$	x -Koordinate des Punktes $P_{i,j,k}^+$ bzw. $P_{r,s,t}^+$
$x_{i,j,k}^-, x_{r,s,t}^-$	x -Koordinate des Punktes $P_{i,j,k}^-$ bzw. $P_{r,s,t}^-$
x_k	x -Koordinate des Punktes P_k
$\overline{x_{k+w}}$	Mittelwert der x -Koordinaten von $\lambda_u^{i,j}$ Punkten des k -ten Konturabschnitts von K
$\frac{x_{u,w}^{I,J}}{x_{u,w+1}^{I,J}}$	x -Koordinate des Punktes $P_{u,w}^{I,J}$
	Mittelwert der x -Koordinaten der Punkte $P_{u,w}^{I,J}$ in $s_u^{I,J}$
$y_{i,j,k}, y_{r,s,t}$	y -Koordinate des Punktes $P_{i,j,k}$ bzw. $P_{r,s,t}$
$\tilde{y}_{i,j,k}, \tilde{y}_{r,s,t}$	y -Koordinate des Punktes $\tilde{P}_{i,j,k}$ bzw. $\tilde{P}_{r,s,t}$
$\hat{y}_{i,j,k}, \hat{y}_{r,s,t}$	y -Koordinate des Punktes $\hat{P}_{i,j,k}$ bzw. $\hat{P}_{r,s,t}$
$\bar{y}_{i,j,k}, \bar{y}_{r,s,t}$	Mittelwert der y -Koordinaten der Punkte des Konturabschnitts $\hat{A}_{i,j,k}^{l_{i,j,k}}$ bzw. $\hat{A}_{r,s,t}^{l_{r,s,t}}$
$y_{i,j,k}^+, y_{r,s,t}^+$	y -Koordinate des Punktes $P_{i,j,k}^+$ bzw. $P_{r,s,t}^+$
$y_{i,j,k}^-, y_{r,s,t}^-$	y -Koordinate des Punktes $P_{i,j,k}^-$ bzw. $P_{r,s,t}^-$
y_k	y -Koordinate des Punktes P_k
$y_{u,w}^{I,J}$	y -Koordinate des Punktes $P_{u,w}^{I,J}$
z_p^{sta}	Index des ersten Elementes in $\Delta Q_p^{(I,J),(R,S)}$
z_p^{end}	Index des letzten Elementes in $\Delta Q_p^{(I,J),(R,S)}$
$Z^{(i,j),(r,s)}$	Anzahl der Elemente in der Korrespondenzliste $Q^{(i,j),(r,s)}$
$\tilde{Z}^{(i,j),(r,s)}$	Anzahl der Elemente in der Korrespondenzliste $\tilde{Q}^{(i,j),(r,s)}$

Anhang B

Glossar

4er-Nachbarschaft	Als 4er-Nachbarn eines bestimmten \rightarrow <i>Pixels</i> bezeichnet man die Pixel, die nördlich, südlich, westlich oder östlich dieses bestimmten Pixels auf einem rechteckigen oder quadratischen Pixelraster liegen.
8er-Nachbarschaft	Als 8er-Nachbarn eines bestimmten \rightarrow <i>Pixels</i> bezeichnet man die Pixel, die nördlich, südlich, westlich, östlich, nordwestlich, nordöstlich, südwestlich oder südöstlich dieses bestimmten Pixels auf einem rechteckigen oder quadratischen Pixelraster liegen.
Φ-s-Funktion	Es werden die aufkumulierten Krümmungswinkel gegenüber der aufkumulierten Bogenlänge eines Polygons oder Polygonzuges aufgetragen.
Active Vision	Englischer Fachausdruck für die Fähigkeit eines Bildverarbeitungssystems die inneren Parameter seiner Kamera(s) (Fokus, \rightarrow <i>Zoom</i> , Blende) und deren Position, Blickrichtung und Bewegung zu steuern.
A/D-Wandler	Analog/Digital-Wandler. Eine kontinuierliche Messgröße wird in eine \rightarrow <i>diskrete</i> Messgröße gewandelt, so dass ein Digitalrechner die Werte übernehmen kann.
affine Transformation	Unter einer affinen Transformation werden die Transformationen verstanden, die die Kollinearität (d.h. alle Punkte auf einer Linie liegen auch nach der Transformation auf einer korrespondierenden Linie) und die Distanzverhältnisse (z.B. der Mittelpunkt einer Linie ist auch nach der Transformation der Mittelpunkt einer korrespondierenden Linie) bewahren. Zu den affinen Transformationen gehören z.B. die Skalierung, die Translation und die Rotation.

agglomerative Clusteranalyse	Ausgehend von einer Klasseneinteilung, in der jedem Objekt eine eigene Klasse zugeordnet ist, werden die Klassen sukzessive zu umfangreicheren Klassen fusioniert.
attributierter relationaler Graph	Hierbei handelt es sich um einen \rightarrow <i>Graph</i> , der aus Knoten besteht, die mit Hilfe der sie verbindenden Kanten die relativen Positionen von beispielsweise Konturabschnitten oder Flächensegmenten repräsentieren. Die Attribute sind dann Eigenschaften der Knoten oder Kanten, wie z.B. die Fläche der Flächensegmente oder die Abstände der Schwerpunkte der Flächensegmente.
Aufgabendomäne	Menge aller \rightarrow <i>Objekte</i> , die für eine bestimmte Aufgabenstellung relevant sind.
Ausbeute	Verhältnis von richtig in eine bestimmte \rightarrow <i>Klasse</i> eingeordnete Testmuster(-konturen) und Gesamtanzahl der Testmuster(-konturen) in dieser \rightarrow <i>Klasse</i> .
axiales Moment	Ein \rightarrow <i>Moment</i> , das sich ausschließlich auf eine Koordinatenachse bezieht.
Backpropagation	Bei <i>Backpropagation</i> handelt es sich um ein Lernverfahren für mehrstufige vorwärtsgerichtete \rightarrow <i>künstliche neuronale Netze</i> , das sich mathematisch als Gradientenabstiegsverfahren des Netzfehlers herleiten lässt.
bijektiv, Bijektion	Unter einer Bijektion versteht man eine 1-zu-1-Abbildung, die alle Werte aus einem Definitionsbereich auf alle Werte im Funktionsbereich abbildet, so dass dort keine Werte übrigbleiben und umgekehrt.
Bounding Box	Engl. für ein umgebendes Rechteck, das gerade groß genug ist, ein 2D-Objekt komplett zu beinhalten. Typischerweise sind die Seiten des Rechtecks dabei parallel zu den Koordinatenachsen.
Clusteranalyse	Die Clusteranalyse ist ein mathematisches Werkzeug zur systematischen Unterteilung einer i. allg. großen Menge ungeordneter Objekte in kleinere und homogene \rightarrow <i>Klassen</i> (Gruppen) aufgrund von Ähnlichkeiten zwischen den Objekten.
Computer Vision	Der wissenschaftliche Bereich, der sich damit beschäftigt, welche visuelle Information für eine bestimmte Aufgabe notwendig ist und wie diese Information aus Bildern extrahiert werden kann.
deduktiv	Das Besondere, Einzelne vom Allgemeinen ableitend. Gegensatz von \rightarrow <i>induktiv</i> .

Diskretisierung	Einen kontinuierlichen Zahlen- oder Wertebereich aufteilen in voneinander getrennte, einzelne, gesonderte Zahlen oder Werte.
dynamische Programmierung	Der Begriff <i>dynamische Programmierung</i> ist etwas irreführend. Es handelt sich dabei um eine Methode zur Lösung komplexer Optimierungsprobleme, die von Richard Bellman [Bel57] entwickelt wurde. Wo die Werkzeuge der Analysis nicht mehr ausreichen, sucht die dynamische Programmierung nach dem Maximum oder dem Minimum einer Funktion. Ein Gesamtproblem wird dabei in Teilprobleme zerlegt und, ohne das Gesamtproblem aus den Augen zu verlieren, gelöst. Siehe auch Anhang C.
Exzentrizität	Verhältnis von minimalem zu maximalem \rightarrow <i>polarem Abstand</i> .
Fourieranalyse	J.B. Fourier zeigte 1822 [Fou22], dass sich periodische Funktionen vollständig aus der Summe von Schwingungen aufbauen lassen. Die Grund- und Oberschwingungen nennt man Fourierkomponenten. Die Amplituden, Phasen und Frequenzen dieser Schwingungen lassen sich zur Beschreibung der periodischen Funktion heranziehen.
Fourierdeskriptor	Fourierdeskriptoren sind die bei einer \rightarrow <i>Fourieranalyse</i> berechneten Amplituden eines Spektrums diskreter Frequenzen.
Frame	Ein Frame ist der komplette Scan eines optischen Bildes und besteht aus M Linien mit jeweils N Pixeln.
Framegrabber	Ein Gerät zum Einlesen eines einzelnen \rightarrow <i>Frames</i> eines analogen Videosignals (von einer Videokamera oder einem Videoabspielgerät), der dann als digitales Bild gespeichert wird.
Gaußfilter	Unter einem eindimensionalen Gaußfilter wird die Gaußfunktion $f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma^2}}$ mit σ als sogenannter Halbwertsbreite verstanden, die zum Filtern von Signalen herangezogen wird. Der Gaußfilter hat die Eigenschaft, Störungen zu vermindern und dabei gleichzeitig große Sprünge in der Funktion nicht allzu sehr zu glätten.
gemischtes Moment	Ein \rightarrow <i>Moment</i> , das sich auf mehrere Koordinatenachsen bezieht.

Graph	Ein Graph G ist ein Paar $G = (V, E)$ aus einer Menge $V \neq \emptyset$ und einer Menge E von zweielementigen Teilmengen von V . Die Elemente von V heißen Punkte (auch Knoten oder Ecken), die Elemente von E Kanten (auch Bögen). Für eine Kante $e = \{a, b\}$ heißen a und b die Endpunkte von e (nach [Jun94]). Graphen werden oft durch Zeichnungen veranschaulicht. Die Knoten werden dabei durch Punkte markiert und die Kanten werden mit Hilfe von Kurven oder Geradenstücken dargestellt, die die Knoten verbinden.
Graph-Isomorphismus-Problem	Das Graph-Isomorphismus-Problem (GIP) behandelt die Fragestellung, ob 2 Graphen „gleich“ sind. 2 Graphen werden als „gleich“ angesehen, wenn sie $\rightarrow isomorph$ sind.
Greebles	Greebles sind von Michael Tarr geschaffene künstliche Objekte, die sich bzgl. der Erkennung ähnlich verhalten wie Gesichter.
Hauptachse	Unter den Hauptachsen eines Objektes versteht man diejenigen Achsen, auf die bezogen die zweiten $\rightarrow zentrierten$, $\rightarrow axialen Momente$ minimal bzw. maximal werden.
heuristisch	heuristisch sind solche Verfahren, Grundsätze oder Methoden, die "[...] etwas zur Erkenntniserweiterung beitragen, ohne selbst die Sicherheit der gewonnenen Erkenntnisse begründen zu können", nach [Rit74].
Hooke'sches Federgesetz	Wird eine elastische Feder gespannt, so ist die Kraft F die dazu notwendig ist proportional zur Auslenkung s der Feder aus ihrem Ruhezustand heraus. Die Proportionalitätskonstante D wird als Federkonstante bezeichnet. D.h. es gilt: $F = Ds$. Die Energie E , die eine gespannte Feder speichert, ergibt sich dann folgendermaßen: $E = \int_s F ds = \frac{1}{2}Ds^2$.
ICR	Abk. für intelligent character recognition. Das ist der englische Ausdruck für intelligente Buchstabenerkennung. Eine ICR-Anwendung erkennt Buchstaben flexibler als eine $\rightarrow OCR$ -Anwendung. Sie arbeitet mit Beschreibungen der grundlegenden Merkmale der Buchstaben. Ein i beispielsweise kann beschrieben werden durch einen senkrechten Strich über dem ein Punkt schwebt.
Image Retrieval	Engl. für das Suchen von Bildern in Datenbanken oder im Internet. Dabei gibt der Benutzer ein ihn interessierendes Beispielbild vor und das <i>Image Retrieval</i> -System sucht nach ähnlichen Bildern.

