Adaptive Verfahren zur Bewertung texturierter Oberflächen

Von der Fakultät für Maschinenbau der
Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von
Jens Pannekamp
aus Ulm

Hauptberichter:	Univ.-Prof. Dr.-Ing. Prof. e. h. Dr.-Ing. e. h. Dr. h. c. mult. Engelbert Westkämper
Mitberichter:	Prof. Dr. Günter Radons

Tag der Einreichung:	03. Juli 2003
Tag der mündlichen Prüfung:	11. Mai 2004

Institut für Industrielle Fertigung und Fabrikbetrieb der
Universität Stuttgart

2005

IPA-IAO Forschung und Praxis

Berichte aus dem
Fraunhofer-Institut für Produktionstechnik und
Automatisierung (IPA), Stuttgart,
Fraunhofer-Institut für Arbeitswirtschaft und
Organisation (IAO), Stuttgart,
Institut für Industrielle Fertigung und
Fabrikbetrieb (IFF), Universität Stuttgart
und Institut für Arbeitswissenschaft und
Technologiemanagement (IAT), Universität Stuttgart

Herausgeber:
Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper
und
Univ.-Prof. Dr.-Ing. habil. Prof. e.h. Dr. h.c. Hans-Jörg Bullinger

Universität Stuttgart
Institut für Industrielle Fertigung und Fabrikbetrieb

Fraunhofer Institut Produktionstechnik und Automatisierung

Jens Pannekamp

Adaptive Verfahren zur Bewertung texturierter Oberflächen

Nr. 414

JOST-JETTER VERLAG
Fachverlag · 71296 Heimsheim

Dr.-Ing. Jens Pannekamp
Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper
ord. Professor an der Universität Stuttgart
Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

Univ.-Prof. Dr.-Ing. habil. Prof. e.h. Dr. h.c. Hans-Jörg Bullinger
ord. Professor an der Universität Stuttgart
Präsident der Fraunhofer-Gesellschaft, München

D 93

ISBN 3-936947-53-8 Jost Jetter Verlag, Heimsheim

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils gültigen Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.
© Jost Jetter Verlag, Heimsheim 2005.
Printed in Germany.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Sollte in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z. B. DIN, VDI, VDE) Bezug genommen oder aus ihnen zitiert worden sein, so kann der Verlag keine Gewähr für die Richtigkeit, Vollständigkeit oder Aktualität übernehmen. Es empfiehlt sich, gegebenenfalls für die eigenen Arbeiten die vollständigen Vorschriften oder Richtlinien in der jeweils gültigen Fassung hinzuzuziehen.

Druck: printsystem GmbH, Heimsheim

Geleitwort der Herausgeber

Über den Erfolg und das Bestehen von Unternehmen in einer marktwirtschaftlichen Ordnung entscheidet letztendlich der Absatzmarkt. Das bedeutet, möglichst frühzeitig absatzmarktorientierte Anforderungen sowie deren Veränderungen zu erkennen und darauf zu reagieren.

Neue Technologien und Werkstoffe ermöglichen neue Produkte und eröffnen neue Märkte. Die neuen Produktions- und Informationstechnologien verwandeln signifikant und nachhaltig unsere industrielle Arbeitswelt. Politische und gesellschaftliche Veränderungen signalisieren und begleiten dabei einen Wertewandel, der auch in unseren Industriebetrieben deutlichen Niederschlag findet.

Die Aufgaben des Produktionsmanagements sind vielfältiger und anspruchsvoller geworden. Die Integration des europäischen Marktes, die Globalisierung vieler Industrien, die zunehmende Innovationsgeschwindigkeit, die Entwicklung zur Freizeitgesellschaft und die übergreifenden ökologischen und sozialen Probleme, zu deren Lösung die Wirtschaft ihren Beitrag leisten muss, erfordern von den Führungskräften erweiterte Perspektiven und Antworten, die über den Fokus traditionellen Produktionsmanagements deutlich hinausgehen.

Neue Formen der Arbeitsorganisation im indirekten und direkten Bereich sind heute schon feste Bestandteile innovativer Unternehmen. Die Entkopplung der Arbeitszeit von der Betriebszeit, integrierte Planungsansätze sowie der Aufbau dezentraler Strukturen sind nur einige der Konzepte, welche die aktuellen Entwicklungsrichtungen kennzeichnen. Erfreulich ist der Trend, immer mehr den Menschen in den Mittelpunkt der Arbeitsgestaltung zu stellen - die traditionell eher technokratisch akzentuierten Ansätze weichen einer stärkeren Human- und Organisationsorientierung. Qualifizierungsprogramme, Training und andere Formen der Mitarbeiterentwicklung gewinnen als Differenzierungsmerkmal und als Zukunftsinvestition in *Human Resources* an strategischer Bedeutung.

Von wissenschaftlicher Seite muss dieses Bemühen durch die Entwicklung von Methoden und Vorgehensweisen zur systematischen Analyse und Verbesserung des Systems Produktionsbetrieb einschließlich der erforderlichen Dienstleistungsfunktionen unterstützt werden. Die Ingenieure sind hier gefordert, in enger Zusammenarbeit mit anderen Disziplinen, z. B. der Informatik, der Wirtschaftswissenschaften und der Arbeitswissenschaft, Lösungen zu erarbeiten, die den veränderten Randbedingungen Rechnung tragen.

Die von den Herausgebern langjährig geleiteten Institute, das

- Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA),
- Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO),
- Institut für Industrielle Fertigung und Fabrikbetrieb (IFF), Universität Stuttgart,
- Institut für Arbeitswissenschaft und Technologiemanagement (IAT), Universität Stuttgart

arbeiten in grundlegender und angewandter Forschung intensiv an den oben aufgezeigten Entwicklungen mit. Die Ausstattung der Labors und die Qualifikation der Mitarbeiter haben bereits in der Vergangenheit zu Forschungsergebnissen geführt, die für die Praxis von großem Wert waren. Zur Umsetzung gewonnener Erkenntnisse wird die Schriftenreihe „IPA-IAO - Forschung und Praxis“ herausgegeben. Der vorliegende Band setzt diese Reihe fort. Eine Übersicht über bisher erschienene Titel wird am Schluss dieses Buches gegeben.

Dem Verfasser sei für die geleistete Arbeit gedankt, dem Jost Jetter Verlag für die Aufnahme dieser Schriftenreihe in seine Angebotspalette und der Druckerei für saubere und zügige Ausführung. Möge das Buch von der Fachwelt gut aufgenommen werden.

Engelbert Westkämper Hans-Jörg Bullinger

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit am Institut für Industrielle Fertigung und Fabrikbetrieb (IFF) der Universität Stuttgart sowie am Fraunhofer Institut für Produktionstechnik und Automatisierung (IPA). Im Rückblick auf die Jahre, in denen diese Arbeit entstand, wird mir bewußt, von wie vielen Seiten ich Unterstützung erfahren habe.

Zuallererst danke ich Herrn Prof. Westkämper für die wohlwollende Begleitung und Förderung. Weiterhin gilt mein Dank Herrn Prof. Günter Radons für wertvolle Anregungen und die Übernahme des Mitberichts.

Als Keimzelle dieser Dissertation sehe ich die wissenschaftliche Ausrichtung der Abteilung „Technische Informationsverarbeitung“ durch Herrn Dr. Wolfgang Rauh an. Ihm danke ich auch für die gezielte Förderung und Durchsicht der Arbeit. Meinem Mentor Dr. Manfred Schmutz gilt mein herzlichster Dank. Er hat mein Wissenschaftsbild nachhaltig geprägt. Ohne seine Ideen und inspirierenden Diskussionen wäre diese Arbeit nicht so wie sie ist.

Für die freundschaftliche Zusammenarbeit und vielfache Hilfestellungen bedanke ich mich bei meinen Kollegen, insbesondere bei Herrn Markus Hüttel.

Der Deutschen Forschungsgemeinschaft danke ich für die Unterstützung der Arbeiten im Rahmen des Schwerpunktprogramms „Automatische Sichtprüfung technischer Objekte“ (SPP 735).

Stuttgart, März 2005 Jens Pannekamp

Inhaltsverzeichnis

Abkürzungen **13**

1 Einleitung **15**

2 Grundlagen **18**
2.1 Texturen 18
2.2 Bewertung von Texturen 19
2.3 Texturmerkmale 21
2.4 Klassifikation 23
2.5 Verarbeitungsschemata 24
2.5.1 Standardansatz 24
2.5.2 Neuronaler Gesamtansatz 25
2.5.3 Merkmalsverteilungen 25
2.5.4 Bewertung von Merkmalssätzen 26
2.5.5 Merkmalsselektion 26
2.5.6 Merkmalsoptimierung 27
2.6 Anwendungsbereiche 27
2.7 Defizite vorhandener Ansätze 28

3 Systementwurf **29**
3.1 Anforderungen 30
3.2 Lösungsansatz 30
3.3 Benutzerschnittstellen 33

4 Optimierung von Texturmerkmalen **35**
4.1 Merkmale der 2-Punkt-Statistik 36
4.1.1 Grundlagen 36
4.1.2 Robuste Merkmalsextraktion 38
4.1.3 Multiskalenansatz 41
4.1.4 Rotationsinvarianz 42
4.1.5 Randomisierung 43
4.1.6 Mehrkanalbilder 43
4.2 Lokale binäre Muster 44

4.2.1 Grundlagen 44
4.2.2 Multiskalenansatz 45
4.2.3 Rotations- und Spiegelinvarianz 45
4.2.4 Randomisierung 46
4.2.5 Mehrkanalbilder 47
4.2.6 Klassifikation 47
4.3 Gaborfilter 48
4.3.1 Grundlagen 48
4.3.2 Optimierung der Filterbank 49

5 Aspekte der Klassifikation 51
5.1 Kleine Lernstichproben 51
5.1.1 Schätzfehler 52
5.1.2 Instationarität 53
5.2 Ähnlichkeit zu Texturklassen 54
5.3 Rotationsinvariantes Klassifikationsschema 57
5.4 Implizite Merkmalsgewichtung 58
5.5 Visuelle Ähnlichkeit 61

6 Merkmalsselektion 63
6.1 Motivation 63
6.2 Gütekriterien 64
6.2.1 Klassifikatorspezifische Kriterien 65
6.2.2 A-posteriori Wahrscheinlichkeit 70
6.2.3 Gütekriterium für kontinuierliche Bewertung 71
6.3 Suchverfahren 71
6.3.1 Sequentielle Suche 72
6.3.2 Gleitende sequentielle Suche 72
6.3.3 Branch-And-Bound 73
6.3.4 Vorwärtssuche vs. Rückwärtssuche 73
6.4 Experimente 74

7 Merkmalsoptimierung 77
7.1 CMA-Evolutionsstrategie 78
7.2 Optimierung von Gaborfiltern 79

8 Vergleichende Untersuchungen 82
8.1 Rotationsinvariante Texturklassifikation 82
8.1.1 Konzeption der Experimente 82
8.1.2 Datensatz „Brodatz-Texturen“ 85
8.1.3 Ergebnisse und Diskussion 87
8.2 Textursegmentierung 93
8.2.1 Konzeption der Experimente 93

8.2.2 Datensatz „Mosaic08“ 95
8.2.3 Ergebnisse und Diskussion 96
8.3 Selektion vs. Optimierung 103
8.3.1 Konzeption der Experimente 103
8.3.2 Ergebnisse und Diskussion 104

9 Anwendungsbeispiele 109
9.1 Medizinische Kollagenschwämme 109
9.1.1 Prüfaufgabe 109
9.1.2 Realisierung 111
9.1.3 Validierung des Prüfsystems 111
9.1.4 Fazit 116
9.2 Kontinuierliche Bewertung 117
9.2.1 Bewertung von Schleifpapieren 119
9.2.2 Ergebnisse 120
9.3 Texturanalysesystem Adaptex 122
9.3.1 Zielsetzung 122
9.3.2 Realisierung 122
9.3.3 Beispiel 123

10 Zusammenfassung und Ausblick 126

11 Summary 130
11.1 Introduction 130
11.2 Rotation-Invariant Texture Features 130
11.2.1 Local Binary Patterns (LBP) 131
11.2.2 Sum and Difference Histograms (HP2) 132
11.3 Rotation Invariance by Randomisation 132
11.4 Feature Selection 133
11.5 Experiments 134
11.6 Results and Discussion 134
11.7 Conclusion 138

Literaturverzeichnis 147

Lebenslauf 148

Verwendete Abkürzungen und Formelzeichen

Abkürzungen

DST	Distanzklassifikator
HP2	Histogrammbasierte Merkmale der 2-Punkt-Statistik
HP2-ROT	Rotationsinvariante Merkmale vom Typ HP2
LBP	Lokales binäres Muster (Local Binary Pattern)
LBP-ROT	Rotationsinvariante Merkmale vom Typ LBP
NDS	Normalverteilungsklassifikator
SUBn	Multiskalenanalyse durch n-fache Unterabtastung

Arabische Formelzeichen

B	Anzahl der Nachbarbildpunkte bei LBP
C	Anzahl der Farbkanäle des Bildes
C_{ij}	Cooccurrence- oder Abhängigkeitsmatrix
d	Abstand zweier Merkmalsvektoren
D	Abstand des Bildpunktpaares bei LBP und HP2
f	Texturmerkmal
F	Größe des Analysefensters
g	Bildmatrix
G	Anzahl der Graustufen
h_{dif}	Differenzhistogramm
h_{sum}	Summenhistogramm
J	Gütekriterium für die Merkmalsselektion
J_{Bhat}	Bhattacharyya Abstandsmaß
$J_{\mathrm{Kön}}$	Nichtparametrisches Überlappungsmaß (König)
J_{Kont}	Gütekriterium für die kontinuierliche Bewertung
J_{Kull}	Kullback-Leibler Abstandsmaß
J_{min}	Minimaler Nächste-Klassen-Abstand
J_{Prob}	Gütekriterium auf Basis der a-posteriori Wahrscheinlichkeit
k	Anzahl der nächsten Nachbarn

ABKÜRZUNGEN

K	Anzahl der Skalierungsstufen
L	Anzahl der Texturklassen bzw. Dimension des kontinuierlichen Bewertungsvektors y
m	Anzahl der selektierten Merkmale
M	Anzahl der Merkmale
N	Anzahl der Mustervektoren
O_φ	Texturoperator mit Orientierung φ
p_i	Wahrscheinlichkeitsdichte der Klasse ω_i
P	Gesamtzahl der Parameter bei der Merkmalsoptimierung
P_i	A-priori Wahrscheinlichkeit der Klasse ω_i
q_i	A-posteriori Wahrscheinlichkeit der Klasse ω_i
r	Referenzvektor
R	Anzahl der Analyserichtungen
s_i	Ähnlichkeit von x zur Klasse ω_i
S	Standardabweichung des Gütekriteriums J
T	Anzahl der Texturfilter bei der Merkmalsoptimierung
T	Transformation des Bildes
u	Ortsvektor
U	Untermenge der primären Merkmale
v	Verschiebungsvektor für Cooccurrence-Matrix
x	Merkmalsvektor
y	Kontinuierlicher Bewertungsvektor
Y	Menge der primären Merkmale
z	Klassifikationsvorgabe für Trainingsvektor
Z	Anzahl der Histogrammeinträge

Griechische Formelzeichen

α	Merkmal
Γ	Maß für die Rotationsinvarianz
$\Delta\varphi$	Winkeldifferenz zwischen Analyserichtungen
ε	Schätzfehler eines Merkmals
κ	Klassifikator
λ	Eigenwert
μ_i	Erwartungswert der Klase ω_i
φ	Orientierung des Texturoperators O_φ, speziell des Verschiebungsvektors v
Φ	Eigenvektor
ρ^{-1}	Inverse Winkelauflösung des Texturoperators O_φ
Σ_i	Kovarianzmatrix der Klasse ω_i
Σ	Kovarianzmatrix der CMA-Evolutionsstrategie
ω_i	Klasse i

Kapitel 1

Einleitung

Viele technische und natürliche Gegenstände zeichnen sich durch eine charakteristische Ausprägung ihrer Oberflächenstruktur aus, die eine Unterscheidung und Bewertung der Objekte erlaubt. Diese Oberflächenstrukturen – auch Texturen genannt – sind Untersuchungsgegenstand der vorliegenden Arbeit. Ausgangspunkt für die technische Analyse von Texturen sind zweidimensionale Bilder der Objektoberfläche, die typischerweise von einer digitalen Kamera erzeugt werden. Zielsetzung ist die Interpretation der Bilder, d. h. der Übergang von einer signalbasierten zu einer symbolischen Beschreibung. Beispielsweise soll eine geschliffene Metalloberfläche mit lokalem Defekt nicht mehr als Folge von Grauwerten dargestellt werden, sondern als Fläche der Textur A mit Einschluss der Textur B. Die Erforschung und technische Umsetzung der menschlichen Fähigkeit, Oberflächen gemäß ihrer Struktur zu gruppieren und zuzuordnen, ist Aufgabe der Texturanalyse.

Der Einsatz der Texturanalyse konzentriert sich bisher auf Bereiche, in denen große Datenmengen nach einheitlichem Schema auszuwerten sind, wie etwa in der Luftbildauswertung oder in der Qualitätssicherung. Dem Stand der Technik entsprechen Systeme, die auf einem fest definierten Satz von Texturmerkmalen basieren. Für jeden Bildausschnitt werden die Merkmale berechnet und einem Klassifikator zugeführt, der dem Bildausschnitt die wahrscheinlichste Klasse zuweist. Häufig erfolgt die Klassifikation durch die Spezifikation von Schwellwerten für die Merkmale, alternativ werden statistische Verfahren wie die Nächste-Nachbar-Klassifikation eingesetzt. Nachteil dieser Vorgehensweise ist, dass die Systemadaption ein hohes Maß an Expertenwissen voraussetzt. Insbesondere die Auswahl und Optimierung der Texturmerkmale ist hier zu nennen.

Für die Zukunft zeichnen sich Anwendungsgebiete in dynamischer Umgebung ab, etwa die Qualitätssicherung in der variantenreichen Fertigung, die inhaltsbasierte Suche in Bilddatenbanken oder Anwendungen im Bereich des Robot Vision. Schnell wechselnde Aufgabenstellungen erfordern dann von den Bildanalysesystemen ein höheres Maß an Flexibilität sowie die Fähigkeit zur

Selbstadaption. Weiterhin ist erkennbar, dass der Einsatzbereich maschineller Bildanalysesysteme immer stärker mit der Domäne menschlichen Handelns überlappen wird. Damit nimmt die Notwendigkeit zur Kommunikation zwischen Mensch und Maschine zu. Kommunikationsfähigkeit bedeutet, dass

- ein Austausch über die Definition von Texturklassen möglich ist und
- Entscheidungsprozesse von der Texturinstanz zur Texturklasse nachvollziehbar sind.

Um diese Ziele zu erreichen, ist die Weiterentwicklung der Mensch-Maschine-Schnittstelle erforderlich. Intern muss die maschinelle Verarbeitung so aufgebaut werden, dass eine Korrespondenz zwischen menschlicher und maschineller Texturbewertung vorliegt. Insbesondere muss die maschinell ermittelte Ähnlichkeit zwischen Texturen der visuell empfundenen Ähnlichkeit entsprechen. In die Sprache der Technik übersetzt heißt das, dass das maschinelle System in den Punkten Merkmalsdefinition, Klassenbildung, Ähnlichkeitsmaßen, Invarianzen und Merkmalsoptimierung auf das menschliche Vorbild abgestimmt werden muss, um ein uns vertrautes Verhalten aufzuweisen.

Die vorliegende Arbeit ist in diesen Kontext eingebettet. Ihr konkretes Ziel ist die Realisierung eines Systems zur adaptiven Texturanalyse („Adaptex“). Dieses soll die Übertragung einer manuell durchgeführten Texturprüfung auf ein maschinelles System ermöglichen, ohne dass der Anwender über technisches Spezialwissen verfügen muss. Der Anwender gibt vor, *was* unterschieden werden soll, während das System mit Hilfe von adaptiven Verfahren festlegt, *wie* diese Vorgaben umzusetzen sind. Die Interaktion des Anwenders beschränkt sich darauf, exemplarisches Bildmaterial in einem grafischen Editor zu bewerten. Je nach Komplexität der Prüfaufgabe dauert die sich anschließende automatische Adaption zwischen mehreren Sekunden bis zu einigen Minuten. Damit ist der Einsatz der Verfahren auch für kleine oder schnell wechselnde Prüfaufgaben möglich und wirtschaftlich sinnvoll.

Das nächste Kapitel dieser Arbeit gibt eine Einführung in die *Grundlagen der Texturanalyse*. In Kapitel 3 werden die Anforderungen an das zu entwickelnde Texturanalysesystem zusammengestellt und auf dieser Grundlage das *Systemkonzept* von Adaptex entwickelt.

Die Kapitel 4 bis 7 entsprechen in ihrer Reihenfolge dem Informationsfluss und führen von der Merkmalsdefinition über die Klassifikation bis hin zur Merkmalsselektion und -optimierung.

In Kapitel 4 werden *Texturmerkmale*, die auf der 2-Punkt-Statistik bzw. auf lokalen binären Mustern basieren, optimiert. Ziel ist die Erzeugung von robusten, rotationsinvarianten Merkmalen sowie die Texturcharakterisierung über mehrere Auflösungsstufen hinweg. Weiterhin werden Gaborfilter so parametrisiert, dass ihr Einsatz für die Merkmalsoptimierung möglich wird.

Kapitel 5 beschäftigt sich mit verschiedenen Aspekten der *Klassifikation*. Neben dem Einfluss kleiner Lernstichproben auf das Klassifikationsverhalten wird untersucht, wie die Ähnlichkeit eines Merkmalsvektors zu einer Texturklasse quantifiziert werden kann. Unter dem Stichwort der impliziten Merkmalsgewichtung wird der Frage nachgegangen, welchen Einfluss die Wahl des Abstandsmaßes auf die Robustheit der Klassifikation hat.

Schwerpunkt von Kapitel 6 ist die *Merkmalsselektion*. Verschiedene Gütekriterien und Suchverfahren werden vorgestellt. Dabei wird auf die enge Wechselwirkung zwischen Gütekriterium und verwendetem Klassifikatortyp eingegangen. Der modulare Systemaufbau ermöglicht die freie Kombinierbarkeit von Gütekriterium und Suchstrategie.

Eine Alternative zur Merkmalsselektion ist die *Merkmalsoptimierung*. Der in Kapitel 7 entwickelte Ansatz beruht auf dem Einsatz von Evolutionsstrategien. Mit ihnen werden die Parameter von Gaborfiltern optimiert.

Die in den Kapiteln 4 bis 7 entwickelten Verfahren werden umfangreichen *Vergleichstests* unterzogen. Dabei werden Datensätze herangezogen, die sich in der Literatur als Standards etabliert haben. Die erzielten Ergebnisse werden mit den in der Literatur dokumentierten Resultaten verglichen.

Zum Abschluss werden in Kapitel 9 *Anwendungsaspekte* diskutiert. Neben den Ergebnissen aus industriellen Anwendungen wird die Funktionsweise des entstandenen Werkzeugs „Adaptex“ vorgestellt. Eine *Zusammenfassung* der Arbeit findet sich in Kapitel 10.

Kapitel 2

Grundlagen

Dieses Kapitel gibt eine kurze Einführung in die Texturanalyse: Was ist unter dem Begriff „Textur“ zu verstehen und welche Werkzeuge bieten sich zur Bewertung von Texturen an? Nach Vorstellung der gängigen Analyseverfahren werden typische Anwendungsbereiche beschrieben. Bedingt durch die rasche Verbreitung digitaler Technologien ergeben sich neue Anwendungsszenarien für die Texturanalyse, aus denen sich die Zielsetzungen für die vorliegende Arbeit ableiten.

2.1 Texturen

Der Ursprung des Wortes „Textur“ geht auf lateinisch „texere“ zurück, was soviel bedeutet wie „weben“, „flechten“ oder „(kunstvoll zusammen-) fügen“. In dieser Arbeit liegt der Schwerpunkt jedoch nicht auf der Erzeugung (Synthese) von Texturen, sondern auf ihrer Untersuchung (Analyse), d. h. wir stellen die Frage nach den konstituierenden Elementen einer Textur und den Regeln, wie diese zusammmengefügt werden.

Im Bereich der digitalen Bildverarbeitung wird der Begriff „Textur“ folgendermaßen definiert:

- „An image texture is described by the number and types of its primitives and the spatial organization or layout of its primitives.“ (Haralick [Har79])
- „We consider a texture to be a stochastic, possibly periodic, two-dimensional image field.“ (Cross und Jain [CJ83])
- „Eine grundlegende Eigenschaft der Textur ist also das kleine elementare Muster, das sich periodisch oder quasiperiodisch im Raum wiederholt.“ (Jähne [Jäh97b])

Alle Definitionen zielen auf das Zusammenspiel des Begriffspaars „Texturelement“ und „Anordnungsschema“ ab. Die Frage, welche Elemente zur Texturbeschreibung geeignet sind und wie deren Anordnung beschrieben werden kann, steht im Zentrum der Texturanalyse. Ausgehend von der Abstraktion des „Elemente Anordnens“ lassen sich Texturmodelle entwickeln, die in der Lage sind, natürliche und technische Oberflächen zu charakterisieren und zu beschreiben.

Um sich zu vergegenwärtigen, dass das Konzept „Textur“ nicht nur eine abstrakte Vorstellung ist, genügt ein Blick in unsere Umwelt. Nahezu alle Gegenstände, die uns umgeben, zeichnen sich durch Oberflächenbeschaffenheiten aus, die es uns erlauben, die Gegenstände gegeneinander abzugrenzen und Rückschlüsse auf die Objekteigenschaften zu ziehen. Beispiele für natürliche Texturen sind die Schale einer Orange oder die Struktur einer Wolke, für künstliche Texturen dagegen das Muster einer Tapete oder die Riffelung einer Feile. Weitere Beispiele sind in Abbildung 2.1 zu sehen.

2.2 Bewertung von Texturen

Der Fähigkeit, Texturen erkennen und unterscheiden zu können, kommt eine große Bedeutung zu, da die funktionalen bzw. ästhetischen Eigenschaften eines Gegenstands häufig aus seiner Oberflächenbeschaffenheit abgeleitet werden können.

Die Realisierung einer Textur sei durch Angabe einer Bildmatrix g gegeben, wobei die Elemente von g gleichermaßen für Grau-, Farb- oder Höhenwerte stehen können. Jeder Texturrealisierung g wird durch die Funktion f eine funktionale Bewertung y zuordnet:

$$f : g \longrightarrow y \tag{2.1}$$

Für die Bestimmung der zunächst unbekannten Funktion f bieten sich 2 Möglichkeiten an:

Algorithmische Definition: Die Funktion f ist durch Angabe einer Berechnungsvorschrift gegeben.

Interpolation von Beispielen: Die Funktion f ist an N Stützstellen durch die Wertepaare (g_i, y_i), $i = 1 \ldots N$, gegeben. Für alle anderen Werte g ist y durch Interpolation zu berechnen.

Der Ansatz der algorithmischen Definition bietet sich an, wenn die funktionale Bewertung y physikalisch aus der Bildmatrix g abgeleitet werden kann. Dieser Vorgehensweise entspricht die Definition der Rauheitskenngrößen R_a

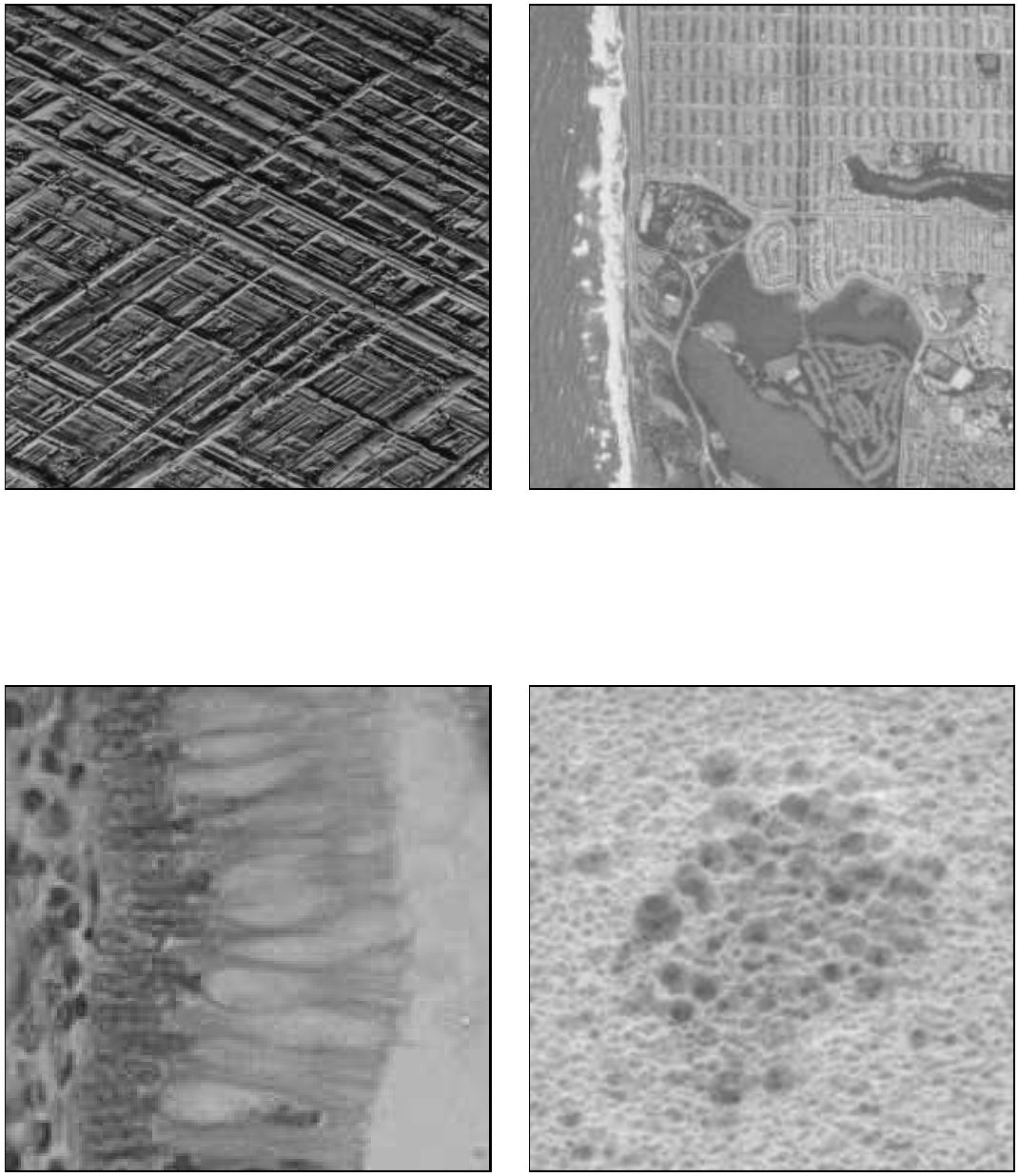


Abbildung 2.1: Beispiele für texturierte Oberflächen: gehonte Oberfläche (links oben), Luftbild (rechts oben), medizinischer Schnitt (links unten) und Schwammstruktur (rechts unten).

und R_z für technische Oberflächen. Sobald die Funktion f algorithmisch festgelegt ist, lässt sich für jede Texturrealisierung g die Bewertung y exakt angeben.

Anders verhält es sich bei dem auf Interpolation beruhenden Ansatz. Hier sind die Bewertungen y_i nur für eine endliche Menge von Texturrealisierungen g_i gegeben. Erst durch Wahl eines Interpolationsschemas wird die Funktion f für alle g festgelegt. Jedoch ist nicht garantiert, dass die so bestimmte Funktion f mit dem menschlichen visuellen Bewertungsempfinden f_{vis} für alle möglichen Texturen g übereinstimmt. Um die Abweichung zwischen f und f_{vis} zu minimieren, muss das Interpolationsschema f in seiner Grundstruktur auf das visuelle Bewertungsschema f_{vis} abgestimmt werden.

Um das Problem zu verdeutlichen, betrachten wir einen Texturausschnitt der Größe 32×32 mit 16 Grauwertstufen. Daraus ergeben sich $16^{32\times 32} \approx 10^{1233}$ unterschiedliche Texturrealisierungen g_i, die sich in einem $32 \times 32 = 1024$ dimensionalen Texturraum anordnen lassen. Leider findet sich in der Literatur kein Hinweis, wieviele visuell unterscheidbare Texturklassen ω_i damit einhergehen. Schätzen wir ihre Zahl L mit 10^6 ab, so werden im Durchschnitt 10^{1227} Texturrealisierungen auf ein- und dieselbe Texturklasse ω_i abgebildet! Aufgabe der Interpolation ist es, von den Bewertungen an den N Stützstellen auf die Bewertungen, d.h. Texturklassen, aller Texturrealisierung zu schließen. Jedoch kann nicht davon ausgegangen werden, dass die Texturklassen im oben erwähnten Texturraum kompakte Cluster bilden.

Zur Lösung des Problems müssen die Eigenschaften der visuellen Wahrnehmung beim Menschen berücksichtigt werden. Im Zentrum steht die Frage nach den Prinzipien, die für die Abbildung einer großen Zahl unterschiedlichster Texturrealisierungen auf den gleichen visuellen Eindruck verantwortlich sind. Julesz [Jul75] konnte beispielsweise zeigen, dass Texturen nicht differenziert werden können, wenn sie in der Statistik erster und zweiter Ordnung übereinstimmen, sich jedoch in der Statistik dritter Ordnung unterscheiden. Fließen derartige Erkenntnisse in die Konstruktion der Funktion f ein, nimmt die Generalisierungsfähigkeit der maschinellen Bewertung zu. Kurzum: Um mit maschinellen Systeme das Bewertungsverhalten von Menschen nachzuahmen, ist es unumgänglich, die Prinzipien der visuellen Wahrnehmung zu berücksichtigen.

2.3 Texturmerkmale

Mit dem Gedankenexperiment des letzten Abschnitts wurde gezeigt, dass eine direkte Interpolation über dem Texturraum aufgrund der komplexen Verteilung der Texturklassen nicht möglich ist. Werden jedoch über dem Texturraum Merkmale definiert, die sich an die visuelle Wahrnehmung des Menschen angelehnen, lässt sich der Texturraum auf einen Merkmalsraum abbilden, in dem

die Texturklassen kompakte Cluster bilden. Die Funktion der Texturmerkmale besteht folglich darin, die Klassifikationsaufgabe zu vereinfachen, indem allgemeine Eigenschaften der visuellen Wahrnehmung berücksichtigt werden.

Welche Texturmerkmale verwendet werden, ergibt sich aus der Wahl des Texturmodells. In Abschnitt 2.1 wurde der Begriff der Textur durch das Zusammenspiel von Texturelementen und einem Anordnungsschema definiert. Gemäß dieser Vorstellung unterscheiden sich die verschiedenen Texturmodelle dadurch, welche Elemente zur Beschreibung verwendet werden und wie deren räumliche Anordnung erfasst wird. Die Aufgabe der Texturanalyse besteht darin, die freien Parameter eines Texturmodells aus einem Bildausschnitt zu schätzen. Die Modellparameter werden auch als Merkmale x der Textur bezeichnet.

Im Folgenden wird eine Auswahl von gängigen Texturmodellen vorgestellt. Das Element jedes Texturmodells und die Modellparameter werden in Klammern angegeben (Texturelement; Modellparameter). Für umfassendere Darstellungen sei auf die Literatur verwiesen [RH99] [Wag99] [Har79] [WDR76].

Cooccurrence-Matrizen: Die Elemente c_{ij} der Cooccurrence-Matrix C – auch Abhängigkeitsmatrix genannt – geben die Wahrscheinlichkeit wieder, dass der Grauwert i und der Grauwert j an Bildpunkten auftreten, die durch einen Verschiebungsvektor v voneinander getrennt sind. Siehe auch 4.1.1. (Bildpunktpaar; Auftrittswahrscheinlichkeiten) [HSD73].

Summen- und Differenzhistogramme: Eine Vereinfachung der Cooccurrence-Matrizen besteht in der Betrachtung von eindimensionalen Summen- und Differenzhistogrammen h_{sum} bzw. h_{dif}. So gibt die i-te Komponente von h_{sum} die Wahrscheinlichkeit an, dass zwei durch einen Verschiebungsvektor v getrennte Bildpunkte in der Summe den Grauwert i aufweisen. (Bildpunktpaar; Auftrittswahrscheinlichkeiten) [Uns86].

Markov-Zufallsfelder: Die Wahrscheinlichkeit für das Auftreten eines Grauwerts i an einem Punkt u ist durch Binomialverteilung gegeben. Der Parameter der Verteilung leitet sich aus der gewichteten Summe der Nachbarbildpunkte ab. (Bildpunkt plus Umgebung; Gewichtsfaktoren der Summe) [CJ83].

Lokale binäre Muster: Ein lokales binäres Muster ist durch den binären Grauwertvergleich eines zentralen Bildpunktes mit seinen Nachbarn gegeben. Siehe auch Abschnitt 4.2.1. (Bildpunkt plus Umgebung; Auftrittswahrscheinlichkeiten der lokalen binären Muster) [OPH96].

Fourierzerlegung: Die Bildmatrix g wird nach harmonischen Funktionen entwickelt. (Harmonische Schwingungen; Amplituden der Zerlegung) [WDR76].

Gabor-Filter: Ein Satz von Bandpassfiltern mit verschiedenen Zentralfrequenzen und Orientierungen wird auf die Bildmatrix g angewendet. Im Ortsraum entsprechen die Filter gaußförmig moduliertenen harmonischen Schwingungen. Siehe auch 4.3.1.(Lokalisierter Wellenzug; Amplituden der Filterung) [JF91].

Überlagerung von Primitiven: Manche Anwendungen legen es nahe, problemspezifische Primitive zu verwenden. Beispielsweise lassen sich gehonte Oberflächen (siehe Abbildung 2.1) durch die Überlagerung von Riefen modellieren, die von einem Poissonprozess erzeugt werden. (Riefen; Parameter der Poissonverteilung (Riefendichte), Riefenorientierung) [Bey96].

2.4 Klassifikation

Aufgabe der Klassifikation ist es, die Zugehörigkeit des Merkmalsvektors x zu den Texturklassen ω_i zu bestimmen. Der ideale Klassifikator (Bayes-Klassifikator) greift dazu auf die klassenspezifischen Wahrscheinlichkeitsdichten $p_i(x)$ zurück und ordnet jeder Klasse eine a-posteriori Wahrscheinlichkeit q_i zu. Sie gibt an, mit welcher Wahrscheinlichkeit das Ereignis x von der Klasse ω_i erzeugt wurde. Insbesondere für nichtparametrische Klassifikatoren ist es jedoch extrem schwierig, gute Dichteschätzungen $p_i(x)$ im hochdimensionalen Merkmalsraum zu erhalten. Aus diesem Grund begnügt man sich häufig damit, die wahrscheinlichste Klasse zu bestimmen. Die Zugehörigkeit von x zu der wahrscheinlichsten Klasse wird auf 1 gesetzt, die Zugehörigkeiten zu den anderen Klassen auf 0. Anschaulich entspricht dies einer Aufteilung des Merkmalsraumes in Bereiche der jeweils dominierenden Klasse.

Ebenso wie bei den Texturmerkmalen, können für die statistische Klassifikation nur einige Verfahren exemplarisch herausgegriffen werden. Ausführliche Darstellungen sind in der Literatur zu finden [DH73] [Fuk90] [Sch96] [Zel94].

Normalverteilungsklassifikator: Die klassenspezifischen Wahrscheinlichkeitsdichten werden durch Normalverteilungen modelliert. Jede Klasse ω_i ist durch Angabe des Klassenschwerpunktes μ_i und der Kovarianzmatrix Σ_i charakterisiert [Fuk90].

Abstandsklassifikator: Jede Klasse ω_i ist durch einen Referenzvektor r_i gegeben. Der Merkmalsvektor x wird mit allen Referenzvektoren r_i verglichen und der Klasse mit dem kleinsten Abstand zugeordnet. Mögliche Abstandsmaße sind der euklidische Abstand, der Kullback-Leibler Abstand, die Kreuzentropie oder die Korrelation [Fuk90].

Nächste-Nachbar-Klassifikator: Jede Klasse ω_i ist durch eine Vielzahl von Referenzvektoren r_{ij} definiert. Zu einem Merkmalsvektor x werden die k

nächsten Nachbarn unter allen Referenzvektoren bestimmt. Die Klasse mit den meisten Treffern gilt als die wahrscheinlichste Klasse [Das91] [Fuk90].

Lernende Vektorquantisierung: Das Klassifikationsprinzip entspricht der Nächsten-Nachbar-Klassifikation, jedoch werden die Referenzvektoren r_{ij} nicht einfach der Trainingsmenge entnommen. Vielmehr wird eine feste Zahl von Referenzvektoren pro Klasse bestimmt und deren Lage im Merkmalsraum optimiert: Referenzvektoren werden von Trainingsvektoren gleicher Klasse angezogen und von Trainingsvektoren ungleicher Klasse abgestoßen. [Koh97] [Zel94]

Neuronale Netze: Die Abbildung eines Eingangs- auf ein Ausgangsmuster erfolgt über die in Schichten angeordneten Neuronen. Die Verbindungsgewichte, die die Signalweiterleitung zwischen den Neuronen steuern, entsprechen den freien Parametern des Klassifikators. Sie werden durch eine Lernregel, beispielsweise den Backpropagation-Algorithmus, an einen Trainingsdatensatz angepasst [Zel94] [Sch96].

Support-Vector-Machine: Die Trennebene zwischen 2 Klassen bestimmt sich durch ausgezeichnete Vektoren des Trainingsdatensatzes, den sogenannten Support-Vektoren. Um auch linear nicht separierbare Klassen zu trennen, werden die Merkmalsvektoren x in einen höherdimensionalen Merkmalsraum transformiert [CST00].

In Kapitel 5 werden weitere Aspekte der Klassifikation diskutiert, insbesondere die Zuverlässigkeit der Klassifikation sowie die Wechselwirkung zwischen Klassifikation und Merkmalen.

2.5 Verarbeitungsschemata

Mit den skizzierten Verfahren der Merkmalsextraktion und Klassifikation können Systeme zur überwachten Texturklassifikation realisiert werden. Um die Bandbreite der möglichen Verarbeitungsschemata aufzuzeigen, seien einige exemplarisch herausgegriffene Verfahren näher beschrieben.

2.5.1 Standardansatz

Unter „Standardansatz“ ist hier das direkte Verarbeitungsschema mit den Verarbeitungsschritten Merkmalsextraktion und Klassifikation zu verstehen. Im ersten Schritt werden aus dem Bild g die festgelegten Merkmale extrahiert und im Merkmalsvektor x zusammengefasst. Im zweiten Schritt bildet der zuvor trainierte Klassifikator den Merkmalsvektor x auf eine der Texturklassen ω_i ab.

Motiviert wird diese Zerlegung, indem jedem der Verarbeitungsschritte eine spezifische Funktion zugewiesen wird. Aufgabe der Merkmalsextraktion ist es, eine Vorverarbeitung des Bildes vorzunehmen, die sich an den Grundprinzipien der visuellen Wahrnehmung orientiert (vgl. Abschnitt 2.2) und damit in erster Näherung problemunabhängig ist. Die Klassifikation dagegen übernimmt die Abbildung des Merkmalsraumes auf die problemspezifisch definierten Texturklassen.

Beispiele für diesen Ansatz sind in den Vergleichstests von Randen *et al.* und Wagner zu finden [RH99] [Wag99].

2.5.2 Neuronaler Gesamtansatz

Eine zentrale Eigenschaft des Standardansatzes liegt in dem problemunabhängig gewählten Merkmalssatz. Dem steht die Beobachtung entgegen, dass die Leistungsfähigkeit eines Klassifikationssystems gesteigert werden kann, wenn die Merkmale an die vorliegende Aufgabe adaptiert werden.

Die von Jain *et al.* formulierte Idee des neuronalen Gesamtansatzes [JK96] sieht vor, die gesamte Signalverarbeitung von der Bildmatrix bis hin zur zugewiesenen Texturklasse durch ein neuronales Netz zu realisieren. Dabei wird ausgenutzt, dass die Extraktion eines Texturmerkmals durch Filterung mathematisch der Operation eines Neurons ähnelt: In beiden Fällen wird das innere Produkt zwischen einem Signal- und einem Gewichtsvektor berechnet und darauf eine nichtlineare Funktion angewandt. Folglich lässt sich die erste Schicht des neuronalen Netzes mit der Merkmalsextraktion identifizieren, die weiteren Schichten mit der Klassifikation. Dadurch wird es möglich, die bekannten Lernverfahren für neuronale Netze nicht nur für das Klassifikatortraining sondern auch für die Optimierung der Merkmalsextraktion zu nutzen. Für beide Teiladaptionen gilt nun das gleiche Optimierungskriterium, nämlich die Minimierung der Fehlerrate auf dem Trainings- bzw. Testdatensatz.

Aufgrund der großen Zahl freier Modellparameter (Verbindungsgewichte) und des Verzichts auf bewährte Merkmale (vgl. Abschnitt 2.2) muss die Generalisierungsfähigkeit des Systems genau im Auge behalten werden. In dem Vergleichstest von Randen *et al.* [RH99] enttäuschte dieser Ansatz.

2.5.3 Merkmalsverteilungen

Eine Form der Texturbeschreibung besteht darin, die Häufigkeitsverteilung der Ausprägungen des Texturelements anzugeben. Ein einfaches Beispiel hierfür ist das Grauwerthistogramm der Textur. Als Texturmerkmale kommen Kennzahlen wie Kontrast, Energie und Entropie (vgl. Abschnitt 4.1.1) in Frage, die sich aus dem Histogramm extrahieren lassen.

Alterativ hierzu wurde von Unser [Uns86] vorgeschlagen, das gesamte Histogramm für die Texturklassifikation zu verwenden, um den Informationsver-

lust zu umgehen, der mit der Merkmalsextraktion verbunden ist. Unter der Annahme, dass die Komponenten x_i des Histogramms x durch eine multinomiale Verteilung gegeben sind, beträgt die Wahrscheinlichkeit p für die Realisierung x des Histogramms

$$p(x) = \Big(\sum_{i=1}^{M} x_i\Big)! \prod_{i=1}^{M} r_i^{x_i}/x_i! \tag{2.2}$$

Die r_i entsprechen den Referenzwahrscheinlichkeiten, dass die Ausprägungen i des Texturelements (hier: Grauwerte i) auftreten. Sie lassen sich in einer Trainingsphase für jede Texturklasse ermitteln. Als Entscheidungsregel für die Texturklassifikation ergibt sich: Wähle die Klasse ω_j, für die der Ausdruck d_{Log}

$$d_{\text{Log}} = -\sum_{i=1}^{M} x_i \ln r_i \tag{2.3}$$

minimal wird. Dieser Ausdruck entspricht der Kreuzentropie in Abschnitt 4.2.6.

