

Automatischer Entwurf und Autokonfiguration von Bildverarbeitungssystemen für die industrielle Oberflächeninspektion

vom
Fachbereich Elektrotechnik und Informationstechnik
der Universität Kaiserslautern zur Verleihung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation

von
Dipl.-Ing. (FH) Stefanie Peters
geb. in Lima (Peru)

D 386

Eingereicht am:
Tag der mündlichen Prüfung:
Dekan des Fachbereichs:

31. März 2009
31. März 2010
Prof. Dipl.-Ing. Dr. Gerhard Fohler

Promotionskommission
Vorsitzender:
Berichterstattende:

Prof. Dr. Remigius Zengerle
Prof. Dr.-Ing. Andreas König
Prof. Dr.-Ing. Bärbel Mertsching

Zusammenfassung

Die industrielle Oberflächeninspektion und insbesondere die Defekterkennung ist ein wichtiges Anwendungsgebiet für die automatische Bildverarbeitung (BV). Diesbezüglich wurde der aktuelle Stand gemäß dem VDMA recherchiert und wiedergegeben. Für den Entwurf und die Konfiguration der entsprechenden Softwaresysteme, in der Regel anwendungsspezifische Einzellösungen, werden im industriellen Umfeld zumeist entweder firmeneigene Bildverarbeitungsbibliotheken, kommerzielle oder freie Toolboxes verwendet. In der Regel beinhalten diese u.a. Standardalgorithmen der Bildverarbeitung in modularer Form, z. B. Filter- oder Schwellwertoperatoren. Die einzelnen BV-Methoden werden in der Regel nach dem Prinzip der visuellen Programmierung in einer grafischen Entwicklungsumgebung ausgewählt und zu einer BV-Kette bzw. einem -Graph zusammengesetzt. Dieses Prinzip ermöglicht es auch einem Programmierunkundigen, BV-Systeme zu erstellen und zu konfigurieren. Eine gewisse Grundkenntnis der Methoden der Bildverarbeitung ist jedoch notwendig. Je nach Aufgabenstellung und Erfahrung des Systementwicklers erfordern manueller Entwurf und Konfiguration eines BV-Systems erheblichen Zeiteinsatz. Diese Arbeit beschäftigt sich mit automatischen Entwurfs-, Konfigurations- und Optimierungsmöglichkeiten dieser modularen BV-Systeme, die es auch einem ungeübten Endnutzer ermöglichen, adäquate Lösungen zu generieren mit dem Ziel, ein effizienteres Entwurfswerkzeug für Bildverarbeitungssysteme mit neuen und verbesserten Eigenschaften zu schaffen. Die Methodenauswahl und Parameteroptimierung reicht von der Bildvorverarbeitung und -verbesserung mittels BV-Algorithmen bis hin zu ggf. eingesetzten Klassifikatoren, wie Nächste-Nachbar-Klassifikator (NNK) und Support-Vektor-Maschinen (SVM) sowie verschiedenen Bewertungsfunktionen.

Der flexible Einsatz verschiedener Klassifikations- und Bewertungsmethoden ermöglichte einen automatischen problemspezifischen Entwurf und die Optimierung des BV-Systems für Aufgaben der Fehlerdetektion und Texturanalyse für 2d-Bilder, sowie die Trennung von Objekten und Hintergrund für 2d- und 3d-Grauwertbilder. Für die Strukturoptimierung des BV-Systems wurden Genetische Algorithmen (GA) und Genetische Programmierung (GP) verwendet. Der optimierte Einsatz bewährter Methoden zur Bildvorverarbeitung und Texturanalyse hat generell zu besseren Lösungen geführt als die Verwendung einer Parametrierung mit Standardwerten oder ad hoc Werten. Die hier eingeführte Mehrzieloptimierung mittels eines akkumulativen Gütemaßes ermöglichte zusätzlich eine Berücksichtigung der für den industriellen Einsatz wichtigen Laufzeit der BV-Lösung. Der Einsatz von Schwarmoptimierung für BV-Systeme mit einer festen Struktur, d. h. einer festgelegten Methodenauswahl, hat im Vergleich zu GA/GP bessere Ergebnisse für die

verwendeten Testdatensätze für die Defekterkennung und Texturklassifikation geliefert. Die Ergebnisse wurden zudem innerhalb einer kürzeren Optimierungszeit erreicht. Die Verwendung von SVM für die Klassifikation und Bewertung ergab deutlich höhere Klassifikationsraten als bei einer üblichen kNN-Klassifikation. Es hat sich gezeigt, dass die Parametrierung der Klassifikatoren sowohl von den Eingangsdaten, als auch von den ausgewählten Bildverarbeitungsmethoden abhängt. Aus diesem Grund hat die in die Gesamtoptimierung integrierte Adaption der Klassifikatorparameter mittels GA/GP und PSO im Vergleich zu einer Rastersuche nach dem Stand der Technik Vorteile bezüglich der Optimierungszeit gebracht. Das in dieser Arbeit geschaffene Werkzeug wird in der nächsten Phase in eine geeignete Oberfläche integriert und für einen komfortablen Einsatz für den Entwurf industrieller BV-Systeme nutzbar gemacht.

Vorwort

Die vorliegende Arbeit entstand während meiner Zeit als Stipendiatin in der Abteilung Bildverarbeitung des Fraunhofer Instituts für Techno- und Wirtschaftsmathematik und am Lehrstuhl Integrierte Sensorsysteme im Fachbereich Elektro- und Informationstechnik der Technischen Universität Kaiserslautern in den Jahren 2004 bis 2009.

Herrn Prof. Dr.-Ing. Andreas König, Inhaber des Lehrstuhls für Integrierte Sensorsysteme gilt mein besonderer Dank für den Vorschlag des Themas, die Förderung und das Interesse an meiner Arbeit. Seine Unterstützung, Ratschläge und Diskussionen haben wesentlich zum Gelingen dieser Arbeit beigetragen.

Bei meinen Kollegen am Fraunhofer Institut möchte ich mich für die freundliche Unterstützung und ein gutes Arbeitsklima bedanken.

Ich danke meinen Eltern und meinem Bruder für ihre Unterstützung, ihre Geduld und dass sie immer ein offenes Ohr für meine Angelegenheiten hatten.

Kaiserslautern, März 2010

Stefanie Peters

Inhaltsverzeichnis

1 Einleitung	1
1.1 Zielsetzung dieser Arbeit.....	3
1.2 Gliederung der Arbeit.....	6
2 Visuelle Inspektionssysteme in der industriellen Bildverarbeitung.....	9
2.1 Sensorik.....	12
2.1.1 Analoge Flächenkameras.....	12
2.1.2 Digitale Flächen- oder Zeilenkameras	12
2.1.3 Intelligente Kameras und „Embedded Systems“	13
2.1.4 Objektiv	14
2.1.5 Automatisierung und Optimierung bei Kameras und Sensorik.....	14
2.2 Beleuchtung.....	14
2.2.1 Lichtquellen	15
2.2.2 Beleuchtungsmethoden.....	15
2.2.3 Automatisierung und Optimierung der Beleuchtung.....	17
2.3 Bildverarbeitungssoftware.....	17
2.3.1 Khoros/Cantata	18
2.3.2 HDevelop	19
2.3.3 NeuroCheck	20
2.3.4 QuickCog	20
2.4 Optimierung und Automatisierung des Systementwurfs	
für visuelle Inspektionssysteme.....	21
2.4.1 Optimierung und Autokonfiguration einzelner BV-Methoden.....	22
2.4.2 Automatische Mustererkennung und -klassifizierung.....	22
2.4.3 Automatischer Entwurf und Autokonfiguration von BV-Systemen.....	23
2.4.4 Kritikpunkte an den aufgeführten Ansätzen	25
2.5 Zusammenfassung	26
3 Bildvorverarbeitung und Texturanalyse	27
3.1 Grundlagen der Texturanalyse.....	27
3.2 Verfahren der Texturanalyse.....	30
3.3 Auswahl und Parametrierung von Methoden für die Texturklassifikation und Defekterkennung	32
3.4 Spezielle Methoden der Bildvorverarbeitung und Texturanalyse.....	34
3.4.1 Methoden der Bildvorverarbeitung und ihre Parameter.....	34
3.5 Bestimmung von Texturmerkmalen.....	35
3.5.1 Methoden der Berechnung von Texturmerkmalen	35
3.6 Nichtlineare und orientierte Kerne (NLOK).....	42
3.6.1 Kodierung der lokalen Orientierung	42
3.6.2 Lokale Autokorrelationsfunktion höherer Ordnung.....	43
3.6.3 Dominante Strukturen in einer lokalen Umgebung.....	44
3.6.4 Zurückweisung nicht dominanter Strukturen.....	44
3.6.5 Texturmerkmale	45
3.6.6 Varianten von NLOK.....	45
3.6.7 Vergleich von NLOK mit anderen Texturoperatoren	46
3.7 Merkmalskombination und -selektion	46

3.8	Übersicht über die Methodenparameter der Texturanalysemethoden.....	46
3.9	Zusammenfassung.....	48
4	Bewertung eines Bildverarbeitungssystems.....	49
4.1	Filter- und Wrappermodelle für die Bewertung von BV-Systemen	49
4.2	Bildähnlichkeit	50
4.2.1	Bildvergleich bei Binärbildern.....	50
4.2.2	Summe der quadrierten Grauwertdifferenzen	50
4.2.3	Gegenseitige Information oder Mutual Information.....	50
4.3	Merkmalsbewertung	51
4.4	Klassifikation	51
4.4.1	Schwellwertklassifikator	51
4.4.2	Abstandsklassifikatoren	52
4.4.3	Nächste-Nachbar-Klassifikator.....	52
4.4.4	Support-Vektor-Maschinen	53
4.5	Bewertung des Klassifikationsergebnisses.....	54
4.5.1	Maximierung der Klassifikationsrate R_{OK} der korrekt klassifizierten Bildregionen.....	54
4.5.2	Minimierung der erwarteten Kosten.....	54
4.6	Kombination verschiedener Bewertungsmethoden.....	54
4.7	Vergleichende Untersuchungen	55
4.7.1	Bildähnlichkeit zwischen zwei Grauwertbildern	55
4.7.2	Ergebnisse und Diskussion.....	56
4.7.3	Sensitivität der Klassifikatorparameter bei der Texturklassifikation.....	58
4.7.4	Ergebnisse und Diskussion.....	60
4.8	Zusammenfassung.....	64
5	Optimierung von Bildverarbeitungssystemen mittels Evolutionärer Algorithmen und Schwarmintelligenz.....	65
5.1	Aufgaben der Optimierung für den Entwurf und die Konfiguration eines BV-Systems....	65
5.2	Evolutionäre Algorithmen	68
5.2.1	Genetische Algorithmen.....	69
5.2.2	Genetische Programmierung	71
5.2.3	Evolutionsstrategien.....	73
5.2.4	Partikelschwarmoptimierung.....	74
5.2.5	Ameisenalgorithmus	77
5.3	Aspekte der verwendeten Architektur bezüglich der Optimierung.....	77
5.3.1	Initialisierung einer Population	77
5.3.2	Elitismus	77
5.3.3	Diversität.....	78
5.3.4	Bewertungsfunktionen und Bestrafungselemente	78
5.4	Optimierung mit GA und GP.....	78
5.4.1	Selektion.....	78
5.4.2	Rekombination.....	79
5.4.3	Mutation.....	79
5.5	Optimierung mittels PSO	80
5.6	Experimente	80
5.6.1	Ergebnisse und Diskussion.....	81
5.7	Zusammenfassung.....	84

6 Konzept und Architektur einer Umgebung für den automatisierten Entwurf und die Konfiguration eines Bildverarbeitungssystems	85
6.1 Verwendung von Vorwissen für die Optimierung.....	86
6.1.1 Sammeln von Vorwissen	87
6.1.2 Einbringen von Vorwissen in die Optimierung.....	87
6.2 Nichtlinearer Verarbeitungsablauf.....	89
6.3 Multispektrale und multidimensionale Eingangssignale.....	90
6.4 Multisensorik.....	90
6.5 Anforderungen an eine Umgebung für den automatischen Entwurf und die Konfiguration eines BV-Systems und des Optimierungsverfahrens.....	90
6.5.1 Hardware und Sensorik	90
6.5.2 Algorithmik	90
6.5.3 Datenfluss und Repräsentation	91
6.5.4 Optimierung.....	91
6.6 Umsetzung der Anforderungen an die Architektur.....	92
6.7 Zusammenfassung	94
7 Anwendungen auf die Texturanalyse und 2d/3d-Segmentierung.....	95
7.1 Texturklassifikation mit C-PSO und GA.....	95
7.1.1 Elitismus und Diversität	97
7.1.2 Populationsgröße und Laufzeit der Optimierung.....	99
7.2 Detektion von Texturfehlern mit C-PSO und GA	99
7.2.1 Synthetische, schwach markierte Texturdefekte.....	100
7.2.2 Applikationsnahe Daten aus der Inspektion von gegerbtem Leder	101
7.2.3 Textiltexturdefekte der Datenbank TILDA	103
7.2.4 Aufbau des Experiments	104
7.2.5 Ergebnisse und Diskussion	105
7.3 Mehrzieloptimierung mit PSO	107
7.4 Automatische Defektdetektion mittels ikonischer Bildsegmentierung	107
7.4.1 Exemplarische Ergebnisse der Bildsegmentierung.....	109
7.5 Optimierte Texturklassifikation	114
7.5.1 Aufbau des Experiments	114
7.5.2 Ergebnisse und Diskussion	115
7.6 Segmentierung von 3d-Volumendaten.....	117
7.6.1 Ergebnisse der automatischen Segmentierung von dünnen und dicken Fasern	119
7.7 Diskussion der Ergebnisse.....	120
8 Diskussion und Zusammenfassung der Ergebnisse, Ausblick.....	123
Abkürzungen und Formelzeichen.....	125
Allgemeine Abkürzungen.....	125
Evolutionäre Algorithmen und Schwarmintelligenz.....	126
Bewertung und Klassifikation.....	126
Implementierte Texturoperatoren und ihre Merkmale.....	127
Weitere Texturoperatoren	127
Allgemeine Formelzeichen.....	128
Evolutionäre Algorithmen und Schwarmintelligenz.....	129
Bewertung und Klassifikation.....	129

Industrielle Bildverarbeitung: Software, Firmen und Messen	131
Frei verfügbare Programmierbibliotheken	131
Kommerzielle Bildverarbeitungssoftware	131
Ausgewählte Bildverarbeitungsfirmen	132
Ausgewählte Bildverarbeitungsmessen	132
Anhang	133
A1 Implementierung in das Programmpaket Genesis	133
A1.1 GenesisOpt	135
A1.1.1 Optimierung	135
A1.1.2 Verarbeitungsmethoden.....	137
A1.1.3 Datenstrukturen	140
A1.1.4 Parametrierung der Optimierung.....	141
A1.1.5 Anwendungsbeispiele für die Initialisierung von GenesisOpt.....	141
A1.1.6 Grafische Oberfläche für GenesisOpt	144
A1.2 GenesisMSeg.....	144
A2 Laws' Filtermasken der Größe 5×5	147

Literaturverzeichnis

Lebenslauf

1 Einleitung

Sensorsignal- und insbesondere Bildverarbeitung findet eine breite Anwendung in vielen Bereichen des täglichen Lebens. Diese Systeme dienen der automatischen Aufnahme von Informationen, der Bestimmung spezieller Kennwerte, der Bewertung oder der Klassifikation. Mit Hilfe von biometrischen Systemen kann zum Beispiel die Identität eines Menschen mit sehr hoher Wahrscheinlichkeit festgestellt werden. Als Merkmale dienen hierbei gescannte Fingerabdrücke oder Bilder der Augen-Iris. In der Medizin werden unter Verwendung von Bildverarbeitung Zelltypen in mikroskopischen Aufnahmen klassifiziert; der Pathologe kann so bei seiner Entscheidungsfindung unterstützt werden. An einer Supermarktkasse wird der Preis eines Produkts anhand des Strichcodes mit Methoden der Optik und Bildverarbeitung automatisch ausgelesen. Fahrerassistenzsysteme unterstützen den Fahrer bei der Erkennung von Kennzeichen, der Verfolgung der Fahrspur oder bei der frühzeitigen Erkennung von Gefahrensituationen. Eine automatische Auswertung von Satellitenbilddaten kann zum Beispiel der Früherkennung von Waldbränden dienen, kann helfen, Bewuchs- und Anbaugelände zu katalogisieren oder für die militärische Aufklärung verwendet werden. Ein Großteil der Anwendungen findet jedoch im industriellen Umfeld, insbesondere in der Fertigung und Produktion (vgl. Abbildung 1-1) statt. Immer mehr Industriezweige erkennen und nutzen das Potenzial, Bildverarbeitung für die Qualitätskontrolle und -sicherung einzusetzen.

Für die Entwicklung geeigneter Bildverarbeitungssysteme erfordern diese sehr heterogenen und komplexen Anwendungen in einem dynamisch wachsenden Anwendungsfeld ein dementsprechend dynamisches und heterogenes Systemdesign. In der Regel werden solche Systeme manuell auf die einzelnen Problemstellungen zugeschnitten und von geschultem Personal entwickelt und optimiert. Zusätzlich ist in den letzten Jahren auch das Bedürfnis des Kunden nach konfigurierbaren, d. h. modularen, parametrierbaren Systemen gestiegen (vgl. Abbildung 1-2). Ein sinnvoller Einsatz dieser Systeme erfordert vom Kunden dann jedoch zumindest gewisse Grundkenntnisse im Bereich der Signal- und Bildverarbeitung. Der Umsatz durch rein anwendungsspezifische Systeme ist hingegen leicht zurückgegangen. Die aufgeführten Daten wurden aktuellen Marktstudien des VDMA (Verband Deutscher Maschinen- und Anlagenbauer e.V.) [91, 130, 131] entnommen, dem der Bereich „Industrielle Bildverarbeitung“ als Fachabteilung des VDMA-Verbandes „Robotik+Automation“ angehört. Die Ergebnisse dieser jährlich durchgeführten Marktstudien für das Jahr 2008 standen zum Zeitpunkt der Erstellung dieser Arbeit noch nicht zur Verfügung.

**Gesamtumsatz Industrielle Bildverarbeitung 2007 (Deutschland)
Anteil nach Branchen**

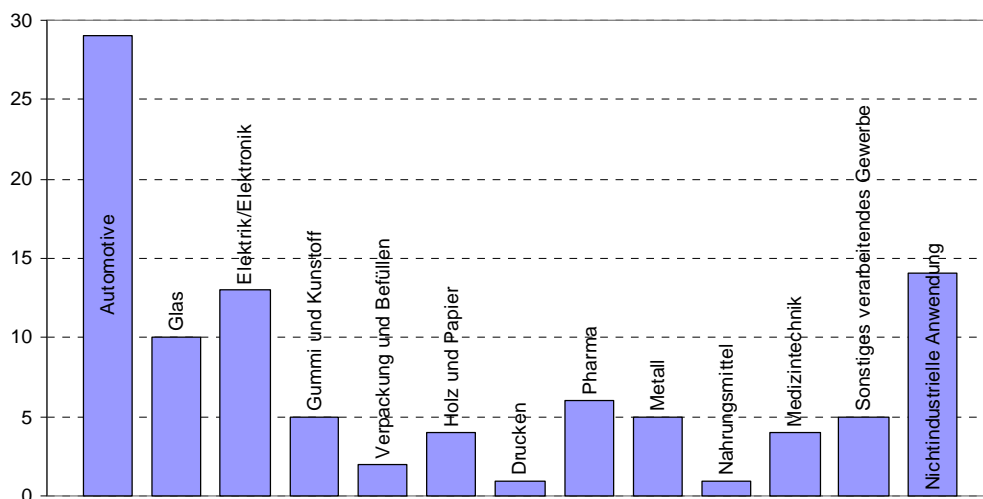


Abbildung 1-1: Branchenanteile am Gesamtumsatz der Industriellen Bildverarbeitung 2007 in Deutschland. Die auf ein Prozent Genauigkeit gerundeten Anteile wurden aktuellen Marktstudien des VDMA [91, 130, 131] entnommen.

**Umsatz von Bildverarbeitung nach Produkten
Veränderungen 2007/2006, 2006/2005 und 2005/2004**

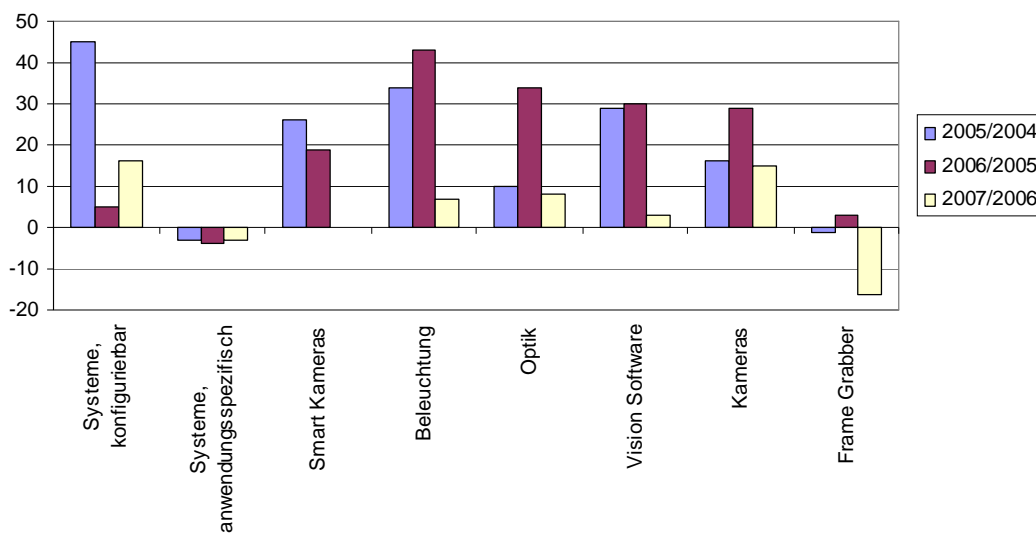


Abbildung 1-2: Veränderung der Anteile des Umsatzes der Industriellen Bildverarbeitung in Deutschland nach Produkten. Die auf ein Prozent Genauigkeit gerundeten Daten wurden aktuellen Marktstudien des VDMA [91, 130, 131] entnommen.

1.1 Zielsetzung dieser Arbeit

Das oberste Ziel dieser Arbeit ist die Entwicklung eines verbesserten Werkzeuges für den automatischen Entwurf, die Autokonfiguration und die Optimierung von heterogenen, modularen, algorithmischen Systemen für die Signalverarbeitung. Für den geschulten Systementwickler bis hin zum unbedarften Endnutzer werden Hilfsmittel aufgezeigt Signalverarbeitungssysteme zu entwickeln, zu optimieren und automatisch an spezielle Gegebenheiten anzupassen. Das Portfolio der verwendeten Methoden reicht von der Autokonfiguration und Optimierung einzelner Module bis hin zu dem kompletten Entwurf und der Optimierung eines Gesamtsystems.

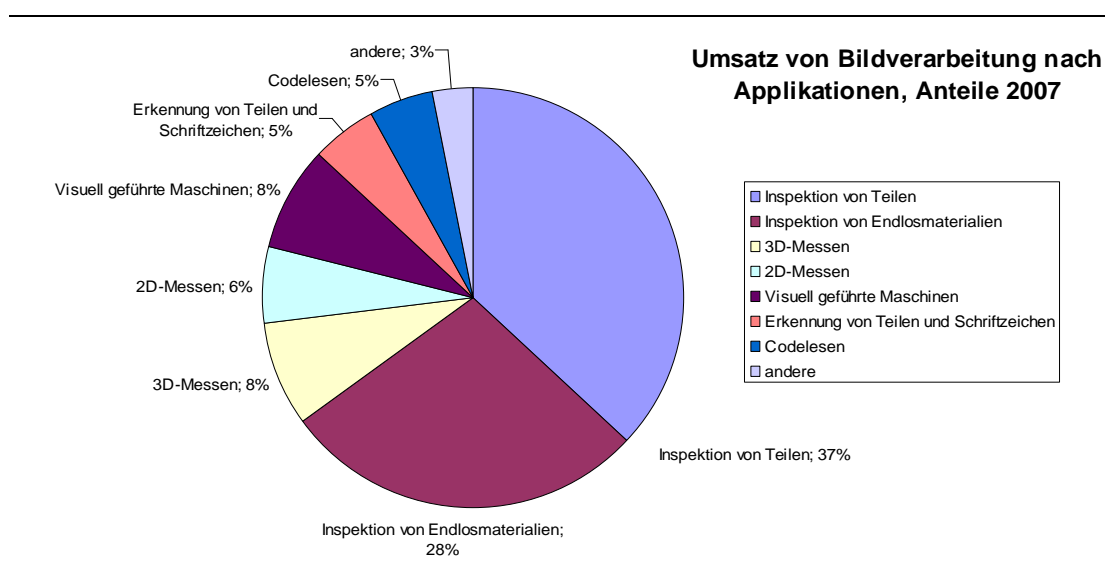


Abbildung 1-3: Anteile des Systemumsatzes der industriellen Bildverarbeitung in Deutschland nach Applikationen. Die auf ein Prozent Genauigkeit gerundeten Umsatzwerte wurden aktuellen Marktstudien des VDMA [91, 130, 131] entnommen.

Als praktische Anwendung dient ein Bildverarbeitungssystem für die automatische visuelle Kontrolle von Oberflächen im industriellen Einsatz. Besonderes Augenmerk wird in dieser Arbeit auf Systeme für die Segmentierung und Texturdefekterkennung geworfen. Gemäß der aktuellen Marktstudie des VDMA liegt der Applikationsanteil am Gesamtumsatz der Bildverarbeitung in Deutschland 2007 für die Inspektion von Teilen bei 37% und für die Inspektion von Endlosmaterialien bei 28% (vgl. Abbildung 1-3). Die Segmentierung (Inspektion von Teilen) und die Texturdefekterkennung (Inspektion von Endlosmaterialien) sind wichtige Werkzeuge für die Lösung solcher Inspektionsaufgaben. Abbildung 1-4 zeigt einen schematischen Aufbau eines industriellen Bildverarbeitungssystems für die Oberflächenprüfung.

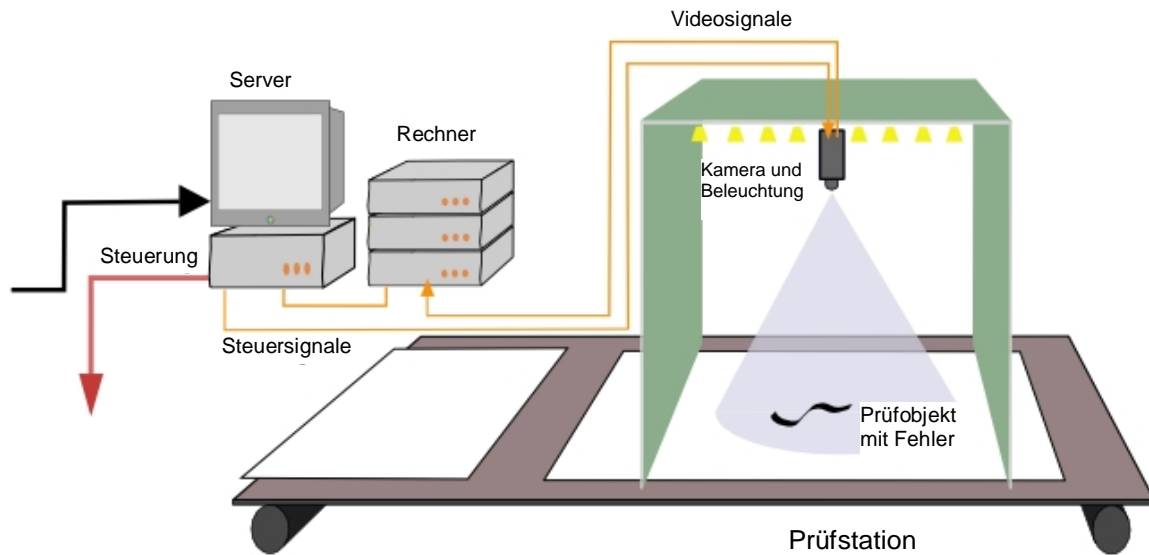


Abbildung 1-4: Schematische Darstellung eines visuellen Inspektionssystems für die industrielle Oberflächenprüfung. Mit geeigneter Hardware, d. h. Kamera und Beleuchtung, wird ein Bild von einem Prüfobjekt aufgenommen. Die aufgenommenen Bilder werden mit einem oder mehreren Rechnern verarbeitet, um die gestellte Prüfaufgabe zu lösen. Eine Server-Komponente übernimmt die Ansteuerung von Kamera und Beleuchtung sowie die Kommunikation mit dem Gesamtsystem, in das das Prüfsystem integriert ist.

Der Entwurf und die Parameterkonfiguration eines Bildverarbeitungssystems haben einen entscheidenden Einfluss auf dessen erreichbare Güte. Für einige Bereiche der Bildverarbeitung, wie zum Beispiel dem Lesen von Barcodes, existieren Standardlösungen von etablierten Bildverarbeitungsfirmen wie z. B. Stemmer, ISRA VISION SYSTEMS, NeuroCheck oder VITRONIC GmbH. In der Regel werden Bildverarbeitungssysteme für komplexe Prüfaufgaben jedoch mit erheblicher Mühe und Zeitaufwand von geschulten Ingenieuren manuell aus bestehenden oder neu erstellten Softwaremodulen zusammengesetzt und parametrisiert. Die Lösungsqualität dieser Einzelsysteme ist damit stark von der Erfahrung des Systementwicklers abhängig. Veränderungen in der Produktionsumgebung, wie z. B. variierende Defekte auf dem Prüfobjekt oder eine Alterung der Beleuchtung erfordern in der Regel eine Rekonfiguration oder im schlimmsten Fall eine Neuentwicklung des eingesetzten BV-Systems. Dies kann neben den Entwicklungs- und Anschaffungskosten zu nicht unerheblichen Folgekosten für die Wartung des Systems führen. Alternativ einsetzbare Lern- und Anpassungsmechanismen werden derzeit nur eingeschränkt verwendet, wie etwa für die Klassifikation oder Merkmalsselektion. Einige wenige kommerzielle Anbieter¹ versprechen so genannte „Ein-Klick“-Lösungen z. B. unter Verwendung neuronaler Netze [82], die mit geringem

¹ Softwarepakete Manto und Minos, Stemmer Imaging GmbH

Aufwand funktionsfähige Lösungen versprechen. Diese Lösung wird in der Regel jedoch als „Black-Box“ geliefert, ohne dass der Anwender nachträglich eine Möglichkeit der Manipulation oder Anpassung der Lösung hat.

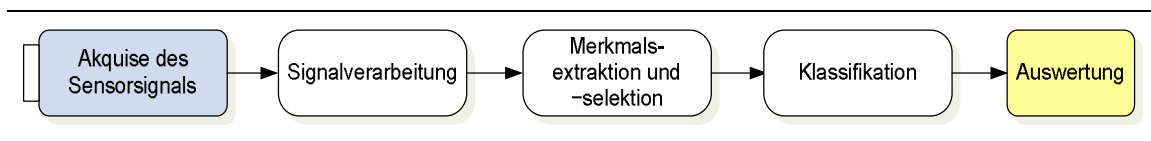


Abbildung 1-5: Schematische Darstellung einer typischen Signalverarbeitungskette.

Bei den Bildverarbeitungsaufträgen von kleinen und mittelständigen Unternehmen im Bereich der Oberflächeninspektion handelt es sich jedoch zumeist um Einzellösungen mit einer großen Vielfalt an Prüfaufgaben und Prüfobjekten bzw. Fehlertypen. Hauptziel dieser Arbeit ist es, für Anwendungen im Bereich der Zwei-Klassen-Segmentierung, d. h. Trennung von Objekt und Hintergrund, Texturanalyse, Texturklassifikation und Defekterkennung automatisch algorithmische anwendungsspezifische Lösungen mit minimalem menschlichen Aufwand zu entwickeln. Hauptaugenmerk liegt hierbei auf dem algorithmischen Bereich nach der Akquise des Sensorsignals bis hin zur Klassifikation (vgl. Abbildung 1-5). Methoden der automatischen Dimensionsreduktion werden in dieser Arbeit nicht explizit bzw. nur untergeordnet betrachtet.

Ein typischer Ansatz für die Entwicklung solch einer Einzellösung ist zunächst das bilddatenabhängige manuelle Erstellen einer geeigneten Bildverarbeitungskette mit bewährten und neu entwickelten BV-Methoden für die Bildvorverarbeitung und -verbesserung, Segmentierung, Merkmalsextraktion und Klassifikation.

Veröffentlichungen, wie z. B. zur Wahl eines Verfahrens zur automatischen Schwellwertbestimmung [96, 125, 132] oder die Bestimmung der optimalen Fenstergröße bei einer Texturklassifikation [112, 127] können bei der Selektion und Parametrierung der einzelnen BV-Methoden Anhaltspunkte geben. Das Leistungsvermögen unterschiedlicher Operatoren in vergleichenden Arbeiten [81, 133, 89] ist in der Regel allerdings abhängig von den gewählten Testdaten und zudem gibt es meist fixierte Methoden oder Parameterwerte bei der Bildvorverarbeitung oder der Klassifikation. Ein Teilziel dieser Arbeit ist eine Untersuchung der Abhängigkeit verschiedener BV-Methoden einer BV-Kette für die Texturklassifikation und Defekterkennung in Bezug auf die Methodenkombination und Parametrierung. Daraus folgend sollen optimale Methodenkombinationen automatisch ausgewählt und parametrierung werden.

Teilabschnitte einer manuell entwickelten BV-Kette können unter Verwendung von Optimierungsverfahren (z. B. Hill Climbing oder Genetische Algorithmen (GA) [41, 70]) automatisch konfiguriert werden. Beispiele hierfür sind die automatische Adaption von

Strukturelementen und morphologischen Operatoren mittels Genetischer Programmierung (GP) oder Genetischer Algorithmen für eine Trennung von Objekt und Hintergrund [3, 4, 49, 66, 114], die automatische Auswahl optimaler Filter für die Texturklassifikation [71] oder die automatische Merkmalsselektion geeigneter Merkmale für eine anschließende Klassifikation [113]. Ein weiteres Teilziel dieser Arbeit ist es zu untersuchen, inwieweit eine Parameteroptimierung einzelner Methoden deren Leistungsfähigkeit gegenüber einer naiven Standard-Parametrierung bzw. einer Rastersuche steigert.

Neben der Konfiguration und Optimierung soll ein Verfahren implementiert werden, welches u.a. auch für die Strukturoptimierung verwendbar ist. Eine viel versprechende Methodik für ein solches automatisch konfigurierbares System ist die Struktur- und Parameteroptimierung eines Bildverarbeitungsgraphen bestehend aus modularen Bildverarbeitungsoperatoren mittels Genetischer Algorithmen und Genetischer Programmierung der ISIS Gruppe der Los Alamos National Laboratories [50, 99, 100]. Ein Teilziel dieser Arbeit ist die Adaption dieser für die Satellitenbildverarbeitung entwickelten Methode für die industrielle Oberflächeninspektion. Unter anderem wird das Einbringen von Randbedingungen durch Mehrzieloptimierung betrachtet. Besonderes Augenmerk liegt hierbei auf der für industrielle Anwendungen in der Regel relevanten Laufzeit eines BV-Systems.

Auf der Suche nach schnelleren und/oder besseren Optimierungsverfahren wird in diesem Zusammenhang Partikelschwarmoptimierung (PSO) als alternative Optimierungsmethode betrachtet und analysiert. Mittels PSO konnten bei verschiedenen Optimierungsaufgaben (keine BV) vergleichbare bzw. sogar bessere Ergebnisse als bei Verwendung von GA/GP erzielt werden [144].

1.2 Gliederung der Arbeit

Im ersten Teil dieser Arbeit wird auf die relevanten Aspekte der industriellen Bildverarbeitung und den Stand der Technik für das Systemdesign und die Optimierung eingegangen. Kapitel 2 erläutert die Anforderungen an Bildverarbeitungssysteme für die industrielle Oberflächenprüfung und beschreibt deren Komponenten. Ein besonderes Augenmerk wird auf modulare Systeme für Segmentierungsaufgaben und Texturanalyse gelegt. In Kapitel 3 werden die theoretischen Grundlagen der Texturanalyse näher betrachtet. Insbesondere werden bewährte Methoden für die Extraktion von Texturmerkmalen eingeführt und deren freie Methodenparameter betrachtet. In Kapitel 4 werden Aspekte der Klassifikation und Bewertung näher beleuchtet.

Der zweite Teil der Arbeit behandelt Entwurfs- und Optimierungsstrategien für Bildverarbeitungssysteme. Hauptaugenmerk dieser Arbeit liegt hierbei auf der Verwendung von Evolutionären Algorithmen (EA) und Partikelschwarmoptimierung (PSO). Kapitel 5 beschäftigt

sich mit den Prinzipien der lokalen und globalen Optimierung von Bildverarbeitungssystemen mittels EA und PSO. Der Einsatz verschiedener Bewertungskriterien, u.a. die Verwendung von Klassifikationsergebnissen, wird untersucht. Bestehende Ansätze für den automatischen Entwurf von industriellen Bildverarbeitungssystemen kommerzieller Anbieter, sowie wissenschaftliche Arbeiten werden vorgestellt und diskutiert. Kapitel 6 beschäftigt sich mit dem Konzept und der Architektur einer Umgebung für den Entwurf und die Autokonfiguration einzelner Module bzw. des kompletten Bildverarbeitungssystems.

Im dritten, experimentellen Teil dieser Arbeit werden die vorgestellten Entwurfs-, Autokonfigurations- und Optimierungsprinzipien anhand von speziellen, besonders typischen Aufgaben aus der industriellen BV getestet und evaluiert. Als Testdaten dienen hierbei speziell an industrielle Aufgabenstellungen angepasste synthetische Daten für die Defekterkennung, Benchmark-Datensätze für die Texturklassifikation sowie konkrete Applikationsdaten aus der Oberflächeninspektion (Kapitel 7).

Kapitel 8 diskutiert die Ergebnisse und gibt einen Ausblick über zukünftige Arbeiten.

Im Anhang wird abschließend ein Einblick in die programmiertechnische Umsetzung des Gesamtsystems für den Entwurf, die Autokonfiguration und die Optimierung von Bildverarbeitungssystemen gegeben und das verwendete Bildmaterial dargestellt.

2 Visuelle Inspektionssysteme in der industriellen Bildverarbeitung

Eine automatische Prüfung mit einem visuellen Inspektionssystem in der industriellen Bildverarbeitung bietet entscheidende Vorteile gegenüber der manuellen, subjektiven menschlichen Prüfung. Ziel dieser Sichtprüfsysteme ist eine möglichst rationelle, objektive und zeitlich konstante Kontrolle jedes einzelnen Prüflings, d. h. eine 100%-Kontrolle in allen Bereichen der Fertigung. Je früher ein Fehler im Fertigungsprozess erkannt wird, desto kostengünstiger ist dies für den Hersteller.

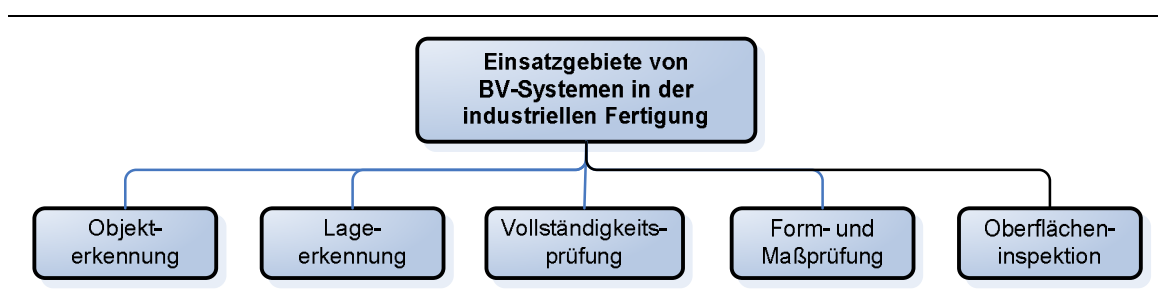


Abbildung 2-6: Einsatzgebiete von Bildverarbeitungssystemen in der industriellen Fertigung nach Jähne et. al. [56] und der VDI/VDE Richtlinie Nr. 2628 [149].

Einsatzgebiete von Bildverarbeitungssystemen in der industriellen Fertigung sind nach Jähne et. al. [56] und der VDI/VDE Richtlinie Nr. 2628 [149] die Objekterkennung, Lageerkennung, Vollständigkeitsprüfung, Form- und Maßprüfung und die Oberflächeninspektion (vgl. Abbildung 2-6). Demant et. al. [27] orientieren sich eher an der zugrunde liegenden Technologie und berücksichtigen in ihrem Buch über industrielle Bildverarbeitung zusätzlich die Gebiete der Kennzeichenidentifikation sowie den Bild- und den Objektvergleich. Die Objekterkennung hingegen wird als Bestandteil vieler Anwendungen und nicht als eigenständiges Ziel eines Bildverarbeitungssystems angesehen.

Einen Einblick in aktuelle Anwenderberichte aus dem Bereich der industriellen Bildverarbeitung erlaubt die Datenbank „Bildverarbeitung in Aktion!“ des VDMA².

Der Entwurf eines Online-Inspektionssystems umfasst nicht nur die Bildverarbeitung und Sensorik. Alle Komponenten bestehend aus Systemsoftware, Bildverarbeitung, Sensorik, Aktorik und Hardware müssen genauestens aufeinander abgestimmt sein und in eine gegebene

² <http://www.vdma.org/vision-at-work> (letzter Zugriff: 30.11.2008)

Produktionsumgebung integriert werden. Aus den produktionstechnischen Rahmenbedingungen, wie z. B. dem Prüfmaterial, der Prüfgeschwindigkeit und der Prüfaufgabe ergeben sich die spezifischen Anforderungen an das Inspektionssystem. Gemäß [118] lassen sich die Anforderungen eines Online-Prüfsystems für die Oberflächenprüfung grob in sechs Klassen aufteilen (vgl. Abbildung 2-7).