Importanz	Zahlenwert für die Wichtigkeit eines wichtigen Konturabschnittes.
Importanzschwelle	Schwellwert für die Distanz (Ähnlichkeit) eines korrespondierenden Konturpunktpaars von 2 verglichenen Konturen, um wichtiger Konturpunkt eines wichtigen Konturabschnitts zu werden.
induktiv	Von beobachteten Einzelfällen (Beispielen) auf das Allgemeine schließend oder allgemeine Regeln oder Gesetzmäßigkeiten ableitend. Gegensatz von \rightarrow <i>deduktiv</i> .
infinitesimal	Zu einem Grenzwert hin gegen Null strebend; unendlich klein werdend.
Interpolation	Bestimmung von Zwischenwerten innerhalb der Abtastwerte einer Funktion oder zwischen 2 Funktionen.
isomorph	\rightarrow <i>Isomorphie</i> aufweisend.
Isomorphie	Gleichheit der Gestalt. Zwei \rightarrow <i>Graphen</i> sind isomorph, wenn sie die gleiche Anzahl an Knoten und Kanten besitzen und eine \rightarrow <i>bijektive</i> Abbildung existiert, die die Knoten des einen Graphen auf die Knoten des anderen Graphen abbildet, so dass auch die korrespondierenden Kanten und Nicht-Kanten aufeinander abgebildet werden.
Kettencode (engl. chain code)	Der Kettencode kodiert die Richtung, in der sich der nächste Konturpunkt einer Konturpunktsequenz befindet.
Klasse	Diejenige Untermenge an \rightarrow <i>Objekten</i> einer \rightarrow <i>Aufgabendomäne</i> , die sich ähnlich sind.
Klassifikation	Ein \rightarrow <i>Objekt</i> einer \rightarrow <i>Klasse</i> zuordnen.
Klassifikator	Ein realisiert die \rightarrow <i>Klassifikation</i> , führt diese also aus.
kollinear	2 oder mehr Punkte sind kollinear, wenn sie auf derselben Geraden liegen.
Kompaktheit	Die Kompaktheit eines 2D-Objektes ist definiert als: $Kompaktheit = Umfang^2 / (4\pi \times Fläche)$. Die Kompaktheit ist im Falle eines Kreises 1.0 und wird umso größer, je länger der Umriss eines Bereiches im Verhältnis zu seinem Flächeninhalt ist.
Konfidenz	Die Konfidenz ist der einem \rightarrow <i>signifikanten Konturabschnitt</i> zugeordnete Schwellwert für die Maximaldistanz (Minimalähnlichkeit) zu dem diesem \rightarrow <i>signifikanten Konturabschnitt</i> ähnlichsten Konturabschnitt einer zu klassifizierenden Kontur, um die Kontur der Klasse des \rightarrow <i>signifikanten Konturabschnitts</i> zuzuordnen.
konstitutiv	Das Wesen von etwas ausmachend; grundlegend; wesentlich bestimmend.

Kostenfunktion	Im Bereich der Ähnlichkeitsanalyse von \rightarrow <i>Objekten</i> ist damit eine Funktion gemeint, mit der die <i>Unähnlichkeit</i> zweier Objekte festgestellt werden kann. Eine Kostenfunktion definiert die Gewichtung einzelner Kosten. Unter Kosten versteht man eine Zahl für den Aufwand, der für eine bestimmte Operation benötigt wird. Kostenfunktionen werden minimiert, um den Gesamtaufwand so gering wie möglich zu halten.
Krümmungskreis	Unter einem Krümmungskreis einer Kurve in einem Punkt wird derjenige Kreis verstanden, der an diesem Punkt mit der Kurve eine Berührung von mindestens 2. Ordnung aufweist (näheres siehe in Abschnitt 2.2.3).
künstliche neuronale Netze	Künstliche neuronale Netze (KNN) sind informationsverarbeitende Systeme, die sich aus einer großen Anzahl einfacher künstlicher Einheiten (künstliche Neuronen oder auch Zellen) zusammensetzen, die sich Informationen über gerichtete Verbindungen schicken.
Lernstichprobe	Diejenige Menge an \rightarrow <i>Objekten</i> aus der \rightarrow <i>Aufgabendomäne</i> , die zum Einlernen eines \rightarrow <i>Klassifikators</i> herangezogen werden.
Machine Vision	Machine Vision (Maschinelles Sehen) ist die Fähigkeit eines Computers zu "sehen". Bilder realer Szenen werden optisch, berührungslos und automatisch erfasst und interpretiert, um Informationen zu erhalten und/oder Maschinen bzw. Prozesse zu steuern.
Maschinelles Lernen	Im Zentrum des Forschungsgebietes <i>Maschinelles Lernen</i> (engl. <i>machine learning</i>) steht die Frage, wie Lern- und Entdeckungsprozesse durch geeignete Rechenverfahren und Softwaresysteme modelliert, unterstützt und automatisiert werden können.
MAT	MAT ist die Abkürzung für <i>Medial Axis Transform</i> und bezeichnet das "Zusammenziehen" der Außenkontur eines \rightarrow <i>Objektes</i> bis eine Strichzeichnung, auch Skelett genannt, übrig bleibt, die möglichst genau in der Mitte zwischen gegenüberliegenden Konturabschnitten liegt.
Merkmal	Unter Merkmal wird i. allg. ein einzelner Zahlenwert verstanden, der eine bestimmte Eigenschaft eines Musters ausdrückt. Dies könnte z.B. die von einer Außenkontur eingeschlossene Fläche sein. Aber Achtung: der Terminus Merkmal wird auch in vielen anderen Zusammenhängen benutzt.

Merkmalsextraktion	Berechnung von \rightarrow <i>Merkmalen</i> für ein \rightarrow <i>Objekt</i> zur Vereinfachung der Objektbeschreibung. Es kann sich dabei um einfache Merkmale wie Ecken und Kanten, aber auch um abstraktere Merkmale wie z.B. das Längen-Breitenverhältnis handeln.
Merkmalsraum	\rightarrow <i>Objekte</i> können durch \rightarrow <i>Merkmalsvektoren</i> repräsentiert werden. Besteht dieser Vektor aus $n \rightarrow$ <i>Merkmalen</i> , so kann das Objekt als Punkt in einem n -dimensionalen Merkmalsraum eingetragen werden. Punkte im Merkmalsraum die eng zusammenliegen, also einen kompakten Bereich abdecken, können als ähnlich bezogen auf ihre Merkmale angesehen werden.
Merkmalsselektion	Auswahl einer möglichst optimalen Untermenge aus einer gegebenen Menge von Merkmalen zur korrekten \rightarrow <i>Klassifikation</i> . Die Merkmale werden dabei durch ein geeignetes Gütemaß bewertet. Man extrahiert dabei zunächst mehr Merkmale als wahrscheinlich tatsächlich gebraucht werden und wählt anhand des Gütemaßes die besten aus.
Merkmalsvektor	Ein Merkmalsvektor fasst alle \rightarrow <i>Merkmale</i> eines Musters in einem Vektor zusammen.
Metrik	Eine Metrik ist eine nicht negative Funktion, die die Distanz zwischen 2 Punkten beschreibt.
Moment	Mit einem Moment ordnet man dem Zusammenhang von Koordinaten eines Körpers und dem zu den Koordinaten gehörenden Funktionswert einen eindeutigen Wert zu. D.h. der Funktionswert wird mit seiner "Lage" gewichtet. Im 2-dimensionalen, diskreten Fall ergibt sich folgender Zusammen für ein Moment $(p + q)$ ter Ordnung: $m_{pq} = \sum_x \sum_y x^p y^q g(x, y).$
Morphismus multivariate Datenanalyse	Strukturähnliche Abbildung. Da- Im Gegensatz zur univariaten oder bivariaten Datenanalyse, in der nur eine oder zwei Variablen betrachtet werden, beziehen multivariate Verfahren mehrere Variablen in ihre Analyse mit ein.
nächster Nachbar (engl. nearest neighbour)	Das \rightarrow <i>Objekt</i> , das den kleinsten Abstand (oft euklidischer Abstand) zu einem \rightarrow <i>Objekt</i> aufweist.

NP-vollständig	NP-Probleme sind Probleme, für die es einen nichtdeterministischen Algorithmus mit polynomialem Aufwand gibt (d.h. die Überprüfung, ob eine Lösung gefunden wurde, ist polynomial). NP-vollständige Probleme sind dann solche Probleme in NP, auf die jedes andere Problem aus NP mit polynomialem Aufwand reduziert werden kann. NP-vollständige Algorithmen finden nur für kleine Problemstellungen eine optimale Lösung in angemessener Zeit.
Objekt	Gegenstand, Sache oder Person der realen Welt, der/die unabhängig von einem menschlichen oder einem maschinellen Bewusstsein existiert.
OCR	Abk. für Optical Character Recognition. Das ist der englische Ausdruck für optische Buchstabenerkennung. Definitionsgemäß kann eine OCR-Anwendung nur Texte bestimmter Zeichensätze erkennen, da sie die zu erkennenden Buchstaben im Gegensatz zur \rightarrow ICR nur mit den Buchstaben eben dieser Zeichensätze vergleicht.
PE	Abkürzung für Pixeleinheit. Eine Pixeleinheit ist der Abstand von 2 horizontal oder vertikal benachbarten Pixeln bei quadratisch angenommenen Pixeln.
Phänomen	Erscheinung, Erscheinungsform, das sich den Sinnen Zeigende.
Pixel	Abkürzung für Picture Element: Ein Rasterpunkt eines Bildes, dem eine quantisierte Helligkeitsinformation zugeordnet ist. Nicht zu verwechseln mit einem \rightarrow Sensel.
polarer Abstand	Der polare Abstand ist der Abstand der Außenkontur vom Schwerpunkt dieser Kontur. Jedem Konturpunkt wird in diesem Zusammenhang eine Masse zugeordnet. Werden alle Konturpunkte gleichbehandelt, so werden diesen die gleichen Massen zugeteilt.
probabilistisch Quantisierung	Die Wahrscheinlichkeit berücksichtigend. \rightarrow <i>Diskretisierung</i> des Wertebereiches (der Helligkeitswerte) eines Bildes.
Rasterung	\rightarrow <i>Diskretisierung</i> eines Bildes im Ortsbereich.

Reinheit	Verhältnis von richtig in eine bestimmte → <i>Klasse</i> eingeordnete Testmuster(-konturen) und Gesamtanzahl der Testkonturen, die in diese → <i>Klasse</i> eingeordnet wurden. Die Reinheit ist bei bestimmten Sortieraufgaben ein wichtiges Gütemaß. So wird beispielsweise in Kuhnert [Kuh98] eine Sortierung von Verpackungsmaterialien aus Kunststoff vorgestellt, deren sortierte Fraktionen einem Recyclingprozess zugeführt werden. Hier ist ganz entscheidend, dass die Reinheit der einzelnen Fraktionen hoch, d.h. die Anzahl der Fehlklassifikationen klein ist, um die einzelnen Fraktionen überhaupt recyceln zu können.
Reliabilität	Zahlenwert für die Verlässlichkeit eines → <i>signifikanten Konturabschnitts</i> für eine korrekte Klassifikation.
Robot Vision	Die zentrale Aufgabe eines Robot-Vision-Systems ist die Identifikation von Objekten anhand ihrer Geometrie. Dies beinhaltet z.B. die Positionsbestimmung, die Ermittlung des Greifpunktes und die präzise Positionierung einzelner Bauteile zur Weiterverarbeitung in den jeweiligen Maschinen oder Montageanlagen (nach [VDM99b]).
Segmentierung	Heraustrennen eines → <i>Objektes</i> aus seinem Hintergrund. Die Segmentierungsverfahren teilen sich grob in 2 unterschiedliche Gruppen auf, die Bereichs- und die Kontursegmentierungsverfahren. In der Bereichssegmentierung werden möglichst homogene Bereiche, d.h. Bereiche mit ähnlichen Farb- bzw. Grauwerten zusammengefasst. In der Kontursegmentierung werden im Bild benachbarte Pixel mit hohem Grauwertunterschied gesucht. Benachbarte Stellen mit hohem Grauwertunterschied werden zu Konturen zusammengefasst. Es existieren aber auch zahlreiche Mischverfahren (sog. hybride Verfahren), die Anteile aus beiden Gruppen in sich vereinigen.
Sensel	Abkürzung für Sensorelement: Ein Element eines lichtempfindlichen Sensors, der das auf seine Fläche einfallende Licht integriert.

Serviceroboter	”Ein Serviceroboter ist eine frei programmierbare Bewegungseinrichtung, die teil- oder vollautomatisch Dienstleistungen verrichtet. Dienstleistungen sind dabei Tätigkeiten, die nicht der direkten industriellen Erzeugung von Sachgütern, sondern der Verrichtung von Leistungen für Menschen und Einrichtungen dienen” (Definition nach Fraunhofer-IPA 1994). Erweitert wurde diese Definition 1997 von der International Federation of Robotics (IFR) mit dem Zusatz: ”They are mobile or manipulative or combinations of both.”
Shape analysis	Engl. für die Analyse der Form bzw. des Umrisses eines 2D-Objekts.
Shape number	Engl. für Formzahl. Die <i>shape number</i> ist ein geordneter Krümmungscode einer Kontur mit dem Konturen verglichen werden können (siehe Abschnitt 2.2.1).
signifikanter Konturabschnitt	Ein signifikanter Konturabschnitt einer Kontur aus einer bestimmten Klasse unterscheidet sich von allen anderen Konturabschnitten der Konturen aus anderen Klassen. Er wird aus sich überlappenden → <i>wichtigen Konturabschnitten</i> synthetisiert.
Signifikanz	Die Signifikanz ist ein Zahlenwert, der einem → <i>signifikanten Konturabschnitt</i> zugeordnet wird. Die Signifikanz setzt sich zusammen aus der Anzahl, der Kompaktheit und den Importanzen der sich überlappenden → <i>wichtigen Konturabschnitte</i> , die zu dem → <i>signifikanten Konturabschnitt</i> synthetisiert wurden und der Konturpunktanzahl des → <i>signifikanten Konturabschnitts</i> .
strukturelle Mustererkennung	Der strukturellen Mustererkennung liegt das Prinzip zu Grunde, ein komplexes Muster durch ein hierarchisches Zusammenfügen von einfacheren Submustern zu beschreiben. Dies geschieht durch Auswahl von geeigneten Muster-Primitiven und von Regeln, wie die Primitiven zu einem komplexeren Muster zusammengesetzt sind. Die Regeln sind die Grammatik. D.h. ein Muster wird durch relationale Strukturen repräsentiert, deren Knoten Unterstrukturen und deren Linien Relationen zwischen den Knoten sind, die sie verbinden. Siehe auch → <i>syntaktische Mustererkennung</i> .