2.5.4 Bewertung von Merkmalssätzen

Um die Leistungsfähigkeit von Texturmerkmalen abzuschätzen, führte Wagner [Wag99] einen breit angelegten Vergleichstest durch. Darin wurden 364 Merkmale aus 18 Merkmalssätzen an 7 Datensätzen erprobt. Als Klassifikator wurde ein Nächste-Nachbar-Klassifikator mit $k = 1$ verwendet. Die Untersuchungen ergaben deutliche Unterschiede in der Leistungsfähigkeit der Merkmalssätze. So lieferten die auf Summen- bzw. Differenzhistogrammen basierenden Merkmale über alle Datensätze gemittelt die besten Ergebnisse. Außerdem zeichneten sich diese Merkmale durch eine geringe Rechenzeit aus.

Vergleichsuntersuchungen dieser Art (siehe auch Randen *et al.* [RH99]) geben wichtige Hinweise, welche Merkmalssätze überdurchschnittlich leistungsfähig sind. Für eine konkrete Anwendung ist jedoch nicht garantiert, dass der Vergleichstestsieger die beste Lösung darstellt. Soll die optimale Lösung gefunden werden, müssen alle Merkmalssätze (manuell) durchgetestet werden.

2.5.5 Merkmalsselektion

Eine Erweiterung zur statischen Bewertung von Merkmalssätzen besteht darin, die Auswahl an Texturmerkmalen für eine konkrete Prüfaufgabe zu optimieren. Erste Untersuchungen dieser Art gehen auf Jain *et al.* [JZ97] zurück. Aus der vergleichsweise kleinen Menge von 18 Merkmalen wurde die Merkmalsuntermenge bestimmt, die die Fehlerrate minimierte. Verfahren dieser Art bieten ein hohes Potential für die automatische Adaption von Bildverarbeitungssystemen.

Ein wesentlicher Vorteil ist darin zu sehen, dass die Optimierung auf Basis von problemunabhängigen Suchverfahren und Gütekriterien durchgeführt werden kann, so dass die Interaktion zwischen Anwender und System auf ein Minimum beschränkt wird.

2.5.6 Merkmalsoptimierung

Während bei der Merkmalsselektion die geeignetesten Merkmale aus einer vordefinierten Menge ausgewählt werden, zielt die Merkmalsoptimierung darauf ab, die Merkmale selbst zu optimieren. Dies kann entweder durch Variation von Parametern in der Merkmalsdefinition erfolgen oder aber durch die Synthese von Merkmalen aus Funktionsbausteinen.

Jacquelin *et al.* [JAH97a] setzte genetische Algorithmen ein, um Merkmale zu optimieren, die den Elementen der Cooccurrence-Matrix entsprechen. Jedes Merkmal c_{ij} ist durch 4 Parameter bestimmt, nämlich die Grauwerte i und j sowie den Verschiebungsvektor $v = (v_1, v_2)^T$. Auch in Disziplinen außerhalb der Bildverarbeitung besteht der Bedarf, Merkmale an spezifische Klassifikationsaufgaben anzupassen. So optimierte Mallet *et al.* [MCKV97] die Koeffizienten von Wavelets mit Hilfe des Quasi-Newton Verfahrens, um mineralogische Spektren zu klassifizieren.

Werden nicht die Parameter von Merkmalen optimiert, sondern die Merkmale durch eine Kombination von Funktionsbausteinen erzeugt, spricht man von genetischer Programmierung [Hüb98] [KTN98]. Jedes Merkmal lässt sich dann als Baumstruktur darstellen, wobei die Knoten des Baumes den Operatoren und die Blätter den Operanden entsprechen. Hübner lehnte sich an die aus den Cooccurrence-Matrizen abgeleiteten Haralick-Merkmale [HSD73] an, so dass Operatoren wie „+“, „×“, „log“ auf die Elemente der Cooccurrence-Matrix c_{ij} und Grauwerte g_i wirken.

2.6 Anwendungsbereiche

Die Bedeutung der Texturanalyse ergibt sich aus der Tatsache, dass Gegenstände anhand ihrer Oberflächenbeschaffenheit erkannt und unterschieden werden können. Ihren Ursprung hat die maschinelle Texturanalyse in der Auswertung von Luft- und Satellitenbilder für militärische und wissenschaftliche Zwecke, die sich im Laufe der 60-iger Jahre etablierte. In der Folge wurden weitere Anwendungsgebiete erschlossen, beispielsweise im Bereich der Geowissenschaften und Fertigungstechnik. Insbesondere in der Qualitätstechnik gewann die Texturanalyse in den 80-iger Jahren zunehmend an Bedeutung. Während anfangs die Fehlerdetektion in homogenen Oberflächen dominierte, erweiterte sich das Anwendungsspektrum nach und nach auf strukturierte Oberflächen wie Holzmaserungen oder geschäumte Materialien. Aufgrund der Komplexität

der eingesetzten Verfahren zur Merkmalsberechnung und Klassifikation ist es bis heute gängige Praxis, dass die Anpassung der Prüfsysteme an neue oder veränderte Prüfaufgaben von Spezialisten durchgeführt wird.

Die anhaltende Leistungssteigerung von Computerhard- und -software sowie deren zunehmende Vernetzung im letzten Jahrzehnt haben der Bildverarbeitung neue Anwendungsbereiche erschlossen. Ihren Ausdruck findet diese Entwicklung in der weiten Verbreitung digitaler Bildinformation in alltäglichen Lebensbereichen wie Kommunikation, Unterhaltung, Medien, Fotografie und Medizin. In dem Maße wie die Zahl der Nutzer anwächst, muß die Leistungsfähigkeit der Verfahren und die Mensch-Maschine-Schnittstelle weiterentwickelt werden. Ziel ist die Entwicklung von Bildverarbeitungssystemen, die auf möglichst natürliche Weise vom Benutzer bedient werden können. Speziell in der Texturanalyse soll der Anwender in die Lage versetzt werden, die Adaption von Prüfsystemen an neue oder veränderte Prüfaufgaben selbst vorzunehmen.

2.7 Defizite vorhandener Ansätze

Trotz hoher Forschungsaktivität auf dem Gebiet der Texturanalyse (vgl. Abschnitt 2.5) existiert keine Standardsoftware, die es dem Benutzer erlaubt, eine Texturprüfaufgabe nach dem Prinzip des überwachten Lernens umzusetzen. Insbesondere der Gedanke, eine automatische Bewertung, Auswahl und Optimierung möglicher Verfahrensschritte durchzuführen, hat sich bisher in der Praxis nicht durchgesetzt. Damit bleibt die Adaption von Texturprüfsystemen an neue oder veränderte Aufgaben nach wie vor dem Experten vorbehalten. Im Einzelnen bestehen die folgenden Defizite:

Suboptimale Merkmalsmengen: Die zu verwendenten Merkmale werden von außen vorgegeben und nicht durch das System selbst an die konkrete Prüfaufgabe adaptiert.

Starre Merkmalsparameter: Ebenso wie die Merkmale selbst werden deren kontinuierliche Parameter von außen vorgegeben und nicht automatisch optimiert.

Unzureichende Rotationsinvarianz: Texturmerkmale weisen in vielen Fällen ein unzureichendes Maß an Rotationsinvarianz auf.

Fehlende Integration: Der Verfahrensvergleich für Merkmalsberechnung, Merkmalsselektion, Merkmalsoptimierung und Klassifikation wird dadurch erschwert, dass konkurrierende Verfahren mangels einheitlicher Schnittstellen nicht gegeneinander ausgetauscht werden können. Damit sind Untersuchungen gemäß dem Ceteris-Paribus-Prinzip nicht möglich.

Kapitel 3

Systementwurf für die adaptive Texturbewertung

Trotz der technologischen Fortschritte im Bereich der Texturanalyse ist die Marktdurchdringung im industriellen Umfeld nach wie vor gering. Die Anwendungen konzentrieren sich auf die Inspektion von Massenprodukten wie Stahlblechen, Kunststofffolien oder Holzoberflächen. Produkte, die sich durch kleine Losgrößen und hohe Variantenvielfalt auszeichnen, werden typischerweise manuell geprüft. Das Potential der automatischen Oberflächeninspektion wird nicht ausgeschöpft, da sich die Verfahren der Texturanalyse bisher nicht fertigungsnah adaptieren lassen und das Verhältnis von Aufwand und Nutzen in einem ungünstigen Verhältnis steht.

Um heute von einer manuellen zu einer automatisierten Inspektion überzugehen, müssen Experten die Brücke zwischen der Prüfaufgabe auf der einen Seite und dem großen Fundus möglicher Analyseverfahren auf der anderen Seite schlagen. Die Kluft zwischen diesen beiden Domänen ist für den Endanwender zu groß. Eine schnelle und flexible Anpassung an sich ändernde Fertigungsbedingungen wird damit unmöglich bzw. zu aufwendig.

Zielsetzung muss deshalb sein, diese Kluft zu schließen, indem die Adaption der Analyseverfahren an eine konkrete Prüfaufgabe automatisiert wird. Der Anwender soll sich darauf beschränken, die Prüfaufgabe zu definieren, während das System selbst die Adaption vornimmt. Neben den Basisalgorithmen der Merkmalsberechnung und Klassifikation muss das System dazu über Metaverfahren verfügen, die in der Lage sind, konkurrierende Verfahren und deren Parametereinstellungen zu bewerten, auszuwählen und zu optimieren. Die Metaverfahren bilden gewissermaßen eine Zwischenschicht zwischen der vom Anwender formulierten Prüfaufgabe und den verfügbaren Basisalgorithmen.

3.1 Anforderungen

Zentrale Anforderung an ein Werkzeug zur adaptiven Texturanalyse ist der schnelle und einfache Übergang von der manuellen zur automatischen Inspektion. Die Anpassung des Systems an eine konkrete Prüfaufgabe muss vom Anwender selbst ohne Spezialwissen durchgeführt werden können.

Weitere Anforderungen, die für den industriellen Einsatz von Bedeutung sind, lauten:

- Robustheit gegenüber Beleuchtungsänderungen: Die Texturbewertung soll nicht von den Beleuchtungsbedingungen abhängen.
- Rotationsinvariante Texturbewertung: Gegeneinander gedrehte Texturen sollen die gleiche Bewertung erhalten.
- Diskrete bzw. kontinuierliche Texturbewertung: Die Texturbewertung soll entweder in diskrete Klassen (Beispiel: iO, niO) oder auf einer kontinuierlichen Skala (Beispiel: $[0, 1]$) erfolgen.
- Bewertung von Mehrkanalbildern: Information aus weiteren spektralen Kanälen soll die Zuverlässigkeit der Klassifikation steigern.
- Ähnlichkeit zu Trainingsdaten: Um Fehlklassifikationen zu vermeiden, sollen Texturen nur dann bewertet werden, wenn sie eine Mindestähnlichkeit zu den Trainingsdaten aufweisen.

3.2 Lösungsansatz

Die Adaption eines konventionellen Texturanalysesystems an eine konkrete Prüfaufgabe (vgl. „Standardansatz“ aus Abschnitt 2.5.1 und Abbildung 3.1 oben), zeichnet sich durch eine lineare Folge von Verarbeitungsschritten aus: Ausgehend von Bildern mit Mustertexturen werden vorab definierte Merkmale extrahiert, die zusammen mit den vorgegebenen Texturbewertungen zum Training eines statistischen Klassifikators verwendet werden. Wesentlicher Nachteil dieses Ansatzes ist, dass Texturmerkmale und Parametereinstellungen vom System nicht variiert werden und damit die spezifischen Eigenschaften der vorliegenden Prüfaufgabe keine Berücksichtigung finden. In der Regel wird dieser Nachteil dadurch kompensiert, dass der Experte eine geeignete Auswahl an Merkmalen bzw. Parametereinstellungen trifft – entweder auf Grund von Erfahrungswissen oder durch zeitaufwendiges Ausprobieren verschiedener Varianten.

Um von dem externen Expertenwissen unabhängig zu werden bzw. das wiederholte händische Durchlaufen des Verarbeitungsschemas zu vermeiden, wird

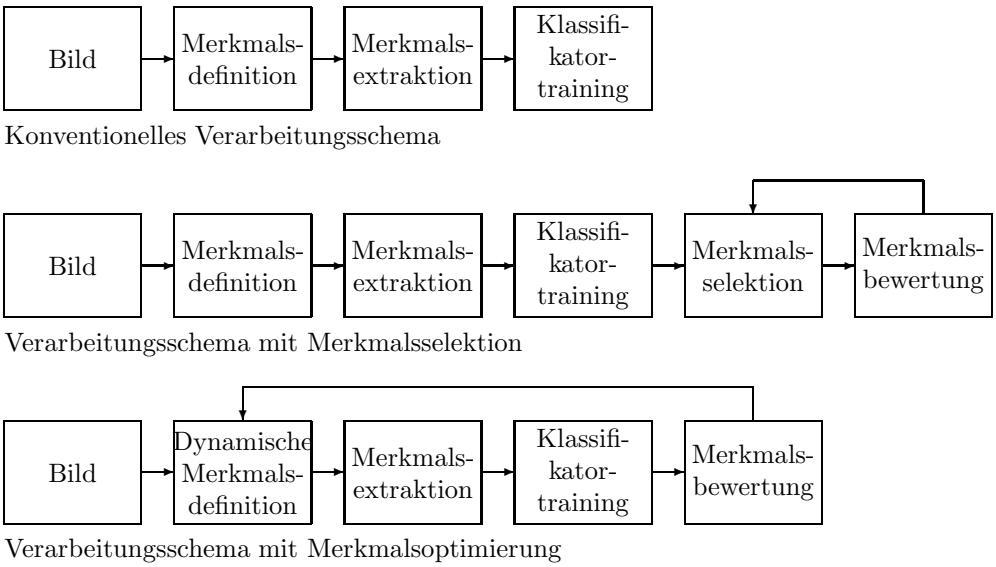


Abbildung 3.1: Verschiedene Verarbeitungsschemata für die Adaption eines Texturanalysesystems. Bei dem konventionellen Vorgehen ist der Satz der Texturmerkmale fest vorgegeben und damit unabhängig von der Prüfaufgabe. Durch Merkmalsselektion bzw. -optimierung werden die Texturmerkmale an die Prüfaufgabe angepasst.

das konventionelle Schema um Rückkopplungen erweitert. Als Prinzip gilt dabei, Merkmalsuntermengen oder Parametereinstellungen solange zu variieren, bis ein Gütekriterium ein Optimum annimmt. In Abbildung 3.1 sind zwei alternative Vorgehensweisen skizziert. Das Verarbeitungsschema mit Merkmalsselektion entspricht zunächst dem konventionellen Schema mit dem Unterschied, dass der berücksichtige Merkmalssatz nicht durch Erfahrungswissen reduziert wird, sondern zunächst alle Merkmale umfasst. Erst in einer Optimierungsschleife, die sich an das Klassifikatortraining anschließt, wird die optimale Merkmalsuntermenge automatisch bestimmt. Bei dem als Merkmalsoptimierung bezeichneten Schema greift die Rückkopplung bis in die Merkmalsdefinition zurück. Hier werden nicht Merkmale aus einem vordefinierten Pool ausgewählt, sondern die Werte von parametrisierten Merkmalen dynamisch verändert. In beiden Fällen bewirkt die Rückkopplung, dass die Texturmerkmale automatisch an die konkrete Prüfaufgabe adaptiert werden.

Die Realisierung der Verarbeitungsschemata mit Merkmalsselektion bzw. Merkmalsoptimierung beruht auf den folgenden Prinzipien und Verfahren:

Modularität

Adaption setzt immer eine Auswahlmöglichkeit voraus. So ist auch bei der adaptiven Texturanalyse das Vorhandensein von alternativen Verfahren von zentraler Bedeutung: In Abhängigkeit von der momentan vorliegenden Prüfaufgabe wird aus dem Pool der möglichen Verfahren das geeignetste Verfahren ausgewählt. Um die Austauschbarkeit zu gewährleisten, müssen die Verfahren über eine einheitliche Schnittstelle verfügen. Eine elegante Implementierung besteht im Einsatz von abstrakten Basisklassen. Die abstrakte Basisklasse gibt die Schnittstellenfunktionen vor, die für jedes der alternativen Verfahren erstellt werden müssen. Auf diese Weise entsteht ein modulares und erweiterungsfähiges System. Abstrakte Basisklassen wurden sowohl für die Texturmerkmale als auch für die Klassifikation entworfen.

Texturmerkmale

Im Vordergrund der Untersuchungen stehen die im Kapitel 4 beschriebenen Merkmale der 2-Punkt-Statistik (HP2) sowie die lokalen binären Muster (LBP). Ebenfalls berücksichtigt werden Merkmale, die auf lokaler Spektralzerlegung beruhen (Gabor- und Fourier-Merkmale). Allen Texturmerkmalen ist gemeinsam, dass sie sowohl in rotationsabhängiger als auch in rotationsinvarianter Form zur Verfügung stehen. Des weiteren ist die Merkmalsextraktion so ausgelegt, dass neben Grauwertbildern auch Bildern mit beliebiger Anzahl von Kanälen C verarbeitet werden können.

Klassifikation

Ebenso wie bei Merkmalsextraktion stehen bei der Klassifikation alternative Verfahren zur Verfügung. Im Einzelnen sind dies Normalverteilungsklassifikatoren (NDS), Abstandsklassifikatoren (DST) mit Kreuzentropie bzw. euklidischem Abstand als Distanzmaß, neuronale Netze (NNT) mit Backpropagation sowie Nächste-Nachbar-Klassifikatoren (KNN). Anders als bei der Merkmalsextraktion findet jedoch bei der Klassifikation keine automatische Auswahl des besten Verfahrens bei der Systemadaption statt. In der Praxis zeigt sich jedoch, dass die Wahl des Klassifikators weit weniger problemspezifisch ist als die Auswahl geeigneter Texturmerkmale.

Merkmalsselektion

Die Merkmalsselektion hat die Auswahl einer optimalen Merkmalsuntermenge zum Ziel. Neben effizienten Suchverfahren sind dazu Gütekriterien notwendig, die die Qualität der Merkmalsuntermengen bewerten. Wichtig ist, dass Gütekriterium und Klassifikatortyp nicht unabhängig voneinander gesehen werden.

Merkmalsoptimierung

Eine Alternative zur diskreten Optimierung der verwendeten Merkmalsuntermenge besteht in der Optimierung reellwertiger Parameter der Merkmale. Als Suchverfahren zur Exploration des Parameterraumes wird die von Rechenberg vorgeschlagende Evolutionsstrategie eingesetzt. Die möglichen Gütekriterien entsprechen denjenigen bei der Merkmalsselektion.

3.3 Benutzerschnittstellen

Mit diesem Lösungsansatz kann der Anwender das System an eine konkrete Prüfaufgabe adaptieren und anschließend weiteres Bildmaterial klassifizieren. Das Vorgehen gliedert sich in die folgenden 3 Schritte:

Interaktive Bewertung

Die interaktive Bewertung ist als Vorbereitungsschritt für die automatische Adaption anzusehen. Gemäß dem Paradigma des überwachten Lernens gibt der Anwender an exemplarischem Bildmaterial vor, wie die unterschiedlichen Texturen zu bewerten sind. Dazu markiert der Anwender in einem grafischen Editor Bildbereiche und ordnet ihnen diskrete oder kontinuierliche Bewertungen zu. Der sich ergebende Satz an bewerteten Mustertexturen definiert die Prüfaufgabe.

Automatische Adaption

Auf Basis des interaktiv bewerteten Bildmaterials wird das System automatisch adaptiert (vgl. Abbildung 3.1). Die Adaption umfasst zum einen das Training des Klassifikators, zum anderen die Adaption der Merkmale. Letztere kann entweder durch Merkmalsselektion oder durch Merkmalsoptimierung erreicht werden. In diesem Schritt kann der Anwender entweder die Grundeinstellungen zur Systemadaption übernehmen oder aber die Vorauswahl von Merkmalen, den Klassifikatortyp und Invarianzeigenschaften selbst bestimmen.

Klassifikation

Ist die Adaption abgeschlossen, steht das System zur automatischen Klassifikation von Texturbildern bereit. Intern setzt sich dieser Prozess aus 4 Schritten zusammen:

1. Parzellierung des Bildes in Flächenelemente
2. Extraktion von Texturmerkmalen für jedes Flächenelement

3. Klassifikation der Flächenelemente
4. Segmentierung des Bildes auf Basis der Klassenzugehörigkeit der Flächenelemente

Der Anwender muss lediglich auswählen, welche Bilddatei mit dem zuvor adaptierten System klassifiziert werden soll.

Kapitel 4

Optimierung von Texturmerkmalen

Jeder Texturbeschreibung liegt eine Modellvorstellung zugrunde, die auf unterschiedliche Eigenschaften der Textur abzielt. Handelt es sich bei den zu analysierenden Texturen um periodische Muster, werden andere Modelle zum Einsatz kommen wie bei der Beschreibung von stochastischen Texturen.

Streng periodische Texturen sind durch die Angabe einer Elementarzelle definiert. Die Frage, ob und in welcher Qualität die Elementarzelle regelmäßig angeordnet ist, weist eine enge Verknüpfung zu den Verfahren des Schablonenvergleichs (Pattern Matchings) auf. Diese Thematik soll hier nicht vertieft werden. Vielmehr steht die Analyse von stochastischen Texturen im Vordergrund. Im Unterschied zu streng periodischen Texturen lassen sich stochastische Texturen nicht durch Angabe einer Elementarzelle beschreiben. Stattdessen müssen die stochastischen Eigenschaften des mustererzeugenden Prozesses herausgearbeitet werden. Die Charakterisierung von stochastischen Texturen darf nicht am unmittelbaren Erscheinungsbild einer Realisierung festgemacht werden, sondern verlangt die Schätzung der zugrundeliegenden (stationären) Modellparametern.

Zielsetzung ist also die Definition von Prozessmodellen, die auf der einen Seite in der Lage sind, eine Vielzahl von Texturen zu charakterisieren und auf der anderen Seite eine robuste Schätzung der Modellparameter zulassen. Dabei stellt sich die Frage, auf welcher Ebene die Modellierung ansetzen soll. Anders ausgedrückt, was sollen die Texturelemente oder Primitive sein, aus deren stochastischer Anordnung die Texturen aufgebaut sind? Im Folgenden werden 2 Ansätze vorgestellt, von denen der erste Ansatz auf der 2-Punkt-Statistik basiert. Als Primitive werden dabei Bildpunktpaare einer Textur betrachtet. Die 2-Punkt-Statistik fragt nach der Wahrscheinlichkeit $p(v, i, j)$, am Ort u den Grauwert i und am Ort $u + v$ den Grauwert j zu finden. Durch den Verschiebungsvektor v ist die Analyserichtung und den Analyseabstand gegeben. Beim

$$\begin{matrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 2 & 2 & 1 & 0 \\ 1 & 1 & 0 & 2 \end{matrix} \qquad C = \frac{1}{4 \cdot 3} \begin{bmatrix} 3 & 1 & 1 \\ 3 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Abbildung 4.1: Links ist ein Bildausschnitt mit Grauwerten aus dem Interval 0 bis 2 dargestellt. Für den Verschiebungsvektor $v = (1, 0)^T$ (Schritt nach rechts) ergibt sich daraus die rechts dargestellte Cooccurrence-Matrix. Das Matrixelement $c_{0,2}$ gibt beispielsweise die Wahrscheinlichkeit für einen Grauwerteübergang von 0 nach 2 an.

zweiten Ansatz werden nicht nur 2 Bildpunkte miteinander verknüpft, sondern es wird nach der Auftrittswahrscheinlichkeit von sogenannten lokalen binären Mustern gefragt, die sich aus 8 oder mehr Bildpunkten zusammensetzen. Die lokalen binären Muster weisen damit als Primitive eine höhere Komplexität auf als die Bildpunktpaare der 2-Punkt-Statistik. Steht über eine Textur Vorwissen zur Verfügung, ist daran zu denken, noch komplexere Primitive zu verwenden: Die Textur eines Baumes etwa lässt sich nicht nur auf der Ebene von Bildpunktpaaren oder lokalen binären Mustern modellieren, sondern auch durch das komplexe Primitiv „Blatt“.

4.1 Merkmale der 2-Punkt-Statistik

4.1.1 Grundlagen

Der Einsatz der 2-Punkt-Statistik in der Texturanalyse geht auf Haralick *et al.* [HSD73] zurück. In einem ersten Schritt wird die 2-Punkt-Statistik für einen gegebenen Bildausschnitt berechnet. Sie kann als Matrix C dargestellt werden, wobei das Matrixelement c_{ij} die Wahrscheinlichkeit angibt, beim Übergang von Bildpunkt u zu Bildpunkt $u+v$ eine Grauwerteänderung von i nach j anzutreffen. Der konstante Vektor v wird als Verschiebungsvektor bezeichnet, die Matrix C selbst als Cooccurrence-Matrix oder Abhängigkeitsmatrix. Ihre Größe $G \times G$ ist durch die Anzahl der Grauwertstufen G gegeben. In Abbildung 4.1 wird das Prinzip an einem einfachen Beispiel demonstriert.

In einem zweiten Schritt geht es darum, aus der Cooccurrence-Matrix geeignete Texturmerkmale abzuleiten. Prinzipiell können die Elemente c_{ij} als Texturmerkmale verwendet werden. Bei der üblichen Quantisierung der Grauwerte in $G = 256$ Stufen ergeben sich jedoch $256^2 = 65563$ Merkmale für einen einzigen Verschiebungsvektor. Sollen verschiedene Analyseabstände und -richtungen berücksichtigt werden, explodiert die Gesamtzahl der Merkmale. Deshalb ist es sinnvoll, die Information der Cooccurrence-Matrix in einige wenige Kennzahlen zu aggregieren. Typische Beispiele sind die Energie, der

Kontrast und die Korrelation [HSD73] [CH80] [Uns86]:

$$\begin{aligned}
&\text{Energie:} && f_0 = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} c_{ij}^2 \\
&\text{Kontrast:} && f_1 = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} (i-j)^2 c_{ij} \\
&\text{Korrelation:} && f_2 = \frac{1}{\sigma_x \sigma_y}\Big(\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} (i-\mu_x)(j-\mu_y) c_{ij}\Big)
\end{aligned} \tag{4.1}$$

Mit:

$$\begin{aligned}
\mu_x &= \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} i\ c_{ij} \\
\mu_y &= \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} j\ c_{ij} \\
\sigma_x^2 &= \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} (i-\mu_x)^2 c_{ij} \\
\sigma_y^2 &= \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} (j-\mu_y)^2 c_{ij}
\end{aligned} \tag{4.2}$$

Daneben finden sich in der genannten Literatur weitere Kenngrößen wie Varianz, Entropie und Homogenität.

In der Praxis kann sich der paradoxe Umstand ergeben, dass die Cooccurrence-Matrix ein größeres Datenvolumen aufweist als der Bildausschnitt, aus dem sie abgeleitet wurde. Die Folge ist eine niedrige Schätzgüte der Matrixelemente, die sich auch auf die abgeleiteten Kenngrößen übertragen kann. Aus diesem Grund wurde von Unser [Uns86] vorgeschlagen, die Cooccurrence-Matrix durch Wahrscheinlichkeitsfunktionen erster Ordnung entlang ihrer Hauptachsen zu ersetzen, d.h. durch das Summen- und Differenzhistogramm h_{sum} und h_{dif}. Die i-te Komponente $h_{\text{sum},i}$ ($h_{\text{dif},i}$) des Summenhistogramms (Differenzhistogramms) gibt die Wahrscheinlichkeit an, dass zwei durch einen Verschiebungsvektor v getrennte Bildpunkte in der Summe (Absolutdifferenz) den Grauwert i aufweisen. Ebenso wie bei der Cooccurrence-Matrix lassen sich aus den Summen- und Differenzhistogrammen Kennzahlen extrahieren. Für das Summenhistogramm gilt:

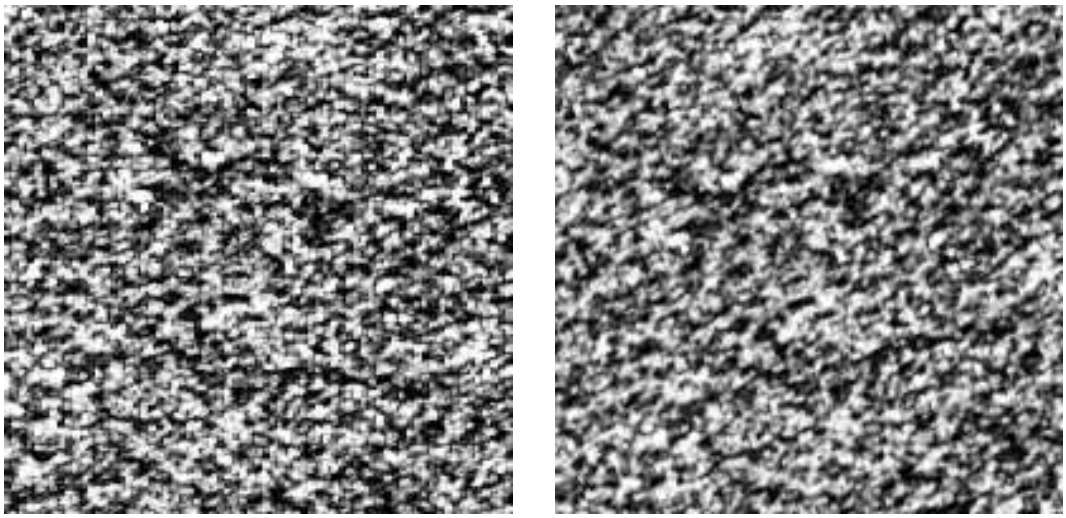
$$
\begin{aligned}
&\text{Mittelwert:} && f_0 = \mu_{\text{sum}} = \sum_i i \cdot h_{\text{sum},i} \\
&\text{Varianz:} && f_1 = \sum_i (i - \mu_{\text{sum}})^2 \cdot h_{\text{sum},i} \\
&\text{Kontrast:} && f_2 = \sum_i i^2 \cdot h_{\text{sum},i} \\
&\text{Homogenität:} && f_3 = \sum_i (1 + i)^{-2} \cdot h_{\text{sum},i} \\
&\text{Cluster Shade:} && f_4 = \sum_i (i - 2\mu_{\text{sum}})^3 \cdot h_{\text{sum},i} \\
&\text{Cluster Prominence:} && f_5 = \sum_i (i - 2\mu_{\text{sum}})^4 \cdot h_{\text{sum},i} \\
&\text{Energie:} && f_6 = \sum_i h_{\text{sum},i}^2 \\
&\text{Entropie:} && f_7 = \sum_i -h_{\text{sum},i} \cdot \log h_{\text{sum},i}
\end{aligned}
\tag{4.3}
$$

Die Kennzahlen für das Differenzhistogramm (f_8 bis f_{15}) werden analog berechnet. Im Weiteren werden die so definierten Merkmale unter dem Kürzel HP2 (histogrammbasierte Merkmale der 2-Punkt-Statistik) geführt. Ausgehend von diesen Merkmale werden sukzessive Verbesserungen eingeführt, um Schwachstellen zu beseitigen und die Leistungsfähigkeit zu steigern. Die vorgeschlagenen Maßnahmen zielen darauf ab, die Merkmalsberechnung robuster gegen Störeinflüsse zu machen, die Diskriminierungsfähigkeit der Merkmale zu steigern sowie Rotationsinvarianz herzustellen. In den nächsten Abschnitten werden die Verfahren vorgestellt, eine Diskussion der Vorschläge anhand von verschiedenen Datensätzen findet in Kapitel 8 statt.

4.1.2 Robuste Merkmalsextraktion

Um eine zuverlässige Texturklassifikation vornehmen zu können, stellen wir an die Texturmerkmale die Forderung, dass visuell als gleich empfundene Texturen zu äquivalenten Merkmalen führen sollen. Dass dies nicht zwangsläufig gilt, soll an drei Beispielen gezeigt werden.

Abtastung: Der visuelle Eindruck einer Textur spiegelt sich nur dann korrekt im digitalen Bild wider, wenn dieses mit hinreichend großer Frequenz abgetastet wurde. Gemäß dem Abtasttheorem ist die Abtastfrequenz mindestens doppelt so hoch zu wählen wie die maximal darzustellende Frequenz. Im anderen Fall treten Moiré-Effekte auf [Jäh97b].



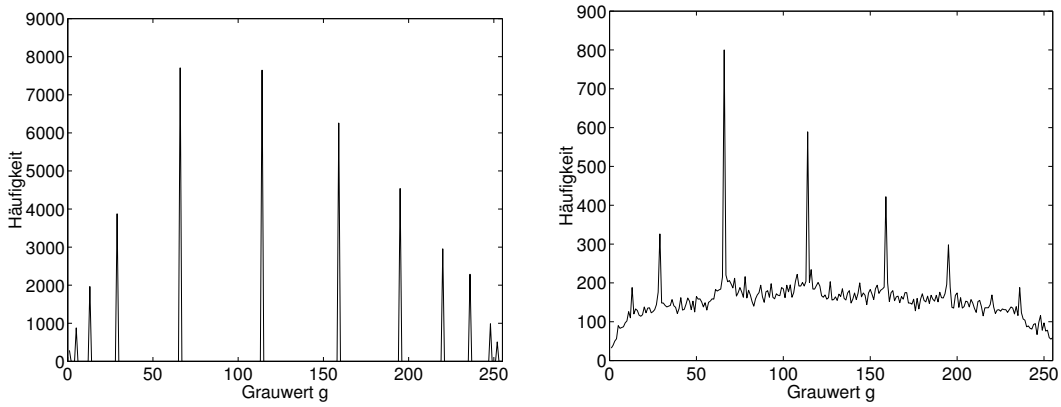


Abbildung 4.2: Die oben rechts dargestellte Textur geht durch Drehung um 20 ° gegen den Uhrzeigersinn aus der Textur oben links hervor. Obwohl der visuelle Eindruck identisch ist, unterscheiden sich die zugehörigen Histogramme signifikant. Ursache ist die (bilineare) Interpolation von Bildpunkten bei der Bilddrehung.

Digitale Vorverarbeitung: Durch digitale Vorverarbeitung können Effekte auftreten, die die Korrespondenz zwischen visuellem Eindruck und algorithmischer Bildbeschreibung aufheben. Ein Beispiel ist in Abbildung 4.2 zu sehen. Während die Drehung des Bildes den visuellen Eindruck nicht beeinflusst, verändert sich das Histogramm signifikant. Ursache für die diskreten Spitzen des linken Histogramms war eine vorausgegangene Histogrammebnung [GW92].

Fenstergröße: Die Robustheit der Merkmalsschätzung hängt von der Größe des Analysefensters ab. Um etwa Histogramme zuverlässig zu schätzen, wird in der Literatur gefordert [OVOP01], dass die durchschnittliche Anzahl von Einträgen pro Quantisierungstufe zwischen 5 und 10 liegen sollte. Bei 256 Quantisierungsstufen ergeben sich daraus minimale Fenstergrößen von $F = 35 \times 35$ bzw. $F = 50 \times 50$.

Um diesem Auseinanderdriften von visuellem Eindruck und berechneten Merkmalen entgegenzuwirken, bieten sich folgende Maßnahmen an:

Bildglättung: Abtasteffekte (Moiré-Effekte) lassen sich vermeiden, wenn bei der Signalaufnahme Frequenzen oberhalb der Grenzfrequenz durch Tiefpassfilterung eliminiert werden [Jäh97b]. Ebenso kann eine vorherige Bildglättung verhindern, dass durch Interpolation bzw. Bilddrehung der visuelle Eindruck verändert wird.

Histogrammebnung: Ein in der Literatur häufig genanntes Verfahren ist die Histogrammebnung [HSD73]. Die zugrunde liegende Grauwerttransformation hat zum Ziel, das Integral über dem Histogramm zu linearisieren und unabhängig von Beleuchtungsbedingungen zu werden. Die Histogrammebnung führt folglich dazu, dass die Auflösung der Grauwertquantisierung an die Häufigkeitsverteilung angepasst [POX00] wird. Problematisch für die weitere Verarbeitung ist, dass durch die Histogrammebnung „Lücken“ im Histogramm entstehen können.

Lineare Merkmale: Bei der Quantisierung der Grauwerte besteht ein Dilemma zwischen Auflösungsvermögen und Schätzgüte. Eine hohe Grauwertauflösung führt zu schwach gefüllten Histogrammen und damit zu einer niedrigen Schätzgüte und umgekehrt. Werden jedoch nicht die einzelnen Quantisierungsstufen des Histogramms als Merkmale betrachtet, sondern die Merkmale als Funktional des Histogramms gebildet, kann das Problem umgangen werden. Hängt das Funktional linear von dem Histogramm ab, wie die Größen f_0 bis f_5 in Gleichung 4.3, liefert die maximale Quantisierung die höchste Genauigkeit für die zu berechnenden Merkmale.

Histogrammglättung: Für Merkmale, die nicht als lineares Funktional des Histogramms gebildet werden, bietet sich eine Glättung des Histogramms an. Wird der Faltungskern zur Histogrammglättung als Gaussfunktion gewählt, wird den Zielen „Auflösungsvermögen“ und „Schätzgüte“ besser entsprochen, als eine bloße Reduktion der Quantisierungsstufen ermöglichen würde, die einem rechteckigen Faltungskern entspräche.

In der Praxis hat sich die Kombination der Ansätze „Lineare Merkmale“ und „Histogrammglättung“ als sinnvoll erwiesen. Die linearen Merkmale werden auf den Summen- und Differenzhistogrammen mit maximaler Quantisierung berechnet. Für Bilder mit $G = 256$ Graustufen ergeben sich 511 Quantisierungsstufen für das Summenhistogramm und 256 für das Differenzhistogramm. Vor Berechnung der nichtlinearen Merkmale werden die Histogramme mit dem Filterkern $(0{,}25; 0{,}5; 0{,}25)$ geglättet.

4.1.3 Multiskalenansatz

Wesentlich für die Merkmale HP2 ist die Betrachtung eines Bildpunktpaares, das durch den Verschiebungsvektor v separiert ist. Der Betrag D des Verschiebungsvektors v bestimmt, über welche Abstände die Korrelation des Bildpunktpaares untersucht wird. Um eine möglichst umfassende Beschreibung der Texturen zu erhalten, genügt es nicht, nur die Korrelation von Nachbarpunkten ($D = 1$) zu betrachten. Vielmehr müssen die Summen- und Differenzhistogramme für einen weiten Bereich von D ausgewertet werden. Die obere Grenze für D lässt sich aus der Autokorrelationsfunktion abgeschätzen. Sie ist gegeben durch den Wert, bei dem die normierten Autokorrelationsfunktion unter eine gegebene Schwelle fällt.

Die erste Form der Multiskalenanalyse – auch als Multi-Predicate Ansatz bezeichnet – bedient sich also eines Satzes von Verschiebungsvektoren v_i mit unterschiedlichem Betrag D_i. Der Vorteil dieses Ansatzes liegt im punktgenauen Vergleich zweier Grauwerte. Insbesondere bei feinen Strukturen mit großer Korrelationslänge können so wertvolle Information über die Textur gewonnen werden. Eine zweite Form der Multiskalenanalyse basiert auf Tiefpassfilterung und Unterabtastung der Textur. Begonnen wird die Analyse auf der originalen Textur mit einem Satz von Verschiebungsvektoren v_i, typischerweise der Länge $D_i = 1$. Sind die Histogramme für diese Auflösungsstufe (SUB0) berechnet, wird die Textur um den Faktor 2 skaliert. Im einfachsten Fall werden die Grauwerte von 4 benachbarten Bildpunkten gemittelt und zu einem neuen Bildpunkt zusammengefasst. Auf der so verkleinerte Textur (SUB1) werden die Histogramme erneut berechnet. Im Gegensatz zum ersten Ansatz geben die Histogramme höherer Stufe hier nicht Auskunft über die Korrelation von einzelnen Bildpunkten, sondern über die Korrelation von zunehmend größer werdenden Bildbereichen. Dabei besteht die Gefahr, dass feinen Struktu-

ren nicht ausreichend Rechnung getragen wird. Ein weiterer Nachteil besteht im Auftreten von Diskretisierungseffekten, die sich in einer verminderten Rotationsinvarianz der Merkmale HP2-ROT äußern. Jedoch bietet der Ansatz mit Unterabtastung die Möglichkeit zur skalierungsinvarianten Texturanalyse [PTCH97]. Selbstverständlich können Merkmale beider Ansätze parallel verwendet werden, um ihre Stärken zu kombinieren. Vergleiche dazu Experiment G in Abschnitt 8.1.

4.1.4 Rotationsinvarianz

Im letzten Abschnitt wurde untersucht, wie durch Längenvariation des Verschiebungsvektors v die Textur auf verschiedenen *Skalen* zu erfassen ist. Nun wenden wir uns der Analyse der Textur in verschiedenen *Richtungen* zu, indem wir die Orientierung von v variieren. Ziel ist die umfassende Erfassung und Charakterisierung von anisotropen Texturen. Die Berechnung der Histogramme erfolgt jetzt nicht nur für einen Satz von K Skalierungsstufen, sondern auch für R Analyserichtungen, insgesamt also für $K \cdot R$ Summen- und Differenzhistogramme. Da die Summen- und (absoluten) Differenzhistogramme für die Orientierungen φ und $\varphi + \pi$ identisch sind, ergibt sich für R Analyserichtungen eine Winkeldifferenz zwischen den Analyserichtungen von $\Delta\varphi = 2\pi/(2R)$. Bei der Berechnung der Histogramme muss berücksichtigt werden, dass der Verschiebungsvektor v im Allgemeinen nicht exakt von Bildpunkt zu Bildpunkt reicht, sondern an einem Zwischengitterplatz endet. Der entsprechende Grauwert wird durch bilineare Interpolation gewonnen.

Für Anwendungen, bei denen rotierte (aber ansonsten identische) Texturen unterschieden werden sollen, können die Texturmerkmale aus diesen Histogrammen direkt in einen Klassifikator gefüttert werden. In der Praxis besteht jedoch meistens der Wunsch, Texturen unabhängig von ihrer Rotationslage zu klassifizieren. Daher fasst man die Merkmale, die für unterschiedliche Rotationslagen berechnet wurden, zu rotationsinvarianten Merkmalen zusammen. Dies geschieht etwa durch die Bildung von Mittelwert und Varianz über die R Analyserichtungen für jedes Merkmal vom Typ f_0 bis f_{15} aus Gleichung 4.3.

In Experimenten zeigt sich, dass die Varianz über die richtungsabhängigen Merkmale ein aussagekräftiges, jedoch weniger robustes Merkmal ist. Insbesondere reagiert sie weitaus sensibler auf Änderungen der Rotationslage der Textur als die entsprechenden Mittelwerte. Ein guter Kompromiss ergibt sich, wenn auf die Berechnung der Varianz für diejenigen Merkmale f_i verzichtet wird, die bereits von einem geschätzten Wert (f_0) abhängen. Gemäß Gleichung 4.3 sind davon die Merkmale f_1 (Varianz), f_4 (Cluster Shade) und f_5 (Cluster Prominence) betroffen.

Um echte Rotationsinvarianz zu erreichen, muss die Winkeldifferenz $\Delta\varphi$ zwischen Analyserichtungen kleiner sein als die inverse Winkelauflösung ρ^{-1}

des rotationsabhängigen Texturoperators O_φ. Für die Merkmale der 2-Punkt-Statistik ergibt sich ρ^{-1} zu $1/D$, was der inversen Winkelauflösung eines Pixels im Abstand D entspricht. Somit gilt:

$$\Delta\varphi \leq \rho^{-1} \quad \Longleftrightarrow \quad \pi/R \leq 1/D \tag{4.4}$$

Für den Pixelpaarabstand $D = 1$ ist Gleichung 4.4 bereits erfüllt, wenn 2 Analyserichtungen verwendet werden ($R = 2$: $\varphi = 0\,°$, $\varphi = 90\,°$). Für steigende Werte von D muss jedoch die Anzahl der Analyserichtungen R erhöht werden, so dass die Berechnung der Merkmale zunehmend aufwendiger wird. Für $D = 5$ etwa wären mindestens $R = 15$ Analyserichtungen für Merkmale des Typs HP2 notwendig. Im nächsten Abschnitt wird eine Methodik vorgeschlagen, die auch für eine reduzierte Anzahl von Analyserichtungen Rotationsinvarianz garantiert.

4.1.5 Randomisierung

Um Gleichung 4.3 zu erfüllen, muss die Winkeldifferenz $\Delta\varphi$ zwischen Analyserichtungen kleiner sein als die inverse Winkelauflösung ρ^{-1} des rotationsabhängigen Texturoperators O_φ. Anstatt dies über eine große Zahl von Analyserichtungen R zu erreichen, wird vorgeschlagen, die inverse Winkelauflösung des Texturoperator O_φ zu erhöhen.

Rekapitulieren wir das bisherige Vorgehen: Für einen Verschiebungsvektor v mit Betrag D und Orientierung φ werden die Histogramme für Grauwertsumme und -differenz über einem Bildausschnitt bestimmt. Die Pixelpaare zu den Z Histogrammeinträgen sind alle identisch orientiert, so dass die inverse Winkelauflösung ρ^{-1} dieses Texturoperators $1/D$ beträgt.

Um die inverse Winkelauflösung ρ^{-1} zu erhöhen, wird die Randomisierung der Orientierung des Verschiebungsvektors v vorgeschlagen. Anstatt alle Bildpunktpaare exakt im Winkel φ auszurichten, wird eine Winkelverteilung für φ vorgegeben, beispielsweise Gleichverteilung im Intervall $[\varphi - \Delta\varphi/2, \varphi + \Delta\varphi/2]$, mit $\Delta\varphi = 2\pi/(2R)$. Dadurch erhöht sich die inverse Winkelauflösung ρ^{-1} auf $\Delta\varphi$, so dass Gleichung 4.3 stets erfüllt ist. Für eine effiziente Implementierung werden die Anfangs- und Endpositionen für die Z Zufallsrealisierungen des Verschiebungsvektors nur einmal bestimmt und dann auf alle Analysefenster angewendet. Die mit dieser Methodik erzielten Verbesserungen sind in Abschnitt 8.1.3 dargestellt.