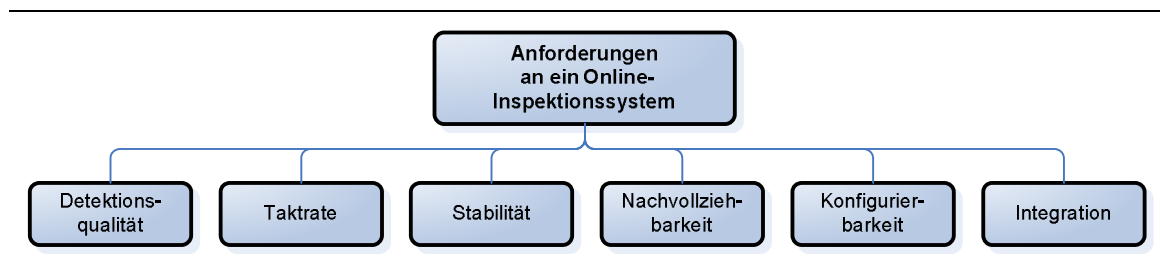


Abbildung 2-7: Anforderungen an ein Online-Inspektionssystem [118].

Die Fehlerdetektion muss in der Regel mindestens so flexibel wie die manuelle Prüfung sein. Hierbei müssen die subjektiven Merkmale der manuellen Prüfung in adäquate objektive Merkmale der automatischen Prüfung überführt werden. Neben einer gewünschten Detektionsrate von 100% soll die Anzahl der „Falsch-Positiven“, d. h. die Anzahl der Defekte die vorhanden sind und nicht erkannt werden, möglichst gering ausfallen.

Die Taktrate und die Reaktionszeit sind durch die Produktion vorgegeben. Das Bildverarbeitungssystem muss dementsprechend im Takt der Produktion arbeiten, ohne die Reaktionszeiten zu überschreiten.

Die Detektionsrate darf sich nicht durch äußere Einflüsse, wie z. B. einer Alterung der Komponenten ändern. Zudem erfordert ein stabiles System eine hohe Ausfallsicherheit. Sollte das System trotzdem ausfallen, muss dies dem Benutzer sofort mitgeteilt werden.

Der Nutzer des Online-Inspektionssystems muss den Systemzustand und die Prüfergebnisse zu jedem Zeitpunkt nachvollziehen können, um die Qualität der Prüfung und der geprüften Produkte einschätzen zu können.

Neue Varianten des zu prüfenden Produkts müssen wenn möglich ebenfalls vom Inspektionssystem inspiziert werden können. Dies erfordert eine gewisse Konfigurierbarkeit des Systems vor Ort.

Für eine reibungslose Integration des Prüfsystems in die Fertigungsanlage muss die Kommunikation mittels produktionsüblicher Protokolle (z. B. Profibus DT oder den Industrial Ethernet Port) ablaufen. Die Verarbeitung von Signalen der Anlage, wie Trigger-Signale bzw. Ansteuerungssignale an z. B. einen Sortierer oder Markierer muss gewährleistet sein.

Zudem sollte der Einbau des Prüfsystems möglichst wenige Änderungen an der Produktionsanlage erfordern.

Aus diesen Anforderungen und den produktionstechnischen Rahmenbedingungen, wie z. B. dem Prüfmaterial, lassen sich dann der Hardwareaufbau, Beleuchtung, Kameraauflösung und -anzahl, der Systemaufbau und die benötigte Algorithmik für den Entwurf des Bildverarbeitungssystems bestimmen (vgl. Abbildung 2-8).

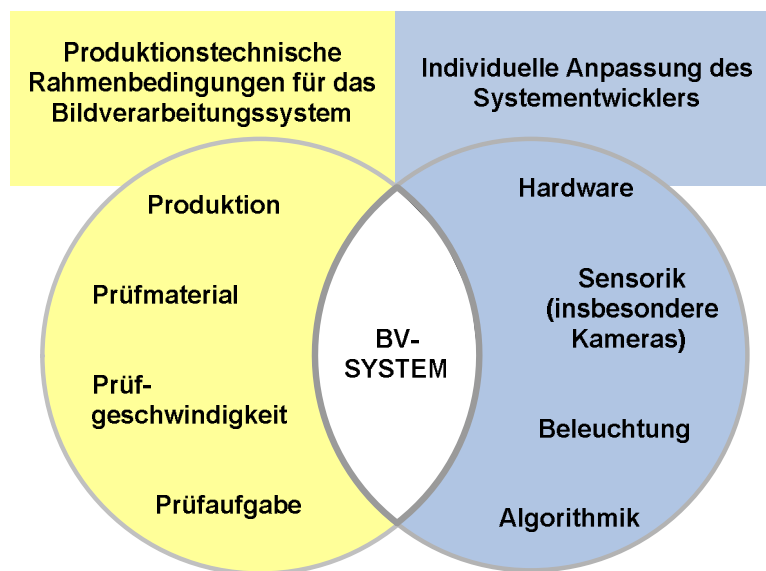


Abbildung 2-8: Produktionstechnische Anforderungen und daraus resultierenden individuelle Anpassungen eines Bildverarbeitungssystems für die visuelle Oberflächeninspektion.

Im Folgenden wird kurz auf die einzelnen Systembereiche gemäß Abbildung 2-8 eingegangen. Tiefer gehende Informationen zu dem optischen Hintergrundwissen und der Funktionsweise von Sensorik und Beleuchtung können beispielsweise [6, 27, 52] entnommen werden. Einen guten Einblick in aktuelle Lösungen und Technologien der deutschen und internationalen industriellen Bildverarbeitungsbranche gibt auch die vier mal im Jahr erscheinende Zeitschrift „INSPECT“ des GIT Verlags³. Neben gedruckten Informationen werden die neuesten Trends auf den jährlich stattfindenden Messen mit dem Thema Bildverarbeitung vorgestellt. Einige interessante Messen sind hierbei die „VISION“⁴ und die „Control“⁵ in Stuttgart, die „Automatica“⁶ in München und die „Hannover-Messe“⁷ in Hannover.

³ GIT VERLAG GmbH & Co. KG, 64293 Darmstadt

⁴ <http://www.vision-messe.de> (letzter Zugriff: 30.11.2008)

⁵ <http://www.control-messe.de> (letzter Zugriff: 30.11.2008)

⁶ <http://www.automatica-muenchen.de> (letzter Zugriff: 30.11.2008)

⁷ <http://www.hannovermesse.de> (letzter Zugriff: 30.11.2008)

2.1 Sensorik

In der industriellen Bildverarbeitung werden je nach Anforderungsprofil hauptsächlich Flächen- oder Zeilenkameras eingesetzt. Flächenkameras werden beispielsweise bei vereinzelt oder ruhenden Objekten eingesetzt, Zeilenkameras vorzugsweise bei der Prüfung von Endlos-Materialien. Die Kameras beinhalten wahlweise einen oder mehrere Sensoren der CCD- oder CMOS-Technologie [5, 40] für die Bildaufnahme. Kameras mit nur einem Sensor werden entweder für reine Intensitäts-, d. h. Graustufenaufnahmen oder mit integrierten Farbfiltern nach Bayer-Muster [27] über den einzelnen Sensorelementen für einfache Farbverarbeitungsaufgaben verwendet. Für anspruchsvollere Aufgaben aus der Farbbildverarbeitung werden Kameras mit drei CCD-Sensoren eingesetzt. Über eine integrierte Optik, in der Regel Prismen, wird das einfallende Licht in drei Spektralbereiche getrennt und auf die Sensoren für rotes, grünes und blaues Licht gelenkt. Neben den üblichen Kameras für Monochrom- und Farbverarbeitung kommen Spezialkameras und weitere Sensorsysteme für die Bildgebung zum Einsatz. Hierbei handelt es sich beispielsweise um Zwei-Sensoren-Kameras für die Bildaufnahme im visuellen und im nahem infraroten Bereich, um Kameras für die 3d-Wahrnehmung wie zum Beispiel Photo-Misch-Detektor-Kameras (PMD-Kameras) [51, 55] oder um bildgebende Sensoren für Ultraschall-, Röntgen- oder neuerdings auch Terahertzstrahlung [63].

2.1.1 Analoge Flächenkameras

Bei den analogen Kameras wird das Bildsignal als analoges Signal üblicherweise über ein Koaxialkabel oder ein Mehr-Pin Kabel an den Rechner übertragen. Mittels einer Bilderfassungskarte (Framegrabber) für analoge Bildsignale wird das Signal abgetastet und diskretisiert. Standardmäßig handelt es sich hierbei um eine Diskretisierung des Signals in 8-Bit, d. h. 256 Intensitätsstufen für Monochromkameras oder 24-Bit für RGB-Farbkameras. Eine höhere Farb- oder Intensitätsauflösung mit 10- oder 12-Bit ist ebenfalls möglich. Neben der Bilderfassung können Framegrabber zusätzliche Aufgaben wie beispielsweise das analoge Filtern des Bildsignals, die automatische Anwendung von Farbtabelle, eine zusätzliche Zwischenspeicherung für mehrere Bilder oder das Ansteuern der Kamera und der Beleuchtung übernehmen. Die typische Bildgröße bei analogen Monochromflächenkameras hat nach der Videostandardnorm CCIR 752 x 582 Bildpunkte.

2.1.2 Digitale Flächen- oder Zeilenkameras

Die Bildübertragung digitaler Kameras unterscheidet sich grundlegend von der Bildübertragung bei analogen Kameras. Die Sensordaten werden hier direkt in der Kamera in digitale Daten umgewandelt. Die Grau- bzw. Farbwerte werden anschließend digital über in der Regel vieladriges Kabel an den Rechner übertragen. Gängige digitale Schnittstellen für die

Datenübertragung sind in Abbildung 2-9 dargestellt. Die relativ neue Gigabit-Ethernet-Schnittstelle [28] setzt sich bei den Kameraherstellern vermehrt durch, steckt aber im Vergleich zu der bereits etablierten FireWire-Schnittstelle noch in den Kinderschuhen. Trotz der digitalen Übertragung der Bilddaten bei Verwendung von digitalen Kameras ist der Einsatz eines Framegrabbers für die industrielle Bildverarbeitung sinnvoll [27].

CCD-Flächenkameras haben im Jahr 2008 eine typische Auflösung⁸ von bis zu 8096×8096 Bildpunkten, CMOS-Kameras liegen im Bereich von bis zu 2048×2048 Bildpunkten. Die Zeilenauflösung der Anfang 2008 bei Stemmer-Imaging⁷ [80] vertriebenen Zeilenkameras liegt im Bereich von 512 bis zu 12288 Bildpunkten.

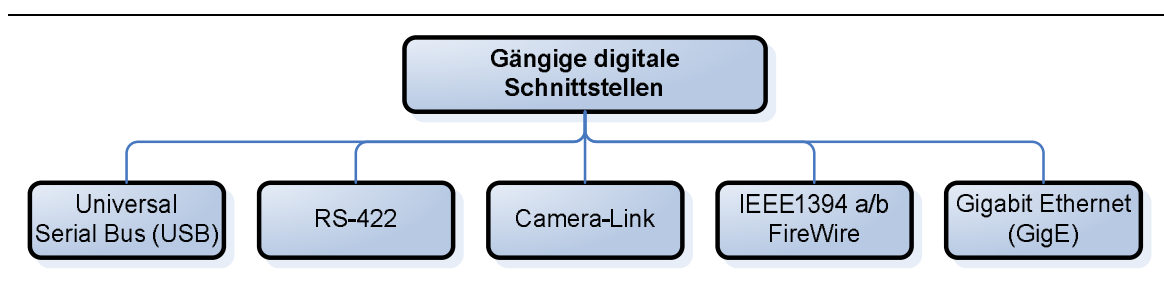


Abbildung 2-9: Gängige digitale Schnittstellen.

2.1.3 Intelligente Kameras und „Embedded Systems“

Bei intelligenten Kameras [139, 164] wird ein CCD-Sensor mit einem schnellen digitalen Signalprozessor (DSP) für die Bildauswertung kombiniert. Die Verarbeitung des Bildsignals und die Auswertung werden komplett auf der Kamera ausgeführt. Ein externer Framegrabber für die Wandlung des analogen Video-Signals in ein digitales Bild und ein angeschlossenes PC-System mit laufender Bildverarbeitungssoftware sind bei Verwendung einer intelligenten Kamera nicht mehr notwendig. Dies erlaubt – nicht zuletzt aufgrund der geringen Baugröße – einen flexiblen industriellen Einsatz dieser Kameras. Je nach Anbieter variieren die Programmierung und die Konfiguration einer intelligenten Kamera. Bildverarbeitungslösungen können so beispielsweise mit einer zu C ähnlichen Programmiersprache selbst erstellt werden oder es wird eine Lösung mittels grafischer Programmierung durch das Zusammenstellen von Bildverarbeitungs-Modulen generiert.

Die maximale Auflösung intelligenter Kameras ist in der Regel geringer als bei herkömmlichen Kameras. Hochauflösende intelligente Kameras haben hier bis zu 1600x1200 Bildpunkte.

Neben intelligenten Kameras mit DSP gibt es Kameras mit einem speziellen Chip-Design, das explizit für das Lernen mittels neuronaler Netze ausgelegt ist.

⁸ Stemmer Imaging GmbH, <http://stemmer-imaging.de> (letzter Zugriff: 30.11.2008)

2.1.4 Objektive

Im Zusammenhang mit der Auswahl der Kamera muss immer auch die Auswahl eines geeigneten Objektivs betrachtet werden. Je nach Glasarten, Linsenformen und -zusammenstellung ergeben sich spezifische Strahlengänge für die Abbildung des zu prüfenden Objekts auf den Sensor und somit Werte für z. B. die Brennweite oder den Abbildungsmaßstab der Objektivs. Auf die Linsen aufgedampfte Schichten oder zusätzliche Filter beeinflussen die Übertragungsfunktion des Objektivs. Je nach Einstellung der Vergrößerung, des Fokusses oder der Blende werden unterschiedliche Bereiche im Bild für eine anschließende Bildauswertung hervorgehoben. Nähere Informationen zu den optischen Grundlagen und über spezielle Objektive können zum Beispiel in [27, 57] oder direkt in den Objektiv-Datenblättern der Objektivhersteller und -vertreiber nachgelesen werden.

2.1.5 Automatisierung und Optimierung bei Kameras und Sensorik

Die meisten Kameras bieten integrierte Funktionen zur automatischen Bildverbesserung. Hierzu gehören beispielsweise ein automatischer Weißabgleich, eine automatische Verstärkung des Signals (Gain) oder eine Verschiebung des Intensitätswertes durch Addition eines festen Wertes (Offset). Für industrielle Anwendungen werden diese automatischen Zusatzfunktionen jedoch häufig abgeschaltet. Dies liegt einerseits an dem zumeist undefinierbaren Verhalten dieser Funktionen bei Veränderungen der Aufnahmeszenarie. Andererseits werden die Kameraeinstellungen dazu verwendet, um relevante Bildbereiche hervorzuheben bzw. Objekte in nichtrelevanten Bildbereichen durch zum Beispiel Übersteuerung zu unterdrücken. Generell bieten die Stellschrauben bei der Kameraeinstellung Potenzial zur Optimierung der Gesamtlösung. Diese werden allerdings weitestgehend manuell eingestellt und fixiert, so dass die Kamera im industriellen Einsatz reproduzierbare Ergebnisse liefert.

Die Einstellung der freien Objektivparameter wird in der Regel ebenfalls manuell getroffen und anschließend für den industriellen Einsatz mit Hilfe von Feststellschrauben fixiert. In einzelnen Anwendungen werden Kameras mit einem Autofokus⁹ verwendet. Aufgrund einer integrierten Mechanik zur Veränderung der Objektiveneinstellungen sind diese zumeist etwas größer als Standardkameras.

2.2 Beleuchtung

Die richtige Beleuchtung ist ein nicht unerheblicher Faktor für den erfolgreichen Einsatz eines industriellen Bildverarbeitungssystems. Je nach Lichtquelle und Beleuchtungsmethode können Oberflächenstrukturen hervorgehoben oder unterdrückt, Kanten verstärkt, Schatten vermieden und Reflexe verhindert werden. Je besser die relevanten Eigenschaften des zu prüfenden

⁹ Stemmer Imaging GmbH, <http://stemmer-imaging.de> (letzter Zugriff: 30.11.2008)

Objekts durch die Beleuchtung sichtbar gemacht werden, desto einfacher wird die anschließende Bildauswertung. Im Gegenzug ist jedoch die algorithmische Auswertung sehr schwer, wenn zu untersuchende Oberflächenstrukturen aufgrund einer falschen Beleuchtung im Bild kaum oder gar nicht erkennbar sind.

Neben der Relevanz der Beleuchtung für die Bildauswertung spielen zusätzlich weitere Faktoren, wie beispielsweise die Lebensdauer, die Kosten, die Lichtintensität und -farbe oder die Abstrahlcharakteristik der Lichtquelle eine Rolle für die industrielle Bildverarbeitung. Neben der Auswahl der eigentlichen Beleuchtung sollte zusätzlich gegen Fremdlicht, wie z. B. aufgrund direkter Sonneneinstrahlung, abgeschirmt werden.

Im Folgenden wird kurz auf verschiedene Lichtquellen und unterschiedliche Beleuchtungsmethoden für die industrielle BV sowie auf potenzielle Verfahren für den automatischen Entwurf und die Autokonfiguration von BV-Systemen in Hinblick auf die Beleuchtung eingegangen.

2.2.1 Lichtquellen

Neben konventionellen Lichtquellen der industriellen Bildverarbeitung, wie Leuchtstoffröhren oder Halogenlampen (Kaltlichtquellen) werden in den letzten Jahren zunehmend Leuchtdioden verwendet. Diese haben den Vorteil, dass sie neben geringen Kosten eine sehr lange Lebensdauer haben. Des Weiteren sind die Dioden im Vergleich zu Lasern für menschliche Augen meist ungefährlich.

Die geringe Größe und die Möglichkeit einer flexiblen Anordnung und Ansteuerung der Leuchtdioden mittels angepasster Schaltungstechnik ermöglichen den optimalen Einsatz von z. B. Linien-, Ring-, Flächen- oder sogar Halbkugelbeleuchtungen mit mono- oder polychromatischem Licht. Polychromatisches Licht wird mittels Farbmischung mehrerer monochromatisch emittierenden Leuchtdioden oder mit Lumineszenzkonversions-LEDs mit einer internen zusätzlichen Leuchtschicht erzeugt.

2.2.2 Beleuchtungsmethoden

Grundsätzlich unterscheidet man zwischen zwei Arten von Beleuchtungsanordnungen, dem Auflicht und dem Durchlicht (vgl. Abbildung 2-10). Für Oberflächeninspektionssysteme werden typischerweise Auflichtbeleuchtungen eingesetzt, wenn es um die Erkennung von Defekten wie Kratzern, Dellen oder Ähnlichem geht. Für die Erkennung von Löchern im Material oder das Prüfen von Konturen eignet sich die Durchlichtbeleuchtung.

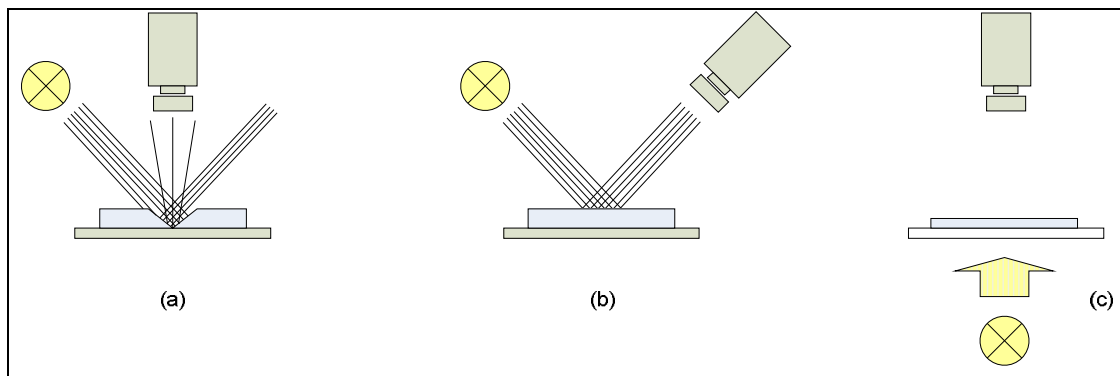


Abbildung 2-10: Verschiedene Beleuchtungsmethoden. Dunkelfeld-Auflichtbeleuchtung (a), Hellfeld-Auflichtbeleuchtung (b) und Durchlichtbeleuchtung (c).

Bei Auflicht befindet sich die Beleuchtung auf der gleichen Seite des Objekts, wie die Kamera. Hier gibt es zwei prinzipiell unterschiedliche Methoden, die Dunkelfeld- und die Hellfeldbeleuchtung.

Bei der Dunkelfeldbeleuchtung ist die Beleuchtungsquelle so positioniert, dass gerichtetes Licht über einen flachen Einfallswinkel auf das zu beobachtende Objekt gebracht wird (vgl. Abbildung 2-10a). An Unebenheiten der Oberfläche bzw. Oberflächenstrukturen wird das Licht in Richtung der Kamera reflektiert. Direkt an der glatten Objektoberfläche reflektiertes Licht erreicht die Kamera hingegen nur mit einem geringen Anteil. Dieses Verfahren kann dementsprechend zum sichtbar machen von Oberflächendefekten wie zum Beispiel Kratzern in metallischen Materialien oder von gravierten Strukturen auf dem Prüfobjekt eingesetzt werden. Das Bild dieser Strukturen erscheint heller. Um Strukturen unabhängig von ihrer Ausrichtung hervorheben zu können, werden gerne ring- oder rahmenförmige Beleuchtungen verwendet.

Für eine Hellfeldbeleuchtung wird die Beleuchtungsquelle so positioniert, dass an einer glatten Oberfläche reflektiertes Licht direkt die Kamera erreicht. Das Bild erscheint in diesem Fall hell. Diffus an Oberflächenstrukturen reflektiertes Licht lässt diese Bereiche im Bild hingegen dunkler erscheinen. Eine Hellfeldbeleuchtung kann durch die Positionierung von Kamera und Lichtquelle unter Berücksichtigung des Reflektionsgesetzes (Einfallswinkel gleich Ausfallswinkel) erreicht werden (vgl. Abbildung 2-10b). Bei einer Positionierung der Kamera senkrecht über dem Prüfobjekt kann das Licht alternativ von der Seite über einen halbdurchlässigen Spiegel eingestrahlt werden.

Bei einer Beleuchtung mit Durchlicht (Abbildung 2-10c) befindet sich die Beleuchtung hinter dem Objekt auf der entgegengesetzten Seite der Kamera. Durchlicht eignet sich generell für Vermessungsaufgaben oder auch für die Erkennung von Löchern, da die Außen- und ggf. Innenkonturen eines zu prüfenden Objekts hervorgehoben werden. Diffuses Durchlicht wird

mittels Durchstrahlung halbtransparenter Glas- bzw. Kunststoffplatten erreicht und kann für ebenfalls für Vermessungsaufgaben verwendet werden. Aufgrund des Seitenlichts wird allerdings die Objektkante zusätzlich beleuchtet, was zu einer Erschwerung bei der Kantendetektion führen kann. Durch das Einfügen einer Kollimatorlinse zwischen Lichtquelle und milchiger Glas- bzw. Kunststoffplatte wird paralleles Licht erzeugt, welches sich in Verbindung mit einem telezentrischen Objektiv für hochgenaue Vermessung eignet. Durch Verwendung von Polarisationsfiltern auf der Leuchtplatte und vor der Kamera können gewisse Strukturen bei transparenten Objekten hervorgehoben werden.

2.2.3 Automatisierung und Optimierung der Beleuchtung

Seitens der Industrie werden zunehmend steuerbare LED-Lichtquellen bzw. Controller angeboten, mit denen beispielsweise die Intensität, die Farbe, die Position und die Leuchtdauer bzw. Taktrate bei Blitzlicht automatisch über eine grafische Schnittstelle oder eine Programmierschnittstelle eingestellt werden kann. Diese Möglichkeiten erlauben es, die Beleuchtung flexibel automatisch oder halb-automatisch an die jeweilige Applikation anzupassen. Eine automatische Adaption der Beleuchtung wird zum Beispiel eingesetzt, um die Hintergrundbeleuchtung bei der automatischen Kontrolle von Schüttgut farblich anzupassen [140]. Erste Arbeiten beschäftigen sich mit Beleuchtungsserien mittels schaltbarer LED-Felder und einer daraus resultierenden applikationsbezogenen Optimierung der Beleuchtung bzw. einer Fusion der aufgenommenen Bilder. In anderen Arbeiten wird die Positionierung von Beleuchtung und Sensorik automatisch geplant [34, 35].

2.3 Bildverarbeitungssoftware

Im Folgenden werden der klassische Ansatz für die Erstellung einer Bildverarbeitungslösung kurz dargestellt und vier kommerzielle Softwareprodukte aufgeführt.

Nach Hermann [53] besteht der klassische Ansatz für die Erstellung von Bildverarbeitungslösungen in der Verwendung von Programmbibliotheken in Programmiersprachen der dritten Generation (3GL). Zu diesen gehören z. B. Pascal, C, C++ oder C#. Es werden entweder Eigenentwicklungen, frei verfügbare oder kostenpflichtige Bibliotheken verwendet. Zu den frei verfügbaren Bibliotheken gehören beispielsweise die „Open Computer Vision Library“ (OpenCV)¹⁰ oder das „Insight Segmentation and Registration Toolkit“ (Insight Toolkit, ITK)¹¹. Beiden Bibliotheken sind gemein, dass sie sehr viel Funktionalität anbieten, dadurch aber auch sehr komplex und teilweise unübersichtlich sind. Zusätzlich erfordert die Verwendung dieser Bibliotheken im Allgemeinen Programmierkenntnisse einer höheren Programmiersprache. Zu den führenden Anbietern von kommerziellen Bibliotheken gehört

¹⁰ <http://sourceforge.net/projects/opencvlibrary> (letzter Zugriff: 30.11.2008)

¹¹ <http://www.itk.org> (letzter Zugriff: 30.11.2008)

MVTec mit ihrer Bibliothek HALCON mit über 1300 Operatoren. Zu den Programmiersprachen der vierten Generation gehören z. B. so genannte Interpreter wie Matlab¹², die grafischen Programmierschnittstelle HDevelop für die Programmbibliothek HALCON oder Heurisko¹³. Diese ermöglichen es auf einfache Weise einzelne Operatoren direkt aufzurufen und anzuwenden. Neben der Verwendung von Softwarebibliotheken werden gerne Programme mit einer grafischen Oberfläche für das Zusammenstellen von Lösungen verwendet. Stellvertretend für eine Vielzahl von kommerziell verfügbaren Programmen werden hier folgende Produkte kurz vorgestellt: Khoros/Cantata, HDevelop, NeuroCheck und das Programm QuickCog. Die Programme unterscheiden sich neben ihrem Funktionsumfang vor allem in der Art, wie fertige Bildverarbeitungslösungen generiert werden können.

Neben der prinzipiellen Vorgehensweise für die Erstellung einer algorithmischen BV-Lösung liegt das Augenmerk der Betrachtung auf der softwareseitigen Unterstützung eines (ungeschulten) Anwenders hinsichtlich der Erstellung und Konfiguration einer Lösung. Im Einzelnen werden folgende Punkte betrachtet:

- sind Programmierkenntnissen einer höheren Programmiersprache notwendig
- die visuelle Darstellung der Struktur einer Lösung
- die Unterstützung eines ungeschulten Anwenders bei der Generierung komplexer (algorithmischer) Lösungswege
- die Unterstützung eines Anwenders bei der Konfiguration der erstellten Lösungen bzw. der Parametrierung der einzelnen verwendeten Module

Die Einbindung und Ansteuerung von Hardwarekomponenten, beispielsweise für den Bildeinzug, werden nicht untersucht.

2.3.1 Khoros/Cantata

Khoros¹⁴ ist ein offenes Softwareentwicklungssystem für Signal- und Bildverarbeitungsaufgaben, das ursprünglich von der Universität von Mexiko entwickelt wurde. Mittels der grafischen Oberfläche Cantata [160] können Verarbeitungsroutinen zu komplexen Datenflußpfaden zusammengesetzt werden. In diesem Zusammenhang wird eine Vielzahl von wieder verwendbaren modularen Softwarebausteinen beispielsweise für das Einlesen verschiedener Datentypen, für die Datenkonversion, für die statistische Auswertung oder als Bild- und Signalverarbeitungsroutinen bereitgestellt. 2d- und 3d-Daten können zusätzlich über vorhandene Zeichenfunktionen visualisiert werden. Die einzelnen Routinen sind thematisch in

¹² The MathWorks, Natick, MA 01760-2098, USA

¹³ AEON Verlag & Studio Walter H. Dorn e.K., 63452 Hanau

¹⁴ Khoros Inc., Albuquerque, NM 87110, USA

so genannten Toolboxen zusammengefasst. Weitere Charakteristika von Khoros sind die Erweiterbarkeit durch eigene Programmierung und die Möglichkeit der Ausführung der Datenflusspfade in einer verteilten Rechnerumgebung.

Für die Erstellung einer Applikation wählt der Anwender die gewünschten Routinen in der grafischen Entwicklungsumgebung aus einem Toolbox-Menü aus. Der entsprechende Glyph, die grafische Repräsentation einer Routine, kann im Arbeitsbereich frei platziert und über seine spezifischen Ein- und Ausgänge mit anderen Glyphen verbunden werden. Die Parameter eines Glyphs können manuell eingestellt werden. Durch die Kombination verschiedener Glyphen entstehen so komplexe Datenflusspfade, die entweder schrittweise oder als Ganzes ausgeführt werden können. Für die reine Verwendung der in Khoros integrierten Funktionen sind keine Programmierkenntnisse erforderlich. Bildverarbeitungsgrundkenntnisse sollten für die applikationsbezogene Zusammenstellung und Konfiguration der einzelnen Glyphen vorhanden sein. Für die Entwicklung und Einbindung eigener Funktionen sind zusätzlich Kenntnisse einer 3GL-Programmiersprache notwendig. Die grafische Darstellung des Datenflusses erleichtert das Verständnis der entwickelten Lösung. Hilfestellung bezüglich einer geeigneten Auswahl bzw. Konfiguration der einzelnen Operatoren zusätzlich zu der Operatorbeschreibung wird jedoch nicht gegeben.

2.3.2 HDevelop

HDevelop ist die grafische Benutzerumgebung für die umfangreiche modulare Softwarebibliothek HALCON der Firma MVTec¹⁵. Die Bibliothek bietet über 1300 Bildverarbeitungsoperatoren von der Bildakquise bis hin zur Visualisierung von Ergebnissen. Im Prinzip wird die komplette Bildverarbeitungskette und spezielle Funktionalität verschiedener Einsatzgebiete von BV-Systemen in der industriellen Fertigung gemäß Abbildung 2-11 abgedeckt. Bei HDevelop handelt es sich um einen Interpreter, der die einzelnen in der grafischen Entwicklungsumgebung ausgewählten und zusammengestellten Funktionen nacheinander abarbeitet. Für die Darstellung des Ablaufs der erstellten Applikation sind die Operatoren über ihre Namen und ihre Parameter aufgelistet. Das Ausführen der einzelnen gelisteten Funktionen kann schrittweise oder als Ganzes erfolgen. Eine farbliche Darstellung unterschiedlicher Funktionsgruppen und zusätzliche Kommentare unterstützen das Lesen der erstellten BV-Lösung. Programmierkenntnisse sind für das Zusammenstellen einer Applikationslösung nicht erforderlich, erleichtern aber das Verstehen des teilweise komplexen Applikationsablaufs.

Eine umfangreiche Beispieldatenbank gibt dem Anwender Anhaltspunkte für die Erstellung eigener problemspezifischer Applikationen. Zusätzlich werden für jeden ausgewählten Operator

¹⁵ MVTec Software GmbH, 81675 München

Tipps für alternative Funktionen, verwandte Themen, sowie mögliche Vorgänger- oder Folgefunktionen gegeben. Für die Konfiguration der einzelnen Operatoren werden einige wenige mögliche Werte vorgeschlagen. Teilweise wird der Einfluss der Parameterwahl auf das zu erwartende Ergebnis in der Operatorbeschreibung des dazugehörigen Hilfeintrag beschrieben. Die vorgeschlagenen Parameterwerte decken in der Regel einen Großteil des Parameterwertebereichs ab, so dass eine falsche Parametrierung auch hier zu ungewollten Ergebnissen führen kann. Die große Menge an Operatoren macht die Bibliothek und ihre grafische Oberfläche zu einem mächtigen Werkzeug für die Erstellung von Bildverarbeitungs-Lösungen, erschwert jedoch gleichzeitig die Auswahl geeigneter applikationsspezifischer Operatoren.

2.3.3 NeuroCheck

Im Gegensatz zu einer auf Beispielen basierten Unterstützung des Anwenders werden bei dem Programm NeuroCheck¹⁶ bei der Erstellung von BV-Lösungen Assistenten verwendet. Je nach Anwendungsgebiet wird der Nutzer durch eine Reihe von Menüs geführt, in denen er beispielsweise Angaben zu der Bildqualität machen kann. Anhand dieser Angaben wird anschließend automatisch ein Prototyp für den Analyseablauf erstellt. Solche Assistenten sind beispielsweise für die Anwesenheitskontrolle, Barcode-Erkennung, Muster- und Teile-Erkennung oder für die Teilevermessung verfügbar. Die Struktur eines Prüfablaufs ist in Form einer Baumstruktur gegeben, ähnlich der Darstellung der Datei- und Ordnerstruktur unter Windows. Einzelprüfungen können benannt und zu dieser Struktur hinzugefügt werden. In der zweiten, untergeordneten Ebene einer Einzelprüfung befindet sich eine Auflistung der dafür notwendigen Prüfschritte. Neben der generellen Struktur des Prüfablaufs wird u.a. zusätzlich der Datenfluss dargestellt. Programmierkenntnisse sind bei der Verwendung von NeuroCheck nicht notwendig. Die Assistenten ermöglichen auch einem ungeschulten Anwender, funktionsfähige BV-Lösungen zu erstellen. Die einzelnen BV-Algorithmen bzw. deren Parameter werden in der Hilfe beschrieben.

2.3.4 QuickCog

Bei QuickCog¹⁷ handelt es sich um ein System für einen transparenten teilautomatischen Entwurf und die interaktive bzw. teilautomatisierte Optimierung von Bildverarbeitungssystemen über Merkmalsraumbewertungsfunktionen nach dem Baukasten-Prinzip. Die einzelnen Module werden vergleichbar zu Cantata auf einem Steckbrett angeordnet und miteinander verknüpft. Neben der Verwendung von Standardmethoden der Bildvorverarbeitung, Segmentierung und

¹⁶ NeuroCheck GmbH, 71686 Remseck

¹⁷ Andreas König, Michael Eberhardt und Robert Wenzel
<http://www.quickcog.de> (letzter Zugriff: 30.11.2008)
Vertrieb: PHYTEC Messtechnik GmbH, 55129 Mainz

Merkmalsextraktion wird hierbei ein besonderes Augenmerk auf Klassifikation und Training im offenen Kreis gelegt. Methoden zur Dimensionsreduktion, z. B. für die Merkmalsselektion, und die im Entwurfsablauf integrierten Visualisierungswerkzeuge zum interaktiven Aufbau von Stichproben und der Merkmalsraumvisualisierung ermöglichen einen transparenten, fehlerarmen Entwurf und erlauben einen einfachen Umgang auch mit hochdimensionalen Merkmalsätzen. Die folgende Tabelle gibt abschließend einen Überblick der einzelnen Softwareprodukte hinsichtlich der Unterstützung eines ungeschulten Anwenders.

Tabelle 2-1.

	Programmier- kenntnisse	Visuelle Darstellung des Lösungsablaufs	Hilfe bei der Lösungserstellung	Hilfe bei der Lösungs- konfiguration
Freie Software- bibliotheken	notwendig	nein	nein	mögliche Parameter- werte sind in der Regel aufgeführt
Khoros/Cantata	nicht erforderlich	grafisch (in Form von Glyphs)	nein	nein
HDevelop	Hilfreich	Ablaufliste der einzelnen Methoden	Umfangreiche Datenbank mit Beispielapplikationen	Parameter und alternative Methoden werden angeboten
NeuroCheck	nicht erforderlich	Ablaufstruktur der einzelnen Prüfungen und deren Methoden	geführte Assistenten, Beispielapplikationen	mögliche Parameter- werte sind in der Regel aufgeführt
QuickCog	nicht erforderlich	grafisch (Baukasten-Prinzip)	Beispielapplikationen	mögliche Parameter- werte sind in der Regel aufgeführt

2.4 Optimierung und Automatisierung des Systementwurfs für visuelle Inspektionssysteme

Bei den in Abschnitt 2.3 vorgestellten Softwarelösungen für Bildverarbeitungsapplikationen handelt es sich prinzipiell um die Verwendung modularer Bildverarbeitungsmethoden mit definierten Eingangs- und Ausgangsdatentypen und fest zu wählenden Parametern. Für den Entwurf der grundsätzlichen Struktur einer BV-Lösung (vgl. Abbildung 2-11) werden Hilfestellungen in Form von Beispielapplikationen oder von Assistenten gegeben. Die automatische Konfiguration der einzelnen ausgewählten Operatoren ist hier nicht oder nur im geringen Maße Gegenstand der Anwenderunterstützung. Im Folgenden werden kurz wissenschaftliche Arbeiten und kommerzielle Produkte vorgestellt, welche sich mit der automatischen Optimierung bzw. der Konfiguration einzelner BV-Methoden und dem automatischen Gesamtentwurf eines BV-Systems für visuelle Inspektionssysteme beschäftigen. Hauptaugenmerk liegt hierbei auf der Verwendung von Evolutionären Algorithmen und Schwarmintelligenz.

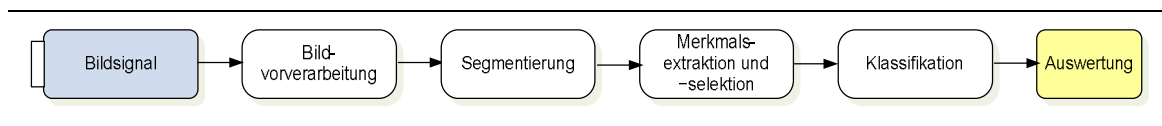


Abbildung 2-11: Schematische Darstellung einer typischen Bildverarbeitungskette.

2.4.1 Optimierung und Autokonfiguration einzelner BV-Methoden

Beispiele für die Optimierung und Autokonfiguration einzelner BV-Methoden in der Literatur sind z. B. die automatische Adaption von Strukturelementen und morphologischen Operatoren mittels Genetischer Programmierung (GP) [3, 4, 49, 66, 114] für eine Trennung von Objekt und Hintergrund, die automatische Auswahl optimaler Filter für die Texturklassifikation [71] oder die automatische Merkmalsselektion geeigneter Merkmale für eine anschließende Klassifikation. Ein weiteres Teilziel dieser Arbeit ist die Untersuchung, inwieweit eine Parameteroptimierung einzelner Methoden deren Leistungsfähigkeit gegenüber einer naiven Standard-Parametrierung bzw. einer Rastersuche steigert.

2.4.2 Automatische Mustererkennung und -klassifizierung

Als Beispiel für eine automatische Mustererkennung und -klassifizierung wird das SmartLearn-System von Cognex¹⁸ vorgestellt, mit dem eine Vielfalt von Defekten automatisch erkannt werden soll. SmartLearn ist ein neues patentiertes Mehrschritt-Klassifizierungswerkzeug für Oberflächeninspektionssysteme SmartView (2007). In einer Vorklassifikation werden mittels Expertenwissen unkritische Defekte herausgefiltert und leicht voneinander zu trennende Defekte klassifiziert. Anschließend werden komplexere Defekte mittels einer lernenden Klassifikation voneinander getrennt. Für die Auswahl geeigneter Merkmale werden hierfür wahlweise regelbasierte oder statistische Klassifikatoren oder Support-Vektor-Maschinen eingesetzt. In einem abschließenden Nachklassifikationsschritt können Defektklassen in Abhängigkeit der Defekteigenschaften wie beispielsweise der Defektgröße mittels Expertenwissen in weitere Unterklassen unterteilt werden. Dieses Verfahren ermöglicht eine applikationsbezogene, flexible, maßgeschneiderte Implementierung von Klassifikations-schemen. Einsatzgebiete von SmartLearn und SmartView sind die automatische Prüfung von Metall, Papier, Vliesstoffen, Kunststoff und Glas. Fehler, die dabei automatisch erkannt werden sind Kratzer, Blasen und Löcher. Die Darstellung dieses Werkzeuges ist der Internetrepräsentation von Cognex entnommen, so dass keine qualifizierte Aussage über die Leistungsfähigkeit des Systems möglich ist.

¹⁸ Cognex Germany, 76131 Karlsruhe

2.4.3 Automatischer Entwurf und Autokonfiguration von BV-Systemen

In diesem Abschnitt werden verschiedene Ansätze für den automatischen Entwurf und die Autokonfiguration eines BV-Systems vorgestellt.

Das SOLUTION-System [76] (ehemals CONNY) [77] wird verwendet, um mittels eines wissensbasierten Ansatzes low-level Bildverarbeitungsmethoden zu konfigurieren, mit dem Ziel geschlossene Konturen im Bild zu finden und zu extrahieren. Hierzu wird ein adaptives Regelwerk mittel Genetischer Algorithmen an die jeweilige Applikation angepasst.

- Bei den hier verwendeten low-level-Bildverarbeitungsmethoden handelt es sich um rein ikonische Operatoren, d. h. Bild-zu-Bild-Transformationen, wie beispielsweise Filter oder morphologische Operatoren.
- Ein applikationsspezifisches Ziel wird durch den Nutzer definiert. Anhand dieses Ziels werden Eigenschaften festgelegt, die das Ergebnisbild haben sollte. Anhand dieser Aufgabenstellung werden geeignete Operatoren aus einer verfügbaren Gruppe von Methoden ausgewählt, zu einer linearen Kette zusammengesetzt und konfiguriert.