Subpixelvermessung	Eine Objektkante muss bei der → <i>Segmentierung</i> von Objekten aus einem gerasterten Bild heraus an einer bestimmten Pixelgrenze angenommen werden, obwohl sie tatsächlich mitten durch einen Bildpunkt verläuft. Durch Interpolation des Grauwertverlaufes (bei Kenntnis des wahren Ortsverlaufes, z.B. linear oder parabolisch) kann man dann jedoch den wahren Ort der Kante ermitteln. Unter Laborbedingungen lassen sich zuverlässig Pixelgenauigkeiten von besser als 1/10 Pixel erreichen und unter Fertigungsbedingungen lässt sich von einer Genauigkeit besser als 1/5 Pixel ausgehen, so dass auch eine Vermessung im Mikrometerbereich mit entsprechenden Komponenten möglich ist (nach [DSAW98]).
syntaktische Mustererkennung	→ <i>Objekte</i> werden zunächst in einer Hierarchie in einfachste Bestandteile wie z.B. Geradenstücke zerlegt und diese anschließend mit Hilfe von Methoden aus der Syntaxanalyse formaler Sprachen zu Bestandteilen höherer Ordnung zusammengefasst.
Teststichprobe	Diejenige Menge an → <i>Objekten</i> aus der → <i>Aufgabendomäne</i> , die zum Testen eines eingelernten → <i>Klassifikators</i> herangezogen werden. Die Teststichprobe sollte unabhängig von der → <i>Lernstichprobe</i> gewählt werden, d.h. → <i>Objekte</i> der Teststichprobe sollten nicht in der → <i>Lernstichprobe</i> enthalten sein.
Topologie	Topologie ist die mathematische Richtung, die sich mit den Eigenschaften von Objekten beschäftigt, die sich unter Deformation, Krümmung und Dehnung nicht ändern. In der Topologie wird so etwas wie Nähe eingeführt, ohne dass zwangsläufig eine → <i>Metrik</i> vorhanden ist.
Torus	Ring mit kreisförmigem Querschnitt.
Trefferquote	Die Trefferquote gibt das Verhältnis von allen richtig eingeordneten Konturen der Teststichprobe und der Gesamtanzahl der Konturen in der Teststichprobe an. Sie ist damit ein Gütemaß für den gesamten Klassifikationsprozess.
TQ	Abkürzung für → <i>Trefferquote</i> .
überwachtes Lernen	Ein Trainer gibt dem lernenden System die zu den einzulernenden → <i>Objekten</i> gehörenden → <i>Klassen</i> vor.
unüberwachtes Lernen	Das lernende System versucht selbständig ähnliche einzulernende → <i>Objekte</i> gemeinsamen → <i>Klassen</i> zuzuordnen. Es wird oft auch von selbstorganisiertem Lernen gesprochen.

wichtiger Konturabschnitt	Ein wichtiger Konturabschnitt einer Kontur unterscheidet sich von genau einem wichtigen Konturabschnitt genau einer anderen Kontur.
VDMA	Verband Deutscher Maschinen- und Anlagenbauer e.V.
zentriertes Moment	Das um den Schwerpunkt bereinigte → <i>Moment</i> .
Zoom(objektiv)	Objektiv mit stufenlos veränderbarer Brennweite.

Anhang C

Dynamische Programmierung

Die Entwicklung der dynamischen Programmierung geht zurück auf David Bellman. In seinem Buch *Dynamic Programming* [Bel57] sind seine Untersuchungen auf diesem Gebiet zusammengefasst. Der von Bellman stammende Name *dynamische Programmierung* ist etwas irreführend. Die Bezeichnung *rekursive Optimierung* trifft den Kern der Sache eigentlich besser.

Bei der dynamischen Optimierung wird versucht, eine beste Lösung aus einer Menge mehrerer möglicher Lösungen zu bestimmen, die das Minimum oder Maximum einer Zielfunktion, die bestimmten Nebenbedingungen genügen soll, detektiert. Diese Optimierungsstrategie kommt dann zum Zuge, wenn sich die beste Lösung nicht mit Hilfe der Analysis berechnen lässt.

Ein Beispiel für ein solches Problem ist das Treibstoffproblem. Hier soll beispielsweise ein Flugzeug mit minimalem Treibstoffverbrauch, ausgehend von einer bestimmten Anfangshöhe h_0 und einer bestimmten Anfangsgeschwindigkeit v_0 , eine bestimmte Endhöhe h_{end} und Endgeschwindigkeit v_{end} erreichen. Bekannt ist der Treibstoffverbrauch, um in einer bestimmten Höhe h_j die Geschwindigkeit von v_i auf v_{i+1} zu erhöhen und bei einer bestimmten Geschwindigkeit v_i die Höhe von h_j auf h_{j+1} zu erhöhen. Die Randbedingung für das Problem soll sein, dass sich nur jeweils eine der beiden Größen steigert, und dass keine der Größen kleiner wird. D.h. das Flugzeug vergrößert entweder seine Geschwindigkeit bei gleichbleibender Höhe oder es steigert seine Höhe bei konstanter Geschwindigkeit. Die Geschwindigkeit und die Höhe liegen quantisiert vor und sind jeweils gleichabständig. In diesem Beispiel gibt es die Höhen $h_0=500$ m, $h_1=1000$ m, $h_2=1500$ m, $h_3=2000$ m, $h_{end}=2500$ m und die Geschwindigkeiten $v_0=100$ km/h, $v_1=200$ km/h, $v_2=300$ km/h, $v_3=400$ km/h, $v_4=500$ km/h, $v_{end}=600$ km/h.

Mit diesen Geschwindigkeits- und Höhenwerten als Indices lässt sich ein gerichteter gewichteter Graph konstruieren (siehe Abbildung C.1). Jedem Geschwindigkeits-Höhen-Paar wird ein Knoten des Graphen zugeordnet. Die Verbindungen zwischen den Knoten sind die mit dem Treibstoff gewichteten Kanten des Gra-

phen.

In horizontaler Richtung bleibt die Geschwindigkeit gleich und die Höhe ändert sich. Z.B. ergibt sich ein Treibstoffverbrauch von 7.3, um bei der Geschwindigkeit v_2 die Höhe von h_1 auf h_2 zu steigern. In vertikaler Richtung ist es genau umgekehrt. Hier bleibt die Höhe gleich und die Geschwindigkeit ändert sich. Z.B. ergibt sich in der Höhe h_3 ein Treibstoffverbrauch von 3.9, um die Geschwindigkeit von v_3 auf v_4 zu erhöhen.

Ziel ist es nun, einen Pfad (eine Liste benachbarter Knoten) durch diesen Graphen zu finden, der vom ersten Knoten $\{v_0, h_0\}$ zum letzten Knoten $\{v_{end}, h_{end}\}$ führt, der den oben genannten Randbedingungen genügt und dessen aufsummierter Treibstoffverbrauch minimal wird.

Die einfachste, aber auch aufwendigste, Art dies zu tun, ist für alle möglichen solchen Pfade die Summe zu berechnen und dann denjenigen mit der minimalsten Summe zu detektieren. Bei m möglichen Geschwindigkeiten v_0, \dots, v_{end} und n möglichen Höhen h_0, \dots, h_{end} darf der Pfad sich insgesamt $(m - 1)$ mal nach unten und $n - 1$ mal nach rechts fortpflanzen. Damit handelt es sich dabei um eine Permutation von insgesamt $(m - 1) + (n - 1)$ Elementen einer Menge, die $(m - 1)$ gleiche Elemente *Nach Unten* und $(n - 1)$ gleiche Elemente *Nach Rechts* beinhaltet. Formal bedeutet das, dass

$$\frac{(m + n - 2)!}{(m - 1)!(n - 1)!} = \binom{m + n - 2}{m - 1} \quad (\text{C.1})$$

erlaubte Pfade existieren. Bei dem Beispiel existieren also schon

$$\binom{6 + 5 - 2}{6 - 1} = 126 \quad (\text{C.2})$$

erlaubte Pfade. Diese Anzahl wäre vielleicht noch handhabbar. Für z.B. $m = 14$ und $n = 14$ existieren allerdings schon 1040060 erlaubte Pfade, die alle bzgl. der minimalen Treibstoffsumme untersucht werden müssten.

Aus diesem Grunde ist es sinnvoll eine weniger aufwendige Methode zu finden, die zum gleichen Ziel führt. Eine dieser Methoden ist die dynamische Programmierung. Das Gesamtproblem wird in einzelne Probleme zerlegt und mit Rücksicht auf das Gesamtproblem rekursiv gelöst.

Dazu wird im 1. Schritt mit dem Zielknoten $\{v_{end}, h_{end}\}$ in der in Abbildung C.1 gestrichelt angedeuteten Ebene 0 begonnen, dem die Treibstoffmenge $T(v_{end}, h_{end}) := 0.0$ zugeordnet wird. Es werden die Treibstoffmengen der Knoten in der Ebene 1 berechnet.

Ebene 1:

$$\begin{aligned}
T(v_4, h_{end}) &:= T(v_{end}, h_{end}) + t(v_4 \rightarrow v_{end}, h_{end}) \\
&:= 0.0 + 3.6 \\
&:= 3.6 \\
T(v_{end}, h_3) &:= T(v_{end}, h_{end}) + t(v_{end}, h_3 \rightarrow h_{end}) \quad (C.3) \\
&:= 0.0 + 7.9 \\
&:= 7.9
\end{aligned}$$

mit T als Treibstoffverbrauch, um aus einer Höhe in eine andere bzw. von einer Geschwindigkeit zur nächsten zu gelangen. Den Knoten der Ebene 1 wird die Richtung R , die auf den Knoten $\{v_{end}, h_{end}\}$ zeigt, zugewiesen:

$$\begin{aligned}
R(v_4, h_{end}) &:= \downarrow \\
R(v_{end}, h_3) &:= \rightarrow . \quad (C.4)
\end{aligned}$$

Im 2. Schritt werden die Treibstoffmengen, die den Knoten von Ebene 2 zugeordnet werden, berechnet und zwar unter der Berücksichtigung der im vorigen Schritt ermittelten Werte.

Ebene 2:

$$\begin{aligned}
T(v_3, h_{end}) &:= T(v_4, h_{end}) + t(v_3 \rightarrow v_4, h_{end}) \\
&:= 3.6 + 1.7 \\
&:= 5.3 \\
T(v_4, h_3) &:= \min \left(T(v_4, h_{end}) + t(v_4, h_3 \rightarrow h_{end}), \right. \\
&\quad \left. T(v_{end}, h_3) + t(v_4 \rightarrow v_{end}, h_3) \right) \\
&:= \min(3.6 + 4.2, 7.9 + 5.8) \quad (C.5) \\
&:= 7.8 \\
T(v_{end}, h_2) &:= T(v_{end}, h_3) + t(v_{end}, h_2 \rightarrow h_3) \\
&:= 7.9 + 9.7 \\
&:= 17.6
\end{aligned}$$

Bei der Berechnung von $T(v_4, h_3)$ wird ein Minimum bestimmt, da es 2 mögliche Wege zum Knoten $\{v_4, h_3\}$ gibt und der minimale Treibstoffverbrauch ermittelt werden soll.

Dem Knoten $\{v_4, h_3\}$ wird diejenige Richtung zugeordnet, die in die Richtung zeigt, die den minimalen Treibverbrauch für diesen Knoten ergibt:

$$R(v_4, h_3) := \rightarrow \quad (\text{C.6})$$

Die beiden anderen Knoten haben wie in der vorigen Ebene 1 nur jeweils einen Vorgängerknoten auf den sie zeigen:

$$\begin{aligned} R(v_{end}, h_2) &:= \rightarrow \\ R(v_3, h_{end}) &:= \downarrow . \end{aligned} \quad (\text{C.7})$$

Die Richtungen sind jeweils in den Knoten in Abbildung C.1 eingezeichnet.