4.1.6 Mehrkanalbilder

Liegen die Texturen nicht als Grauwertbilder vor, sondern als Mehrkanalbilder, sollte diese zusätzliche Information für die Klassifikation ausgenutzt werden. Der hier verfolgte Ansatz sieht vor, für jeden der C Bildkanäle die Merkmale

Beispiel

6	5	2
7	6	1
9	8	7

Binärbild

1	0	0
1		0
1	1	1

Gewichte

1	2	4
128		8
64	**32**	**16**

$$\text{LBP} = (10001111)_2 = 1 + 16 + 32 + 64 + 128 = 241$$

Abbildung 4.3: Berechnung des lokalen binären Musters LBP für eine 3×3 Umgebung durch die Schritte „Binarisierung“ und „Gewichtung“.

separat zu berechnen. Damit vergrößert sich die Gesamtzahl der Texturmerkmale um den Faktor C. Für den Fall, dass eine hohe Korrelation zwischen den Bildkanälen besteht, ist der Informationsgewinn gering. Werden Verfahren der Merkmalsselektion angewendet, muss sichergestellt sein, dass das Gütekriterium Korrelationen zwischen Merkmalen erfasst und stark korrelierende Merkmale nicht gemeinsam ausgewählt werden. Insbesondere scheiden alle Gütekriterien aus, die auf der Betrachtung von Einzelmerkmalen beruhen.

4.2 Lokale binäre Muster

4.2.1 Grundlagen

Das Konzept der lokalen binären Muster (LBP) geht auf Ojala *et al.* [OPH96] zurück und ist in Abbildung 4.3 an einem Beispiel dargestellt. Die 3×3-Umgebung eines Bildpunkts (Bild links) wird mit dessen Grauwert binarisiert (mittleres Bild). Schließlich wird das Binärbild elementweise mit einer konstanten Gewichtsmatrix (rechtes Bild) multipliziert. Das Ergebnis entspricht dem lokalen binären Muster. Eine interessante Eigenschaft der LBP besteht darin, dass Grauwertemuster von 9 Bildpunkten durch einen von lediglich 256 diskreten Zuständen zu beschreiben. Dies wird durch die Beschränkung auf den binären Vergleich von Bildpunkten ermöglicht. Ebenfalls eine Folge des binären Vergleichs ist die Invarianz der LBP gegenüber beliebigen monotonen Grauwerttransformationen.

Soll auf Basis der LBP ein Texturausschnitt charakterisiert werden, berechnet man die Häufigkeitsverteilung der 256 verschiedenen Muster. Anders als bei den Summen- und Differenzhistogrammen der Merkmale HP2 hat die Reihenfolge der Histogrammeinträge hier keine unmittelbare physikalische Bedeutung. Der Index des Histogramms entspricht einer bloßen Durchnummerierung der LBP. Deshalb ist es im Unterschied zu den Merkmalen HP2 nicht sinnvoll, Merkmale der Form $f_0 - f_{15}$ aus dem Histogramm abzuleiten. Vielmehr wird die Häufigkeit eines jeden Musters direkt als Texturmerkmal verwendet.

Nummer des LBP	Binärmuster des LBP	Binärmuster des LBP-ROT	Nummer des LBP-ROT
1	0000	0000	1
2	0001	0001	2
3	0010	0001	2
4	0011	0011	3
5	0100	0001	2
6	0101	0101	4
7	0110	0011	3
8	0111	0111	5
9	1000	0001	2
10	1001	0011	3
11	1010	0101	4
12	1011	0111	5
13	1100	0011	3
14	1101	0111	5
15	1110	0111	5
16	1111	1111	6

Tabelle 4.1: Die 16 LBP, die sich aus dem binären Vergleich eines Bildpunktes mit 4 Nachbarn ergeben, werden durch zyklische Vertauschung auf 6 rotationsinvariante Muster LBP-ROT abgebildet.

Zahl der Nachbarn B	Winkelinkrement $\Delta\varphi$	Zahl der LBP	Zahl der LBP-ROT	Zahl der spiegelsymmetrischen LBP-ROT
4	90,0 °	16	6	6
8	45,0 °	256	36	30
12	30,0 °	4096	352	224
16	22,5 °	65536	4116	2250

Tabelle 4.2: Die Zahl der LBP nimmt exponentiell mit der Zahl der Nachbarn zu. Werden diejenigen LBP, die durch Operationen wie Rotation und Spiegelung auseinander hervorgehen, als identisch angesehen, reduziert sich die Zahl der Muster.

4.2.2 Multiskalenansatz

Ebenso wie bei den Merkmalen HP2 besteht hier der Wunsch, weiter auseinanderliegende Bildpunkte auf ihre Korrelation zu untersuchen. Die in Abschnitt 4.1.3 vorgestellten Verfahren können eins zu eins auf die Merkmale vom Typ LBP übertragen werden. Beim Multi-Predicate Ansatz bezeichnet die Größe D jetzt den Abstand zwischen zentralem Bildpunkt und den kreisförmig angeordneten Nachbarn. Sofern die Nachbarn nicht exakt auf Gitterplätzen liegen, werden deren Grauwerte durch bilineare Interpolation gewonnen.

4.2.3 Rotations- und Spiegelinvarianz

Der Vergleich des zentralen Bildpunkts mit seinen B Nachbarn ergibt 2^B mögliche LBP, die als binäre Zeichenfolge der Länge B darstellbar sind. Eine Rotation der Textur und damit der Nachbarbildpunkte entspricht einer zyklischen

Vertauschung dieser binären Zeichenfolge. Die LBP sind folglich nicht rotationsinvariant. Rotationsinvarianz kann erzeugt werden, indem alle LBP, die durch zyklische Vertauschung auseinander hervorgehen, zu einem rotationsinvarianten Muster zusammengefasst werden. Der reduzierte Satz rotationsinvarianter Muster wird mit LBP-ROT [POX00] bezeichnet. In Tabelle 4.1 sind die LBP und LBP-ROT für den Fall $B = 4$ dargestellt.

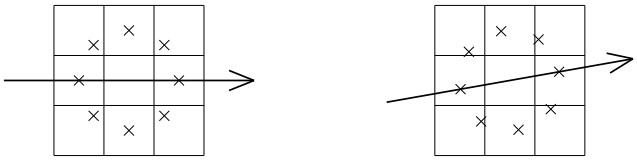
Streng genommen bezieht sich die Bezeichnung „rotationsinvariant“ bisher nur auf die mathematische Operation der zyklischen Vertauschung eines zuvor extrahierten Satzes von B Nachbargrauwerten. Die Muster LBP-ROT sind also nur invariant gegenüber Drehungen um diskrete Winkel $\Delta\varphi$, die einem Vielfachen von $360\,°/B$ entsprechen. Um tatsächlich eine rotationsunabhängige Statistik beliebig orientierter Texturen zu erhalten, ist eine hinreichend große Anzahl an Nachbarbildpunkten B notwendig. Vor allem für größere Bildpunktabstände D reichen $B = 8$ Nachbarn nicht aus. Höhere Werte für B mit kleineren Winkelinkrementen $\Delta\varphi$ führen jedoch zu einer exponentiell anwachsenden Zahl von LBP bzw. LBP-ROT, so dass die eine stabile Schätzung der Häufigkeitsverteilung unmöglich wird, vgl. Tabelle 4.2.

Um gute Rotationsinvarianz zu erhalten, ist eine weitere Reduktion der Musterzahl vordringlich. Dazu wird vorgeschlagen, Symmetrien in den Mustern auszunutzen, die für den visuellen Eindruck unerheblich sind. Neben der bereits berücksichtigten Rotatationssymmetrie kommt die Spiegelsymmetrie in Frage. Analog dem Vorgehen bei der Rotationsinvarianz, werden jetzt alle Muster als identisch angesehen, die durch Spiegelung auseinander hervorgehen. In Tabelle 4.2 sind die sich ergebenden Musterzahlen wiedergegeben.

Gehen wir von Analysefenstern der Größe $F = 32 \times 32$ aus und berechnen für jeden inneren Bildpunkt ein Muster, stehen 900 Histogrammeinträge zur Verfügung. Fordern wir weiterhin 5 Einträge pro Quantisierungsstufe, so darf die Zahl der Muster nicht wesentlich über 180 liegen. Als praxistauglich kommen damit die Varianten LBP-ROT mit $B = 8$ und 36 Mustern sowie mit $B = 12$ und 224 Mustern in Frage.

4.2.4 Randomisierung

Analog zu den Merkmalen HP2 kann bei den Merkmalen LBP die Ausrichtung der Muster randomisiert werden, beispielsweise durch Gleichverteilung im Intervall $[-\Delta\varphi/2, \Delta\varphi/2]$, mit $\Delta\varphi = 2\pi/B$. Abbildung 4.4 zeigt die Ausrichtung der Nachbarbildpunkte um den zufällig ermittelten Winkelversatz von $10\,°$. Um die Histogrammschätzung weiter zu verbessern, ist daran zu denken, pro Bildpunkt des Analysefensters mehr als ein lokales binäres Muster zu extrahieren. Aufgrund der zufälligen Ausrichtung φ werden die LBP im Allgemeinen verschiedene Werte annehmen.



$\varphi = 0\,°$ $\varphi = 10\,°$

Abbildung 4.4: Die Randomisierung der Orientierung des Texturoperators O_φ wird am Beispiel der Merkmale LBP/LBPROT mit $R = 8$ gezeigt. Der zufällig ermittelte Winkelversatz beträgt $10\,°$.

4.2.5 Mehrkanalbilder

Das Vorgehen erfolgt analog zu den Merkmalen HP2, vgl. Abschnitt 4.1.6.

4.2.6 Klassifikation

Für die Klassifikation von Merkmalen des Typs LBP bieten sich Verfahren an, die den Abstand zwischen zwei Häufigkeitsverteilungen messen. Dazu zählt neben dem Kullback-Leibler Abstand (vgl. Abschnitt 6.2.1 und [HOP+95]) die Kreuzentropie d_{Log} [MPO00], die eng miteinander verwandt sind:

$$d_{\text{Log}} = -\sum_{i=1}^{M} x_i \ln r_i \tag{4.5}$$

x_i und r_i bezeichnen die Häufigkeitsverteilungen im Analysefenster bzw. in der Referenztextur, M entspricht der Anzahl der Quantisierungsstufen. Ebenso wie der Kullback-Leibler Abstand wird die Kreuzentropie minimal, wenn beide Verteilungen identisch sind.

Alternativ zu diesen logarithmischen Wahrscheinlichkeitsmaßen kann der euklidische Abstand verwendet werden:

$$d_{\text{Euklid}} = \sum_{i=1}^{M} (x_i - r_i)^2 \tag{4.6}$$

Welches Abstandsmaß in der Praxis zu verwenden ist, hängt u.a. von der Robustheit gegenüber Schätzfehlern in der Häufigkeitsverteilung ab. Diese Eigenschaft wird unter dem Stichwort der „Impliziten Merkmalsgewichtung“ in Abschnitt 5.4 diskutiert.

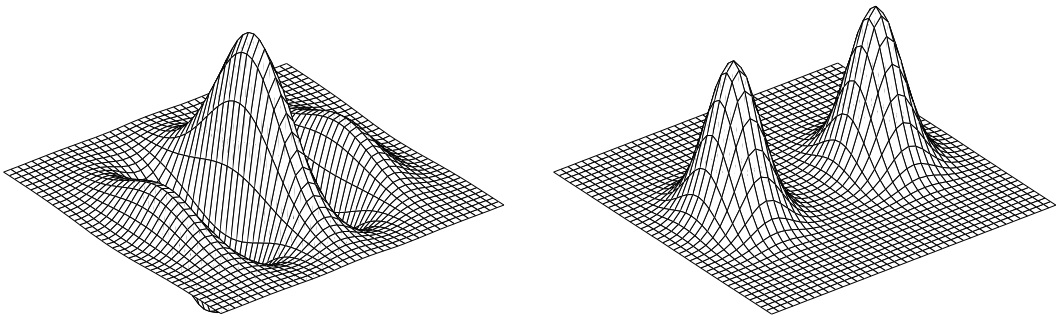


Abbildung 4.5: Darstellung eines geraden Gaborfilters im Ortsraum (links) und Frequenzraum (rechts) .

4.3 Gaborfilter

4.3.1 Grundlagen

Im Gegensatz zu den meisten Texturmerkmalen liegt den Gaborfiltern eine biologische Motivation zu Grunde. In Experimenten konnte gezeigt werden, dass die zweidimensionalen Feldprofile von Neuronen im visuellen Kortex Gaborfiltern entsprechen [Dau88]. Im Ortsraum haben sie die Form von gaußförmig modulierten ebenen Wellen. Der Realteil (gerader Filter) [MP97] [JF91] [CJ00] lautet:

$$h(x,y) = \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right\}\cos(2\pi u_0 x) \tag{4.7}$$

u_0 gibt die Frequenz der ebenen Welle an, σ_x und σ_y beschreiben die Ausdehnung der Einhüllenden in longitudinaler und transversaler Richtung. Im Frequenzraum ist der Realteil des Filters durch zwei symmetrisch liegende Gaußglocken gegeben:

$$\begin{aligned} H(u,v) = \; & A \; \left(\exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\}\right. \\ & + \; \left.\exp\left\{-\frac{1}{2}\left[\frac{(u+u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\}\right) \end{aligned} \tag{4.8}$$

Mit $A = 2\pi\sigma_x\sigma_y$, $\sigma_u = 1/2\pi\sigma_x$ und $\sigma_v = 1/2\pi\sigma_y$. Abbildung 4.5 zeigt einen geraden Gaborfilter im Orts- und Frequenzraum. Durch die Beziehung $\sigma_x \sim 1/\sigma_u$ ist eine Unschärferelation zwischen den Auflösungen im Orts- und Frequenzraum gegeben. Sie besagt, dass eine Erhöhung der Ortsauflösung mit einer Verminderung der Frequenzauflösung einhergeht und umgekehrt. Gaborfilter besitzen die Eigenschaft, dass die gemeinsame Auflösung im Orts-/Orts-Frequenzraum maximal ist [Dau85].

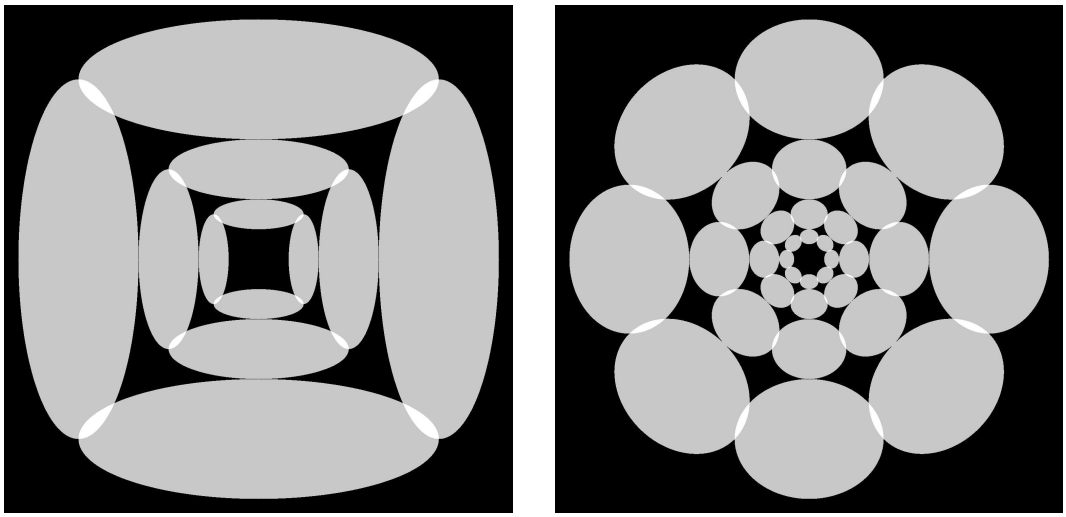


Abbildung 4.6: Polar-logarithmische Gaborfilterbank, dargestellt im Frequenzraum. Links mit 3 Zentralfrequenzen und 2 Analyserichtungen, rechts mit 4 Zentralfrequenzen und 4 Analyserichtungen.

Die Ableitung von Texturmerkmalen aus Gaborfiltern beruht auf der Schätzung der lokalen Energiefunktion [CJ00] [RH99]. Als Nichtlinearität wurde in dieser Arbeit die Quadratsumme von gerader und ungerader Filterantwort gewählt. Um stabilere Merkmale zu erhalten, werden die Ergebnisse häufig räumlich gemittelt. Neben diesem Mittelwert kann auch die Varianz oder Standardabweichung aus einer lokalen Umgebung als Texturmerkmal verwendet werden [CC96].

4.3.2 Optimierung der Filterbank

Um eine umfassende Texturbeschreibung zu erhalten, wird ein Satz von Gaborfiltern zusammengestellt. Eine Möglichkeit ist die polar-logarithmische oder dyadische Anordnung der Filter im Frequenzraum. In Abbildung 4.6 sind zwei Beispiele zu sehen: Die linke Filterbank setzt sich aus 6 Filtern für 3 Zentralfrequenzen und 2 Analyserichtungen zusammen, während die rechte Filterbank 16 Filter für 4 Zentralfrequenzen und 4 Analyserichtungen umfasst. Jeder Filter ist durch ein symmetrisches Paar von Gaußglocken gegeben, die bis zur Halbwertsbreite markiert sind. Der Frequenzursprung ist in der Bildmitte.

Um Texturen zu unterscheiden, ist es nicht notwendig, den gesamten Frequenzraum lückenlos abzudecken. Vielmehr kommt es darauf an, die Bereiche ausfindig zu machen, in denen die Texturen unterschiedliche Energien aufweisen. Ein Ansatz besteht darin, aus einer gegebenen Filterbank die relevanten Elemente mit Methoden der Merkmalsselektion zu bestimmen. Als Alternative bietet sich die Merkmalsoptimierung an. Hier werden die Parameter einer

kleinen Filterbank optimiert. Im Fall der Gaborfilter sind dies für jeden Filter 4 Größen, nämlich die Zentralfrequenz u_0, die Filterbreiten σ_x und σ_y und die Orientierung θ. In den Abschnitten 7 und 8.3 werden Gaborfilterbänke mit Evolutionsstrategien an spezifische Klassifikationsaufgaben adaptiert.

Kapitel 5

Aspekte der Klassifikation

Während das letzte Kapitel der Berechnung von Texturmerkmalen gewidmet war, gehen wir nun einen Schritt weiter entlang der Verarbeitungskette in Richtung Klassifikation. Im Vordergrund stehen zwei Aspekte: Zum einen die Robustheit der Klassifikation, zum anderen die Wechselwirkung zwischen extrahierten Merkmalen und Klassifikator.

Im Hinblick auf die Robustheit der Klassifikation sollen die Einflüsse von endlichen Lernstichproben (5.1) diskutiert und ein Ähnlichkeitsmaß zwischen Merkmalsvektor und Texturklasse vorgeschlagen werden. Schließlich soll die Wechselwirkung zwischen extrahierten Merkmalen und Klassifikator unter den Blickwinkeln „Rotationsinvarianz“ (5.3), „Implizite Merkmalsgewichtung“ (5.4) und „Visuelle Ähnlichkeit“ (5.2) untersucht werden.

5.1 Kleine Lernstichproben

Bevor wir uns mit den Auswirkungen endlicher Lernstichproben auseinandersetzen, sollen die zur Adaption verwendeten Datensätze „Training“, „Test“ und „Validierung“ gegeneinander abgegrenzt werden. Allen drei Datensätzen ist gemeinsam, dass sie Merkmalsvektoren x und deren Bewertungen y miteinander verknüpfen und damit Mengen von (x, y)-Paaren darstellen. Durch sie ist exemplarisch die Abbildung $\kappa : x \rightarrow y$ gegeben, die es bei der Adaption des Klassifikators zu erlernen gilt.

Trainingsmenge: Mit ihr werden die freien Parameter des Klassifikators bestimmt, beispielsweise die Kovarianzmatrix beim Normalverteilungsklassifikator oder die Verbindungsgewichte eines neuronalen Netzes.

Testmenge: Sie dient als Abbruchkriterium für das iterative Klassifikatortraining, etwa bei neuronalen Netzen. Außerdem wird sie herangezogen, um die beste Systemkonfiguration (Merkmalsuntermenge, Merkmalsparameterisierung) auszuwählen.

Validierungsmenge: Als unabhängiger Datensatz dient sie dazu, das fertig adaptierte System zu validieren.

Im theoretischen Grenzfall unendlich großer Lernstichproben fallen die Fehlerraten auf Trainings-, Test- und Validierungsmenge zusammen. In der Praxis führen endliche Lernstichproben zu deutlichen Unterschieden in den Fehlerraten. Um die Ursachen für die Diskrepanzen in den Fehlerraten herauszuarbeiten, soll folgendes Prozessmodell verwendet werden: Jede Texturklasse ω_i wird als ein mustererzeugender Prozess aufgefasst, der Mustervektoren x erzeugt. Die Klassifikation entspricht dem umgekehrten Vorgang, nämlich von den Mustervektoren x auf die Texturklassen ω_i zurückzuschließen.

In der Trainingsphase werden die mustererzeugenden Prozesse modelliert und statistisch beschrieben. Das Training entspricht einer Schätzung der Modellparameter für jede Klasse. In der Klassifikationsphase wird analysiert, mit welchem Modellparametersatz der vorliegende Vektor x am besten darstellt werden kann.

5.1.1 Schätzfehler

Die Qualität der Modellparameterschätzung in der Trainingsphase hängt von vielen Einflussgrößen ab, beispielsweise von der wahren Statistik der Texturklassen, der Anzahl der Trainingsvektoren, der Anzahl der Merkmale und der Anzahl der Modellparameter. Je nach Blickwinkel werden die Schätzfehler einem dieser Einflussfaktoren zugeordnet. So spricht man im Fall zu weniger Trainingsvektoren vom „Finite Sample Size Effect“, im Fall zu vieler Merkmale vom „Curse of Dimensionality“ und im Fall zu vieler Modellparameter vom „Overfitting“ [JZ97] [Tru79] [Rau97].

Beim Entwurf eines Klassifikationssystems besteht die Kunst darin, die Komplexität des Prozessmodells so zu wählen, dass ein Optimum zwischen den Polen „Hohe Aussagekraft des Prozessmodells“ und „Niedriger Schätzfehler der Modellparameter“ erreicht wird. Als Optimierungskriterium dient die Fehlerrate auf dem unabhängigen Validierungsdatensatz. Ein Beispiel soll dies veranschaulichen: Die Verteilung einer Klasse wird entweder durch Statistik erster Ordnung (Schwerpunkt μ) oder durch Statitik zweiter Ordnung (Schwerpunkt μ und Kovarianzmatrix Σ) beschrieben. Im ersten Fall ist das Prozessmodell einfach und die Schätzgüte der Modellparameter μ hoch während im zweiten Fall das Prozessmodell aussagekräftiger und die Schätzgüte von Σ geringer ist. Dieses Optimieren der Komplexität des Prozessmodells kann von 2 Seiten angegangen werden. Der erste Ansatz besteht darin, die Anzahl der inneren Freiheitsgrade des Prozessmodells einzustellen, während der zweite Ansatz die Anzahl bzw. Auswahl der Eingangsgrößen des Prozessmodells optimiert.

Aus der Literatur bekannte Beispiele für den ersten Ansatz sind die Minimierung von neuronalen Netzen [Zel94] oder die gezielte Vereinfachung der

Kovarianzmatrix [HL96] [HFT96]. In der vorliegenden Arbeit jedoch wird der zweite Ansatz mit Hilfe von Merkmalsselektion und -optimierung verfolgt (vgl. Kapitel 6, 7 und 8).

5.1.2 Instationarität

Als weitere Ursache für Diskrepanzen zwischen den Fehlerraten kommen Instationaritäten in den mustererzeugenden Prozessen in Frage [Hän01]. Im Unterschied zu den im letzten Abschnitt besprochenen Schätzfehlern liegt hier das Problem nicht in der endlichen Größe der Lernstichprobe, sondern in einem Prozess, der Veränderungen unterworfen ist. Anders ausgedrückt: Die Lernstichprobe erlaubt eine zuverlässige Schätzung der Modellparameter, jedoch unterscheiden sich diese für die verschiedenen Datensätze aufgrund der Instationarität der Prozesse.

In der Praxis offenbart sich dieser Umstand häufig in Form mangelnder Robustheit des Klassifikators. Leichte Variationen der zu klassifizierenden Muster werden nicht mehr richtig erkannt. Ein Beispiel dafür ist in Abschnitt 8.2.3 zu finden. Als Ausweg aus diesem Problem bietet sich eine veränderte Zusammensetzung der Lernstichprobe an - nicht im Sinne einer Vergrößerung der Stichprobe sondern im Sinne einer repräsentativeren Zusammensetzung. Ist die Instationarität des Prozesses nicht zyklisch, sondern liegt eine gerichtete Drift vor, sind die erzielbaren Verbesserungen beschränkt. Ein weiterer Ansatz zur Erhöhung der Robustheit kann in einem vereinfachten Prozessmodell bestehen, so dass die Modellparameter von Lern- und Validierungsphase besser übereinstimmen.

Zum Abschluss sei auf eine Wechselwirkung zwischen instationären Prozessen und Merkmalsselektion hingewiesen. Ohne Merkmalsselektion, d.h. unter Verwendung aller verfügbaren erfüllen die Merkmale den Zweck, jede Klasse für sich zu beschreiben. Durch die Selektion der effektivsten Merkmale verändert sich die Zielsetzung. Anstelle der (absoluten) *Beschreibung* der Klassen steht nun die (relative) *Abgrenzung* der Klassen im Vordergrund. Änderungen der Prozessparameter aufgrund von Instationaritäten können sich bei Verwendung eines reduzierten Merkmalssatzes stärker auswirken und die Fehlerrate ansteigen lassen. Ein einfaches Beispiel soll dies verdeutlichen: Zitronen und Orangen seien durch Form und Farbe charakterisiert. Die Farbe als effektivstes Merkmal erlaubt eine sichere Unterscheidung. Liegen die Objekte jedoch nur noch in schwarz-weiß Darstellung vor (Instationarität des mustererzeugenden Prozesses) ist anhand der Farbe keine Unterscheidung mehr möglich. Die vollständige Beschreibung (Form und Farbe) lässt jedoch nach wie vor eine Diskriminierung zu.

5.2 Ähnlichkeit zu Texturklassen

Aufgabe der Klassifikation ist die Bestimmung der wahrscheinlichsten Klasse ω_i für einen gegebenen Merkmalsvektor x. Die Bayessche Entscheidungsregel [Fuk90] besagt, dass dies die Klasse mit der höchsten a-posteriori Wahrscheinlichkeit $q_i(x)$ ist:

$$q_i = \frac{p_i(x)}{p(x)} \tag{5.1}$$

Mit:

$$p(x) = \sum_{l=1}^{L} p_i(x) \tag{5.2}$$

Dabei steht $p_i(x)$ für die klassenspezifischen Verteilungsfunktionen und $p(x)$ für die Mischdichtefunktion (mixture density function).

Von dieser Entscheidungsregel unberührt ist die Frage, wie *ähnlich* der Merkmalsvektor x zu der so bestimmten *wahrscheinlichsten* Klasse ω_i ist. Es besteht die Möglichkeit, dass die a-posteriori Wahrscheinlichkeiten eine eindeutige Klassenzuordnung ergeben, obwohl der Merkmalsvektor x weitab des Zentrums der ermittelten Klasse ω_i liegt. Für viele Anwendungen in der Praxis ist dies nicht akzeptabel. Vielmehr soll in diesen Fällen auf eine Klassifikation verzichtet und der Merkmalsvektor x der Rückweisungsklasse zugewiesen werden. Um den Abstand zwischen Merkmalsvektor x und Klasse ω_i zu quantifizieren, werden in diesem Abschnitt geeignete Abstandsmaße eingeführt.

An einem einfachen Beispiel soll das Grundproblem verdeutlicht werden. In Abbildung 5.1 sind die Verteilungsfunktionen für 2 Klassen wiedergegeben. Die Merkmalsvektoren x_a und x_b wurden so gewählt, dass x_a nahe dem Schwerpunkt der Verteilung p_1 von Klasse 1 und x_b weit außerhalb der Verteilung p_1 liegen. Trotz ihrer unterschiedlichen Lage weisen sie nahezu identische a-posteriori Wahrscheinlichkeiten auf, nämlich $q_1(x_\mathrm{a}) \approx q_1(x_\mathrm{b}) \approx 1$ und $q_2(x_\mathrm{a}) \approx q_2(x_\mathrm{b}) \approx 0$. Damit werden sowohl x_a als auch x_b der Klasse 1 zugeschlagen. Die unterschiedliche Lage der Merkmalsvektoren bezüglich Klasse 1 wird durch die a-posteriori Wahrscheinlichkeit nicht wiedergegeben.

Während die a-posteriori Wahrscheinlichkeit q_i die *relativen* Wahrscheinlichkeiten für das Auftreten von x unter *allen* gegebenen Klassen betrachtet, macht die klassenspezifische Wahrscheinlichkeitsdichte p_i eine *absolute* Aussage über die Auftrittswahrscheinlichkeit von x für *eine* Klasse ω_i. Daher lässt sich auf Basis von p_i ein absolutes Abstandsmaß zwischen einem Merkmalsvektor x und einer Klasse ω_i definieren, das hier als Ähnlichkeit s_i bezeichnet werden soll. Zwei Varianten können unterschieden werden:

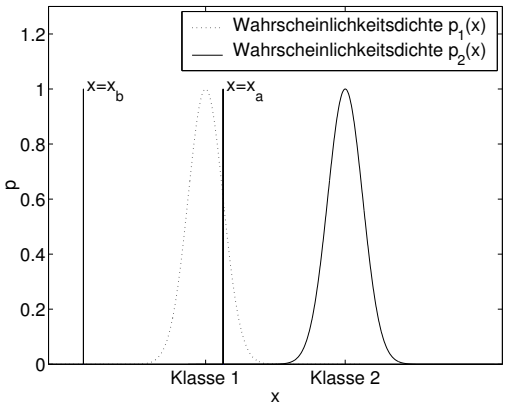


Abbildung 5.1: Die Merkmale x_a und x_b wurden so gewählt, dass x_a nahe dem Schwerpunkt der Verteilung $p_1(x)$ und x_b am Rande der Verteilung $p_1(x)$ liegen. Trotz ihrer unterschiedlichen Lage weisen sie nahezu identische a-posteriori Wahrscheinlichkeiten $q_i(x) = p_i(x)/\sum_{l=1}^{L} p_l(x)$ auf, nämlich $q_1(x_\mathrm{a}) \approx q_1(x_\mathrm{b}) \approx 1$ und $q_2(x_\mathrm{a}) \approx q_2(x_\mathrm{b}) \approx 0$. Die a-posteriori Wahrscheinlichkeit ist folglich nicht geeignet, die absolute Ähnlichkeit eines Merkmalsvektors x zu den verschiedenen Klassen zu quantifizieren.

$$\text{Absolute Ähnlichkeit:} \quad s_{\mathrm{abs},i}(x) = p_i(x) \tag{5.3}$$

$$\text{Relative Ähnlichkeit:} \quad s_{\mathrm{rel},i}(x) = \int_{p_i(x') \leq p_i(x)} p_i(x')dx' \tag{5.4}$$

„Absolut“ steht hier für die Ähnlichkeit zwischen einem Merkmalsvektor x und einer Klasse ω_i, „Relativ“ dagegen für die Ähnlichkeit eines Merkmalsvektors x im Vergleich zu allen anderen Merkmalsvektoren. Die Unterschiede zwischen diesen Definitionen sind in Abbildung 5.2 visualisiert. Im linken Bild wurden für beide Klassen die Bereiche markiert, die über eine absolute Ähnlichkeit $s_\mathrm{abs} \geq 0{,}2$ verfügen und damit eine Wahrscheinlichkeitsdichte p_i über 0,2 aufweisen. Im rechten Bild dagegen ist die relative Ähnlichkeit das Auswahlkriterium. Sie wurde so gewählt ($s_\mathrm{rel} = 0{,}07$), dass der markierte Bereich für Klasse ω_1 unverändert bleibt. Bedingt durch die flachere Verteilung vergrößert sich der Bereich für Klasse ω_2 deutlich.

Für die praktische Arbeit erweist sich das Konzept der relativen Ähnlichkeit als besonders zweckmäßig. Im Unterschied zur absoluten Ähnlichkeit muss der Anwender keine Kenntnis über die absolute Größe der Wahrscheinlichkeitsdichte p_i haben, was insbesondere bei variierender Dimensionalität des Merkmalsraumes von Vorteil ist. Die Vorgabe einer Mindestähnlichkeit von $s_{\mathrm{rel},i} = 0{,}2$ führt beispielsweise stets dazu, dass $1 - 0{,}2 = 80\,\%$ aller Realisierungen der Klasse ω_i dargestellt werden.

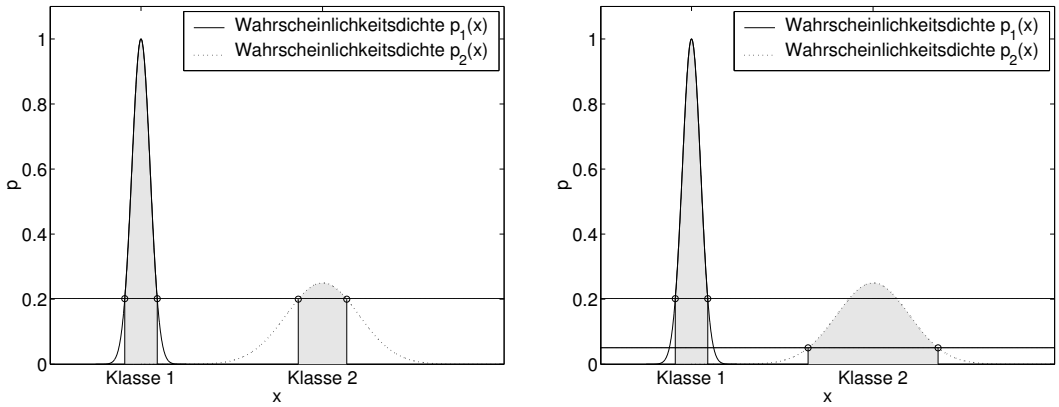


Abbildung 5.2: Visualisierung des Konzepts der absoluten und relativen Ähnlichkeit. Links sind die Bereiche markiert, deren absolute Ähnlichkeit größer als 0,2 ist. Rechts dagegen sind die Bereiche mit einer relativen Ähnlichkeit größer als 0,07 markiert.

Beruht die Klassifikation auf Normalverteilungen, gegeben durch die Erwartungswerte μ_i, Kovarianzmatrizen Σ_i, ergibt sich die relative Ähnlichkeit $s_{\text{rel}}(x)$ für den eindimensionalen Fall $n = 1$ aus der komplementären Fehlerfunktion [PTVF92]:

$$s_{\text{rel}}(x) = \text{erfc}\left(\frac{d}{\sqrt{2}}\right) \tag{5.5}$$

Mit:

$$d^2(x) = (x - \mu)^T \Sigma^{-1} (x - \mu) \tag{5.6}$$

Für den allgemeinen Fall ($n \geq 1$) muss die unvollständige Gammafunktion verwendet [PTVF92] werden:

$$s_{\text{rel}}(x) = \text{gammq}\left(\frac{n}{2}, \frac{d^2}{2}\right) \tag{5.7}$$

Für andere Klassifikatoren können in der Regel keine geschlossenen Ausdrücke angegeben werden, da das Integral über die Wahrscheinlichkeitsdichte $p_i(x)$ nicht bekannt ist. Ein eleganter Ausweg ergibt sich aus der Tatsache, dass die relative Ähnlichkeit eines Vektors x zur Klasse ω_i dem Anteil der Testvektoren x_k mit $p(x) \geq p(x_k)$ entspricht (vgl. Gleichung 5.4). Es genügt also, für jeden Testvektor x_k mit Hilfe des adaptierten Klassifikators die Wahrscheinlichkeitsdichte $p(x_k)$ zu schätzen und die Integration über p anstatt über x auszuführen. Damit lässt sich das Integral über x durch eine Summe über die Testvektoren x_k abschätzen, die nach aufsteigenden Wahrscheinlichkeitsdichten $p(x_k)$ sortiert sind. Das Vorgehen sei an einem Beispiel demonstriert. In Abbildung 5.3 sehen wir links eine Wahrscheinlichkeitsverteilung $p(x)$, die aus

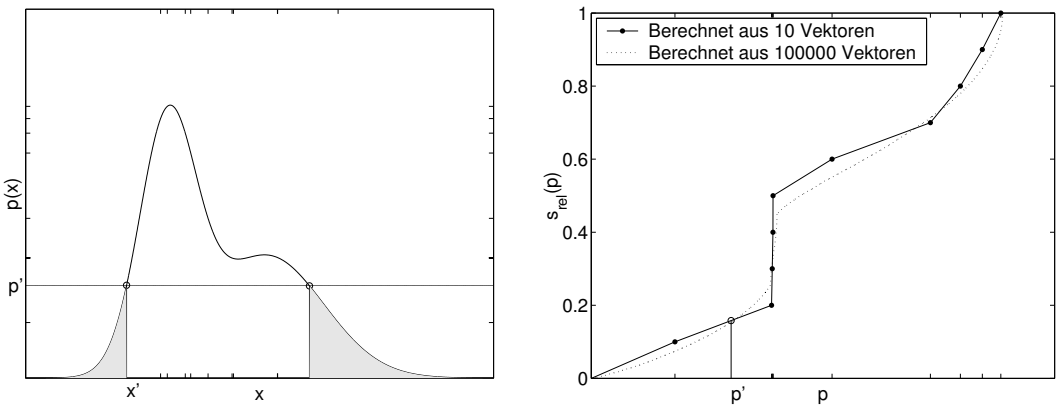


Abbildung 5.3: Ableitung der relativen Ähnlichkeit s_{rel} aus der Testmenge für eine beliebige Wahrscheinlichkeitsverteilung $p(x)$.

der Überlagerung von 2 Normalverteilungen hervorgeht und hier als bekannt vorausgesetzt wird. Die Testmenge bestehend aus 10 Vektoren ist durch die Häkchen auf der x-Achse markiert. Für ein willkürlich gewähltes x' mit einer Wahrscheinlichkeitsdichte p' soll nun die relative Ähnlichkeit bestimmt werden. Dazu werden die p_k der Testvektoren von der Ordinate des linken Diagramms auf die Abszisse des rechten Diagramms übertragen. Jedem p_k wird die relative Häufigkeit der Testvektoren mit $p \leq p_k$ zugewiesen. Die gesuchte relative Ähnlichkeit $s_{\text{rel}}(x')$ ergibt sich als $s_{\text{rel}}(p')$ durch (lineare) Interpolation.

Das hier entwickelte Konzept der Ähnlichkeit kann in der Praxis dazu eingesetzt werden, um die Klassifizierbarkeit eines Merkmalsvektors x zu überprüfen. Weist der Merkmalsvektor x eine hohe relative Ähnlichkeit zur wahrscheinlichsten Klasse ω_i auf, ist mit einer Fehlerrate zu rechnen, die derjenigen auf den Testdaten entspricht. Bei einer niedrigen Ähnlichkeit dagegen kann keine zuverlässige Aussage über die Wahrscheinlichkeit einer Fehlklassifikation gemacht werden, da das entsprechende Gebiet im Merkmalsraum nicht exploriert wurde.

5.3 Rotationsinvariantes Klassifikationsschema

Eine wesentliche Anforderung an Klassifikationssysteme besteht darin, Objekte auch in veränderter Erscheinungsform sicher zu erkennen – man spricht von invarianter Erkennung bezüglich einer Transformation T. In der Texturanalyse kommt der Rotation eine besondere Bedeutung zu. Im Folgenden sollen Lösungsansätze für die rotationsinvariante Klassifikation vorzustellt werden, die auf „Invarianten Merkmalen“ bzw. „Invarianter Klassifikation“ beruhen.

Rotationsinvariante Merkmale zeichnen sich dadurch aus, dass sie bei Rotation des Objekts unverändert bleiben. In Abschnitt 4.2.3 und 4.1.4 wurden

zwei Mechanismen zur Konstruktion von rotationsinvarianten Merkmalen vorgestellt. Für die Merkmale vom Typ LBP-ROT beruht die Invarianzerzeugung auf dem Prinzip „Invarianz durch Normierung“. Aus einem Bildausschnitt g werden lokale binäre Muster LBP berechnet. Diese sind nicht rotationsinvariant. Die Rotationsinvarianz wird durch Abbildung der LBP auf die Muster LBP-ROT erzeugt. Für Merkmale des Typs HP2-ROT wird der Ansatz „Invarianz durch Momentenbildung“ verfolgt. Aus einem Bildausschnitt g werden Merkmale für R unterschiedliche Rotationslagen berechnet. Aus diesem Satz rotationsabhängiger Merkmale lassen sich durch Momentenbildung rotationsinvariante Merkmale ableiten, etwa Mittelwerte und Varianzen. Eine ähnliche Vorgehensweise findet sich in [SM95] [SB98] unter dem Stichwort „Invariante Integration über die Transformationsgruppe“.

Eine Alternative zu der Berechnung invarianter Merkmale stellt die rotationsinvariante Klassifikation dar. Sie wurde im Rahmen dieser Arbeit nicht näher untersucht, jedoch soll sie zur Abgrenzung kurz vorgestellt werden. Im Unterschied zum Ansatz „Invariante Merkmale“ wird hier die Invarianz nicht über die Merkmale sondern über das Klassifikationsschema hergestellt. Ähnlich wie bei den Merkmalen $HP2$ wird zunächst ein Satz von R richtungsabhängigen Merkmalen berechnet, beispielsweise $x_{0\circ}$, $x_{45\circ}$, $x_{90\circ}$, $\ldots$, $x_{315\circ}$ für $R = 8$. Die Klassifikation besteht in einem Vergleich dieses Merkmalsvektors $x = (x_{0\circ}, x_{45\circ}, x_{90\circ}, \ldots, x_{315\circ})^T$ mit abgespeicherten Referenzvektoren r_i durch Kreuzkorrelation. Im Unterschied zum Ansatz „Invariante Merkmale“ liefert dieser Ansatz nicht nur die wahrscheinlichste Klasse ω_i sondern auch noch die Größe des Transformationsschritts, hier den Rotationswinkel φ. Systeme, die auf dieser Basis realisiert wurden, sind in [PTCH97] [Bre02] [HG84] beschrieben.

5.4 Implizite Merkmalsgewichtung

Die Leistungsfähigkeit eines Klassifikationssystems hängt wesentlich von den Eingangsgrößen, den sogenannten „Merkmalen“ ab. Um deren Qualität zu beurteilen, müssen zwei Einflussfaktoren berücksichtigt werden: 1) Die Separabilität beschreibt die Fähigkeit der Merkmale, zwischen den gegebenen Klassen zu unterscheiden. 2) Die Schätzgüte gibt an, wie zuverlässig die Verteilungen dieser Merkmale bestimmt werden können. In Kapitel 6 werden ausführlich Verfahren behandelt, die die Selektion von Merkmalsuntermengen mit maximaler Separabiltät zum Ziel haben. In diesem Abschnitt liegt der Fokus dagegen auf der Gewichtung von Merkmalen gemäß ihres relativen Schätzfehlers. Wir fordern, dass Merkmale, die einen hohen Schätzfehler aufweisen, schwächer gewichtet werden als Merkmale mit niedrigen Schätzfehlern.

In Abschnitt 4.2.6 wurden Merkmale vorgestellt, die den relativen Auftrittswahrscheinlichkeiten von lokalen binären Mustern LBP entsprechen. In

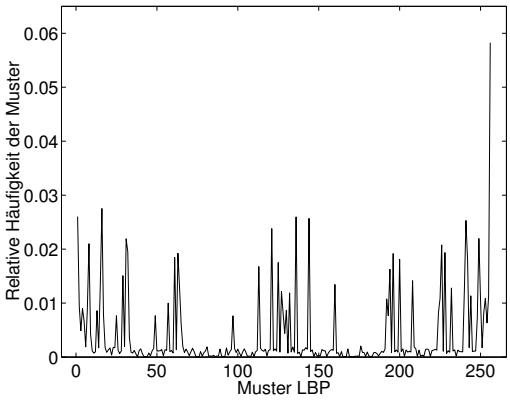


Abbildung 5.4: Beispiel für die Verteilung der Auftrittswahrscheinlichkeiten der lokalen binären Muster LBP.

einer Histogrammdarstellung wie in Abbildung 5.4 ist zu sehen, dass sich die Auftrittswahrscheinlichkeiten der Muster sehr stark unterscheiden. Um ihren relativen Schätzfehler $\varepsilon_{\text{rel},i}$ abzuschätzen, nähern wir die Binomialverteilung für das Auftreten des Musters i durch Normalverteilung mit Hilfe der Näherungsformel von De Moivre-Laplace an:

$$\begin{aligned} \mu_i &= np_i \\ \sigma_i &= \sqrt{np_iq_i} \end{aligned} \tag{5.8}$$

wobei n für die Zahl der Histogrammeinträge und p_i für die Auftrittswahrscheinlichkeit des Musters i steht. q_i ist durch $q_i = 1 - p_i$ gegeben. Den relativen Schätzfehler schätzen wir mit $\varepsilon_{\text{rel},i} = \sigma_i/\mu_i = \sqrt{q_i/np_i}$ ab. Für kleine p_i folgt $\varepsilon_{\text{rel},i} \sim 1/\sqrt{p_i}$. Das bedeutet, der relative Schätzfehler $\varepsilon_{\text{rel},i}$ ist umso größer, je seltener ein Muster auftritt. Neben dieser statistischen Motivation, die Merkmale mit niedriger Auftrittswahrscheinlichkeiten schwächer zu gewichten, besteht ein weiteres Argument in ihrer niedrigen visuellen Relevanz. Relative Änderungen in den Auftrittswahrscheinlichkeiten der dominierenden Merkmalen wirken sich auf das visuelle Erscheinungsbild der Textur stärker aus als bei Merkmalen mit kleinen Auftrittswahrscheinlichkeiten.

Es stellt sich nun die Frage, inwieweit die in Abschnitt 4.2.6 vorgestellten Klassifikatoren diesen Anforderungen and die Merkmalsgewichtung entsprechen. Dazu untersuchen wir, wie sich der Abstand $d_{\text{Euklid},i}$ bzw. $d_{\text{Log},i}$ verändert, wenn ein Merkmal x_i von dem Referenzwert p_i um Δx_i abzuweicht. Für den Abstandsklassifikator ergibt sich eine Abstandsänderung $\Delta d_{\text{Euklid},i} = \Delta x_i^2$, für die Kreuzentropie dagegen von $\Delta d_{\text{Log},i} = -\Delta x_i \ln p_i$. Setzen wir Δx_i mit der zu erwartenden Standardabweichung $\sigma_i = \sqrt{np_iq_i} \approx \sqrt{np_i}$ für kleine p_i gleich, so lassen sich für den Abstandsklassifikator folgende Zusammenhänge ableiten:

$$\begin{aligned}
\Delta d_{\text{Euklid},i} &\sim \Delta x_i^2 \\
&\sim p_i && (5.9)\\
\varepsilon_{\text{rel},i} &\sim \frac{1}{\sqrt{p_i}} && (5.10)\\
\Longrightarrow \Delta d_{\text{Euklid},i} &\sim \frac{1}{\varepsilon_{\text{rel},i}^2} && (5.11)
\end{aligned}$$

Gleichung 5.11 besagt, dass die Gewichtung $\Delta d_{\text{Euklid},i}$ eines Merkmals mit steigendem relativen Fehler gemäß $1/\varepsilon_{\text{rel},i}^2$ abnimmt. Der Abstandsklassifikator verfügt damit über eine implizite Merkmalsgewichtung, die der eingangs gestellte Forderung entspricht.