Bei dem Programm GENIE [50, 99] der ISIS Gruppe der Los Alamos National Laboratories¹⁹ wird eine Lösung für die Segmentierung von multispektralen Satellitenbildaufnahmen automatisch zusammengestellt und optimiert. Es werden folgende Prinzipien und Methoden angewendet:

- Es handelt sich um einen überwachten Ansatz, bei dem kleine Bildregionen manuell einer Klasse zugeordnet werden. Anhand dieser Trainingsdaten wird das System entworfen und optimiert und abschließend das Gesamtbild in einem Erkennungsschritt klassifiziert.
- Auch hier werden rein ikonische, modulare Bildverarbeitungsoperatoren, d. h. Bild-zu-Bild-Operatoren verwendet.
- Einzelne Texturmerkmale, wie zum Beispiel der Grauwertmittelwert können verwendet werden, indem das Ergebnis für jeden Pixel in einem neuen Bild gespeichert wird. So bleibt die ikonische Bildverarbeitungsfolge erhalten, gleichzeitig können jedoch nur eine eingeschränkte Anzahl von Texturmerkmalen verwendet werden (pro Merkmal ein Bild).
- Die einzelnen Bildkanäle der multispektralen Satellitendaten werden getrennt verarbeitet.

¹⁹ <http://www.isis.lanl.gov> (letzter Zugriff: 30.11.2008)

- Mehrere Ergebnisbilder bzw. Bildkanäle können in einem temporären Zwischenspeicher gespeichert werden. Durch einen flexiblen Zugriff auf diesen Speicher können neben einer linearen Methodenabfolge auch Baum- und Graphenstrukturen realisiert werden.
- Die am Ende des Verarbeitungsablaufs im Speicher befindlichen Ergebnisbilder werden mittels Linearkombination zusammengeführt und klassifiziert.
- Für die Struktur- und Parameteroptimierung werden Verfahren von GA und GP angewendet. Insbesondere kommen dabei drei verschiedene Mutationsoperatoren zum Einsatz. Ein Mutationsoperator wird dazu verwendet, eine Methode komplett gegen eine andere auszutauschen. Mit dem zweiten Operator werden die Parameter einer Methode optimiert und mit dem dritten Operator wird der Bildzugriff auf den temporären Speicher variiert, was zu Veränderung der Ablaufsstruktur führen kann.

Mit GeniePro [100] wird Mitte 2008 eine kommerzielle neue Version des Programms auf den Markt kommen. In dieser Version werden drei separate Optimierungsschritte miteinander kombiniert. Zunächst werden anhand verschiedener Texturanalysenmethoden Merkmalsbilder generiert. Diese werden über eine Linearkombination zusammengefasst. In einem zweiten Schritt werden optimale morphologische Operationen ausgeführt und verschiedene Ergebnisbilder wiederum kombiniert. In einem abschließenden Schritt werden Schwellwertoperatoren und ein regionenbasierte Auswahlverfahren angewendet um Fehlklassifikationen zu vermeiden. In der Literaturangabe wird nur das prinzipielle Vorgehen bei GeniePro beschrieben. Eine Evaluation der Betaversion ist aufgrund sehr restriktiver Einschränkungen auf dem amerikanischen Markt nicht möglich.

Für die Realisierung eines adaptiven Texturanalysesystems mit Merkmalsselektion nach Pannekamp [98] wurden folgende Prinzipien und Verfahren angewendet:

- Das System muss auch in diesem Fall modular sein. Einzelne Methoden des Systems müssen austauschbar sein, d. h. sie müssen über einheitliche Schnittstellen verfügen.
- Es werden verschiedene Texturmerkmale betrachtet. Pannekamp legt besonderen Wert auf beleuchtungsunabhängige und rotationsinvariante Texturmerkmale. Aus diesem Grund werden Zwei-Punkt-Statistiken, d. h. Co-Occurrence- oder Grauwertabhängigkeitsmatrizen [47, 48, 105], lokale binäre Muster (LBM) [105], sowie Gabor- und Fourierspektren [105] verwendet, die alle sowohl rotationsvariant als auch rotationsinvariant eingesetzt werden können.

- Verschiedene Klassifikatoren, u.a. Abstandsklassifikatoren oder neuronale Netze, stehen für die Bewertung zur Verfügung.
- Anhand verschiedener Gütekriterien wird optional eine Merkmalsselektion ausgeführt. Klassifikatortyp und Gütefunktion werden hierbei in Relation zueinander betrachtet.
- Für die Merkmalsoptimierung der Texturanalyseparameter wird die Evolutionsstrategie (ES) von Rechenberg [119, 120] verwendet.

2.4.4 Kritikpunkte an den aufgeführten Ansätzen

Die aufgeführten Systeme bieten gemeinsam gesehen einen guten Ansatz für den automatischen Entwurf und die Autokonfiguration eines BV-Systems. Im Einzelnen betrachtet weisen die Verfahren jedoch Defizite auf:

- Bei der Optimierung eines einzelnen BV-Operators, wie zum Beispiel des Strukturelementes einer morphologischen Operation, wird in der Regel ggf. ausgeführte Bildvorverarbeitung bei der Optimierung nicht berücksichtigt. Das gleiche gilt bei der optimalen Anwendung verschiedener Klassifikatoren (Cognex).
- Bei einem regelbasierten Ansatz kann es bei einigen Algorithmen schwer fallen, die Wirkung auf ein Bild in eine überschaubare Menge an Regeln zu fassen. Wie können beispielsweise die Wirkungen der 2d-Laws-Filter [105] beschrieben werden? Aufgrund der Beschränkung von SOLUTION auf das Finden zusammenhängender Konturen ist es zudem nur für eine geringe Anzahl von industriellen Applikationen für die Oberflächeninspektion verwendbar.
- Aufgrund der Beschränkung des GENIE-Systems auf ikonische Bildverarbeitung und damit auch auf eine begrenzte Anzahl von Texturmerkmalen wird die Mächtigkeit des Ansatzes begrenzt.
- In keinem der dargestellten Verfahren wird die Regionengröße bei der Berechnung von Texturmerkmalen adaptiert. Diese kann jedoch einen entscheidenden Einfluss auf die Güte der Lösung haben.
- Nach Pannekamp ist die Auswahl eines Klassifikators in der Praxis weit weniger wichtig als eine geeignete Auswahl von Merkmalen. Der optimale Einsatz des Klassifikators spielt bei den vorgestellten Verfahren eine untergeordnete Rolle. Eine applikationsbezogene Parametrierung eines Klassifikators kann jedoch durchaus eine Rolle bezüglich des erzielbaren Ergebnisses spielen.

- Mehrzieloptimierung eines Systementwurfs zum Beispiel hinsichtlich einer für den industriellen Einsatz relevanten maximalen Laufzeit wird nicht berücksichtigt.

2.5 Zusammenfassung

Ein komplexes visuelles Inspektionssystem für die industrielle Bildverarbeitung hat viele Stellschrauben, die ausschlaggebend für einen erfolgreichen Einsatz des Systems in der Praxis sind. Neben der Auswahl und Konfiguration geeigneter Hardware in Form des Sensors, der Kamera, des Objektivs und der Beleuchtung spielt die Verwendung, Zusammenstellung und Konfiguration geeigneter Operatoren der Bild- und Signalverarbeitung eine entscheidende Rolle. Der Systementwickler wird hierbei in unterschiedlicher Form unterstützt. Eine visuelle Hilfe bei der Zusammenstellung verschiedener Methoden, Beispielapplikationen oder Assistenten helfen bei dem manuellen Entwurf und der Konfiguration eines visuellen Inspektionssystems für die industrielle Bildverarbeitung.

Im folgenden Kapitel werden grundsätzliche Methoden der Bildvorverarbeitung und Texturanalyse vorgestellt und hinsichtlich der freien Parameter beleuchtet.

3 Bildvorverarbeitung und Texturanalyse

In diesem Kapitel wird auf die relevanten Grundlagen der Bildvorverarbeitung und Texturanalyse eingegangen. Es wird, neben der Vorstellung eines im Rahmen dieser Arbeit entwickelten neuen Operators, eine praktische Auswahl bekannter und bewährter Verfahren aufgeführt und hinsichtlich ihrer Konfigurationsparameter beleuchtet. Eine geeignete Auswahl, Zusammenstellung und Parametrierung der Methoden der Vorverarbeitung und Texturanalyse kann einen entscheidenden Einfluss auf die Leistungsfähigkeit der gesamten BV-Lösung haben.

3.1 Grundlagen der Texturanalyse

Gemäß [84] sind Texturen komplexe visuelle Muster, bestehend aus Untermustern mit charakteristischen Eigenschaften wie beispielsweise der Helligkeit, der Farbe oder der Größe. In der gängigen Bildverarbeitungsliteratur sind u.a. folgende Beschreibungen von Texturen zu finden:

- In dem IEEE-Standard 610.4-1990 werden Grauwerttexturen folgendermaßen beschrieben: „*Texture is an attribute representing the spatial arrangement of the gray levels of pixels in a region.*“ (Übersetzung aus dem Englischen: Textur ist ein Attribut, das die räumliche Anordnung der Grauwerte von Bildpunkten in einer Region repräsentiert.)
- Nach Haralick [47, 48] wird eine Textur durch die Anzahl und Art ihrer Primitiven und der räumlichen Anordnung dieser Primitive beschrieben.
- Van Gool et al. [147] beschreiben Textur als eine Struktur, die aus einer großen Anzahl von mehr oder weniger geordneten ähnlichen Elementen oder Mustern zusammengesetzt ist, ohne dass eines dieser Elemente besonders hervorsteht. Die Textur vermittelt dem Betrachter den Eindruck einer globalen Zusammengehörigkeit. Sie ordnen der Textur zusätzlich zwei Hauptcharakteristika zu, Rauigkeit und Ausrichtung.
- Jähne [57] ordnet der Textur die grundlegende Eigenschaft zu, dass sich kleine elementare Muster periodisch oder quasiperiodisch im Raum wiederholen.

Texturanalyse von zweidimensionalen flächigen Texturen lässt sich in die Hauptaufgabengebiete gemäß Abbildung 3-12 unterteilen.

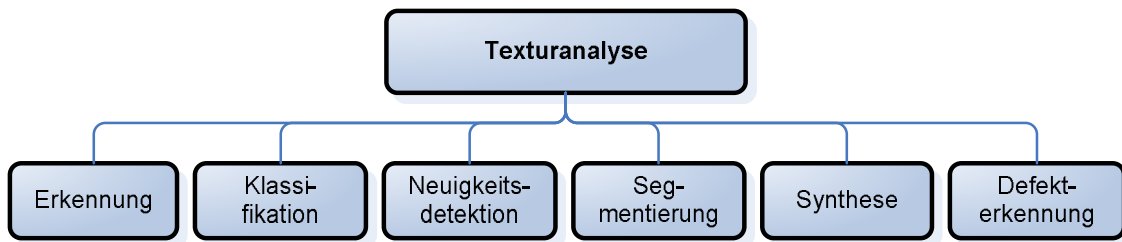


Abbildung 3-12: Hauptaufgabengebiete bei der Betrachtung von 2d-Textur mittels Methoden der Bildverarbeitung.

Bei der Texturerkennung oder –detektion geht es um die Anwesenheit und Position von Texturen in einem gegebenen Bild. Diese Fragestellung wird zum Beispiel verwendet, um eine texturabhängige Kompression von Bildern oder Videostreamen zu realisieren.

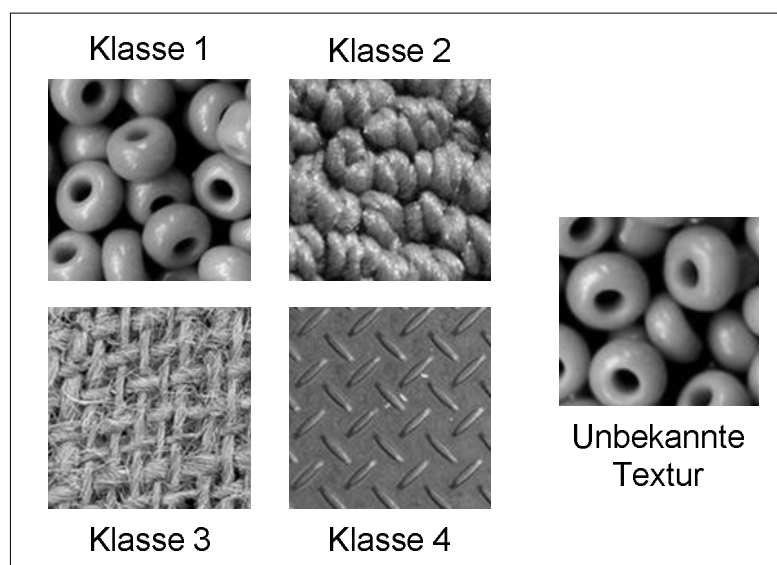


Abbildung 3-13: Beispiel für eine Aufgabe der Texturklassifikation. Ein unbekanntes Texturbeispiel soll einer bekannten Klasse zugeordnet werden.

Bei der Texturklassifikation geht es um die Unterscheidung und Zuordnung verschiedener Texturen. Ein Texturmuster einer unbekanntes Klasse soll einer bekannten Klasse zugeordnet werden (vgl. Abbildung 3-13). Typischerweise handelt es sich bei der Texturklassifikation um überwachte Lernmethoden. Anhand einer Reihe a priori bekannter Trainingsmuster werden mittels unterschiedlicher Analyseverfahren signifikante Merkmale generiert. Anhand der Merkmale wird ein Klassifikator trainiert, mit dem Ziel die Texturen im Trainingsdatensatz möglichst gut zu separieren. In einem zweiten Schritt werden nun unbekannte Muster anhand des trainierten Klassifikators zugeordnet. Bei nichtüberwachten Lernmethoden werden in der

Regel Clustermethoden verwendet, um Texturmodelle ohne vorherige Kenntnis der Zugehörigkeit der Texturmuster in dem Trainingsdatensatz zu erstellen. Die Leistungsfähigkeit bei der Texturklassifikation hängt stark von den verwendeten Merkmalen ab. Diese wird in der Regel anhand des Klassifikationsfehlers bei der Klassifikation von vorklassifizierten Testdaten bestimmt. In der Literatur lassen sich viele Arbeiten [2, 24, 79, 86, 95, 106, 148, 159] mit dem Thema der Texturklassifikation finden, die genannten Literaturverweise bieten nur einen kleinen Ausschnitt aus den Arbeiten, insbesondere bezüglich entwickelter oder verbesserter Algorithmen für die Texturklassifikation.

Bei der Neuigkeits- oder Anomaliedetektion [89, 135, 156] wird bestimmt, ob ein unbekanntes Muster zu einer gelernten Texturklasse gehört oder tatsächlich unbekannt ist. In der Regel wird hierbei die Distanz der Texturmerkmale des zu prüfenden Musters zu den Merkmalen der Trainingsdaten anhand eines Distanzmaßes bestimmt. Die Zuweisung der unbekanntes Textur erfolgt anhand eines zuvor festgelegten Schwellwertes bezüglich des Distanzmaßes. Im Gegensatz zur überwachten Texturklassifikation werden für das Training nur Beispieltexturen der Normaltextur, d. h. regelmäßig auftretender Textur(en), benötigt.

Bei der Textursegmentierung wird ein Bild anhand der Textur in signifikante Regionen gleichartiger Textur unterteilt. Prinzipiell wird hierbei zwischen überwachten und nichtüberwachten Methoden unterschieden. Bei der nichtüberwachten Textursegmentierung wird keinerlei Vorwissen über die Objekte oder homogene Texturregionen im Bild verwendet. Die entsprechenden Algorithmen müssen diese automatisch finden. Bei der überwachten Textursegmentierung werden hingegen a priori Informationen über das Bild, wie beispielsweise die Anzahl der Objekte oder Texturklassen verwendet um das Bild zu segmentieren. In der Literatur werden typischerweise kontur- oder regionenbasierte Verfahren beschrieben. Regionenbasierte Verfahren versuchen, homogene Regionen innerhalb des Bildes zu finden, konturbasierte Verfahren ordnen die Textur anhand von Inhomogenitäten an den Texturgrenzen zu. Textursegmentierung wird u.a. dazu verwendet, um Bilder natürlicher Szenen zu analysieren und einzuordnen, z. B. mit dem Ziel einzelne Objekte innerhalb des Bildes oder ähnliche Bildkompositionen in einer Datenbank zu finden.

Bei der Textursynthese [15, 105] werden Texturen anhand einer Mustertextur so generiert, dass es einem menschlichen Betrachter erscheint, als ob die Texturen mit dem gleichen stochastischen Prozess erzeugt wurden [155]. Textursynthese wird zum Beispiel für Applikationen im Bereich der Computergrafik oder der Bildkompression [141] angewendet.

Bei der Texturdefekterkennung (TDE) handelt es sich in der Regel um die Erkennung von Defekten im Rahmen industrieller Sichtprüfungsaufgaben. Ansätze für die Kategorisierung von Texturfehlern lassen sich beispielsweise in [143] finden. Tatari beschreibt in seiner Arbeit

Texturfehler zum Beispiel über ihre räumliche Ausprägung, Texturierung und in Bezug auf die Unterschiede zur Normaltextur. Zu den Fehlern nach ihrer räumlichen Ausprägung gehören beispielsweise offene Fehler, wie Löcher oder Risse.

Grundsätzlich sind die Problemstellungen der Defekterkennung sehr spezifisch, d. h. sie hängen von dem zu prüfenden Material und der Fehlerart ab, so dass es in der Regel keinen allgemeingültigen Ansatz für deren Lösung gibt. Eine Möglichkeit für den Einsatz von Texturmerkmalen für ein großes Gebiet von Fehlertypen ist die Verwendung invarianter Merkmale [42, 105]. Die Invarianz kann sich hierbei beispielsweise auf die Größe, die Rotation, eine Translation oder auf Beleuchtungsschwankungen beziehen. Mit der Generierung invarianter Texturmerkmale beschäftigen sich unter anderem [95, 98, 123, 127, 159, 162].

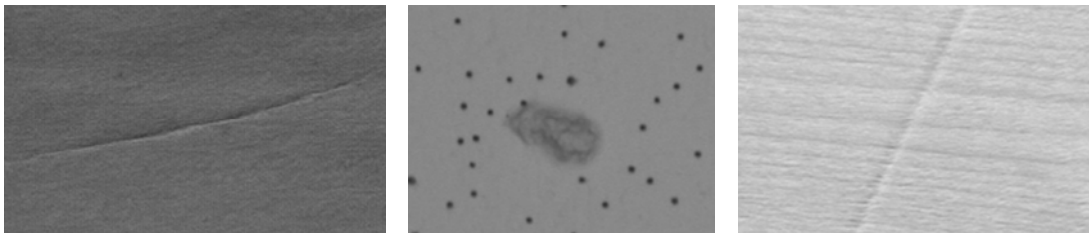


Abbildung 3-14: Beispiele für Texturdefekte²⁰. Links: Ein Kratzer auf Leder. Mitte: Ein Fleck auf einer Deckenplatte. Rechts: Eine Riefe im Holz.

Die untersuchten Materialien reichen von Holz, über Deckenplatten bis hin zu metallischen Oberflächen und Leder, um nur einige Beispiele zu nennen, vgl. Abbildung 3-14. Im Bereich der Erkennung von Fehlern in und auf Textilien gibt es eine Reihe von Arbeiten. Beispielhaft wird hier auf [17, 22] verwiesen. Weiterführende Informationen kann man in u.a. in [127] finden.

Für die TDE werden sowohl Verfahren der Texturklassifikation, der Textursegmentierung und der Neuigkeits- oder Anomaliedetektion [89, 135, 156] verwendet.

3.2 Verfahren der Texturanalyse

Die Verfahren für die Texturanalyse von zweidimensionalen Texturen lassen sich gemäß Abbildung 3-15 in fünf verschiedenen Hauptgruppen unterteilen [84, 121, 127, 147].

Bei den statistischen Methoden werden Merkmale aus der Verteilung und Relation der Grauwerte im Bild oder Bildausschnitt gewonnen. Einfache Texturmerkmale können bereits durch die Berechnung statistischer Größen, wie Mittelwert oder Varianz aus dem Grauwert-Histogramm gewonnen werden. Die Leistungsfähigkeit dieser anhand einer Statistik erster Ordnung berechneten Merkmale ist für komplexe Aufgaben der Texturanalyse in der

²⁰ Bildmaterial: Fraunhofer ITWM

Regel jedoch nicht ausreichend. Rechenzeit- und speicherintensiver ist die Verwendung von Statistiken zweiter Ordnung. Bei der Zwei-Punkt Statistik, oder auch Co-Occurrence-Matrix [47, 48, 105] wird neben den Grauwerten auch die räumliche Relation zwischen zwei Bildpunkten berücksichtigt. Weitere statistische Methoden sind die Autokorrelation (AK) [105], Lokale Binärmuster (*local binary pattern*, LBP) [94, 95] und registrierungsbasierte Verfahren.

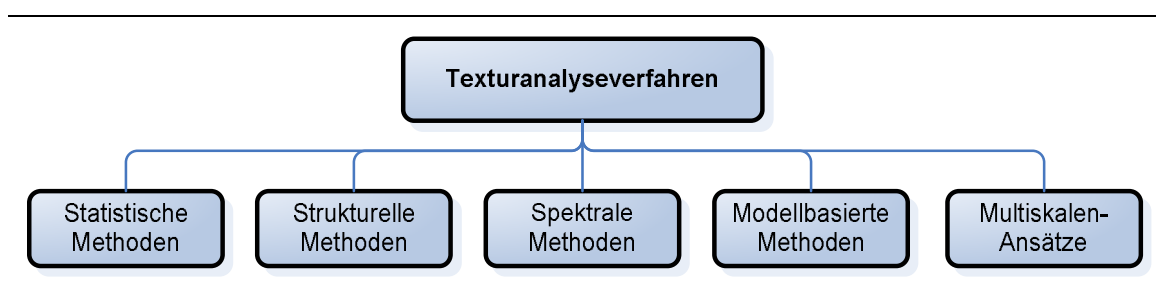


Abbildung 3-15: Hauptmethoden der Texturanalyse.

Bei strukturellen Ansätzen wird die Textur durch definierte Texturprimitive oder auch Textur-Elemente, kurz: Texel, repräsentiert. Für die Beschreibung dieser Texturen müssen die Texel und ihre Anordnung definiert werden. Strukturelle Ansätze liefern gute symbolische Beschreibungen einer Textur, was u.a. bei der Textursynthese von Vorteil sein kann. Zu den strukturellen Methoden zählt beispielsweise die mathematische Morphologie [84, 138] und Merkmale, die auf Regularität und lokaler Orientierung basieren [17].

Zu den spektralen Methoden gehören beispielsweise Merkmale, die auf dem Fourierspektrum basieren [105, 127].

Bei modellbasierten Methoden wird eine Textur mittels eines parametrischen Ansatzes beschrieben. Die Modellparameter werden in der Regel applikationsspezifisch gelernt und als Merkmale verwendet. Beispiele für modellbasierte Methoden sind Markov Zufallsfelder [105]. Modellbasierte Methoden werden unter anderem für die Textursynthese [105], die Texturklassifikation [16] oder die Textursegmentierung [69] verwendet.

In dieser Arbeit werden hauptsächlich statistische Methoden verwendet, da diese sich für eine Vielzahl von Texturen (stochastischen aber auch periodischen) einsetzen lassen.

Eine Übersicht über die Anwendungsgebiete der Texturanalyse gemäß Abbildung 3-12, die Einteilung der Texturanalysemethoden (Abbildung 3-15) und die Beschreibung einer Vielzahl von Operatoren für die Bestimmung von Texturmerkmalen lässt sich beispielhaft in folgenden Übersichtsarbeiten und Büchern finden: [42, 57, 84, 105, 136, 121, 147, 162].

3.3 Auswahl und Parametrierung von Methoden für die Texturklassifikation und Defekterkennung

Aufgrund der vielfältigen Aufgabenstellungen im Bereich der Texturanalyse, –klassifikation und Defekterkennung ist es nicht einfach generelle Aussagen zu treffen, welche Methoden mit welchen Parameterwerten für einen spezifischen Anwendungsfall zum Erfolg führen werden.

Tatari [143] beschäftigt sich in seiner Arbeit unter anderem mit der Kategorisierung von Texturfehlern und der Auswahl von geeigneten Texturanalyseverfahren im Bereich von industriellen Sichtprüfungsaufgaben. Zu diesem Zweck werden beispielsweise Textur- und Fehlerklassen definiert, in die eine Normaltextur ohne Fehler und die auftretenden Fehlertypen eingeordnet werden. Neben dieser Einordnung unterscheidet Tatari verschiedene Texturanalyseverfahren nach ihrem prinzipiellen Vorgehen. Abbildung 3-16 zeigt beispielhaft die von ihm erstellte Einteilung von Texturanalyseverfahren in Verfahren mit und ohne Segmentierung der Texturtextel. Insbesondere untersucht Tatari Analyseverfahren zur Erfassung von Grauwerteigenschaften von Textur und Defekt, sowie Verfahren für die Erfassung von Größen-, Ausrichtungs-, Form- und Anordnungseigenschaften.

Anhand von Defektbildern verschiedener texturierter Materialien, wie beispielsweise Teppich oder Holz, werden Normaltexturen und Fehler charakterisiert und zugeordnet und geeignete Texturanalyse- und Prüfverfahren bzw. einzelne Texturmerkmale für die Defekterkennung ausgewählt. Die Parametrierung der einzelnen Verfahren, wie beispielsweise die Wahl der Größe der Beobachtungsfenster erfolgt manuell anhand der Analyseergebnisse von Normal- und Fehlertextur.

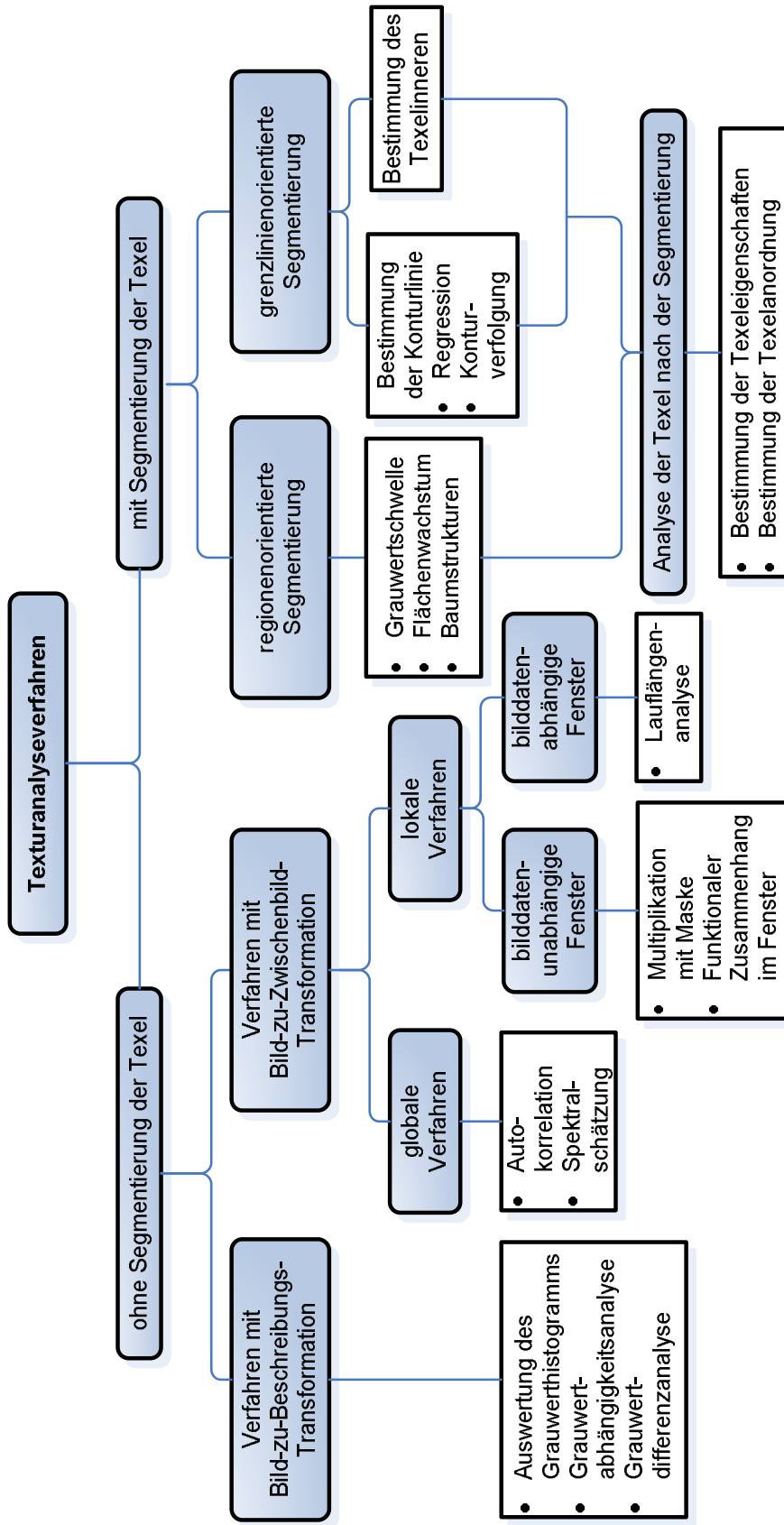


Abbildung 3-16: Einteilung von Texturanalyseverfahren nach Tatarı [143] in Texturanalyseverfahren mit und ohne Segmentierung der Texel.

3.4 Spezielle Methoden der Bildvorverarbeitung und Texturanalyse

In den folgenden Abschnitten werden einige ausgewählte bewährte Methoden der Vorverarbeitung, der Merkmalsberechnung und der Merkmalsauswahl vorgestellt und kurz auf die entsprechenden Methodenparameter eingegangen. Abschließend wird ein Texturoperator vorgestellt, der im Rahmen dieser Arbeit entwickelt und angewendet wurde.

3.4.1 Methoden der Bildvorverarbeitung und ihre Parameter

Es folgen ein kurzer Auszug von Standard-Methoden der Bildvorverarbeitung und eine Beschreibung möglicher Methodenparameter. Eine grafische Übersicht aller verwendeten Methoden für die Bildvorverarbeitung ist im Anhang zu finden. Die Auswahl an Methoden orientierte sich zunächst am Funktionsumfang verschiedener kommerzieller Produkte (Stand 2004) und an den in den Literaturquellen [100, 98, 143] beschriebenen Verfahren und wurde im Laufe dieser Arbeit stetig erweitert. Die Parameter der einzelnen Methoden werden im Laufe der Optimierung an die jeweilige Prüfaufgabe angepasst.

Filtermasken und Filtergrößen: Als Vorverarbeitungsschritt für die Bildsegmentierung und Texturanalyse können unter anderem Glättungsfilter, Kantenfilter oder Filter für die Hervorhebung einzelner Strukturen verwendet werden. Bei den meisten dieser Filter lässt sich die Größe der Filtermaske variieren. Neben der Anpassung der Filtergröße besteht die Möglichkeit des Züchtens neuer Filter mit variablen Filterelementen oder die Optimierung von Filterbänken [11, 71, 137].

Schwellwerte: Zur Trennung von Objekt und Hintergrund oder zur Hervorhebung verschiedener Textursegmente werden in einigen Fällen Schwellwertoperatoren verwendet. Im einfachsten Fall werden hierbei alle Grauwerte unterhalb und oberhalb einer Schwelle auf konstante Intensitätswerte gesetzt (Binarisierung). Andere Methoden unterdrücken nur einen bestimmten Grauwertebereich, während die restlichen Grauwerte unverändert in das Ergebnisbild übernommen werden. Alle Schwellwertverfahren haben eines gemein, es müssen ein oder mehrere datenabhängige Schwellwerte bestimmt werden. Für eine automatische Bestimmung einer Grauwertschwelle aus dem Grauwert histogramm gibt es verschiedene Verfahren, wie beispielsweise nach Otsu [96]. Einen guten Überblick über die Verfahren zur automatischen Bestimmung von Grauwertschwellen geben [61, 125, 132].

Strukturelemente: Bei Verwendung morphologischer Funktionen spielt die Wahl des Strukturelementes eine entscheidende Rolle. In Abhängigkeit des Strukturelementes können unterschiedliche Objektstrukturen hervorgehoben oder unterdrückt werden. Gängige 2d-Strukturelemente sind beispielsweise Quadrate, Quader, Linienelemente, Kreuze oder diskretisierte Kreiselemente. Arbeiten zur automatischen Adaption der Strukturelemente morphologischer Operationen sind beispielsweise [3, 4, 49, 66, 114]. Über die

Bildvorverarbeitung werden mittels morphologischer Operationen strukturelle Texturanalysemethoden in das System integriert [84]. Neben den Standardmethoden der Morphologie, wie Erosion, Dilatation, Öffnung und Schließung werden zusätzlich Gradienten-, TopHat- und adaptive Verfahren mit einer zusätzlichen Eingangsmaske verwendet.

3.5 Bestimmung von Texturmerkmalen

Im Folgenden werden einige grundlegende Methoden für die Bestimmung von Texturmerkmalen beschrieben. Zunächst erfolgt eine Auflistung globaler Parameter, die unabhängig von der gewählten Methode für die Texturmerkmalsberechnung zum Einsatz kommen können.

Grauwertquantelung: Die Anzahl der verwendeten Grauwerte G kann eine entscheidende Rolle bezüglich der Klassifikationsergebnisse und des Rechenaufwands bieten. Gerade für Grauwertstatistiken höherer Ordnung reduziert sich zusätzlich der aufzubringende Speicheraufwand. Zwei Verfahren für die Grauwertquantelung werden in dieser Arbeit optional angewendet. Der gesamte Grauwertwertebereich, i.A. ganzzahlige Grauwerte im Bereich $[0, 256]$, wird für die Quantelung verwendet. Jeweils 2^b , $b = 0..7$ benachbarte Grauwerte werden zu einem neuen Wert zusammengefasst. Dieses Verfahren wird beispielsweise auch bei QuickCog eingesetzt. Alternativ werden nur die tatsächlich vorkommenden Grauwerte für die Quantelung herangezogen.

Fenstergröße: Die Robustheit und Aussagekraft berechneter Texturmerkmale hängt bei vielen Verfahren entscheidend von der Fenstergröße des betrachteten Bildfensters $M \times N$ ab [98]. Diese Fenstergröße ist in der Regel abhängig von den zu prüfenden Texturen. Verschiedene Arbeiten beschäftigen sich mit einer a priori-Bestimmung der optimalen Fenstergröße [112, 127].

In dieser Arbeit sind die Adaption der Regionengröße und die automatische Grauwertquantelung in die Optimierung integriert.

3.5.1 Methoden der Berechnung von Texturmerkmalen

In diesem Abschnitt wird näher auf die Methoden der Merkmalsberechnung für die Texturanalyse eingegangen. Hierbei wird das Hauptaugenmerk auf bekannte Verfahren für die statistische Texturanalyse, im allgemeinen auf Statistiken erster und zweiter Ordnung gelegt, da hier aufgrund der Betrachtung einer Bildregion robustere Merkmale gewonnen werden können. Für die Verwendung von Grauwert-Histogrammen und daraus berechneter Merkmale spricht zusätzlich der geringe Berechnungsaufwand. Je nach Anwendungsfall liefert die viel verwendete und zitierte Methode der Co-Occurrence-Matrix, einer Statistik zweiter Ordnung, schlechtere, vergleichbare und auch bessere Ergebnisse als aktuelle Methoden, weswegen sie hier ebenso verwendet wird.

Inspiziert von der umfangreichen Evaluierung dieser Texturanalysemethoden von [133, 136], einer aktuellen Übersicht über Texturanalysetechniken für die Erkennung von Oberflächendefekten [156] und dem Funktionsumfang gängiger Softwaretools bei Beginn dieser Arbeit (2004) werden neben den oben genannten Methoden die ebenfalls etablierten Methoden für die Berechnung der Autokorrelationskoeffizienten und die Anwendung der Laws-Filter [105, 133] verwendet.

3.5.1.1 Grauerthistogramm

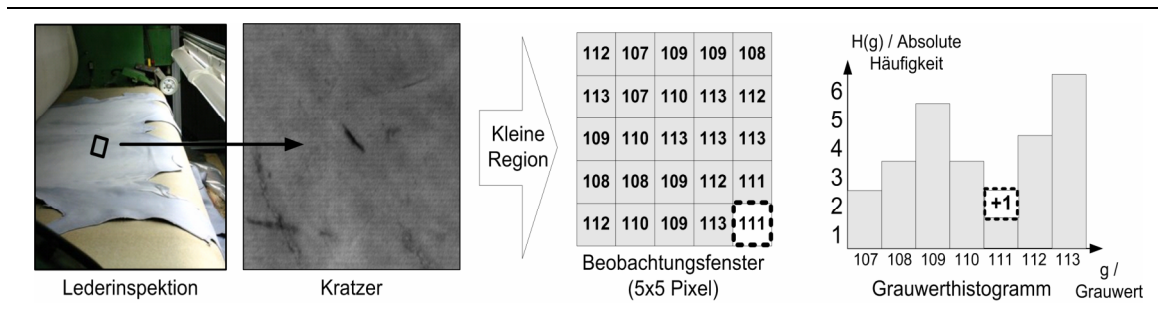


Abbildung 3-17: Bei der Bildung eines Grauerthistogramms, hier für eine kleine Bildregion, wird die relative bzw. absolute Häufigkeit des Auftretens der Grauwerte im Bild betrachtet.

Das Grauerthistogramm (vgl. Abbildung 3-17) beschreibt die Häufigkeitsverteilung der Grauwerte g in einem Bild. Die relative Häufigkeit $P(g)$ ist die Normierung der absoluten Grauwert-Häufigkeit $H(g)$ auf die Anzahl der Bildpunkte Z in einem Bild der Größe $M \times N$. $P(g)$ gibt somit die Wahrscheinlichkeiten des Auftretens des entsprechenden Grauwertes im Bild an. Die Position eines Pixels im Bild beeinflusst ist für die Bildung des Histogramms nicht relevant.

$$P(g) = \frac{1}{Z} H(g) \quad (3.1)$$

Aus dem Histogramm können statistische Merkmale, wie beispielsweise der **Mittelwert** μ berechnet werden:

$$\mu = \sum_{g=0}^{G-1} gP(g) = \frac{1}{Z} \sum_{g=0}^{G-1} gH(g) \quad (3.2)$$

Ein Maß für die Streuung der Grauwerte im Bild ist die **Varianz**:

$$\text{var} = \sum_{g=0}^{G-1} (g - \mu)^2 P(g) \quad (3.3)$$

Die Quadratwurzel aus der Varianz ist die **Standardabweichung**:

$$s = \sqrt{\text{var}} \quad (3.4)$$

Die **Schiefe** ist ein Maß für die Symmetrie des Histogramms zum Mittelwert. Ist die Schiefe größer als null, handelt es sich um eine rechtsschiefe Verteilung. Werte, die kleiner als der Mittelwert sind, treten in diesem Fall häufiger auf als Werte, die größer als der Mittelwert sind. Bei einem linksschiefen Histogramm mit einer Schiefe kleiner null ergibt sich ein komplementäres Verhalten.

$$sk = \frac{\sum_{g=0}^{G-1} (g - \mu)^3 P(g)}{s^3} \quad (3.5)$$

Die **Wölbung** oder **Kurtosis** beschreibt die Abweichung des Histogrammverlaufs bezüglich eines normalverteilten Histogramms. Ist die Wölbung gleich null, so spricht man von einer normalgipfligen Verteilung. Ist die Wölbung größer als null, handelt es sich um steilgipflige Verteilung mit im Vergleich zur Normalverteilung spitzeren Peaks.

$$kur = \frac{\sum_{g=0}^{G-1} (g - \mu)^4 P(g)}{s^4} - 3 \quad (3.6)$$

Die **Entropie** eines Bildes ist ein Maß für den Informationsgehalt des Bildes:

$$ent = - \sum_{g=0}^{G-1} P(g) \log_{10}(P(g)). \quad (3.7)$$

Der **Kontrast** ist ein Maß für die maximale Grauwertdifferenz im Bild.

$$con = \begin{cases} \frac{g_{\max} - g_{\min}}{g_{\max} + g_{\min}} & \text{für } g_{\min} \neq g_{\max} \\ 0 & \text{sonst} \end{cases} \quad (3.8)$$

Anhand der **Energie** oder **Quadratsumme** kann eine Aussage über die Homogenität der Grauwertverteilung eines Bildes gemacht werden.

$$qsum = \sum_{g=0}^{G-1} P^2(g) \quad (3.9)$$

Als Merkmale für die Texturanalyse dienen entweder die Histogrammeinträge direkt oder die hier beschriebenen Merkmale. Diese Merkmale sind automatisch translations- und rotationsinvariant.

3.5.1.2 Co-Occurrence-Matrizen und Haralick-Merkmale

Neben der Statistik erster Ordnung wird auch eine Statistik zweiter Ordnung für die Merkmalsberechnung herangezogen. Bei der Berechnung der Co-Occurrence-Matrix (Grauwertabhängigkeitsmatrix) wird die relative bzw. absolute Häufigkeit des Auftretens von Grauwertepaaren betrachtet, wobei die betrachteten Bildpunkte in einer festen räumlichen Relation $(\Delta x, \Delta y)$ zueinander stehen. Je nach Anwendungsfall kann es sinnvoll sein, eine gewisse Richtungsunabhängigkeit bzw. sogar Rotationsinvarianz bezüglich der räumlichen Relation zu fordern. Bei der Richtungsunabhängigkeit wird das Auftreten einer Grauwertkombination g_1, g_2 sowohl in dem Histogrammtopf $H(g_1, g_2)$ als auch in dem Topf $H(g_2, g_1)$ erfasst. Für die Rotationsinvarianz werden in der Regel neben der vorgegebenen Relation $(\Delta x, \Delta y)$ zwischen den betrachteten Pixeln auch komplementäre Abstände betrachtet: $(\Delta x, \Delta y) \rightarrow (\Delta x, \Delta y), (-\Delta x, \Delta y), (\Delta x, -\Delta y)$ und $(-\Delta x, -\Delta y)$.