Diese Verfahrensweise wird bis zum Startknoten $\{v_0, h_0\}$ in Ebene 9 über alle Ebenen hinweg durchgeführt und dem Knoten $\{v_0, h_0\}$ ist dann der minimale Gesamtreibstoffverbrauch zugeordnet, der benötigt wird, um von der Anfangsgeschwindigkeit v_0 und der Anfangshöhe h_0 zur angestrebten Endgeschwindigkeit v_{end} und Endhöhe h_{end} zu gelangen.

Werden die den Knoten zugeordneten Richtungen, angefangen beim Anfangsknoten und bis hin zum Endknoten, zurückverfolgt (engl. backtracking), so erhält man den dem minimalen Treibstoffverbrauch zugehörigen Pfad (in Abbildung C.1 fett markiert). Dieses Flugzeug steigert also zunächst seine Geschwindigkeit in der Höhe h_0 von v_0 auf v_3 . Danach fliegt es mit der Geschwindigkeit v_3 aus der Höhe h_0 in die Höhe h_{end} , um dann in dieser Höhe nochmals die Geschwindigkeit von v_3 auf v_{end} zu erhöhen.

Bei dieser Art der Suche müssen bei m Geschwindigkeiten und n Höhen $(mn - 1)$ Knoten untersucht werden. D.h. in unserem Beispiel, dass $(5 \cdot 6 - 1) = 29$ Knoten untersucht werden müssen. Für das Beispiel $m = 14, n = 14$ ergeben sich so nur 195 Knotenuntersuchungen, also weit weniger als die Gesamtanzahl erlaubter Pfade.

Die Suche kann selbstverständlich auch in der Ebene 9 beginnen und in Ebene 0 enden. Das Ergebnis für die minimalen Gesamtreibstoffkosten ist dasselbe. Nur der Pfad verläuft dann durch das Zurückverfolgen vom Endknoten zum Anfangsknoten, was nicht unbedingt immer erwünscht ist.

Das Gesamtproblem wurde in Einzelprobleme zerlegt. Ein Einzelproblem besteht darin zu bestimmen, ob auf die Geschwindigkeit v_{i+1} oder auf die Höhe h_{j+1} erhöht werden muss, um den Gesamtreibstoffverbrauch bzgl. eines Knotens $\{v_i, h_j\}$ minimal zu halten. Die Vorgängerknoten gehen in die Berechnung mit ein, in deren Treibstoffberechnung wiederum deren Vorgängerknoten eingingen usw. . Formal besteht die Lösung eines Einzelproblems in folgender Minimierung:

$$\begin{aligned} T(v_i, h_j) = \min & \left(T(v_{i+1}, h_j) + t(v_i \rightarrow v_{i+1}, h_j), \right. \\ & \left. T(v_i, h_{j+1}) + t(v_i, h_j \rightarrow h_{j+1}) \right) \end{aligned} \quad (\text{C.8})$$

Analog können so Maximum-Probleme gelöst werden. Die Minimumsuche in Gleichung (C.8) wird in dem Fall einfach durch eine Maximumsuche ersetzt.

Manche Probleme lassen sich nicht wie im Treibstoffbeispiel schematisieren. Hier sei nur angedeutet, dass die Theorie der dynamischen Programmierung weit allgemeiner gehalten ist:

Gegeben sei eine Zielfunktion

$$z(x_1, x_2, \dots, x_i, \dots, x_N) \quad (\text{C.9})$$

mit den Variablen x_i , die ausschließlich diskrete Werte größer oder gleich 0 annehmen dürfen. Für die Zielfunktion wird ein Minimum oder Maximum gesucht.

Lässt sich die Funktion C.9 folgendermaßen zerlegen:

$$z(x_1, x_2, \dots, x_i, \dots, x_N) = z(x_1, x_2) + z(x_2, x_3) + \dots + z(x_{N-1}, x_N), \quad (\text{C.10})$$

dann handelt es sich um ein *serielles Optimierungsproblem*. Wenn nicht, wie z.B. in dieser Gleichung:

$$z(x_1, x_2, x_3, x_4) = z(x_1, x_2) + z(x_1, x_4) + z(x_2, x_3, x_4), \quad (\text{C.11})$$

dann handelt es sich um ein *nichtserielles Optimierungsproblem*. Beide Optimierungsprobleme lassen sich mit Hilfe der dynamischen Programmierung lösen.

Anhang D

Clusteranalyse

Die Clusteranalyse (im deutschen Sprachraum oft *automatische Klassifikation* genannt) ist ein Verfahren zur \rightarrow *multivariaten Datenanalyse*. Ihr Ziel ist es, Objekte möglichst optimal in verschiedene Gruppen (engl. cluster) oder auch Klassen einzuteilen. Optimal bedeutet in diesem Zusammenhang, dass die entstehenden Cluster möglichst homogen sein sollten. Die Objekte selbst werden durch eine Anzahl von Eigenschaften zahlenmäßig beschrieben, die die Objekte charakterisieren.

Oft wird eine sogenannte *Rohdatenmatrix* erstellt, in der die Eigenschaften für ein Objekt in jeweils einer Zeile stehen. Dies setzt allerdings voraus, dass die Objekte alle die gleiche Anzahl an Eigenschaften besitzen.

Aus der Rohdatenmatrix lässt sich eine *Ähnlichkeits-* oder auch *Distanzmatrix* S bzw. D mit Hilfe einer *Ähnlichkeits-* oder *Distanzfunktion* erstellen, indem alle Objektvektoren der Rohdatenmatrix paarweise miteinander verglichen werden. Eine Distanzfunktion könnte beispielsweise die Euklidische Distanz zwischen je zwei solcher Objektvektoren sein. Bei N Objekten entsteht eine $N \times N$ -Matrix S bzw. D , die meist, aber nicht zwangsläufig, symmetrisch ist und deren Elemente die Distanzen oder Ähnlichkeiten zwischen den einzelnen Objekten sind. In der i -ten Zeile und j -ten Spalte steht die Distanz $d_{i,j}$ bzw. Ähnlichkeit $s_{i,j}$ zwischen Objekt i und Objekt j :

$$D = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,j} & \cdots & d_{1,N} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,j} & \cdots & d_{2,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{i,1} & d_{i,2} & \cdots & d_{i,j} & \cdots & d_{i,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{N,1} & d_{N,2} & \cdots & d_{N,j} & \cdots & d_{N,N} \end{pmatrix}, \quad (\text{D.1})$$

$$S = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,j} & \cdots & s_{1,N} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,j} & \cdots & s_{2,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{i,1} & s_{i,2} & \cdots & s_{i,j} & \cdots & s_{i,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{N,1} & s_{N,2} & \cdots & s_{N,j} & \cdots & s_{N,N} \end{pmatrix}. \quad (\text{D.2})$$

Die Erstellung einer Distanz- bzw. Ähnlichkeitsmatrix aus einer Rohdatenmatrix ist aber nicht unbedingt notwendig. Das in Abschnitt 4.3.2 vorgestellte Verfahren zur Erzeugung von globalen Distanz- bzw. Ähnlichkeitsmatrizen mit den globalen Distanzen bzw. Ähnlichkeiten von Konturen ist ein Beispiel dafür. Die Anzahl der Konturpunkte, d.h. die Anzahl der Eigenschaften der Konturen, ist unterschiedlich, so dass keine Rohdatenmatrix erzeugt werden kann. Trotzdem wird dort ein Verfahren vorgeschlagen, die Distanz zwischen 2 Konturen zu bestimmen.

Da aus einer Ähnlichkeitsmatrix durch Inversion der einzelnen Elemente leicht eine Distanzmatrix erzeugt werden kann, werden sich die nachfolgenden Erklärungen nur noch auf Distanzmatrizen beziehen.

Eine wichtige Gruppe der Clusteranalyseverfahren ist die der *hierarchischen* Verfahren. Bei ihnen werden Folgen von Gruppierungen erzeugt, so dass von einer Hierarchieebene zur nächsten jeweils eine Gruppierung mehr oder weniger erzeugt wird. Zwei Cluster in aufeinanderfolgenden Hierarchieebenen sind entweder elementfremd oder vollständig ineinander enthalten.

Man unterscheidet sogenannte *divisive* und *agglomerative* Verfahren. Bei den divisiven Verfahren wird die Gruppierungsfolge durch eine stufenweise Verfeinerung einer größten Einteilung erreicht. Bei den agglomerativen Verfahren ist dies genau andersherum. Hier werden die einzelnen Objekte sukzessiv zu Clustern und diese Cluster nach und nach zu umfangreicheren Clustern zusammengefasst.

Das schrittweise Zusammenfassen oder Aufteilen von Clustern lässt sich mit sogenannten *Dendogrammen* darstellen (siehe Abbildung D.1). Zum einen erkennt man in der Abbildung das schrittweise Zusammenfassen (agglomerativ) der Objekte o_i von unten nach oben bzw. das Aufteilen (divisiv) der Cluster von oben nach unten auf verschiedenen Distanzebenen E_i , denen wie später beschrieben verschiedene Werte zugeordnet werden. Für die verschiedenen Ebenen in der Abbildung können folgende Cluster angegeben werden:

$$\begin{aligned} E_1 & : \{o_1\}, \{o_2\}, \{o_3\}, \{o_4\}, \{o_5\}, \{o_6\} \\ E_2 & : \{o_1\}, \{o_2\}, \{o_3\}, \{o_4\}, \{o_5, o_6\} \\ E_3 & : \{o_1, o_2\}, \{o_3\}, \{o_4\}, \{o_5, o_6\} \\ E_4 & : \{o_1, o_2\}, \{o_3\}, \{o_4, o_5, o_6\} \\ E_5 & : \{o_1, o_2, o_3\}, \{o_4, o_5, o_6\} \end{aligned}$$

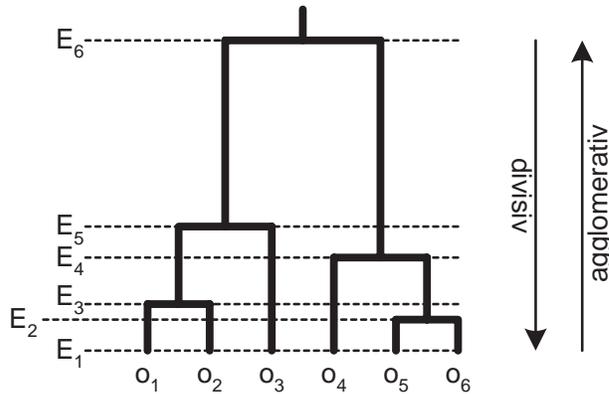


Abbildung D.1: Dendrogramm zur Darstellung des Zusammenfassens (agglomerativ) oder Aufteilens (divisiv) von Clustern auf verschiedenen Distanzebenen E_i - die feinste Partition mit den Objekten $o_1 \dots o_6$ auf Ebene E_1 , die größte Partition mit nur einem Cluster auf Ebene E_6 .

$$E_6 : \{o_1, o_2, o_3, o_4, o_5, o_6\}.$$

Im Nachfolgenden werden die agglomerativen Verfahren näher beschrieben. Ein agglomeratives Verfahren beinhaltet folgende grundlegenden Schritte:

1. Beginne mit der feinsten Partition

$$G = \{g_1, g_2, \dots, g_i, \dots, g_N\} \quad (\text{D.3})$$

mit

$$g_i = \{o_i\}. \quad (\text{D.4})$$

2. Suche diejenigen Gruppen g_p und g_q , die von allen Gruppen die minimalste Distanz besitzen:

$$d_{p,q} = \min_{i \neq j} d_{i,j}. \quad (\text{D.5})$$

3. Fusioniere die Gruppen g_p und g_q zu einer neuen Gruppen g_p^{neu} . Die Gruppenanzahl erniedrigt sich dadurch um 1:

$$g_p^{neu} = g_p \cup g_q. \quad (\text{D.6})$$

4.
 - Ändere die Elemente der p -ten Zeile und Spalte, indem die Distanzen zwischen der neuen Gruppe g_p^{neu} und allen anderen Gruppen neu berechnet werden.

- Streiche die q -te Zeile und Spalte aus der Distanzmatrix.

5. Fahre solange ($N - 1$ mal) mit Schritt 2 fort, bis alle Gruppen zu einer einzigen Gruppe zusammengefasst sind.

Es gibt verschiedene agglomerative Verfahren, die sich ausschließlich in Schritt 4, d.h. der Neuberechnung der Distanz zwischen der neu entstandenen Gruppe g_p^{neu} und allen übrigen Gruppen, unterscheiden.

Im einzelnen werden im Folgenden jene agglomerativen Verfahren vorgestellt, die nicht voraussetzen, dass die Distanzen basierend auf der Euklidischen Distanz entstanden sind, da dies bei der Distanzbestimmung in Kapitel 4 nicht der Fall ist:

1. Single-Linkage (Nächster Nachbar)

Die neue Distanz $d_{p,i}^{neu}$ berechnet sich folgendermaßen:

$$d_{p,i}^{neu} = d_{i,p}^{neu} = \min(d_{p,i}, d_{q,i}) \quad (\text{D.7})$$

$d_{p,i}$ und $d_{q,i}$ sind die alten Distanzen der zusammengefassten Gruppen p und q zu einer nicht zusammengefassten Gruppe i . In diesem Fall wird jeweils die minimale Distanz genommen. Das bedeutet, dass die Distanz der jeweils nächsten Nachbarn (Objekte) in der zusammengefassten Gruppe und den nicht zusammengefassten Gruppen i genommen wird.