Für die Kreuzentropie ergibt sich in analoger Weise:

$$\begin{aligned}
\Delta d_{\text{Log},i} &\sim -\Delta x_i \ln p_i \\
&\sim -\sqrt{p_i} \ln p_i && (5.12)\\
\varepsilon_{\text{rel},i} &\sim \frac{1}{\sqrt{p_i}} && (5.13)\\
\Longrightarrow \Delta d_{\text{Log},i} &\sim \frac{1}{\varepsilon_{\text{rel},i}} \ln \varepsilon_{\text{rel},i} && (5.14)
\end{aligned}$$

Hier nimmt die Gewichtung $\Delta d_{\text{Log},i}$ eines Merkmals mit steigendem relativen Fehler $\varepsilon_{\text{rel},i}$ langsamer ab. Die Kreuzentropie weist somit hinsichtlich der impliziten Merkmalsgewichtung nicht die Vorteile der Abstandsklassifikation auf.

Vergleichen wir abschließend den Ansatz der impliziten Merkmalsgewichtung mit dem Ansatz der Merkmalsselektion (vgl. Abschnitt 6), so lässt sich sagen, dass die implizite Merkmalsgewichtung eher dem Prinzip der „Signalrepräsentation“ entspricht, die Merkmalsselektion dagegen dem Prinzip der „Signaldiskriminierung“. Während die implizite Merkmalsgewichtung die dominanten Merkmale bevorzugt, wählt die Merkmalsselektion Merkmale aus, die besonders unterscheidungskräftig sind (vgl. [Fuk90], S. 442 'Difference between Signal Representation and Classification'). Experimentelle Bestätigung findet dieser Befund in Abschnitt 8.2.3.

Überleitend zum nächsten Abschnitt sei angemerkt, dass die in diesem Abschnitt betrachteten Klassifikatoren (Abstandsklassifikator und Kreuzentropie) nicht auf die Modellierung der Verteilungen im Merkmalsraum abzielen, sondern der Abstand eines Merkmalsvektors x zu einem Referenzvektor r die zentrale Größe darstellt. Sowohl dem euklidischen Abstand als auch der Kreuzentropie liegen die einfache Annahme zugrunde, dass mit wachsendem Abstand

zwischen Merkmalsvektor x und Referenzvektor r die visuell empfundene Ähnlichkeit der zugrundeliegenden Texturen abnimmt. Eine visuelle Gewichtung der Merkmale hinsichtlich der durch sie ausgelösten Sinneseindrücke findet in der Regel nicht statt.

5.5 Visuelle Ähnlichkeit

Zu den Kernaufgaben der Texturanalyse gehört es, die Ähnlichkeit zwischen zwei Texturen festzustellen. Doch was ist unter Ähnlichkeit zu verstehen? Betrachten wir 2 texturerzeugende Prozesse ω_1 und ω_2, für die sich jeweils eine (unendliche) Sequenz von Merkmalsvektoren x berechnen lässt. In einem ersten Schritt müssen die Objekte festgelegt werden, die auf Ähnlichkeit zu untersuchen sind. 3 Paarungen sind denkbar, nämlich die Ähnlichkeit zwischen Klassen $\omega_i - \omega_j$, zwischen Klasse und Merkmalsvektor $\omega_i - x_k$ sowie zwischen Merkmalsvektoren $x_k - x_l$. Der zweite Schritt besteht in der Definition von Ähnlichkeitsmaßen. Dabei kann zwischen den Ansätzen „Statistische Ähnlichkeit“ und „Visuelle Ähnlichkeit“ unterschieden werden.

Statistische Ähnlichkeit: Dieser Ansatz basiert auf dem Konzept der Texturklasse, die gegeben ist durch die Menge aller möglichen Realisierungen. Im Merkmalsraum wird sie durch Angabe der klassenspezifischen Wahrscheinlichkeitsdichte $p_i(x)$ beschrieben. Die Ähnlichkeit zwischen Texturklassen $\omega_i - \omega_j$ ergibt sich aus der Durchdringung der Verteilungen $p_i(x)$ und $p_j(x)$ (vgl. Abschnitt 6.2). Die Ähnlichkeit zwischen Klasse und Merkmalsvektor $\omega_i - x_k$ hängt von $p_i(x_k)$ ab (vgl. Abschnitt 5.2. Ein direkter Vergleich zwischen zwei Merkmalsvektoren $x_k - x_l$ ist nicht möglich. Zielsetzung für die Merkmalsauswahl ist die *Unterscheidung* von Texturen, d. h. die Merkmale werden so gewählt, dass die Durchdringung der Texturklassen im Merkmalsraum minimiert wird.

Visuelle Ähnlichkeit: Wesentliches Element ist der Merkmalsvektor x als Punkt im Merkmalsraum. Das Konzept der Klasse ist bedeutungslos. Die Ähnlichkeit zwischen zwei Merkmalsvektoren $x_k - x_l$ wird über eine Metrik $d_{\text{vis}} = d_{\text{vis}}(x_k, x_i)$ im Merkmalsraum definiert, die den Unterschied in der visuellen Empfindung quantifiziert. Sollen Klassen auf ihre Ähnlichkeit untersucht werden, sind die Klassen auf ihre konstituierenden Referenzvektoren x_k zurückzuführen. Zielsetzung für die Merkmalsauswahl ist die *Beschreibung* des visuellen Eindrucks. Nur wenn die Merkmalsvektoren x den visuellen Eindruck vollständig wiedergeben, kann die Ähnlichkeit zwischen beliebigen Merkmalsvektoren korrekt abgeleitet werden.

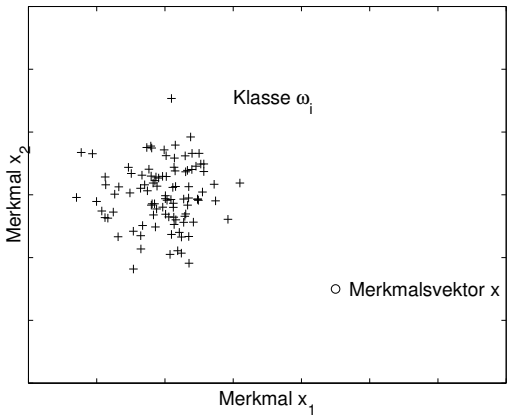


Abbildung 5.5: Im Sinne der statistischen Verteilung ist der Merkmalsvektor x unähnlich zur Klasse ω_i. Über die visuelle Ähnlichkeit ist damit aber noch keine Aussage gemacht.

Die meisten Verfahren der Texturklassifikation beruhen heute auf dem Ansatz der „Statistische Ähnlichkeit“, da eine Quantifizierung der visuell empfundenen Ähnlichkeit für allgemeine Texturen nach wie vor am Mangel geeigneter Merkmale und Metriken scheitert. Der Nachteil dieses Ähnlichkeitsbegriffs sei am Beispiel von Abbildung 5.5 skizziert. Neben einer Klasse ω_i, die durch eine Menge von Referenzvektoren x_k gegeben sei, liegt ein weiterer Merkmalsvektor x vor. Im Sinne der statistischen Ähnlichkeit besteht ein großer Abstand zwischen der Klasse ω_i und dem Vektor x. Über die visuelle Ähnlichkeit kann ohne Angabe einer Metrik d_{vis} keine Aussage gemacht werden. Ansätze für Beurteilung der visuellen Ähnlichkeit finden sich in [RW90] [Dau88] [MP00] [Jul75] [Wer58]. Für die Zukunft besteht die Aufgabe darin, eine Verknüpfung zwischen mathematisch definierten Texturmerkmalen und der durch sie ausgelösten visuellen Empfindung herzustellen. Ein Maß, um die visuelle Ähnlichkeit experimentell zu bestimmen, könnte auf der Wahrscheinlichkeit basieren, dass Texturen durch den Menschen verwechselt werden.

Kapitel 6

Merkmalsselektion

Ein wesentlicher Aspekt bei der Adaption eines Klassifikationssystems besteht in der Auswahl eines geeigneten Merkmalssatzes. Aus der Menge der verfügbaren Primärmerkmale α_i, $i = 1 \ldots M$ sollen $m \leq M$ Merkmale abgeleitet werden, um das gewählte Gütekriterium zu optimieren. In Abhängigkeit von der Art und Weise, wie dieser Merkmalssatz aus den Primärmerkmalen gewonnen wird, sprechen wir von Merkmalsselektion, Merkmalsgewichtung, Merkmalsextraktion oder Merkmalsoptimierung. Merkmalsselektion [JZ97] [SS88] [PS98] bedeutet, dass aus den M Primärmerkmalen m Merkmale ausgewählt werden. Die nicht selektierten Merkmale werden komplett verworfen. Im Unterschied zu dieser binären Auswahl wird bei der Merkmalsgewichtung jedem Merkmal ein kontinuierlicher Gewichtsfaktor zugeordnet. Bei der Merkmalsextraktion [BNS98] [LGAD99] [MJ95] [PSK98] werden neue Merkmale aus den Primärmerkmalen gewonnen, indem diese transformiert oder kombiniert werden. Neben linearen Funktionen können dazu auch nicht-lineare Funktionen eingesetzt werden. Die Merkmalsoptimierung [JAH97a] [MCKV97] [Wag97] schließlich setzt auf die Optimierung von freien Parametern in den Primärmerkmalen. Schwerpunkt der Untersuchungen in diesem Abschnitt ist die Merkmalsselektion. In den Abschnitten 5.4 und 7 wird näher auf die (implizite) Merkmalsgewichtung bzw. die Merkmalsoptimierung eingegangen.

6.1 Motivation

Die Motivation, die Menge an Primärmerkmalen auszudünnen, beruht zum einen auf einer Beschleunigung der Merkmalsberechnung, zum anderen auf einer möglichen Reduzierung der Fehlerrate. Dieser zunächst erstaunliche Umstand, dass mehr Merkmale nicht zwangsläufig in einer niedrigeren Fehlerrate münden, wurde in einer Vielzahl von Untersuchungen beschrieben [JZ97] [Tru79] [OD92] [PS98] [PMW00]. Als Ursache für dieses Phänomen sind Schätzfehler anzusehen, die bei endlicher Lernstichprobe mit der Anzahl der Merk-

male zunehmen. Wie bereits in Abschnitt 5.1 beschrieben, werden sie je nach Blickwinkel auf den „Finite Sample Size Effect“ oder den „Curse of Dimensionality“ zurückgeführt. Zur Veranschaulichung betrachten wir die Auswirkungen, die mit der Hinzunahme eines weiteren Merkmals zu der bereits selektierten Merkmalsuntermenge verbunden sind:

- Unter der Annahme, dass die wahre Statistik des Merkmals bekannt sei, bringt dieses Merkmal einen Zugewinn an Information ΔI zur Unterscheidung der Klassen mit sich.
- In der Praxis ist die Größe der Lernstichprobe beschränkt. Dadurch ergeben sich Schätzfehler $\Delta\varepsilon$ in den Modellparametern.

Überwiegt der Zugewinn an Information ΔI die Auswirkungen des erhöhten Schätzfehlers $\Delta\varepsilon$, so sinkt die Fehlerrate, im anderen Fall steigt sie. Welcher der Effekte die Oberhand behält, ist in hohem Maße von dem verwendeten Klassifikator abhängig – beispielsweise kann eine veränderte Komplexität des Klassifikationsmodells eine Verringerung der Fehlerrate mit sich bringen [Tru79] [HL96] [HFT96]. Hier genügt es festzuhalten, dass die Hinzunahme von Merkmale unter gewissen Umständen zu einem Anstieg der Fehlerrate führt und damit Verfahren der Merkmalsselektion geeignet sind, die Fehlerrate zu senken.

6.2 Gütekriterien

Prinzipiell kommen verschiedene Auswahlkriterien für die Merkmalsselektion in Frage, so z.B. eine hohe Diskriminierungsfähigkeit, geringe Rechenzeit oder eine hohe Robustheit der Merkmale gegenüber instationären Texturen. Im Folgenden konzentrieren wir uns auf das Kriterium „Diskriminierungsfähigkeit“. Soll darüber hinaus die Rechenzeit berücksichtigt werden, so kann dies auf einfache Weise realisiert werden, indem sie mit entsprechender Gewichtung in ein Verbundkriterium einfließt.

Um das Kriterium „Diskriminierungsfähigkeit“ genauer zu fassen, sollen zwei Vorgehensweisen gegeneinander abgegrenzt werden. Beim direkten Ansatz werden die Vorgaben des Testdatensatzes mit den Ergebnissen des Klassifikators verglichen, der zuvor mit einer Untermenge U aller Merkmale trainiert wurde. Dieser Ansatz wird in Abschnitt 6.2.2 behandelt. Der zweite Ansatz beruht auf einem indirekten Messprinzip. Anstatt die Klassifikionsergebnisse auszuwerten, wird die Separierbarkeit der Klassen im Merkmalsraum analysiert. Ob und inwieweit gegebene Klassenverteilungen im Merkmalsraum noch getrennt werden können, hängt jedoch in hohem Maße vom verwendeten Klassifikatortyp ab. In Abbildung 6.1 sind zwei ineinander verschlungene Verteilungen

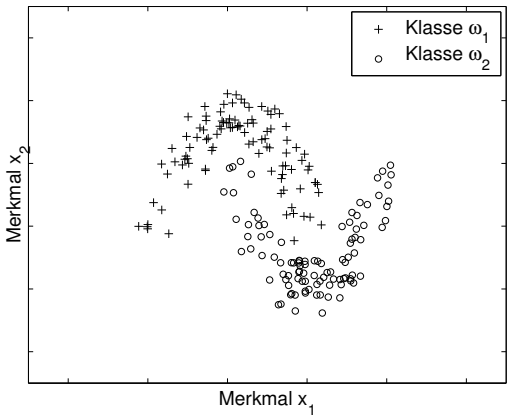


Abbildung 6.1: Die Überlappung zwischen den beiden Klassen im Merkmalsraum ist gering und damit die prinzipielle Separierbarkeit hoch. Ob sich mit diesen Merkmale niedrige Fehlerraten erzielen lassen, hängt jedoch stark von dem verwendeten Klassifikator ab.

dargestellt. Während diese Verteilungen mit einem nichtparametrischen Klassifikator wie z.B. einem Nächste-Nachbar-Klassifikator noch separierbar sind, stößt ein Normalverteilungsklassifikator als Beispiel für einen parametrischen Klassifikator an seine Grenzen. Die Qualitätsbeurteilung einer Merkmalsuntermenge U anhand der Klassenseparierbarkeit darf daher nicht losgelöst vom eingesetzten Klassifikatortyp betrachtet werden, sondern muss sich an dessen Funktionsweise und Leistungsfähigkeit orientieren. Im nächsten Abschnitt werden Separabilitätskriterien für verschiedene Klassifikatoren vorgestellt.

6.2.1 Klassifikatorspezifische Kriterien

Kullback-Leibler-Abstand

Um den Abstand zweier Verteilungen $p_1(x)$ und $p_2(x)$ zu quantifizieren ist aus der Informationstheorie der Kullback-Leiber-Abstand, auch Divergenz oder Information Gain genannt, bekannt [Kul59] [Hak88] [Fuk90]. Seine symmetrischen Form lautet:

$$J_{\text{Kull}} = \int p_1(x) \ln \frac{p_1(x)}{p_2(x)} dx + \int p_2(x) \ln \frac{p_2(x)}{p_1(x)} dx \tag{6.1}$$

Wichtige Eigenschaften des Kullback-Leiber-Abstands sind unter anderem die Invarianz gegenüber Koordinatentransformation und der monotone Anstieg des Abstandsmaßes bei Hinzunahme weiterer Merkmale. Diese Eigenschaft ist darauf zurückzuführen, dass die Verteilungen p_i als wahr angenommen werden. Mögliche Schätzfehler $\Delta\varepsilon$ der Verteilungen, die dem Informationsgewinn ΔI entgegenwirken könnten, gehen in den Kullback-Leibler-Abstand nicht ein (vgl.

Abschnitt 6.1).

Die Auswertung des Funktionals $J_{\text{Kull}}(p_1, p_2)$ wird praktikabel, wenn die Verteilungen p_1 und p_2 durch Normalverteilungen approximiert werden [Fuk90]:

$$\begin{aligned} J_{\text{Kull}} &= \frac{1}{2}\,\text{tr}\,\{(\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_2 - \mu_1)(\mu_2 - \mu_1)^T\} \\ &\quad + \frac{1}{2}\,\text{tr}\,(\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 - 2I) \end{aligned} \tag{6.2}$$

Um das Kullback-Leibler Abstandsmaß effektiv bei der Merkmalsselektion zu verwenden, werden in einem ersten Schritt Erwartungswerte μ_i und Kovarianzmatrizen Σ_i aus den Trainingsvektoren x_k der Dimension M berechnet. Für reduzierte Merkmalsräume der Dimension $m \leq M$ ergeben sich die reduzierten Größen μ_i und Σ_i durch Streichen von Zeilen bzw. Spalten. Bei der Matrixinvertierung in Gleichung 6.2 können in der Praxis Probleme durch singuläre Matrizen auftreten, insbesondere wenn die Zahl der Trainingsvektoren die Dimension des Merkmalsraumes M nicht deutlich übersteigt. Daher wird vorgeschlagen, die Matrixinvertierung auf Basis der Spektralwertzerlegung durchzuführen [Fuk90]:

$$\Sigma^{-1} = \sum_{i=1}^{M} \frac{1}{\lambda_i} \Phi_i \Phi_i^T \tag{6.3}$$

Selbst wenn die Determinante $|\Sigma|$ extrem klein wird, bleiben die Eigenwerte λ_i in der Regel oberhalb kritischer Werte. Eine weitere Möglichkeit die Robustheit zu steigern, besteht darin, alle Eigenwerte, die unterhalb einer Schwelle λ_{min} liegen, auf den Wert λ_{min} zu setzen. Durch das Heraufsetzen der Eigenwerte auf λ_{min} wird garantiert, dass jede Klasse ein minimales Volumen im Merkmalsraum einnimmt. Dies läst sich interpretieren als eine minimale Messunsicherheit λ_{min}, die jedem Messwert zugeordent wird – in loser Analogie zur Größe h^M im quantenmechanischen Phasenraum. Wurden die Trainingsvektoren zuvor komponentenweise auf einen Mittelwert von 0 und eine Varianz von 1 normiert, hat sich in der Praxis ein Schwellwert $\lambda_{\text{min}} = 0{,}001$ bewährt. Für weitere Details zur Berechnung der Kullback-Leibler-Distanz J_{Kull} sei auf [PS98] verwiesen.

Bhattacharyya-Abstand

Aufgrund seiner besseren Kopplung an den Bayes-Fehler (Fehler des optimalen Klassifikators) wird häufig der Bhattacharyya-Abstand dem Kullback-Leibler-Abstand vorgezogen. Er stellt eine obere Schranke für den Bayes-Fehler dar. [Fuk90]:

$$J_{\text{Bhat}} = \int \sqrt{p_1(x)p_2(x)}dx \tag{6.4}$$

Wie bei dem Kullback-Leibler-Abstand existiert auch hier ein geschlossener Ausdruck für den Fall, dass die Verteilungen p_i durch Normalverteilungen angenähert werden:

$$\begin{aligned} J_{\text{Bhat}} &= \frac{1}{8}(\mu_2 - \mu_1)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (\mu_2 - \mu_1) \\ &\quad + \frac{1}{2} \ln \frac{\left|\frac{\Sigma_1+\Sigma_2}{2}\right|}{\sqrt{|\Sigma_1| \cdot |\Sigma_2|}} \end{aligned} \tag{6.5}$$

Die Formel gliedert sich in 2 Terme, von denen der erste für $\mu_1 = \mu_2$ und der zweite für $\Sigma_1 = \Sigma_2$ verschwindet. Der erste Term gibt daher die Separabilität der Klassen aufgrund des Unterschieds im Erwartungswert wieder, der zweite Term die Separabilität aufgrund des Unterschieds in der Kovarianzmatrix. Empfehlungen aus der Literatur zufolge liefert der Ausdruck 6.5 selbst dann gute Ergebnisse, wenn die wahren Verteilungen p_i von der Normalverteilung abweichen [Fuk90]. Untersuchungen zum Schätzfehler des Bhattacharyya-Abstands wurden von Djouadi *et al.* [DSG90] durchgeführt.

Mehrklassenproblem

Sowohl der Kullback-Leibler als auch der Bhattacharyya-Abstand messen die Separabilität zwischen 2 Klassen. Um die Separabilität einer Merkmalsmenge für L Klassen zu erhalten, müssen die paarweisen Abstande $J(p_i, p_j)$, oder kurz J_{ij}, aggregiert werden.

Zunächst definieren wir den Nächste-Klasse-Abstand als den minimalen Abstand einer Klasse ω_i zu allen anderen Klassen:

$$J_{\text{min},i} = \min_{i \neq j}(J_{ij}) \tag{6.6}$$

Damit ergeben sich die 3 naheliegendsten Varianten einer globalen Merkmalsgüte J_{glob} zu:

Mittlerer Abstand:

$$J_{\text{glob},1} = \frac{2}{L(L-1)} \sum_{i=1}^{L} \sum_{j=i+1}^{L} J_{ij} \tag{6.7}$$

Mittlerer Nächste-Klassen-Abstand:

$$J_{\text{glob},2} = \frac{1}{L} \sum_{i=1}^{L} J_{\text{min},i} \tag{6.8}$$

Minimaler Nächste-Klasse-Abstand:

$$J_{\text{glob},3} = \min_i (J_{\min,i}) \tag{6.9}$$

Für $L = 2$ fallen alle 3 Definitionen zusammen. Da jede Definition unterschiedliche Eigenschaften aus der Verteilung der paarweisen Abstände J_{ij} extrahiert, kann je nach Applikation die geeignete Variante gewählt werden. Zusätzlich ist daran zu denken, die in diese Definitionen eingehenden Abstände J_{ij} mit den Klassenauftrittswahrscheinlichkeiten P_i, P_j zu gewichten.

Nichtparametrisches Überlappungsmaß

Während der Kullback-Leibler- und der Bhattacharyya-Abstand in den verwendeten Näherungen an Normalverteilungsklassifikatoren angepasst sind, soll nun ein Separabilitätskriterium vorgestellt werden, dass auf dem Prinzip der Nächsten-Nachbar-Klassifikation beruht. Dazu wird die Überlappung der Klassen im Merkmalsraum ermittelt. Grundidee ist, die Klassenzugehörigkeit eines jeden Mustervektors mit den Klassenzugehörigkeiten seiner k nächsten Nachbarn zu vergleichen.

Das von König vorgeschlagene Gütekriterium [Kön95] [Lam98] lautet:

$$J_{\text{Kön}} = \frac{1}{N}\sum_{i=1}^{N} q_i\,, \qquad q_i = \frac{\sum_{j=1}^{k} q_{ij} + \sum_{j=1}^{k} n_{ij}}{2\sum_{j=1}^{k} n_{ij}} \tag{6.10}$$

$$n_{ij} = 1 - \frac{d_{ij}}{d_{ik}}\,, \qquad q_{ij} = \begin{cases} n_{ij} & ,\omega_i = \omega_j \\ -n_{ij} & ,\omega_i \neq \omega_j \end{cases}$$

Mit:

- N : Größe der Stichprobe
- k : Anzahl der nächsten Nachbarn
- d_{ij} : Euklidischer Abstand zwischen Datenpunkt x_i und seinem j-ten nächsten Nachbarn x_{ij}
- d_{ik} : Euklidischer Abstand zwischen Datenpunkt x_i und dem entferntesten seiner k Nachbarn x_{ik}
- ω_i : Klasse des Datenpunktes x_i
- ω_j : Klasse des j-ten Nachbarn x_{ij}

Die Güte $J_{\text{Kön}}$ ergibt sich als das arithmetische Mittel der lokalen Überlappungsgrade q_i aller Mustervektoren. Für Mustervektoren, deren k nächste Nachbarn der eigenen Klasse angehören, ergibt sich eine lokale Überlappung von $q = 1$, für Mustervektoren, deren k nächste Nachbarn nicht der eigenen Klasse angehören, ist $q = 0$. $J_{\text{Kön}}$ liegt damit zwischen 0 (vollständige Überlappung) und 1 (keine Überlappung).

Eine auf ähnlichen Überlegungen basierende Kenngröße wird in [Fuk90] unter dem Namen „Nonparametric between-class scatter matrix“ diskutiert.

Relevanz eines Neurons

Für neuronale Netze sind eine Vielzahl von Verfahren zur Netzminimierung bekannt [Zel94]. Werden sie eingesetzt, um Eingabeneuronen zu löschen, entsprechen sie in ihrer Wirkung der Merkmalsselektion. Von Mozer *et al.* [MS89] wurde ein Verfahren vorgeschlagen, das sich Skelettierung nennt. Das Kriterium, ob ein Neuron gelöscht werden soll, wird als Relevanz ρ bezeichnet und lässt sich in ähnlicher Weise wie die Anpassung der Gewichte beim Backpropagation aus der Differenz von Lernvorgabe und Ausgabe des Klassifikators berechnen. Die Skelettierung sieht vor, mit dem vollen Merkmalssatz zu beginnen und sukzessive die am wenigsten relevanten Eingabeneuronen zu löschen, bis ein Abbruchkriterium erfüllt ist. Das Vorgehen entspricht einer sequentiellen Rückwärtssuche (Sequential backward search). Anders als bei den zuvor besprochenen Gütekriterien ist es nicht möglich, die Güte für eine beliebige Merkmalsuntermenge U direkt zu berechnen. Vielmehr muss das neuronale Netz bei jeder Änderung der Merkmalsuntermenge U nachtrainiert werden. Damit sind Netztraining, Gütekriterium und Suchstrategie auf das Engste miteinander verbunden. Suchstrategien wie die sequentielle *Vorwärts*suche oder gleitende Suchverfahren lassen sich mit der Skelettierung im Rahmen des „Stuttgarter neuronale Netze Simulators“ [Uni02] nicht kombinieren.

Um die Leistungsfähigkeit der Skelettierung zu untersuchen, wurden Experimente zur Textursegmentierung durchgeführt. Die erzielten Ergebnisse [MP97] blieben jedoch hinter den Resultaten zurück, die sich mit den übrigen Gütekriterien und Suchstrategien ergaben. Da auch Zell [Zel94] von einer „sehr schlechten Generalisierungsfähigkeit“ berichtet, wurde dieser Ansatz nicht weiter verfolgt.

Zuordnung der Gütekriterien zu Klassifikatoren

Jedes der vorgestellten Gütekriterien ist durch seine Konstruktion eng mit einem Klassifikatortyp verbunden. Die entsprechenden Zuordnungen lauten:

- Kullback-Leibler-Distanz (Normalverteilungsnäherung)
  $\longrightarrow$ Normalverteilungsklassifikator
- Bhattacharyya-Distanz (Normalverteilungsnäherung)
  $\longrightarrow$ Normalverteilungsklassifikator
- Nichtparametrisches Überlappungsmaß
  $\longrightarrow$ Nächste-Nachbar-Klassifikator
- Relevanz eines Neurons
  $\longrightarrow$ Neuronales Netz

Wie zu Beginn von Abschnitt 6.2 beschrieben, sollte die Merkmalsselektion prinzipiell mit Gütekriterien durchgeführt werden, die sich an der Leistungsfähigkeit des nachgeschalteten Klassifikators orientieren. Diese Forderung wird von den angegebenen Paarungen in idealer Weise erfüllt. Dennoch kann es Gründe geben von dieser Regel abzuweichen. So kann das Gütekriterium „Relevanz eines Neurons“ die Erwartungen nicht erfüllen. Sollen dennoch neuronale Netze als Klassifikator eingesetzt werden, kann die Merkmalsselektion auf Basis des Bhattacharyya-Abstandes in Kombination mit sequentieller Vorwärtssuche durchgeführt werden. Das neuronale Netz wird mit der so bestimmten Merkmalsuntermenge U trainiert.

6.2.2 A-posteriori Wahrscheinlichkeit

Um die Qualität einer Merkmalsuntermenge U zu bestimmen, wurde im letzten Abschnitts die Separabilität der Klassen im Merkmalsraum untersucht. Eine Ausnahme bildete das Gütekriterium „Relevanz eines Neurons“, das sich aus der Abweichung zwischen Lernvorgabe und Ausgabe des Klassifikators ableitet. Dieser Ansatz lässt sich auf alle anderen Klassifikatoren übertragen. Die Grundidee besteht darin, zunächst einen Klassifikator zu trainieren und anschließend seine Klassifikationsleistung zu bewerten. Derartige Gütekriterien, auch als Verifikationsmaße bezeichnet, lassen sich aus den a-posteriori Wahrscheinlichkeiten $q_i(x)$ ableiten, indem für jeden Vektor x_k die prognostizierten Wahrscheinlichkeiten $q_i(x_k)$ mit der Klassifikationsvorgabe $z_i(x_k)$ verglichen werden (k: Index des Vektors, i: Klassenindex, t: Index der wahren Klasse, p: Index der Klasse mit höchster a-posteriori Wahrscheinlichkeit). Viele Verifikationsmaße haben die Form:

$$J_{\text{Prob}} = \frac{1}{N} \sum_{k=1}^{N} a_k \tag{6.11}$$

J_{Prob} liegt im Intervall $[0, 1]$, wobei $J_{\text{Prob}} = 1$ für die perfekte Prognose und $J_{\text{Prob}} = 0$ für die vollständige Fehlprognose steht. Die Wahl von a_k bestimmt die Eigenschaften des Verifikationsmaßes J_{Prob}. Typische Varianten sind [MCKV97] [Bri50]:

- Wahrscheinlichkeit, die richtige Klasse zu prognostizieren J_{Prob1}:

$$a_k = \begin{cases} 1\,, & t = p \\ 0\,, & \text{sonst} \end{cases} \tag{6.12}$$

- Mittlere a-posteriori Wahrscheinlichkeit der wahren Klasse J_{Prob2}:

$$a_k = q_t(x_k) \tag{6.13}$$

- Quadratisches Verifikationsmaß nach Mallet *et al.* J_{Prob3}:

$$a_k = \frac{1}{2} + q_t(x_k) - \frac{1}{2}\sum_{i=1}^{L} q_i(x_k)^2 \tag{6.14}$$

- Quadratisches Verifikationsmaß nach Brier J_{Prob4}:

$$a_k = 1 - \sum_{i=1}^{L} \left(q_i(x_k) - z_i(x_k)\right)^2 \tag{6.15}$$

Das einfachste Verifikationsmaß nach Gleichung 6.12 erweist sich in der Praxis als wenig geeignet. Die Ursache dafür ist, dass sich ein perfektes Verifikationsmaß $J_{\text{Prob}} = 1$ auch dann ergeben kann, wenn erhebliche Unterschiede zwischen den prognostizierten a-posteriori Wahrscheinlichkeiten $q_i(k)$ und den Klassifikationsvorgaben $z_i(k)$ bestehen. Bei kleinen Stichproben und einfachen Klassifikationsproblemen kann dies dazu führen, dass J_{Prob} ab einer gewissen Zahl der selektierten Merkmale identisch 1 wird. Die optimale Merkmalsmenge kann dann nicht gefunden werden, weil das Verifikationsmaß J_{Prob} über mangelhaftes Differenzierungsvermögen verfügt.

6.2.3 Gütekriterium für kontinuierliche Bewertung

Handelt es sich bei der zugrundeliegenden Aufgabe nicht um (diskrete) Klassifikation sondern um die Prognose eines reellwertigen Bewertungsvektors, bietet sich als Gütekriterium J_{Kont} der mittlere quadratische Abstand zwischen den prognostizierten Bewertungsvektoren y_k und den Vorgabevektoren z_k an:

$$a_k = \sum_{i=1}^{L} \left(y_i(x_k) - z_i(x_k)\right)^2 \tag{6.16}$$

Dieser Ausdruck stimmt in seiner Struktur mit dem Verifikationsmaß nach Brier (6.15) überein. Lediglich die Bedeutung von L hat sich verändert: Bei der diskrete Klassifikation steht L für die Anzahl der Klassen, hier für die Dimension des kontinuierlichen Bewertungsvektors y. Die Größe $y_i(x_k)$ entspricht der i-ten Komponente des Bewertungsvektors y, der aus dem Merkmalsvektor x_k vorhergesagt wurde.

6.3 Suchverfahren

Wie bei jeder Optimierungsaufgabe ist auch bei der Merkmalsselektion neben dem Gütekriterium J eine Suchstrategie festzulegen, die den Suchvorgang steuert. Ziel der Optimierung ist die Bestimmung der Merkmalsuntermenge

U mit m aus M Merkmalen, die maximale Separabilität J aufweist. Die Anzahl möglicher Kombinationen ist durch $\binom{M}{m}$ gegeben. Eine vollständige Suche scheidet bei realistischen Werten für M und m aus, denn bereits für $M = 100$ und $m = 10$ wären $\binom{M}{m} \approx 1{,}7 \cdot 10^{13}$ Kombinationen zu durchsuchen. Effiziente Suchverfahren spielen daher eine wichtige Rolle in der Merkmalsselektion. Begünstigend ist der Umstand, dass es in der Praxis nicht darauf ankommt, die optimale Lösung zu finden. Vielmehr genügt es, aus dem großen Vorrat von M Merkmalen eine kleine Zahl m effektiver Merkmale auszuwählen und nicht relevante Merkmale auszuschließen.

Im Folgenden sollen 3 Suchstrategien vorgestellt werden, die sich als besonders effizient herausgestellt haben [JZ97] [Wag97]. Wichtig ist, dass jedes der Suchverfahren mit einem der Gütekriterien (Ausnahme: Relevanz eines Neurons) aus Abschnitt 6.2 frei kombiniert werden kann.

6.3.1 Sequentielle Suche

Bei der sequentiellen Suche [Kit78] [PS98] wird in jedem Iterationsschritt zu den bereits selektierten m Merkmalen ein weiteres Merkmal hinzugefügt. Dieses Merkmal wird aus der Menge der noch nicht selektierten $M - m$ Merkmale so bestimmt, daß es zusammen mit den bereits gewählten Merkmalen die Güte J maximiert. Mit der sequentiellen Suche können vor allem dann gute Ergebnisse erzielt werden, wenn aus einer großen Merkmalsmenge eine kleine Untermenge bestimmt werden soll, also $m \ll M$ ist. Die Abweichungen von der optimalen Lösung nehmen in der Regel mit wachsendem m zu, bevor sie bei Annäherung von m an M wieder auf null zurückgehen. Eine Ursache für die suboptimale Auswahl ist der sog. „Nesting-Effect“, der darin besteht, dass einmal selektierte Merkmale nicht wieder deselektiert werden können. Neben der soeben skizzierten Vorwärtssuche, die mit der leeren Merkmalsmenge beginnt, kann die sequentielle Suche auch als Rückwärtssuche realisiert werden. Startpunkt ist dann die volle Merkmalsmenge, aus der die am wenigsten effektiven Merkmale entfernt werden. Der Rechenaufwand für die sequentielle Suche ist gering.

6.3.2 Gleitende sequentielle Suche

Die gleitende sequentielle Suche (Sequential-Floating-Search) [PNK94] beruht auf der sequentiellen Suche, beseitigt aber deren entscheidende Schwachstelle. Der „Nesting-Effect“ wird vermieden, indem bereits selektierte Merkmale zur Disposition gestellt und gegebenenfalls wieder entfernt werden. Nach Auswahl eines weiteren Merkmals wird untersucht, welches der selektierten Merkmale den kleinsten Beitrag zur Maximierung des Güte J leistet. Nur wenn das so bestimmte uneffektivste Merkmal mit dem zuletzt hinzugefügten Merkmal

übereinstimmt, wird die sequentielle Suche unmittelbar fortgesetzt. Im anderen Fall werden zuvor die uneffektivsten Merkmale solange deselektiert, bis die Güte $J_{i,\downarrow}$ unter den bereits erreichten Stand $J_{i,\uparrow}$ fällt. Wie die sequentielle Suche ist auch die gleitende sequentielle Suche sowohl in Vorwärts- als auch in Rückwärtsrichtung möglich. Mit dem Verfahren können bei mäßigem Zeitaufwand nahezu optimale Merkmalsuntermengen bestimmt werden [JZ97].

6.3.3 Branch-And-Bound

Das Branch-And-Bound Verfahren [NF77] [Fuk90] [YY93] beruht auf der Annahme einer Monotonieeigenschaft für des Gütekriteriums J:

$$J(U_1 \cup U_2) \geq J(U_1) \quad \forall \quad U_1, U_2 \subseteq Y \tag{6.17}$$

Das bedeutet, dass die Hinzunahme von Merkmalen zu einer Merkmalsuntermenge das Gütekriterium nur erhöhen kann. Gütekriterien wie der Kullback-Leibler- oder der Bhattacharyya-Abstand erfüllen diese Monotoniebedingung. Die Monotoniebedingung ermöglicht es, bei der Rückwärtssuche die optimale Merkmalsuntermenge zu finden, ohne alle $\binom{M}{m}$ Kombinationen auszutesten. Für Primärmerkmalsmengen Y mit einer Größe von $M \geq 30$ ist das Verfahren jedoch aufwendig, da die Komplexität des Algorithmus im ungünstigsten Fall exponentiell mit M anwächst.

6.3.4 Vorwärtssuche vs. Rückwärtssuche

Aufgrund der großen Zahl verfügbarer Texturmerkmale ist die Anzahl der Primärmerkmale M in der Regel weitaus größer als die Anzahl der auszuwählenden Merkmale m. Typische Werte liegen im Bereich $M = 50 \ldots 400$ und $m = 5 \ldots 20$. Unter diesen Bedingungen weist die Vorwärtssuche, auch Bottom-Up-Strategie genannt, große Geschwindigkeitsvorteile gegenüber der Rückwärtssuche (Top-Down-Strategie) auf. Diese beruhen zum einen darauf, dass für die genannten Werte die Anzahl der hinzuzufügenden Merkmale bei der Vorwärtssuche kleiner ist als die Anzahl der auszuschließenden Merkmale bei der Rückwärtssuche. Zum anderen liegen die Vorteile in der weitaus einfacheren Auswertung des Gütekriteriums J für kleine Merkmalsuntermengen U.

Da das Branch-And-Bound Verfahren nur für die Rückwärtssuche eingesetzt werden kann, ist es für die Merkmalsselektion bei der Texturanalyse nicht das bevorzugte Verfahren.

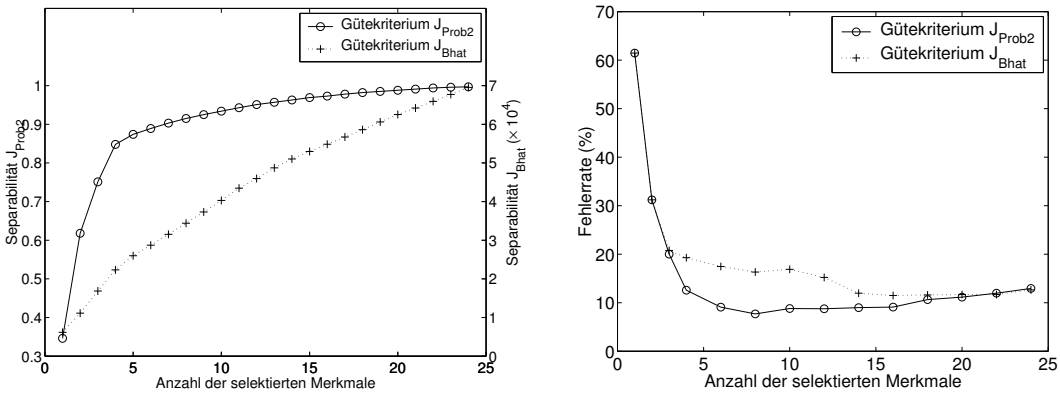


Abbildung 6.2: Verlauf der Separabilität (links) und Fehlerrate (rechts) für die Gütekriterien J_{Prob2} bzw. J_{Bhat} bei Selektion von $m = 1 \ldots 24$ aus $M = 270$ Merkmalen.

6.4 Experimente

Um die Leistungsfähigkeit der Merkmalsselektion zu demonstrieren, greifen wir 2 Konfigurationen heraus. In der ersten Konfiguration wird als Gütekriterium der Bhattacharyya-Abstand J_{Bhat} eingesetzt, in der zweiten Konfiguration die mittlere a-posteriori Wahrscheinlichkeit der wahren Klasse J_{Prob2}. Diese Gütekriterien werden jeweils mit der sequentiellen Vorwärtssuche kombiniert. Für die Experimente werden Texturen aus dem Brodatz-Album herangezogen, die in Abschnitt 8.1 detailliert beschrieben sind. Hier genügt die Information, dass es sich um ein 15-Klassen-Problem handelt und der Merkmalspool 270 primäre Merkmale des Typs HP2-ROT umfasst. Für jede der 15 Texturen stehen 64 Mustervektoren x der Dimension $N = 270$ zum Training bereit (vgl. Experiment G aus Abschnitt 8.1). Auf diesem Datensatz wird neben dem Training auch die Merkmalsselektion durchgeführt. Die Validierung der Merkmalsselektion findet auf Texturen statt, die gegenüber den Trainingsmustern rotiert sind. Insgesamt stehen 6720 Vektoren für die Validierung bereit. Diese Zahl ergibt sich aus der Anzahl der Texturen (15), der Anzahl der Rotationslagen (7) und der Anzahl der Muster (64) für jede der Rotationslagen.

Im linken Teil der Abbildung 6.2 ist der Verlauf des Gütekriteriums für die Experimente dargestellt. In beiden Fällen wächst die Separabilität J_{Bhat} bzw. J_{Prob2} monoton an. Dieses Verhalten ist typisch, wenn Klassifikatortraining und Auswertung des Gütekriteriums auf dem gleichen Datensatz durchgeführt werden, weil dem Zugewinn an Information ΔI kein Schätzfehler $\Delta\varepsilon$ gegenüber steht (vgl. Abschnitt 6.1). Im Gegensatz dazu sinkt die Fehlerrate auf dem Validierungsdatensatz zunächst mit wachsender Merkmalsanzahl, um nach Durchlaufen eines Minimums anzusteigen. Mit J_{Bhat} als Gütekriterium wird die Fehlerrate für $m = 16$ minimal und liegt mit 11,49 % deutlich über

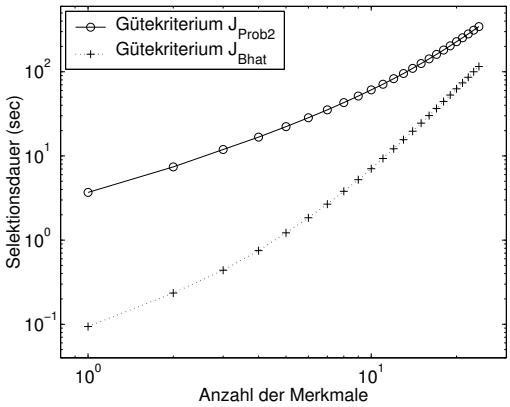


Abbildung 6.3: Zeitaufwand für die Selektion von $m = 1 \ldots 24$ aus $M = 270$ Merkmalen in doppelt logarithmischer Darstellung. Die Zeiten wurden auf einem Intel Pentium 4 Prozessor mit 2,4 GHz ermittelt.

der Fehlerrate von 7,69 %, die mit J_{Prob2} bei $m = 8$ erreicht wird. Offensichtlich ist die Kopplung von J_{Prob2} zur Fehlerrate stärker als diejenige von J_{Bhat}. Dieses Ergebnis ergibt sich auch in anderen Experimenten, so dass J_{Prob2} als das bevorzugte Gütekriterium gelten kann.

Spielt jedoch neben der Fehlerrate die für die Merkmalsselektion benötigte Zeit eine Rolle, bietet J_{Bhat} Vorteile. In Abbildung 6.3 ist der Zeitbedarf dargestellt, um m Merkmale aus $M = 270$ Primärmerkmalen zu selektieren. Dabei bleibt die Zeit unberücksichtigt, die das vorausgegangene Training des Klassifikators mit dem vollen Merkmalssatz erfordert. Es ist erkennbar, dass das Gütekriterium J_{Bhat} wesentlich schneller zu berechnen ist als J_{Prob2}. Jedoch nimmt mit wachsender Anzahl der zu selektierenden Merkmale der Zeitvorteil von J_{Bhat} ab. Ursache hierfür ist die Matrixinvertierung in Gleichung 6.5. Ein weiterer Einflussfaktor auf die Rechenzeit ist die Größe N des Datensatzes, auf dem das Gütekriterium berechnet wird. Unter der Annahme, dass die die Kovarianzmatrizen Σ_i für den vollen Merkmalssatz vorab berechnet wurden, hängt die Auswertedauer von J_{Bhat} nicht von der Größe des Datensatzes ab, während für J_{Prob2} der Zeitbedarf linear mit N ansteigt.

Zusammenfassend kann festgehalten werden, dass sich in der Praxis die folgenden Verfahren für die Merkmalsselektion bewährt haben. Sie bilden die Grundlage für die vergleichenden Untersuchungen in Kapitel 8:

- Gütekriterium: Mittlere a-posteriori Wahrscheinlichkeit der wahren Klasse J_{Prob2}
- Suchverfahren: Sequentielle Vorwärtssuche

Ist die Merkmalsselektion zeitkritisch, empfiehlt sich das Bhattacharyya-Abstandsmaß mit diesen Einstellungen:

- Vermeidung von Singularitäten der Kovarianzmatrizen: $\lambda_{\min} = 0{,}001$
- Aggregation der paarweisen Klassenabstände: Mittlerer paarweiser Abstand $J_{\text{glob},1}$

Kapitel 7

Merkmalsoptimierung mittels Evolutionsstrategie

Zielsetzung der im letzten Kapitel vorgestellte Merkmalsselektion ist es, einen kompakten Satz an Merkmalen mit hoher Diskriminierungsfähigkeit zu erhalten. Dieser Zielsetzung kann umso besser entsprochen werden, je größer der zur Verfügung stehende Merkmalspool ist. Aus Gründen der Rechenzeit ist seine Größe auf einige Hundert Merkmale begrenzt. Diese Begrenzung wirkt sich vor allem bei Texturmerkmalen störend aus, die über kontinuierliche Parameter verfügen. Man sieht sich dem Problem gegenübergestellt, die Parameter vor der Merkmalsselektion festlegen zu müssen. Vor allem bei Texturmerkmalen, die über mehrere Parameter verfügen, kann der Suchraum nur ungenügend dicht abgedeckt werden.