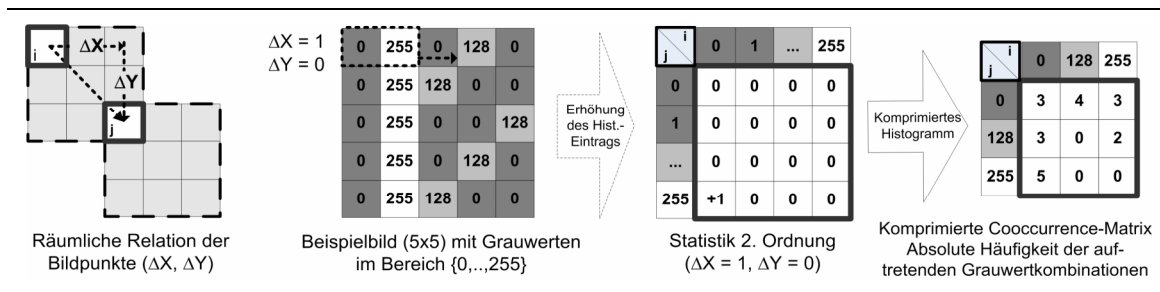


Abbildung 3-18: Co-Occurrence- oder Grauwertabhängigkeitsmatrix.

Im Folgenden werden die Formeln für die relative Häufigkeit des Auftretens eines Grauwertpaares im Bild ohne Rotationsinvarianz und die am häufigsten gebrauchten Haralick-Merkmale [105] aus der Co-Occurrence-Matrix: Energie (Eng), Kontrast (Con), Entropie (Ent), Homogenität (Hom) und Korrelation (Cor) angegeben.

$$P_{ij} = P(g_1, g_2) \Big|_{\Delta x, \Delta y} = \frac{1}{MN} H(g_1, g_2) \Big|_{\Delta x, \Delta y} \quad (3.10)$$

mit

$$i = g_1 = g(x, y) \quad (3.11) \quad \text{und} \quad j = g_2 = g(x + \Delta x, y + \Delta y) \quad (3.12)$$

$$Eng(i, j) = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (P_{ij})^2 \quad (3.13)$$

$$Con(i, j) = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} ((i-j)^2 \cdot P_{ij}) \quad (3.14)$$

$$Ent(i, j) = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (P_{ij} \cdot \log_{10} P_{ij}) \quad (3.15)$$

$$Hom(i, j) = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \left(\frac{P_{ij}}{1+|i-j|} \right) \quad (3.16)$$

$$Cor(i, j) = \frac{1}{\sigma_x \cdot \sigma_y} \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (P_{ij} \cdot i \cdot j) - \mu_x \cdot \mu_y \quad (3.17)$$

mit

$$\mu_x = \sum_{i=0}^{G-1} i \sum_{j=0}^{G-1} (P_{ij}) \quad (3.18), \quad \mu_y = \sum_{j=0}^{G-1} j \sum_{i=0}^{G-1} (P_{ij}) \quad (3.19)$$

und

$$\sigma_x = \sum_{i=0}^{G-1} (i - \mu_x)^2 \sum_{j=0}^{G-1} (P_{ij}) \quad (3.20), \quad \sigma_y = \sum_{j=0}^{G-1} (j - \mu_y)^2 \sum_{i=0}^{G-1} (P_{ij}) \quad (3.21)$$

Zu adaptierende Parameter bei der Berechnung der Grauwertabhängigkeitsmatrix bzw. der aus ihr resultierenden Haralick-Merkmale sind neben der bereits aufgeführten Grauwertquantelung und der Regionengröße die räumliche Relationen Δx , Δy der betrachteten Grauwertpaare und die Anzahl der betrachteten Richtungen bezüglich der Rotation.

Als Merkmale für die Texturanalyse dienen auch hier entweder direkt die Histogrammeinträge, die oben beschriebenen fünf meist genutzten Haralick-Merkmale oder der komplette Merkmalssatz von 14 Merkmalen [47, 48].

3.5.1.3 Laws' Texturfilter und Texturenergien (LTEM)

Laws [74] hat eine Reihe von eindimensionalen Filtermasken unterschiedlicher Länge entwickelt, die bestimmte Texturmerkmale, wie beispielsweise Kanten (engl.: *edges*), Wellen (engl.: *waves*) oder Flecken (engl.: *spots*) innerhalb eines Bildes hervorheben. Die einzelnen eindimensionalen Filtermasken tragen die englischen Bezeichnungen *Level (L)*, *Edge (E)*, *Spot (S)*, *Wave (W)* und *Ripple (R)* in Anlehnung an die Mikrostrukturen im Bild, die mit dem jeweiligen Filtern hervorgehoben werden sollen. Neben dem Namenskürzel wird in der Regel die Länge der eindimensionalen Filtermaske angegeben. So ergibt sich für einen 1d-Edge-Filter der Länge fünf die Bezeichnung *E5*. Bei Kombination zweier eindimensionaler Filtermasken mittels Faltung zu 2d-Filtern, werden die Bezeichnungen der Einzelfilter aneinander gereiht:

$$E5L5 = E5^T * L5. \quad (3.22)$$

In Abbildung 3-19 sind alle eindimensionalen Filtermasken der Länge fünf und die Bezeichnungen der daraus resultierenden 2d-Filter dargestellt.

<i>Level</i>	$L5 = [1 \ 4 \ 6 \ 4 \ 1]$	$L5L5$	$L5E5$	$L5S5$	$L5W5$	$L5R5$
<i>Edge</i>	$E5 = [-1 \ -2 \ 0 \ 2 \ 1]$	$E5L5$	$E5E5$	$E5S5$	$E5W5$	$E5R5$
<i>Spot</i>	$S5 = [-1 \ 0 \ 2 \ 0 \ -1]$	$\Rightarrow S5L5$	$S5E5$	$S5S5$	$S5W5$	$S5R5$
<i>Wave</i>	$W5 = [-1 \ 2 \ 0 \ -2 \ 1]$	$W5L5$	$W5E5$	$W5S5$	$W5W5$	$W5R5$
<i>Ripple</i>	$R5 = [1 \ -4 \ 6 \ -4 \ 1]$	$R5L5$	$R5E5$	$R5S5$	$R5W5$	$R5R5$

Abbildung 3-19: Aus den fünf eindimensionalen Filtermasken von Laws $L5$, $E5$, $S5$, $W5$, und $R5$ der Länge fünf werden mittels Faltung 25 zweidimensionale Filtermasken erstellt.

Die zweidimensionalen Filtermasken der Größe 5×5 gemäß Abbildung 3-19 werden verwendet, um geeignete Texturmerkmale, die Texturenergien von Laws, zu berechnen. Die fünfundzwanzig Ergebnisbilder der Faltungen des Eingangsbildes mit der jeweiligen Filtermaske werden zunächst über den geglätteten Mittelwert ($L5L5$) normiert und anschließend mittels einer nichtlinearen Fensterfunktion in einer Bildregion $M \times N$ zu einem Merkmal zusammengefasst. Als Kompromiss zwischen Rechenaufwand und Klassifikationsgenauigkeit verwendet Laws Bildregionen der Größe $M=N=15$. Als Fensterfunktion werden beispielsweise die Standardabweichung, die Summe der Absolutwerte oder die Summe der Quadrate verwendet [105]. Es ergeben sich für jede betrachtete Bildregion 24 Texturmerkmale, Laws' Texturenergien.

$L3$	$L5$	$L7$
$\begin{bmatrix} 1 & 2 & 1 \\ -1 & 0 & 1 \\ -1 & 2 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ -1 & -2 & 0 & 2 & 1 \\ -1 & 0 & 2 & 0 & -1 \\ -1 & 2 & 0 & -2 & 1 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ -1 & -4 & -5 & 0 & 5 & 4 & 1 \\ -1 & -2 & 1 & 4 & 1 & -2 & -1 \\ -1 & 0 & 3 & 0 & -3 & 0 & 1 \\ 1 & -2 & -1 & 4 & -1 & -2 & 1 \\ -1 & 6 & -15 & 20 & -15 & 6 & -1 \end{bmatrix}$
$E3$	$E5$	$E7$
$S3$	$S5$	$S7$
$W5$	$W7$	$R7$
	$R5$	$O7$

Abbildung 3-20: Laws' eindimensionale Filtermasken: *Level* (L), *Edge* (E), *Spot* (S), *Wave* (W) und *Ripple* (R).

Alternativ zu der Berechnung der Texturenergien werden verschiedene Kombinationen der eindimensionalen Filtermasken $L3$, $L5$, $L7$, $E3$, $E5$, $E7$ usw. für die Bildvorverarbeitung verwendet, um Bildstrukturen hervorzuheben oder zu unterdrücken.

In den Abbildungen 3-20 und 3-21 sind die gängigsten 1d- und 2d-Filtermasken im Detail aufgeführt. Die Filtermasken der Größe 5×5 werden im Anhang dargestellt. In der Regel werden die einzelnen Einträge der Filtermasken zusätzlich über die Summe der Filterelemente normiert, falls diese ungleich null ist.

$L3L3$			$L3E3$			$L3S3$		
1	2	1	-1	0	1	-1	2	-1
2	4	2	-2	0	2	-2	4	-2
1	2	1	-1	0	1	-1	2	-1
$E3L3$			$E3E3$			$E3S3$		
-1	-2	-1	1	0	-1	1	-2	1
0	0	0	0	0	0	0	0	0
1	2	1	-1	0	1	-1	2	-1
$S3L3$			$S3E3$			$S3S3$		
-1	-2	-1	1	0	-1	1	-2	1
2	4	2	-2	0	2	-2	4	-2
-1	-2	-1	1	0	-1	1	-2	1

Abbildung 3-21: Laws' Filtermasken der Größe 3×3 .

3.5.1.4 Autokorrelationskoeffizienten (AK)

Lineare räumliche Eigenschaften der Texturen innerhalb einer Bildregion $M \times N$ werden über die Korrelationskoeffizienten Akk_{ff} bestimmt. Für jeden Koeffizienten wird hierbei die normierte Summe des Produkts der Grauwerte der Region mit den Grauwerten der gleichen, verschobenen Region gebildet:

$$Akk_{ff}(p, q) = \frac{M \cdot N}{(M - p) \cdot (N - q)} \cdot \frac{\sum_{x=1}^{M-p} \sum_{y=1}^{N-q} g(x, y) \cdot g(x + p, y + q)}{\sum_{x=1}^M \sum_{y=1}^N g^2(x, y)}, \quad (3.23)$$

wobei es sich bei p und q um die ganzzahlige Verschiebungen der Region in x - bzw. y -Richtung handelt. In Abhängigkeit des Wertebereichs von p und q ergeben sich $(\max(p) - \min(p) + 1) \cdot (\max(q) - \min(q) + 1)$ Texturmerkmale. Für den Bereich für p und q von $[-5, 5]$ ergeben sich so beispielsweise 121 Merkmale. Im Folgenden wird die Anzahl der Merkmale über die ungerade Anzahl aller Verschiebungen in positive und negative Richtung p_{max} und q_{max} angegeben. Der Wertebereich für p ergibt sich somit zu $[-0,5 \cdot (p_{max} - 1), 0,5 \cdot (p_{max} - 1)]$ mit

insgesamt $p_{max} \cdot q_{max}$ Merkmalen. Bei einer rauen Textur nimmt die Autokorrelationsfunktion langsam mit zunehmendem Abstand ab. Bei einer feinen Textur nimmt die Autokorrelationsfunktion bei zunehmendem Abstand schnell ab.

3.6 Nichtlineare und orientierte Kerne (NLOK)

Inspiziert von der Kodierung der lokalen Orientierung (LOC) [43, 44] und lokalen Autokorrelationsmethoden (LAC) [87] wurde im Rahmen dieser Arbeit ein von Mayr und König [65, 85] entwickelter Texturoperator für die Vorverarbeitung und Merkmalsberechnung um die Auswertung zweiter Ordnungen erweitert [102, 103].

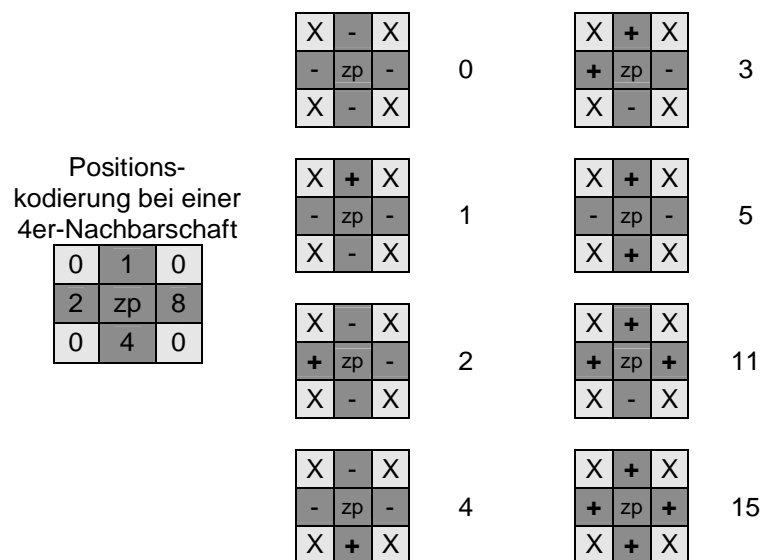


Abbildung 3-22: Ausgewählte Kodierungsergebnisse für Grauwertkombinationen in der 4er-Nachbarschaft. Positionen, die mit einem X markiert sind, werden aufgrund der Nachbarschaftskonfiguration nicht betrachtet. Ein Minuszeichen markiert Positionen, bei denen die Grauwertdifferenz des entsprechenden Bildpunktes zum Zentralpixel (zp) kleiner als ein festgelegter Schwellwert ist. Das Pluszeichen kennzeichnet Positionen, wo diese Differenz größer oder gleich dem Schwellwert ist.

3.6.1 Kodierung der lokalen Orientierung

Bei der Kodierung der lokalen Orientierung werden anhand der Relation eines Bildpunktes einer Nachbarschaft zu dem entsprechenden Zentral- oder Referenzpixel eindeutige Kodierungen zugewiesen. Neben der Position des Bildpunktes der Nachbarschaft wird die Grauwertdifferenz bezüglich des entsprechenden Zentralpixels betrachtet. Liegt diese oberhalb eines festgelegten Schwellwertes, wird die entsprechende Positionskodierung berücksichtigt. Die Kodierung der lokalen Orientierung ergibt sich aus der Summe aller berücksichtigten Positionskodierungen. Da

diese Summe in ihren Komponenten separierbar ist, bleibt die Information über die lokale Orientierung in der Kodierung erhalten. Abbildung 3-22 zeigt die eindeutige Positionskodierung und die daraus resultierende Kodierung der lokalen Orientierung beispielhaft an acht mögliche Grauwertkonfigurationen einer 4er-Nachbarschaft.

3.6.2 Lokale Autokorrelationsfunktion höherer Ordnung

Die diskrete lokale Autokorrelationsfunktion n -ter Ordnung (*higher order local autocorrelation*, HLAC) ist folgendermaßen definiert:

$$hlac(\mathbf{d}_1, \dots, \mathbf{d}_N) = \sum_{\mathbf{r}} h_1(\mathbf{r}) = \sum_{\mathbf{r}} g(\mathbf{r})g(\mathbf{r} + \mathbf{d}_1)g(\mathbf{r} + \mathbf{d}_2) \cdots g(\mathbf{r} + \mathbf{d}_N), \quad (3.24)$$

wobei \mathbf{r} der Positionsvektor der jeweiligen Referenzposition im Bild, $g(\mathbf{r})$ der Grauwert dieser Position ist und \mathbf{d}_i Verschiebungsvektoren repräsentieren. $h_1(\mathbf{r})$ ist ein lokales Bildmerkmal aus dem Produkt der Grauwerte an den entsprechenden relativen Bildpositionen und N die Ordnung von HLAC ist. Für $N = 1$ und $\mathbf{r} = [x, y]^T$ ergibt sich somit für das lokale Bildmerkmal $h_1(\mathbf{r})$, vgl. auch Formel (3.23):

$$h_1(x, y) = g(x, y)g(x + p, y + q). \quad (3.25)$$

Für unterschiedliche Ordnungs- und Verschiebungsconfigurationen lassen die dazugehörigen Autokorrelationskoeffizienten bestimmen und beispielsweise für die Gesichtserkennung verwenden [87].




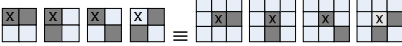
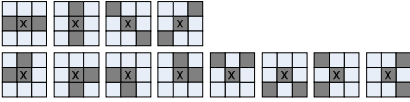
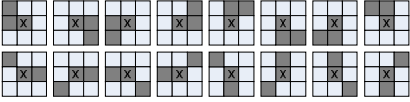
 <p>3x3 Umgebung</p>	<p>□: nicht betrachtete Bildpunkte ■: relevante Bildpunkte, Zentrum: , </p>
<p>K1: 4 Kerne</p>	 <p>Horizontal (H), vertikal (V) und diagonal(D)</p>
<p>K2: 12 Kerne</p>	 <p>H, V, D, rechte Winkel 1, rechte Winkel 2</p>
<p>K3: 16 Kerne</p>	 <p>Spitze und stumpfe Winkel</p>

Abbildung 3-23: 32 lokal-orientierte Strukturen.

Betrachtet man nur die lokale Autokorrelationsfunktion erster und zweiter Ordnung für ein 2d-Bild, so ergeben sich 32 mögliche Bildpunktkombinationen für eine lokale 3×3 Nachbarschaft gemäß Abbildung 3-23 und somit 32 lokale Bildmerkmale $h_{I_i}(\mathbf{r})$ mit $i = 1..32$. Die Bildpunktkombinationen entsprechen diskreten lokalen Strukturen, wie rechten, stumpfen oder spitzen Winkeln, sowie geraden Linien. Diese Strukturen sind in drei Gruppen mit je zwei oder drei relevanten Bildpunkten pro Nachbarschaft zusammengefasst.

3.6.3 Dominante Strukturen in einer lokalen Umgebung

Anstelle einer direkten Summation der lokalen Merkmale $h_{I_i}(\mathbf{r})$ in einem Bild oder einem Bildausschnitt gemäß Formel (3.24), wird, inspiriert von LOC, die dominante Struktur innerhalb der 3×3 Umgebung bestimmt. Die Dominanz einer Struktur ergibt sich aus der Größe des Merkmalswertes, also dem Produkt der Grauwerte an den zwei bzw. drei relevanten Grauwertpositionen. Für die Vergleichbarkeit der Merkmale erster bzw. zweiter Ordnung, werden die entsprechenden Grauwerte zusätzlich mittels der Quadrat- bzw. Kubikwurzel normiert. Als Rückgabewert dient die Identität i des zu der dominanten Struktur gehörenden Kerns, welche sich auf eine eindeutige lokale Kodierung gemäß Abschnitt 3.6.1 abbilden lässt.

3.6.4 Zurückweisung nicht dominanter Strukturen

Um zu verhindern dass aufgrund von statistischem Rauschen auch Strukturen in einer nahezu homogenen Umgebung oder nur schwach ausgeprägte Strukturen gefunden werden, können dominante Kerne zurückgewiesen werden.

Hierfür wird die Standardabweichung aller neun Grauwerte innerhalb der 3×3 Nachbarschaft bestimmt. Liegt der Wert unterhalb einer festgelegten Schwelle T_R , wird der zum höchsten Merkmalswert gehörende dominante Kern zurückgewiesen und die Kennzahl null zurückgegeben.

Tabelle 3-2. Mögliche Texturmerkmale bei NLOK

Texturmerkmal	Abkürzung	Anzahl der Merkmale
Histogramm erster Ordnung, Histogrammeinträge	NLOK-HIST	17 oder 33 (abhängig von der Anzahl der verwendeten Kerne)
Histogramm erster Ordnung, statistische Werte (NLOK-HISTS)	NLOK-HISTS	8
Histogramm zweiter Ordnung, Histogrammeinträge	NLOK-COOC	17^2 oder 33^2
Histogramm zweiter Ordnung, Haralick-Merkmale	NLOK-HARA1 NLOK-HARA2	5 (vgl. Abschn. 0) 13

3.6.5 Texturmerkmale

Das so gewonnene Ergebnisbild mit den Identitäten der dominanten bzw. der Kennzahl für zurückgewiesene Kerne kann beispielsweise direkt als Zwischenbild für die ikonische Bildvorverarbeitung verwendet werden. Alternativ werden innerhalb einer Bildregion Histogramme erster bzw. zweiter Ordnung und daraus abgeleitete Merkmale gemäß Abschnitt 3.5.1.1 und 3.5.1.2 aus den Identitäten für die Gewinnung aussagekräftiger Texturmerkmale verwendet (vgl. Tabelle 3-2).

3.6.6 Varianten von NLOK

Zurückweisung anhand lokaler Grauwertdifferenzen. Als Alternative zur Berechnung des Produkts der Grauwerte der betrachtenden Bildpunkte für die Bestimmung des dominanten Kerns wird die maximal auftretende Grauwertdifferenz für die Zurückweisung einer Struktur herangezogen. Dies erspart die rechentechnisch aufwendigere Berechnung der Kubik- oder Quadratwurzel. Allerdings entspricht die so bestimmte dominante Struktur nicht der mittels Multiplikation bestimmten Struktur (vgl. Abbildung 3-24). In Abhängigkeit der tatsächlich auftretenden Strukturen in einem Bild kann eine Variation der Bestimmung dominanter Strukturen jedoch durchaus von Vorteil sein.

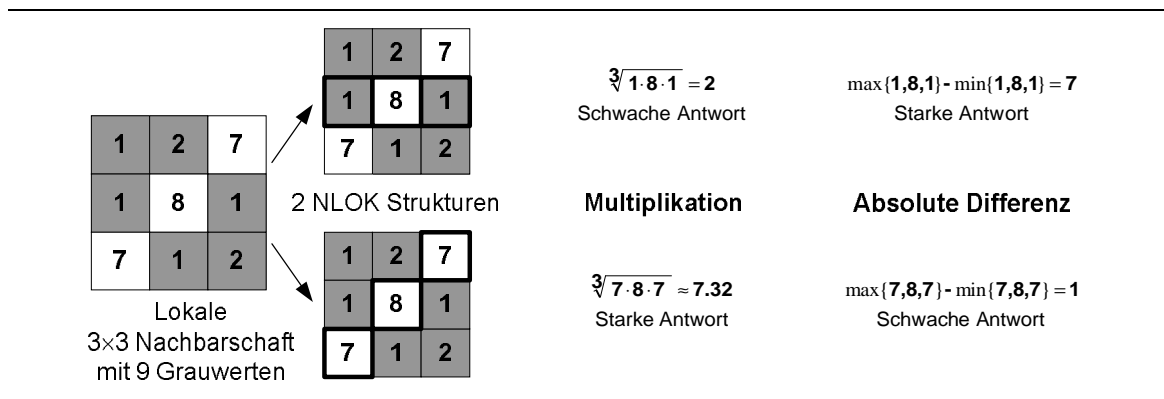


Abbildung 3-24: Vergleich zweier lokaler, orientierter Strukturen in einer 3x3 Nachbarschaft anhand der Kriterien „Multiplikation“ und „Absolute Differenz“. Je nach verwendetem Kriterium dominieren Strukturen mit unterschiedlichen Grauwertkombinationen. Mit dem Kriterium „Multiplikation“ gewinnen Strukturen, die hauptsächlich hohe Grauwerte beinhalten, bei der Verwendung der „Absoluten Differenz“ dominieren hingegen Strukturen mit einem hohen Grauwertkontrast innerhalb der Struktur.

Kerne höherer Ordnung. Die hier beschriebenen Verfahren für die Gewinnung von Texturmerkmalen lässt sich gleichermaßen auf Strukturen höherer Ordnung, d. h. mit einer größeren Anzahl relevanter Bildpunkte innerhalb einer lokalen Umgebung anwenden (vgl. Abbildung 3-25). Allerdings erhöht sich mit zunehmender Anzahl der relevanten Bildpunkte der Rechenaufwand für die Bestimmung der dominanten Strukturen.

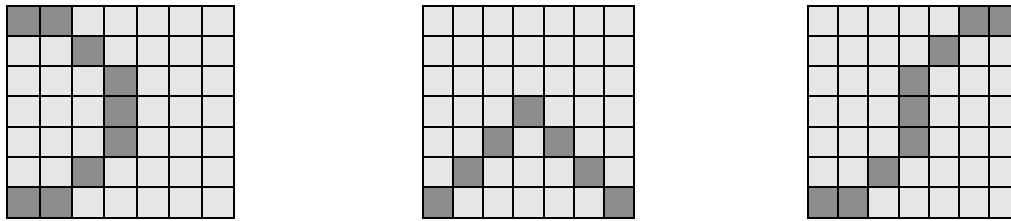


Abbildung 3-25: Drei mögliche Beispiele für Kerne höherer Ordnung in einer 7x7 Umgebung.

3.6.7 Vergleich von NLOK mit anderen Texturoperatoren

In vorangegangenen Untersuchungen [102, 103] hat sich gezeigt, dass mit NLOK berechnete Texturmerkmale im Vergleich zu den Texturoperatoren HIST, COOC, HISTS, LTEM und AK zu besseren Ergebnissen bei einer Texturklassifikation von 16 Beispieltexturen und bei der Defekterkennung von Kratzern auf Leder geführt haben. Die optimale Parametrierung aller Texturoperatoren sowie der Bildvorverarbeitung wurde in diesen Fällen automatisch optimiert. Dasselbe Verhalten zeigt sich ebenso bei der Untersuchung der Sensitivität der Klassifikatorparameter gemäß Kapitel 4.7.3.

3.7 Merkmalskombination und -selektion

Die Auswahl oder gezielte Kombination einzelner Texturmerkmale wird in dieser Arbeit nicht explizit behandelt. Eine optimale minimale Merkmalskombination kann jedoch unter Umständen zu besseren Ergebnissen bei der Texturanalyse führen. Bezüglich der Auswahl geeigneter Merkmale und der damit einhergehenden Dimensionsreduktion für die Texturanalyse sei beispielhaft auf folgende Literaturstellen verwiesen: [25, 64, 113, 142].

Implizit findet in der in dieser Arbeit betrachteten Optimierung gemäß Kapitel 6 eine Auswahl von Merkmalen einer Gruppe, d. h. zum Beispiel der Haralick-Merkmale oder der Laws' Texturenergien statt. Zusätzlich können Merkmale zweier Gruppen durch die optionale Verwendung eines Verarbeitungsoperators „Kombiniere Merkmale“ zu einem größeren Merkmalsvektor verknüpft werden.

3.8 Übersicht über die Methodenparameter der Texturanalysemethoden

Die folgende Tabelle zeigt die Konfigurationsparameter und der hier angeführten Methoden zur Texturanalyse:

Tabelle 3-3. Übersicht der Konfigurationsparameter und der resultierende Merkmale von Methoden zur Texturanalyse

Beschreibung	Parameter bzw. Abkürzung	Mögliche Werte bzw. Anzahl der Merkmale
Größe der betrachteten 2d-Bildregionen	M, N	$M, N \in \{2, 4, 6, \dots, 64\}$
Anzahl der verwendeten Grauwerte (Grauwertquantelung)	Q	$Q \in \{2, 4, 8, 16, 32, 64, 128, 256\}$
Grauerthistogramm		
<i>Histogrammeinträge</i>	HIST	G bzw. Q maximale Anzahl der Grauwerte
<i>Statistische Kennzahlen</i>	HISTS	8
Co-Occurrence-Matrix und Haralick-Merkmale		
Abstand zweier betrachteter Bildpunkte	$\Delta x, \Delta y$	$\Delta x, \Delta y \in \{-6, -5, \dots, 6\}$
Betrachtete Rotationsrichtungen	N_{Rot}	1, 2, 4, 8
<i>Histogrammeinträge</i>	COOC	G^2 bzw. Q^2
<i>Haralick-Merkmale</i>	HARA1 HARA2	5 13
Laws' Texturenergien		
<i>Texturenergien</i>	LTEM	24
Autokorrelationskoeffizienten		
Maximal Verschiebung	p_{max}, q_{max}	$p_{max}, q_{max} \in \{5, 7, 9, \dots, 19\}$
<i>Autokorrelationskoeffizienten</i>	AK	$p_{max} \cdot q_{max}$
Nicht-lineare und orientierte Kerne		
Anzahl der betrachteten Strukturen	NLOK _{16,32}	16 (K1 und K2) oder 32
Bestimmung dominanter Strukturen	NLOK ^M NLOK ^D	Multiplikation Abs. Differenz
Normalisierte Rückweisungsschwelle	T_R	$T_R \in \{1, \dots, 50\}$
<i>Histogramm erster Ordnung, Histogrammeinträge</i>	NLOK-HIST	17 oder 33 (abhängig von der Anzahl der verwendeten Kerne)
<i>Histogramm erster Ordnung, statistische Werte (NLOK-HISTS)</i>	NLOK-HISTS	8
<i>Histogramm zweiter Ordnung, Histogrammeinträge</i>	NLOK-COOC	17^2 oder 33^2
<i>Histogramm zweiter Ordnung, Haralick-Merkmale</i>	NLOK-HARA1 NLOK-HARA2	5 13

Die Parameterbereiche der einzelnen Parameter beruhen einerseits auf Erfahrungswerten oder orientieren sich an in der Literatur üblichen Werten. So hat sich beispielsweise der Wertebereich für die Rückweisungsschwelle T_R bei verschiedenen Texturen herausgestellt, um zwischen orientierter Vordergrundstruktur und Rauschen zu unterscheiden.

Je nach Anwendungsfall können die Parametergrenzen bzw. die Schrittweiten bei der Verwendung von Vorwissen eingegrenzt bzw. erweitert werden.

3.9 Zusammenfassung

In diesem Kapitel wurden einige bewährte Texturanalysemethoden, ihre Parameter und die resultierenden Merkmale näher vorgestellt. Neben dem neu entwickelten Texturoperator NLOK werden diese im Folgenden bei der Optimierung von Texturklassifikationsaufgaben herangezogen. In diesem Fall gelten, soweit nicht anders angegeben, die Parameterwertebereiche gemäß Tabelle 3-3.

Im folgenden Kapitel wird auf die verschiedenen Möglichkeiten der Bewertung eines Bildverarbeitungssystems eingegangen.

4 Bewertung eines Bildverarbeitungssystems

Die Bewertung eines Bildverarbeitungssystems kann je nach Einsatzgebiet des Systems global am Ende der Bildverarbeitungskette oder lokal an unterschiedlichen Stellen im Bildverarbeitungsablauf (vgl. Abbildung 4-26) erfolgen. Zur automatischen Optimierung oder Adaption einer BV-Kette bedarf es eines Mechanismus zur objektiven Bewertung ihrer Güte. Hierzu werden beispielsweise Fehlermaße zum Vergleich der Ergebnisse der Bildvorverarbeitung und erwünschten Ergebnisbildern eingesetzt.

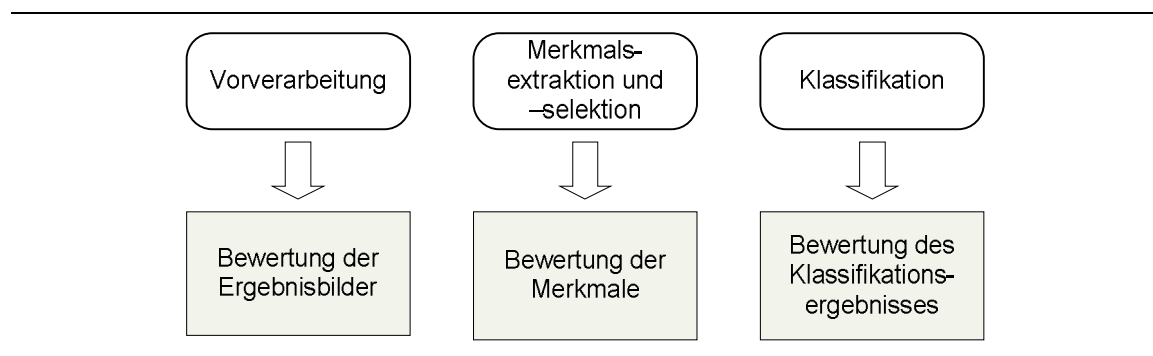


Abbildung 4-26: Bewertung verschiedener Module eines Bildverarbeitungssystems.

4.1 Filter- und Wrappermodelle für die Bewertung von BV-Systemen

Die Begriffe Filter- und Wrapper-Modelle [142] oder auch Methoden werden hauptsächlich im Zusammenhang mit der Merkmalsextraktion und -selektion verwendet. Sie lassen sich gleichermaßen für die Bewertung eines gesamten BV-Systems anwenden.

Der Unterschied zwischen Filter- und Wrapper-Modellen bei der Bewertung von BV-Systemen besteht darin, ob für die Bewertung des Gesamtsystems auf lernende Algorithmen wie beispielsweise Nächste-Nachbar-Klassifikatoren zurückgegriffen wird. Das Filter-Modell ist hierbei unabhängig von lernenden Algorithmen. Es liefert in der Regel bessere Generalisierungsergebnisse und ist weniger rechenintensiv [64].

Bei dem Wrapper-Modell werden lernende Algorithmen in die Bewertung mit einbezogen. Dadurch werden in der Regel bessere Klassifikationsergebnisse erreicht. Allerdings ist dieses Modell insbesondere für große Datensätze deutlich rechenzeitintensiver und das Leistungsvermögen kann unter Umständen deutlich von der Wahl des Lerndatensatzes abhängen [25].

Je nachdem, welche der folgenden Kriterien für die Bewertung der einzelnen BV-Module bzw. des Gesamtsystems verwendet werden, handelt es sich dementsprechend um ein Filter- bzw. ein Wrapper-Modell.

4.2 Bildähnlichkeit

Für die Bewertung der Bildvorverarbeitung eignen sich Ähnlichkeitsmaße zwischen den Sollbildern, d. h. der manuellen Segmentierung, und dem Ergebnisbild der Vorverarbeitung. Bei dem Ergebnis der Vorverarbeitung kann es sich um Binärbilder aufgrund von Schwellwertoperationen oder um Grauwertbilder handeln.

4.2.1 Bildvergleich bei Binärbildern

Für einen einfachen Bildvergleich von Binärbildern wird die auf eins normierte Anzahl der übereinstimmenden Grauwerte $s=s(x,y)$ bzw. $b=b(x,y)$ in einem Soll- (S) und einem Ist-Bild (B) gemäß der folgenden Formel gebildet:

$$VBB(s,b) = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \delta_{sb}(x,y), \text{ mit } \begin{cases} \delta_{sb} = 1 & \text{für } s = b \\ \delta_{sb} = 0 & \text{sonst} \end{cases} \quad (4.26)$$

Je höher die Summe ist, desto ähnlicher sind die betrachteten Bilder.

4.2.2 Summe der quadrierten Grauwertdifferenzen

Als einfaches Maß für die Ähnlichkeit zweier Bilder oder Bildregionen wird die Summe der quadrierten Grauwertdifferenzen SQG gemäß der folgenden Formel betrachtet:

$$SQG(s,b) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (s(x,y) - b(x,y))^2 \quad (4.27)$$

Je geringer die Summe ist, desto ähnlicher sind die betrachteten Bilder.

4.2.3 Gegenseitige Information oder Mutual Information

Unter der „Mutual Information“ (MI) oder der „gegenseitigen Information“ versteht man ein auf der relativen Entropie basierendes Maß aus der Informationstheorie [7]. Die gegenseitige Information ist die relative Entropie oder Kullback-Leibler-Abstand KLD [7] zwischen der gemeinsamen Verteilung $p(x, y)$ und dem Produkt der Verteilungen $p(x)$ und $q(x)$ gemäß der folgenden Formeln.

Relative Entropie oder Kullback-Leibler-Distanz KLD zwischen zwei Verteilungen $p(x)$ und $q(x)$:

$$KLD(p \parallel q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \quad (4.28)$$

Berechnung der gegenseitigen Information:

$$MI(p(x, y) \parallel p(x)q(y)) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)q(y)} \quad (4.29)$$

Für die Bildverarbeitung werden für $p(x)$ und $q(x)$ die relativen Histogramme $P(g_1)$ und $P(g_2)$ (Verteilungsdichten der Grauwertintensitäten) zweier Bildausschnitte verwendet. In der BV wird MI beispielsweise für die Bildregistrierung multimodaler Bildsignale eingesetzt [152].

4.3 Merkmalsbewertung

Sobald Verfahren für die Merkmalsberechnung und die Klassifikation eingesetzt werden, kann eine Bewertung des Systems optional anhand der Merkmale oder des Klassifikationsergebnisses erfolgen. Die Bewertung von Merkmalen wird beispielsweise für die Merkmalsselektion verwendet. Insbesondere wird hierbei untersucht, wie gut verschiedene Klassen im Merkmalsraum voneinander separierbar sind. Diese Trennbarkeit hängt dabei stark von dem verwendeten Klassifikator ab. In der Regel werden aus diesem Grund Gütemaße für Bewertung und Auswahl von Merkmalen verwendet, die sich an der Leistungsfähigkeit des nachfolgenden Klassifikators orientieren. Beispiele für Gütemaße bei der Merkmalsselektion sind die Kullback-Leibler-Distanz bei Verwendung von Normalverteilungsklassifikatoren [7], nichtparametrische Überlappungsmaße bei Verwendung von Nächste-Nachbar-Klassifikatoren oder die Relevanz eines Neurons bei Verwendung von Neuronalen Netzen [98].

4.4 Klassifikation

Anschließend an die Berechnung und Auswahl von Merkmalen werden Klassifikatoren eingesetzt, um die Zugehörigkeit eines Merkmalsvektors $\mathbf{m} = [m_1, m_2, \dots, m_n]$ zu einer bestimmten Klasse $\omega_i, i = 1..L$ zu bestimmen. Im Folgenden werden einige Klassifikatortypen und ihre Parameter vorgestellt, die im Rahmen der Bildsegmentierung und Texturanalyse verwendet werden.

4.4.1 Schwellwertklassifikator

Bei diesem Klassifikationsverfahren werden pro Merkmal jeweils eine obere und untere Schwelle anhand der Verteilungen der Merkmale pro Klasse bestimmt. Dieser Akzeptanzbereich ist in der Regel ein Vielfaches der Streuung und liegt symmetrisch um den

Mittelwert. Werte, die außerhalb des Akzeptanzbereichs liegen, führen zur Rückweisung des zu prüfenden Merkmalsvektors. Dieser Klassifikator lässt sich beispielsweise für die Erkennung von Neuheiten oder Anomalien in einer Textur [89, 135, 156] einsetzen, wenn aufgrund fehlender Daten keine überwachte Klassifikation eingesetzt werden kann. Der Schwellwertklassifikator lässt sich in der Regel nur für zwei Klassen praktikabel anwenden. Das gleiche Prinzip wird auch bei Entscheidungsbäumen, wie beispielsweise Klassifikations- und Regressionsbäume (CART) [7] verwendet.

4.4.2 Abstandsklassifikatoren

Bei einem Abstandsklassifikator wird für jede Klasse ω_i ein Merkmalsvektor als Prototyp oder Referenz r_i abgespeichert. Die Zuweisung eines zu klassifizierenden Merkmalsvektors erfolgt über die Abstände d der Merkmale zu den gespeicherten Prototypen. Der Abstand wird in der Regel über die Summe der quadratischen Abweichungen über alle Merkmale bestimmt. Weitere Abstandsmaße sind beispielsweise die *KLD* oder die Korrelation.

4.4.3 Nächste-Nachbar-Klassifikator

Beim Nächste-Nachbar-Klassifikator wird ein zu prüfender Merkmalsvektor des Testdatensatzes anhand seiner nächsten Nachbarn aus dem Lern- oder Trainingsdatensatz klassifiziert. Der Lerndatensatz besteht aus einer Vielzahl von Referenzen für jede Klasse. m wird der Klasse seines nächsten Nachbarn mit dem geringsten Abstand zugeordnet. Werden die k nächsten Nachbarn betrachtet, so handelt es sich um einen k -Nächste-Nachbar (kNN) Klassifikator. Tritt bei einer kNN-Klassifikation eine Patt-Situation mit einer gleichen Anzahl von nächsten Nachbarn für mehrere Klassen auf, kann beispielsweise die mittlere Distanz zu den Nachbarn pro Klasse für die Entscheidungsfindung herangezogen werden. Um solche Patt-Situation zu vermeiden wird bei einer 2-Klassen-Klassifikation k zweckmäßigerweise ungerade gewählt.