Dies führt i. allg. zu einer typischen Kettenbildung der Cluster. Deshalb sollte dieses Verfahren nur herangezogen werden, wenn kettenförmige Cluster erwartet werden.

2. Complete-Linkage (Entferntester Nachbar)

$$d_{p,i}^{neu} = d_{i,p}^{neu} = \max(d_{p,i}, d_{q,i}) \quad (\text{D.8})$$

Dieses Verfahren ist der Antipode zum Single-Linkage-Verfahren. Hier wird jeweils die maximalste Distanz eines Objektes aus der zusammengefassten Gruppe zu einem Objekt aus den nicht zusammengefassten Gruppen genommen. In anderen Worten bedeutet das, dass die Distanzen zwischen den jeweils entferntesten Objekten der jeweiligen Gruppen benutzt werden.

Dieses Verfahren führt zur Konstruktion von vielen kleinen Gruppen.

3. Average-Linkage (Durchschnitt)

$$d_{p,i}^{neu} = d_{i,p}^{neu} = \frac{d_{p,i} + d_{q,i}}{2} \quad (\text{D.9})$$

Dieses Verfahren gilt als Kompromiss zwischen dem Single-Linkage- und dem Complete-Linkage-Verfahren. Hier wird jeweils die mittlere Distanz

zwischen den Objekten der zusammengefassten Gruppe zu den Distanzen der nicht zusammengefassten Gruppen als neuer Distanzwert herangezogen.

4. Weighted Average-Linkage (Gewichteter Durchschnitt)

$$d_{p,i}^{neu} = d_{i,p}^{neu} = \frac{n_p d_{p,i} + n_q d_{q,i}}{n_p + n_q} \quad (\text{D.10})$$

n_p und n_q sind die Anzahl der Objekte in den Gruppen p und q vor der Zusammenfassung. Dies ist die verallgemeinerte Version des Average-Linkage-Verfahrens. Hier wird jeweils die mit n_p und n_q gewichtete mittlere Distanz zwischen den Objekten der zusammengefassten Gruppe und den Distanzen der nicht zusammengefassten Gruppen als neuer Distanzwert zu den nicht zusammengefassten Gruppen ermittelt. Dieses Verfahren erweist sich als vorteilhaft, wenn Gruppen mit einer stark unterschiedlichen Anzahl an Objekten erwartet werden.

5. Verfahren nach Ward

$$d_{p,i}^{neu} = d_{i,p}^{neu} = \frac{(n_p + n_i)d_{p,i} + (n_q + n_i)d_{q,i} + n_i d_{p,q}}{n_p + n_q + n_i} \quad (\text{D.11})$$

mit n_i als Anzahl der Objekte in Gruppe i . Gleichung (D.11) ergibt sich aus dem Ansatz möglichst homogene Gruppen dadurch zu konstruieren, dass jeweils diejenigen Gruppen fusioniert werden, die durch das Zusammenfassen den geringsten Zuwachs an Varianz, bezogen auf die Rohdaten der Objekte, haben. Die Distanzmatrix zu Beginn enthält hier die Elemente

$$d_{i,j} = \frac{(\bar{o}_i - \bar{o}_j)^2}{2} \quad (\text{D.12})$$

mit \bar{o}_i bzw. \bar{o}_j als Mittelwert der Elemente der Objektvektoren der Objekte i bzw. j .

Dieses Verfahren basiert auf quadrierten Euklidischen Distanzen. Nach [SL77] lassen sich mit diesem Verfahren aber auch für andere Distanzfunktionen zufriedenstellende Ergebnisse erzielen.

An einem Beispiel mit 6 Objekten mit jeweils 2 Variablen sollen jetzt die Single-Linkage-Methode und die Complete-Linkage-Methode anschaulich

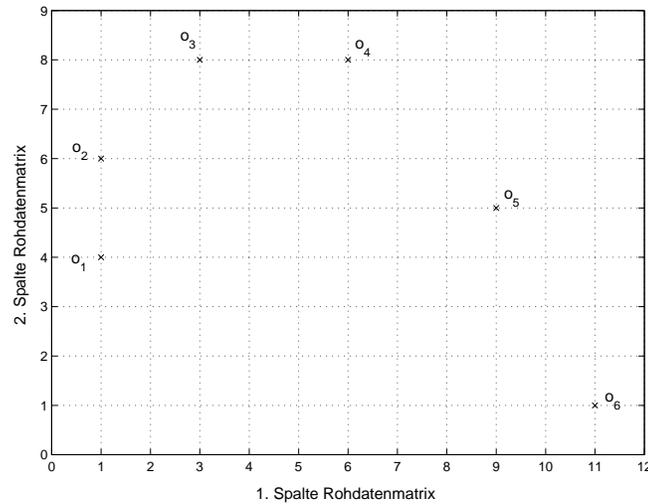


Abbildung D.2: Objekte der Rohdatenmatrix in Gleichung (D.13) in Karthesisches Koordinatensystem eingetragen.

gemacht werden. Die bei beiden Verfahren herangezogene Rohdatenmatrix

$$\begin{pmatrix} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \\ o_6 \end{pmatrix} = \begin{pmatrix} 1 & 4 \\ 1 & 6 \\ 3 & 8 \\ 6 & 8 \\ 9 & 5 \\ 11 & 1 \end{pmatrix} \quad (\text{D.13})$$

ist in Abbildung D.2 veranschaulicht.

Als Distanzfunktion zwischen den Objekten o_i und o_j wird die quadrierte Euklidische Distanz

$$d_{i,j} = (o_{i,1} - o_{j,1})^2 + (o_{i,2} - o_{j,2})^2 \quad (\text{D.14})$$

herangezogen. Es ergibt sich die anfängliche Distanzmatrix

$$D = \begin{pmatrix} 0 & 4 & 20 & 41 & 65 & 109 \\ 4 & 0 & 8 & 29 & 65 & 125 \\ 20 & 8 & 0 & 9 & 45 & 113 \\ 41 & 29 & 9 & 0 & 18 & 74 \\ 65 & 65 & 45 & 18 & 0 & 20 \\ 109 & 125 & 113 & 74 & 20 & 0 \end{pmatrix} \quad (\text{D.15})$$

Single-Linkage:

Im Folgenden werden ausführlich die einzelnen Schritte der Clusterung für jede

einzelne Distanzebene E_i formal dargestellt. Die kleinen Zahlen in den auf jeder Distanzebene neu entstehenden Distanzmatrizen D^{neu} markieren die jeweils gestrichene Zeile und Spalte:

Schritt 2 - E_2 :

$$E_2 := d_{1,2} = \min_{i \neq j} d_{i,j} = 4 \quad (\text{D.16})$$

Schritt 3 - E_2 :

$$\{o_1, o_2\} := \{o_1\}, \{o_2\} \quad (\text{D.17})$$

Schritt 4 - E_2 :

$$\begin{aligned} d_{1,1}^{neu} &:= 0 \\ d_{1,2}^{neu} = d_{2,1}^{neu} &:= \min(d_{1,2}, d_{2,2}) = \min(4, 0) = 0 \\ d_{1,3}^{neu} = d_{3,1}^{neu} &:= \min(d_{1,3}, d_{2,3}) = \min(20, 8) = 8 \\ d_{1,4}^{neu} = d_{4,1}^{neu} &:= \min(d_{1,4}, d_{2,4}) = \min(41, 29) = 29 \\ d_{1,5}^{neu} = d_{5,1}^{neu} &:= \min(d_{1,5}, d_{2,5}) = \min(65, 65) = 65 \\ d_{1,6}^{neu} = d_{6,1}^{neu} &:= \min(d_{1,6}, d_{2,6}) = \min(109, 125) = 109 \end{aligned} \quad (\text{D.18})$$

$$\Rightarrow D^{neu} := \begin{pmatrix} 0 & 0 & 8 & 29 & 65 & 109 \\ 0 & 0 & 8 & 29 & 65 & 125 \\ 8 & 8 & 0 & 9 & 45 & 113 \\ 29 & 29 & 9 & 0 & 18 & 74 \\ 65 & 65 & 45 & 18 & 0 & 20 \\ 109 & 125 & 113 & 74 & 20 & 0 \end{pmatrix} \quad (\text{D.19})$$

Schritt 2 - E_3 :

$$E_3 := d_{1,2} = \min_{i \neq j} d_{i,j} = 8 \quad (\text{D.20})$$

Schritt 3 - E_3 :

$$\{o_1, o_2, o_3\} := \{o_1, o_2\}, \{o_3\} \quad (\text{D.21})$$

Schritt 4 - E_3 :

$$\begin{aligned} d_{1,1}^{neu} &:= 0 \\ d_{1,2}^{neu} = d_{2,1}^{neu} &:= \min(d_{1,2}, d_{2,2}) = \min(8, 0) = 0 \\ d_{1,3}^{neu} = d_{3,1}^{neu} &:= \min(d_{1,3}, d_{2,3}) = \min(29, 9) = 9 \\ d_{1,4}^{neu} = d_{4,1}^{neu} &:= \min(d_{1,4}, d_{2,4}) = \min(65, 45) = 45 \\ d_{1,5}^{neu} = d_{5,1}^{neu} &:= \min(d_{1,5}, d_{2,5}) = \min(109, 113) = 109 \end{aligned} \quad (\text{D.22})$$

$$\Rightarrow D^{neu} := \begin{pmatrix} 0 & 0 & 9 & 45 & 109 \\ 0 & 0 & 9 & 45 & 113 \\ 9 & 9 & 0 & 18 & 74 \\ 45 & 45 & 18 & 0 & 20 \\ 109 & 113 & 74 & 20 & 0 \end{pmatrix} \quad (\text{D.23})$$

Schritt 2 - E_4 :

$$E_4 := d_{1,2} = \min_{i \neq j} d_{i,j} = 9 \quad (\text{D.24})$$

Schritt 3 - E_4 :

$$\{o_1, o_2, o_3, o_4\} := \{o_1, o_2, o_3\}, \{o_4\} \quad (\text{D.25})$$

Schritt 4 - E_4 :

$$\begin{aligned} d_{1,1}^{neu} &:= 0 \\ d_{1,2}^{neu} = d_{2,1}^{neu} &:= \min(d_{1,2}, d_{2,2}) = \min(9, 0) = 0 \\ d_{1,3}^{neu} = d_{3,1}^{neu} &:= \min(d_{1,3}, d_{2,3}) = \min(45, 18) = 18 \\ d_{1,4}^{neu} = d_{4,1}^{neu} &:= \min(d_{1,4}, d_{2,4}) = \min(109, 74) = 74 \end{aligned} \quad (\text{D.26})$$

$$\Rightarrow D^{neu} := \begin{pmatrix} 0 & 0 & 18 & 74 \\ 0 & 0 & 18 & 74 \\ 18 & 18 & 0 & 20 \\ 74 & 74 & 20 & 0 \end{pmatrix} \quad (\text{D.27})$$

Schritt 2 - E_5 :

$$E_5 := d_{1,2} = \min_{i \neq j} d_{i,j} = 18 \quad (\text{D.28})$$

Schritt 3 - E_5 :

$$\{o_1, o_2, o_3, o_4, o_5\} := \{o_1, o_2, o_3, o_4\}, \{o_5\} \quad (\text{D.29})$$

Schritt 4 - E_5 :

$$\begin{aligned} d_{1,1}^{neu} &:= 0 \\ d_{1,2}^{neu} = d_{2,1}^{neu} &:= \min(d_{1,2}, d_{2,2}) = \min(18, 0) = 0 \\ d_{1,3}^{neu} = d_{3,1}^{neu} &:= \min(d_{1,3}, d_{2,3}) = \min(74, 20) = 20 \end{aligned} \quad (\text{D.30})$$

$$(\text{D.31})$$

$$\Rightarrow D^{neu} := \begin{pmatrix} 0 & 0 & 20 \\ 0 & 0 & 20 \\ 20 & 20 & 0 \end{pmatrix} \quad (\text{D.32})$$

Schritt 2 - E_6 :

$$E_6 := d_{1,2} = \min_{i \neq j} d_{i,j} = 20 \quad (\text{D.33})$$

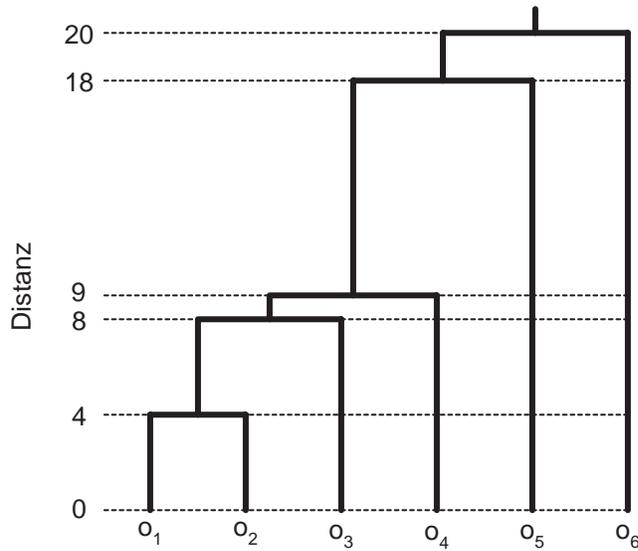


Abbildung D.3: Die Abbildung zeigt das Dendrogramm für das Single-Linkage-Verfahren basierend auf der Rohdatenmatrix D.13 (siehe auch Abbildung D.2) und der quadrierten Euklidischen Distanz. Man erkennt deutlich die für das Single-Linkage-Verfahren typische Kettenbildung.