Als Alternative zur Merkmalsselektion bietet sich in diesen Fällen die Merkmalsoptimierung an. Anstatt aus einem Pool von M Merkmalen die m effektivsten Merkmale auszuwählen, werden nun die Parameter von m Merkmalen optimiert. Der endliche Suchraum der Merkmalsselektion mit $\binom{M}{m}$ Kombinationen erweitert sich dadurch auf den $\mathbb{R}^P$, wobei P die Gesamtzahl der zu optimierenden Parameter angibt.

Zwei wesentliche Änderungen gehen mit dem Übergang von der Merkmalsselektion zur -optimierung einher. Zum einen bringt die Variation der Parameter mit sich, dass die Merkmale für jeden Iterationsschritt neu aus den Mustertexturen berechnet werden müssen (vgl. Abbildung 3.1). Zum anderen müssen die bisher eingesetzten Suchverfahren durch Suchverfahren für den $\mathbb{R}^P$ ersetzt werden. Die Gütekriterien zur Bewertung eines Merkmalssatzes können dagegen unverändert übernommen werden.

Such- bzw. Optimierungsverfahren für diese Art von Problemstellung müssen ohne die Berechnung von Ableitungen auskommen. Falls der Definitionsbereich nicht über den ganzen $\mathbb{R}^P$ ausgedehnt ist, müssen die entsprechenden Nebenbedingungen durch geeignete Kodierung des Problems oder durch Straf-

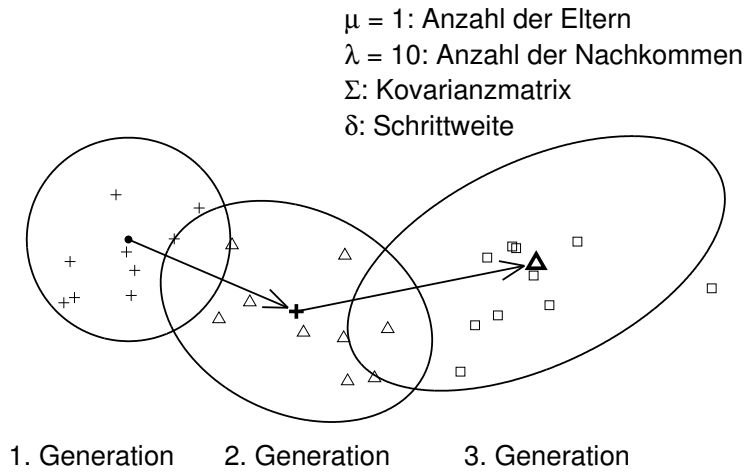


Abbildung 7.1: Prinzip der CMA-Evolutionsstrategie: In einer Generation wird das Individuum mit der höchsten Fitness bestimmt (fettgedrucktes Symbol), das als Elter für die nächste Generation dient. Die Kovarianzmatrix Σ (angedeutet durch Ellipse) und Schrittweite δ werden aktualisiert und bestimmen die Richtungs- und Größenverteilung der Mutationen.

terme realisiert werden [Mic92].

Laut Press *et al.* [PTVF92] bieten sich zwei Klassen von Optimierungsverfahren an. In die erste Klasse fallen die Verfahren der „konjugierten Gradienten“, zu denen die Methode von Powell gehört. Auch Mathematikprogramme wie Mathematica [Wol97] greifen bei der Minimierung auf diese Verfahren zurück. Die zweite Klasse wird unter dem Schlagwort des „Simulated Annealing“ geführt. Sie eignet sich für Probleme, in denen das „globale Extremum zwischen vielen, schlechteren lokalen Extrema versteckt ist“. In seiner Grundstruktur entspricht die von Rechenberg [Rec94] vorgeschlagene Evolutionsstrategie diesem Ansatz. Ebenfalls weite Verbreitung haben genetische Algorithmen gefunden [Gol89]. Im Gegensatz zur Evolutionsstrategie ist dort der dominierende Mechanismus zur Variantenerzeugung nicht die (gerichtete) Mutation, sondern die Kreuzung (Rekombination) von erfolgreichen Varianten.

7.1 CMA-Evolutionsstrategie

Evolutionsstrategien [KS97] zeichnen sich dadurch aus, dass in jeder Iteration, die *Generation* genannt wird, eine *Population* von *Individuen* betrachtet wird. Jedes Individuum besteht aus einem reellwertigen Parametervektor x, der die Erscheinung des Individuums bestimmt sowie Strategieparameter, die die Mutation steuern. Die Güte bzw. *Fitness* eines jeden Individuums entscheidet darüber, mit welcher Wahrscheinlichkeit seine *Gene* (Objekt- und Strategiepa-

rameter) in die nächste Generation einfließen.

Im Folgenden wird die Covariance-Matrix-Adaptation (CMA) Evolutionsstrategie von Hansen und Ostermeier [HO96] [HO97] vorgestellt. Im Unterschied zu anderen Evolutionsstrategien sind die Strategieparameter nicht den einzelnen Individuen zugeordnet, sondern haben für die gesamte Population Gültigkeit. Die im Laufe der Evolution akkumulierte Information über die Funktionslandschaft ist daher nicht ausschließlich durch die Gene aller Individuen der aktuellen Population und deren Fitness gegeben, sondern explizit in Form einer Kovarianzmatrix Σ abgelegt. Sie gibt an, in welchen Richtungen mit erfolgreichen Mutationen zu rechnen ist und ergibt sich aus der Sequenz der erfolgreichen Evolutionsschritte. Zu Beginn der Evolution liegt noch keine Information über die Funktionslandschaft vor, die Kovarianzmatrix Σ ist proportional zur Einheitsmatrix, d.h. keine Richtung ist bei der Mutation ausgezeichnet. In jeder Generation wird durch Auswahl des besten Individuums $x^{\mathrm{E}}_{i,g+1}$ (x^{E}: Elter, g: Generation) das Elter für die nächste Generation festgelegt. Die Kovarianzmatrix Σ wird mit dem Differenzvektor $x^{\mathrm{E}}_{g+1} - x^{\mathrm{E}}_{g}$ aktualisiert. Die Individuen der nächsten Generation ergeben sich aus dem selektierten Elter durch Mutationen, die gemäß der Kovarianzmatrix verteilt sind. Die Kondition der Kovarianzmatrix Σ (Verhältnis von größtem zu kleinstem Eigenwert) gibt an, wie stark die Verteilung der Mutationsschritte von einer isotropen Verteilung abweicht.

Neben der Kovarianzmatrix Σ hat die Schrittweite δ Einfluss auf die Erzeugung der Nachkommen, indem sie die Mutationsschritte skaliert. Die Schrittweite δ selbst ist nicht konstant, sondern passt sich an den Evolutionsfortschritt an. Ist der zuletzt realisierte Evolutionsschritt $x^{\mathrm{E}}_{g+1} - x^{\mathrm{E}}_{g}$ größer als der Erwartungswert der aktuellen Mutationsschrittverteilung, wird die Schrittweite δ vergrößert, im anderen Fall verkleinert.

In Abbildung 7.1 ist die Evolution über 3 Generationen schematisch dargestellt. Ausgehend von einem Startwert x^{E}_{0} wird die Population mit einer Kovarianzmatrix erzeugt, die der Einheitsmatrix entspricht. Das beste Individuum wird ausgewählt und die Kovarianzmatrix aktualisiert. Daraus ergibt sich die nächste Generation. Im Laufe der Evolution passt sich die Kovarianzmatrix Σ und die Schrittweite δ der Funktionslandschaft an.

Einsatzgebiete für die CMA-Evolutionsstrategie sind vor allem Optimierungsaufgaben im hochdimensionalen, multimodalen („mehrgipfeligen“) Suchraum [Pan98] [WKPR99].

7.2 Optimierung von Gaborfiltern

Mit Hilfe der CMA-Evolutionsstrategie soll die Diskriminierungsfähigkeit zwischen Texturen gesteigert werden, indem Texturmerkmale an die spezifische Prüfaufgabe angepasst werden. Die in Abschnitt 4.3 vorgestellten Gaborfilter

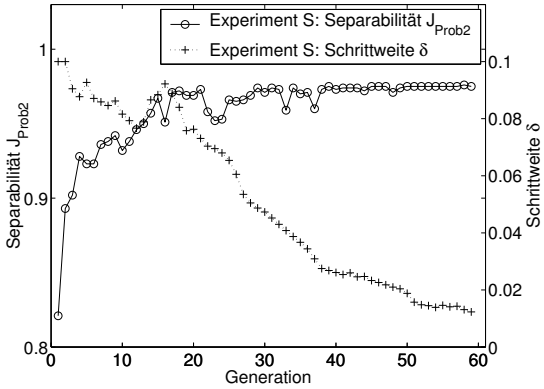


Abbildung 7.2: Typischer Verlauf des Evolutionsfortschritts bei der Optimierung von $T = 3$ Gaborfiltern. Während das Gütekriterium J_{Prob2} wächst, verringert sich die Schrittweite δ über die Generationen.

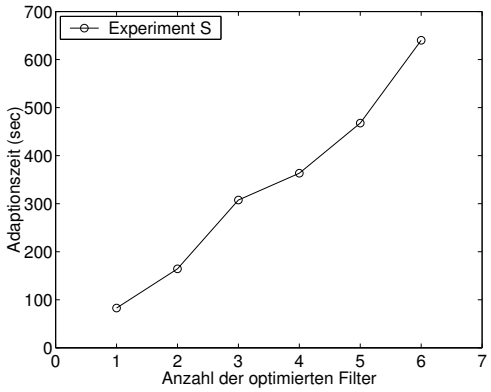


Abbildung 7.3: Zeitbedarf für die Optimierung von Gaborfiltern mittels CMA-Evolutionsstrategie über 30 Generationen à 10 Individuen. Die Zeiten wurden auf einem Intel Pentium 4 Prozessor mit 2,4 GHz ermittelt.

mit ihren jeweils 4 Parametern Zentralfrequenz u_0, Filterbreiten σ_x und σ_y und Orientierung θ sind dazu gut geeignet.

Im Vorgriff auf die Experimente von Abschnitt 8.3 sollen hier typische Ergebnisse für den Optimierungsverlauf gezeigt werden. Die Aufgabe besteht darin, die 10 Texturen des Datensatzes „Mosaic08“ mit $T = 3$ Gaborfiltern zu klassifizieren. Jeder Gaborfilter liefert 2 Merkmale, so dass die Dimension der Merkmalsvektoren x für die Optimierung $N = 3 \cdot 2 = 6$ beträgt. Im Unterschied zur Merkmalsselektion müssen die 200 Trainingsvektoren pro Texturklasse für jede Parameterkombination neu berechnet werden. Die Validierung der Optimierung wird auf der Texturszene vorgenommen, so dass Trainings- und Validierungsvektoren aus unterschiedlichen Bildausschnitten stammen. Insgesamt stehen 5490 Valdierungsvektoren für die 10 Texturen bereit.

Abbildung 7.2 zeigt den Optimierungsverlauf über 60 Generationen. Zunächst wächst das Gütemaß J_{Prob2} (vgl. Gleichung 6.13) steil an und geht ab der 30. Generation in ein Plateau über. Insgesamt verbessert sich die Diskriminierungsfähigkeit von 0,82 auf 0,98, wobei der theoretische Maximalwert von J_{Prob2} bei 1,0 liegt. Die Schrittweite δ verringert sich über alle Generationen um eine Größenordnung von 0,10 auf 0,012 während die Kondition der Kovarianzmatrix Σ von 1,0 auf 1,8 zunimmt. Die eigentliche Qualität der Optimierung lässt sich an der verringerten Fehlerrate auf der Validierungsmenge ablesen, die von 36,9 % auf 25,3 % zurückgeht. Für eine detaillierte Diskussion der Ergebnisse sei auf die vergleichenden Untersuchungen in Abschnitt 8.3 verwiesen.

Um den Zeitbedarf für die Merkmalsoptimierung einzuschätzen, wurde eine

Versuchsserie durchgeführt, bei der die Anzahl der Gaborfilter variiert wurde. Die Filterzahl reichte von $T = 1 \ldots 6$, dementsprechend lag die Parameterzahl zwischen $P = 4 \ldots 24$. Alle Optimierungsläufe umfassten 30 Generationen à 10 Individuen. In Abbildung 7.3 ist ein linearer Zusammenhang zwischen der Anzahl der Gaborfilter und der Adaptionszeit zu erkennen. Jedoch gilt diese Linearität nur unter gewissen Einschränkungen. Erstens ist zu berücksichtigen, dass mit zunehmender Dimensionaltät des Suchraums mehr Generationen notwendig werden, um das Optimum zu finden. Zweitens nimmt der Zeitbedarf für das wiederholte Klassifikatortraining superlinear mit der Merkmalszahl ($M = 2T$) zu und macht sich nur solange nicht bemerkbar, wie die Merkmalsextraktion länger dauert als das Klassifikatortraining.

Generell bleibt festzuhalten, dass der Aufwand für die Merkmalsoptimierung hoch ist. Ursache ist die sich wiederholende Berechnung von Merkmalen und Klassifikator für jede Parameterkombination, die Voraussetzung für die Ermittlung der Fitness (Gütekriterium) ist.

Kapitel 8

Vergleichende Untersuchungen

Die in den vorangegangenen Kapiteln entwickelten Verfahren sollen nun auf reale Texturen angewendet werden. Im Sinne einer vergleichenden Untersuchung wird dabei auf Bilddaten zurückgegriffen, die sich in der Literatur als Standards herausgebildet haben [POX00] [RH99]. Der Schwerpunkt des Abschnitt 8.1 liegt auf der rotationsinvarianten Klassifikation von Texturen. In Abschnitt 8.2 steht die Textursegmentierung mit Merkmalen des Typs LBP und HP2 im Vordergrund. Abschnitt 8.3 schließlich vergleicht die Leistungsfähigkeit von Merkmalsselektion und -optimierung.

8.1 Rotationsinvariante Texturklassifikation

Um Texturen unabhängig von ihrer Rotationslage zu erkennen, kommen die in Abschnitt 5.3 beschrieben Ansätze „Rotationsinvariante Merkmale“ und „Rotationsinvarianter Klassifikator“ in Frage. Hier wird der erste Ansatz mit dem Ziel verfolgt, die Rotationsinvarianz der Merkmale LBP-ROT und HP2-ROT zu untersuchen und zu vergleichen.

8.1.1 Konzeption der Experimente

Alle Experimente dieser Versuchsserie werden auf dem gleichen Datensatz durchgeführt. Die Modularität des Modularität des Systementwurfs erlaubt es, unterschiedliche Verfahren der Merkmalsberechnung, der Klassifikation sowie der Merkmalsselektion zu aktivieren. Für die Merkmalsberechnung kommen die Verfahren LBP-ROT (rotationsinvariante lokale binäre Muster) und HP2-ROT (rotationsinvariante histogrammbasierte Merkmale der 2-Punkt-Statistik) zum Einsatz. Für die Klassifikation werden Kreuzentropie-, Abstands- und Normalverteilungsklassifikatoren berücksichtigt. Die Konfigurationen der einzelnen Experimente sind in Tabelle 8.1 zusammengestellt.

x_2	x_3	x_4
x_1		x'_1
x'_4	x'_3	x'_2

Abbildung 8.1: Anordnung der 4 punktsymmetrischen Bildpunktpaare für das Merkmal SCOV.

In jedem Experiment werden für alle Bilder der Größe $F = 32 \times 32$ Pixel die Merkmale LBP-ROT bzw. HP2-ROT berechnet. Die Merkmalsvektoren der Trainingsbilder mit den zugehörigen Klassenkennungen dienen zur Adaption des Klassifikationssystems während die Merkmalsvektoren der Validierungsbilder zur Überprüfung des Systems herangezogen werden.

Experiment A und B geben Versuchskonfigurationen wieder, die von Pietikäinen *et al.* [POX00] verwendet wurden. Sie dienen als Referenzexperimente für die Versuchsserie und basieren auf der Grundversion der rotationsinvarianten lokalen binären Muster LBP-ROT 4.2 sowie Kreuzentropie-Klassifikation. In Experiment B wird neben den Merkmalen LBP-ROT das Histogramm des symmetrischen Kovarianzmaßes SCOV (Symmetric covariance measure) verwendet. Die Größe SCOV selbst berechnet sich wie die lokalen binären Muster aus einer 3×3 Umgebung und quantifiziert Korrelation und Kontrast der Umgebung:

$$\mathrm{SCOV} = \frac{1}{4} \sum_{i}^{4} (x_i - \mu)(x'_i - \mu) \tag{8.1}$$

μ steht für den mittleren Grauwert der 3×3 Umgebung. Die Anordnung der Pixelpaare x_i, x'_i ist in Abbildung 8.1 dargestellt.

Experiment C bildet den Ausgangspunkt der Experimentserie C–G, in der sukzessive Verbesserungen in der Merkmalsberechnung und Klassifikation eingeführt werden. Experiment C entspricht bis auf die Wahl des Klassifikators dem Referenzexperiment A. Damit besteht die Möglichkeit, die erzielten Ergebnisse der Experimentserie mit den Literaturwerten zu vergleichen.

Experiment D stellt eine erste Verbesserung gegenüber dem Experiment C dar. Hier wurde die Anzahl der Nachbarpixel und damit der Anzahl der Analyserichtungen von 8 auf 12 erhöht. Um die Anzahl der invarianten Muster zu begrenzen, wurde neben der Rotations- auch Spiegelinvarianz für die lokalen binären Muster gefordert. Die insgesamt $2^{12} = 4096$ möglichen binären Muster reduzieren sich damit auf 224 rotations- und spiegelinvariante Muster (vgl. Abschnitt 4.2.3).

Während in Experiment D die lokalen binären Muster aus der unmittelbaren Nachbarschaft ($D = 1$) berechnet werden, wird in Experiment E der

Experiment	Texturmerkmale	Zahl der Merkmale	Klassifikator	Bemerkung
A	LBP-ROT (8 Nachbarn, $D = 1$)	36	Kreuzentropie	Ergebnis von Pietikäinen [POX00]
B	LBP-ROT, SCOV (8 Nachbarn, $D = 1$)	68	Kreuzentropie	Bestes Ergebnis von Pietikäinen [POX00]
C	LBP-ROT (8 Nachbarn, $D = 1$)	36	Abstand	Äquivalentes Experiment zu A
D	LBP-ROT (12 Nachbarn, Spiegelinvarianz, $D = 1$)	224	Abstand	
E	LBP-ROT (12 Nachbarn, Spiegelinvarianz, $D = 1, 2, 4, 6$)	896	Abstand	
F	LBP-ROT (12 Nachbarn, Spiegelinvarianz, $D = 1, 2, 4, 6$, Randomisiert)	896	Abstand	
G	HP2-ROT (12 Analyserichtungen für $D = 1 - 6$, 8 Analyserichtungen für SUB0–SUB2, Randomisiert)	8 aus 270	Normalverteilung	Merkmalsselektion

Tabelle 8.1: Experimente zur Rotationsinvarianz

Nachbarschaftsbereich erweitert ($D = 1, 2, 4, 6$). Für jeden der vier Pixelpaarabstände D wird nun ein invariantes lokales binäres Muster errechnet, dessen Statistik in einem separaten Histogramm erfasst wird (vgl. Abschnitt 4.2.2).

Ziel von Experiment F ist die Verbesserung der Rotationsinvarianz gegenüber Experiment E. Durch Randomisierung der Orientierung der lokalen binären Muster wird vermieden, dass Vorzugsrichtungen existieren, die sich vor allem bei der Klassifikation von regelmäßigen Texturen negativ auswirken (vgl. Abschnitt 4.2.4).

Alle bislang beschriebenen Experimente beruhen auf der Erfassung der Statistik von lokalen binären Mustern in Form ihrer Histogramme und der anschließenden Klassifikation mit Hilfe eines Kreuzentropie- bzw. eines Abstandsklassifikators. In die Klassifikation fließen *alle* Merkmale ein, d.h. die Häufigkeiten aller lokalen binären Muster. Sowohl der Kreuzentropie- als auch der Abstandklassifikator führen eine implizite Gewichtung der Merkmale durch.

So gehen lokale binäre Muster hoher Auftrittswahrscheinlichkeit stärker in das Abstandsmaß ein als solche mit niedriger Auftrittswahrscheinlichkeit. Zu einem gewissen Grad ist dieser Effekt erwünscht, da der relative Schätzfehler mit sinkender Auftrittswahrscheinlichkeit zunimmt. Allerdings nützt dieser Ansatz nicht aus, dass die Diskriminierungsfähigkeit der Merkmale von den gegebenen Texturen abhängt. Eine gezielte Auswahl der unterscheidungsstärksten Merkmale bzw. das gezielte Weglassen der unterscheidungsschwächsten Merkmalen führt in der Regel (vgl. Abschnitt 6 zu niedrigeren Fehlerraten. Experiment G nützt diesen Effekt gezielt aus. Als Merkmale kommen die histogrammbasierten Merkmale der 2-Punkt-Statistik HP2-ROT zum Einsatz (vgl. Abschnitt 4.1). Die geeignetesten Merkmale werden mit dem Gütekriterium J_{Prob2} (vgl. Abschnitt 6.2.2) und dem Suchverfahren „Sequentielle Suche“ (vgl. Abschnitt 6.3.1) bestimmt. Die Klassifikation erfolgt mit einem Normalverteilungsklassifikator (vgl. Abschnitt 2.4).

8.1.2 Datensatz „Brodatz-Texturen“

Die verwendeten Texturen entstammen der Arbeit von Pietikäinen *et al.* [POX00] und sind elektronisch zugänglich. Der Datensatz leitet sich aus 15 Texturen des Brodatz-Albums [Bro66] ab, die in Abbildung 8.2 wiedergegeben sind. Es handelt sich um die Texturen gepresster Kork (D4), Rasen (D9), Fischgrätenmuster (D16), Wolltuch (D19), Leinwand (D21), Kalbsleder (D24), Sand (D29), gepresster Kork (D32), Strohmatte (D53), handgeschöpftes Papier (D57), Holzmaserung (D68), Baumwolltuch (D77), Raffiabast (D84), Schweinsleder (D92) und Kalbsfell (D93). Alle Bilder sind grauwertnormiert und liegen in 7 verschiedenen Rotationslagen ($11\,^\circ$, $11\,^\circ + 30\,^\circ$, $11\,^\circ + 60\,^\circ$, $11\,^\circ + 90\,^\circ$, $11\,^\circ + 120\,^\circ$, $11\,^\circ + 150\,^\circ$, $11\,^\circ + 200\,^\circ$) vor, wobei die Bildpunkte bikubisch interpoliert wurden. Aus diesen rotierten Bildern wurden jeweils 64 Subbilder der Größe 32×32 Pixeln ausgeschnitten, so dass insgesamt $15 \times 7 \times 64 = 6720$ Bilder zur Verfügung stehen. Die 960 Bilder der Rotationslage $11\,^\circ$ (Referenzrotationslage) dienen zum Training der Texturen, die 5760 Bilder aller anderen Rotationslagen zur Validierung. Auf diese Weise kann die Rotationsinvarianz der Merkmale LBP-ROT und HP2-ROT experimentell überprüft werden. Der Einfachheit wegen wird der konstante Winkelversatz von $11\,^\circ$ zur Kennzeichnung der verschiedenen Rotationslagen im Folgenden unterdrückt. In Abbildung 8.3 ist exemplarisch für jede Textur ein Subbild der Größe 32×32 Pixel abgebildet, um einen Eindruck vom Schwierigkeitsgrad der Klassifikationsaufgabe zu geben.

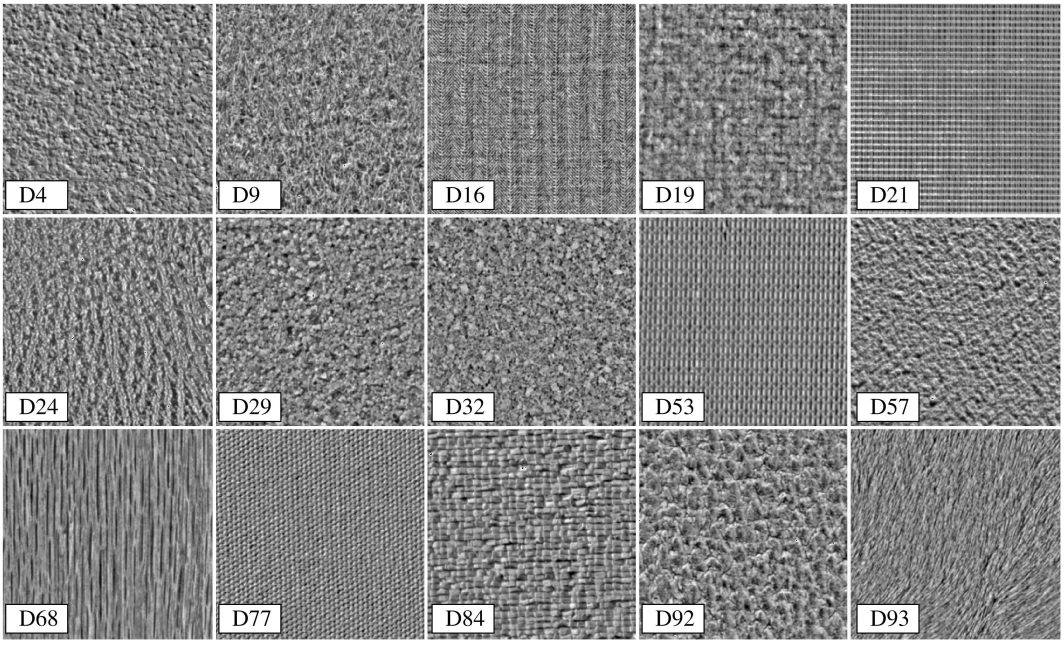


Abbildung 8.2: Originalbilder der Brodatztexturen, aus denen der Datensatz für die Experimente A–G gewonnen wurde.

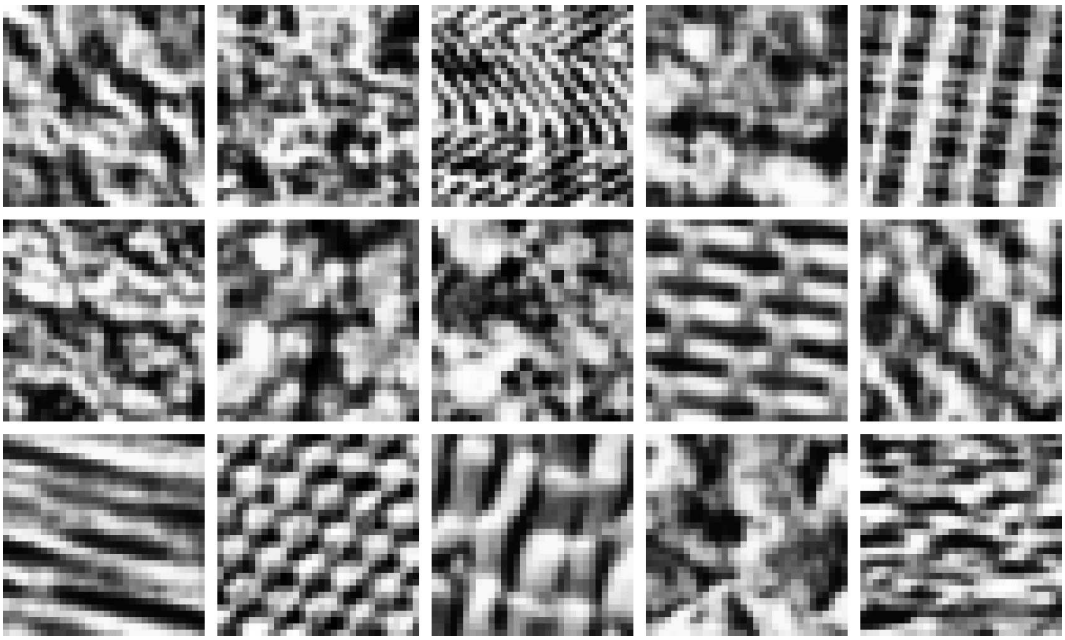


Abbildung 8.3: Subbilder der Größe 32×32 Pixel. Für jede der 15 Texturen ist ein Subbild in der Referenzrotationslage ($11\,^{\circ}$) dargestellt.

Experiment	Fehlerrate auf Testdaten (%)	Fehlerraten auf Validierungsdaten (%) (Rotierte Texturen)
A	Nicht verfügbar	47,7
B	Nicht verfügbar	24,0
C	37,7	41,3
D	35,2	39,5
E	14,7	22,5
F	16,7	19,7
G	5,7	7,7

Tabelle 8.2: Fehlerraten für Test- und Validierungsdaten.

8.1.3 Ergebnisse und Diskussion

Fehlerraten

Das wichtigste Kriterium zur Beurteilung der Experimente ist die Fehlerrate auf dem Test- und Validierungsdatensatz. Sie sind in Tabelle 8.2 für alle Versuche wiedergegeben. Die Werte für Experiment A und B wurden dem Artikel von Pietikäinen entnommen und belaufen sich auf 47,7 % bzw. 24,0 %. Testfehlerraten sind dort leider nicht angegeben.

Experiment C als Ausgangspunkt der Experimentserie C–G liefert gegenüber dem Referenzexperiment A eine um 6,4 % verringerte Fehlerrate. Über die Ursachen für die Differenz kann nur spekuliert werden. Möglicherweise ist das unterschiedliche Klassifikationsprinzip die wichtigste Einflussgröße.

Betrachtet man die Experimente C bis F, so fällt auf, dass die Fehlerrate von 41,3 % auf 19,7 % reduziert werden konnte. Jede der eingeführten Modifikationen gegenüber der Grundversion zieht eine Verbesserung nach sich. Der größte Sprung ergibt sich aus der Einführung eines erweiterten Nachbarschaftsbereichs (Multiskalenansatz) von Experiment D auf E mit einer Verringerung der Fehlerrate um 17,0 %. Aber auch die Erhöhung der Anzahl der Analyserichtungen (C → D) sowie die Randomisierung der Analyserichtungen (E → F) senken die Fehlerrate um 1,8 % bzw. 2,8 %.

Das beste Ergebnis wird jedoch durch ein alternatives Verarbeitungsschema in Experiment G erzielt, das auf histogrammbasierten Merkmalen der 2-Punkt-Statistik in Verbindung mit Merkmalsselektion beruht. Damit wird eine Fehlerrate von 7,7 % erreicht, also weniger als ein Drittel des Fehlers des besten Versuchs von Pietikäinen (Experiment B). Ebenso wie bei den Merkmalen LBP-ROT konnte diese drastische Reduktion der Fehlerrate gegenüber dem Referenzexperiment nur durch ein Bündel von Maßnahmen erreicht werden: Bei den lokalen binären Mustern zählen dazu die Erhöhung der Anzahl der Analyserichtungen, die Multiskalenanalyse sowie die Randomisierung der Analyserichtungen. Bei den histogrammbasierten Merkmalen der 2-Punkt-Statistik Merkmalen spielt darüber hinaus eine verbesserte Berechnung von Kenngrößen aus den Differenz- und Summenhistogrammen eine wichtige Rolle (vgl.

Abschnitt 4.1).

Rotationsinvarianz

Neben dem Ziel niedriger Fehlerraten auf den gesamten Validierungsdaten, liegt der Schwerpunkt dieser Untersuchungen darin, die Rotationsinvarianz der Merkmale LBP-ROT und HP2-ROT zu überprüfen. Außerdem soll analysiert werden, durch welche Maßnahmen die Rotationsinvarianz erhöht werden kann. Hierzu bieten sich die Erhöhung der Zahl der Analyserichtungen sowie die Randomisierungen der Analyserichtungen an. Die Grauwerte an etwaigen Zwischengitterplätzen werden bilinear interpoliert, um den Einfluss der Bildrasterung zu reduzieren, der sich bei Bilddrehung besonders bemerkbar macht.

Tabelle 8.3 und Abbildung 8.4 geben den Verlauf der Fehlerrate über der Rotationslage der Texturen wieder. Parallel zu den Gesamtfehlerraten aus Tabelle 8.2 sinken auch hier die Fehlerraten für die unterschiedlich stark rotierten Texturen von Experiment C zu Experiment G. Betrachtet man jedoch die Gleichmäßigkeit G der Fehlerrate, beispielsweise charakterisiert durch den Quotienten μ/σ aus Mittelwert und Standardabweichung der Fehlerrate, ergibt sich ein anderes Bild. Experiment E weist mit $G = 3{,}4$ eine wesentlich niedrigere Gleichmäßigkeit auf als die anderen Experimente. Insbesondere die Fehlerrate für die Rotationslage 200 ° sticht negativ hervor.

Um diese Beoabachtung richtig einzuordnen, betrachten wir nochmals die Versuchskonfiguration. Die Experimente D–F wurden alle mit 12 Analyserichtungen durchgeführt, was einer Schrittweite von 30 ° entspricht. Folglich fallen bis auf die Rotationslage 200 ° alle Rotationslagen mit einer der 12 Analyserichtung zusammen. Die Rotationslage 200 ° dagegen ist um 10 ° zur nächstgelegenen Analyserichtung versetzt und nimmt damit eine Sonderstellung ein.

Für Experiment D ergibt sich eine hohe Gleichmäßigkeit der Fehlerrate von $G = 14{,}6$. Als Ursache ist die hohe Zahl der Analyserichtungen (12) im Vergleich zur Zahl der Nachbarbildpunkte (8) anzusehen. Wird der Nachbarschaftsbereich wie in Experiment E erweitert, genügen die 12 Analyserichtungen nicht mehr, um die Nachbarschaft hinreichend dicht abzudecken, G sinkt auf 3,4. Für den Fall, dass Rotationslage und Analyserichtung nicht deckungsgleich sind (200 °) ergibt sich eine sprunghaft gesteigerte Fehlerrate von 35,9 %. Hier kommt die Bedeutung der Randomisierung der Analyserichtung ins Spiel. Sie bewirkt im Fall von LBP-ROT, dass die 12 Analyserichtungen nicht immer gleich ausgerichtet sind, sondern vielmehr für jede Realisierung um einen zufällig ermittelten Versatzwinkel gedreht sind (vgl. Abschnitt 4.1.5). Experiment F, das sich von Experiment E lediglich in der aktivierten Randomisierung unterscheidet, verfügt über eine drastisch verbesserte Gleichmäßigkeit der Fehlerrate von $G = 23{,}4$. Auch das letzte Experiment G mit den Merkmalen HP2-ROT weist eine hohe Gleichmäßigkeit von $G = 11{,}5$ auf, die ebenfalls auf Randomisierung der Merkmale zurückzuführen ist.

Experiment	30 °	60 °	90 °	120 °	150 °	200 °	μ	σ	μ/σ
C	41,8	41,3	39,2	39,9	42,3	43,5	41,3	1,6	26,2
D	41,5	39,5	34,7	40,8	38,5	42,1	39,5	2,7	14,6
E	20,4	20,6	17,8	20,2	19,9	35,9	22,5	6,7	3,4
F	18,9	19,1	19,7	20,4	19,1	21,0	19,7	0,84	23,4
G	7,8	7,9	7,1	8,7	7,9	6,8	7,7	0,67	11,5

Tabelle 8.3: Numerische Darstellung der Fehlerraten (%) für die Validierungsdaten (aller Texturen) über dem Rotationswinkel. μ und σ geben den Mittelwert bzw. die Standardabweichung der Fehlerrate über dem Rotationswinkel an. G quantifiziert die Gleichmäßigkeit der Klassifikation über dem Rotationswinkel.

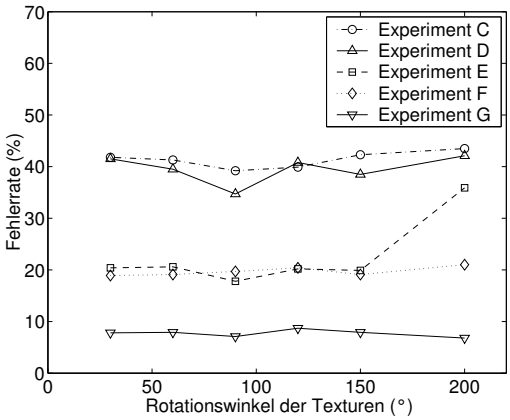


Abbildung 8.4: Grafische Darstellung der Fehlerraten für die Validierungsdaten über dem Rotationswinkel.

Als Fazit kann festgehalten werden, dass die intrinsische Rotationsinvarianz der Merkmale LBP-ROT erheblich gesteigert werden kann, wenn die Analyserichtungen randomisiert werden. Dies gilt umso mehr, je größer der Nachbarschaftsbereich ist. Außerdem kann die Randomisierung dazu beitragen, die Anzahl der Analyserichtungen zu begrenzen. Durch die Randomisierung lässt sich eine rechentechnisch billige Abdeckung des Nachbarschaftsbereichs realisieren, die anderenfalls durch eine weitere Erhöhung der Anzahl der Analyserichtungen mit wesentlich größerem Aufwand erkauft werden müsste.

Periodische Textur D021

Bei der Durchführung der Experimente zeigte sich, dass periodische Texturen in mancherlei Hinsicht ein anderes Verhalten aufweisen als die stochastisch geprägten Texturen. Aus dem Datensatz sei deshalb die Textur D021 herausgegriffen, die eine feine Gitterstruktur zeigt.

Wie bereits in Abschnitt 4.2 erwähnt, ist die Erscheinungsvielfalt periodischer Texturen klein im Vergleich zu stochastischen Texturen. Im Extremfall

Experiment	90 °	200 °	μ (über alle Rotationslagen)
C	43,8	71,9	64,6
D	26,6	76,6	63,8
E	0,0	60,9	10,2
F	0,0	0,0	0,0
G	0,0	0,0	0,2

Tabelle 8.4: Fehlerraten (%) für die Validierungsdaten der Textur D021.

einer rein periodischen Textur lässt sich diese durch Aneinanderreihung einer Elementarzelle darstellen. Werden an einer periodischen Textur Messungen durchgeführt – beispielsweise die Ermittlung von Auftrittswahrscheinlichkeiten der LBP-ROT – haben diese Messungen eine geringe Schwankungsbreite. In einer stochastischen Textur dagegen wird die Verteilung eine höhere Varianz aufweisen. Andererseits bedeutet dies, dass die Klassifikation einer periodischen Textur weitaus anfälliger gegenüber Störungen der Textur ist. Bereits Quantisierungseffekte, die mit einer Bilddrehung verbunden sind, können die Messergebnisse signifikant beeinflussen und die Klassifikation verfälschen.

In Tabelle 8.4 sind die Fehlerraten für die periodische Textur D021 für die Rotationslagen 90 ° und 200 ° sowie für die mittlere Fehlerrate über alle Rotationslagen angegeben. In Experiment D treten sehr große Fehlerraten für D021 auf. Insbesondere die Fehlerrate für die Rotationslage 200 ° mit 76,6 % und die mittlere Fehlerrate über alle Rotationslagen mit 63,8 % liegen deutlich über den entsprechenden Werten für alle Texturen mit 42,1 % bzw. 39,5 % (vgl. Tabelle 8.3). Durch die Einführung eines größeren Nachbarschaftsbereichs in Experiment E reduzieren sich die Fehlerraten erheblich. Ursache dafür ist die hohe Korrelation über große Distanzen in periodischen Texturen. Stochastische Texturen mit ihrer geringeren Korrelation über große Distanzen profitieren von dem größeren Nachbarschaftsbereich weniger stark: So fällt die über die Rotationslagen gemittelte Fehlerrate für D021 von 63,8 % auf 10,2 %, die entsprechende Fehlerrate für alle Texturen dagegen nur von 39,5 % auf 22,5 %. Auffallend ist weiterhin, dass in Experiment E der Fehler für die Rotationslage 200 ° mit 60,9 % sehr groß ist. Darin zeigt sich die oben erwähnte Anfälligkeit des Histogramms gegenüber Störungen (Rotation) der Textur. Erst die Randomisierung in Experiment F kann das Histogramm derart stabilisieren, dass der Fehler für die Textur D021 völlig verschwindet! Experiment G mit den Merkmalen HP2-ROT ist ebenfalls in der Lage, die Textur D021 nahezu fehlerfrei zu klassifizieren [PW03b].

Verwechslungsmatrix

Um abschließend einen Eindruck von der Klassifikationsgüte zu erhalten, ist in Tabelle 8.5 die Verwechslungsmatrix von Experiment G für die Validierungs-

Wahre Klasse	Prognostizierte Klasse D4	D9	D16	D19	D21	D24	D29	D32	D53	D57	D68	D77	D84	D92	D93
D4	361	5	0	0	0	13	0	0	0	5	0	0	0	0	0
D9	11	344	0	5	0	11	8	5	0	0	0	0	0	0	0
D16	0	0	384	0	0	0	0	0	0	0	0	0	0	0	0
D19	0	0	0	323	0	1	6	1	0	2	0	0	2	49	0
D21	0	0	0	0	383	0	0	0	0	0	0	0	0	0	1
D24	7	17	0	0	0	350	1	0	0	9	0	0	0	0	0
D29	14	1	0	13	0	0	314	13	0	0	0	0	5	24	0
D32	1	1	0	3	0	0	17	348	0	0	0	0	0	14	0
D53	0	0	0	0	0	0	0	0	384	0	0	0	0	0	0
D57	7	0	0	0	0	14	0	0	0	363	0	0	0	0	0
D68	0	0	0	0	0	0	0	0	0	0	384	0	0	0	0
D77	0	0	0	0	0	0	0	0	0	0	0	384	0	0	0
D84	0	0	0	6	0	0	23	0	0	0	0	0	355	0	0
D92	2	0	0	79	0	0	38	8	0	0	0	0	1	256	0
D93	0	0	0	0	0	0	0	0	0	0	0	0	0	0	384

Tabelle 8.5: Verwechslungsmatrix für Experiment G. Für jede Klasse standen 384 Subbilder zur Klassifikation bereit.

daten dargestellt. Ihr ist zu entnehmen, dass die 5 Texturen D16, D53, D68, D77 und D93 in 100 % der Fälle richtig klassifiziert wurden. Auf der anderen Seite sind die 3 Texturen D19, D29 und D92 für 58 % der Fehlklassifikationen verantwortlich. Betrachtet man diese 3 Texturen in Abbildung 8.2 und vor allem die exemplarischen Subbilder der Größe 32×32 Pixel in 8.3, so wird deutlich wie schwierig die Unterscheidung dieser 3 Texturen ist.

Fazit

Die Untersuchungen zur rotationsinvarianten Klassifikation von Brodatz-Texturen zielten in einer ersten Experimentserie (C–F) darauf ab, die Merkmale LBP-ROT zu optimieren. Dazu wurden sukzessive Verbesserungen eingeführt, mit denen die Fehlerrate auf 19,7 % reduziert werden konnte. Der in der Literatur dokumentierte Wert von 47,7 % des entsprechenden Referenzexperiments A konnte somit um mehr als die Hälfte unterschritten werden. Neben einer insgesamt niedrigen Fehlerrate bestand ein weiteres Optimierungskriterium in einer möglichst konstanten Fehlerrate für die verschiedenen Rotationslagen. Die realisierten Maßnahmen umfassen die Einführung eines erweiterten Nachbarschaftsbereichs (Multiskalenansatz), die Erhöhung der Anzahl der Analyserichtungen, die Spiegelinvarianz der Merkmale sowie die Randomisierung der Analyserichtungen.

Neben der Klassifikation mit lokalen binären Mustern wurde ein Verarbeitungsschema entwickelt, dessen Merkmale auf Histogrammen der 2-Punkt-Statistik beruhen (Experiment G). Weiteres Charakteristikum dieses Ansatzes ist die Selektion der effektivsten Merkmale auf Basis eines Normalverteilungsmodells. Die mit diesem Ansatz erzielte Fehlerrate von 7,7 % liegt um mehr als

zwei Drittel unter dem besten Literaturwert von 24,0 % (Referenzexperiment B).

Schließlich wurde eine Analyse des Klassifikationsverhaltens für die periodischen Textur D021 durchgeführt. Es zeigte sich, dass für die sichere Klassifikation ein erweiterter Nachbarschaftschaftsbereich unabdingbar ist, um die hohe Korrelation der periodischen Textur über größere Distanzen zu erfassen. Eine konstante Fehlerrate nahe 0 % für alle Rotationslagen konnte durch die Randomisierung der Analyserichtungen erreicht werden.

8.2 Textursegmentierung

Nachdem im Abschnitt 8.1 die Rotationsinvarianz der Merkmale LBP-ROT und HP2-ROT nachgewiesen wurde, soll nun eine Texturszene mit Hilfe dieser rotationsinvarianten Merkmale segmentiert werden. Zu Vergleichszwecken wird die gleiche Texturszene mit den rotationsabhängigen Merkmalen LBP und HP2 segmentiert. Aus den Ergebnissen lassen sich Hinweise für den Praxiseinsatz ableiten. Besonderes Augenmerk bei der Ergebnisanalyse wird auf die Robustheit der Merkmale für die Klassifikation von instationären Texturen gerichtet.

8.2.1 Konzeption der Experimente

Die Aufgabe besteht in der Segmentierung einer Bildszene, die 10 verschiedene Texturen umfasst. Jedes Experiment setzt sich aus zwei Schritten zusammen: Zunächst wird das Klassifikationssystem anhand von Trainingsbildern, die für jede Textur vorliegen, adaptiert. Anschließend findet eine pixelweise Klassifikation der Texturszene statt. Die Experimente unterscheiden sich in der Merkmalsberechnung, den Invarianzeigenschaften der Merkmale sowie in den Klassifikationsverfahren. Als Texturmerkmale kommen die bereits aus dem letzten Abschnitt bekannten lokalen binären Muster (LBP) und histogrammbasierten Merkmale der 2-Punkt-Statistik (HP2) zum Einsatz, sowohl in ihrer rotationsabhängigen (LBP, HP2) als auch in einer rotationsinvarianten Form (LBP-ROT, HP2-ROT). Die Merkmale vom Typ LBP/LBP-ROT werden stets mit einem Abstandsklassifikator klassifiziert (vgl. Abschnitt 4.2.6), die Merkmale vom Typ HP2/HP2-ROT dagegen mit neuronalen Netzen mit vorgeschalteter Merkmalsselektion, die auf einer Modellierung mit Normalverteilungen beruht. Die neuronalen Netze selbst sind als Backpropagation-Netze mit einer verborgenen Schicht ausgelegt. Eingangsgrößen für das neuronale Netz sind die extrahierten Merkmale, Ausgangsgrößen die prognostizierten Wahrscheinlichkeiten für die verschiedenen Texturklassen.

Ausgangspunkt der Versuchsserie sind Untersuchungen von Mäenpää *et al.* [MPO00], die hier als Referenzexperiment H geführt werden. Im Unterschied zu dem Referenzexperiment A aus dem letzten Abschnitt wird in Referenzexperiment H die rotationsabhängige Variante der lokalen binären Muster verwendet (LBP anstatt LBP-ROT).