Der Vorteil bei einem kNN besteht darin, dass keinerlei Voraussetzungen über die statistische Verteilung der einzelnen Merkmale getroffen wird und dass ein Training des Klassifikators im eigentlichen Sinne nicht notwendig ist, da m nur mit den Referenzen des Lerndatensatzes verglichen wird. Bei sehr großen Datenmengen oder hochdimensionalen Merkmalsvektoren kann die Suche nach den nächsten Nachbarn mitunter jedoch sehr zeitaufwendig sein. Mit einer zusätzlichen Trainingsphase kann in diesem Fall Abhilfe durch eine Beschränkung auf repräsentative Prototypen liefern

4.4.4 Support-Vektor-Maschinen

Bei der Verwendung von Support-Vektor-Maschinen [7] für die Klassifikation wird die Trennebene zwischen zwei Klassen durch Stützvektoren, den so genannten Support-Vektoren bestimmt (Abbildung 4-27). Dies sind die Vektoren aus dem Lerndatensatz, die der Trennebene am nächsten liegen. Um auch linear nicht trennbare Klassen voneinander separieren zu können, werden die Merkmalsvektoren mittels nichtlinearer Kernfunktionen transformiert und in einen höherdimensionalen Merkmalsraum überführt. In einem Raum mit genügend hoher Dimensionszahl wird jede Vektormenge linear separierbar. Dies entspricht dann einer nichtlinearen Klassifikation im Merkmalsraum. Als Kernfunktion können beispielsweise Radial-Basis-Funktionen (RBF) oder polynomische Kernfunktionen verwendet werden:

$$RBF : kn(\mathbf{m}, \mathbf{m}') = e^{(-\gamma \|\mathbf{m} - \mathbf{m}'\|^2)}, \gamma > 0 \quad (4.30)$$

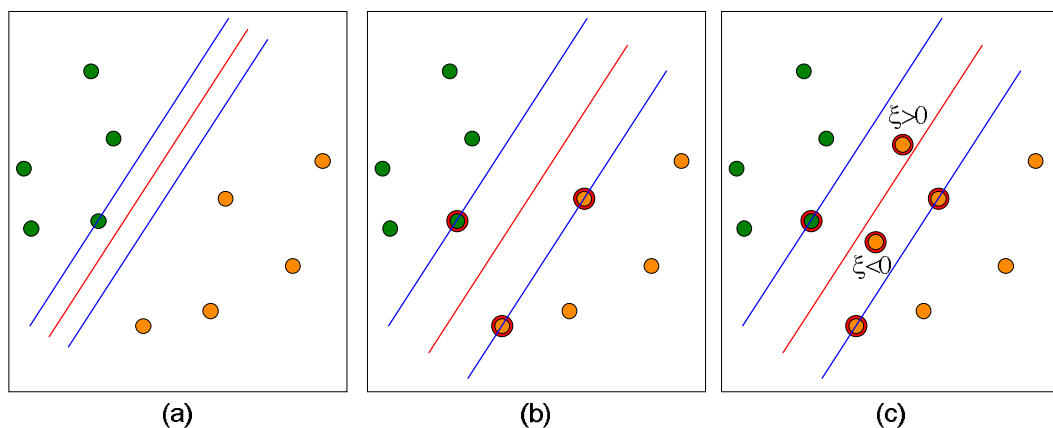


Abbildung 4-27: Bestimmung linearer Trennebenen mit SVM (a). Der Abstand zwischen der Trennebene (rot) und den Merkmalsvektoren wird maximiert. Die Vektoren, die der Trennlinie am nächsten liegen, sind die so genannten Stützvektoren (b). Die Verwendung von Schlupfvariablen ξ_i erlaubt es dem Klassifikator einzelne Merkmalsvektoren bei der Suche nach der optimalen Trennebene falsch zu klassifizieren (c).

Um zu berücksichtigen, dass die Trainingsdaten in der Regel nicht streng linear separierbar sind, werden bei der C -SVM zusätzlich Schlupfvariablen ξ_i eingeführt. Diese erlauben es dem Klassifikator, einzelne Merkmalsvektoren bei der Suche nach der optimalen Trennebene falsch zu klassifizieren. Eine mit dem Regularisierungsfaktor $C > 0$ gewichtete Summe aller ξ_i fließt in das der SVM zugrunde liegende Minimierungsproblem mit ein, wobei Merkmalsvektoren, die jenseits der Trennebene liegen, mit einem positiven ξ_i bestraft werden [7].

4.5 Bewertung des Klassifikationsergebnisses

Unabhängig von der Art des Klassifikators kann das Ergebnis der Klassifikation nach dem Training anhand der Klassifikationsraten bewertet werden (Wrapper-Modell). Die Klassifikationsrate wird in unterschiedlicher Form häufig für die Bewertung von Defekterkennungsverfahren herangezogen.

4.5.1 Maximierung der Klassifikationsrate R_{OK} der korrekt klassifizierten Bildregionen

Als Grundlage für die Bewertung regionenbasierter Verfahren dient die Klassifikationsrate R_{OK} der korrekt klassifizierten Bildregionen R :

$$R_{OK} = \frac{1}{L} \sum_{i=1}^L \frac{R_{i,OK}}{R_i}, R_i > 0 \quad (4.31)$$

mit $R_{i,OK}$: korrekt klassifizierten Regionen der Klasse ω_i , R_i : Anzahl aller Regionen der Klasse ω_i und L : Klassenanzahl. Bei einer zusätzlichen Kreuzvalidierung (CV) wird die mittlere Klassifikationsrate $R_{OK,CV}$ verwendet

4.5.2 Minimierung der erwarteten Kosten

Je nach Klassifikationsaufgabe kann es zusätzlich sinnvoll sein, Fehlklassifikationen verschiedener Klassen unterschiedlich zu gewichten, d. h. eine spezifische Kostenmatrix (vgl. Abbildung 4-28) verwenden. Diese wird in der industriellen Bildverarbeitung beispielsweise eingesetzt, wenn ein zu prüfendes Objekt nur wenige Cent kostet, ein nicht erkannter Defekt jedoch hohe Folgekosten verursacht. Die erwarteten Kosten sollen minimal sein.

	Defekt	kein Defekt
Defekt	0	100
kein Defekt	1	0

Abbildung 4-28: Beispiel für eine Kostenmatrix. Die Nicht-Erkennung eines Defektes kostet 100 Einheiten. Wird hingegen ein fehlerfreies Objekt der Defektklasse zugeordnet, so kostet dies nur eine Einheit. Ziel der Klassifikation ist eine Minimierung der Kosten.

4.6 Kombination verschiedener Bewertungsmethoden

Durch eine Kombination verschiedener Bewertungsmethoden bzw. Gütemaße lässt sich unter Umständen ein besseres Gesamtergebnis eines BV-Systems bezüglich einer oder mehrerer Problemstellungen erreichen.

Eine Verknüpfung der Gütemaße zu einem einzigen Gesamtmaß lässt sich beispielsweise über eine gewichtete Akkumulierung der Einzelmaße erreichen (vgl. Abbildung 4-29). Hierbei ist es sinnvoll Maße zu verwenden, die auf einen vergleichbaren Wertebereich normierbar sind, um ungewollte Dominanz einzelner Maße in der gewichteten Summe zu vermeiden.

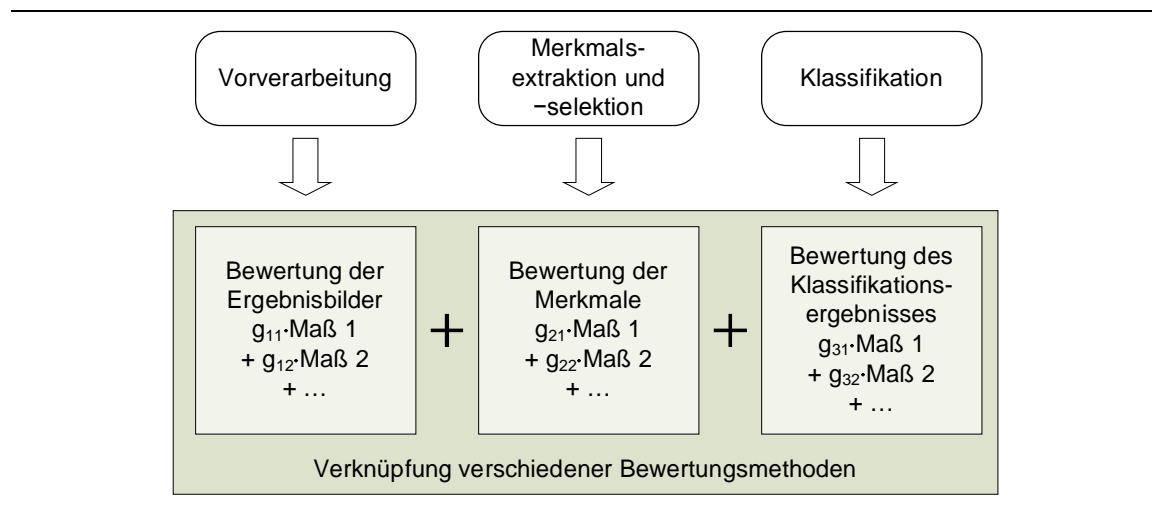


Abbildung 4-29: Gewichtete akkumulative Verknüpfung verschiedener Bewertungsmethoden und -maße zu einem Gesamtmaß.

Die Gütemaße der einzelnen Stufen eines BV-Systems können beispielsweise dazu verwendet werden, eine hierarchische Optimierung oder eine Mehrzieloptimierung, z. B. über eine gewichtete Summe oder Pareto-Optimierung, zu realisieren.

Eine konkrete Kombination verschiedener Gütemaße wird in dieser Arbeit nur am Rande im Rahmen der Mehrzieloptimierung mittels Bestrafungs- oder Belohnungsthermen diskutiert.

4.7 Vergleichende Untersuchungen

4.7.1 Bildähnlichkeit zwischen zwei Grauwertbildern

Im Folgenden werden die aufgeführten Methoden *SQG*, *KLD* und *MI* für die Bestimmung der Bildähnlichkeit zwischen zwei Bildern näher beleuchtet. Zu diesem Zweck werden verschiedene potenzielle Ergebnisbilder eines Segmentierungsalgorithmus mit einem manuell erstellten Sollbild (S1, siehe Abbildung 4-30) mit einer Markierung eines Oberflächenfehlers (hier: schwarz auf weißem Hintergrund) verglichen und die Bildähnlichkeitsmaße berechnet. Bei den untersuchten Bildern gemäß der Abbildung handelt es sich um das originale Grauwertbild einer Deckenplatte mit einem Oberflächenfehler (B1), eine verkleinerte und vergrößerte Markierung des Fehlers (B2, B3), den Rand des Fehlers (B4), ein Bild mit gleichförmig verteiltem Rauschen (B5), S1 mit zusätzlichem Gaußschen Rauschen (B6) und ein Segmentierungsergebnis (B7), in dem neben dem Fehler auch die Struktur des Hintergrundes segmentiert wurde. Zusätzlich

wurde die Bildähnlichkeit mit dem Sollbild (maximale Bildähnlichkeit) selbst, dem invertierten Sollbild, einem rein schwarzen (gesamtes Bild als Fehler) bzw. einem weißen Bild (gesamtes Bild ist Hintergrund) bestimmt.

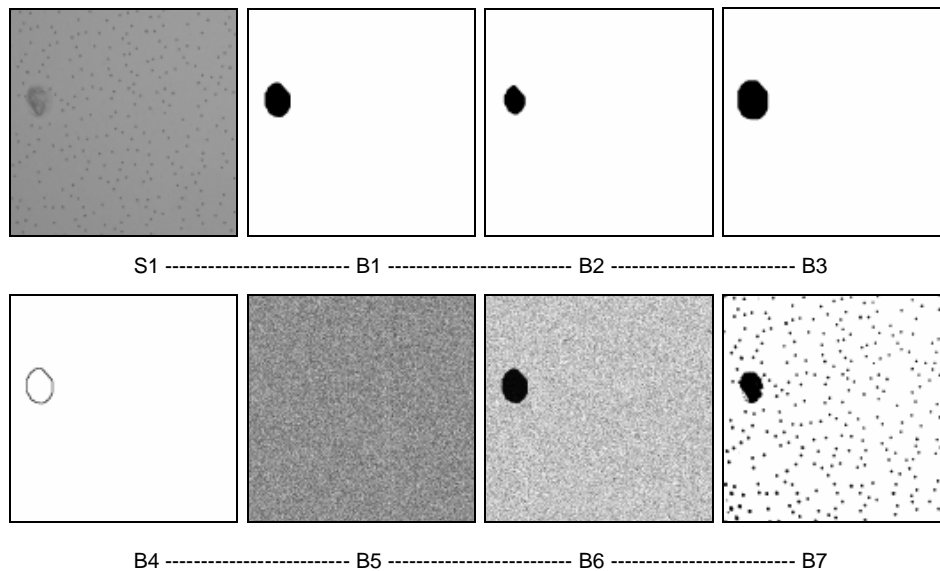


Abbildung 4-30: Soll- und Ist-Bilder für die Berechnung der Bildähnlichkeit. Obere Reihe von links nach rechts: Deckenplatte mit Fehler, Markierung des Fehlers, verkleinerte Markierung des Fehlers, Vergrößerte Markierung des Fehlers. Untere Reihe: Rand der Fehlermarkierung, uniform verteiltes Rauschen, Gaußsches Rauschen, Fehler und Hintergrundstruktur²¹.

B2 und B3 wurden gewählt, da es im Bereich der Oberflächenprüfung in vielen Fällen akzeptabel ist, wenn ein Oberflächenfehler nur grob segmentiert wird. Eine pixelgenaue Segmentierung spielt für eine erste Detektion und die Frage nach der Anwesenheit des Fehlers in der Regel keine bzw. nur eine untergeordnete Rolle. Ein geeignetes Maß für die Bildähnlichkeit sollte auch in diesen Fällen einen hohen Ähnlichkeitswert liefern. Zusätzlich ist der zeitliche Aufwand für die Erstellung einer manuellen Segmentierung geringer, wenn auch diese nicht pixelgenau sein muss. Neben der Segmentierung kann der Einfluss des Rauschens auf die Bildähnlichkeit eine Rolle spielen. Dies wird anhand der Bilder B5 und B6 untersucht. Mit B7 wird der Fall untersucht, wenn der eigentliche Fehler prinzipiell gut erkannt wird, es aber zusätzlich Segmentierungsfehler im Hintergrund gibt.

4.7.2 Ergebnisse und Diskussion

In Tabelle 4-4 sind die berechneten Maße KLD , MI und SQG für die Bildähnlichkeit der Bildvergleiche aufgeführt. Bei KLD und SQG ist die Bildähnlichkeit am größten, wenn der Wert

²¹ Bildmaterial: Fraunhofer ITWM

dieser Abstandsmaße gegen Null geht. Für eine bessere Vergleichbarkeit wurden die Wertebereiche auf das Intervall $[0,1]$ abgebildet, wobei der Wert 1 einer maximalen Bildähnlichkeit, d. h. identische Bilder, entspricht. MI und SQG lassen sich einfach über die maximal mögliche Bildähnlichkeit $MI(S1, S1)$ bzw. über die in einem Bild maximal mögliche Grauwertdifferenz normieren. Für KLD ist dies nicht ohne weiteres möglich. Um trotzdem eine gewisse Vergleichbarkeit zu erreichen, wurde KLD über einen subjektiv gewählten Faktor (angelehnt an die maximal aufgetretene Distanz) normiert.

Tabelle 4-4. Bildähnlichkeit zu S1

	KLD	MI	$1-SQG$	$MI/MI(S1,S1)$	$1-KLD/10$
S1	0,00000	3,14894	1,00000	1,00000	1,00000
S1 Invertiert	4,16473	3,14894	0,00000	1,00000	0,58353
B1	9,41686	2,61217	0,71880	0,82954	0,05831
B2	0,00098	3,11905	0,99520	0,99051	0,99990
B3	0,00103	3,13710	0,99443	0,99624	0,99990
B4	0,00766	3,08201	0,98639	0,97875	0,99923
B5	6,85252	1,37671	0,66711	0,43720	0,31475
B6	2,49949	1,60856	0,83949	0,51083	0,75005
B7	3,92828	3,10045	0,03746	0,98460	0,60717
Schwarzes Bild	4,29463	3,08510	0,01358	0,97973	0,57054
Weißes Bild	0,01363	3,08510	0,98642	0,97973	0,99864

Der optimale Ähnlichkeitswert wird wie erwartet in allen drei Fällen bei identischen Bildern erreicht. Bei MI ist die Bildähnlichkeit ebenfalls maximal, wenn anstelle von S1 das invertierte Bild von S1 verwendet wird. Bei KLD nimmt die Bildähnlichkeit in diesem Fall ab, bei SQG wird sie sogar minimal. Soll ein Bild unter Verwendung des Ähnlichkeitsmaßes automatisch binarisiert, d. h. in zwei Klassen unterteilt werden, sollte bei KLD und SQG darauf geachtet werden, für helle Objekte auf dunklem Untergrund eine helle Objektmarkierung bzw. umgekehrt zu wählen oder ggf. einen optionalen Invertierungsoperator zu verwenden. MI liefert in beiden Fällen dasselbe Ergebnis. MI nimmt deutlich ab, wenn sich Rauschen im Vergleichsbild befindet, da es hierbei deutliche Unterschiede in den Grauerthistogrammen der beiden Bilder gibt. Für B2 und B3 werden wie erwünscht in allen drei Fällen hohe Ähnlichkeitswerte erreicht.

Für Vergleichsbilder mit ausschließlich Hintergrund (hier: schwarzes Bild) bzw. Vordergrund (weißes Bild) wird bei SQG und KLD eine sehr große bzw. eine sehr kleine Bildähnlichkeit erreicht. Dies ist abhängig von der Pixelanzahl der Objektpixel. Bei MI ist die Bildähnlichkeit in beiden Fällen sehr groß. Für den Einsatz bei Optimierung und der Bildbinarisierung kann dieses Verhalten von Nachteil sein, da ein monochromes Ergebnisbild in der Regel nicht sinnvoll weiterverwendet werden kann, wenn vorhandene Strukturen oder Objekte im Bild erkannt werden sollen.

Für den Einsatz der Bildähnlichkeit für den Vergleich von automatisch generierten Ergebnisbildern mit einem Referenzbild bieten sich abschließend sowohl *SQG* als auch *MI* an. Im Fall eines bimodalen Referenzbildes und eines monochromatischen Vergleichsbildes wird der Ähnlichkeitswert in beiden Fällen auf null, d. h. auf eine minimale Bildähnlichkeit gesetzt, um solche Ergebnisbilder bei der Autokonfiguration oder Optimierung zu vermeiden. Im Fall von *SQG* wird zusätzlich ein optionaler Operator für die Bildinvertierung verwendet.

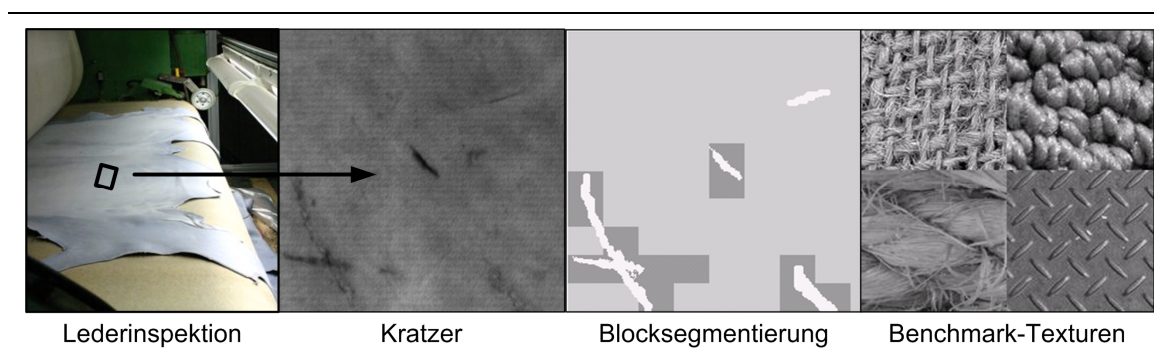


Abbildung 4-31: Kratzererkennung auf Leder sowie Beispieltexturen für die Texturklassifikation.

4.7.3 Sensitivität der Klassifikatorparameter bei der Texturklassifikation

Für die Bewertung von Verfahren für die Erkennung von Oberflächendefekten oder die Texturklassifikation wird in vielen Fällen die Klassifikationsrate [156] herangezogen.

Im Folgenden wird die Sensitivität der Klassifikatorparameter von kNN und C-SVM in Abhängigkeit der Fenstergröße und des Texturoperators bei einer reinen Texturklassifikation ohne zusätzliche Bildvorverarbeitung untersucht. Hierfür werden zwei Datensätze *TC00001* und *TC00002* der OUTEX-Datenbank²² [95] verwendet. Diese Datensätze enthalten Beispielbilder der Größe 64x64 (*TC00001*) bzw. 32x32 (*TC00002*) Bildpunkte von 24 verschiedenen Texturklassen. Abbildung 4-32 zeigt je ein Beispielbild der verwendeten stochastischen und periodischen Texturen.

²² Outex Texture Database, <http://www.outex.oulu.fi> (letzter Zugriff: 30.11.2008)

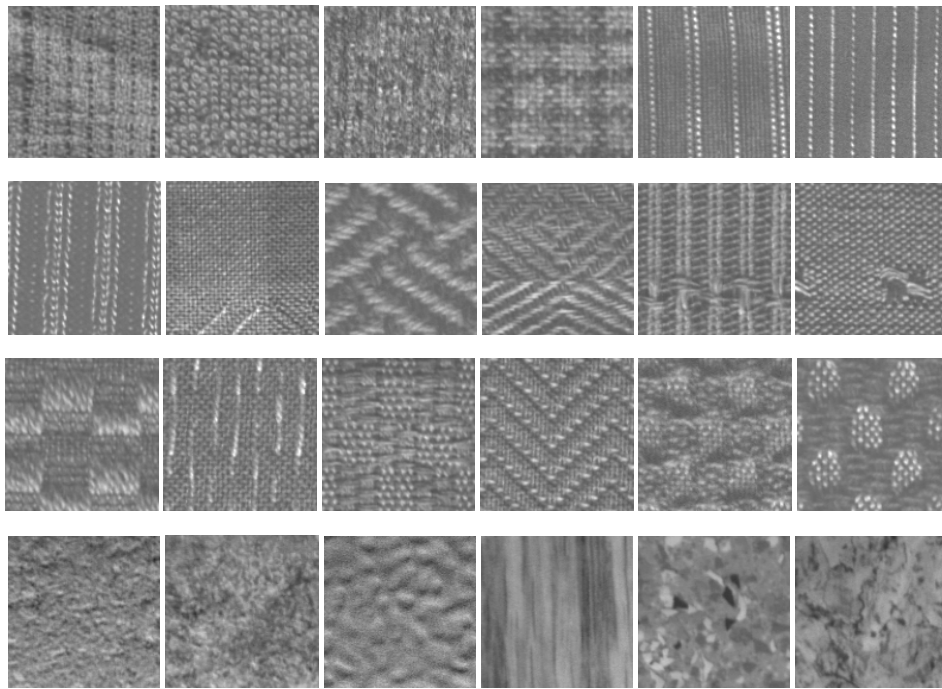


Abbildung 4-32: Beispielbilder der 24 Texturklassen der OUTEX TC-Datensätze.

Die Sensitivität der Klassifikatorparameter wird in Abhängigkeit des Texturoperators und der Fenstergröße untersucht. Hierbei werden je 50 Texturbilder pro Texturklasse verwendet. Die gewählte Regionengröße entspricht der Bildgröße, d. h. 32×32 bzw. 64×64 Bildpunkte pro Bild. Die Parameter und die Anzahl der Texturmerkmale der betrachteten Texturoperatoren LTEM, AK, HIST, HARA1, NLOK-HIST und NLOK-COOC sind in Tabelle 4-5 aufgeführt. Die Parametrierung wurde so gewählt, dass bei voller Grauwertauflösung eine minimale Anzahl von Merkmalen generiert wird. Insbesondere bei HARA1 und NLOK sind die Parameter nicht optimiert, d. h. an die Klassifikationsaufgabe angepasst.

Tabelle 4-5. Parameter der Texturoperatoren

	LTEM	AK	HIST	HARA1	NLOK-HIST	NLOK-COOC
Parameter	---	$p_{max}=5$ $q_{max}=5$	$G=256$	$G=256$ $\Delta x=1$ $\Delta y=-1$ $N_{Rot}=1$	$NLOK_{16}^{-M}$ $\Delta x=1$ $\Delta y=-1$ $N_{Rot}=1$ $T_R=0$	$NLOK_{16}^{-M}$ $\Delta x=1$ $\Delta y=-1$ $N_{Rot}=1$ $T_R=0$
Anzahl der Merkmale	24	25	256	5	17	289

Tabelle 4-6 zeigt die untersuchten Parameter der Klassifikatoren kNN und C-SVM. Für kNN werden elf Parameterkonfigurationen betrachtet, für C-SVM 1116 Konfigurationen. Für die Bestimmung der mittleren Klassifikationsrate $R_{OK,CV}$ wird eine 2-fach Kreuzvalidierung verwendet.

Tabelle 4-6. Parameter der Klassifikatoren

	<i>kNN</i>	<i>SVM</i>
Parameter	$k = 1 + i \cdot 2, i \in \{0, 1, 2, \dots, 10\}$	$\gamma = 2^a, a \in \{-15, -14, \dots, 15\}$ $C = 2^b, b \in \{-10, -9, \dots, 25\}$
Anzahl der Parameterkonfigurationen	11	1161

4.7.4 Ergebnisse und Diskussion

Abbildung 4-33 zeigt die mittleren Klassifikationsraten von kNN für die Outex-Texturdaten *TC00002* (links) und *TC00001* (rechts) bei Verwendung der einzelnen Texturoperatoren. In Abhängigkeit von der Fenstergröße und dem gewählten Texturoperator steigt (z. B. NLOK32, (32×32 Bildpunkte), LTEM64 (64×64 Bildpunkte)) oder sinkt (AK, HIST) die Klassifikationsrate mit einer zunehmendem Anzahl von Nachbarn k . Eine generelle Operator-übergreifende Aussage bezüglich des besten Wertes für k kann somit nicht getroffen werden.

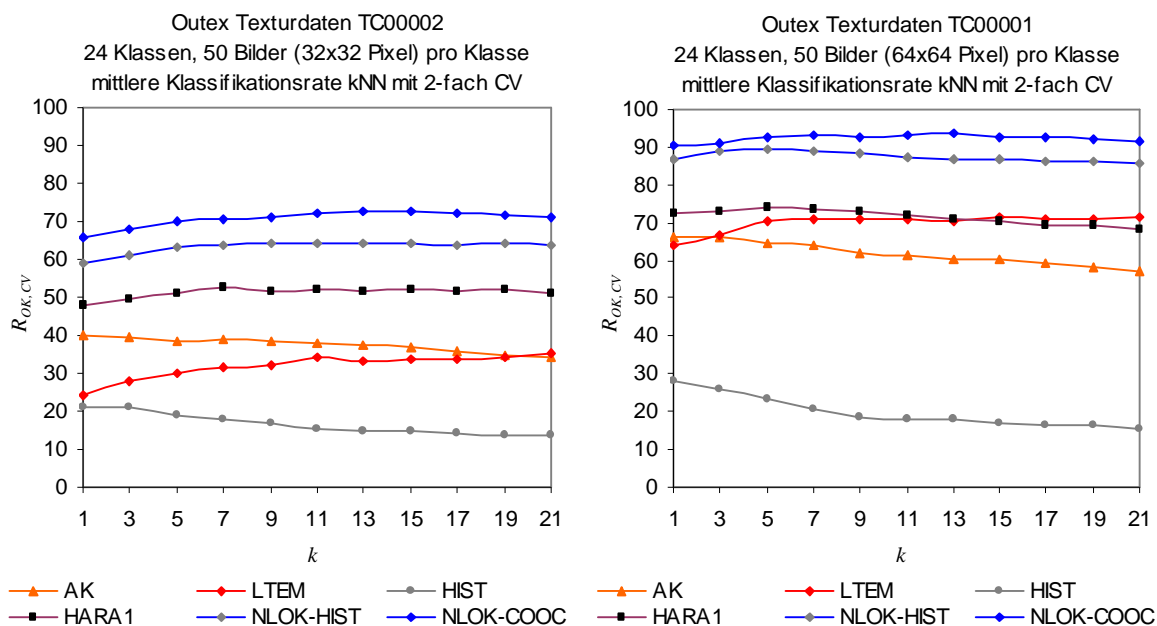


Abbildung 4-33: Mittlere Klassifikationsrate (2-fach Kreuzvalidierung) in Abhängigkeit von k bei einer kNN-Texturklassifikation. Links: Je 50 Bilder pro Texturklasse der Outex Texturdatenbank *TC00002* (32×32 Pixel). Rechts: Je 50 Bilder pro Klasse der Outex Texturdatenbank *TC00001* (64×64 Pixel).

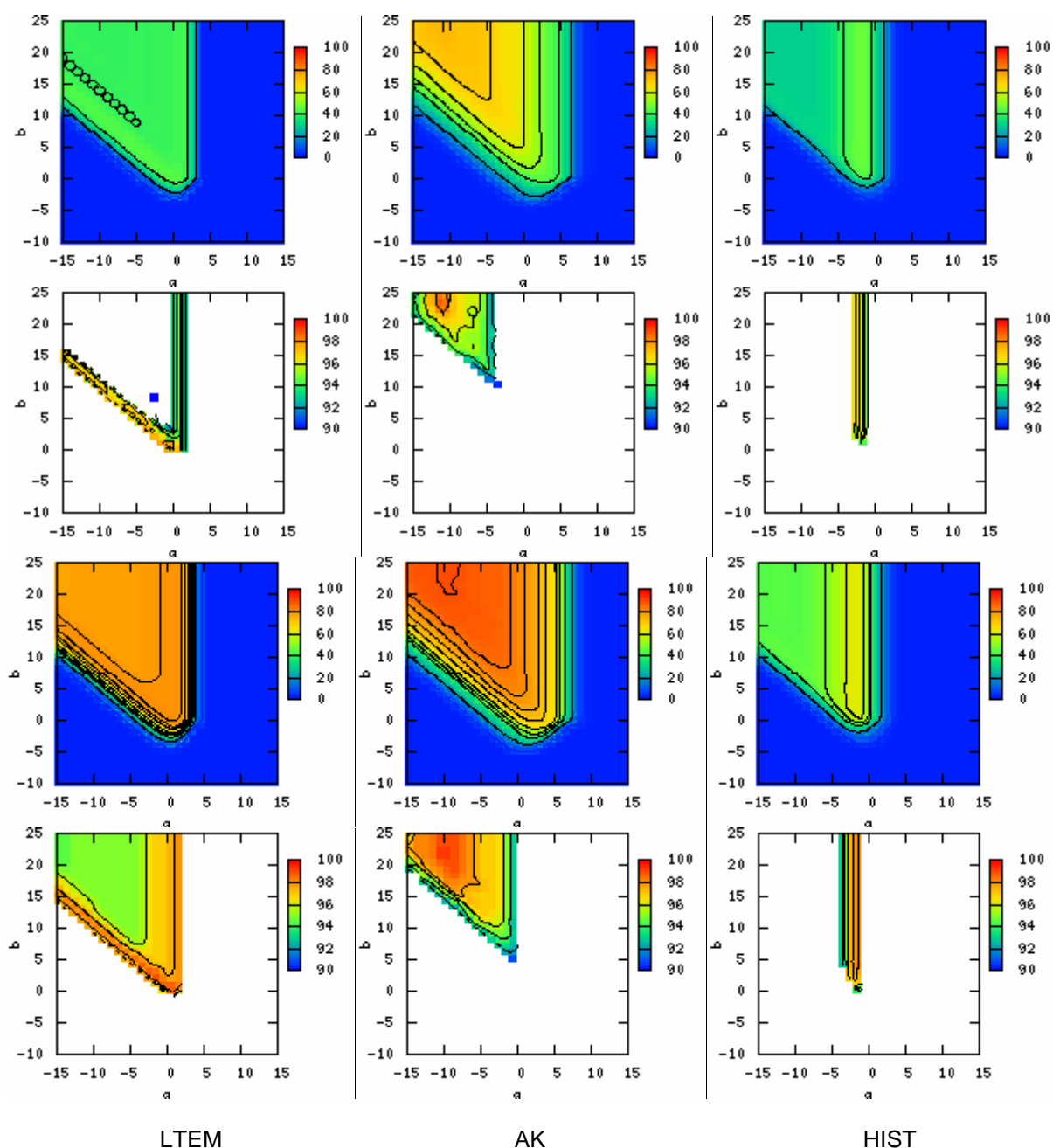


Abbildung 4-34: Mittlere Klassifikationsrate (2-fach Kreuzvalidierung) in Abhängigkeit von γ ($\gamma=2^a$) und C ($C=2^b$) bei einer C-SVM-Texturklassifikation mit RBF-Kern. Oben: Outex-Datenbank *TC00002* (32×32 Bildpunkte). Unten: Outex-Datenbank *TC00001* (64×64 Bildpunkte). Zeile zwei und vier zeigen jeweils die auf das Maximum normierten besten 10 Prozent der Klassifikationsraten.

Abbildung 4-34 und Abbildung 4-35 zeigen die mittleren Klassifikationsraten von C-SVM für die Outex-Texturdaten *TC00002* (erste Reihe) und *TC00001* (dritte Reihe) bei Verwendung der einzelnen Texturoperatoren. Reihe zwei und vier zeigen die jeweils auf das Maximum normierten besten zehn Prozent der Klassifikationsraten, um die Konfigurationsbereiche der Klassifikatorparameter mit den besten Ergebnissen hervorzuheben.

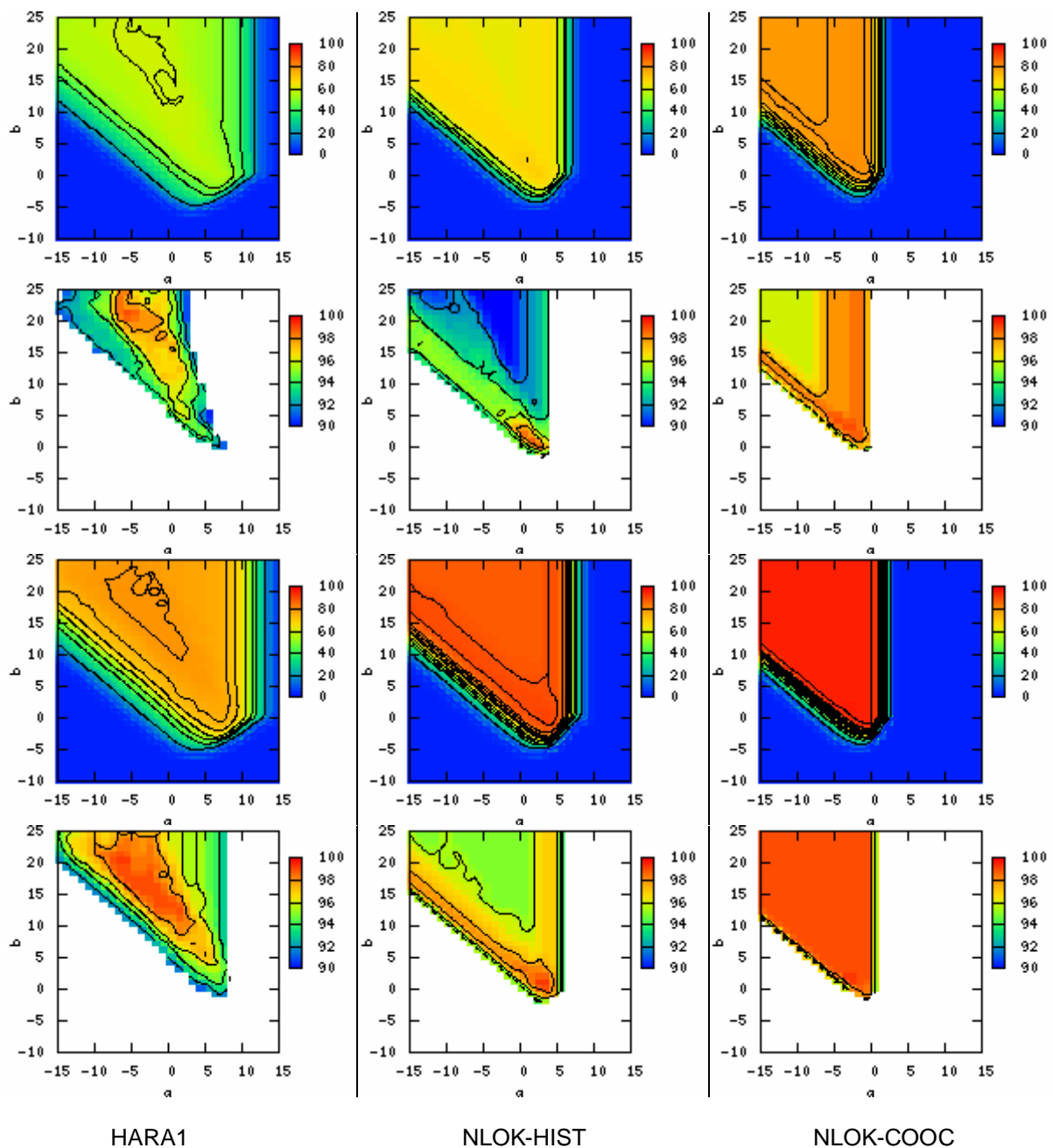


Abbildung 4-35: Mittlere Klassifikationsrate (2-fach Kreuzvalidierung) in Abhängigkeit von $\gamma (\gamma=2^a)$ und $C (C=2^b)$ bei einer C-SVM-Texturklassifikation mit RBF-Kern. Oben: Outex-Datenbank *TC00002* (32×32 Bildpunkte). Unten: Outex-Datenbank *TC00001* (64×64 Bildpunkte). Zeile zwei und vier zeigen jeweils die auf das Maximum normierten besten 10 Prozent der Klassifikationsraten.

Betrachtet man nur die besten Klassifikationsergebnisse eines Texturoperators bei verschiedenen Bildgrößen ((zweite bzw. vierte Reihe), so stellt man fest, dass optimale Bereiche für C und γ für die Klassifikation mit einer C-SVM je nach Konfiguration an

vergleichbaren Positionen liegen. Die Position der optimalen Bereiche bei Verwendung von unterschiedlichen Texturoperatoren variiert jedoch zum Teil deutlich.

Für jeden Texturoperator ist es also sinnvoll, diese Klassifikatorparameterbereiche gezielt zu bestimmen.

Tabelle 4-7 zeigt die jeweils besten erreichten Klassifikationsergebnisse bei kNN und C-SVM. Der Einsatz von C-SVM im Vergleich zu kNN für die Texturklassifikation hat in allen betrachteten Fällen zu einer teilweise erheblichen (AK, HIST) Verbesserung der Klassifikationsergebnisse geführt. Der Aufwand für die Bestimmung der optimalen Klassifikatorparameter mittels einer umfassenden Rastersuche ist jedoch deutlich größer (1116 bzw. 11 Parameterkonfigurationen).

Generell liefert die Klassifikation mit NLOK-Merkmalen (NLOK-HIST und NLOK-COOC) für beide Datensätze deutlich bessere Ergebnisse als alle anderen Texturoperatoren, ohne dass deren zusätzliche Parameter an die Klassifikationsaufgabe angepasst wurden.

Tabelle 4-7. Beste Klassifikationsergebnisse Texturklassifikation.

	LTEM	AK	HIST	HARA1	NLOK-HIST	NLOK-COOC
kNN, 32×32	39,97	35,22	21,27	52,87	64,47	72,57
C-SVM, 32×32	45,59	75,39	48,69	61,64	70,73	81,91
kNN, 64×64	71,37	66,27	28,11	73,87	89,30	93,39
C-SVM, 64×64	80,74	90,59	64,03	81,16	92,45	97,10

Da eine umfangreiche Rastersuche im Rahmen der Erstellung, Konfiguration und Optimierung eines BV-Systems unter Umständen zeitlich sehr aufwendig sein kann, wird die Anpassung der Klassifikatorparameter in die Optimierung des Gesamtsystems integriert. Für den Fall, dass ausreichend Zeit für die Optimierung zur Verfügung steht, kann jedoch optional auch auf eine umfassende Rastersuche zurückgegriffen werden.

In einer früheren Arbeit [101] wurde die Sensitivität der Klassifikatorparameter bei kNN und C-SVM bereits an 16 heterogenen Beispieltexturen der Größe 128×128 (vgl. Abbildung 4-36) im Rahmen einer Texturklassifikationsoptimierung mittels PSO untersucht. Auch in diesem Fall lies sich keine Aussage über eine optimale Anzahl von nächsten Nachbarn treffen. Interessanterweise stieg aber auch hier die Klassifikationsrate bei Verwendung von LTEM tendenziell mit einer wachsenden Anzahl von Nachbarn, wohingegen die Klassifikationsrate bei Verwendung von AK leicht abfiel. Zukünftige Untersuchungen an weiteren Texturdatensätzen müssen bestätigen, ob es sich hierbei um ein Operator-spezifisches Verhalten oder um ein in diesen Fällen zufälliges Verhalten handelt.

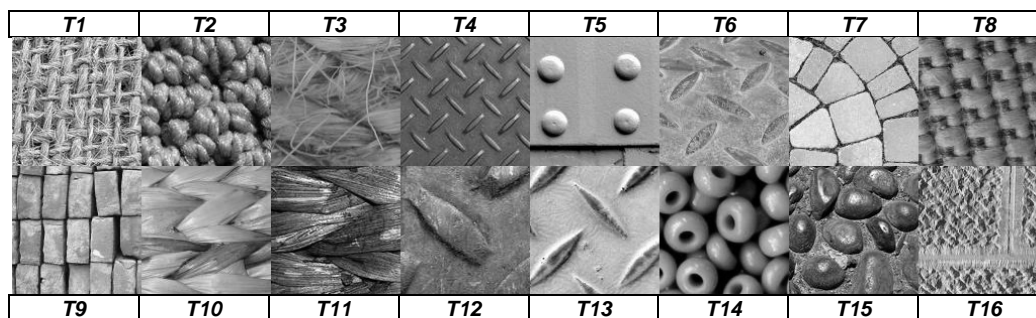


Abbildung 4-36: Sechzehn verschiedene Texturen für die Texturanalyse.

Die in dieser früheren Arbeit beispielhaft durchgeführte Untersuchung der Parametersensitivität bei C-SVM bezüglich C und γ bei einer fixen Texturparameterkonfigurationen für AK ($p=15$, $q=17$) und LTEM führte zu vergleichbaren Ergebnissen bezüglich der unterschiedlichen Position optimaler Parameterbereiche bei verschiedenen Texturoperatoren. Untersucht wurden die Klassifikationsraten bei variierenden in Abhängigkeit des Texturoperators und der betrachteten Regionengröße (32×32 bzw. 64×64 Bildpunkte). Allerdings ist die Variation der optimalen Bereiche bei dem gleichen Texturoperator, aber einer unterschiedlichen Regionengröße deutlicher. Dies kann unter Umständen an der Vielfalt der räumlichen Abmessungen der Texturlemente in den 16 Beispielt Texturen liegen.