Schritt 3 - E_6 :

$$\{o_1, o_2, o_3, o_4, o_5, o_6\} := \{o_1, o_2, o_3, o_4, o_5\}, \{o_6\} \quad (\text{D.34})$$

Schritt 4 - E_6 :

$$d_{1,2}^{neu} = d_{2,1}^{neu} := \min(d_{1,2}, d_{2,2}) = \min(20, 0) = 0 \quad (\text{D.35})$$

$$\Rightarrow D^{neu} := \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (\text{D.36})$$

Zusammengefasst ergibt sich das in Abbildung D.3 gezeigte Dendrogramm. Man erkennt deutlich die, für das Single-Linkage-Verfahren, typische Kettenbildung. D.h. es entsteht nach und nach ein größer werdender Cluster, dem Einzelobjekte zugeordnet werden.

Complete Linkage:

Für das Complete-Linkage-Verfahren sei an dieser Stelle nur das Dendrogramm

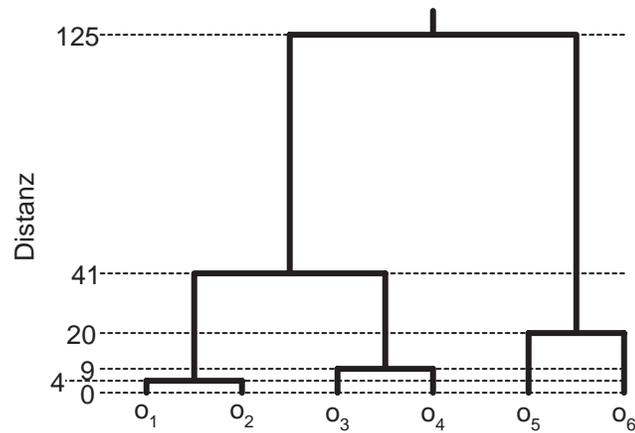


Abbildung D.4: Die Abbildung zeigt das Dendrogramm für das Complete-Linkage-Verfahren basierend auf der Rohdatenmatrix D.13 (siehe auch Abbildung D.2) und der quadrierten Euklidischen Distanz. Man erkennt deutlich die Tendenz des Complete-Linkage-Verfahrens kleine Gruppen zu bilden.

in Abbildung D.4 dargestellt. Man erkennt deutlich den Unterschied zum Single-Linkage-Verfahren. Deutlich erkennbar ist die Tendenz des Complete-Linkage-Verfahrens, kleine Gruppen zu bilden. So entstehen zunächst 3 kleine Cluster mit jeweils 2 Objekten, die dann später zusammengefasst werden.

Anhang E

Algorithmen

In diesem Anhang sind die Algorithmen aufgeführt, auf die im Kapitel 4 verwiesen wird.

$\Delta_{(i,j,k),(r,s,t)} = \text{Algorithmus Ia}(A_{i,j,k}^{l_{i,j,k}}, A_{r,s,t}^{l_{r,s,t}}):$

Kommentar: Eingang: $A_{i,j,k}^{l_{i,j,k}}, A_{r,s,t}^{l_{r,s,t}}$ sind 2 Konturabschnitte mit $l_{i,j,k} = l_{r,s,t}$.

Ausgang: $\Delta_{(i,j,k),(r,s,t)}$ ist Distanz zwischen $A_{i,j,k}^{l_{i,j,k}}, A_{r,s,t}^{l_{r,s,t}}$.

$l := l_{i,j,k} \quad (= l_{r,s,t})$

$\alpha_{i,j,k} := \text{atan2}(y_{i,j,k+l} - y_{i,j,k-l}, x_{i,j,k+l} - x_{i,j,k-l})$

$\alpha_{r,s,t} := \text{atan2}(y_{r,s,t+l} - y_{r,s,t-l}, x_{r,s,t+l} - x_{r,s,t-l})$

$a := -l$

$sum_{i,j,k} := 0$

$sum_{r,s,t} := 0$

while $a \leq l$

$\tilde{P}_{i,j,k+a} := P_{i,j,k+a} - P_{i,j,k-l}$

$\tilde{P}_{r,s,t+a} := P_{r,s,t+a} - P_{r,s,t-l}$

$\hat{P}_{i,j,k+a} := \begin{pmatrix} \cos(\alpha_{i,j,k}) & \sin(\alpha_{i,j,k}) \\ -\sin(\alpha_{i,j,k}) & \cos(\alpha_{i,j,k}) \end{pmatrix} \begin{pmatrix} \tilde{x}_{i,j,k+a} \\ \tilde{y}_{i,j,k+a} \end{pmatrix}$

$\hat{P}_{r,s,t+a} := \begin{pmatrix} \cos(\alpha_{r,s,t}) & \sin(\alpha_{r,s,t}) \\ -\sin(\alpha_{r,s,t}) & \cos(\alpha_{r,s,t}) \end{pmatrix} \begin{pmatrix} \tilde{x}_{r,s,t+a} \\ \tilde{y}_{r,s,t+a} \end{pmatrix}$

$sum_{i,j,k} := sum_{i,j,k} + \hat{x}_{i,j,k+a}$

$sum_{r,s,t} := sum_{r,s,t} + \hat{x}_{r,s,t+a}$

$a := a + 1$

end

$\xi := \frac{sum_{i,j,k} - sum_{r,s,t}}{2l+1}$

$a := -l$

$\Delta_{(i,j,k),(r,s,t)} := 0$

while $a \leq l$

$\Delta_{(i,j,k),(r,s,t)} := \Delta_{(i,j,k),(r,s,t)} + (\hat{x}_{i,j,k+a} - \hat{x}_{r,s,t+a} - \xi)^2 + (\hat{y}_{i,j,k+a} - \hat{y}_{r,s,t+a})^2$

end

Abbildung E.1: Algorithmus Ia: Berechnung der Distanz $\Delta_{(i,j,k),(r,s,t)} = \Delta(A_{i,j,k}^{l_{i,j,k}}, A_{r,s,t}^{l_{r,s,t}})$ basierend auf der absoluten Orientierung. $\text{atan2}(\Delta y, \Delta x)$ berechnet den inversen Tangens für alle 4 Quadranten.

$\Delta_{(i,j,k),(r,s,t)}$ = Algorithmus Ib ($A_{i,j,k}^{l_{i,j,k}}$, $A_{r,s,t}^{l_{r,s,t}}$):

Kommentar: Eingang: $A_{i,j,k}^{l_{i,j,k}}$, $A_{r,s,t}^{l_{r,s,t}}$ sind 2 Konturabschnitte mit $l_{i,j,k} = l_{r,s,t}$.

Ausgang: $\Delta_{(i,j,k),(r,s,t)}$ ist Distanz zwischen $A_{i,j,k}^{l_{i,j,k}}$, $A_{r,s,t}^{l_{r,s,t}}$.

$l := l_{i,j,k}$ ($= l_{r,s,t}$)

$z := 0$

$P_{i,j,k}^- := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $P_{i,j,k}^+ := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $P_{r,s,t}^- := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $P_{r,s,t}^+ := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

while $z \leq l$

$P_{i,j,k}^- := P_{i,j,k}^- + P_{i,j,k-z}$, $P_{i,j,k}^+ := P_{i,j,k}^+ + P_{i,j,k+z}$

$P_{r,s,t}^- := P_{r,s,t}^- + P_{r,s,t-z}$, $P_{r,s,t}^+ := P_{r,s,t}^+ + P_{r,s,t+z}$

$z := z + 1$

end

$P_{i,j,k}^- := \frac{P_{i,j,k}^-}{l+1}$, $P_{i,j,k}^+ := \frac{P_{i,j,k}^+}{l+1}$

$P_{r,s,t}^- := \frac{P_{r,s,t}^-}{l+1}$, $P_{r,s,t}^+ := \frac{P_{r,s,t}^+}{l+1}$

$\alpha_{i,j,k} := \text{atan2}(y_{i,j,k}^+ - y_{i,j,k}^-, x_{i,j,k}^+ - x_{i,j,k}^-)$

$\alpha_{r,s,t} := \text{atan2}(y_{r,s,t}^+ - y_{r,s,t}^-, x_{r,s,t}^+ - x_{r,s,t}^-)$

$a := -l$

$sum_{i,j,k} := 0$, $sum_{r,s,t} := 0$

while $a \leq l$

$\tilde{P}_{i,j,k+a} := P_{i,j,k+a} - P_{i,j,k}^-$

$\tilde{P}_{r,s,t+a} := P_{r,s,t+a} - P_{r,s,t}^-$

$\hat{P}_{i,j,k+a} := \begin{pmatrix} \cos(\alpha_{i,j,k}) & \sin(\alpha_{i,j,k}) \\ -\sin(\alpha_{i,j,k}) & \cos(\alpha_{i,j,k}) \end{pmatrix} \begin{pmatrix} \tilde{x}_{i,j,k+a} \\ \tilde{y}_{i,j,k+a} \end{pmatrix}$

$\hat{P}_{r,s,t+a} := \begin{pmatrix} \cos(\alpha_{r,s,t}) & \sin(\alpha_{r,s,t}) \\ -\sin(\alpha_{r,s,t}) & \cos(\alpha_{r,s,t}) \end{pmatrix} \begin{pmatrix} \tilde{x}_{r,s,t+a} \\ \tilde{y}_{r,s,t+a} \end{pmatrix}$

$sum_{i,j,k} := sum_{i,j,k} + \hat{x}_{i,j,k+a}$

$sum_{r,s,t} := sum_{r,s,t} + \hat{x}_{r,s,t+a}$

$a := a + 1$

end

$\xi := \frac{sum_{i,j,k} - sum_{r,s,t}}{2l+1}$

$a := -l$

$\Delta_{(i,j,k),(r,s,t)} := 0$

while $a \leq l$

$\Delta_{(i,j,k),(r,s,t)} := \Delta_{(i,j,k),(r,s,t)} + (\hat{x}_{i,j,k+a} - \hat{x}_{r,s,t+a} - \xi)^2 + (\hat{y}_{i,j,k+a} - \hat{y}_{r,s,t+a})^2$

$a := a + 1$

end

Abbildung E.2: Algorithmus Ib: Berechnung der Distanz $\Delta_{(i,j,k),(r,s,t)} = \Delta(A_{i,j,k}^{l_{i,j,k}}, A_{r,s,t}^{l_{r,s,t}})$ basierend auf der mittleren Orientierung. $\text{atan2}(\Delta y, \Delta x)$ berechnet den inversen Tangens für alle 4 Quadranten.

$\Lambda^{(i,j)(r,s)}$ = Algorithmus IIa ($K_{i,j}, K_{r,s}, l_{const}$):

Kommentar: Eingang: $K_{i,j}, K_{r,s}$ sind 2 Konturpunktlisten.

Eingang: l_{const} ist konstante Länge aller Konturabschnitte

Ausgang: $\Lambda^{(i,j)(r,s)}$ ist die lokale Distanzmatrix.

$k := 1$

while $k \leq n_{i,j}$

$t := 1$

while $t \leq n_{r,s}$

$A_{i,j,k}^{l_{const}} := \{P_{i,j,k-l_{const}}, \dots, P_{i,j,k+l_{const}}\}$

$A_{r,s,t}^{l_{const}} := \{P_{r,s,t-l_{const}}, \dots, P_{r,s,t+l_{const}}\}$

$\Lambda_{k,t}^{(i,j)(r,s)} := \Delta(A_{i,j,k}^{l_{const}}, A_{r,s,t}^{l_{const}})$ mit Algorithmus Ia oder Ib

$t := t + 1$

end

$k := k + 1$

end

Abbildung E.3: Algorithmus IIa zur Berechnung der lokalen Distanzmatrix

$\Lambda^{(i,j)(r,s)}$ für konstante $l_{i,j,k} = l_{r,s,t} = l_{const}$.

$\tilde{\Lambda}^{(i,j),(r,s)}$ = Algorithmus IIb($K_{i,j}, K_{r,s}, E$):

Kommentar: Eingang: $K_{i,j}, K_{r,s}$ sind 2 Konturpunktlisten.
 Eingang: E ist maximal erlaubte Distanz zwischen
 2 Konturabschnitten.
 Ausgang: $\tilde{\Lambda}^{(i,j),(r,s)}$ ist die lokale Ähnlichkeitsmatrix.