Experiment I dient zum Abgleich unserer Verfahren mit dem in der Literatur dokumentierten Referenzexperiment. Unterschiede bestehen lediglich im Klassifikator (Abstandsklassifikator anstelle eines Kreuzentropie-Klassifikators) und in der Form des Analysefensters (Quadratisches Analysefenster der Größe 34×34 Pixel anstelle eines ähnlich großen, kreisförmigen Analysefensters mit 41 Pixeln Durchmesser).

Experiment	Texturmerkmale	Zahl der Merkmale	Klassifikator	Bemerkung
H	LBP (8 Nachbarn, $D = 1, 3, 5$, Kreisförmiges Fenster, mit $F = 41$)	768	Log-Likelihood	Bestes Ergebnis von Mäenpää [MPO00]
I	LBP (8 Nachbarn, $D = 1, 3, 5$, $F = 34$)	768	Abstand	Äquivalentes Experiment zu H (Originalbilder)
J	LBP (8 Nachbarn, $D = 1 - 6$, $F = 32$)	1536	Abstand	Ab hier doppelt rotierte Bilder
K	HP2 (4 Analyserichtungen, $D = 1 - 6$, $F = 32$, Randomisiert)	10 aus 456	Neuronal	Merkmalsselektion
L	LBP-ROT (12 Nachbarn, Spiegelinvarianz, $D = 1 - 6$, $F = 32$, Randomisiert)	1344	Abstand	
M	HP2-ROT (12 Analyserichtungen, $D = 1 - 6$, $F = 32$, Randomisiert)	10 aus 180	Neuronal	Merkmalsselektion

Tabelle 8.6: Experimente zur Segmentierung von Texturszenen

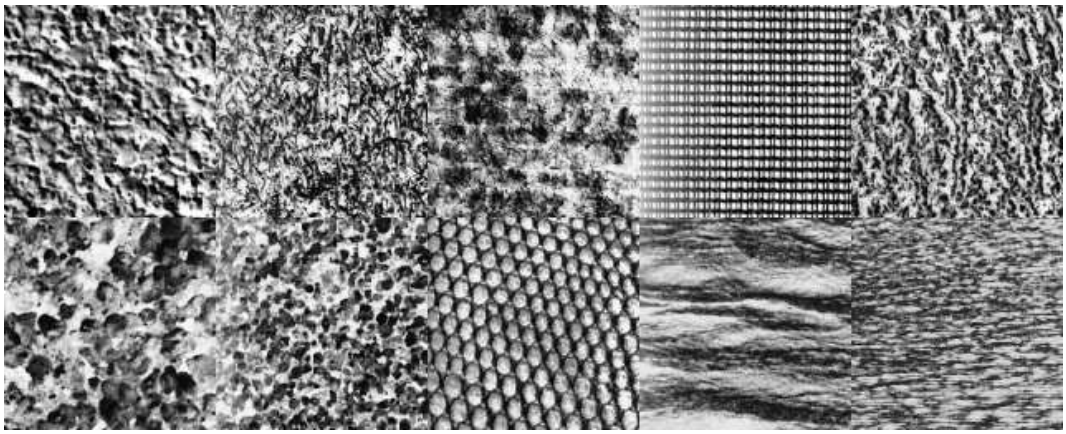


Abbildung 8.5: Texturszene „Mosaic08“ bestehend aus 10 Texturen.

Die Experimenten J–M vergleichen die Leistungsfähigkeit von rotationsabhängigen und rotationsinvarianten Merkmalen. Allen Experimenten ist gemeinsam, dass sie mit einem quadratischen Analysefenster der Größe 32×32 Pixel und mit Nachbarabständen $D = 1 - 6$ durchgeführt werden. Da in diesen Experimenten gedrehte Versionen der Texturszene in den Rotationslagen 0 °, 10 °, 20 °, 40 ° und 90 ° verwendet werden, ist darauf zu achten, dass alle Bilder den gleichen Vorverarbeitungsschritten unterworfen werden. Insbesondere kann eine Bilddrehung mit Interpolation eine Veränderung des Grauwerthistogramms nach sich ziehen. Aus diesem Grund wurden alle Bilder zweimal rotiert: Zunächst um 45 ° im Uhrzeigersinn und anschießend um 45 ° plus den Zielwinkel (0 °, 10 °, 20 °, 40 °, 90 °) gegen den Uhrzeigersinn. Das Training wurde immer auf den Trainingsbildern in Rotationslage 0 ° nach zweimaliger, sich kompensierender Drehung durchgeführt.

In Experiment J kommen die rotationsabhängigen Merkmale LBP in Kombination mit einem Abstandsklassifikator zum Einsatz. Ebenfalls rotationsabhängig sind die Merkmale HP2 in Experiment K. Hierbei werden aus dem Pool von 456 Merkmalen die 10 besten Merkmale selektiert und mit einem neuronalen Netz klassifiziert.

Im Gegensatz zu den Experimenten J und K werden in den Experimenten L und M die rotationsunabhängige Merkmale LBP-ROT bzw. HP2-ROT verwendet. Wie zuvor werden die Merkmale LBP mit einem Abstandsklassifikator weiterverarbeitet, die Merkmale HP2 dagegen mit einem neuronalen Netz mit vorgeschalteter Merkmalsselektion.

8.2.2 Datensatz „Mosaic08“

Der Datensatz „Mosaic08“ setzt sich aus einer Texturszene der Größe 640×256 und 10 separaten Trainingsbildern der Größe 256×256 zusammen. Die Daten der Texturszene und der Trainingsbilder stammen aus verschiedenen Bereichen der originalen Texturbilder und sind damit disjunkt. Randen und Husoy stellten diesen Datensatz, der unter [POX00] elektronisch verfügbar ist, ursprünglich zusammen [RH99].

Wie bereits erwähnt, wurden die Texturszene für die Experimente J bis M durch zweifache Drehung in unterschiedliche Rotationslagen (0 °, 10 °, 20 °, 40 °, 90 °) gebracht. Trainings- und Testdatensatz wurden aus den Trainingsbildern gewonnen, indem aus jedem der 10 Trainingstexturen die Merkmalsvektoren in 200 zufällig positionierten Analysefenstern berechnet wurden. Als Validierungsdatensatz dient die Texturszene in den unterschiedlichen Rotationslagen.

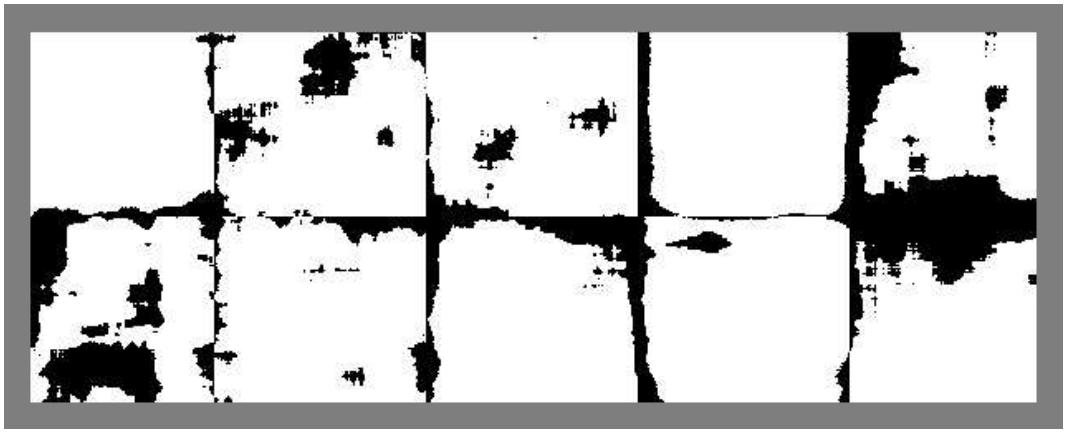


Abbildung 8.6: Fehlerbild der Texturszene „Mosaic08“ gemäß Experiment I. Weiß steht für korrekt klassifizierte Bildpunkte, Schwarz für falsch klassifizierte Bildpunkte, Grau für nicht klassifizierte Bildpunkte im Randbereich.

Experiment	Fehlerrate auf Testdaten (%)	Fehlerraten auf Validierungsdaten (%) (Nicht rotierte Texturszene)
H	Nicht verfügbar	17,6
I	3,1	16,8
J	7,1	20,0
K	2,4	25,5
L	19,5	36,9
M	5,1	30,5

Tabelle 8.7: Fehlerraten (%) für Test- und Validierungsdaten der nicht rotierten Texturszene.

8.2.3 Ergebnisse und Diskussion

Klassifikation der nicht-rotierten Texturen

In der Praxis auftretenden Klassifikationsaufgaben können sich darin unterscheiden, dass die zu untersuchenden Texturen entweder immer in der gleichen Rotationslage oder aber in beliebigen Rotationslagen auftreten. Im ersten Fall ist eine rotationsinvariante Klassifikation nicht sinnvoll, da durch die Erzeugung von Rotationsinvarianz stets Information verloren geht und höhere Fehlerraten in Kauf genommen werden müssen. Anwendungsfälle diesen Typs haben wir vor Augen, wenn in diesem Abschnitt die Fehlerraten von nicht rotierten Texturen analysiert werden.

Zunächst seien die Ergebnisse des Referenzexperiements H von Mäenpää *et al.* genannt, die auf der nicht rotierten Texturszene eine Fehlklassifikationsrate von 17,6 % erreichen. Implementierung I dieses Ausgangsexperiments ergibt ein leicht verbessertes Ergebnis von 16,8 %, dessen Fehlerbild in Abbildung 8.6 dargestellt ist. Damit ist ein Abgleich zu den Ergebnissen von Ojala und Mäen-

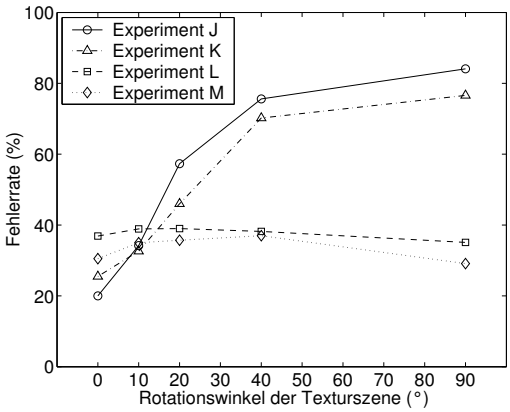


Abbildung 8.7: Grafische Darstellung der Fehlerraten bei Klassifikation der Texturszene in 5 verschiedenen Rotationslagen.

Experiment	0 °	10 °	20 °	40 °	90 °	μ
J	20,0	34,2	57,3	75,6	84,1	54,4
K	25,5	32,6	46,0	70,2	76,6	50,2
L	36,9	38,9	39,0	38,2	35,1	37,7
M	30,5	35,0	35,7	37,0	29,1	33,6

Tabelle 8.8: Numerische Darstellung der Fehlerraten (%) bei Klassifikation der Texturszene in 5 verschiedenen Rotationslagen.

pää hergestellt. Auf dieser Basis wird die Experimentserie J–M durchgeführt. Bedingt durch leicht veränderte Grundeinstellungen und die Verwendung von doppelt rotierten Bildern ergibt sich eine Erhöhung der Fehlerrate von Experiment I zu J um gut 3 Prozentpunkte auf 20,0 %.

Innerhalb der Experimentserie J–M ist erwartungsgemäß zu beobachten, dass die rotations*abhängigen* Merkmale gegenüber den rotations*invarianten* Merkmalen Vorteile bei der Klassifikation der nicht gedrehten Bilder aufweisen: Experiment J (LBP) liegt mit 20,0 % deutlich besser als Experiment L (LBP-ROT) mit 36,9 %. Ebenso übertrifft Experiment K (HP2) mit 25,5 % das Ergebnis von Experiment M (HP2-ROT) mit 30,5 %.

Beim Vergleich der Merkmals*typen* LBP/LBP-ROT und HP2/HP2-ROT ergibt sich ein weniger klares Bild. Während sich bei der rotationsabhängigen Klassifikation ein Vorteil für LBP gegenüber HP2 von 20,0 % zu 25,5 % ergibt, schneiden bei der rotationsinvarianten Klassifikation die Merkmale HP2-ROT mit 30,5 % besser ab als LBP-ROT mit 36,9 %.

Rotationsabhängige vs. rotationsinvariante Klassifikation

Nachdem im letzten Abschnitt die Klassifikation der *nicht-rotierten* Texturszene untersucht wurde, soll nun die Leistungsfähigkeit der verschiedenen Tex-

turmerkmale an der *gedrehten* Texturszene verglichen werden.

Tabelle 8.8 und Abbildung 8.7 geben den Verlauf der Fehlerrate für die Experimente J–M über dem Rotationswinkel der Texturszene wieder. Die Rotationslage 0 ° entspricht den Ergebnissen des letzten Abschnitts. Die Experimente J und K mit rotationsabhängigen Merkmalen weisen erwartungsgemäß steil ansteigende Fehlerraten auf, da die berechneten Texturmerkmale mit wachsendem Rotationswinkel immer stärker von den im Training berechneten Texturmodell abweichen. Ganz anders verlaufen dagegen die Kurven der Experimente mit rotationsinvarianten Merkmalen. Sie weisen relativ konstante Fehlerraten auf, die für kleine Winkel (0 ° und 10 °) allerdings über den Werten von Experiment J und K liegen. Ab einer Rotationslage von ungefähr 15 ° kommt die Rotationsinvarianz dagegen zum Tragen: Die Fehlerraten unterschreiten die Werte der Experimente J und K signifikant.

Während sich bei der Betrachtung der nicht-rotierten Texturszene im letzten Abschnitt noch keine klare Aussage treffen ließ, welcher Merkmalstyp leistungsstärker sei, ergibt sich hier ein eindeutiger Trend: Lediglich in einem einzigen Fall ist die Fehlerrate von LBP/LBP-ROT niedriger als bei dem korrespondierenden Experiment mit HP2/HP2-ROT, nämlich bei dem Experimentpaar J/K für 0 °. In allen anderen Fällen übertreffen die Merkmale HP2/HP2-ROT die Merkmale LBP/LBP-ROT in ihrer Leistungsfähigkeit (vgl. Tabelle 8.8).

Robustheit der Klassifikation

Klassifikationssysteme, die auf Verfahren der statistischen Mustererkennung beruhen, müssen stets auf ihre Robustheit analysiert werden:

- Welches sind geeignete Merkmale, um den Prozess robust zu beschreiben? (Vgl. Kapitel 6 und 7).
- Wie groß ist die Lernstichprobe zu wählen, um ein zuverlässiges Prozessmodell aufzubauen? (Vgl. Abschnitt 5.1).
- Ist sichergestellt, dass es sich um einen stationären Prozess handelt? (Vgl. Abschnitt 5.1).
- Sind Merkmale und Prozessmodell robust gegen eine Drift des Prozesses? Kann eine etwaige Instationarität innerhalb des Modells detektiert werden? (Vgl. Abschnitt 5.2).

Die Bedeutung dieser Fragestellungen wird offensichtlich, wenn die Ergebnisse der Experimente aus Abschnitt 8.1 und 8.2 verglichen werden. Der Übersichtlichkeit wegen sind die Daten nochmals in Tabellen 8.9 und 8.10 zusammengefasst. Als erstes sticht ins Auge, dass die Differenzen zwischen Test- und Validierungsfehler bei dem Datensatz „Brodatz“ (Experiment F und G) sehr

Datensatz	Experiment	Merkmale	Fehlerrate auf Testdaten (%)	Fehlerraten auf rotierten Validierungsdaten (%)
Brodatz	F	LBP-ROT	16,7	19,7
Brodatz	G	HP2-ROT	5,7	7,7
Mosaic08	L	LBP-ROT	19,5	37,7
Mosaic08	M	HP2-ROT	5,1	33,6

Tabelle 8.9: Gegenüberstellung der Fehlerraten für Test- und Validierungsdaten für die beiden Datensätze.

Datensatz	Experiment	Merkmale	Fehlerrate auf Testdaten (%)	Fehlerraten auf Validierungsdaten (%) (Nicht rot. Texturszene)
Mosaic08	J	LBP-ROT	7,1 (10,5)	20,0 (19,2)
Mosaic08	K	HP2-ROT	2,4 (6,6)	25,5 (28,4)
Mosaic08	L	LBP-ROT	19,5 (23,6)	36,9 (36,3)
Mosaic08	M	HP2-ROT	5,1 (7,9)	30,5 (35,8)

Tabelle 8.10: Gegenüberstellung der Fehlerraten für Test- und Validierungsdaten für die Experimentserie J–M. Die zuerst angegebenen Werte entsprechen dem Originalexperiment, die in Klammern angegebenen Werte dem modifizierten Experiment mit disjunkten Trainings- und Tesdaten.

viel geringer ausfallen als bei dem Datensatz „Mosaic08“ (Experiment L und M). Außerdem fällt auf, dass der Vorteil der Merkmale HP2-ROT in Kombination mit Merkmalsselektion gegenüber den Merkmalen LBP-ROT bei Datensatz „Mosaic08“ wesentlich kleiner ist als bei Datensatz „Brodatz“.

Zur Erklärung dieser Beobachtung kommen verschiedene Ursachen in Betracht (vgl. Abschnitt 5.1). Zum einen sind dies Schätzfehler der Modellparameter aufgrund einer zu kleinen Lernstichprobe für den Datensatz „Mosaic08“, zum anderen systematische Abweichungen zwischen Lernstichprobe und Validierungsdaten aufgrund von instationären Bilddaten.

Um eine Klärung herbeizuführen, wurde die Experimentserie J–M in modifizierter Form wiederholt. Während bisher die Trainings- und Testdaten aus den gesamten Trainingsbildern extrahiert wurden, wobei es zu Überlappungen zwischen diesen Datensätzen kommen konnte, werden nun die Trainingsdaten aus der oberen Bildhälfte, die Testdaten aus der unteren Bildhälfte der Trainingsbilder extrahiert. Auf diese Weise entstehen disjunkte Trainings- und Testdatensätze. Der Validierungsdatensatz (Texturszene) bleibt unverändert.

Tabelle 8.10 gibt die Ergebnisse der originalen Experimentserie J–M sowie der modifizierten Experimentserie wieder. Obwohl die Fehlerraten auf den Testdaten gegenüber den originalen Experimenten um einige Prozentpunkte angestiegen sind, liegen sie auf wesentlich niedrigerem Niveau als die entsprechenden Validierungsfehler. Da sowohl Test- als auch Validierungsdaten disjunkt zu den Trainingsdaten sind, lässt sich der Unterschied zwischen ihren Fehlerraten nicht durch eine unzuverlässige Schätzung der Modellparameter

begründen. Vielmehr ist die Ursache für die Diskrepanz zwischen Testfehler und Validierungsfehler in der Instationarität des Prozesses zu suchen, d.h. in der Unterschiedlichkeit der Muster in Trainings- und Validierungsdaten.

Eine Analyse im Detail offenbart weiter, dass die Fehlerrate auf den Validierungsdaten bei Experiment J und L nahezu konstant bleibt. Das bedeutet, dass die Veränderung der Trainingsdaten in der modifizierten Experimentserie keinen Einfluss auf die Größe des Schätzfehlers der Modellparameter hat. Als Ursache für die robuste Schätzung der Modellparameter ist sicher auch die Einfachheit des zugrundeliegenden Modells in Form eines Abstandsklassifikators anzusehen, in den lediglich die Mittelwerte der einzelnen Merkmale eingehen.

Experiment K und M weisen dagegen einen mäßigen Anstieg des Validierungsfehlers um 2,9 bzw. 5,5 Prozentpunkte auf. Dies ist ein Indiz für einen vergrößerten Schätzfehler der Modellparameter, die hier den Gewichten des neuronalen Netzes entsprechen. Offensichtlich reicht das reduzierte Datenmaterial in den Trainingsdaten (obere Hälfte der Trainingsbilder) für eine robuste Schätzung des komplexeren Klassifikators nicht mehr aus.

Der zu ziehende Schluss, die wesentliche Ursache für die stark voneinander abweichenden Test- und Validierungsfehler in einer Instationarität des Prozesses zu suchen, wird durch den visuellen Eindruck der Bilder bestätigt. Abbildung 8.8 zeigt die Teiltextur 10 im Trainingsbild und in der Texturszene. Insbesondere der linke obere Bereich des Trainingsbildes weicht in seiner Struktur von der Textur des Szenenausschnitts ab.

Als Ergebnis bleibt festzuhalten, dass die Merkmale HP2 mit Merkmalsselektion gegenüber den Merkmalen LBP von ihrem Vorsprung einbüßen, wenn instationäre Prozesse in Kombination mit kleinen Lernstichproben zu bearbeiten sind. Die Ursachen dafür lauten:

Merkmale: Die Merkmale LBP umfassen stets den vollen Merkmalssatz während von den Merkmalen HP2 nur die effektivsten Merkmale verwendet werden. Die LBP zielen also auf eine Beschreibung der Textur, während die Merkmale HP2 auf eine Unterscheidung der Texturen ausgerichtet sind. Bei instationären Prozesse oder unzuverlässig geschätzten Modellparametern verringert sich dadurch der Vorteil der Merkmale HP2.

Klassifikator: Der mit den LBP verwendete Abstandsklassifikator zeichnet sich durch seine Einfachheit aus, die geringe Schätzfehler nach sich zieht. In Kombination mit seiner Eigenschaft der impliziten Merkmalsgewichtung kann er bei instationären Prozessen oder kleinen Lernstichproben zu den komplexeren Klassifikatoren (Normalverteilungsklassifikator, neuronale Netze) aufschließen.

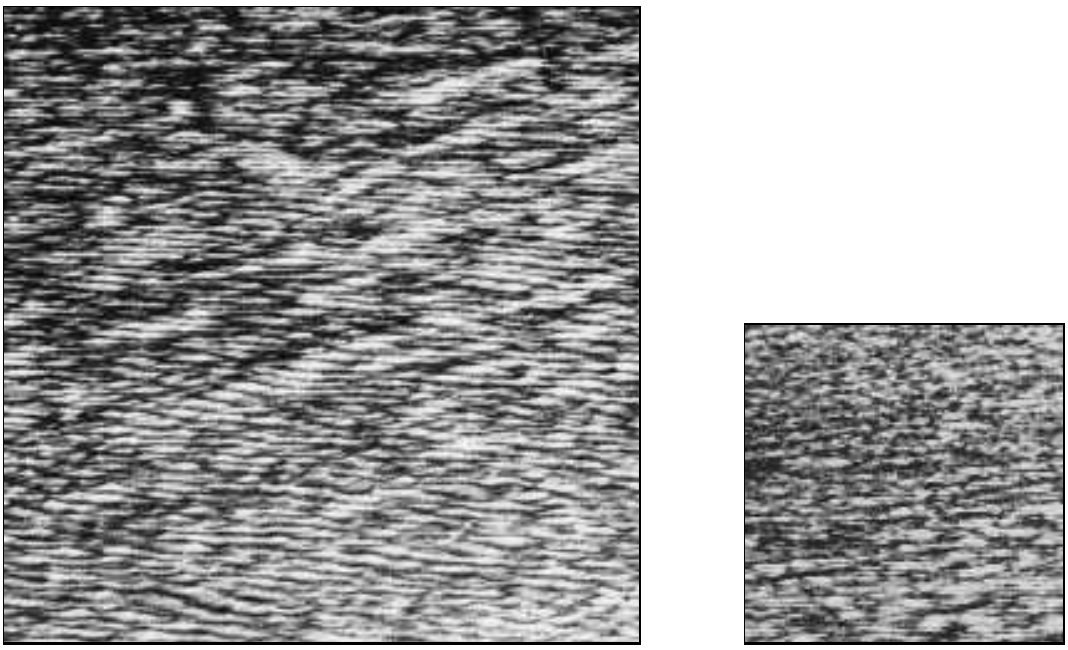


Abbildung 8.8: Textur 10: Trainingsbild und Ausschnitt aus Texturszene

Fazit

In den Untersuchungen an der Texturszene „Mosaic08“ zeigte sich, dass die niedrigsten Fehlerraten erzielt werden, wenn die Texturen in einer konstanten Rotationslage vorliegen. Die hierzu eingesetzten rotationsabhängigen Texturmerkmale haben allerdings einen eingeschränkten Anwendungsbereich: Weicht die Rotationslage von Trainings- und Zieltextur um mehr als 10 ° voneinander ab, überschreitet die steil ansteigende Fehlerkurve die Fehlerraten, die mit rotationsinvarianten Texturmerkmalen erreicht werden. Die Klassifikation mit rotationsinvarianten Merkmalen ist folglich der allgemeinere Ansatz. Mit ihm konnten Fehlerraten realisiert werden, die nahezu unabhängig von der Rotationslage der Textur sind. Es zeigte sich, dass sich mit den histogrammbasierten Merkmalen der 2-Punkt-Statistik HP2-ROT durchweg bessere Ergebnisse erzielen lassen als mit den lokalen binären Mustern LBP-ROT.

Bei instationären Texturen in Kombination mit kleinen Lernstichproben verringert sich der Vorteil der Merkmale HP2-ROT mit neuronaler Klassifikation gegenüber den Merkmalen LBP-ROT mit Abstandsklassifikation. Zwei Faktoren zeichnen dafür verantwortlich: Zum einen begünstigt eine kleine Lernstichprobe ein einfaches Prozessmodell (Abstandsklassifikator). Zum anderen birgt die Merkmalsselektion das Risiko in sich, mit einer kleinen Anzahl von Merkmalen, die unter nicht stationären Bedingungen ausgewählt wurden, weniger robust gegenüber Schwankungen des Prozesses zu sein. Als Ausweg bietet sich bei kleinen Lernstichproben und/oder instationären Texturen an, verein-

fachte Prozessmodelle (vgl. [HL96] [HFT96]) zu verwenden. Reizvoll erscheinen Forschungsanstrengungen mit dem Ziel, die Komplexität des Prozessmodells automatisch zu adaptieren.

8.3 Selektion vs. Optimierung

Verfahren der Merkmalsselektion kamen bereits in den vorangegangenen Abschnitten (Experimente G, K, M) zum Einsatz. Zielsetzung dieses Abschnitts ist der direkte Vergleich von Merkmalsselektion und Merkmalsoptimierung. Im ersten Fall werden die effektivsten Merkmale aus einer dyadischen Gaborfilterbank ausgewählt, während im zweiten Fall die Parameter der Gaborfilterbank optimiert werden.

8.3.1 Konzeption der Experimente

Ebenso wie in den Experimenten des Abschnitts 8.2 besteht die Aufgabe in der Klassifikation der Texturszene „Mosaic08“. In allen Experimenten werden Merkmale verwendet, die auf Gaborfiltern beruhen (vgl. Abschnitt 4.3). Als Klassifikator wird ein Normalverteilungsklassifikator eingesetzt. Lediglich das Referenzexperiment N von Randen basiert auf der lernenden Vektorquantisierung (LVQ).

Randen *et al.* [RH99] führte einen breit angelegten Vergleichstest gängiger Texturmerkmale an einer Reihe von Texturszenen durch. Als eine der schwierigsten Aufgaben stellte sich die Segmentierung der Szene „Mosaic08“ heraus, die dort als „11h“ bezeichnet wurde. Die niedrigste Fehlerrate konnte mit einer dyadischen Gaborfilterbank erreicht werden. Dieses Experiment (N) dient uns als Referenzexperiment.

Experiment O entspricht bis auf den Klassifikator dem Referenzexperiment. Weiterhin bestehen Detailunterschiede in der Nichtlinearität der Energiefunktion und in der Glättung (vgl. Abschnitt 4.3.1). Sowohl im Referenzexperiment N als auch in Experiment O werden mit Hilfe der 20 dyadisch angeordneten Gaborfiltern (5 Zentralfrequenzen $\times$ 4 Analyserichtungen) 20 Merkmale extrahiert.

In den Experimente P und Q werden die Merkmale mittels Merkmalsselektion bestimmt. Der Merkmalspool umfasst in Experiment P 12 Merkmale von 6 Filtern (3 Zentralfrequenzen $\times$ 2 Analyserichtungen), in Experiment Q 40 Merkmale von 20 Filtern (5 Zentralfrequenzen $\times$ 4 Analyserichtungen). In beiden Fällen wird neben dem Mittelwert der Filterantwort innerhalb des Analysefensters auch die Varianz als Merkmal berücksichtigt. Beide Experimente werden mehrfach durchgeführt, wobei die Anzahl der zu selektierenden Merkmale variiert wird.

Experiment R gibt die Konfiguration wieder, die als Ausgangspunkt für die Merkmalsoptimierung in Experiment S dient. In beiden Experimenten wird die Anzahl der Gaborfilter von 1 bis 6 variiert. Während in Experiment R die Filter aus der dyadischen Filterbank stammen und somit unveränderlich sind, werden in Experiment S für jeden Filter die Zentralfrequenz u_0, die Filterbreiten σ_x und σ_y sowie die Analyserichtung θ optimiert. Die CMA-Evolutionsstrategie

Experiment	Texturmerkmale	Zahl der Merkmale	Klassifikator	Bemerkung
N	Gabor (20 Filter, 4 Analyserichtungen)	20	LVQ	Bestes Ergebnis von Randen [RH99]
O	Gabor (20 Filter, 4 Analyserichtungen, $F = 32$)	20	Normalverteilung	Äquivalentes Experiment zu N
P	Gabor (6 Filter, 2 Analyserichtungen, $F = 32$)	1 - 12	Normalverteilung	Merkmalsselektion
Q	Gabor (20 Filter, 4 Analyserichtungen, $F = 32$)	1 - 40	Normalverteilung	Merkmalsselektion
R	Gabor (1–6 Filter, 2 Analyserichtungen, $F = 32$)	2 - 12	Normalverteilung	
S	Gabor (1–6 Filter, jeweils mit separater Analyserichtung, $F = 32$)	2 - 12	Normalverteilung	Merkmalsoptimierung

Tabelle 8.11: Experimente zum Vergleich von Merkmalsselektion und Merkmalsoptimierung

läuft jeweils über 30 Generationen à 10 Individuen. Die Schrittweite δ wird mit 0,1 initialisiert (vgl. Kapitel 7).

Die Anordnung der Gaborfilter entspricht in allen Experimenten der üblichen dyadischen Anordnung [JF91] mit den Zentralfrequenzen $\sqrt{2}/4$, $\sqrt{2}/8$, $\sqrt{2}/16$ usw. und den Analyserichtungen $0\,°$, $90\,°$ für die Experimente P, R und S (Initialisierung) bzw. $0\,°$, $45\,°$, $90\,°$, $135\,°$ für die Experimente N, O und Q. In Tabelle 8.11 sind alle Experimente kurz zusammengefasst.

8.3.2 Ergebnisse und Diskussion

Fehlerraten

In Tabelle 8.12 sind die jeweils besten Ergebnisse der Experimente zusammengefasst. Trotz der leichten Implementierungsunterschiede zwischen dem Referenzexperiment N und Experiment O entsprechen sich ihre Fehlerraten exakt. Bei den Experimenten P und Q mit Merkmalsselektion ist erwartungsgemäß festzustellen, dass die Qualität der Merkmale umso größer ist, je mehr Merk-

Experiment	Beste Einstellung	Fehlerrate auf Testdaten (%)	Fehlerraten auf Validierungsdaten (%)
N	-	Nicht verfügbar	32,3
O	-	2,1	32,3
P	11 von 12 Merkmalen	5,0	34,3
Q	4 von 40 Merkmalen	4,0	25,5
R	6 Filter (12 Merkmale)	4,8	35,0
S	3 Filter (6 Merkmale)	1,9	25,3

Tabelle 8.12: Fehlerraten für Test- und Validierungsdaten. In den Experimenten mit mehreren Varianten (P–S) wurde jeweils das Ergebnis für die beste Einstellung wiedergegeben.

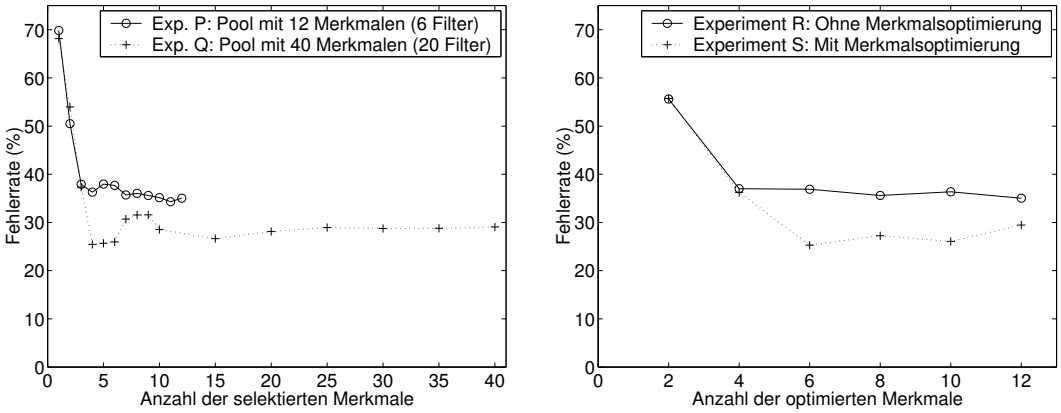


Abbildung 8.9: Verlauf des Validierungsfehlers für die verschiedenen Varianten der Experimente P bis S. Links: Experimente mit Merkmalsselektion, rechts: Experimente mit Merkmalsoptimierung.

male der Ausgangspool umfasst. Experiment R mit fixem Merkmalssatz liefert das schlechteste Ergebnis. Wird jedoch ausgehend von diesen Filterkonfigurationen die Merkmalsoptimierung dazugeschaltet, erhalten wir die niedrigste Fehlerrate von 25,3 % für Gabormerkmale. Noch niedriger liegt die Fehlerrate des Experiments I (Merkmale LBP) aus Abschnitt 8.2 mit 16,8 %, das jedoch mit Merkmalen des Typs LBP erzielt wurde. Offensichtlich sind lokalen binären Muster besser geeignet als Gabormerkmale, die spezifischen Charakteristika einer Textur zu erfassen.

Merkmalsselektion

Um Eigenschaften der Merkmalsselektion zu analysieren, betrachten wir den Verlauf des Validierungsfehlers über der Anzahl der selektierten Merkmale. Abbildung 8.9 links zeigt den Valdierungsfehler für Experiment P mit einem Merkmalspool der Größe $M = 12$ und für Experiment Q mit $M = 40$. Es fällt auf, dass die Fehlerraten von Experiment Q bis auf eine einzige Ausnahme un-

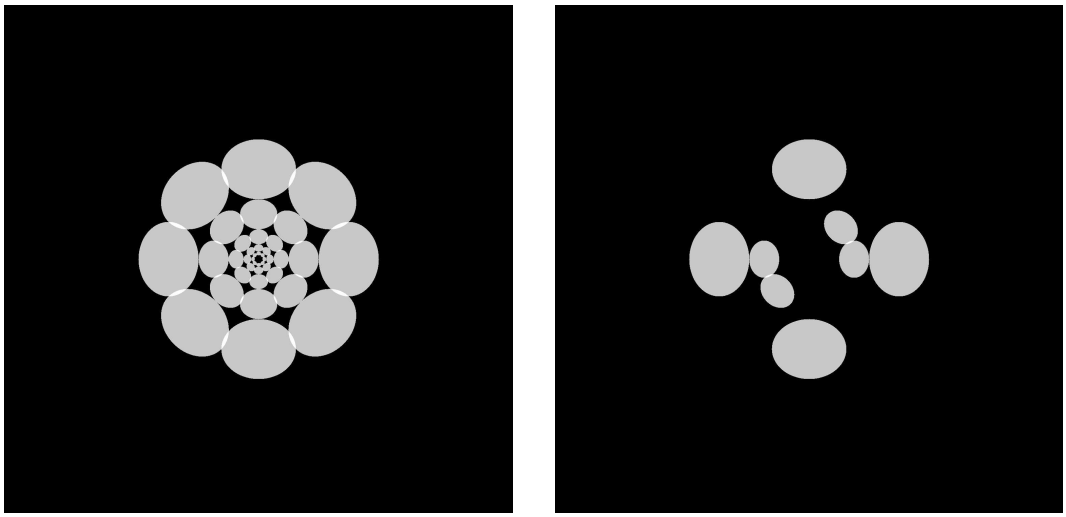


Abbildung 8.10: Darstellung der Gaborfilterbank im Frequenzraum vor und nach der Merkmalsselektion in Experiment Q. Aus $M = 40$ Merkmalen von 20 Filtern (5 Zentralfrequenzen $\times$ 4 Analyserichtungen) wurden die $m = 4$ Merkmale mit der höchsten Diskriminierungsfähigkeit ausgewählt. Der Ursprung der Frequenzebene liegt in der Bildmitte.

ter den Fehlerraten von Experiment P liegen. Das Potential der Merkmalsselektion ist folglich umso größer, je umfassender der Merkmalspool ist. Weiterhin ist zu erkennen, dass die minimale Fehlerrate in keinem der beiden Fälle bei dem vollen Merkmalssatz ($m = M$) liegt, sondern stets bei $m < M$. Der Verlauf von Experiment Q kann als typisch angesehen werden: Nach Durchlaufen eines Minimums folgt ein schwacher Anstieg der Fehlerrate. Dieser Anstieg ist auf den „Finite Sample Size Effect“ zurückzuführen (vgl. Abschnitt 6.1). Das beste Ergebnis in Experiment Q wird mit 4 selektierten Merkmalen ($m = 4$) erzielt und liegt 3,6 Prozentpunkte unter der Fehlerrate mit dem vollen Merkmalssatz ($m = M = 40$). Neben der niedrigeren Fehlerrate ist ein weiterer Vorteil der Merkmalsselektion in der wesentlichen Beschleunigung der Klassifikation zu sehen. Während die Zahl der Merkmale um eine Größenordnung von 40 auf 4 sinkt, beschleunigt sich die Klassifikation um den Faktor 10.

Der Zeitbedarf für die Merkmalsselektion ist gering. Um in Experiment Q die 4 besten Merkmale aus dem Pool von $M = 40$ Merkmalen zu bestimmen, werden lediglich 29,9 Sekunden auf einem Intel Pentium 4 Prozessor mit 2,4 GHz benötigt. Die Ausdünnung der Filterbank lässt sich gut im Frequenzraum visualisieren. Abbildung 8.10 zeigt die Filterbank vor und nach der Selektion. Offenbar verfügen für die gegebene Texturszene die Merkmale mit niedrigeren Zentralfrequenzen über keine große Diskriminierungsfähigkeit und fallen dementsprechend bei der Merkmalsselektion weg.

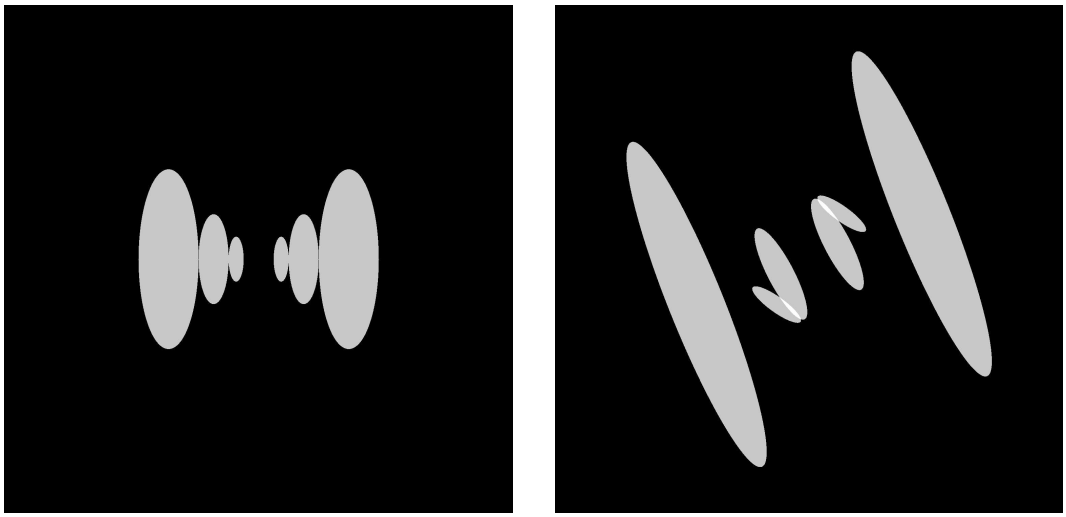


Abbildung 8.11: Darstellung der Gaborfilterbank im Frequenzraum vor und nach der Merkmalsoptimierung in Experiment S. Die Parameter von 3 Gaborfilter wurden optimiert.

Merkmalsoptimierung

Um das Potential der Merkmalsoptimierung abzuschätzen, vergleichen wir den Verlauf des Validierungsfehlers für die Experimente R und S über der Zahl der berücksichtigten Merkmale bzw. Filter. In Abbildung 8.9 ist zu sehen, dass der Validierungsfehler für Experiment S mit Merkmalsoptimierung immer gleichauf oder niedriger als beim Vergleichsexperiment R mit unveränderlichen Filtern liegt. Durch die Optimierung wird eine Absenkung des Fehlers um bis zu 11,6 Prozentpunkte erreicht. Diese Verbesserung ist umso bemerkenswerter, als Experiment R nicht mit einer beliebigen Filteranordnung durchgeführt wurde, sondern mit Filtern der als besonders geeignet geltenden dyadischen Anordnung. Noch bessere Ergebnisse lassen sich erzielen, wenn die Optimierung nicht mit der Anordnung aus Experiment R startet, sondern die Filterparameter zu Beginn der Optimierung randomisiert werden und mehrere Optimierungsläufe durchgeführt werden. Der Suchraum wird dann noch besser abgedeckt und die Wahrscheinlichkeit verringert, dass die Optimierung in einem lokalen Extremum hängen bleibt.

Als Detailunterschied zwischen der Merkmalsselektion und -optimierung sei erwähnt, dass in der vorliegenden Implementierung die von jedem Filter abgeleiteten Texturmerkmale „Mittelwert“ und „Varianz“ unterschiedlich stark gekoppelt sind. Bei der Selektion kann jedes dieser beiden Merkmale eines Texturfilters unabhängig vom anderen ausgewählt werden. Bei der Optimierung dagegen werden stets beide Merkmale verwendet. Die Optimierung ist in dieser Hinsicht implementierungsbedingt benachteiligt, weil das weniger effektive Merkmal anders als bei der Selektion nicht abgespalten werden kann.

Die Variante des Optimierungsexperiments S mit dem niedrigsten Validierungsfehler verwendete 3 Filter mit 6 Merkmalen. Für die Optimierung über 30 Generationen à 10 Individuen wurden 308 Sekunden auf einem Intel Pentium 4 Prozessor mit 2,4 GHz benötigt. Die durch die Optimierung bewirkte Veränderung der Filter ist in Abbildung 8.11 dargestellt. Interessant ist, dass ebenso wie bei der Selektion Filter bevorzugt werden, deren Ausrichtung in nordöstlicher Richtung verläuft.

Fazit

Als Ergebnis des Vergleichstests kann festgehalten werden, dass sowohl Merkmalsselektion als auch Merkmalsoptimierung geeignet sich, die Fehlerrate deutlich zu senken. Das Prinzip besteht in beiden Fällen darin, eine kleine Zahl von problemspezifischen Merkmalen zu generieren. Das beste Ergebnis lieferte die Merkmalsoptimierung mit 25,3 %, dicht gefolgt von der Merkmalsselektion mit 25,5 %. Wesentlich größer sind die Unterschiede bei dem Zeitbedarf für die Adaption. Während die Selektion nach 29,9 Sekunden abgeschlossen war, benötigte die Optimierung 308 Sekunden. Der höhere Zeitbedarf der Optimierung ist in der wiederholten Berechnung der Texturmerkmale für jede Parameterkonstellation begründet

Für Anwendungsfälle, in denen die Adaptionszeit eine wichtige Rolle spielt, ist folglich die Merkmalsselektion das zu bevorzugende Verfahren. Jedoch spielt die Optimierung ihre Vorteile umso besser aus, je mehr Parameter einzustellen sind. Aufgrund der kombinatorischen Explosion möglicher Parametereinstellungen ist eine hinreichend dichte Abdeckung des Parameterraums mit Verfahren der Merkmalsselektion dann nicht mehr möglich. Das Potential der Merkmalsoptimierung kann in Zukunft noch besser ausgeschöpft werden, wenn weitere Parameter wie die Größe des Analysefensters in die Optimierung einbezogen werden.

Kapitel 9

Anwendungsbeispiele

Auf Basis der vorgestellten Verfahren zur Oberflächenbewertung wurden Prüfsysteme für den industriellen Einsatz entwickelt. Anhand von drei Beispielen sollen Aspekte beleuchtet werden, die für eine Vielzahl von Anwendungen relevant sind. Dazu zählen die Validierung des Prüfsystems, die Prognose von kontinuierlichen Oberflächenparametern sowie die schnelle Adaption des Prüfsystems an neue Prüfaufgaben.

9.1 Medizinische Kollagenschwämme

9.1.1 Prüfaufgabe

Medizinische Kollagenschwämme werden eingesetzt, um den Heilungsprozess von inneren Wunden – wie sie etwa bei Operationen auftreten – zu beschleunigen. Um optimale Wirkung zu entfalten, müssen die Schwammstreifen über eine wohldefinierte Oberfläche verfügen. So gelten beispielsweise für die Porengröße und die Gleichmäßigkeit der Porenverteilung enge Toleranzbänder. Ebenso sind lokale Unregelmäßigkeiten in der Porenverteilung, die als „Blumen“ bezeichnet werden, unerwünscht. Verschiedene Ausprägungen von Kollagenschwämmen sind in Abbildung 9.1 dargestellt.