4.8 Zusammenfassung

In diesem Kapitel wurden verschiedene Bewertungsverfahren eines Bildverarbeitungssystems vorgestellt, die Bildähnlichkeit anhand verschiedener Kriterien bestimmt und die Sensitivität der Klassifikatorparameter bei einer Texturklassifikation und einer Defekterkennung untersucht. GA und PSO wurden anhand mathematischer Benchmark-Testfunktionen miteinander verglichen.

Im folgenden Kapitel werden Methoden für die Optimierung eines BV-Systems mittels evolutionärer Algorithmen und Schwarmintelligenz vorgestellt.

5 Optimierung von Bildverarbeitungssystemen mittels Evolutionärer Algorithmen und Schwarmintelligenz

Evolutionäre Algorithmen (EA) und Schwarmintelligenz werden für eine Vielzahl von Applikationen im Umfeld von Konfigurations- und Optimierungsaufgaben verwendet. Es handelt sich hierbei um biologisch inspirierte stochastische Suchverfahren, die auf Prinzipien der Evolutionstheorie nach Darwin bzw. auf Beobachtungen der Interaktion vieler Individuen in einem Schwarm basieren.

Methoden der EA werden bereits in vielen Bereichen der Bildverarbeitung, wie beispielsweise für die Merkmalsselektion oder die Optimierung von Parametern einzelner BV-Methoden erfolgreich eingesetzt. Ein Vorteil dieser Verfahren ist, dass sich sehr heterogene Problemstellungen durch geeignete Repräsentations- und Bewertungsmethoden auf diese Suchverfahren abbilden lassen.

Schwarmintelligenz und insbesondere Partikelschwarmoptimierung (PSO) hat sich in den letzten Jahren als ein zu EA alternatives Optimierungsverfahren etabliert, das bei der Optimierung vergleichbare oder sogar bessere Ergebnisse liefert [144]. Neben EA wird der Einsatz von PSO für Aufgaben der Bildverarbeitung und insbesondere für den Entwurf und die Autokonfiguration analysiert und evaluiert. Zielsetzung ist es ein Verfahren zu finden, das im Vergleich zu EA eine geringere Anzahl an Konfigurationsmöglichkeiten aufweist und ggf. schnellere und bessere Lösungen generiert.

In diesem Kapitel wird kurz auf die verschiedenen Aufgaben der Optimierung bei dem Entwurf und der Konfiguration eines BV-Systems eingegangen und die Grundlagen von EA und Schwarmintelligenz für die in dieser Arbeit verwendeten Methoden für den Entwurf und die Konfiguration beschrieben. In einem experimentellen Teil wird die Leistungsfähigkeit von GA und PSO anhand von mathematischen Testfunktionen demonstriert.

5.1 Aufgaben der Optimierung für den Entwurf und die Konfiguration eines BV-Systems

Nach [153] lassen sich grundsätzlich drei Arten von Optimierungsaufgaben für den Entwurf und die Konfiguration eines BV-Systems unterscheiden:

- Parameteroptimierung
- Auswahlverfahren

- Erzeugung neuer Algorithmen bzw. ganzer Systeme

Bei der Parameteroptimierung wird eine feste Anzahl von Parametern an eine bestimmte Aufgabe angepasst. So wird in einem einfachen Fall zum Beispiel der Schwellwert einer Binarisierung so verändert, dass Objekt und Hintergrund bestmöglich voneinander getrennt werden.

Auswahlverfahren wählen aus einer Obermenge eine geeignete Untermenge aus, was zu besseren oder auch schnelleren Lösungen führen kann. In der Bildverarbeitung handelt es sich zum Beispiel um die Auswahl geeigneter Merkmale für die Unterscheidung zweier Objekte: Schraube und Mutter aus Eisen lassen sich eher über Form beschreibende Merkmale (wie z. B. der maximale Durchmesser oder die Anzahl der Löcher) als über den mittleren Grauwert der Objekte unterscheiden.

Bei der Erzeugung neuer Algorithmen bzw. Systeme werden problemangepasste elementare Funktionen mit Variablen bzw. Konstanten so kombiniert, dass eine vordefinierte Aufgabe möglichst gut erfüllt wird. Bei den Funktionen kann es sich um grundlegende mathematische Funktionen, wie +, - und *, Elemente einer Programmiersprache wie „if-then-else“ aber auch um Bildverarbeitungsmodule, wie z. B. Segmentierungsalgorithmen, Filteroperationen, oder Morphologie handeln.

Voraussetzung für den Einsatz von Optimierungsverfahren ist, dass man für den jeweiligen Anwendungsfall eine so genannte Zielfunktion (auch Fitnessfunktion genannt) definieren kann, die im anschließenden Optimierungsprozess minimiert oder maximiert wird. In die Zielfunktion bei der Optimierung und Konfiguration von BV-Systemen fließen in der Regel die Bewertungsmaße nach Kapitel 4 mit ein. Bei der Mustererkennung und der Fehlerdetektion verwendet man für die Zielfunktion beispielsweise die Erkennungsraten bezüglich eines festen Trainingsdatensatzes. Je mehr Muster richtig erkannt werden, desto besser sind die Parameter des Systems eingestellt (Maximierung der Zielfunktion). In komplexeren Zielfunktionen kann zusätzlich berücksichtigt werden, wie sicher ein Muster erkannt wird oder wie viel Aufwand (Zeit, Rechenoperationen, usw.) für die Erkennung von Nöten ist. Die Fitnessfunktion muss jedoch nicht unbedingt ausschließlich von den Ergebnissen eines ggf. im System integrierten Klassifikators abhängig sein. Unter Umständen kann es durchaus sinnvoll sein, bereits die Ergebnisse früherer Systemmodule für eine Bewertung heranzuziehen. So ist z. B. ein Rückschluss anhand der Güte einer Lösung über eine korrekt arbeitende Beleuchtung sicherlich einfacher, wenn mittels der Zielfunktion eine Aussage über die Grauwertdynamik des Originalbildes gemacht werden kann. Weiterhin ist bei der Wahl einer geeigneten Fitnessfunktion darauf zu achten, dass diese zum einen eine geringe Rechenleistung erfordert, andererseits jedoch sensibel genug auf kleine Veränderungen reagieren kann.

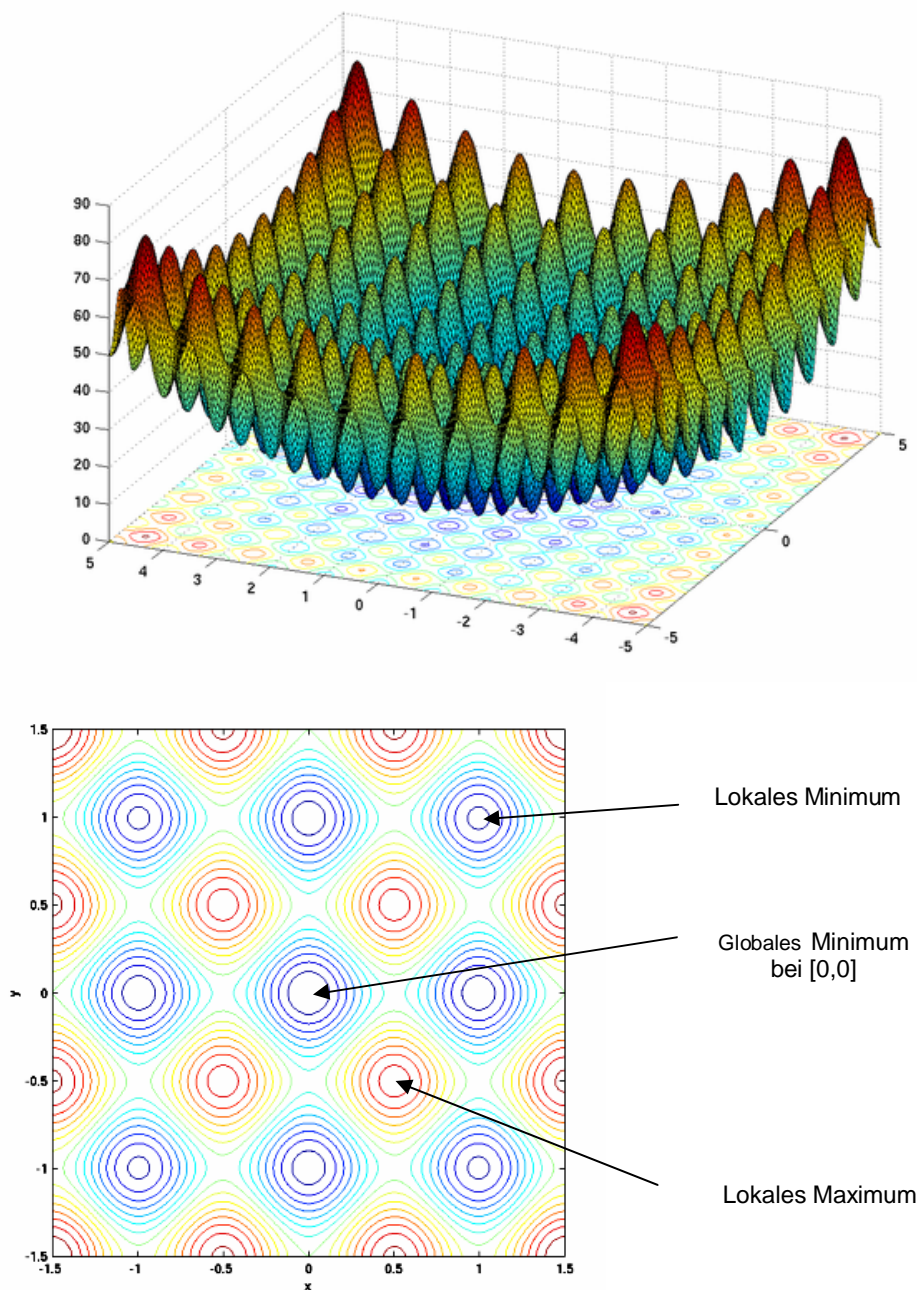


Abbildung 5-37: Rastrigrin-Funktion: $RAS(x) = nA + \sum_{i=1}^n (x_i^2 - A \cos 2\pi x_i)$. Für die Dimension $n = 2$ und die Steilheit $A=10$ hat diese Funktion ein globales Minimum bei $[0,0]$.

Ist die Zielfunktion nur von zwei Parametern abhängig, kann man sie sich als Gebirge vorstellen: ausgehend von einem Startpunkt in dem Gebirge sucht ein Optimierungsverfahren nun nach dem nächstgelegenen Tal oder Berg (lokale Optimierung) oder nach dem tiefsten Tal

bzw. höchsten Berg im gesamten Suchraum (globale Optimierung). Abbildung 5-37 zeigt beispielhaft die für den Vergleich von EA und PSO verwendete Rastrigrin-Funktion [90, 144]. Die Wahl des Startpunktes kann zufällig erfolgen oder durch Verwendung von Vorwissen beeinflusst werden und spielt unter Umständen eine entscheidende Rolle.

Eine festzulegende Nebenbedingung für die Optimierung ist die physikalische bzw. mathematisch sinnvolle Beschränkung der zu optimierenden Parameter auf einen definierten Bereich und Datentyp (Binär-, Integer- oder Fließkommawerte). In der Regel handelt es sich in der Bildverarbeitung aufgrund der diskreten Bilddaten auch um diskrete ganzzahlige Parameter: eine Binarisierungsschwelle für ein übliches 8bit-kodiertes Grauwertbild kann nur die ganzzahligen Werte von 0 bis 255 annehmen. Der Datentyp beeinflusst die Wahl eines möglichen Optimierungsverfahrens. Leider ist nicht jedes Verfahren für jeden Datentyp geeignet.

Generell ist es das Ziel der Optimierung, die „beste“ Lösung zu finden, also eine globale Suche im gesamten Suchraum. Leider gibt es für beliebige Zielfunktionen im Allgemeinen kein praktikables mathematisches Verfahren, das in endlicher Zeit mit endlichem Aufwand immer das globale Optimum findet.

Eine komplette Suche, d. h. das Abrastern jedes einzelnen Punktes im definierten Suchraum ist in der Realität rechentechnisch nicht machbar. Abhilfe bietet, wie bereits erwähnt, ggf. unter Abstrichen bei der Lösungsqualität, die Unterteilung in Teilprobleme und/oder die Verwendung von Vorwissen.

In Fällen, in denen kein mit Sicherheit zum Erfolg führendes Verfahren bekannt oder praktikabel ist, bieten Methoden, die in geeigneter Form den Zufall zur Hilfe nehmen, ein probates Mittel für die Suche nach einem Optimum. Es wird versucht, relativ gute Lösungen mit wenig Rechenaufwand unter Zuhilfenahme von „systematisch intelligentem Raten“ zu finden. In der Regel kann hier jedoch nicht garantiert werden, dass bei mehreren Optimierungsläufen (für das gleiche Problem) immer dieselbe Lösung gefunden wird und prognostiziert werden kann, wie gut diese Lösung nach vergleichbarer endlicher Zeit wirklich ist. Eine Abwägung zwischen der erreichbaren Güte der Lösung und der aufzuwendenden Rechenzeit ist nicht vorab möglich.

5.2 Evolutionäre Algorithmen

Evolutionäre Algorithmen, zu denen unter anderen die Genetischen Algorithmen (GA) und die Genetische Programmierung (GP) zählen, sind stochastische Suchverfahren, die bei der Suche eine Schar von unterschiedlichen Lösungsansätzen verwenden. Für die sukzessive Verbesserung der Lösungen werden Prinzipien der Evolution und soziale Verhaltensweisen von Gruppen von Lebewesen verwendet. Eine Einordnung von Evolutionären Algorithmen in die um

GP erweiterte Taxonomie der „Computational Intelligence“ nach [19] ist in Abbildung 5-38 zu finden.

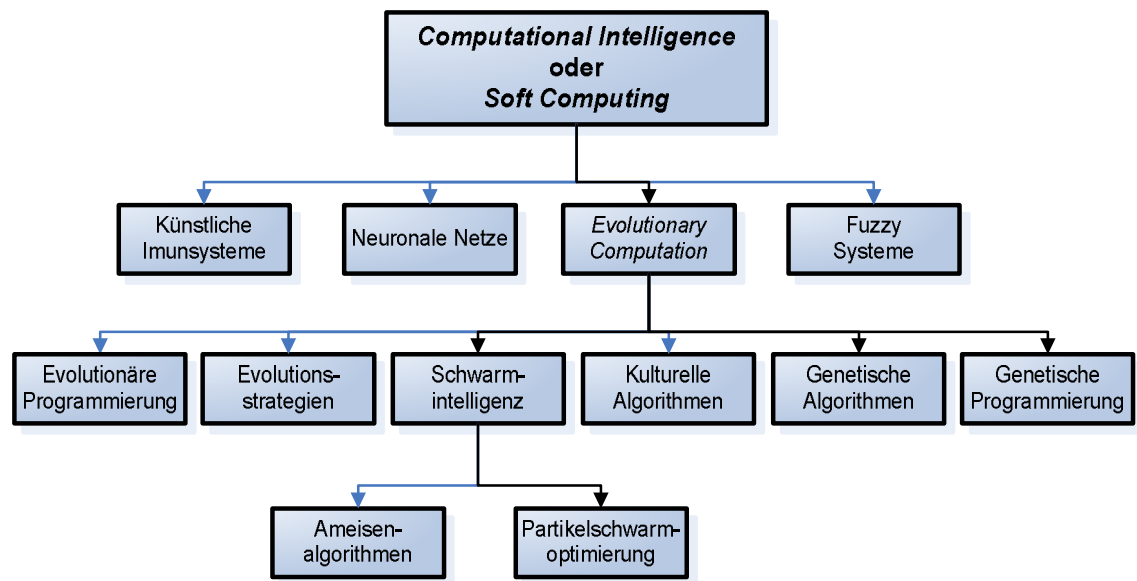


Abbildung 5-38: Taxonomie von *Computational Intelligence*. Angepasst nach [19].

5.2.1 Genetische Algorithmen

Bei GA [41] werden Prinzipien der Evolution, wie Vererbung, Selektion, Mutation und Rekombination (auch Kreuzung genannt) verwendet um Lösungsvorschläge über mehrere Generationen hinweg zu verbessern. Die zu optimierenden Parameter werden in der Regel als Kette (Chromosom) fester Länge mit den Ziffern Null und Eins kodiert. Bei der Auswahl geeigneter Merkmale für die Unterscheidung zweier Objekte hat die Kette beispielsweise für jedes Merkmal einen Eintrag (Gen). Hat das Gen den Wert eins, wird das entsprechende Merkmal verwendet, hat es den Wert null, wird es nicht verwendet. Neben der linearen Kodierung sind auch komplexere Strukturen, wie Bäume oder Graphen bei den GA möglich. Insbesondere wird auch die binäre Darstellung von ganzen Zahlen verwendet.

Den Startpunkt der Optimierung bildet eine zufällig initialisierte Population von Chromosomen. Jedes dieser Chromosome repräsentiert einen Lösungsansatz für das zugrunde liegende Problem. Nach Ausführung und Bewertung des Ansatzes werden einzelne Individuen (Chromosome) aus der Population ausgewählt. Diese Selektion basiert üblicherweise auf der Güte (Fitness) der Lösung. Bessere Lösungen werden bevorzugt bzw. öfter ausgewählt als schlechtere. Ein typisches und viel verwendetes Beispiel ist die Rouletterad-Selektion. Je besser der Fitnesswert eines Individuums ist, desto mehr Anteile eines Rouletterades erhält dieses Individuum (Abbildung 5-39a). Eine Lösung wird somit mit einer Wahrscheinlichkeit

ausgewählt, die proportional zu ihrem Gütewert ist. Die Selektion erfolgt durch wiederholtes Drehen des Rades.

Die ausgewählten Individuen erzeugen durch Rekombination Nachfolger. Hierbei werden je zwei der selektierten Chromosomen an einer zufällig bestimmten Stelle auseinander geschnitten und kreuzweise zu zwei neuen Individuen der nächsten Generation zusammengesetzt (Abbildung 5-39b).

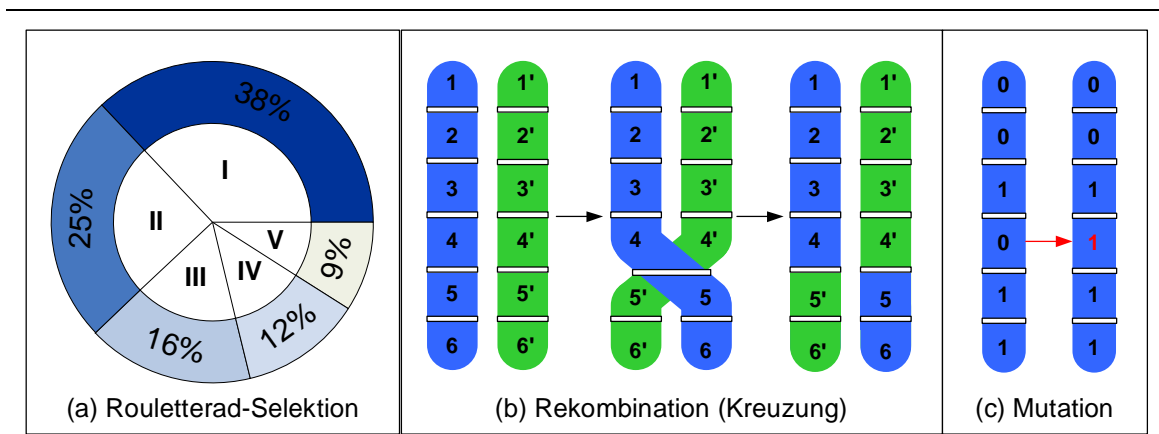


Abbildung 5-39: Rouletterad-Selektion mit fünf Individuen (a), Kreuzung (b) und Mutation (c).

Einzelne wenige Individuen der neuen Generation erfahren zusätzlich eine Mutation, d. h. die Veränderung eines Gens. Im Fall der Merkmalsauswahl wird eine Null der Ziffernkette zu einer Eins bzw. eine Eins zu einer Null (Abbildung 5-39c).

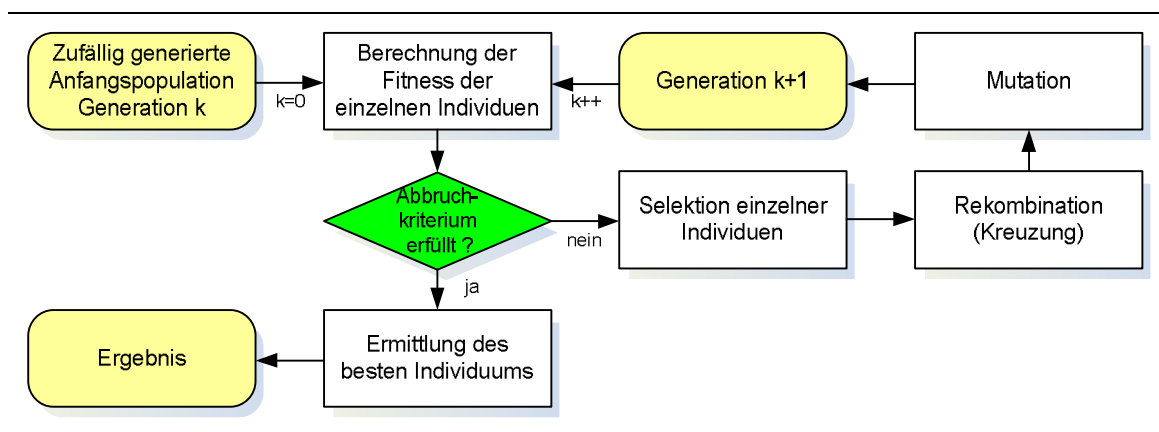


Abbildung 5-40: Ablaufdiagramm für die Optimierung mittels Genetischer Algorithmen.

Die so gebildete neue Generation wird wiederum ausgeführt und bewertet, und der Kreislauf Selektion, Rekombination und Mutation beginnt aufs Neue. Die Optimierung wird beendet, wenn ein definiertes Abbruchkriterium erfüllt wird: ein Individuum einer Generation erreicht

einen festgelegten Fitness-Wert oder eine maximale Anzahl von Generationen wird erreicht (Abbildung 5-40).

5.2.2 Genetische Programmierung

Bei der Genetischen Programmierung (GP) [70] werden komplexe Computerprogramme, Objektstrukturen oder mathematische Formeln generiert, indem problemangepasste elementare Funktionen aus einer Funktionenmenge F mit Variablen und Konstanten aus der so genannten Terminalenmenge T kombiniert werden. Für die Repräsentation der Lösung werden Baumstrukturen verwendet, wobei Elemente aus F die Knoten und Elemente aus T die Blätter eines Baumes darstellen. Für F werden oft grundlegende mathematische Funktionen, wie $+$, $-$, $*$, die Sinus- oder Kosinusfunktion oder logische Verknüpfungen verwendet.

Für die Optimierung mittels GP werden zu GA vergleichbare Methoden der Selektion, Mutation und Rekombination angewendet. Abbildung 5-41 zeigt beispielhaft zwei Lösungen von GP für eine Approximation mathematischer Funktionen. Für die Rekombination wird pro Individuum eine Kante des Baumes zertrennt und die abgetrennten Unterbäume zwischen den Individuen ausgetauscht.

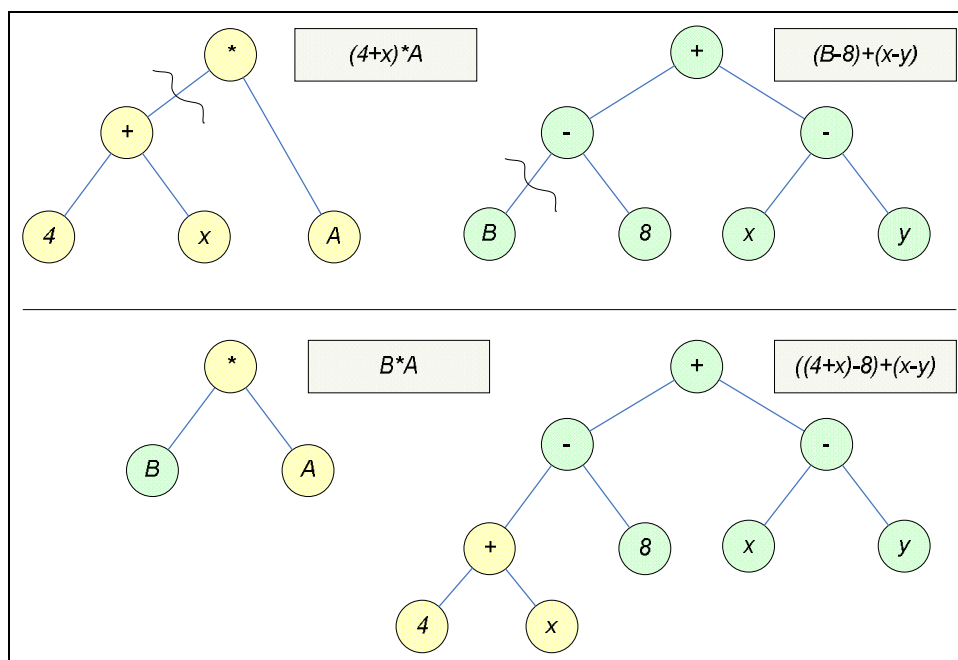


Abbildung 5-41: Eltern (obere Reihe) und durch Kreuzung an den gewellten Linien erzeugte Kinder (untere Reihe) bei der Genetischen Programmierung.

Bei der Mutation wird wahlweise ein Knoten oder ein Blatt des Baumes ausgetauscht (vgl. Abbildung 5-42).

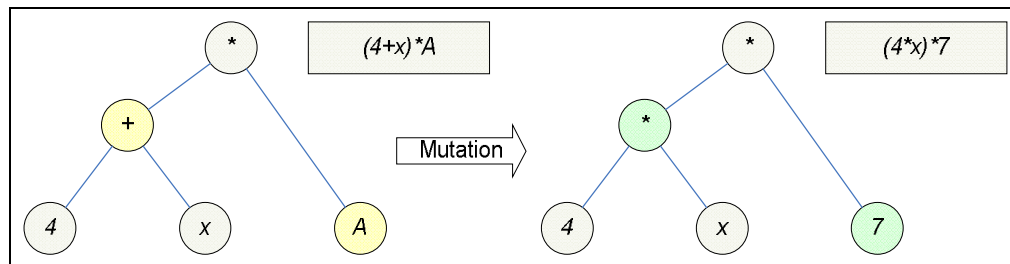


Abbildung 5-42: Mutation eines Elters bei der Genetischen Programmierung.

Wichtige Einstellungsparameter für GA und GP sind die Populationsgröße N_p und die Wahrscheinlichkeiten für die Rekombination (\rightarrow Kreuzungsrate Kr) und die Mutation (\rightarrow Mutationsrate Mu). Typischerweise sollte eine Population mindestens 20-30 Individuen umfassen, bei hochdimensionalen Problemen kann eine Population durchaus aber auch hunderte von Individuen beinhalten. Da die Rekombination der Individuen der treibende Faktor der Optimierung mit Genetischen Algorithmen ist, sollte die Kreuzungsrate mindestens 50% umfassen. Es sind auch Anwendungen bekannt, bei der die Rate bis zu 95% beträgt. Die Mutation hingegen spielt bei der klassischen GA nur eine untergeordnete Rolle. Typische Werte für die Mutationsrate in der Literatur liegen bei 0,1 bis 1%. Allerdings haben sich auch höhere Mutationsraten bei diversen Anwendungen als erfolgreich herausgestellt. Eine Variation der Parameter kann bei der Optimierung ggf. zu schnelleren und besseren Ergebnissen führen.

Besonderes Augenmerk sollte auch auf die Vielfalt (Diversifizierung) der Lösungsansätze innerhalb einer Generation gelegt werden. Sind die Lösungen zu ähnlich, kann es passieren, dass die Optimierung aufgrund mangelnder Variationen frühzeitig zu einem lokalen Optimum konvergiert und dieses nicht mehr verlassen werden kann. Dies kann insbesondere bei der Anpassung einer bereits bestehenden Lösung an veränderte Bedingungen passieren. Abhilfe bietet zum Beispiel der Ausschluss gleichartiger Lösungen: d. h. jedes Individuum einer Generation muss einzigartig sein. Auch eine komplette Reinitialisierung einiger weniger Individuen in jeder Generation kann helfen, die Vielfalt während der Optimierung zu erhalten. Komplexere GA-Verfahren arbeiten u.a. mit so genannten Inselösungen: mehrere Populationen werden parallel auf verschiedenen Inseln optimiert. Zwischen den Inseln kann jedoch auch Migration von guten oder den besten Lösungen stattfinden.

5.2.3 Evolutionsstrategien

Im Vergleich zu GA wird bei den Evolutionsstrategien (ES) keine Güte-bezogene Auswahl der Eltern für eine nachfolgende Rekombination ausgeführt. Die Eltern werden zufällig mit derselben Wahrscheinlichkeit ausgewählt. Nach Rechenberg und Schwefel [119, 120] gibt es verschiedenen Selektionsstrategien.

5.2.3.1 $(\mu + \lambda)$ - Evolutionsstrategie

Aus einer Population von μ Eltern wird λ mal ein Elter zufällig ausgewählt. λ und μ sind beliebige ganze Zahlen, wobei die Relation $\lambda \geq \mu \geq 1$ gelten muss. Aufgrund dieser Bedingung können Eltern mehrfach gezogen werden. Jeder ausgewählte Elter erzeugt eine exakte Kopie seiner selbst. Diese Kopie wird anschließend mittels eines geeigneten Mutationsoperators leicht verändert. Die μ besten Individuen von den $(\mu+\lambda)$ Eltern und Kindern bilden die neue Elternpopulation. Ein Sonderfall der $(\mu+\lambda)$ -ES ist „zweigliedrige“ $(1+1)$ -ES. Da die besten Individuen bei der Auswahl erhalten bleiben, ergibt sich in diesem Fall eine monoton steigende Gütefunktion. Den Quotienten aus μ und λ bezeichnet man als Selektionsdruck. Ist der Selektionsdruck zu groß kann es vorkommen, dass die Population schnell in ein Nebenmaxima konvergiert. Bei einem zu kleinen Selektionsdruck erhält man unter Umständen keine genügende Konvergenz der Lösungen.

5.2.3.2 (μ, λ) - Evolutionsstrategie

Im Vergleich zu der $(\mu+\lambda)$ -ES werden bei der (μ, λ) -ES die μ besten Individuen nur aus den λ Nachkommen bestimmt. Zuvor gefundene Optima können bei dieser Strategie wieder verloren gehen.

5.2.3.3 $(\mu/\rho + \lambda)$ - und $(\mu/\rho, \lambda)$ - Evolutionsstrategie

Bei diesen Strategien werden die λ Nachkommen nicht mehr durch Kopie sondern durch Rekombination von ρ Eltern erzeugt und anschließend mutiert. Für die Auswahl der μ besten Individuen gelten die gleichen Regeln wie bei der $(\mu+\lambda)$ - bzw. (μ, λ) -Evolutionsstrategie.

5.2.3.4 Kodierung, Mutation und Rekombination der Evolutionsstrategien

Bei Verwendung der Evolutionsstrategien werden die zu optimierenden Parameter in der Regel reellwertig kodiert. Im Fall der Mutation werden einzelne Parameter um einen kleinen Betrag verändert. Die Rekombination wird entweder durch eine 1-Punkt-Rekombination zwischen zwei Eltern oder durch eine Parametermittelung bei mehr als zwei Eltern realisiert.

Neben diesem Ansatz wird die Verwendung von Schwarmintelligenz, insbesondere der Partikelschwarmoptimierung für den Entwurf und die Autokonfiguration von Bildverarbeitungssystemen analysiert.

5.2.4 Partikelschwarmoptimierung

In den letzten Jahren finden weitere populationsbasiertes Optimierungsverfahren der so genannten Schwarmintelligenz vermehrte wissenschaftliche und applikative Beachtung [39, 46]. Hierzu gehört auch die Partikelschwarmoptimierung [31, 109]. Inspiriert von dem Verhalten eines Vogel- oder Fischeschwarms auf der Suche nach Nahrung erfolgt die Suche nach einem Optimum hier an Stelle von Fortpflanzung und Mutation durch eine Bewegungsdynamik in Form der kollektiven Veränderung von Position und Geschwindigkeit der Individuen (hier: Partikel). Die Position bereits gefundener lokaler Optima im Suchraum, sowohl vom Individuum selbst, als auch von anderen Individuen im Schwarm, beeinflusst die aktuelle Geschwindigkeit und damit die nächste Suchposition eines Partikels.

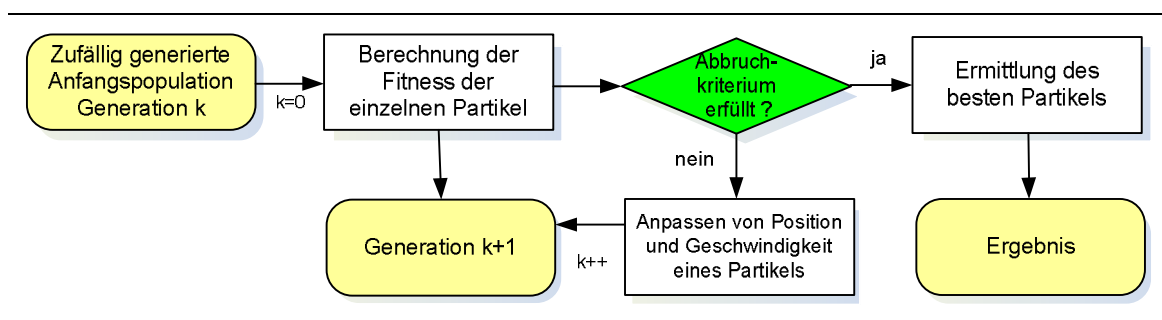


Abbildung 5-43: Schematischer Ablauf für die Optimierung mittels PSO.

Die Optimierung beinhaltet somit eine soziale und eine persönliche Komponente, die die Suche beeinflussen. Der schematische Ablauf einer Optimierung mittels PSO ist in Abbildung 5-43 dargestellt. Im Folgenden wird auf die für diese Arbeit relevanten Aspekte der PSO eingegangen. Einen sehr guten Überblick über die verschiedenen Varianten der PSO bieten beispielsweise [109, 134].

5.2.4.1 Grundalgorithmus nach Kennedy und Eberhardt (1995)

Das i -te Partikel eines Schwarms zu einem definierten Zeitpunkt t wird über seine Position $\mathbf{x}_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{in}^t]$ im n -dimensionalen Suchraum beschrieben. Eine Population besteht aus N_P solcher Partikel: $\mathbf{x}_i^t = [\mathbf{x}_1^t, \mathbf{x}_2^t, \dots, \mathbf{x}_{N_P}^t]$. Die Veränderung im Suchraum erfolgt anhand der Anpassung einer dem Partikel zugewiesenen Geschwindigkeit $\mathbf{v}_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{in}^t]$ gemäß der folgenden Formeln [31]:

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + U(0, \phi_1) \otimes (\mathbf{lb}_i^t - \mathbf{x}_i^t) + U(0, \phi_2) \otimes (\mathbf{gb}_{NP}^{t''} - \mathbf{x}_i^t) \quad (5.32)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad (5.33)$$

$U(0, \phi_1)$ bzw. $U(0, \phi_2)$ sind Vektoren mit gleich verteilten Zufallswerten im Bereich $[0, \phi]$. \mathbf{lb} ist die beste bisher gefundene Position im Suchraum des i -ten Partikels, \mathbf{gb} die beste bisher gefundene Suchposition des gesamten Schwarms. Der Operator \otimes ist eine komponentenweise Multiplikation zweier Vektoren.

Gemäß Formel (5.1) werden Individuen im Suchraum so verschoben werden, dass sie sich in Richtung der bis dahin gefundenen globalen (\mathbf{gb}) und lokalen (\mathbf{lb}) Maxima bewegen. Die Gewichtung zwischen diesen zwei Extrema erfolgt mittels zweier zufälliger Gewichte, so dass auch bis zum Zeitpunkt t unbekannte Regionen des Parameterraums ausgewertet werden können.

Der Vorteil der Partikelschwarmoptimierung besteht darin, dass er nur wenige Parameter besitzt, die festgelegt werden müssen. ϕ_1 und ϕ_2 beeinflussen das „soziale“ und das „persönliche“ Verhalten eines Partikels im Schwarm. Um die Suche zu kontrollieren wurde in den frühen Applikationen der PSO die Geschwindigkeit auf den Bereich $[-v_{max}, v_{max}]$ begrenzt. Diese Werte sind jedoch stark applikationsbezogen und beeinflussen maßgeblich das Verhalten des Schwarms bezüglich Erkundung und Konvergenz.

5.2.4.2 Variationen von PSO

Um den Einfluss des Wertebereichs der Geschwindigkeit zu begrenzen, wurde ein zusätzlicher Begrenzungsfaktor τ für die Geschwindigkeit, ein Trägheitsgewicht („Inertia weight“) eingefügt, mit dem die Bestimmung der Geschwindigkeit folgendermaßen verändert wird:

$$\mathbf{v}_i^{t+1} = \tau \mathbf{v}_i^t + U(0, \phi_1) \otimes (\mathbf{lb}_i^t - \mathbf{x}_i^t) + U(0, \phi_2) \otimes (\mathbf{gb}_{NP}^t - \mathbf{x}_i^t) \quad (5.34)$$

Je nach Applikation haben ein mit der Zeit kleiner werdendes Trägheitsgewicht zwischen 0,9 bis 0,4 [108], eine zufällige Bestimmung des Gewichts oder sogar steigende Gewichte zu einer stabileren und besseren Leistung der PSO geführt.

Bei der zweiten Adaption des Originalalgorithmus, die im Rahmen dieser Arbeit analysiert und implementiert wurde, wird mit einem Begrenzungsfaktor χ die neue Geschwindigkeit begrenzt:

$$\mathbf{v}_i^{t+1} = \chi (\mathbf{v}_i^t + U(0, \phi_1) \otimes (\mathbf{lb}_i^t - \mathbf{x}_i^t) + U(0, \phi_2) \otimes (\mathbf{gb}_{NP}^t - \mathbf{x}_i^t)) \quad (5.35)$$

mit

$$\phi = \phi_1 + \phi_2 > 4 \quad (5.36) \quad \text{und} \quad \chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \quad (5.37)$$

Mit $\phi_1 = \phi_2 = 2,05$ und daraus folgend $\chi \approx 0,7298$ und einer Begrenzung der maximalen Geschwindigkeit v_{max} auf x_{max} ergibt sich die so genannte kanonische

Partikelschwarmoptimierung, eine PSO mit nahezu keinen applikationsspezifischen Parametern. Die kanonische PSO wurde bereits in vielen Fällen erfolgreich eingesetzt [145].

Bei der Suche nach der besten Lösung der Partikel des Schwarms können zusätzlich eine unterschiedlich große Anzahl von Nachbarn und unterschiedliche Nachbarschaftstopologien betrachtet werden. In dieser Arbeit wird immer der gesamte Schwarm betrachtet, eine spezielle Topologie ist für diesen Fall also nicht notwendig.

Geeignete Populationsgrößen sind je nach Applikation 20-50 Partikel.

Um eine frühzeitige Konvergenz der Schwarms zu lokalen Optima zu vermeiden, wird in Anlehnung an das „No-Hope“ und „Re-hope“-Prinzip von [21] die Geschwindigkeit der einzelnen Partikel im Schwarm reinitialisiert, wenn die mittlere betragsmäßige Geschwindigkeit des Schwarms und die Veränderung des Schwarmsschwerpunktes unter ein gewisses Epsilon fallen.

5.2.4.3 Applikationen von PSO im Bereich der Bild- und Videoanalyse

Gemäß einer umfangreichen Studie über Applikationen von Partikelschwarmoptimierung von Poli [110] anhand der Referenzen des IEEE Xplore-Portals²³ entfallen etwa 8% der zwischen den Jahren 1995 und 2006 veröffentlichten Beiträge in den Bereich der Bild- und Videoverarbeitung. Es werden 50 Referenzen zu Anwendungen beispielsweise aus den Bereichen der Gesichtsdetektion und -erkennung, der Bildsegmentierung oder der Bildregistrierung aufgeführt. Das folgende Zitat aus der Arbeit gibt den gesamten Abschnitt bezüglich der Bild- und Videoanalyse wieder:

„3.16 Image and Video

A large proportion of the applications bibliography (7.6%) of PSO applications is devoted to image and video analysis applications. Image analysis applications include: iris recognition [67, 81], fruit quality grading [226], face detection and recognition [478, 455, 77, 454], image segmentation [601, 99, 534, 289, 626], synthetic aperture radar imaging [51, 582], locating treatment planning landmarks in orthodontic x-ray images [86], image classification [61], inversion of ocean colour reflectance measurements [469, 556], image fusion [391], traffic stop-sign detection [621], defect detection [504], image retrieval [250, 65], human detection in infrared imagery [403], image registration [523, 488, 489], pixel classification [100], detection of objects [402], pedestrian detection and tracking [449], texture synthesis [636], microwave imaging [214, 215, 154, 112, 213], scene matching [468], photo time-stamp recognition [159], contrast enhancement [277, 393], 3D recovery with structured beam matrix [483], auto cropping for digital photographs [625], character recognition [294], shape matching [123], image noise cancellation [80]. Video applications include: MPEG optimisation [24], motion estimation [122], object tracking [23], body posture tracking [443, 220], traffic incident detection [474].“

Riccardo Poli, An Analysis of Publications on Particle Swarm Optimisation Applications. Technical Report CSM-469.