$k := 1$

while $k \leq n_{i,j}$

$t := 1$

while $t \leq n_{r,s}$

$l := 0$

do

$l := l + 1$

$A_{i,j,k}^l = \{P_{i,j,k-l}, \dots, P_{i,j,k+l}\}$

$A_{r,s,t}^l = \{P_{r,s,t-l}, \dots, P_{r,s,t+l}\}$

$\Delta_{(i,j,k),(r,s,t)} := \Delta(A_{i,j,k}^l, A_{r,s,t}^l)$ mit Alg. Ia oder Ib

while $\Delta_{(i,j,k),(r,s,t)} \leq E$

$\tilde{\Lambda}_{k,t}^{(i,j),(r,s)} := l - 1$

$t := t + 1$

end

$k := k + 1$

end

Abbildung E.4: Algorithmus IIb zur Berechnung der lokalen Ähnlichkeitsmatrix $\tilde{\Lambda}^{(i,j),(r,s)}$ für variable $l_{i,j,k} = l_{r,s,t}$.

$(Dir, g) = \text{Algorithmus IIIa}(M^{t+})$

Kommentar: Eingang: M^{t+} ist $(n_{i,j} + 1) \times (n_{r,s} + 1)$ -Matrix in der der Pfad von $M_{1,1}$ nach $M_{n_{i,j}, n_{r,s}}$ berechnet werden soll.

Ausgang: Dir ist eine $(n_{i,j} + 1) \times (n_{r,s} + 1)$ -Matrix mit den Richtungen auf einen bestimmten Vorgängerknoten.

Ausgang: g ist minimale Summe der Ersetzungskosten für M .

$sum_{(i,j,n_{i,j}+1),(r,s,n_{r,s}+1)} := 0$

$Dir_{n_{i,j}+1, n_{r,s}+1}^{(i,j),(r,s)} := end$

$k := n_{i,j} + 1$

$t := n_{r,s}$

while $k \geq 1$

while $t \geq 1$

if $(k \neq (n_{i,j} + 1)) \wedge (t \neq (n_{r,s} + 1))$

$sum_{(i,j,k),(r,s,t)} := \min(sum_{(i,j,k+1),(r,s,t)}, sum_{(i,j,k),(r,s,t+1)},$
 $sum_{(i,j,k+1),(r,s,t+1)} + M_{k,t})$

if $sum_{(i,j,k+1),(r,s,t+1)} \leq \min(sum_{(i,j,k+1),(r,s,t)}, sum_{(i,j,k),(r,s,t+1)})$

$Dir_{k,t}^{(i,j),(r,s)} := SO$

elseif $sum_{(i,j,k+1),(r,s,t)} < sum_{(i,j,k),(r,s,t+1)}$

$Dir_{k,t}^{(i,j),(r,s)} := S$

else

$Dir_{k,t}^{(i,j),(r,s)} := O$

end

elseif $k = n_{i,j} + 1$

$sum_{(i,j,k),(r,s,t)} := sum_{(i,j,k),(r,s,t+1)} + M_{k,t}$

$Dir_{k,t}^{(i,j),(r,s)} := O$

else

$sum_{(i,j,k),(r,s,t)} := sum_{(i,j,k+1),(r,s,t)} + M_{k,t}$

$Dir_{k,t}^{(i,j),(r,s)} := S$

end

$t := t - 1$

end

$t := n_{r,s} + 1$

$k := k - 1$

end

$g := sum_{(i,j,1),(r,s,1)}$

Abbildung E.5: Algorithmus IIIa zur Berechnung eines bestimmten kostengünstigsten Pfades, der vom Element $M_{1,1}^{t+}$ zum Element $M_{n_{i,j}, n_{r,s}}^{t+}$ läuft, basierend auf $\Lambda^{(i,j),(r,s)}$.

$(Dir, g) = \text{Algorithmus IIIb}(M^{t+})$

Kommentar: Eingang: M^{t+} ist $(n_{i,j} + 1) \times (n_{r,s} + 1)$ -Matrix, in der der Pfad von $M_{1,1}^{t+}$ nach $M_{n_{i,j}, n_{r,s}}^{t+}$ berechnet werden soll.
 Ausgang: Dir ist eine $(n_{i,j} + 1) \times (n_{r,s} + 1)$ -Matrix mit den Richtungen auf einen bestimmten Vorgängerknoten.
 Ausgang: g ist maximale Summe der Ersetzungskosten für M .

$sum_{(i,j,n_{i,j}+1),(r,s,n_{r,s}+1)} := 0$

$Dir_{n_{i,j}+1, n_{r,s}+1}^{(i,j),(r,s)} := end$

$k := n_{i,j} + 1$

$t := n_{r,s}$

while $k \geq 1$

while $t \geq 1$

if $(k \neq (n_{i,j} + 1)) \wedge (t \neq (n_{r,s} + 1))$

$sum_{(i,j,k),(r,s,t)} := \max(sum_{(i,j,k+1),(r,s,t)}, sum_{(i,j,k),(r,s,t+1)},$
 $sum_{(i,j,k+1),(r,s,t+1)} + M_{k,t})$

if $sum_{(i,j,k+1),(r,s,t+1)} \geq \max(sum_{(i,j,k+1),(r,s,t)}, sum_{(i,j,k),(r,s,t+1)})$

$Dir_{k,t}^{(i,j),(r,s)} := SO$

elseif $sum_{(i,j,k+1),(r,s,t)} > sum_{(i,j,k),(r,s,t+1)}$

$Dir_{k,t}^{(i,j),(r,s)} := S$

else

$Dir_{k,t}^{(i,j),(r,s)} := O$

end

elseif $k = n_{i,j} + 1$

$sum_{(i,j,k),(r,s,t)} := sum_{(i,j,k),(r,s,t+1)} + M_{k,t}$

$Dir_{k,t}^{(i,j),(r,s)} := O$

else

$sum_{(i,j,k),(r,s,t)} := sum_{(i,j,k+1),(r,s,t)} + M_{k,t}$

$Dir_{k,t}^{(i,j),(r,s)} := S$

end

$t := t - 1$

end

$t := n_{r,s} + 1$

$k := k - 1$

end

$g := sum_{(i,j,1),(r,s,1)}$

Abbildung E.6: Algorithmus IIIb zur Berechnung eines bestimmten kostengünstigsten Pfades, der vom Element $M_{1,1}^{t+}$ zum Element $M_{n_{i,j}, n_{r,s}}^{t+}$ läuft basierend auf $\tilde{\Lambda}^{(i,j),(r,s)}$.

$(g^{(i,j),(r,s)}, Dir^*, \psi^{min}) = \text{AlgorithmusIVa}(\Lambda^{(i,j),(r,s)})$

Kommentar: Eingang: $\Lambda^{(i,j),(r,s)}$ ist $n_{i,j} \times n_{r,s}$ -Matrix, in der der Pfad mit minimalen Kosten berechnet werden soll.

Ausgang: $g^{(i,j),(r,s)}$ ist die globale Distanz zwischen den Konturen $K_{i,j}$ und $K_{r,s}$.

Ausgang: Dir^* ist diejenige Richtungsmatrix, die zu dem Pfad mit minimalsten Kosten gehört.

Ausgang: ψ^{min} ist der Index t desjenigen Konturpunktes in $K_{r,s}$, der mit dem 1. Konturpunkt aus $K_{i,j}$ bzgl. minimaler Kosten korrespondiert.

$t := 1$

$g^{(i,j),(r,s)} := \infty$

while $t \leq n_{r,s}$

$M^t := \begin{pmatrix} \Lambda_{1..n_{i,j}, t..n_{r,s}}^{(i,j),(r,s)} & \Lambda_{1..n_{i,j}, 1..t-1}^{(i,j),(r,s)} \end{pmatrix}$

$M^{t+} := \begin{pmatrix} M^t & M_{1..n_{i,j}, 1}^t \\ M_{1, 1..n_{r,s}}^t & M_{1, 1}^t \end{pmatrix}$

$(Dir, g^*) := \text{AlgorithmusIIIa}(M^{t+})$

if $g^* \leq g^{(i,j),(r,s)}$

$g^{(i,j),(r,s)} := g^*$

$Dir^* := Dir$

$\psi^{min} := t$

end

$t := t + 1$

end

Abbildung E.7: Algorithmus IVa zur Berechnung des günstigsten Pfades in $\Lambda^{(i,j),(r,s)}$.

$(\tilde{g}^{(i,j),(r,s)}, Dir^*, \psi^{min}) = \text{AlgorithmusIVb}(\tilde{\Lambda}^{(i,j),(r,s)})$

- Kommentar:** Eingang: $\tilde{\Lambda}^{(i,j),(r,s)}$ ist $n_{i,j} \times n_{r,s}$ -Matrix, in der der Pfad mit maximaler Summe berechnet werden soll.
- Ausgang: $\tilde{g}^{(i,j),(r,s)}$ ist die globale Ähnlichkeit zwischen den Konturen $K_{i,j}$ und $K_{r,s}$.
- Ausgang: Dir^* ist diejenige Richtungsmatrix, die zu dem Pfad mit maximalster Summe gehört.
- Ausgang: ψ^{min} ist der Index t desjenigen Konturpunktes in $K_{r,s}$, der mit dem 1. Konturpunkt aus $K_{i,j}$ bzgl. maximaler Summe korrespondiert.

$t := 1$

$g^{(i,j),(r,s)} := 0$

while $t \leq n_{r,s}$

$M^t := \begin{pmatrix} \Lambda_{1..n_{i,j}, t..n_{r,s}}^{(i,j),(r,s)} & \Lambda_{1..n_{i,j}, 1..t-1}^{(i,j),(r,s)} \end{pmatrix}$

$M^{t+} := \begin{pmatrix} M^t & M_{1..n_{i,j}, 1}^t \\ M_{1, 1..n_{r,s}}^t & M_{1, 1}^t \end{pmatrix}$

$(Dir, g^*) := \text{AlgorithmusIIIb}(M^{t+})$

if $g^* \geq g^{(i,j),(r,s)}$

$g^{(i,j),(r,s)} := g^*$

$Dir^* := Dir$

$\psi^{min} := t$

end

$t := t + 1$

end

Abbildung E.8: Algorithmus IVb zur Berechnung des günstigsten Pfades in $\tilde{\Lambda}^{(i,j),(r,s)}$.

$(Q^{(i,j),(r,s)}, Z^{(i,j),(r,s)}) = \text{AlgorithmusV}(Dir^*, \psi^{min})$

Kommentar: Eingang: Dir^* ist diejenige Richtungsmatrix, die zu dem Pfad mit minimalsten Kosten gehört.

Eingang: ψ^{min} ist der Index t desjenigen Konturpunktes in $K_{r,s}$, der mit dem 1. Konturpunkt aus $K_{i,j}$ bzgl. minimaler Kosten korrespondiert.

Ausgang: $Q^{(i,j),(r,s)}$ ist die Korrespondenzliste von $K_{i,j}$ und $K_{r,s}$ für minimale Kosten.

Ausgang: $Z^{(i,j),(r,s)}$ ist die Anzahl der Elemente q_z in $Q^{(i,j),(r,s)}$.

$q_1 := (1, \psi^{min})$

$z := 2$

while $Dir_{q_z}^* \neq \text{end}$

if $Dir_{q_{z-1}}^* = S$

$q_z := (q_{z-1,1} + 1, (q_{z-1,2} + 0) \bmod n_{r,s})$

elseif $Dir_{q_{z-1}}^* = O$

$q_z := (q_{z-1,1} + 0, (q_{z-1,2} + 1) \bmod n_{r,s})$

else

$q_z := (q_{z-1,1} + 1, (q_{z-1,2} + 1) \bmod n_{r,s})$

end

$z := z + 1$

end

$Z^{(i,j),(r,s)} := z - 1$

Abbildung E.9: Algorithmus V für das Erzeugen der Korrespondenzliste $Q^{(i,j),(r,s)}$ in $\Lambda^{(i,j),(r,s)}$. mod steht für modulo-Rechnung.

$$(w^{(I,J),(R,S)}, a^{(I,J),(R,S)}) = \text{AlgorithmusVIa} \left(Q^{(I,J),(R,S)}, Z^{(I,J),(R,S)}, \Lambda^{(I,J),(R,S)}, \Theta, n_{I,J}, n_{R,S} \right)$$

Kommentar: Eingang: $Q^{(I,J),(R,S)}$ ist Korrespondenzliste.
 Eingang: $Z^{(I,J),(R,S)}$ ist Anzahl der Elemente in $Q^{(I,J),(R,S)}$.
 Eingang: $\Lambda^{(I,J),(R,S)}$ ist lokale Distanzmatrix.
 Eingang: Θ ist Importanzschwelle.
 Eingang: $n_{I,J}$ ist Anzahl der Konturpunkte von Kontur $K_{I,J}$.
 Eingang: $n_{R,S}$ ist Anzahl der Konturpunkte von Kontur $K_{R,S}$.
 Ausgang: $w^{(I,J),(R,S)}$ ist Liste wichtiger Konturabschnitte von $K_{I,J}$, bezogen auf Kontur $K_{R,S}$.
 Ausgang: $a^{(I,J),(R,S)}$ ist die Anzahl wichtiger Konturabschnitte in $w^{(I,J),(R,S)}$.