Ziel der objektiven und reproduzierbaren Erfassung der Oberflächeneigenschaften ist die Qualitätssicherung und Prozessverbesserung [PM00]. Im Einzelnen werden dazu die folgenden Größen ermittelt:

- Porengröße auf einer Skale von 1–5
- Gleichmäßigkeit der Porengrößenverteilung auf einer Skala von 1–5
- Detektion von lokalen Unregelmäßigkeiten („Blumen“)
- Detekion von Löchern

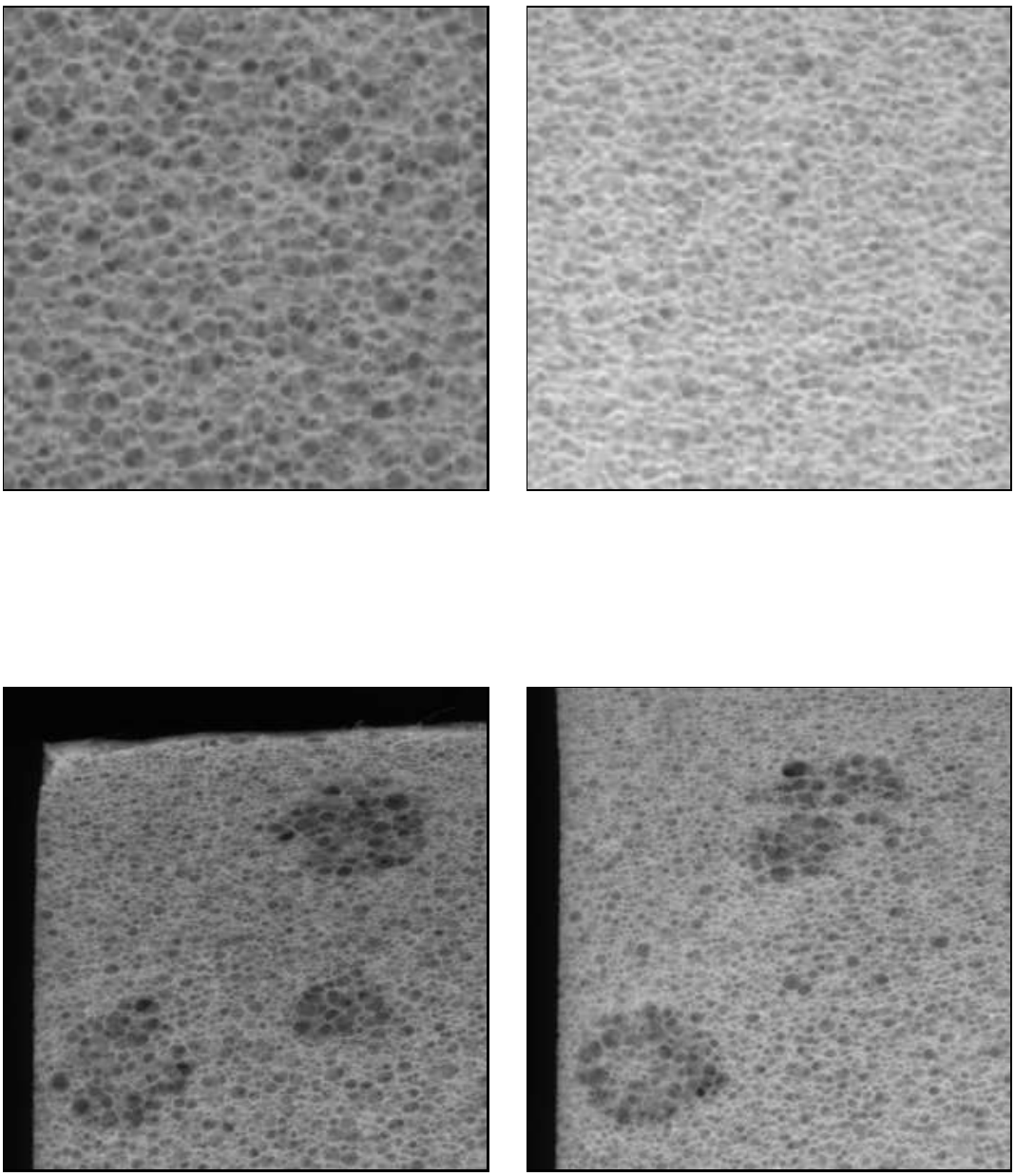


Abbildung 9.1: Kollagenschwämme mit grober (oben links) und feiner (oben rechts) Struktur sowie mit Strukturdefekten (unten), sog. „Blumen“.

9.1.2 Realisierung

Während der Kollagenschwammstreifen auf einem Schlitten unter der Prüfstation hindurchgeführt wird, zeichnet eine Zeilenkamera seine Oberflächenstruktur auf (vgl. Abbildung 9.2).

Die digitale Verarbeitung des aufgenommenen Bildes beginnt mit der Rasterung des Bildes in sich überlappende Zellen der Größe 40×40 Pixel, aus denen u. a. rotationsinvariante Merkmale der 2-Punkt-Statistik (HP2-ROT) extrahiert werden. Aus diesen primären Merkmalen wird mit Hilfe eines neuronalen Klassifikators für jede Zelle die lokale Porengröße geschätzt. Daraus lassen sich die gewünschten Messgrößen „Porengröße“ und „Gleichmäßigkeit der Porengrößenverteilung“ für den Kollagenschwamm ableiten.

Die Detektion lokaler Defekte wie der „Blumen“ erfordert ein zweistufiges Vorgehen. Die Ursache dafür ist, dass die Betrachtung einer isolierten Rasterzelle für die Detektion einer Blume nicht ausreicht. Eine Blume als lokale Störung der Textur kann erst in ihrem Kontext erkannt werden. Deshalb werden aus der bereits berechneten lokalen Porengröße Merkmale extrahiert, die die Eigenschaften der zentralen Rasterzelle im Verhältnis zu ihrer weiteren Umgebung charakterisiert. Diese sekundären Merkmale werden schließlich mit einem neuronalen Klassifikator auf das Interval $[0, 1]$ abgebildet, wobei 0 für „Keine Blume“ und 1 für „Blume“ stehen. Die endgültige Entscheidung, ob eine Rasterzelle die Eigenschaft „Blume“ trägt, wird mittels eines Schwellwertverfahrens mit Hysterese getroffen [Can86]. Abbildung 9.3 zeigt, wie das Prüfergebnis auf einem Prozessmonitor visualisiert wird. Für statistische Analysen werden die Prüfergebnisse chargenweise abgespeichert.

9.1.3 Validierung des Prüfsystems

Voraussetzung für die Inbetriebnahme des Prüfsystems – insbesondere um die Zulassung des medizinischen Produkts für den US-amerikanischen Markt zu erhalten – war eine umfassende Validierung der Verfahren. Validierung bedeutet stets, einen Abgleich zwischen Soll- und Ist-Verhalten durchzuführen. In diesem Zusammenhang sei auf Abschnitt 2.2 verwiesen, in dem das Soll-Verhalten auf 2 Arten definiert wird. Im ersten Fall ist das Soll-Verhalten durch eine algorithmische Definition gegeben. Validierung heißt dann, die Anwendung des Algorithmus sicherzustellen. Im zweiten Fall definiert sich das Soll-Verhalten über Bewertungsbeispiele. Eine Validierung besteht darin, das Interpolationsverhalten auf einem unabhängigen Validierungsdatensatz zu überprüfen. Die vorliegende Prüfaufgabe fällt in die zweite Kategorie mit der Besonderheit, dass die Bewertungsvorgaben mit einem Bewertungsfehler behaftet sind. Im Folgenden wird ein Validierungsschema vorgeschlagen, dass den Bewertungsfehler des Prüfsystems mit dem Bewertungsfehler der Bewertungsvorgaben in Relation setzt.

Abbildung 9.2: Medizinischer Kollagenschwamm bei der Prüfung seiner Oberflächenstruktur.

Abbildung 9.3: Visualisierung des Prüfergebnisses eines medizinischen Kollagenschwamms. Auf der rechten Seite wurde eine fehlerhafte Struktur detektiert.

Der Trainingsdatensatz setzt sich aus einer repräsentativen Stichprobe von N_T Objekten zusammen, die von P Prüfern unabhängig bewertet werden. Anhand dieses Trainingsdatensatzes wird das Prüfsystem adaptiert. Ein weiterer Datensatz dient zur Validierung des Systems. Er setzt sich aus N_V Objekten zusammen, die wiederum von allen Prüfern unabhängig bewertet werden. Darüber hinaus wird der Validierungsdatensatz auch von dem adaptierten Messsystem klassifiziert.

Um die Bewertungsfehler von Mensch und Maschine zu quantifizieren, wird in einem ersten Schritt der Bewertungsfehler innerhalb des menschlichen Validierungsdatensatzes bestimmt. Dazu werden die Bewertungen x_k, $k = 1 \ldots N_\mathrm{V}$, zwischen jeweils zwei Prüfern verglichen. Das Verifikationsmaß $J_{i,j}$ zwischen den Prüfern i und j ergibt sich als mittlere quadratische Abweichung über dem Validierungsdatensatz zu

$$J_{i,j} = \frac{1}{N_\mathrm{V}} \sum_{k=1}^{N_\mathrm{V}} a_{i,j,k} \tag{9.1}$$

Mit:

$$a_{i,j,k} = (x_{i,k} - x_{j,k})^2$$

Das Verifikationsmaß $J_{i,j}$ entspricht damit dem Gütekriterium J_Kont für die kontinuierliche Bewertung einer skalaren Größe ($L = 1$), vgl. Gleichungen 6.11 und 6.16. Neben dem mittleren quadratischen Fehler ist von Interesse, wie stark der Fehler über den Validierungsdatensatz streut. Als Maßzahl für die Gleichmäßigkeit der Bewertungsqualität bietet sich die Standardabweichung $S_{i,j}$ des quadratischen Fehlers zwischen den Prüfern i und j an:

$$S_{i,j}^2 = \frac{1}{N_\mathrm{V}} \sum_{k=1}^{N_\mathrm{V}} (a_{i,j,k} - J_{i,j})^2,$$

In einem zweiten Schritt werden nun die Abweichungen zwischen den Bewertungen der Prüfer und den maschinellen Bewertungen analysiert. Dies geschieht analog zu dem oben beschriebenen Vergleich zwischen den Prüfern.

Porengröße und -verteilung

Das Validierungsverfahren wird nun auf die Bewertung der Porengröße und -verteilung angewendet. Eingangsgrößen sind der von drei Prüfern bewertete Validierungsdatensatz ($V1$, $V2$, $V3$) sowie die Ergebnisse der maschinellen Prüfung (M). Die Ergebnisse sind in Tabelle 9.1 und 9.2 dargestellt. Es fällt auf, dass die Abweichungen zwischen den geschulten Prüfern deutlich höher ausfallen als die Abweichungen zwischen den Prüfern und den maschinellen Prüfergebnissen. Für die Porengröße ergibt sich ein Vorteil von 0,206 zu 0,360

Vergleichspaare	J(Porengröße)	S(Porengröße)
$V1$-$V2$	0,414	0,703
$V1$-$V3$	0,364	0,530
$V2$-$V3$	0,303	0,432
Mittelung über den Vergleich zwischen den Prüfer	**0,360**	0,555
$V1$-M	0,242	0,437
$V2$-M	0,224	0,305
$V3$-M	0,153	0,234
Mittelung über den Vergleich zwischen Prüfern und Maschine	**0,206**	0,325

Tabelle 9.1: Vergleich der Abweichungen für die Bewertung der Porengröße sowohl zwischen den Prüfern als auch zwischen Prüfern und Maschine.

Vergleichspaare	J(Porenverteilung)	S(Porenverteilung)
$V1$-$V2$	0,739	1,142
$V1$-$V3$	1,329	2,032
$V2$-$V3$	0,663	0,770
Mittelung über den Vergleich zwischen den Prüfern	**0,910**	1,315
$V1$-M	0,550	0,762
$V2$-M	0,293	0,467
$V3$-M	0,477	0,728
Mittelung über den Vergleich zwischen Prüfern und Maschine	**0,440**	0,652

Tabelle 9.2: Vergleich der Abweichungen für die Bewertung der Porenverteilung sowohl zwischen den Prüfern als auch zwischen Prüfern und Maschine.

für die maschinelle Prüfung, für die Porenverteilung sogar Werte von 0,440 zu 0,910. Daraus kann gefolgert werden, dass die Adaption des Prüfgeräts anhand von streuenden Bewertungsvorgaben zu einem interpolierenden Verhalten geführt hat. Das Prüfgerät ahmt auf diese Weise einen „gemittelten“ Prüfer nach.

Blumen

Während im letzten Abschnitt die Validierung von kontinuierlichen Messgrößen wie der Porengröße und -verteilung behandelt wurde, steht nun mit der Blumendetektion die Validierung einer diskreten Messgröße an. Jeder Rasterzelle ist ein Messergebnis zugeordnet, wobei $x_{i,k} = 0$ bedeutet, dass Prüfer i in der k-ten Rasterzelle keine Blume detektiert hat. $x_{i,k} = 1$ steht für die Detektion einer Blume. Das Verifikationsmaß $J_{i,j}$ ergibt sich als Mittelung über die Kosten von Fehlklassifikationen, wie sie in einer Kostenmatrix C_{ij} abgelegt sind:

Vergleichspaare	J(Blumen)	Fehler der 1. Art	Fehler der 2. Art
$V1$-$V2$	1748	779	969
$V1$-$V3$	1553	694	859
$V2$-$V3$	1464	745	719
Mittelung über den Vergleich zwischen den Prüfern	**1588**		
$V123$-M	**1228**	644	584

Tabelle 9.3: Vergleich der Abweichungen für die Blumendetektion sowohl zwischen den Prüfern als auch zwischen Prüfern und Maschine.

$$J_{i,j} = \frac{1}{N_\mathrm{V}} \sum_{k=1}^{N_\mathrm{V}} C_{x_{i,k};x_{j,k}} \tag{9.2}$$

Im einfachsten Fall werden abweichende Klassifikationsergebnisse gleich gewichtet. Diagonalelemente von C entsprechen übereinstimmenden Klassifikationsergebnissen und sind kostenfrei:

$$C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{9.3}$$

Dieses Schema wurde für die Validierung der Blumendetektion eingesetzt. Wie im letzten Abschnitt existieren wieder ein Trainings- und Validierungsdatensatz, der durch jeden der Prüfer bewertet wurde. Das Training bzw. die Validierung des Prüfsystems wurde hier jedoch nicht mit den 3 unterschiedlich bewerteten Varianten des Traininsdatensatzes ($T1$, $T2$, $T3$) bzw. des Validierungsdatensatzes ($V1$, $V2$, $V3$) durchgeführt, sondern mit den gemittelten Bewertungen der 3 Prüfer. Diese gemittelten Datensätze seien mit $T123$ und $V123$ bezeichnet. In Tabelle 9.3 sind die Ergebnisse der Validierung für die Blumendetektion zusammengefasst. Wiederum ist festzustellen, dass die Abweichungen zwischen den Prüfern mit 1588 fehlklassifizierten Rasterzellen höher ausfallen als die Abweichungen zwischen den Prüfern und dem adaptierten Prüfsystem mit 1228. Bei diesem Vergleich ist allerdings zu berücksichtigen, dass Rasterzellen aus dem Randbereich der Kollagenschwämme bei der maschinellen Bewertung unberücksichtigt bleiben.

Eine Alternative zu dem hier verwendeten Verifikationsmaß sind die in Abschnitt 6.2.2 diskutierten Abstandsmaße. Sie weisen eine Reihe von Vorteile auf, deren wichtigster darin besteht, eine übertriebene Vorhersage der wahrscheinlicheren Klasse zu bestrafen (vgl. [Bri50]). Voraussetzung für ihren Einsatz ist jedoch eine Bewertung der Aussage „Blume vorhanden“ auf der kontinuierlichen Skala von 0 bis 1 durch die Prüfer, die hier nicht verfügbar war.

9.1.4 Fazit

Die Validierung des Kollagenschwammprüfsystems ergab für alle 3 Prüfaufgaben stabilere Ergebnisse, als sie durch menschliche Prüfer erzielt werden konnten. Interessanterweise liegt die Ursache dieses Phänomens in der schwankenden Bewertung durch die Prüfer. Das Prüfsystem adaptiert sich auf ein gemitteltes Prüfverhalten und interpoliert auf diese Weise das Erfahrungswissen aller Prüfer. Folglich ist der Bewertungsfehler zwischen den Prüfern ist höher als zwischen der Maschine und den Prüfern. Im Falle identischen Prüfverhaltens aller Prüfer könnte das maschinelle System im günstigsten Fall die einheitliche Bewertung reproduzieren, jedoch nie übertreffen.

Die Vorteile eines derart adaptierten Prüfsystems bestehen in

- niedrigeren Schwankungen in den Bewertungen,
- zeitlicher Konstanz des Bewertungsverhaltens,
- Formalisierung des Bewertungsverhaltens; Unabhängigkeit von Prüfern, die geschult werden müssen.

9.2 Kontinuierliche Bewertung

Viele Prüfaufgaben der Texturanalyse haben zum Ziel, einen Texturausschnitt in eine von L diskreten Texturklassen ω_i einzuordnen. Beispiele dafür waren die Untersuchungen an Brodatz-Texturen in Kapitel 8. Im letzten Abschnitt wurde eine weitere Art der Klassifikation vorgestellt, nämlich die Zuordnung einer kontinuierlichen Größe y (Porengröße, Porenverteilung) zu einem Texturausschnitt. Diese Aufgabenstellung soll nun noch detaillierter untersucht werden.

Charakteristisch für die Klassifikation in diskrete Texturklassen ist der Vorgang der Klassenbildung. Unter „Klasse“ verstehen wir das Aggregat aller Trainingsmustern mit der gleichen Klassenkennung. Der von den Trainingsvektoren x eingenommene Bereich im Merkmalsraum muss dazu beschrieben werden, insbesondere müssen die klassenspezifischen Wahrscheinlichkeitsdichten bzw. die Grenzen zwischen den verschiedenen Klassen festgelegt werden. Für diese Modellierung kann unter einer Vielzahl von statistischen Klassifikatoren ausgewählt werden, u.a. können Normalverteilungsklassifikatoren, Nächste-Nachbar-Klassifikatoren, neuronale Netze oder Support Vector Machines eingesetzt werden [Fuk90] [CST00]. Zwischen den einzelnen Klassen, die von außen vorgegeben werden, besteht dabei keine Ordnungsrelation.

Das Problem der Zuordnung einer kontinuierlichen Größe y zu einem Texturausschnitt unterscheidet sich dadurch, dass die Trainingsmuster nicht mehr in Klassen zusammengefasst werden. Damit entfallen die mit den Klassen verbundenen Wahrscheinlichkeitsdichten bzw. die Trenngrenzen im Merkmalsraum. Vielmehr gilt es, eine interpolierende Funktion für die als Stützstellen fungierenden Trainingsvektoren zu finden. Dies kann beispielsweise durch neuronale Netze oder Nächste-Nachbar-Klassifikatoren realisiert werden.

Mit anderen Worten, die Aufgabe der *Klassifikation* besteht in der Abbildung des Merkmalsvektors x auf den Vektor der klassenspezifischen Wahrscheinlichkeiten p_i. Diese lassen sich aus den Verteilungen der Trainingsvektoren der jeweiligen Klasse ω_i im Merkmalsraum ableiten.

$$x = (x_1, x_2, \ldots, x_M)^T \longrightarrow p = (p_1, p_2, \ldots, p_L)^T, \tag{9.4}$$

Aufgabe der *Funktionenapproximation* ist dagegen die Abbildung des Merkmalsvektors x auf eine Zielgröße y, definiert durch die Trainingsvektoren als Stützstellen.

$$x = (x_1, x_2, \ldots, x_M)^T \longrightarrow y = (y_1, y_2, \ldots, y_L)^T, \tag{9.5}$$

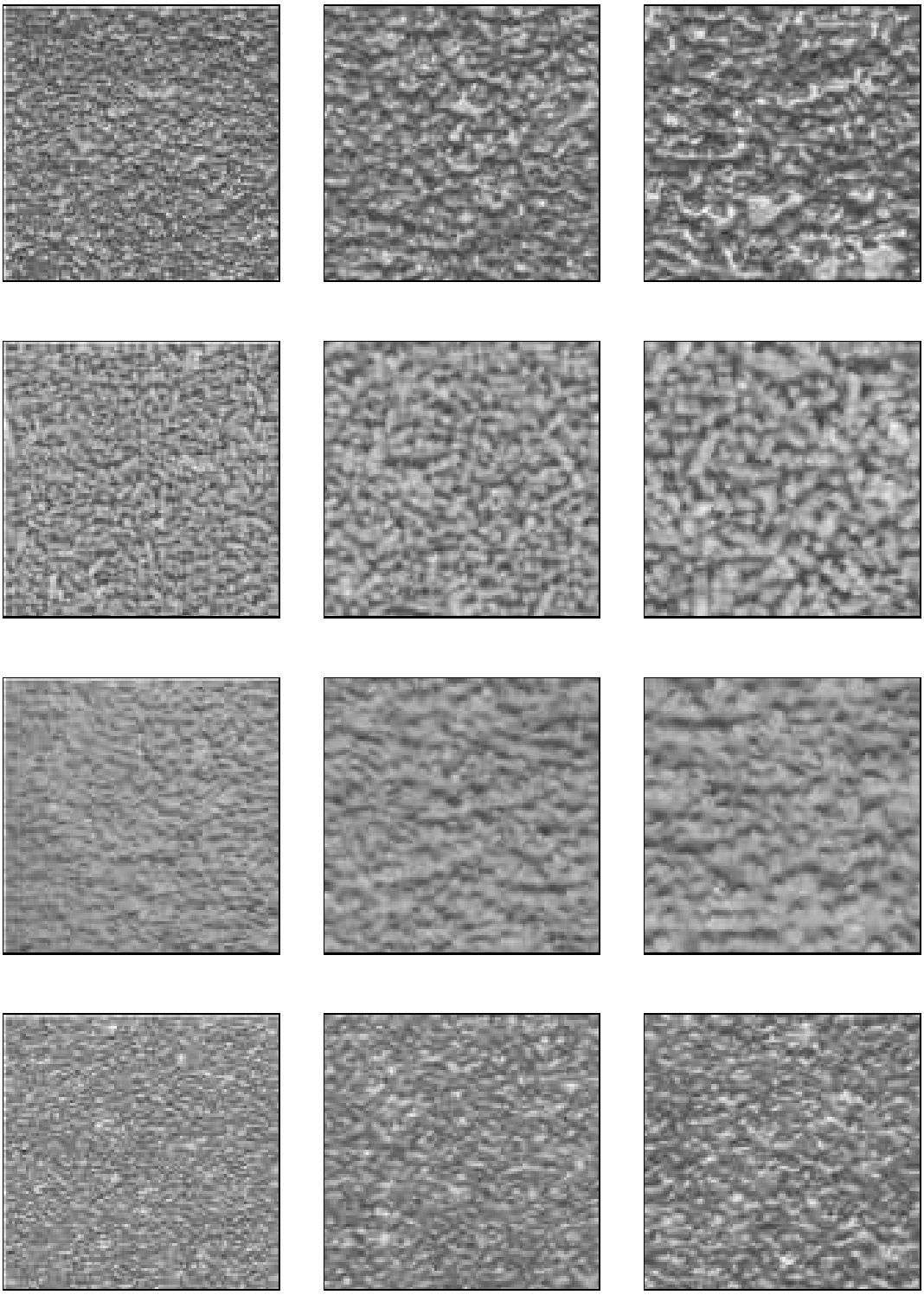


Abbildung 9.4: Ausschnitte aus den Texturen KK511, KK712, KK717 und KP510, jeweils in den Vergrößerungsstufen 50 %, 75 % und 100 %. Die Größe der dargestellten Bildausschnitte beträgt 100×100 Pixel.

Experiment	Texturmerkmale	Zahl der Merkmale	Klassifikator
KK511 KK712 KK717 KP510	HP2-ROT (12 Nachbarn, $D = 1 - 8$, $F = 32$)	5 aus 240	Nächste-Nachbar-Klassifikator ($k = 3$)

Tabelle 9.4: Konfiguration des Klassifikationssystems zur kontinuierlichen Bewertung der Oberflächeneigenschaften von Schleifpapieren

9.2.1 Bewertung von Schleifpapieren

Gegeben sind 4 Typen von Schleifpapieren (KK511, KK712, KK717 und KP510), die in unterschiedlichen Vergrößerungen aufgenommen wurden. Die Aufgabe besteht hier nicht in der (diskreten) Klassifikation des Papiertyps, sondern in der Prognose des (kontinuierlichen) Vergrößerungsfaktors für Analysefenster der Größe 32×32 Pixel. Zur Veranschaulichung sind in Abbildung 9.4 von jedem Papiertyp Bildausschnitte in den Vergrößerungsstufen 50 %, 75 % und 100 % dargestellt.

Zur Lösung der Prüfaufgabe wird für jeden Papiertyp ein separates Klassifikationssystem trainiert, wobei stets noch dem gleichen Schema vorgegangen wird (vgl. Tabelle 9.4). Die Daten zur Adaption des Systems setzen sich jeweils aus 5 Bildern der Größe 256×512 in den Skalierungen 50,0 %, 59,5 %, 70,7 %, 84,1 %, 100,0 % zusammen. Der logarithmischen Anordnung liegt ein Skalierungsfaktor von $\sqrt[4]{2} = 1{,}189$ von Bild zu Bild zugrunde, so dass eine Verdoppelung der Bildgröße über 4 Schritte erreicht wird. Trainings- bzw. Testdaten umfassen jeweils 100 Subbilder der Größe 32×32, die aus den oberen bzw. unteren Bildhälften entnommen werden.

Eine weitere Bildserie steht für die Validierung zur Verfügung. Diese Bilder haben ebenfalls die Größe 256×512 und liegen in den gleichen Skalierungen wie die Trainingdaten vor. Um das Interpolationsverhalten des adaptierten Systems zu testen, steht außerdem ein Bild der Skalierung 75,0 % bereit.

Das Klassifikationsschema basiert auf rotationsunabhängigen Merkmale der 2-Punkt-Statistik (LBP-ROT). Sie bilden die Eingangsgrößen zu dem Nächste-Nachbar-Klassifikator, der für jede Skalierungsstufe 100 Prototypen umfasst. Im Klassifikationsmodus werden die $k = 3$ nächsten Nachbarn (Prototypen) zum momentan vorliegenden Merkmalsvektor x gesucht. Das prognostizierte Resultat ergibt sich als das arithmetische Mittel der 3 zugeordneten Trainingsvorgaben. Die Tatsache, dass die Trainingsvektoren hier nur mit 5 verschiedenen Bewertungen vorkommen, liegt einzig und allein an dem verfügbaren Bildmaterial. Prinzipiell könnten alle Trainingsdaten mit unterschiedlichen Skalierungswerten verbunden sein.

Experiment	Skalierungsstufe					μ_S
	50,0	59,5	70,7	84,1	100,0	
KK511	0,71	4,0	3,3	1,1	6,3	3,1
KK712	0,32	0,72	2,4	6,0	4,1	2,7
KK717	1,0	3,1	4,3	9,0	7,3	4,9
KP510	1,0	1,3	6,1	7,2	5,6	4,2
μ_T	0,76	2,3	4,0	5,8	5,8	$\mu = 3{,}7$

Tabelle 9.5: Prognosefehler der Skalierung auf den Testdaten, aufgeschlüsselt nach Skalierungsstufen. μ_S gibt den gemittelten Prognosefehler über den Skalierungsstufen wieder, μ_T den Mittelwert über den Papiertypen und μ den Mittelwert über allen Experimenten.

9.2.2 Ergebnisse

In Tabelle 9.5 sind die Ergebnisse der Systemadaption für die 4 Papiertypen zusammengestellt. Die über die Skalierungsstufen gemittelten Prognosefehler μ_S liegen für die verschiedenen Papiertypen innerhalb des engen Intervalls von 2,7 bis 4,9 Prozentpunkten in der Skalierung. Ein anderes Bild ergibt sich bei der Betrachtung des über die Papiertypen gemittelten Fehlers μ_T für die verschiedenen Skalierungen. Während sich für kleine Skalierungen eine nahezu perfekte Prognose ergibt, nimmt der Fehler für wachsende Skalierung zu. Als Ursache für diesen Effekt kommt die Konstanz von Analysefenstergröße und Nachbarschaftsbereich in Frage. Offensichtlich besitzt die gewählte Kombination aus Fenstergröße $F = 32$ und Nachbarschaftsbereich $D = 1$–8 für kleine Skalierungen eine höhere Aussagekraft als bei großen Skalierungen. Dieser Effekt ließe sich durch ein verändertes Optimierungskriterium für die Merkmalsselektion beseitigen. Anstelle lediglich den mittleren Fehler über alle Skalierungen zu minimieren, könnte ein Strafterm eingeführt werden, der einen möglichst gleichmäßigen Fehler über alle Skalierungen begünstigt. Insgesamt bleibt anzumerken, dass durch ein vergrößerte Analysefenster F wesentlich bessere Resultate erzielt werden können.

Nach der Systemadaption wird für jedes Validierungsbild eine Skalierung prognostiziert. Diese ergibt sich als Mittelwert über die prognostizierten Skalierungen in den Subbildern der Größe 32×32 Pixel. Die Ergebnisse sind in Abbildung 9.5 dargestellt. Ähnlich wie bei den Testdaten nimmt auch hier der Fehler für wachsende Skalierung zu, ausgehend von sehr niedrigem Niveau. Der maximale Prognosefehler in Höhe von 6,0 Prozentpunkten tritt beim Papiertyp KK717 für die Sollskalierung 84,1 % auf. Der gemittelte Prognosefehler liegt bei 1,5 Prozentpunkten.

Eine weitere Beobachtung ist die systematisch zu kleine Prognose für die Skalierungsstufe 100 %. Sie ist in dem Klassifikationsschema begründet, das auf Nächster-Nachbar-Klassifikation beruht. Konstruktionsbedingt ist ein Nächste-Nachbar-Klassifikator nicht in der Lage, größere Werte zu prognostizieren, als

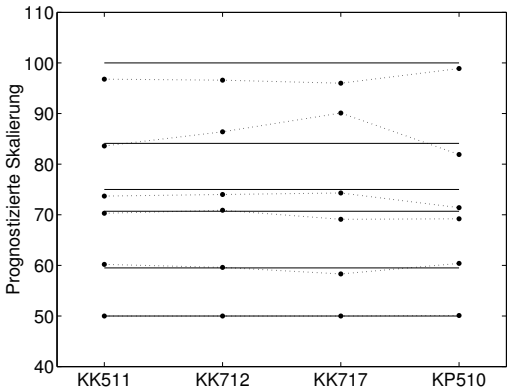


Abbildung 9.5: Prognosefehler der Skalierung für die Validierungsbilder gemittelt über die Subbilder. Die durchgezogenen Linien deuten die Sollwerte an.

im Trainingsdatensatz enthalten sind. Im vorliegenden Fall liegt der maximale Prognosewert also bei 100,0. Bedingt durch die Mittelung über die nächsten $k = 3$ Nachbarn sowie die Mittelung über die Subbilder eines Validierungsbildes ergeben sich in der Praxis Werte unterhalb des Grenzwerts von 100,0. Kurzum, die systematischen Schätzfehler an den Rändern des Trainingsbereichs sind auf die dort herrschende Asymmetrie der Trainingsdatenverteilung zurückzuführen. Sie können für den Prüfbetrieb vermieden werden, indem der Trainingsbereich über das abzudeckende Prüfintervall ausgeweitet werden.

Im Gegensatz zu Nächsten-Nachbar-Klassifikatoren sind neuronale Netze auch für die Extrapolation geeignet, d.h. sie können Werte außerhalb des trainierten Bereichs prognostizieren. Wie bei jeder Art der Extrapolation ist mit großer Vorsicht vorzugehen [PTVF92]. Im Fall der Prognose von Porengröße und -verteilung im letzten Abschnitt wurde die Ausgabe des neuronalen Netzes unter Verwendung der logistischen Funktion $f(x) = 1/(1 + \exp(-x))$ auf das Intervall $0{,}5 \ldots 5{,}5$ abgebildet. Dadurch wird ausgeschlossen, dass Porengrößen bzw. Porenverteilungen prognostiziert werden, die weitab des trainierten Bereichs $1 \ldots 5$ liegen.

9.3 Texturanalysesystem Adaptex

9.3.1 Zielsetzung

Obwohl die Leistungsfähigkeit von Texturanalyseverfahren in den vergangenen Jahren enorm gesteigert wurde, entspricht die Verbreitung im industriellen Einsatz nach wie vor nicht den potentiellen Anwendungsmöglichkeiten. Als großes Hemmnis ist die aufwendige Adaption der Prüfsysteme an neue oder veränderte Prüfaufgaben anzusehen. Die notwendigen Teilschritte zur Adaption (Zusammenstellung des Bildmaterials, Auswahl und Optimierung von Texturmerkmalen, Training eines Klassifikators und Validierung des Systems) werden in aller Regel nur von den Systemherstellern oder spezialisierten Dienstleistern beherrscht, so dass eine schnelle und kostengünstige Anpassung durch den Anwender selbst nicht möglich ist. In der industriellen Praxis kommen Oberflächenprüfsysteme daher überwiegend in der Massenproduktion von Gütern wie Stahlblech, Papier, Kunststoff, Textilien oder in der Holzbearbeitung zum Einsatz.

Eine Zielsetzung für die weitere Entwicklung der Texturanalyse ist die Erschließung von Märkten, die nach kostengünstigeren und flexibleren Verfahren verlangen. Ein Schritt in diese Richtung stellt das Texturanalysesystem „Adaptex“ zur *adap*tiven Bewertung *tex*turierter Oberflächen dar. Es fasst die in dieser Arbeit untersuchten und entwickelten Verfahren der Merkmalsextraktion, Merkmalsselektion und Klassifikation zusammen und stellt sie unter einer einfach zu bedienenden grafischen Benutzeroberfläche zur Verfügung. Grundidee ist, den Anwender in die Lage zu versetzen, das Prüfsystem selbstständig zu adaptieren. Der Anwender bringt dabei sein Know-How über die zu bewertenden Oberflächen und deren Fehlerklassen ein während das System alle technischen Aspekte der Mustererkennung und Klassifikation übernimmt.

9.3.2 Realisierung

Das Programmpaket gliedert sich in die drei Teile „AdaptexEvaluate“, „AdaptexTrain“ und „AdaptexClassify“. Die beiden ersten Teile AdaptexEvaluate und AdaptexTrain erlauben es dem Anwender Texturen einzulernen. AdaptexClassify dient dazu, auf Grundlage der eingelernten Texturen weitere Bilder zu klassifizieren. Die Philosophie von Adaptex basiert auf dem Ansatz des überwachten Lernens. Das bedeutet, der Anwender gibt vor, welche Texturklassen zu erkennen sind und mit welchen Klassenkennungen diese bezeichnet werden. Anhand dieser Information adaptiert sich das System selbstständig, indem das System Texturmerkmale auswählt, die für die gestellte Aufgabe eine hohe Unterscheidungskraft besitzen und für die Klassifikation geeignet sind. Die Verfahren von Adaptex ahmen aufgrund der vorgegebenen Bewertungen das Bewertungsschema des Anwenders nach und können ihm so Routinear-

beit bei der Klassifikation von Bildserien abnehmen. Neben Bewertungen in Form von diskreten Klassen wie beispielsweise „iO“ und „niO“ sind auch kontinuierliche Bewertungen in numerischer Form möglich, etwa die Prognose von Oberflächenrauheiten.

Interaktive Bewertung

Das Programm „AdaptexEvaluate“ erlaubt dem Anwender, Bildmaterial interaktiv zu bewerten, d.h. Bildbereiche per Maus zu markieren und mit einer Klassenkennung zu versehen. Zusätzlich zu der Klassenkennung kann der Anwender festlegen, ob der markierte Bildbereich für das Training (Trainingsdatensatz) oder für die Überprüfung (Testdatensatz) des Adaptionssystems verwendet werden soll. Aufgrund dieser Bewertungsinformation führt AdaptexTrain anschließend die Adaption des Systems durch.

Adaption

Auf Basis der interaktiv erstellten Bewertungsinformation adaptiert AdaptexTrain das Klassifikationssystem. Der modulare Aufbau ermöglicht es dem Anwender, zwischen alternativen Verarbeitungsmethoden zu wählen. So kann beispielsweise eine Vorauswahl für die zu berücksichtigenden Texturmerkmale getroffen werden, ein Klassifikatortyp gewählt oder das Trennschärfekriterium für die Merkmalsselektion bestimmt werden. In allen Fällen existieren jedoch Grundeinstellungen, die in der Regel zu guten Prüfergebnissen führen.

Als Ergebnis liegt ein fertig adaptiertes Prüfsystem vor. Die Qualität und Zuverlässigkeit des Prüfsystems lassen sich anhand der Fehlerraten auf den Testdaten ablesen. Eine Vertauschungsmatrix zeigt an, welche Texturklassen sicher erkannt werden.

Klassifikation

Mit dem Werkzeug AdaptexClassify können Bilder automatisch klassifiziert werden. Das adaptierte Prüfsystem ahmt dabei das anhand von Beispielen gelernte Prüfverhalten nach. Die Ergebnisse werden visualisiert, indem die gefundenen Texturklassen dem Bild farblich kodiert überlagert werden. Durch Angabe einer geforderten Mindestähnlichkeit zu den trainierten Klassen (vgl. Abschnitt 5.2) lässt sich die Klassifikation von Bildbereichen unterbinden, die nicht den trainierten Klassen entsprechen.

9.3.3 Beispiel

Die Vergehensweise zur Adaption eines Prüfsystems soll hier an einem einfachen Beispiel demonstriert werden. Die beiden Texturen D53 und D68 aus dem

Wahre Klasse	Prognostizierte Klasse D53	D68
D53	100	0
D68	0	100

Tabelle 9.6: Verwechslungsmatrix für die Klassifikation von Textur D53 und D68.

Brodatzalbum werden eingelernt, indem in einem ersten Schritt entsprechende Texturbereiche interaktiv markiert werden (AdaptexEvaluate, vgl. Abbildung 9.6). In einem zweiten Schritt wird das System mit diesen Daten adaptiert (AdaptexTrain, vgl. Abbildung 9.7). Die Adaptionszeit beträgt weniger als 0,5 Sekunden auf einem 2,4 GHz Intel Pentium 4 Prozessor. Auf den Testdaten wird in diesem einfachen Beispiel eine Fehlerrate von 0,0 % erreicht. Entsprechend sind die Nicht-Diagonalelemente der Vertauschungsmatrix in Tabelle 9.6 gleich 0. Schließlich wird die Texturszene aus Abbildung 9.8 klassifiziert, die den Schriftzug „IPA“ in Textur D53 vor dem Hintergrund in Textur D68 zeigt. Das Ergebnis ist in Abbildung 9.9 zu sehen.

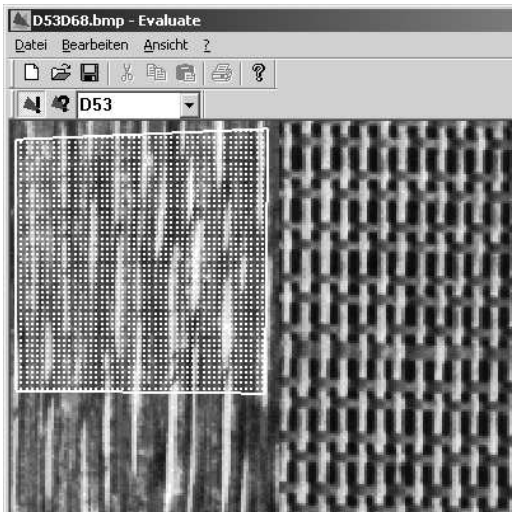


Abbildung 9.6: Bei der interaktive Bewertung von beispielhaftem Bildmaterial wird für jede Texturklasse ein typischer Bildbereich markiert.

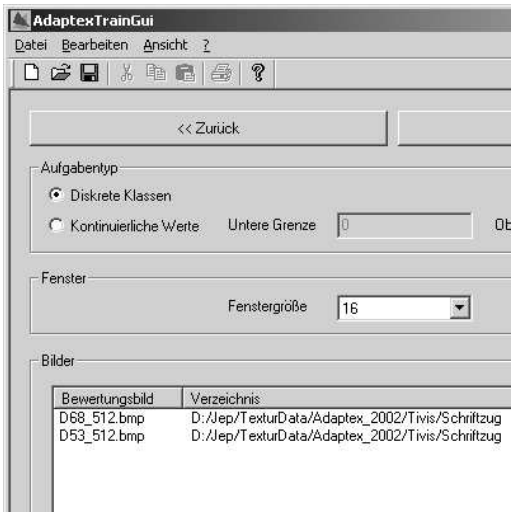


Abbildung 9.7: Das Training des Klassifikationssystems läuft automatisch ab. Falls gewünscht, können Grundeinstellungen wie die Vorauswahl der zu berücksichtigenden Texturmerkmale geändert werden.

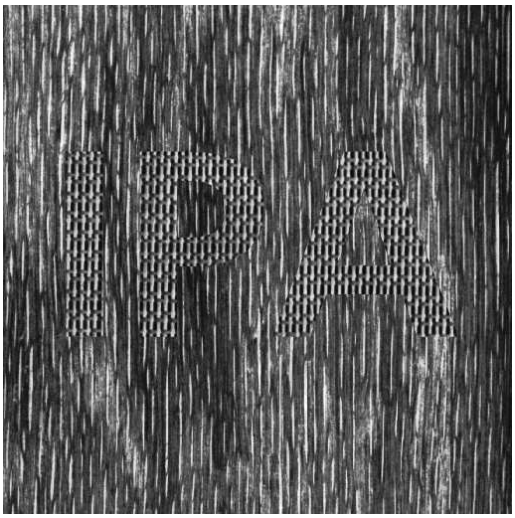


Abbildung 9.8: Zu klassifizierendes Bild, das die beiden Texturen enthält.

Abbildung 9.9: Ergebnis der Klassifikation. Die beiden Texturen sind farbcodiert als schwarze und weiße Pixel dargestellt.

Kapitel 10

Zusammenfassung und Ausblick

Zielsetzung der vorliegenden Arbeit war die Entwicklung von adaptiven Verfahren zur Bewertung texturierter Oberflächen. Im Gegensatz zu traditionellen Verfahren der Oberflächenbewertung ermöglicht die entwickelte Systematik der Selbstadaption eine schnelle Anpassung an neue oder veränderte Prüfaufgaben, ohne vom Anwender Expertenwissen über die zugrundeliegenden Algorithmen der Merkmalsextraktion vorauszusetzen. Der Anwender gibt vor, *was* unterschieden werden soll, während das System mit Hilfe von adaptiven Verfahren festlegt, *wie* diese Vorgaben umzusetzen sind. Die Interaktion des Anwenders mit dem System beschränkt sich darauf, exemplarisches Bildmaterial in einem grafischen Editor zu bewerten. Je nach Komplexität der Prüfaufgabe dauert die sich anschließende automatische Adaption zwischen mehreren Sekunden bis zu einigen Minuten. Damit ist der Einsatz der Verfahren auch für kleine oder schnell wechselnde Prüfaufgaben möglich und wirtschaftlich sinnvoll.

Um dieses Ziel zu erreichen, wurden Verfahren der Texturanalyse von der Merkmalsberechnung über die Klassifikation bis hin zur Merkmalsselektion bzw. -optimierung weiterentwickelt, optimiert und in das modular aufgebaute Gesamtsystem „Adaptex“ integriert. Die Verfahren wurden an Benchmark-Datensätzen validiert. Die in der Literatur dokumentierten Ergebnissen konnten in allen Fällen übertroffen werden. Im industriellen Einsatz zur Qualitätssicherung wurde nachgewiesen, dass derartige Systeme äußerst zuverlässige Oberflächenbewertungen liefern.

Im Einzelnen wurden die folgenden Arbeiten durchgeführt:

Zielsetzung von Kapitel 4 war die *Optimierung von Merkmalen der 2-Punkt-Statistik und lokalen binären Mustern* im Hinblick auf Rotationsinvarianz, Multiskalenanalyse und Robustheit. Als Maßnahmen wurden vorgeschlagen und untersucht: Randomisierung der Analyserichtung, Erhöhung der Anzahl der Analyserichtungen, Verwendung eines Satzes von Distanzvektoren unterschiedlicher Länge, Unterabtastung der Texturen, Analyse der Einflüsse von Quantisierung und Histogrammglättung sowie Berücksichtigung von Spiegelinvarianz neben der Rotationsinvarianz für lokale binäre Muster. Die Wirkung

dieser Maßnahmen kann eindrucksvoll an den Experimenten C bis G in Abschnitt 8.1 nachvollzogen werden. Die in der Literatur [POX00] angegebene Fehlerrate für lokale binäre Muster konnte um mehr als die Hälfte von 47,7 % auf 19,7 % reduziert werden. Die niedrigste dokumentierte Fehlerrate für kombinierte Merkmale wurde sogar von 24,0 % um mehr als zwei Drittel auf 7,7 % verringert.

Der Fokus von Kapitel 5 lag auf *Aspekten der Klassifikation*. Zunächst wurde die Zuverlässigkeit der Klassifikation unter dem Einfluss von kleinen Lernstichproben diskutiert und eine Motivation für den Einsatz von Verfahren der Merkmalsselektion gegeben. Um die Wahrscheinlichkeit von Fehlklassifikationen zu verringern, wurde außerdem das Konzept der relativen Ähnlichkeit vorgeschlagen. Sie quantifiziert die Ähnlichkeit eines Merkmalsvektors x zu einer Klasse ω_i und unterdrückt die unsichere Klassifikation von Merkmalsvektoren, die in der Trainingsmenge nicht ausreichend repräsentiert sind. Schließlich wurde untersucht, wie sich Schätzfehler in den Merkmalen auf die Robustheit der Klassifikation auswirken. Insbesondere konnte gezeigt werden, dass die Verwendung eines euklidischen Abstandsmaßes gegenüber der Kreuzentropie Vorteile besitzt. Dieser Effekt, der als implizite Merkmalsgewichtung bezeichnet werden soll, konnte in den Experimentpaaren A–C in Abschnitt 8.1 bzw. H–I in Abschnitt 8.2 nachgewiesen werden. Liegen instationäre Texturen wie bei den Experimenten L und M in Abschnitt 8.2.3 vor, weist die Klassifikation mit impliziter Merkmalsgewichtung einen geringeren Anstieg in der Fehlerrate auf als mit Merkmalsselektion.

Kapitel 6 beschäftigte sich mit *Verfahren der Merkmalsselektion*. Zunächst wurden diverse Gütekriterien vorgestellt und die enge Beziehung zwischen Gütekriterien und verwendetem Klassifikatortyp herausgearbeitet. Für das Mehrklassenproblem wurde vorgeschlagen, wie globale Gütekriterien aus den paarweisen Klassenabständen abgeleitet werden können. Um die Robustheit der Gütekriterien „Kullback-Leibler“ und „Bhattacharyya“ gegenüber Schätzfehlern in den Merkmalen zu steigern, wurde jeder Klasse im Merkmalsraum ein minimales Volumen zuzuordnet (vgl. Abschnitt 6.2.1). Nach einem kurzen Überblick über effiziente Suchverfahren wurden Empfehlungen für den praktischen Einsatz der Merkmalsselektion gegeben. Experimente G und K aus den Abschnitten 8.1 bzw. 8.2 demonstrieren, dass die Merkmalsselektion auch auf großen Merkmalsmengen von $M = 270$ bzw. $M = 456$ Merkmalen anwendbar ist und zu sehr niedrigen Fehlerraten führt. Der entscheidende Vorteil der Merkmalsselektion ist in der anwendungsspezifischen Auswahl der effektivsten Merkmale zu sehen. Damit lassen sich niedrige Fehlerraten bei hoher Verarbeitungsgeschwindigkeit realisieren, wie ein Vergleich der Experimentserien P und Q aus Abschnitt 8.3 bestätigt: Durch die Auswahl von $m = 4$ aus $M = 40$ Merkmalen verringerte sich die Fehlerrate um 3,6 Prozentpunkte auf 25,5 % während sich die Klassifikation um Faktor 10 beschleunigte. Die Merkmalsse-

lektion selber benötigte nur 29,9 Sekunden.