²³ IEEE: Institute of Electrical and Electronics Engineers
<http://ieeexplore.ieee.org> (letzter Zugriff 30.11.2008)

5.2.5 Ameisenalgorithmus

Eine weitere Methode der Schwarmintelligenz ist der Ameisenalgorithmus (ACO) [39, 46]. Der Ameisenalgorithmus ist dem Verhalten von Ameisen nachempfunden, die bei dem Transport von Nahrung optimale Wege finden. Zu Beginn bewegen sich die Ameisen ohne Präferenz bezüglich des gewählten Weges. Bei jeder Bewegung legt eine Ameise eine Pheromonspur. Nachfolgende Ameisen wählen verschiedene Wege, orientieren sich aber bevorzugt an Wegen mit einer starken Pheromonspur. Kürzere Wege werden schneller und dadurch auch von einer höheren Anzahl von Ameisen durchquert, was zu Ausbildung von Vorzugswegen aufgrund der starken Pheromonspur führt. Diese Methode wird bei dem Ameisenalgorithmus in Form eines Pheromonvektors nachgebildet, in dem die Pheromonwerte an Weggabelungen gespeichert werden. Der Pheromonwert jedes Weges entspricht der Wahrscheinlichkeit mit der dieser Weg ausgewählt wird. Der Pheromonvektor wird nach jeder Iteration aktualisiert, d. h. nachdem eine oder mehrere Ameisen einen Weg gewählt haben. Ein mögliches Gütekriterium für diese Art der Algorithmen ist zum beispielsweise die Länge des gewählten Weges.

Der Ameisenalgorithmus wird sehr effizient vor allem zur Lösung von Kürzesten-Wege-Problemen, wie beispielsweise dem Problem des Handlungsreisenden (*Traveling-Salesman-Problem*, TSP) und anderen Problemen der „Operation-Research“ eingesetzt. Für die Struktur- und Parameteroptimierung ist er eher ungeeignet.

5.3 Aspekte der verwendeten Architektur bezüglich der Optimierung

Die folgenden Aspekte gelten sowohl für die Optimierung mittels GA/GP wie auch für die Optimierung mittels PSO.

5.3.1 Initialisierung einer Population

Bei der Initialisierung einer Population werden nur überlebensfähige Individuen zugelassen. Die Erzeugung eines Individuums wird solange wiederholt, bis die gewünschte Populationsgröße N_p erreicht worden ist. Die verfügbaren Methoden können frei aus einer Liste ausgewählt und kombiniert werden oder es wird eine feste Ablaufstruktur für die Optimierung vorgegeben.

5.3.2 Elitismus

Um das Überleben guter Individuen zu gewährleisten, werden die N_E besten Individuen ohne Veränderung, d. h. Mutation in die nächste Generation übernommen. Diese Individuen können trotzdem bei der Rekombination und dem Klonen verwendet werden. Die Zahl der elitären Individuen liegt typischerweise im Bereich von null bis drei.

5.3.3 Diversität

Um eine frühzeitige Konvergenz bzw. die Dominanz einzelner Individuen zu mindern, können in jede neue Generation vollständig reinitialisierte Individuen eingefügt werden. Damit wird die Vielfalt der Lösungen innerhalb einer Generation erhöht. Typische Werte für die Anzahl N_D neuer Individuen liegen auch hier im Bereich von null bis drei.

5.3.4 Bewertungsfunktionen und Bestrafungselemente

Auch die Wahl der Bewertungsfunktion beispielsweise anhand eines speziellen Klassifikators kann manuell ausgewählt werden. Im Zusammenhang mit dem Klassifikator kann zusätzlich die Behandlung der verwendeten Datensätze festgelegt werden. Insbesondere gilt dies für eine Unterteilung der Daten in Trainings-, Test- und Generalisierungsdatensatz oder um die Anwendung einer n-fachen Kreuzvalidierung.

In Zusammenhang mit der Bewertungsfunktion können variable Bestrafungs- oder Belohnungselemente festgelegt werden. Insbesondere kann hierbei die Anzahl der verwendeten Methoden oder die Verarbeitungszeit nach bedarf positiv oder negativ bewertet werden. Die Konfiguration erfolgt nach dem Prinzip: Schlüsselwort, Bedingungsart, Schwellwert, Art der Bewertung und Größe des Bewertungswertes. Alternativ können den einzelnen Methoden bei deren Verwendung feste Bestrafungs- oder Belohnungswerte zugeordnet werden.

5.4 Optimierung mit GA und GP

Im Folgenden werden relevante Aspekte für den in dieser Arbeit verfolgten Entwurf und die Autokonfiguration von Bildverarbeitungssystemen von GA, GP und ES kurz aufgezeigt. Bei dem Entwurf werden die Struktur und Parameter einer BV-Kette, bei der Autokonfiguration die Algorithmenparameter einer festen Struktur optimiert. Die hier angesprochenen Variationsmöglichkeiten können mittels Initialisierungsdateien frei für den Systementwurf oder die Autokonfiguration angepasst werden.

5.4.1 Selektion

Bei der Selektion von geeigneten Lösungen für eine folgende Rekombination oder dem Klonen von bestehenden Lösungen können sowohl güteabhängige oder -unabhängige Ansätze verfolgt werden. Bei einem von der Güte einer Lösung unabhängigen Ansatz werden die Eltern zufällig ausgewählt.

Alternativ dazu erfolgt eine Selektion in Abhängigkeit von der Güte der Lösungen als Turnier- oder Wettkampfselektion oder Rouletterad-Selektion.

- Rouletterad-Selektion: Je besser der Fitnesswert eines Individuums ist, desto mehr Anteile eines Rouletterades erhält dieses Individuum. Die Auswahl der Eltern erfolgt durch wiederholtes Drehen des Rades.
- Bei der Turnierselektion treten zwei oder mehrere potenzielle Eltern gegeneinander an. Das Individuum mit dem größten Fitnesswert wird für die Rekombination oder das Klonen ausgewählt.

5.4.2 Rekombination

Aufgrund der linearen Beschreibung potenzieller Lösungen für das BV-System wird für die Rekombination wahlweise Ein-Punkt- oder Zwei-Punkte-Kreuzung verwendet. Für die Optimierung der Struktur und der Parameter eines Systems liegen die Schnittpunkte hierfür zwischen zwei Systemmethoden. Bei der Ein-Punkt-Kreuzung wird wahlweise der gleiche Schnittpunkt für beide Elternteile oder unterschiedliche Schnittpunkte verwendet. Im ersten Fall bleibt die Gesamtzahl der Methoden erhalten, im zweiten Fall kann die maximale Anzahl an Methoden variieren.

Wird ein System mit einer festen Ablaufkette, d. h. auch mit einem festen Parametersatz optimiert, kann der Schnittpunkt für die Rekombination auch zwischen zwei Parametern einer Methode liegen. Die Rekombination wird ggf. wiederholt, wenn keine überlebensfähige, d. h. in der Struktur oder Abfolge der einzelnen Methoden richtige Lösung generiert wird.

5.4.3 Mutation

Entsprechend dem GENIE-System [50, 100] werden in dieser Arbeit drei verschiedene Mutationsoperatoren verwendet:

- Parametermutation (PM): Die Parameter einer Methode werden mutiert. Dies kann entweder durch eine kleine Änderung oder durch eine Reinitialisierung des Wertes erfolgen.
- Methodenmutation (MM): Eine Methode wird komplett gegen eine andere ausgetauscht.
- Eingangs- und Ausgangsmutation (EAM): Die Verbindung zwischen zwei Methoden wird verändert.

Die Mutation ist neben der Rekombination in diesem Ansatz ein treibender Faktor für die Veränderung des Entwurfs und der Konfiguration eines BV-Systems. Aus diesem Grund sollten die Mutationsraten hier deutlich höher als bei der klassischen GA gewählt werden. Auch in diesem Fall werden bei der Mutation nur sinnvolle Lösungen zugelassen. Typische

Mutationsraten sind Werte bis zu 0,5. Das Verhältnis von PM zu MM zu EAM kann ebenfalls eingestellt werden. Ein sinnvolles Verhältnisse ist nach [100] 2:1:1.

5.5 Optimierung mittels PSO

Für die Optimierung mittels PSO fällt die Anzahl der Variationsmöglichkeiten deutlich geringer aus. Nach Abschnitt 5.2.4.2 kann zwischen einem festen oder zeitlich veränderlichen Trägheitsgewicht, der kanonischen PSO oder einer freien Anpassung von ϕ_1 und ϕ_2 gewählt werden. Zusätzlich kann die Geschwindigkeit der Partikel ggf. nach vorgegebenen Regeln reinitialisiert werden, um eine frühzeitige Konvergenz der Lösungen zu vermeiden.

5.6 Experimente

Um die prinzipielle Leistungsfähigkeit von GA und PSO zu demonstrieren, werden beide Verfahren in diesem Abschnitt anhand von mathematischen Testfunktionen getestet. Die Optimierung wird eingesetzt, um das globale Minimum dieser Funktionen zu finden. Bei den verwendeten Testfunktionen handelt es sich um unimodale, d. h. Funktionen mit einem globalen Extrempunkt oder multimodale D -dimensionale Testfunktionen. Diese Funktionen werden oft zu Vergleichszwecken im Bereich der Evolutionären Algorithmen verwendet werden.

Die Sphere-Funktion ist eine unimodale Testfunktion mit einem globalen Minimum f^* bei $\mathbf{x} = \mathbf{0}$. Diese Funktion hat keine weiteren Nebenminima.

$$f(\mathbf{x}) = SPH(\mathbf{x}) = \sum_{i=1}^n (x_i^2) \quad (5.38)$$

Die multimodale Rastrigrin-Funktion (vgl. Abbildung 5-37) besitzt für den zweidimensionalen Fall ebenfalls ein Minimum im Punkt (0, 0).

$$f(\mathbf{x}) = RAS(\mathbf{x}) = nA + \sum_{i=1}^n (x_i^2 - A \cos 2\pi x_i) \quad (5.39)$$

Als dritte Testfunktion dient in diesem Rahmen die Griewank-Funktion.

$$f(\mathbf{x}) = GRI(\mathbf{x}) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \left(\frac{\cos x_i}{\sqrt{i}} \right) + 1 \quad (5.40)$$

Bei den Optimierungsmethoden handelt es sich um Standard-GA mit Turnierselektion und verschiedenen Kreuzungsraten, die kanonische PSO (C-PSO) ohne Konfigurationsparameter, eine Variante der PSO mit einem festen Trägheitsgewicht (11-PSO). Zusätzlich wurde die Geschwindigkeit der einzelnen Partikel im Schwarm bei der kanonischen PSO reinitialisiert, wenn die mittlere betragsmäßige Geschwindigkeit des Schwarms und die Veränderung des Schwarmsschwerpunktes unter ein gewisses Epsilon fallen (CV-PSO). Jede Optimierung wurde

25-mal wiederholt. Gemäß Tabelle 5-8 wurden folgende Parameter für die Optimierung verwendet:

Tabelle 5-8. Parameter für die Optimierung

Allgemeine Parameter		
	Größe der Population	20, 40, 60
	Anzahl der Durchläufe	25
	Maximale Anzahl von Generationen	5000
	Dimension D der Testfunktionen	30
GA		
	Turnier-Selektion, Anzahl der Konkurrenten	3
	1-Punkt-Kreuzung, Kreuzungsraten	70%, 90%
	Mutationsrate	1%
PSO		

C-PSO	Trägheitsfaktor	1,1
11-PSO	Epsilon Geschwindigkeit	0,001
CV-PSO	Epsilon Schwerpunkt	0,001

5.6.1 Ergebnisse und Diskussion

Die Ergebnisse sind in Abbildung 5-44 bis 5-46 dargestellt. Aufgetragen ist jeweils der logarithmische mittlere Abstand über alle Optimierungsläufe des Funktionswertes f zum globalen Minimum f^* .

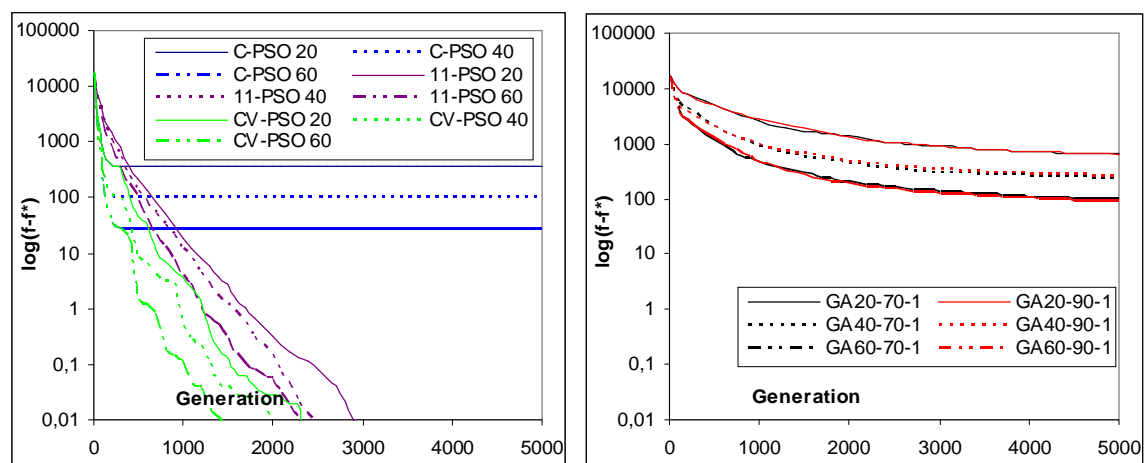


Abbildung 5-44: Mittlerer Abstand zum globalen Minimum f^* bei der Sphere-Funktion ($D = 30$). Die Größe einer Generation umfasst 20, 40 oder 60 Individuen/Partikel. Links: Ergebnisse bei Verwendung verschiedener PSO-Varianten. Rechts: GA mit Ein-Punkt-Kreuzung mit einer Kreuzungsraten von 0,7 bzw. 0,9 und einer Mutationsrate von 0,01.

Für die unimodale Sphere-Funktion und die multimodale Griewank-Funktion erreicht jede Variante der PSO bessere Ergebnisse nach 5000 Generationen. Auffällig ist hierbei, dass PSO in den ersten Generationen deutlich schneller bessere Ergebnisse liefert. Die kanonische PSO konvergiert in relativ kurzer Zeit in ein lokales Minimum oder einen Sattelpunkt der Funktion.

Die mittlere Geschwindigkeit des Schwarms nimmt in diesen Fällen rapide ab und der Schwarmsschwerpunkt bleibt konstant. Es tritt keine weitere Verbesserung des Funktionswertes im Laufe der Optimierung ein. Bei einer Reinitialisierung der Geschwindigkeit (CV-PSO) oder der Verwendung eines Trägheitsfaktors (11-PSO) werden deutlich bessere Ergebnisse erzielt. Eine frühzeitige Konvergenz tritt nicht ein.

In Tabelle 5-9 sind der Mittelwert und die Streuung von $f - f^*$ der 25 Durchläufe für die Optimierung mittels C-PSO, CV-PSO und GA für 60 Individuen nach 5000 Generationen für die Sphere-Funktion angegeben. Zusätzlich wird aufgeführt, in wie vielen Durchläufen drei funktionsspezifische Schranken S_1 , S_2 und S_3 (hier 130, 65 und 4), unterschritten wurden und wie viele Generationen dies im Mittel erforderte. Für C-PSO wurde die zweite Schranke beispielsweise in 22 von 25 Optimierungsdurchläufen nach im Mittel 133 Generationen erreicht, bei GA hingegen wurde die Schranke in nur 15 Durchläufen nach im Mittel 1303 Generationen erreicht.

Tabelle 5-9. Mittelwert und Streuung von $f-f^*$ (Sphere-Funktion)

Erfolgreiche Unterschreitung der Schranken S_1 , S_2 und S_3 in m Optimierungsläufen ($m \leq 25$) nach im Mittel n Generationen.

	$f-f^*$	$S_1=130$	$S_2=65$	$S_3=4$
GA (60 Individuen) <i>Kr=0,9, Mu=0,1</i>	87,83±87,53	19 / 2099	15 / 1303	0 / ---
C-PSO (60 Partikel)	27,90±33,81	24 / 101	22 / 133	5 / 220
CV-PSO (60 Partikel)	$1 \cdot 10^{-8} \pm 1 \cdot 10^{-8}$	25 / 115	25 / 116	25 / 393

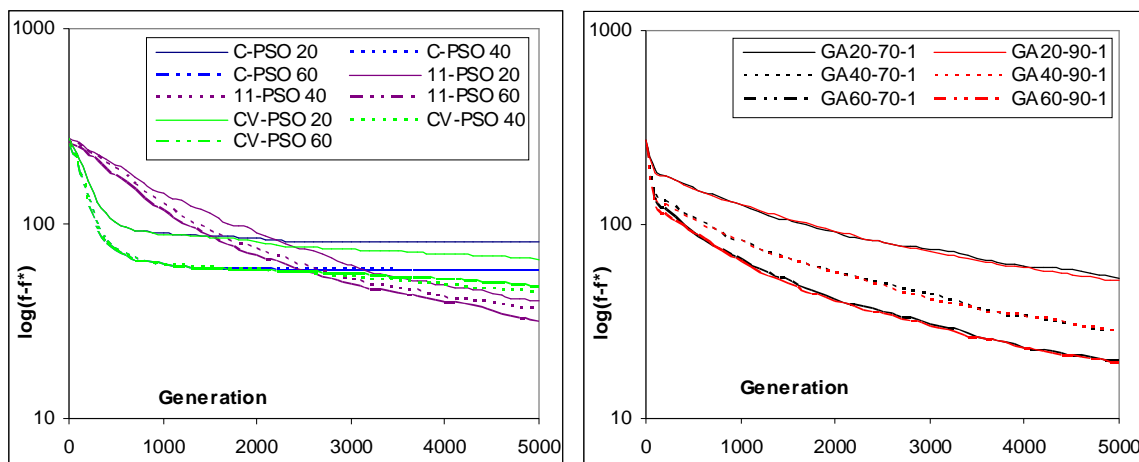


Abbildung 5-45: Mittlerer Abstand zum globalen Minimum bei der Rastrigrin-Funktion ($D = 30$). Die Größe einer Generation umfasst 20, 40 oder 60 Individuen/Partikel. Links: Ergebnisse bei Verwendung verschiedener PSO-Varianten. Rechts: GA mit 1-Punkt-Kreuzung mit einer Kreuzungsrate von 0,7 bzw. 0,9 und einer Mutationsrate von 0,01.

Bei der Suche nach dem Minimum der multimodalen Rastrigrin-Funktion liefert GA bessere Ergebnisse nach 5000 Generationen als alle PSO-Varianten. Allerdings ist auch in diesem Fall die Verbesserung bei C-PSO am Anfang der Optimierung vergleichbar mit GA. Für alle Verfahren nimmt die Leistung mit wachsender Populationsgröße zu. Die Optimierung liefert schneller bessere Ergebnisse.

Tabelle 5-10. Mittelwert und Streuung von $f-f^*$ (Rastrigrin-Funktion)

Erfolgreiche Unterschreitung der Schranken S_1 , S_2 und S_3 in m Optimierungsläufen ($m \leq 25$) nach im Mittel n Generationen.

	$f-f^*$	$S_1=110$	$S_2=55$	$S_3=3$
GA (60 Individuen) $Kr=0,9, Mu=0,1$	17,68±3,84	25 / 146	25 / 972	0 / ---
C-PSO (60 Partikel)	57,61±12,03	25 / 233	11 / 1082	0 / ---
CV-PSO (60 Partikel)	47,45±16,14	25 / 233	18 / 1925	0 / ---

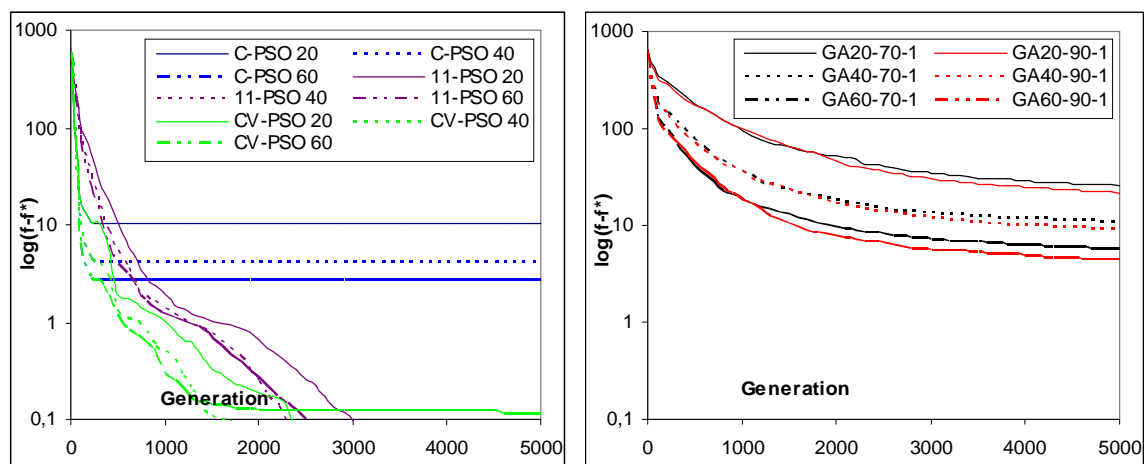


Abbildung 5-46: Mittlerer Abstand zum globalen Minimum bei der Griewank-Funktion ($n = 30$). Die Größe einer Generation umfasst 20, 40 oder 60 Individuen/Partikel. Links: Ergebnisse bei Verwendung verschiedener PSO-Varianten. Rechts: GA mit 1-Punkt-Kreuzung mit einer Kreuzungsrate von 0,7 bzw. 0,9 und einer Mutationsrate von 0,01.

Tabelle 5-11. Mittelwert und Streuung von $f-f^*$ (Griewank-Funktion)

Erfolgreiche Unterschreitung der Schranken S_1 , S_2 und S_3 in m Optimierungsläufen ($m \leq 25$) nach im Mittel n Generationen.

	$f-f^*$	$S_1=110$	$S_2=55$	$S_3=3$
GA (60 Individuen) $Kr=0,9, Mu=0,1$	4,39±3,52	25 / 130	25 / 413	25 / 3375
C-PSO (60 Partikel)	2,69±3,07	25 / 26	25 / 83	20 / 124
CV-PSO (60 Partikel)	0,11±0,12	25 / 26	25 / 38	25 / 202

5.7 Zusammenfassung

In diesem Kapitel wurden verschiedene Optimierungsverfahren der EA vorgestellt und GA und PSO anhand mathematischer Benchmark-Testfunktionen miteinander verglichen. Die kanonische PSO war der Standard-GA in zwei von drei Testfällen überlegen, ohne dass zusätzliche Einstellungen getroffen werden mussten. Anhand der hier gewonnenen Erkenntnisse des Vergleichs von PSO und GA bietet sich die kanonische PSO gerade für den Entwurf und die Autokonfiguration von Ein-Klick-Lösungen für die Bildverarbeitung an, da ein potenzieller Anwender keine zusätzliche Konfiguration des Optimierungsverfahrens mehr treffen muss. Ein Nachteil von PSO ist allerdings, dass bei Standard-PSO nur eine feste Anzahl von Merkmalen optimiert werden kann. PSO kann nicht wie GA oder GP dynamisch wachsen.

Im folgenden Kapitel wird das Konzept und die Architektur einer Umgebung für den automatischen Entwurf von BV-Systemen und eine ad hoc Adaption von PSO für den Entwurf, d. h. eine Strukturoptimierung, vorgestellt.

6 Konzept und Architektur einer Umgebung für den automatisierten Entwurf und die Konfiguration eines Bildverarbeitungssystems

In der Bildverarbeitung werden zumeist komplexe (algorithmische) Systeme mit einer Vielzahl von Parametern verwendet, um spezielle Aufgaben wie Oberflächeninspektion oder Mustererkennung durchzuführen. In der Regel werden diese Systeme von geschulten Ingenieuren manuell erstellt, entweder durch Zusammenfügen bestehender Softwaremodule und/oder durch das Entwickeln neuer Algorithmen. Dieser Prozess ist oftmals arbeitszeit- und damit auch kostenintensiv. Der Einsatz von Optimierungsverfahren kann helfen, bestehende Systeme bzw. einzelne Module der Systeme automatisch zu verbessern (Konfiguration) oder gar komplett neue Systeme zu entwerfen. Für den automatischen Entwurf und die Konfiguration des Bildverarbeitungssystems wird die zunächst die lineare Abfolge von Verarbeitungsschritten einer konventionellen BV-Kette betrachtet (Abbildung 6-47).

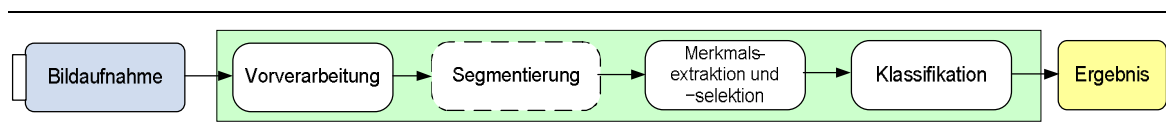


Abbildung 6-47: Konventionelles lineares Verarbeitungsschema eines Bildverarbeitungssystems.

Aufgrund der Komplexität von BV-Systemen ist es sehr schwierig, diese als Ganzes zu optimieren. In der Realität wird meist auf eine Reihe von lokalen Teilloptimierungen („divide-and-conquer“) zurückgegriffen (vgl. Abbildung 6-48). Die Teilloptimierung der einzelnen Module eines BV-Systems kann prinzipiell in beliebiger Reihenfolge geschehen. Sinnvollerweise sollte jedoch eine aufeinander aufbauende Optimierung („Bottom Up“ von der Bildaufnahme bis hin zur Klassifikation verwendet werden. Als Bewertungsmaß für die Vorverarbeitung, die Merkmalsextraktion und -Selektion und die Klassifikation können die in Kapitel 4 aufgeführten Gütemaße verwendet werden. Eine Gesamtoptimierung des Systems kann ebenfalls Lösungen anbieten. Dies erfordert bei komplexen Systemen jedoch einige Rechenzeit für die Optimierung.

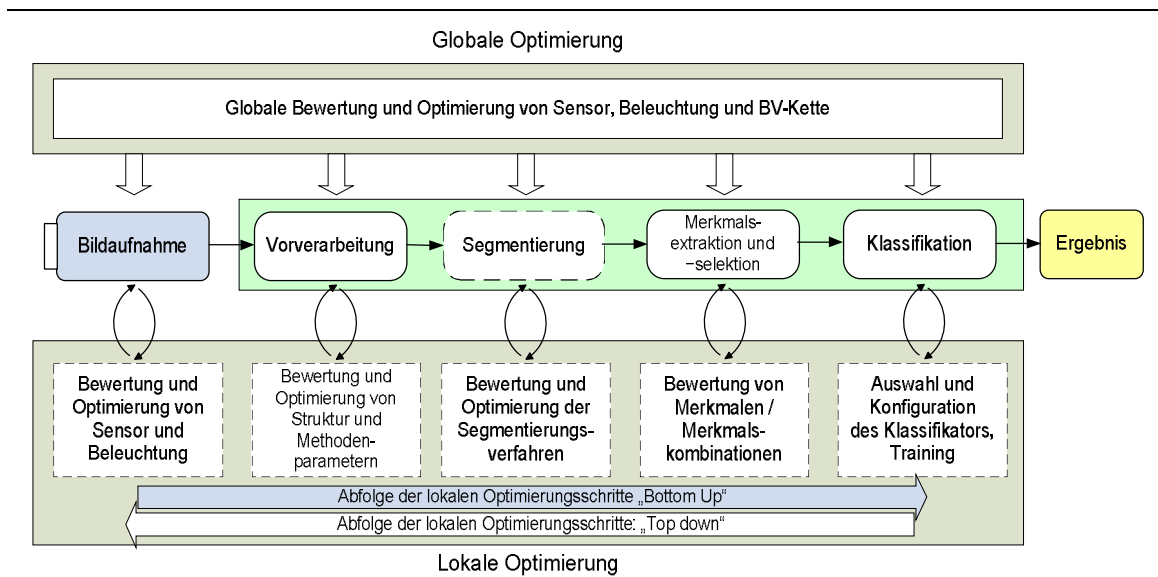


Abbildung 6-48: Lineares Verarbeitungsschema mit lokaler und globaler Optimierung.

6.1 Verwendung von Vorwissen für die Optimierung

Gerade in der Bildverarbeitung spielt das Einbringen von Vorwissen (vgl. Abbildung 6-49) eine große Rolle. Je mehr Vorwissen in die Optimierung eingebracht wird, desto schneller lässt sich ein Problem ggf. lösen oder anpassen. Vorwissen muss dabei nicht unbedingt ein konkreter Wert eines zu optimierenden Parameters sein, sondern kann sich auch auf ein applikationsspezifisches Intervall (kleiner als das Definitionsintervall) pro Parameter beziehen. Daraus folgt eine zusätzliche Einengung der Suche auf einen ggf. besser beherrschbaren Unterraum. Des Weiteren gibt es in der Bildverarbeitung bereits viele erprobte, anwendungsspezifische Abläufe von der Datenerfassung über Vorverarbeitung, Segmentierung, Merkmalsgewinnung, ggf. Datenreduktion, bis hin zur Klassifikation. Auch Wissen über vergleichbare Lösungen oder Teillösungen zu ähnlichen Problemen ist als Vorwissen anzusehen und kann bei einer Optimierung verwendet werden. Im Allgemeinen möchte man ein BV-System nicht komplett neu entwickeln, sondern, unter anderem aus Kostengründen, falls möglich nur in einigen wenigen Bereichen anpassen.

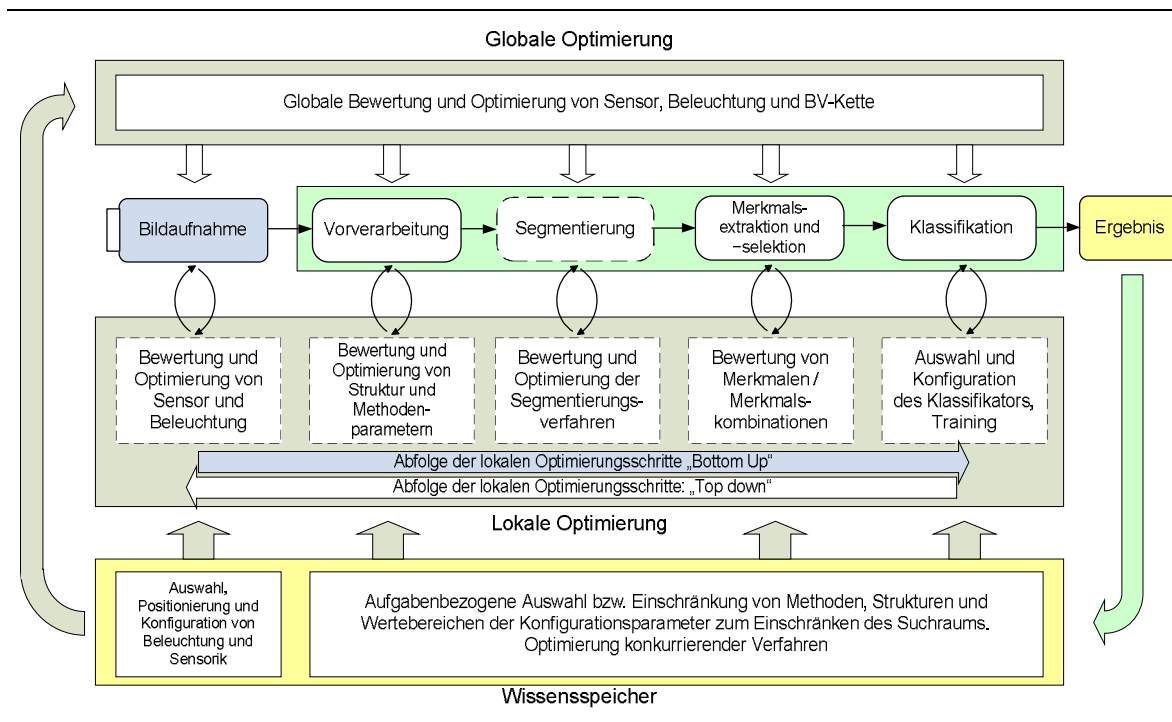


Abbildung 6-49: Einfluss von Vorwissen in die lokale und globale Optimierung.

6.1.1 Sammeln von Vorwissen

Ein entscheidender Punkt in diesem Zusammenhang ist das Sammeln und Kategorisieren von Vorwissen in einen gemeinsamen Wissensspeicher. Im Prinzip müsste jede manuell oder automatisch erstellte erfolgreiche Applikation bezüglich der produktionstechnischen Randbedingungen und daraus entwickelten BV-System analysiert und die relevanten Systemkomponenten, der Programmablauf, Beispielbilder und Ergebnisse in einer Datenbank gesichert werden. Hilfreich für die Datenerfassung kann die Verwendung von Fragebögen sein, wie sie zum Beispiel zur Erstellung einer Analyse des Anwenderprofils für das Programm QuickCog verwendet wird. Hier werden unter anderem das Einsatzgebiet des BV-Systems, die produktionstechnischen Rahmenbedingungen und die Eigenschaften des zu prüfenden Materials bzw. der zu erwarteten Fehler erfasst. Anhand dieser Daten lassen sich beispielsweise Rückschlüsse für Auswahl und Positionierung von Sensorik und Beleuchtung treffen.

6.1.2 Einbringen von Vorwissen in die Optimierung

6.1.2.1 Verfahrensauswahl

Eine Möglichkeit für das Einbringen von Vorwissen ist eine gezielte Auswahl von Verfahren, die sich bei vergleichbaren Problemstellungen bewährt haben. Im Rahmen der Optimierung können

dann einzelne Verfahren parametrisiert oder konkurrierende Verfahren miteinander verglichen werden. Dies kann den Suchraum entscheidend einschränken und somit den Entwurf und die Optimierung deutlich beschleunigen. Tatari [143] hat in seiner Arbeit über die Erkennung von Texturfehlern beispielsweise Fehler und BV-Verfahren in verschiedene Klassen eingeteilt und untersucht, welche Verfahren sich für welche Fehlertypen besonders eignen.

6.1.2.2 Bestrafungsterme für die Mehrzieloptimierung

Durch den gezielten Einsatz von Bestrafungstermen in die bei der Optimierung verwendete Fitnessfunktion lässt sich eine Mehrzieloptimierung erreichen. In Zusammenhang mit industriellen Inspektionssystemen soll hier die Rechenzeit aufgeführt werden. Wird ein zu langsame Methode oder Methodenkombination bei der Optimierung ausgewählt, wird der Gütewert dieser Lösung prozentual oder absolut reduziert.

6.1.2.3 Manuelle Interaktion

Eine manuelle Interaktion während der Optimierung kann die Leistungsfähigkeit der entwickelten Lösungen steigern, da hier explizit Anwenderwissen in die Optimierung einfließt. Einem Anwender kann zum Beispiel erlaubt werden, relevante Bildregionen interaktiv zu markieren oder für eine Defekterkennung auszuschließen. Dies reduziert zum einen den Rechenaufwand und kann zusätzlich die Erkennungsraten erhöhen.

6.1.2.4 Analyse von Bildmaterial mit und ohne Fehler

Bei der industriellen Oberflächeninspektion ist es oftmals so, dass einige Fehlertypen nur sehr selten in der Produktion auftauchen. Unter diesen Umständen kann sehr schwierig sein, ausreichend Datenmaterial für eine überwachte Klassifikation sammeln zu können. Eine Möglichkeit besteht nun darin, nur gute Objekte, d. h. ohne Fehler zu betrachten. Die Ergebnisse der Analyse gehen als Vorwissen in die Optimierung ein, in der beispielsweise Schwellwertklassifikatoren für die Erkennung von Anomalien verwendet werden.

Des Weiteren kann eine Analyse des Hintergrundes dazu verwendet werden, um ortsabhängige Strukturen innerhalb der Bilder zu finden. Dies wird beispielsweise in Form einer Shading-Korrektur dazu verwendet um Inhomogenitäten der Beleuchtung auszugleichen. Prinzipiell können eine getrennte Analyse von Vorder- und Hintergrund und das Einbinden dieser Analyseergebnisse in die BV-Lösung (vgl. Abbildung 6-50) die Ergebnisqualität positiv beeinflussen.

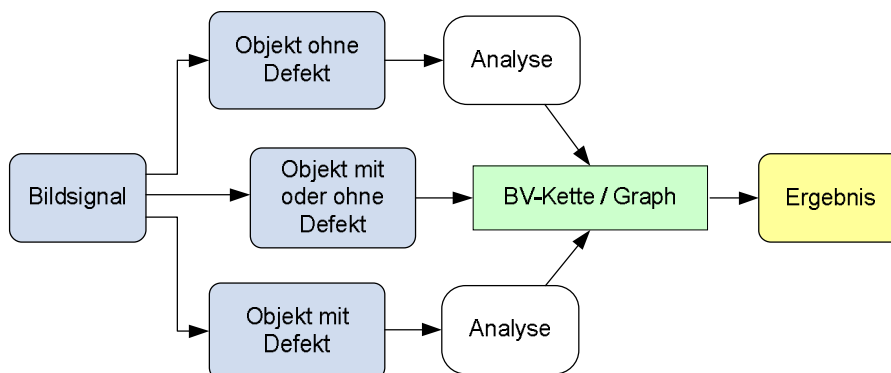


Abbildung 6-50: Eine getrennte Analyse von Objekten mit und ohne Defekte fließt in das BV-System mit ein.

6.2 Nichtlinearer Verarbeitungsablauf

Alternativ zu der dargestellten traditionellen linearen Verarbeitungskette können komplexere Strukturen verwendet werden. Neben einer graphenähnlichen Zusammensetzung von Methoden der Bildvorverarbeitung, können die Ergebnisse von Merkmalsextraktion, –selektion und der Klassifikation im Rahmen einer umfassenden Vorverarbeitung verwendet werden. Merkmale können für jeden Bildpunkt berechnet werden und ergeben so Zwischenbilder, die mittels ikonischer Bildverarbeitungsmethoden, d. h. Bild-zu-Bild Methoden, weiter verarbeitet werden können. Auch das Ergebnis einer Klassifikation lässt sich ggf. als Bild darstellen und kann somit in die Vorverarbeitung mit einfließen. So lassen sich dann auch hierarchische Klassifikationsstrukturen automatisch realisieren.

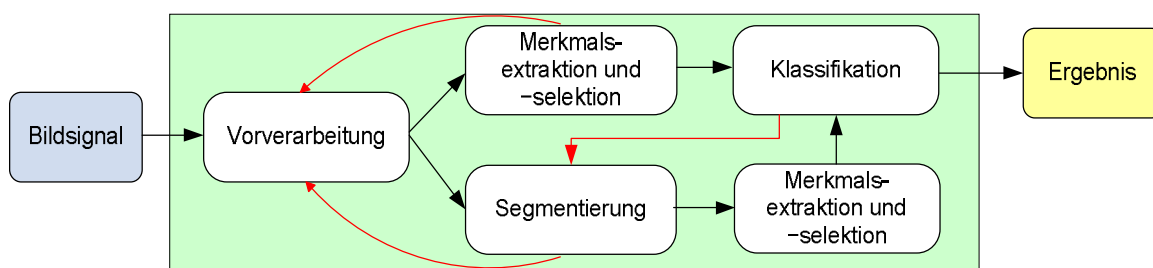


Abbildung 6-51: Aufbruch der linearen Abfolge von BV-Methoden.

6.3 Multispektrale und multidimensionale Eingangssignale

Aufgrund der gestiegenen Rechnerleistung werden in der industriellen BV zunehmend Systeme eingesetzt, die anstelle von zweidimensionalen Intensitätsdaten mit einer Grauwertauflösung von 8bit auch höher auflösende, multispektrale oder multidimensionale Daten für die Analyse und Problemlösung verwenden. Eine Umgebung für den automatischen Entwurf von BV-Systemen muss also flexibel bezüglich der Dimensionalität der Eingangsdaten sein. Bei den multispektralen und multidimensionalen Daten kann es sich beispielsweise um Farbbilder, Zeitserien oder Volumenbilder handeln. Die Datenspeicherung, der Datentransfer und die verwendeten BV-Algorithmen müssen hinsichtlich ihrer Funktionalität an diese Daten angepasst werden.

6.4 Multisensorik

Neben der Analyse und Verarbeitung reiner Bilddaten wird in Zukunft auch die Verknüpfung und gemeinsame Verarbeitung multisensorischer Daten, wie beispielsweise Infrarot-, SAR- (Synthetic Aperture Radar) oder Terahertz-Daten eine Rolle spielen.

6.5 Anforderungen an eine Umgebung für den automatischen Entwurf und die Konfiguration eines BV-Systems und des Optimierungsverfahrens

Aus dem Konzept für eine Umgebung für den automatischen Entwurf und die Konfiguration eines Bildverarbeitungssystems ergeben sich die folgenden Anforderungen an die einzelnen Module des BV-Systems.

6.5.1 Hardware und Sensorik

Hardware, Beleuchtung und Sensorik sollten automatisch ansteuerbar sein und flexibel positioniert werden können.

6.5.2 Algorithmik

- Verwendung von modularen BV-Funktionen.
- Algorithmen für die Verarbeitung multispektraler oder multidimensionaler Daten
- Beschreibung und Wertebereich der Freiheitsgrade für jede Funktion
- Beschreibung der notwendigen Eingangs- und Ausgangssignale
- Systematische Analyse der Methoden bezüglich ihrer Wirkung, ihrer Laufzeit u.a.
- Einteilung der Algorithmen in ähnliche Gruppen

- Funktionale Kombination von Methoden zu „Hypermethode“ mit einem gemeinsamen Parametersatz
- Erweiterbarkeit durch eigene Entwicklungen

6.5.3 Datenfluss und Repräsentation

Folgende Datenformate und Datenrepräsentationen müssen bei der Architektur des Systems verfügbar sein.

- Bild zu Bild (z. B. Filter oder morphologische Operatoren)
- Bild zu Merkmal (z. B. Histogramm oder Co-Occurrence-Matrix)
- Merkmal zu Merkmal (z. B. Merkmalsselektion oder Kombination)
- Merkmal zu Bild (z. B. Auswahl einzelner Merkmale oder über Klassifikation)
- Multidimensionale und multispektrale Daten.