$z := 0$

$p := 0$

do

$z := z + 1$

if $\Lambda_{Q_{z,1}^{(I,J),(R,S)}, Q_{z,2}^{(I,J),(R,S)}}^{(I,J),(R,S)} > \Theta$

$p := p + 1$

$q := 0$

while $\Lambda_{Q_{z,1}^{(I,J),(R,S)}, Q_{z,2}^{(I,J),(R,S)}}^{(I,J),(R,S)} > \Theta$

$q := q + 1$

if $(z > 1) \wedge (q > 1) \wedge (Q_{z,1}^{(I,J),(R,S)} \ll Q_{z-1,1}^{(I,J),(R,S)})$

$w_{p,q}^{(I,J),(R,S)} := P_{I,J,Q_{z,1}^{(I,J),(R,S)}}$

end $z := z + 1$

if $z > Z^{(I,J),(R,S)}$

goto Konturschlussbehandlung

end

end

end

while $z < Z^{(I,J),(R,S)}$

Konturschlussbehandlung:

$a^{(I,J),(R,S)} := p$

if $\left(\Lambda_{Q_{1,1}^{(I,J),(R,S)}, Q_{1,2}^{(I,J),(R,S)}}^{(I,J),(R,S)} > \Theta \right) \wedge \left(\Lambda_{Q_{Z^{(I,J),(R,S)},1}^{(I,J),(R,S)}, Q_{Z^{(I,J),(R,S)},2}^{(I,J),(R,S)}}^{(I,J),(R,S)} > \Theta \right)$

$w_1^{(I,J),(R,S)} := \{w_p^{(I,J),(R,S)}, w_1^{(I,J),(R,S)}\}$

$a^{(I,J),(R,S)} := p - 1$

end

Abbildung E.10: Algorithmus VIa zur Berechnung von $w^{(I,J),(R,S)}$, basierend auf der lokalen Distanzmatrix $\Lambda^{(I,J),(R,S)}$.

$(\tilde{w}^{(I,J),(R,S)}, \tilde{a}^{(I,J),(R,S)}) = \text{AlgorithmusVib} (\tilde{Q}^{(I,J),(R,S)}, \tilde{Z}^{(I,J),(R,S)}, \tilde{\Lambda}^{(I,J),(R,S)}, \tilde{\Theta}, n_{I,J}, n_{R,S})$

Kommentar: Eingang: $\tilde{Q}^{(I,J),(R,S)}$ ist Korrespondenzliste.
 Eingang: $\tilde{Z}^{(I,J),(R,S)}$ ist Anzahl der Elemente in $\tilde{Q}^{(I,J),(R,S)}$.
 Eingang: $\tilde{\Lambda}^{(I,J),(R,S)}$ ist lokale Distanzmatrix.
 Eingang: $\tilde{\Theta}$ ist Importanzschwelle.
 Eingang: $n_{I,J}$ ist Anzahl der Konturpunkte von Kontur $K_{I,J}$.
 Eingang: $n_{R,S}$ ist Anzahl der Konturpunkte von Kontur $K_{R,S}$.
 Ausgang: $\tilde{w}^{(I,J),(R,S)}$ ist Liste wichtiger Konturabschnitte von $K_{I,J}$ bezogen auf Kontur $K_{R,S}$.
 Ausgang: $\tilde{a}^{(I,J),(R,S)}$ ist die Anzahl wichtiger Konturabschnitte in $\tilde{w}^{(I,J),(R,S)}$.

$z := 0$

$p := 0$

do

$z := z + 1$

if $\tilde{\Lambda}_{\tilde{Q}_{z,1}^{(I,J),(R,S)}, \tilde{Q}_{z,2}^{(I,J),(R,S)}}^{(I,J),(R,S)} < \tilde{\Theta}$

$p := p + 1$

$q := 0$

while $\tilde{\Lambda}_{\tilde{Q}_{z,1}^{(I,J),(R,S)}, \tilde{Q}_{z,2}^{(I,J),(R,S)}}^{(I,J),(R,S)} < \tilde{\Theta}$

$q := q + 1$

if $(z > 1) \wedge (q > 1) \wedge (\tilde{Q}_{z,1}^{(I,J),(R,S)} <> \tilde{Q}_{z-1,1}^{(I,J),(R,S)})$

$\tilde{w}_{p,q}^{(I,J),(R,S)} := P_{I,J,\tilde{Q}_{z,1}^{(I,J),(R,S)}}$

end $z := z + 1$

if $z > \tilde{Z}^{(I,J),(R,S)}$

goto Konturschlussbehandlung

end

end

end

while $z < \tilde{Z}^{(I,J),(R,S)}$

Konturschlussbehandlung:

$\tilde{a}^{(I,J),(R,S)} := p$

if $\left(\tilde{\Lambda}_{\tilde{Q}_{1,1}^{(I,J),(R,S)}, \tilde{Q}_{1,2}^{(I,J),(R,S)}}^{(I,J),(R,S)} > \tilde{\Theta} \right) \wedge \left(\tilde{\Lambda}_{\tilde{Q}_{\tilde{Z}^{(I,J),(R,S)},1}^{(I,J),(R,S)}, \tilde{Q}_{\tilde{Z}^{(I,J),(R,S)},2}^{(I,J),(R,S)}}^{(I,J),(R,S)} > \tilde{\Theta} \right)$

$\tilde{w}_1^{(I,J),(R,S)} := \{ \tilde{w}_p^{(I,J),(R,S)}, \tilde{w}_1^{(I,J),(R,S)} \}$

$\tilde{a}^{(I,J),(R,S)} := p - 1$

end

Abbildung E.11: Algorithmus Vib zur Berechnung von $\tilde{w}^{(I,J),(R,S)}$, basierend auf der lokalen Distanzmatrix $\tilde{\Lambda}^{(I,J),(R,S)}$.

$(W^{I,J}, C^{I,J}) = \text{AlgorithmusVIIa}(K_{I,J}, L, \Theta, l_{const})$

Kommentar: Eingang: $K_{I,J}$ ist Kontur, für die alle wichtigen Konturabschnitte bestimmt werden sollen.
 Eingang: L ist die Lernstichprobe.
 Eingang: Θ ist die Importanzschwelle.
 Eingang: l_{const} ist die Länge der Konturabschnitte zur Bestimmung der lokalen Distanzmatrix.
 Ausgang: $W^{I,J}$ ist Liste aller wichtigen Abschnitte in $K_{I,J}$.
 Ausgang: $C^{I,J}$ ist Anzahl wichtiger Konturabschnitte in $W^{I,J}$.

$c := 0$

$r := 1$

$s := 1$

while $r \leq m_r$

while $s \leq n_{r,s}$

if $r \neq I$

$\Lambda^{(I,J),(r,s)} := \text{AlgorithmusIIa}(K_{I,J}, K_{r,s}, l_{const})$

$(g^{(I,J),(r,s)}, Dir^*, \psi^{min}) := \text{AlgorithmusIVa}(\Lambda^{(I,J),(r,s)})$

$(Q^{(I,J),(r,s)}, Z^{(I,J),(r,s)}) := \text{AlgorithmusV}(Dir^*, \psi^{min})$

$(w^{(I,J),(r,s)}, a^{(I,J),(r,s)}) := \text{AlgorithmusVIa}(Q^{(I,J),(r,s)}, Z^{(I,J),(r,s)}, \Lambda^{(I,J),(r,s)}, \Theta, n_{I,J}, n_{r,s})$

$p := 1$

while $p \leq a^{(I,J),(r,s)}$

$c := c + 1$

$W_c^{I,J} := w_p^{(I,J),(r,s)}$

$p := p + 1$

end

end

$s := s + 1$

end

$r := r + 1$

end

$C^{I,J} := c$

Abbildung E.12: AlgorithmusVIIa zur Berechnung von $W^{I,J}$ basierend auf den lokalen Distanzmatrizen $\Lambda^{(I,J),(r,s)}$. In $W^{I,J}$ sind alle wichtigen Konturabschnitte einer bestimmten Kontur $K_{I,J}$, bezogen auf alle Konturen anderer Klassen $r \neq I$, enthalten.

$(\tilde{W}^{I,J}, \tilde{C}^{I,J}) = \text{AlgorithmusVIIb}(K_{I,J}, L, \tilde{\Theta}, E)$

Kommentar: Eingang: $K_{I,J}$ ist Kontur, für die alle wichtigen Konturabschnitte bestimmt werden sollen.
 Eingang: L ist die Lernstichprobe.
 Eingang: $\tilde{\Theta}$ ist die Importanzschwelle.
 Eingang: E ist maximal zulässiger Fehler zur Bestimmung der lokalen Ähnlichkeitsmatrix.
 Ausgang: $W^{I,J}$ ist Liste aller wichtigen Abschnitte in $K_{I,J}$.
 Ausgang: $C^{I,J}$ ist Anzahl wichtiger Konturabschnitte in $W^{I,J}$.

$c := 0$

$r := 1$

$s := 1$

while $r \leq m_r$

while $s \leq n_{r,s}$

if $r \neq I$

$\tilde{\Lambda}^{(I,J),(r,s)} := \text{AlgorithmusIIb}(K_{I,J}, K_{r,s}, E)$

$(\tilde{g}^{(I,J),(r,s)}, Dir^*, \psi^{min}) := \text{AlgorithmusIVb}(\tilde{\Lambda}^{(I,J),(r,s)})$

$(\tilde{Q}^{(I,J),(r,s)}, \tilde{Z}^{(I,J),(r,s)}) := \text{AlgorithmusV}(Dir^*, \psi^{min})$

$(\tilde{w}^{(I,J),(r,s)}, \tilde{a}^{(I,J),(r,s)}) := \text{AlgorithmusVIb}(\tilde{Q}^{(I,J),(r,s)}, \tilde{Z}^{(I,J),(r,s)}, \tilde{\Lambda}^{(I,J),(r,s)}, \tilde{\Theta}, n_{I,J}, n_{r,s})$

$p := 1$

while $p \leq a^{(I,J),(r,s)}$

$c := c + 1$

$\tilde{W}_c^{I,J} := \tilde{w}_p^{(I,J),(r,s)}$

$p := p + 1$

end

end

$s := s + 1$

end

$r := r + 1$

end

$\tilde{C}^{I,J} := c$

Abbildung E.13: AlgorithmusVIIb zur Berechnung von $\tilde{W}^{I,J}$, basierend auf den lokalen Distanzmatrizen $\tilde{\Lambda}^{(I,J),(r,s)}$. In $\tilde{W}^{I,J}$ sind alle wichtigen Konturabschnitte einer bestimmten Kontur $K_{I,J}$, bezogen auf alle Konturen anderer Klassen $r \neq I$, enthalten.

$\hat{g} = \text{AlgorithmusVIII}(\mathcal{W}, K)$

Kommentar: Eingang: \mathcal{W} ist Wissensbasis mit M nach Reliabilität geordneten signifikanten Konturabschnitten und zugeordneten Konfidenzen in N Klassen.

Eingang: K ist zu klassifizierende Kontur mit n_K Konturpunkten.

Ausgang: \hat{g} ist Klassifikationsergebnis (Klassenindex).

Bemerkung: Konturpunktindizes müssen alle modulo bzgl. der entsprechenden Konturpunktanzahlen berechnet werden !

$\hat{g} := N + 1$

$no := 1$

while $no \leq M$

$Min := \infty$

$k := 1$

while $k \leq n_K$

$\lambda := \epsilon^{no} - \sigma^{no} + 1$

$A_k := \{P_k, P_{k+1}, \dots, P_{k+\lambda-1}\}$

$$\overline{x_{i,j,\sigma^{no+w}}} := \frac{\sum_{w=0}^{\lambda-1} x_{i,j,\sigma^{no+w}}}{\lambda}$$

$$\overline{x_{k+w}} := \frac{\sum_{w=0}^{\lambda-1} x_{k+w}}{\lambda}$$

$$\Delta_k := \frac{\sum_{w=0}^{\lambda-1} \left((x_{i,j,\sigma^{no+w}} - \overline{x_{i,j,\sigma^{no+w}}}) - (x_{k+w} - \overline{x_{k+w}}) \right)^2 + (y_{i,j,\sigma^{no+w}} - y_{k+w})^2}{\lambda}$$

if $\Delta_k < Min$

$Min := \Delta_k$

end

$k := k + 1$

end

if $Min \leq \tau^{no}$

$\hat{g} := \hat{g}^{no}$

beende Algorithmus VIII

end

$no := no + 1$

end

Abbildung E.14: Algorithmus VIII für den hierarchischen Klassifikator.