Eine Alternative zur Merkmalsselektion stellt die *Merkmalsoptimierung* dar, die Thema von Kapitel 7 war. Während der Suchraum bei der Merkmalsselektion auf die abzählbar vielen Kombinationen $\binom{M}{m}$ beschränkt ist, werden bei der Merkmalsoptimierung die P freien Parameter des Merkmalssatzes im $\mathbb{R}^P$ optimiert. Als Optimierungsverfahren wurde die CMA-Evolutionsstrategie [HO96] ausgewählt. Sie eignet sich für die Optimierung im hochdimensionalen Parameterraum und zeichnet sich durch ihre Robustheit gegenüber multimodalen oder stochastisch gestörten Zielfunktionen aus. Das Verfahren wurde für die Optimierung einer Gaborfilterbank eingesetzt. Jeder Gaborfilter war durch die 4 freien Parameter Zentralfrequenz u_0, die Filterbreiten σ_x und σ_y und die Orientierung θ festgelegt. In den Experimentserien R und S aus Abschnitt 8.3 wurden bis zu 6 Gaborfilter mit 12 Merkmalen verwendet. Die erzielte Fehlerrate lag 0,2 Prozentpunkten unter dem besten Ergebnis mit Merkmalsselektion und 7,0 Prozentpunkte unter dem besten Literaturergebnis [RH99]. Dagegen dauerte die Optimierung mit 308 Sekunden wesentlich länger als die Merkmalsselektion.

Kapitel 9 widmete sich der *Umsetzung der entwickelten Verfahren* in die industrielle Praxis. Untersuchungsgegenstand in Abschnitt 9.1 waren medizinische Kollagenschwämme. Für die Bewertung von Porengröße und Porenverteilung wurde ein Prüfgerät für den industriellen Einsatz realisiert. Im Vergleich zur manuellen Prüfung konnte der Bewertungsfehler für die Porengröße von 0,36 auf 0,21 verringert werden. Für die Porenverteilung ergab sich sogar eine Halbierung von 0,91 auf 0,44. In Abschnitt 9.2 wurde die Prognose von kontinuierlichen Oberflächenparametern am Beispiel von Schleifpapieren untersucht. Hierzu wurde ein Verfahren entwickelt, bei dem die Zuordnung in diskrete Klassen durch ein kontinuierliches Interpolationsschemas ersetzt wurde. Die konkrete Aufgabe bestand darin, den Skalierungsgrad der Oberflächenstruktur zu prognostizieren. Trotz kleiner Analysefenster der Größe 32×32 Pixel lag der Prognosefehler bei lediglich 3,7 Prozentpunkten. Schließlich wurde in Abschnitt 9.3 das Texturanalysesystem „Adaptex“ vorgestellt. Es ermöglicht die schnelle Adaption an neue Prüfaufgaben, ohne von den Anwendern Expertenwissen der Texturanalyse vorauszusetzen. Die Vorgehensweise wurde an einem einfachen Beispiel demonstriert. Sie setzt sich aus den beiden Schritten „Interaktive Bewertung“ und „Automatische Adaption“ zusammen. Das Texturanalysesystem Adaptex integriert alle in dieser Arbeit vorgestellten Verfahren. Der zugrunde liegende Quellcode in C/C++ wurde im Rahmen dieser Arbeit erstellt und umfasst mehr als 50 000 Zeilen. Bis auf mathematische Hilfsroutinen aus „Numerical Recipes in C“ [PTVF92] wurden keine fremden Quellen oder Bibliotheken eingebunden.

Kennzeichnend für den Verlauf dieser Arbeit war das enge Ineinandergreifen von Forschungs- und Industrieprojekten. Verfahren, die im DFG-Schwerpunkt-

programm „Automatische Sichtprüfung technischer Objekte“ oder im BMBF-Rahmenkonzept „Forschung für die Produktion von morgen“ entwickelt wurden, bildeten die Basis für zahlreiche Industrieprojekte. Im Gegenzug lieferten die industriellen Kontakte die Impulse und Herausforderungen, um neue Fragestellungen anzugehen.

Als lohnende Themen für die Zukunft werden angesehen:

Erweiterte Optimierung: Um das Potential der Merkmalsoptimierung auszuschöpfen, sollten weitere Parameter wie die Größe des Analysefensters oder die Anzahl der zu verwendenden Merkmale in die Optimierung mit einbezogen werden.

Berücksichtigung der Schätzgüte: In viele Gütekriterien der Merkmalsselektion bzw. -optimierung (Kullback-Leibler- und Bhattacharyya-Abstand, nichtparametrisches Überlappungsmaß, vgl. Abschnitt 6.2) fließt lediglich die Diskriminierungsfähigkeit, nicht jedoch die Schätzgüte der Modellparameter ein. Es ist zu erwarten, dass durch Berücksichtigung der Schätzgüte die Zuverlässigkeit der Gütekriterien zunimmt.

Komplexität des Prozessmodells: In Abschnitt 5.1.1 wurde darauf hingewiesen, dass der Anpassung der Komplexität des Prozessmodells eine hohe Bedeutung zukommt. Bisher erfolgt diese Anpasssung, indem Anzahl und Auswahl der Merkmale mit Merkmalsselektion bestimmt werden. Als Erweiterung bietet sich die Optimierung der inneren Freiheitsgrade des Prozessmodells an.

Unüberwachtes Lernen: Um auch Klassen unterscheiden zu können, ohne diese vorab einzulernen, muss der Prozess der Klassenbildung automatisiert werden. Neben den Klassenverteilungen im Merkmalsraum sollte dabei auch die räumliche Verteilung der Texturen berücksichtigt werden [Jäh97b] [BHR81].

Kapitel 11

Summary

11.1 Introduction

In general, technical as well as natural surfaces exhibit a characteristic surface structure (texture) permitting their discrimination and classification. The objective of this work consists in the development of methods supporting the adaptation of surface inspection systems to new or modified inspection tasks. Its focus is set on the automatic selection of texture features and the optimisation of texture parameters. The automatic adaptation is based on sample sets which have been classified by the user in advance.

Characteristic for this work is a feedback loop in the processing chain which enables the optimisation of texture features and parameters. The fitness criteria for the optimisation are based on the analysis of class-specific feature distributions as well as on verification scores obtained from the classification of sample sets. These fitness criteria are used both for feature selection and parameter optimisation with evolution strategies. Another topic of the work is the development of rotation-invariant texture features which is achieved by randomising the operator orientation.

11.2 Rotation-Invariant Texture Features

As human beings do, machine vision systems should be able to classify textures independent of their orientation. An overview of methods for rotation-invariant texture classification is given by Pietikäinen *et al.* [POX00].

The basic task of texture classification consists in determining the texture class C of an input image at position P. Although quite different in implementation, most approaches share the following steps: First, an rotation-dependent operator O_φ is defined in order to extract information from neighboring pixels. Applying O_φ to a set of R evenly-spaced directions results in rotation-dependent features f_φ^{dep}. In the next step, rotation-invariant features

Example

6	5	2
7	6	1
9	8	7

Binary Image

1	0	0
1		0
1	1	1

Weights

1	2	4
128		8
64	**32**	**16**

$$\text{LBP} = (10001111)_2 = 1 + 16 + 32 + 64 + 128 = 241$$

Figure 11.1: Calculation of LBP for a 3×3 neighborhood ($R = 8$): After thresholding the neighborhood by the value of the center pixel, the LBP is derived as inner product with a constant weight matrix.

f^{inv} are derived from f_φ^{dep} by means of normalization or integration. Alternatively, the features f_φ^{dep} are fed into an invariant classifier such as symmetric phase matched only filtering (SPMOF) [TPH95]. In both cases, true rotation-invariance is only achieved if the inverse angular resolution ρ^{-1} of operator O_φ is greater than its angular spacing $\Delta\varphi = 2\pi/R$. Gabor filter banks for example as given by Jain *et al.* [JF91] agree to this condition. For features such as local binary patterns (LBP) [OPM02] or features based on sum and difference histograms (SUMDIF) [Uns86] true rotation-invariance results only if $\rho^{-1} = 1/D \geqq \Delta\varphi = 2\pi/R$, where D signifies the length of the displacement vector.

11.2.1 Local Binary Patterns (LBP)

Ojala *et al.* [OPH96] introduced the Local Binary Pattern (LBP) texture operator shown in Fig. 11.1. The basic idea consists in comparing the gray-value at position P with a set of R neighbors, thus creating a binary number of length R which corresponds to a local binary pattern. The LBP histogram computed over a region is used for texture description. For $R = 8$, a 256-bin histogram is obtained. Rotation-invariant histograms can be derived if all cyclic permutations of a binary LBP are mapped onto the same rotation-invariant pattern LBP-ROT [POX00] [OPM02]. In this way, rotation-invariance is generated by means of normalization.

No. of orientations R	Angular operator spacing $\Delta\varphi$	No. of LBP	No. of LBP-ROT	No. of mirror-symmetric LBP-ROT
4	90.0 °	16	6	6
8	45.0 °	256	36	30
12	30.0 °	4096	352	224
16	22.5 °	65536	4116	2250

Table 11.1: The number of LBP rises exponentially with the number of operator orientations R. If LBPs emanating from rotation or mirroring are not to be distinguished, the number of LBP-ROTs is reduced.

The discriminating power of LBP/LBP-ROT histograms can be significantly increased by using a multipredicate approach: Histograms are not only calculated for neighbors with distance $D = 1$ to the central pixel P but for multiple distances D.

LBP/LBP-ROT histograms are classified by means of a log-likelihood or distance classifier.

11.2.2 Sum and Difference Histograms (HP2)

Unser [Uns86] introduced sum and difference histograms for texture description as an alternative to the widespread co-occurrence matrices [HSD73]. The advantages are savings in computation time and more reliable estimates of the (1-dimensional) histograms. Like co-occurrence matrices, sum and difference histograms are calculated for a specific displacement vector v with orientation φ and length D over an image region.

Typical texture features HP2 (histogram of 2-point statistics) derived from the histograms are mean, variance, energy, correlation, entropy, contrast, homogeneity, cluster shade and cluster prominence. Histograms and features HP2 may be calculated for a set of R evenly spaced orientations. Rotation-invariant features HP2-ROT are obtained as moments (mean, variance) of features HP2 over the R orientations. Thus, invariance is achieved by integration.

As with LBP/LBP-ROT, a multipredicate approach improves on the discriminating power. A normal distribution classifier [Fuk90] is suggested for classification.

11.3 Rotation Invariance by Randomisation

As stated above, true rotation invariance requires the inverse angular resolution ρ^{-1} of the operator to be greater than its angular spacing $\Delta\varphi = 2\pi/R$:

$$\rho^{-1} \geqq \Delta\varphi = 2\pi/R \tag{11.1}$$

For features LBP-ROT and HP2-ROT, the inverse angular resolution is given by $\rho^{-1} = 1/D$ corresponding to the inverse angular resolution of a pixel at distance D. Thus, features calculated for $D = 1$ and $R = 8$ can be considered as rotation-invariant ($\rho^{-1} = 1 \geqq 0.79$). If LBP-ROT or HP2-ROT are calculated for an increased predicate, eg. $D = 3$, condition (11.1) no longer holds for $R = 8$ ($\rho^{-1} \approx 0.33 \ngeqq 0.79$). Increasing the number R of orientations is an obvious choice for restoring the inequality. Unfortunately, there are limitations for increasing R. For LBP, the number of patterns rises exponentially with R (cf. Tab. 11.1), thus leading to sparsly populated histograms. For HP2 on the

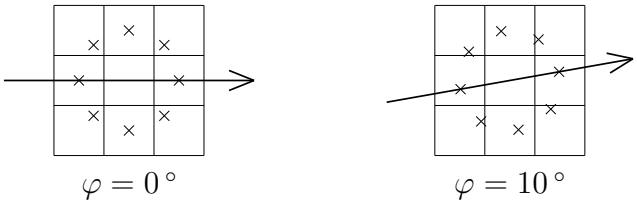


Figure 11.2: The randomisation of operator orientation is demonstrated for local binary patterns (LBP) with $R = 8$.

other hand, the calculation of large numbers of histograms becomes computationally inefficient. Typical values for D such as $D = 5$ would necessitate the evaluation of more than 30 orientations.

Instead of adjusting the number of operator orientatons R, we propose to increase the inverse angular resolution ρ^{-1} of the operator by randomising its orientation O_φ: Every time a histogram entry for LBP-ROT or HP2 is calculated, the orientation of O_φ is offset by a random angle from the interval $[-\pi/R, \pi/R]$. Figure 11.2 displays the randomization of operator orientation for LBP/LBP-ROT. Additionally, it may be considered to calculate more than one histogram entry for each position P in order to obtain statistically more reliable histograms. Due to the random orientation and interpolation effects the extracted binary pattern resp. sum/difference of gray values might take different values.

For an efficient implementation one might determine the random orientation and neighbor positions for each histogram entry in advance. They can be re-used for every region for which texture properties have to be extracted.

11.4 Feature Selection

The motivation to reduce the number of features is given by a reduced computational complexity and by a possible reduction in error rate. The circumstance that more features do not lead necessarily to lower error rates has been described in a large number of studies [JZ97] [Tru79] [OD92] [PS98] [PMW00]. The reason for this phenomenon are estimation errors which increase with the number of features in finite sample sets. Depending on the point of view, this phenomenon is called „Finite Sample Size Effect“ or „Curse of Dimensionality“.

Since feature selection is an optimisation problem appropriate fitness criteria and search strategies are needed.

The fitness criteria used in this work are based on the analysis of class-specific feature distributions (Kullback-Leibler distance or divergence [Kul59] [Hak88] [Fuk90] and the Bhattacharyya distance [Fuk90]) as well as on verification scores obtained from the classification of sample sets [MCKV97] [Bri50].

Search strategies which have turned out to be very efficient [JZ97] [Wag97] are the sequential search [Kit78] [PS98] and the sequential floating search [PNK94].

The feature selection of experiment G described in the next section uses the mean a-posteriori probability of the true class as verification score combined with a sequential search strategy.

An extension of feature selection is given by feature optimisation with evolutionary strategies [Rec94] [HO96] [HO97].

11.5 Experiments

The textures for the experiments originate from Pietikäinen *et al.* [POX00]. They comprise 15 different Brodatz textures in 7 rotation angles. For each rotation angle and texture 64 samples of size 32×32 pixels are provided. Samples of rotation angle $0\,^\circ$ were used for training while samples of rotation angles $30\,^\circ$, $60\,^\circ$, $90\,^\circ$, $120\,^\circ$, $150\,^\circ$ and $200\,^\circ$ were used as test data.

Seven experiments with different settings are compared (Tab.11.2). Experiment A shows the best result of Pietikäinen using only local binary patterns and serves as reference experiment. In Experiment B additional features (SCOV = symmetric covariance measure) are used by Pietikäinen. Experiments C–G were implemented in this work. Each experiment improves upon the feature extraction or classification in order to reduce error rates. Experiment C replaces the log-likelihood by a distance-classifier. In Experiment D, the number of neighbors is increased from $R = 8$ to $R = 12$. At the same time, mirrored patterns are considered to be equivalent, thus limiting the number of histogram bins to 224, cf. Tab. 11.1. Experiment E introduces a multipredicate approach. Finally, experiment F uses randomised orientations for the LBP-ROT. Experiment G is based on a completely different approach. Here, features HP2-ROT are extracted from sum and difference histograms and classified by a normal distribution classifier. Moreover, feature selection [JZ97] is applied in order to reduce the number of features from 270 to 8, keeping only the features with the highest power of discrimination.

11.6 Results and Discussion

In Tab. 11.3 overall error rates (averaged over all textures and all rotation angles) are shown. From experiment A/B to G, the error rate could be reduced from 47.7 % resp. 24.0 % to 7.7 % proving the effectiveness of the proposed modifications. Moreover, the modest increase in error rates from training to test set demonstrate the rotation-invariance of all features. Every single modification in the feature extraction or the classification process (C: distance

Experiment	Texture Feature	No. of Features	Classifier	Remark
A	LBP-ROT (8 neighbors with $\Delta\varphi = 45\,^\circ$, $D = 1$)	36	Log-Likelihood	Result of Pietikäinen [POX00]
B	LBP-ROT, SCOV (8 neighbors with $\Delta\varphi = 45\,^\circ$, $D = 1$)	68	Log-Likelihood	Best result of Pietikäinen [POX00]
C	LBP-ROT (8 neighbors with $\Delta\varphi = 45\,^\circ$, $D = 1$)	36	Distance	Experiment corresponding to A
D	LBP-ROT (12 neighbors with $\Delta\varphi = 30\,^\circ$, Mirror Invariance, $D = 1$)	224	Distance	
E	LBP-ROT (12 neighbors with $\Delta\varphi = 30\,^\circ$, Mirror Invariance, $D = 1, 2, 4, 6$)	896	Distance	
F	LBP-ROT (12 neighbors with $\Delta\varphi = 30\,^\circ$, Mirror Invariance, $D = 1, 2, 4, 6$, Randomised)	896	Distance	
G	HP2-ROT (For $D = 1 - 6$: 12 directions with $\Delta\varphi = 15\,^\circ$, Otherwise: 8 directions with $\Delta\varphi = 22.5\,^\circ$, Randomised)	8 out of 270	Normal distribution	With feature selection

Table 11.2: Overview of experiments.

Experiment	Error rate on training set (%)	Error rate on test set (%)
A	not available	47.7
B	not available	24.0
C	37.7	41.3
D	35.2	39.5
E	14.7	22.5
F	16.7	19.7
G	5.7	7.7

Table 11.3: Error rates (%) on training and test set. Each introduced modification reduces the error rate.

Experiment	All Textures at 200 °	Texture D21 only at 200 °
A	not available	not available
B	not available	not available
C	43.5	71.9
D	42.1	76.6
E	35.9	60.9
F	21.0	0.0
G	6.8	0.0

Table 11.4: Error rates (%) on test set for textures with rotation angle 200 °. Here, the effect of randomization (E→F) is much greater than in Tab. 11.3, especially for the periodic texture D21.

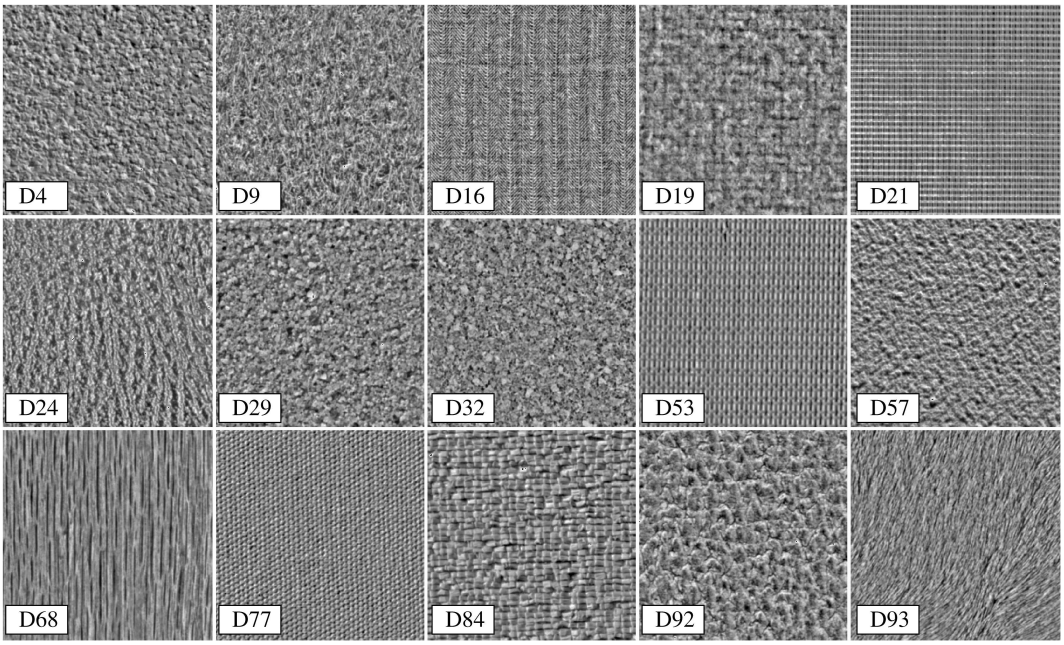


Figure 11.3: Original Brodatz images from which the training and test samples of size 32×32 pixels were taken.

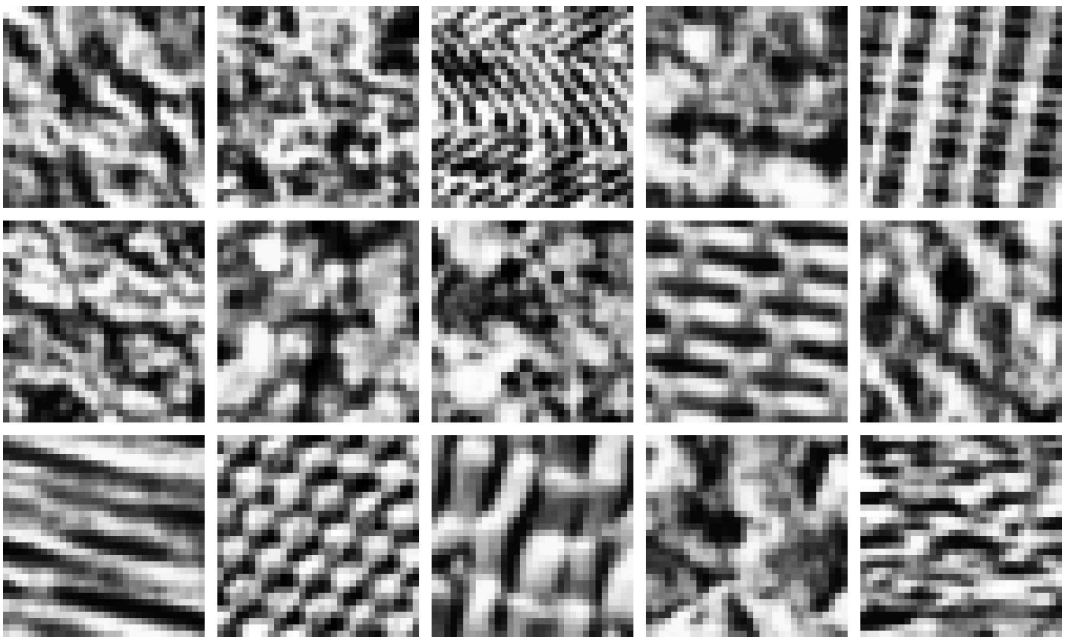


Figure 11.4: A sample of size 32×32 pixels is displayed for each of the 15 textures.

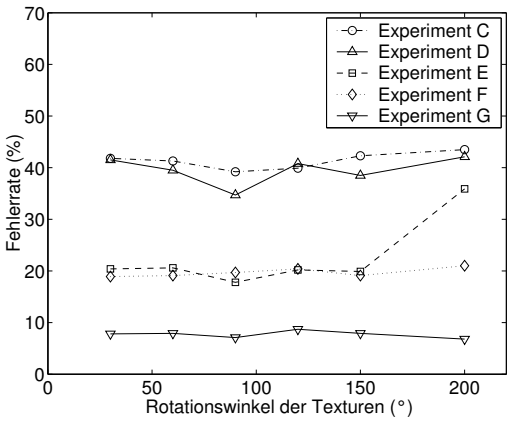


Figure 11.5: Error rates on test set for different rotation angles.

classifier, D: increased no. of neighbors, E: multipredicate approach, F: randomization, G: features HP2-ROT with feature selection) lowered the error rates. The largest reduction was due to the introduction of the multipredicate approach (D→E) whereas the randomization seems to be less important at first glance (E→F).

A different rating arises if only textures with rotation angle 200 ° are considered (Tab. 11.4). Now, the largest drop in error rate occurs from experiment E to F due to randomization. A careful examination of the experimental setup reveals that the rotation angle 200 ° is particularly meaningful. Using $R = 12$ neighbors for LBP-ROT corresponds to an intrinsic rotation invariance of $360\,°/12 = 30\,°$. Thus, the rotation angles 30 °, 60 °, 90 °, 120 °, 150 ° are not as significant as 200 ° for the verification of true rotation-invariance.

Figure 11.5 underlines these findings. Additionally, it shows a near-constant error rate over the rotation angles for all experiments except for experiment E. Due to the introduction of enlarged predicates ($D = 1, 2, 4, 6$) condition (11.1) for true rotation-invariance no longer holds. Error rates rise sharply for rotation angles not being multiples of $360\,°/R = 30\,°$. True rotation invariance is restored by the randomization of operator orientation in experiment F.

The importance of the randomization is emphasized by another observation. Texture D21 shows a near periodic structure making it much more error-prone to small rotations than the more stochastic textures. For texture D21, the error rate drops from 60.9 % to 0.0 % through randomization, cf. Tab. 11.4!

The best results are obtained if feature selection is applied. Table 11.5 displays the results of experiment G in detail. The confusion matrix reveals that 5 textures (D16, D53, D68, D77 and D93) are classified perfectly. On the other hand 3 textures (D19, D29, D92) account for 58 % of the misclassifications. Considering the similarity of these textures in Fig. 11.4 points up the the difficulty of the given classification task.

True class	Predicted class D4	D9	D16	D19	D21	D24	D29	D32	D53	D57	D68	D77	D84	D92	D93
D4	361	5	0	0	0	13	0	0	0	5	0	0	0	0	0
D9	11	344	0	5	0	11	8	5	0	0	0	0	0	0	0
D16	0	0	384	0	0	0	0	0	0	0	0	0	0	0	0
D19	0	0	0	323	0	1	6	1	0	2	0	0	2	49	0
D21	0	0	0	0	383	0	0	0	0	0	0	0	0	0	1
D24	7	17	0	0	0	350	1	0	0	9	0	0	0	0	0
D29	14	1	0	13	0	0	314	13	0	0	0	0	5	24	0
D32	1	1	0	3	0	0	17	348	0	0	0	0	0	14	0
D53	0	0	0	0	0	0	0	0	384	0	0	0	0	0	0
D57	7	0	0	0	0	14	0	0	0	363	0	0	0	0	0
D68	0	0	0	0	0	0	0	0	0	0	384	0	0	0	0
D77	0	0	0	0	0	0	0	0	0	0	0	384	0	0	0
D84	0	0	0	6	0	0	23	0	0	0	0	0	355	0	0
D92	2	0	0	79	0	0	38	8	0	0	0	0	1	256	0
D93	0	0	0	0	0	0	0	0	0	0	0	0	0	0	384

Table 11.5: Confusion matrix for experiment G. The test data for each class is made up of 384 subimages.

11.7 Conclusion

The focus of this work has been on adaptive methods for the classification of textured surfaces.

It has been demonstrated that real rotation invariance presumes the inverse angular resolution ρ^{-1} of the texture operator to be greater than its angular spacing $\Delta\varphi = 2\pi/R$, with R specifying the number of operator orientations. For texture features such as standard LBP-ROT or HP2-ROT this condition is only fulfilled for small displacement vectors v of moderate length D. The randomization of operator orientation has been proposed as an efficient approach to obtain true rotation invariance, even for large predicates. Its applicability has been demonstrated for texture features LBP-ROT and HP2-ROT. Especially, for near-periodic textures the error rate could be reduced significantly. The approach can easily be extended to other texture features.

Moreover, methods for the selection and optimisation of features have been investigated. Special attention has been attached to the evaluation of fitness criteria and search strategies. The optimisation of features has been performed by means of evolution strategies. In order to validate the methods extensive benchmark tests have been carried out. As a result, error rates have been achieved being significantly below error rates reported in the literature. All implemented methods have been integrated in the modular system „Adaptex“ for adaptive texture analysis. The industrial applicability of the methods has been proved, eg. for the inspection of medical foams and cold-forged parts.

Literaturverzeichnis

[Bey96] Beyerer, J.: Modellgestützte Sichtprüfung spanend bearbeiteter Oberflächen. In: *Technisches Messen* 63 (1996), Nr. 5, S. 182–190

[BHR81] Burt, P. J. ; Hong, T. H. ; Rosenfeld, A.: Segmentation and Estimation of Image Region Properties Through Cooperative Hierarchial Computation. In: *IEEE Trans. Systems, Man, and Cybernetics* 11 (1981), December, Nr. 12, S. 802–808

[BNS98] Backer, S. D. ; Naud, A. ; Scheunders, P.: Non-linear dimensionality reduction techniques for unsupervised feature extraction. In: *Pattern Recognition Letters* 19 (1998), S. 711–720

[Bre02] Bresch, M.: Optimizing filter banks for supervised texture recognition. In: *Pattern Recognition* 35 (2002), April, Nr. 4, S. 783–790

[Bri50] Brier, G.: Verification of Forecasts Expressed in Terms of Probability. In: *Monthly Weather Report* 78 (1950), January, Nr. 1, S. 1–3

[Bro66] Brodatz, P.: *Textures: A Photographic Album for Artists and Designers.* New York : Dover, 1966

[Can86] Canny, J.: A Computational Approach to Edge Detection. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 8 (1986), November, Nr. 6, S. 679–698

[CC96] Chen, C. C. ; Chen, D. C.: Multi-resolution Gabor filter in texture analysis. In: *Pattern Recognition Letters* 17 (1996), S. 1069–1076

[CH80] Conners, R. W. ; Harlow, C. A.: A Theoretical Comparison of Texture Algorithms. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 2 (1980), May, Nr. 3, S. 204–222

[CJ83] Cross, G. R. ; Jain, A. K.: Markov Random Field Texture Models. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 5 (1983), January, Nr. 1, S. 25–39

[CJ00] Clausi, D. A. ; Jernigan, M. E.: Designing Gabor filters for optimal texture separability. In: *Pattern Recognition* 33 (2000), S. 1835–1849

[CST00] Christianini, N. ; Shawe-Taylor, J.: *An Introduction to Support Vector Machines.* Cambridge University Press, 2000

[Das91] Dasarathy, B. V.: *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.* Los Alamitos : IEEE Computer Society Press, 1991

[Dau85] Daugman, J. G.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. In: *J. Opt. Soc. Am. A* 2 (1985), July, Nr. 7, S. 1160–1169

[Dau88] Daugman, J. G.: Complete Discrete 2-D Gabor Transform by Neural Networks for Image Analysis and Compression. In: *IEEE Trans. on Acoustics, Speech, and Signal Processing* 36 (1988), July, Nr. 7, S. 1169–1179

[DH73] Duda, R. O. ; Hart, P. E.: *Pattern Classification and Scene Analysis.* New York : Wiley, 1973

[DSG90] Djouadi, A. ; Snorrason, Ö. ; Garber, F. D.: The Quality o Training-Sample Estimates of the Bhattacharyya Coefficient. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 12 (1990), January, Nr. 1, S. 92–97

[Fuk90] Fukunaga, K.: *Introduction to Statistical Pattern Recognition.* New York : Academic Press, 1990

[Gol89] Goldberg, D.: *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, 1989

[GW92] Gonzalez, R. ; Woods, R.: *Digital Image Processing.* Addison-Wesley, 1992

[Hak88] Haken, H.: *Information and Self-Organization: A Macroscopic Approach to Complex Systems.* Berlin : Springer-Verlag, 1988

[Hän01] Hänsler, E.: *Statistische Signale.* 3. Auflage. Berlin : Springer-Verlag, 2001

[Har79] Haralick, R. M.: Statistical and Structural Approaches to Texture. In: *Proceedings of the IEEE* 67 (1979), May, Nr. 5, S. 786–804

[HFT96] Hamamoto, Y. ; Fujumoto, Y. ; Tomita, S.: On the Estimation of a Covariance Matrix in Designing Parzen Classifiers. In: *Pattern Recognition* 29 (1996), Nr. 10, S. 1751–1759

[HG84] Horner, J. ; Gianino, P: Phase-only matched filtering. In: *Applied Optics* 23 (1984), March, Nr. 6, S. 812–816

[HL96] Hoffbeck, J. P. ; Landgrebe, D. A.: Covariance Matrix Estimation and Classification With Limited Training Data. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 18 (1996)

[HO96] Hansen, N. ; Ostermeier, A.: Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation. In: *Proceedings of the 1996 IEEE Intern. Conf. on Evolutionary Computation (ICEC '96)*, 1996, S. 312–317

[HO97] Hansen, N. ; Ostermeier, A.: Convergence Properties of Evolution Strategies with the Derandomized Covariance Matrix Adaptation: The $(\mu/\mu_I, \lambda)$-CMA-ES. In: *Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT '97)*, 1997, S. 650–654

[HOP+95] Harwood, D. ; Ojala, T. ; Pietikäinen, M. ; Kelman, S. ; Davis, L.: Texture classification by center-symmetric autocorrelation, using Kullback discrimination of distributions. In: *Pattern Recognition Letters* 16 (1995), January, S. 1–10

[HSD73] Haralick, R. M. ; Shanmugam, K. ; Dinstein, I.: Textural Features for Image Classification. In: *IEEE Trans. Systems, Man, and Cybernetics* 3 (1973), November, Nr. 6, S. 610–621

[Hüb98] Hübner, W.: *Erzeugung genetischer Programme für die Texturanalyse*, Lehrstuhl für Technische Elektronik an der Friedrich-Alexander Universität Erlangen-Nürnberg, Diplomarbeit, 1998

[JAH97a] Jacquelin, C. ; Aurengo, A. ; Hejblum, G.: Evolving Descriptors for Texture Segmentation. In: *Pattern Recognition* 30 (1997), Nr. 7, S. 1069–1079

[Jäh97b] Jähne, B.: *Digitale Bildverarbeitung.* 4. Auflage. Springer-Verlag, 1997

[JF91] Jain, A. ; Farrokhnia, F.: Unsupervised Texture Segmentation Using Garbor Filters. In: *Pattern Recognition* 24 (1991), S. 1167–1186

[JK96] Jain, A. K. ; Karu, K.: Learning Texture Discrimination Masks. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 18 (1996), S. 195–205

[Jul75] Julesz, B.: Experiments in the Visual Perception of Texture. In: *Scientific American* 232 (1975), S. 34–44

[JZ97] Jain, A. ; Zongker, D.: Feature Selection: Evaluation, Application, and Small Sample Performance. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 10 (1997), February, Nr. 2, S. 153–158

[Kit78] Kittler, J.: Feature Set Search Algorithms. In: Chen, C. H. (Hrsg.): *Pattern Recognition and Signal Processing.* Alphen aan den Rijn, Netherlands : Sijthoff and Noordhoff, 1978, S. 41–60

[Koh97] Kohonen, T.: *Self-Organizing Maps.* 2. Berlin : Springer, 1997 (Information Sciences)

[Kön95] König, A.: *Neuronale Strukturen zur sichtgestützten Oberflächeninspektion von Objekten in industrieller Umgebung*, Technische Hochschule Darmstadt, Institut für Mikroelektronik, Diss., 1995

[KS97] Kursawe, F. ; Schwefel, H. P.: Optimierung mit Evolutionären Algorithmen. In: *atp Automatisierungstechnische Praxis* 37 (1997), Nr. 9, S. 10–17

[KTN98] Köppen, M. ; Teunis, M. ; Nickolay, B.: A Framework for the Evolutionary Generation of 2D-Lookup Based Texture Filters. In: *Proc. Iizuka 1998.* Iizuka, Japan, 1998, S. 965–970

[Kul59] Kullback, S.: *Information Theory and Statistics.* New York : John Wiley, 1959

[Lam98] Lambert, G.: *Echtzeit-Bildverarbeitung zur Qualitätssicherung bei texturierten Oberflächen unter besonderer Berücksichtigung strukturierter Texturen*, TU Darmstadt, Diss., 1998

[LGAD99] Lerner, B. ; Guterman, H. ; Aladjem, M. ; Dinstein, I.: A Comparative Study of Neural Network based Feature Extraction Paradigms. In: *Pattern Recognition Letters* 20 (1999), S. 7–14

[MCKV97] Mallet, Y. ; Coomans, D. ; Kautsky, J. ; Vel, O. D.: Classification Using Adaptive Wavelets for Feature Extraction. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 19 (1997), October, Nr. 10, S. 1058–1066

[Mic92] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, 1992 (Artificial Intelligence)

[MJ95] Mao, J. ; Jain, A. K.: Artificial neural networks for feature extraction and multivariate data projection. In: *IEEE Trans. Neural Networks* 7 (1995), S. 296–317

[MP97] Melchior, K. W. ; Pannekamp, J.: Adaptive Farbtexturanalyse mit neuronalen Netzen. In: *at Automatisierungstechnik* 12 (1997), Dezember, S. 594–598

[MP00] Mirmehdi, M. ; Petrou, M.: Segmentation of Color Textures. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 22 (2000), Nr. 2, S. 142–159

[MPO00] Mäenpää, T. ; Pietikäinen, M. ; Ojala, T.: Texture Classification by Multi-Predicate Local Binary Pattern Operators. In: *Proc. 15th International Conference on Pattern Recognition, Barcelona, Spain* Bd. 3, 2000, S. 951–954

[MS89] Mozer, M. ; Smolensky, P.: Skeletonization: A technique for trimming the fat from a network via relevance assessment. In: Touretzky, D. S. (Hrsg.): *Advances in Neural Information Processing Systems 1.* Morgan Kaufmann, 1989, S. 107–115

[NF77] Narendra, P. M. ; Fukunaga, K.: A Branch and Bound Algorithm for Feature Subset Selection. In: *IEEE Trans. Computers* 26 (1977), S. 917–922

[OD92] Ohanian, P. P. ; Dubes, R. C.: Performance Evaluation for Four Classes of Textural Features. In: *Pattern Recognition* 25 (1992), S. 819–833

[OPH96] Ojala, T. ; Pietikäinen, M. ; Harwood, D.: A Comparative Study of Texture Measures with Classification Based on Feature Distributions. In: *Pattern Recognition* 29 (1996), S. 51–59

[OPM02] Ojala, T. ; Pietikäinen, M. ; Mäenpää, T.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 24 (2002), July, Nr. 7, S. 971–987

[OVOP01] Ojala, T. ; Valkealahti, K. ; Oja, E. ; Pietikäinen, M: Texture discrimination with multidimensional distributions of signed gray-level differences. In: *Pattern Recognition* 34 (2001), S. 727–739

[Pan98] Pannekamp, J.: Produktdesign mit evolutionären Algorithmen. In: *Neue Methoden für Optimierung und Design*. Stuttgart : Fraunhofer IPA-/IAO-Technologieforum, Februar 1998, S. 99–109

[PH03] Pannekamp, J. ; Hüttel, M.: Selbstadaptive Fehlerdetektion in texturierten Oberflächen. In: *Oberflächenmesstechnik* Bd. 1806. Düsseldorf : VDI, 2003, S. 233–238

[PM00] Pannekamp, J. ; Melchior, K.: Augenmerk auf Ebenmäßigkeit, Qualitätsprüfung von Kollagenschwämmen. In: *QZ Qualität und Zuverlässigkeit* (2000), October, Nr. 10, S. 1304–1305

[PMW00] Pannekamp, J. ; Melchior, K. ; Westkämper, E.: Adaptex: Adaptive Bewertung texturierter Oberflächen. In: *Bildverarbeitung im industriellen Einsatz* Bd. 1572. Düsseldorf : VDI Verein Deutscher Ingenieure, October 2000, S. 199–206

[PNK94] Pudil, P. ; Novovicova, J. ; Kittler, J.: Floating search methods in feature selection. In: *Pattern Recognition Letters* 15 (1994), November, S. 1119–1125

[POX00] Pietikäinen, M. ; Ojala, T. ; Xu, Z.: Rotation-invariant texture classification using feature distributions. In: *Pattern Recognition* 33 (2000), S. 43–52. – http://www.ee.oulu.fi/research/imag/texture/

[PS98] Pannekamp, J. ; Schmutz, M.: Methodology for Evaluation and Selection of Features. In: Burkhardt, H. (Hrsg.): *Workshop on Texture Analysis 1998 (WTA '98)*, Albrecht-Ludwigs-Universität Freiburg, 1998, S. 151–164

[PSK98] Pannekamp, J. ; Schmidberger, E. ; Kischkat, R.: Using the Karhunen-Loeve Expansion for Feature Extraction on Small Sample Sets. In: *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (IECON '98)* Bd. 3, 1998, S. 1582–1586

[PTCH97] Pichler, O. ; Teuner, A. ; Conde, J. E. S. ; Hosticka, B. J.: Rotations- und skaleninvariante Textursegmentierung mit Mehrkanalfilterung. In: *19. DAGM-Symposium „Mustererkennung 1997“* (1997), S. 79–86

[PTVF92] Press, W. ; Teukolsky, S. ; Vetterling, W. ; Flannery, B.: *Numerical Recipes in C*. Second. Cambridge University Press, 1992

[PW01] Pannekamp, J. ; Westkämper, E.: Oberflächenstrukturbeurteilung mit Bildverarbeitung. In: Breitmeier, U. (Hrsg.): *Oberflächenprofilmeßtechnik*. Düsseldorf : VDI Wissensforum, September 2001

[PW03a] Pannekamp, J. ; Westkämper, E.: Generating Rotation-Invariant Texture Features by Randomization of Operator Orientation. In: Michaelis, B. (Hrsg.) ; Krell, G. (Hrsg.): *Pattern Recognition* Bd. 2781. Berlin : Springer, September 2003, S. 515–522

[PW03b] Pannekamp, J. ; Westkämper, E.: Improving rotation-invariance for texture classification through randomization of operator orientation. In: Chen, K. (Hrsg.): *Proceedings of JCIS 2003*. Durham, North Carolina, September 2003, S. 773–776

[Rau97] Raudys, S.: On Dimensionality, Sample Size, and Classification Error of Nonparametric Linear Classification Algorithms. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 19 (1997), June, Nr. 6, S. 667–671

[Rec94] Rechenberg, I.: *Werkstatt Bionik und Evolutionstechnik*. Bd. 1: *Evolutionsstrategie '94*. Stuttgart : Frommann-Holzboog, 1994

[RH99] Randen, T. ; Husoy, J.: Filtering for Texture Classification: A Comparative Study. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 21 (1999), April, Nr. 4, S. 291–310. – http://www.ux.his.no/~tranden/

[RW90] Reed, T. ; Wechsler, H.: Segmentation of Textured Images and Gestalt Organization Using Spatial/Spatial-Frequency Representations. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 12 (1990), January, Nr. 1, S. 1–12

[SB98] Schael, M. ; Burkhardt, H.: Error Detection on Textures using Invariant Grey-Scale Features. In: Burkhardt, H. (Hrsg.): *Workshop on Texture Analysis 1998, (WTA '98)*, Albrecht-Ludwigs-Universität Freiburg, 1998, S. 165–179

[Sch96] Schürmann, J.: *Pattern classification: a unified view of statistical and neural approaches*. John Wiley, 1996

[SM95] Schulz-Mirbach, H.: *Anwendung von Invarianzprinzipien zur Merkmalsgewinnung in der Mustererkennung*, Technische Universität Hamburg-Harburg, Diss., 1995

[SS88] Siedlecki, W. ; Sklansky, J.: On Automatic Feature Selection. In: *International Journal on Pattern Recognition and Artificial Intelligence* 2 (1988), Nr. 2, S. 197–220

[TPH95] Teuner, A. ; Pichler, O. ; Hosticka, B. J.: Unsupervised Texture Segmentation of Images Using Tuned Matched Gabor Filters. In: *IEEE Trans. Image Processing* 4 (1995), S. 863–870

[Tru79] Trunk, G. V.: A Problem of Dimensionality: A Simple Example. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 1 (1979), S. 306–307

[Uni02] Universität Stuttgart: Institut für Parallele und Verteilte Systeme (IPVS): SNNS - Stuttgart Neural Network Simulator. In: *http://www-ra.informatik.uni-tuebingen.de/SNNS/* (2002)

[Uns86] Unser, M.: Sum and Difference Histograms for Texture Classification. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 8 (1986), January, Nr. 1, S. 118–125

[Wag97] Wagner, T.: *Automatische Konfiguration von Bildverarbeitungssystemen.* Aachen : Shaker, 1997

[Wag99] Wagner, T.: Texture Analysis. In: Jähne, B. (Hrsg.) ; Haussecker, H. (Hrsg.) ; Geissler, P. (Hrsg.): *Computer Vision and Applications* Bd. 2. Boston : Academic Press, 1999

[WDR76] Weszka, J. S. ; Dyer, C. R. ; Rosenfeld, A.: A Comparative Study of Texture Measures for Terrain Classification. In: *IEEE Trans. Systems, Man, Cybernetics* 6 (1976), April, S. 269–285

[Wer58] Wertheimer, M.: Principles of Perceptual Origanization. In: Beardslee, D. C. (Hrsg.) ; Wertheimer, M. (Hrsg.): *Readings in Perception.* Princeton NJ : Van Nostrand, 1958, S. 115–135

[WKPR99] Westkämper, E. ; Kischkat, R. ; Pannekamp, J. ; Radons, G.: Configuration Support for Flexible Production Systems. In: *Production Engineering* 6 (1999), Nr. 2, S. 101–104

[Wol97] Wolfram, S.: *Das Mathematica-Buch.* 3. Auflage. Bonn : Addison-Wesley-Longman, 1997

[YY93] Yu, B. ; Yuan, B.: A More Efficient Branch and Bound Algorithm for Feature Selection. In: *Pattern Recognition* 26 (1993), Nr. 6, S. 883–889

[Zel94] Zell, A.: *Simulation Neuronaler Netze.* Bonn : Addison-Wesley, 1994

Lebenslauf

Persönliche Daten

Name:	Jens Pannekamp
Geburtsdatum:	20. Oktober 1969
Geburtsort:	Ulm

Schule

76–89	Grundschule und Gymnasium Ulm-Wiblingen

Zivildienst

05/89–09/90	Individuelle Schwerbehindertenbetreuung beim Arbeiter-Samariter-Bund Ulm

Studium

10/90–09/96	Studium der Physik, Universität Ulm
09/92–04/93	Studienaufenthalt an der University of Leeds (GB)
09/94–08/95	Diplomarbeit: „Thermische Umverteilung von optisch erzeugten Ladungsträgern in Halbleiter-Quantentopfstrukturen“
08/95–05/96	Fulbright-Stipendiat an der Universität Knoxville (USA)

Beruf

10/96–10/99	Wissenschaftlicher Mitarbeiter am Institut für Industrielle Fertigung und Fabrikbetrieb, Universität Stuttgart
Seit 10/99	Wissenschaftlicher Mitarbeiter am Fraunhofer Institut für Produktionstechnik und Automatisierung, Stuttgart
Seit 03/04	Leiter der Fachgruppe „Digitale Signalanalyse“ am Fraunhofer Institut für Produktionstechnik und Automatisierung, Stuttgart