6.5.4 Optimierung

Ein Optimierungsverfahren für die Autokonfiguration und den Entwurf eines BV-Systems muss folgenden Anforderungen genügen:

- Optimierung von Auswahlprobleme
- Parameteroptimierung
- Funktionen- und Systemgenerierung (Strukturoptimierung)
- Verarbeitung diskreter Parameter mit sehr unterschiedlichen Wertebereichen und Intervallschritten, sowie einer Verarbeitung kontinuierlicher Merkmale.
- Einflussmöglichkeit auf alle Module des BV-Systems
- Überwachte, halb-überwachte, nicht-überwachte und interaktive Optimierung
- Mehrzieloptimierung
- Flexible und austauschbare Bewertungsfunktionen
- Geringer Konfigurationsaufwand für die Optimierung
- Robuste Lösungen, d. h. keine Überzüchtung

6.6 Umsetzung der Anforderungen an die Architektur

Aus den Anforderungen an die Hardware, die Algorithmik und die Optimierung wurde eine Struktur für die tatsächliche softwareseitige Implementierung entwickelt. Die softwareseitige Ansteuerung der Hardware wurde jedoch nicht berücksichtigt.

Bezüglich der Anforderungen an die Algorithmik wurden folgende der aufgeführten Punkte berücksichtigt und implementiert:

Die BV-Funktionen wurden grundsätzlich modular implementiert und können je nach Anwendungsfall gezielt in die Umgebung eingebunden werden. Dies ermöglicht auch die Erweiterbarkeit durch eigene Entwicklungen.

Neben Standard-1-Kanal-Grauwertbildern können 3d-Volumendaten und multispektrale Daten eingelesen werden. Allerdings sind nicht alle BV-Module für die Verarbeitung von 3d-Daten geeignet. Insbesondere einige Methoden für die Merkmalsberechnung und die Bildfilterung lassen sich zurzeit nur auf 2d-Daten anwenden. Die Kanäle von multispektralen Daten dürfen nur ganzzahlige Werte zwischen 0 und 255 (8bit) beinhalten.

Eine Beschreibung der Freiheitsgrade der Methoden und Informationen bezüglich ihrer Eingangs- und Ausgangssignale ist verfügbar (vgl. auch Anhang A1).

Die Laufzeit eines BV-Graphen kann zurzeit nur für den Gesamtgraph ermittelt werden, eine systematische Analyse bezüglich einzelner Algorithmen ist somit nicht möglich.

Einige ausgewählte Module (z. B. für die adaptive Morphologie) wurden bereits zu Hypermodulen kombiniert. Zusätzlich existiert auch ein Hypermodul für alle aufgeführten Texturanalysemethoden. Die Einteilung der Algorithmen in ähnliche Gruppen erfolgte bisher nur rudimentär, d. h. alle morphologischen Module wurden beispielsweise in einer Bibliothek zusammengefasst. Eine Einteilung aufgrund einer Analyse der Wirkungsweise einzelner Methoden wurde nicht getroffen.

Neben den aufgeführten Anforderungen an Datenfluss und Repräsentation wurde zusätzlich der Bildstapel und daraus resultierend eine Merkmalsmatrix eingeführt. Dies erleichtert den Umgang mit Bildserien. Die einzelnen Repräsentationen wurden in einem sogenannten Datenstapel integriert, was den Datentransfer zwischen zwei heterogenen BV-Modulen erleichtert.

Bezüglich der Optimierung wurde das Hauptaugenmerk auf Parameter- und Strukturoptimierung gelegt. Kontinuierliche Merkmale können hierbei nur über den Umweg sehr kleiner Intervallschritte verarbeitet werden. Die in dieser Arbeit entstandene Umgebung ist nur für eine überwachte Optimierung in Form von manuell erstellten pixelgenauen Referenzbildern

oder Klasseninformationen bzw. für schwach-überwachte Optimierungen verwendbar. Eine Mehrzieloptimierung wird nur in Form von Bestrafungs- oder Belohnungselementen bezüglich eines globalen Gütemaßes berücksichtigt. Diese Elemente können beispielsweise von der Laufzeit eines BV-Graphen oder der Anzahl der Knoten in einem Graphenast abhängen.

Um einer Überzüchtung entgegenzuwirken, besteht die Möglichkeit eine n-fachen Kreuzvalidierung auszuführen. Dabei ist zu berücksichtigen, dass sich die Laufzeit der Optimierung deutlich erhöhen kann.

Insbesondere die Verwendung der kanonischen Partikelschwarmoptimierung für die Parameteroptimierung reduziert den Konfigurationsaufwand der Optimierung. Neben der Anzahl der Individuen in einer Population müssen praktisch keine weiteren Parameter festgelegt werden. Die Optimierung mittels C-PSO kommt somit einer 1-Klick-Optimierung am nächsten. Allerdings können zusätzliche Einstellungen, wie beispielsweise zeitabhängige Veränderungen von Parametern zu besseren Ergebnissen führen (vgl. auch Kapitel 5.6).

Bei der aus den Anforderungen entwickelten Struktur wurden insbesondere die spezifischen Anforderungen der gewählten Optimierungsverfahren, d. h. GA/GP und PSO berücksichtigt. GA/GP bieten den Vorteil, dass sich eine graphenähnliche Struktur einer Bildverarbeitungslösung eins zu eins auf die Optimierung abbilden lässt. Jede Bildverarbeitungsmethode ist mit ihren Parametern sozusagen ein Knoten von GA/GP. Die Repräsentation einer BV-Lösung erfolgt durch eine lineare Abfolge ikonischer BV-Methoden. Die Originaldaten und Ergebnisbilder aus der Verarbeitung werden hierbei in global verfügbaren, temporären Datenspeichern gespeichert. Die Originaldaten können dabei nicht von den Ergebnisbildern überschrieben werden (vgl. Abbildung 6-52).

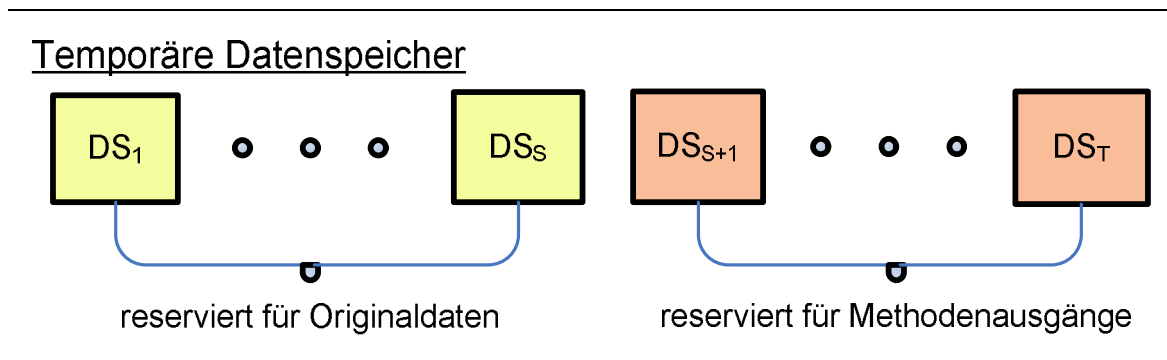
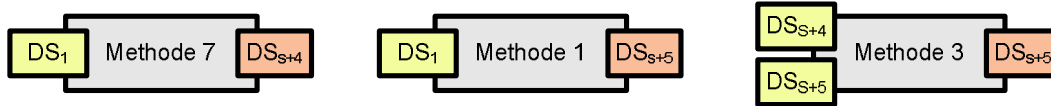


Abbildung 6-52: Für die Umsetzung einer Graphenstruktur stehen temporäre Datenspeicher für die Originaldaten und die Ergebnisbilder der BV-Methoden zur Verfügung.

Eine komplexe Graphenstrukturen ergibt sich durch den variablen Zugriff der BV-Methoden auf diese Speicherbereiche (vgl. Abbildung 6-53).

Eine mögliche Methodenkette mit den entsprechenden Ein- und Ausgängen



Resultierender Graph

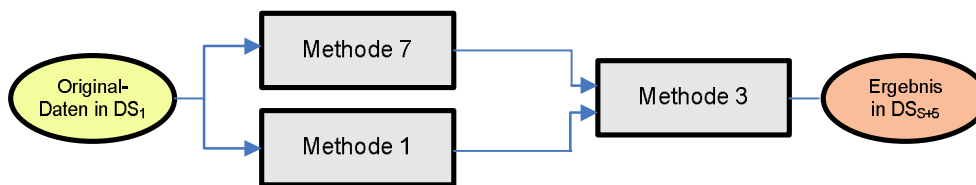


Abbildung 6-53: Durch den flexiblen Zugriff auf die temporären Datenspeicher können sich so komplexe Graphenstrukturen ergeben.

Für die Autokonfiguration eines BV-Systems mit einer festen Struktur und einer festen Anzahl von Parametern kann die gleiche Repräsentation der Daten verwendet werden wie bei GA. Eine automatische Strukturoptimierung mit PSO ist mit dieser Repräsentation allerdings nicht möglich, da es kein definiertes Abstandsmaß für zwei Lösungen mit unterschiedlichen BV-Methoden und deren Parametern gibt.

Bei der in dieser Arbeit verwendeten ad hoc Anpassung der Repräsentation der Daten für die Strukturoptimierung mit PSO wird für jeden Knoten in der linearen Repräsentation des Bildverarbeitungsgraphen der komplette Parametersatz aller möglichen BV-Methoden gespeichert. Allerdings ist pro Knoten immer nur eine BV-Methode in der BV-Lösung aktiv. Da jetzt für jeden Knoten der komplette Parametersatz gespeichert wird, kann ohne Probleme der benötigte Abstand für die Anpassung der Geschwindigkeit berechnet werden.

6.7 Zusammenfassung

In diesem Kapitel wurde ein Konzept und die daraus resultierende Architektur für die Umgebung für den automatischen Entwurf und die Autokonfiguration von BV-Systemen vorgestellt, die auf bestehenden Ansätzen, wie u.a. dem QuickCog-System, aufsetzt und die wesentlichen Teile der üblichen Verarbeitungskette erweitert. In dem folgenden Kapitel werden die automatische Konfiguration und der Entwurf neuer Systeme mit der erarbeiteten Implementierung der Entwurfssystemarchitektur und den innewohnenden GA/GP und PSO Verfahren an praktischen Beispielen vorgestellt.

7 Anwendungen auf die Texturanalyse und 2d/3d-Segmentierung

Anhand der Anforderungen an die Architektur einer Umgebung für den automatischen Entwurf und die Autokonfiguration gemäß Abschnitt 6.5 und der aus den Optimierungsverfahren folgenden Repräsentation eines BV-Systems wurde im Rahmen dieser Arbeit ein Werkzeug, d. h. eine Softwareimplementierung, entwickelt, mit der Anwendungen auf die Texturanalyse und 2d- und 3d- Bildsegmentierung automatisch anhand von Referenz- bzw. Solldaten generiert werden können. Die einzelnen Module der softwareseitigen BV-Kette sind modular implementiert und können bequem über Initialisierungsdateien konfiguriert werden. Dies umfasst eine Auswahl einzelner Methoden der Bildvorverarbeitung, der Merkmalsextraktion, der Bewertung und der Klassifikation. Des Weiteren kann angegeben werden, wie die Bilddaten verwendet werden sollen. Für Aufgaben der klassischen Texturanalyse können die Trainingsdaten beispielsweise in voneinander unabhängige Bildregionen unterteilt werden. Mittels Kreuzvalidierung während des Entwurfs und der Konfiguration einer Lösung werden dann stabile Lösungen geschaffen. Bei der Erkennung von Texturdefekten hingegen können die Regionen eines Bildes als Einheit angesehen werden.

Neben einer Optimierung ohne explizite Verwendung von Vorwissen können bestehende Bildverarbeitungsabläufe in die Software geladen und beispielsweise für die Autokonfiguration herangezogen werden. Zusätzlich besteht die Möglichkeit, Parameterwerte einzelner Methoden gezielt festzulegen oder zu beschränken. Für die überwachte Optimierung können entweder Referenzbilder mit einer manuellen Segmentierung (schwarz/weiß) oder mit ganzzahliger Klasseninformation verwendet werden. Zusätzlich besteht die Möglichkeit Referenzbilder automatisch anhand von geometrischen Kennzahlen zu erstellen. Die Konfiguration der Optimierung, d. h. von PSO und GA/GP erfolgt über Initialisierungsdateien.

In den folgenden Abschnitten werden einige exemplarische Beispiele für den Entwurf und die Autokonfiguration von Systemen für die Segmentierung von 2d- und 3d-Daten und für die Texturklassifikation und Defekterkennung vorgestellt. Als Daten werden unterschiedliche Texturbilder für die Texturklassifikation und synthetische und Applikationsnahe Daten für die Defekterkennung verwendet.

7.1 Texturklassifikation mit C-PSO und GA

Die Leistungsfähigkeit von GA und PSO bei der Konfiguration und Methodenauswahl eines BV-Systems für die überwachte Texturklassifikation wird im Folgenden untersucht. Die Texturbilder

werden hierzu in rechteckige Regionen R der der Größe $M \times N$ zerteilt. Die Regionen werden so platziert, dass eine maximale Anzahl von Regionenmittelpunkten (Abbildung 7-54) innerhalb des Texturbildes liegt. Es können verschiedene Randbehandlungen, wie beispielsweise eine periodische oder konstante Fortsetzung des Bildes gewählt werden. In diesem Fall werden Regionenpunkte außerhalb der Grauwerttextur auf den konstanten Grauwert $g = 0$ gesetzt. Eine Region wird der Klasse zugeordnet, der gemäß des dazugehörigen Masken- oder Sollbildes S die meisten Bildpunkte innerhalb dieser Bildregion angehören.

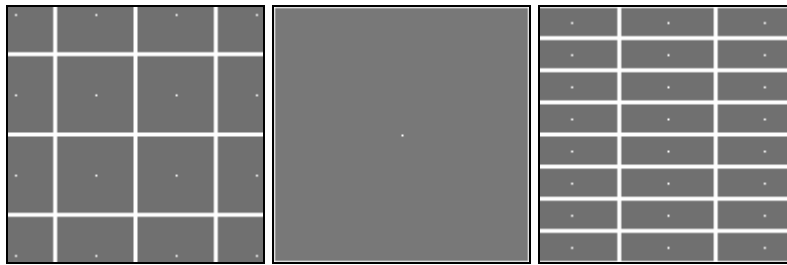


Abbildung 7-54: Beispiele für die Zerteilung eines Bildes in disjunkte Bildregionen. Die Anzahl der im Bild liegenden Mittelpunkte wird in diesem Fall maximiert.

Für die Berechnung der Texturmerkmale werden verschiedene Methoden, insbesondere die in Kapitel 3 aufgeführten NLOK, AK, LTEM und Statistiken erster (HIST) und zweiter Ordnung (COOC) verwendet. Die Bewertung einer Lösung erfolgt anhand des Klassifikationsergebnisses R_{Ok} mit einer C-SVM mit einem RBF-Kern [7]. Die Mehrklassen-Klassifikation wird hierbei durch einen „one-against-one“ Ansatz mit insgesamt $L(L-1)/2$ Klassifikatoren erreicht. Mittels der Optimierung wird die optimale Größe einer Bildregion automatisch ermittelt, der Texturoperator ausgewählt und konfiguriert, ggf. Texturmerkmale bei der Verwendung von Statistiken erster und zweiter Ordnung ausgewählt und der Klassifikator konfiguriert. Das generelle Ablaufschema für die Texturklassifikation ohne zusätzliche Bildvorverarbeitung ist in Abbildung 7-55 dargestellt. Das dargestellte Texturanalysesystem hat in diesem Fall eine feste Ablaufstruktur. Die Parametrierung und Auswahl der einzelnen Methoden für die Berechnung von Texturmerkmalen erfolgt gemäß der in Tabelle 3-3 in Kapitel 3.8 angegebenen Konfigurationsbereiche. Dieses prinzipielle Schema wurde ebenfalls bei früheren eigenen Publikation [101, 102] angewendet.

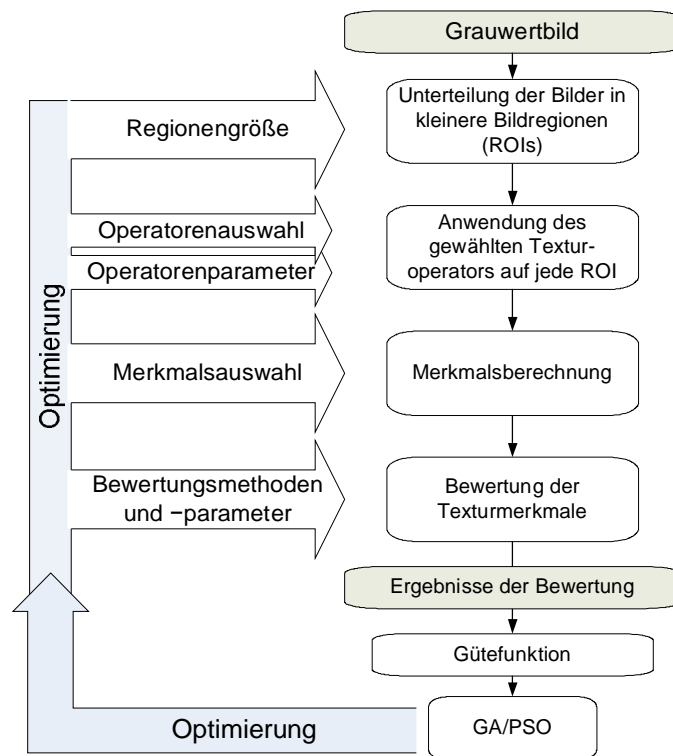


Abbildung 7-55: Konfiguration eines Systems für die Texturklassifikation.

7.1.1 Elitismus und Diversität

Anhand eines Problems der Texturklassifikation wird zunächst untersucht, in wieweit sich der Einsatz von Elitismus und Diversität auf die Klassifikationsergebnisse auswirken. Als Daten werden beispielhaft verschiedene Grauwerttexturen der Größe 128×128 Pixel (vgl. Abbildung 7-56) verwendet.

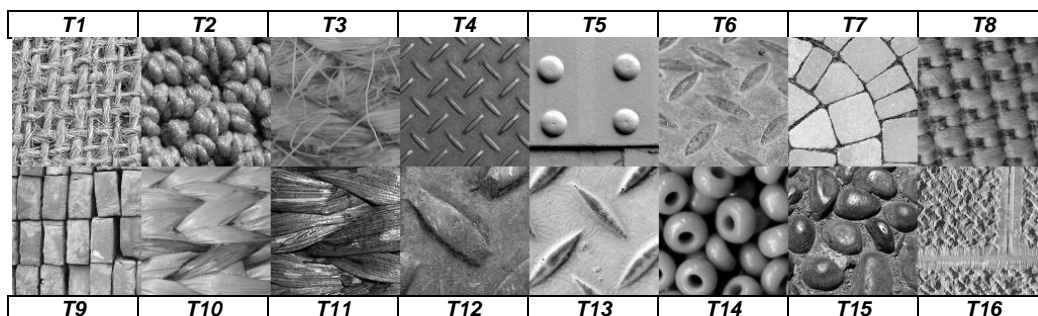


Abbildung 7-56: Sechzehn verschiedene Texturen für die Texturanalyse.

Gemäß Abbildung 7-55 und Tabelle 3-3 werden sowohl die Auswahl und die Parametrierung der Texturoperatoren, die Regionengröße und die Klassifikatorparameter angepasst. Für die

Optimierung wird pro Klasse je eine Texturbild verwendet und in disjunkte Bildregionen unterteilt. Die mittlere Fitness f_o wird während der Optimierung aus den Klassifikationsraten einer 2-fachen Kreuzvalidierung berechnet. Um die Generalisierungsfähigkeit der optimierten Lösung zu testen, wird die C-SVM anhand der kompletten Bilddaten der Optimierung trainiert und anhand der Bildregionen von je zwei weiteren Texturbildern pro Klasse getestet. Die Fitness f_g ist somit die auf hundert normierte Klassifikationsrate der Testregionen. Die Optimierung wird abgebrochen, wenn f_o den maximal möglichen Wert erreicht, d. h. dass während der Kreuzvalidierung alle Regionen richtig klassifiziert wurden.

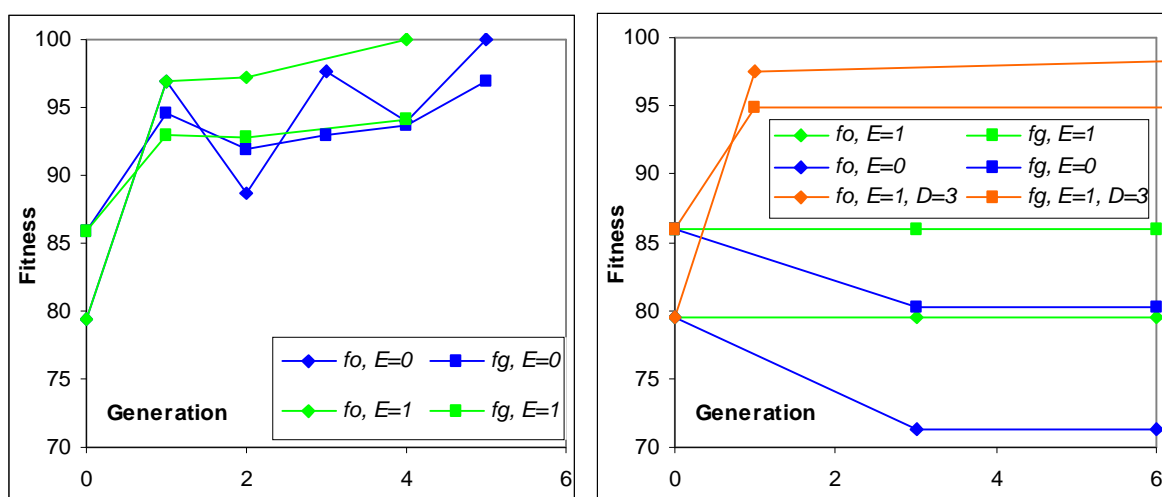


Abbildung 7-57: Korrekt klassifizierte Texturregionen während der Optimierung (2-fach Kreuzvalidierung) und für die Generalisierung. Links: C-PSO, Populationsgröße $N_p = 20$. Rechts: GA, $N_p = 20$, $Kr = 0,7$, $Mu = 0,1$.

In Abbildung 7-57 sind die Ergebnisse für die Texturklassifikation von acht verschiedenen Texturklassen T_1 bis T_8 mit C-PSO und GA dargestellt. Ohne Elitismus ($N_E = 0$) können gute Lösungen sowohl bei C-PSO als auch bei GA im Laufe der Optimierung wieder verloren gehen. Aufgrund einer Turnierselektion mit zwei Teilnehmern und fehlender Diversität ($N_D = 0$) bei der GA beinhaltet die Population nach 10 Generationen nahezu identische Individuen. Eine Verbesserung der Lösung wird dadurch erschwert. Bei gleicher Ausgangskonfiguration und einer Diversität von $N_D = 3$, d. h. pro Generation werden drei Individuen neu initialisiert, erhält man auch für GA geeignete Ergebnisse. Ein vergleichbares Verhalten lässt sich bei der Klassifikation von 4 oder 16 verschiedenen Texturen beobachten.

7.1.2 Populationsgröße und Laufzeit der Optimierung

Wird die Anzahl der Individuen pro Population auf 40 Individuen verdoppelt, werden aufgrund der Vielfalt der Lösungen bereits in der ersten Generation Fitnesswerte f_o größer als 95 erreicht. Allerdings verdoppelt sich dadurch auch die Zeit für das Verarbeiten einer Generation. Mittels C-PSO wird der maximal mögliche Fitnesswert bereits nach einigen wenigen Generationen erreicht. Die Optimierung mit GA weist hingegen eher einen langsamen Anstieg für f_o auf.

Vergleich man die Laufzeit von GA und C-PSO für ein einfaches Texturklassifikationsproblem mit nur vier oder acht unterschiedlichen Texturen, lässt sich feststellen, dass mit PSO im Mittel schneller optimale Lösungen, d. h. eine Fitness von $f_o = 100$ erreicht wird. Ein vergleichbares Verhalten lässt sich auch bei der Texturklassifikation von sechzehn Texturklassen beobachten. Hierbei wird das Optimum allerdings deutlich später erreicht.

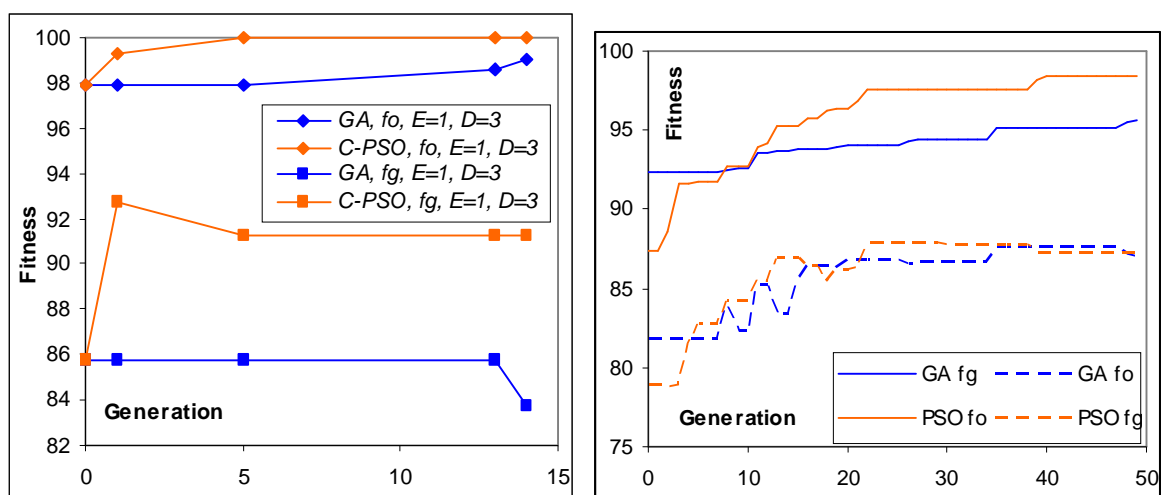


Abbildung 7-58: Korrekt klassifizierte Texturregionen während der Optimierung (2-fach Kreuzvalidierung) und für die Generalisierung. $N_p = 40$. Links: 8 Texturklassen, rechts: 16 Texturklassen (mittlere Fitness, 5 Durchläufe).

7.2 Detektion von Texturfehlern mit C-PSO und GA

Das in Abschnitt 7.1 beschriebene Verfahren für die Texturklassifikation lässt sich ebenso für die Detektion von Texturfehlern einsetzen. Je nach Problemstellung, d. h. beispielsweise der Art des Defekts oder der Textur des Hintergrundes können hier unterschiedliche Texturoperatoren, Regionengrößen oder Klassifikatoreinstellungen optimale Ergebnisse liefern. Im Folgenden wird das in Abbildung 7-55 dargestellte Texturklassifikationssystem für verschiedene Bilddaten mit Texturfehlern angewendet. Hierbei wird die Leistungsfähigkeit von GA und C-PSO, verschiedener Texturoperatoren und die Auswirkung einer 10-fach Kreuzvalidierung auf die Generalisierungsfähigkeit der optimierten Lösung untersucht.

7.2.1 Synthetische, schwach markierte Texturdefekte

Bei den DAGM-Datensätzen für die Erkennung von Defekten in texturierten Bildern handelt es sich um synthetische Datensätze, die in Anlehnung an vergleichbare, reale und anspruchsvolle Defekterkennungsproblemen in der industriellen Fertigung erstellt wurden. Diese Daten wurden im Rahmen der Konferenz DAGM 2007 als Trainingsdaten für den Wettbewerb 'Weakly Supervised Learning for Industrial Optical Inspection' vorgestellt. Bei den Daten handelt es sich um sechs Bilddatensätze mit verschiedenen Hintergrundtexturen. Auf jeweils 150 der pro Datensatz zur Verfügung stehenden 1150 Grauwertbilder der Größe 512×512 Bildpunkte befindet sich zusätzlich ein Texturdefekt. Die Defekte sind schwach markiert (vgl. Abbildung 7-59), d. h. die Defektposition und Ausrichtung sind zusammen über eine umschreibende Ellipse angegeben. Die Ellipsenparameter sind die X - und Y -Position des Ellipsenmittelpunktes $P(x', y')$, die Längen der Halbachsen a und b und der Winkel α zwischen der positiven X -Achse und der Hauptachse der Ellipse.

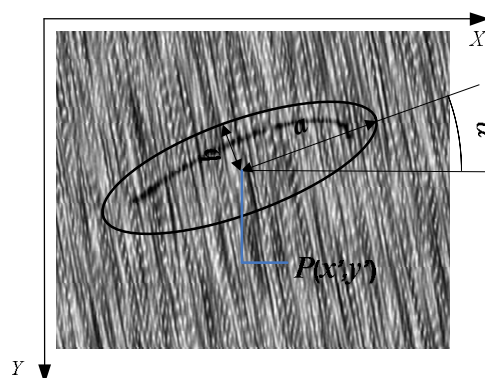


Abbildung 7-59: Schwache Markierung eines Defektes mittels einer Ellipse in den texturierten Bilddaten des DAGM-Datensatzes. Die Ellipsenparameter sind die X - und Y -Position des Ellipsenmittelpunktes $P(x', y')$, die Längen der Halbachsen a und b sowie der Winkel α zwischen der Positiven X -Achse und der Hauptachse der Ellipse.

Die Bilder innerhalb eines Datensatzes sind sich sehr ähnlich. Die sechs unterschiedlichen Datensätze hingegen wurden jeweils mittels eines eigenen Texturmodells und eines eigenen Defektmodells erzeugt. An dieser Stelle wird zunächst der sechste Datensatz *DAGM-6* verwendet, da dieser sehr große und deutliche Texturdefekte enthält. Diese Defekte lassen sich vergleichsweise einfach detektieren. Bei den Defekten handelt es sich um grobe dunkle linienförmige Strukturen, deren Länge in etwa der Bildbreite entspricht. Beispiele für die Normaltextur ohne Fehler und für Bilder mit Defekt sind in Abbildung 7-60 dargestellt.

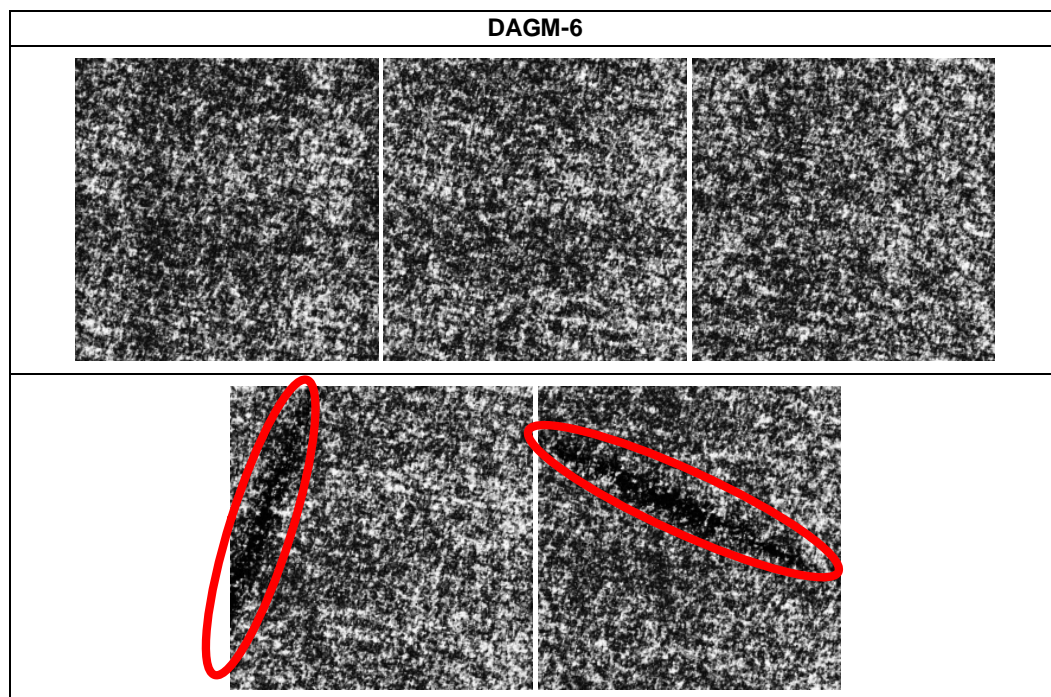


Abbildung 7-60: DAGM Datensatz 6. Die Defekte sind parallele dunkle Striche mit variierender Breite. DAGM-6: Bei den Defekten handelt es sich grobe dunkle linienförmige Strukturen, deren Länge in etwa der Bildbreite entspricht.

Anhand der schwachen Markierung der Fehler wird während der Optimierung für jede Regionengröße ein passendes Referenzbild S erstellt. Eine rechteckige Bildregion wird hierbei als Defektregion gewertet, wenn mindestens fünf Prozent der Bildpunkte einer Region zu der markierten Ellipse gehören. Zwangsläufig wird die markierte Fehlerregion somit ggf. deutlich größer als der eigentliche Defekt. Eine hundertprozentige Klassifikationsrate für alle Regionen ist dadurch nicht zu erwarten. Aus Gründen des Aufwands wurde darauf verzichtet, den kompletten Datensatz für die Optimierung zu verwenden. Ein einzelner Optimierungslauf würde bei Verwendung aller Daten mehrere Tage in Anspruch nehmen. Um die Zeit der Optimierung weiter zu verringern wurden die Bilder zusätzlich auf ein Viertel ihrer Größe verkleinert. Da die Defekte selbst sehr groß sind, spielt diese Verkleinerung bei der Defekterkennung nur eine untergeordnete Rolle.

7.2.2 Applikationsnahe Daten aus der Inspektion von gegerbtem Leder

Gegerbtes Leder ist ein Grundstoff für eine breite Palette hochwertiger Endprodukte wie zum Beispiel Möbel, Schuhe oder Autositze. Anhand der Defektanzahl und der Defektpositionen kann beispielsweise eine Einschätzung der Ledergüte erfolgen und eine ökonomisch sinnvolle Platzierung von Schnittschablonen vorgenommen werden. Die folgenden Beispiele (vgl. Abbildung 7-61) zeigen Kratzer auf gegerbten Leder. Die Bilder wurden mittels einer

Hellfeldbeleuchtung und einer Zeilenkamera aufgenommen. In Abbildung 7-62 sind neben einigen Beispiel-Kratzern pixelbasierte manuelle Segmentierungen dargestellt.

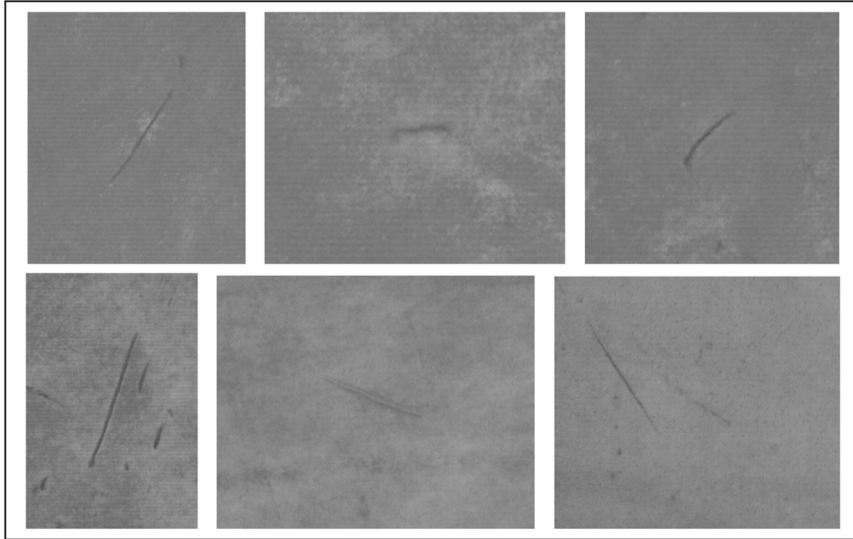


Abbildung 7-61: Kratzer auf gegerbtem Leder.

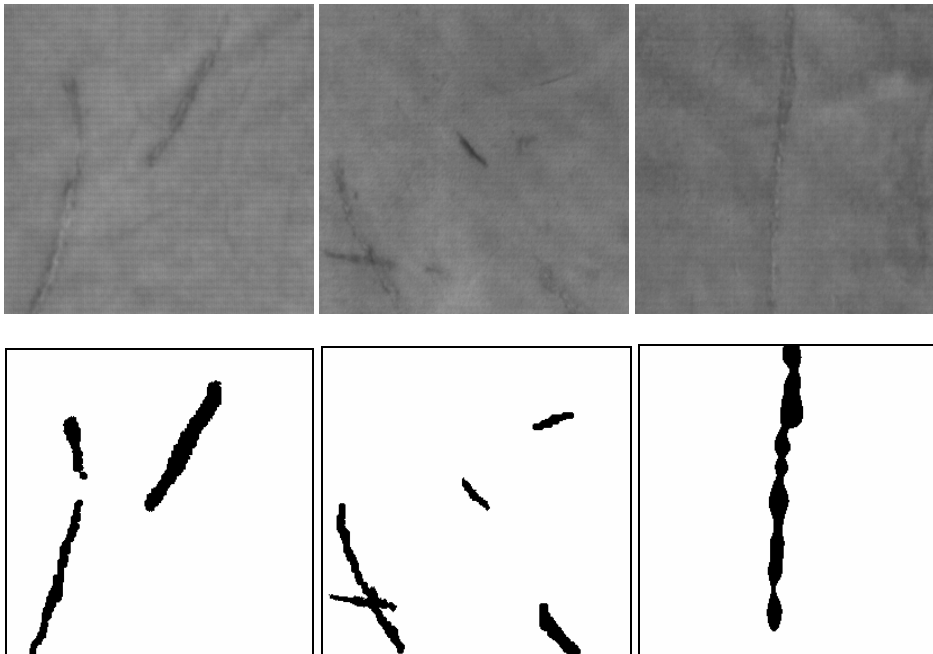


Abbildung 7-62: Pixelbasierte manuelle Kratzer-Klassifikation.

7.2.3 Textiltexturdefekte der Datenbank TILDA

Bei der Bilddatenbank TILDA²⁴ (Textiltexturdatenbank) [129] handelt es um Bilder von vier verschiedenen Textilstoffklassen, deren zugrunde liegende Textilstruktur sich voneinander unterscheidet (vgl. Abbildung 7-63). Für jede Stoffklasse gibt es Referenz und Defektbilder, wobei sich die Defekte in sieben Fehlerklassen unterteilen lassen. Bei den Defekten im oder am Gewebe handelt es sich beispielsweise um Löcher und Schnitte im Textilstoff, Flecken oder Verunreinigungen auf dem Stoff, strukturelle Fehler im Gewebe (Webfehler), Fremdkörper auf dem Stoff oder Falten im Stoff. Zusätzlich wurden „Fehlerbilder“ durch eine Variation der Beleuchtung und eine Veränderung der Kameraposition erzeugt. Zu jedem Bild gibt es eine kurze Fehlerbeschreibung, eine eindeutige manuelle Segmentierung der Fehler existiert hingegen nicht. Aus diesem Grund wurden für die Bilder der TILDA-Datenbank zusätzlich manuelle Referenzbilder unter Verwendung eines kreisförmigen Pinselwerkzeugs für die Markierung der Fehler erzeugt. Für die Detektion von Texturfehlern wurden hier zunächst Textilstoffklasse *c1* und die Fehlerklasse *e4* mit Fremdkörpern auf dem Stoff ausgewählt. Aus Gründen des Aufwands wurden auch diese Bilder mit dem Faktor zwei für die Optimierung verkleinert.

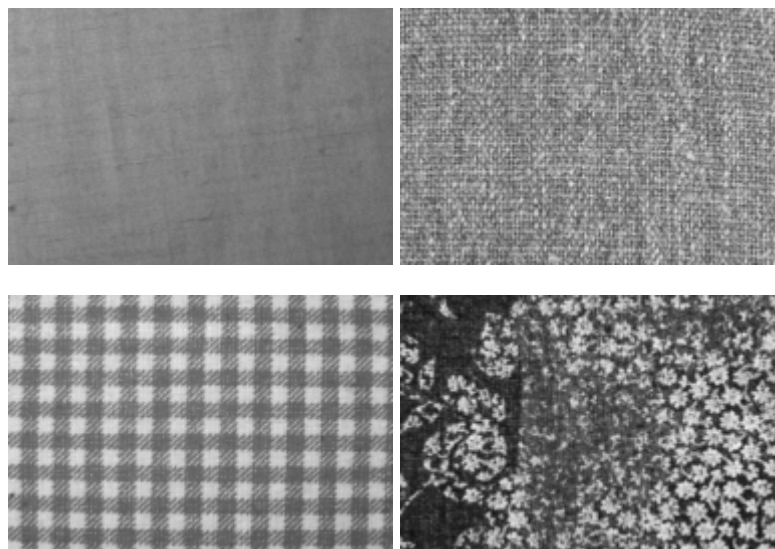


Abbildung 7-63: Die vier Textilstoffklassen der Datenbank TILDA. Oben links: Klasse *c1*, oben rechts: Klasse *c2*, unten links: Klasse *c3*, unten rechts: Klasse *c4*.

²⁴ <http://imb.informatik.uni-freiburg.de/research/dfg-texture/tilda/index.de.html> (letzter Zugriff 30.11.08), Lehrstuhl für Mustererkennung und Bildverarbeitung, Albert-Ludwigs-Universität Freiburg.

7.2.4 Aufbau des Experiments

Für die Detektion der Texturfehler wurden wie für die Texturklassifikation die in Kapitel 3 aufgeführten Methoden der Texturanalyse NLOK, AK, LTEM und Statistiken erster (HIST) und zweiter Ordnung (COOC) für die Berechnung der Texturmerkmale herangezogen. Neben den Histogrammeinträgen werden optional die statistischen Merkmale HISTS und HARA1 verwendet. Die Bewertung einer Lösung erfolgt anhand des Klassifikationsergebnisses R_{Ok} mit einer C-SVM mit einem RBF-Kern.

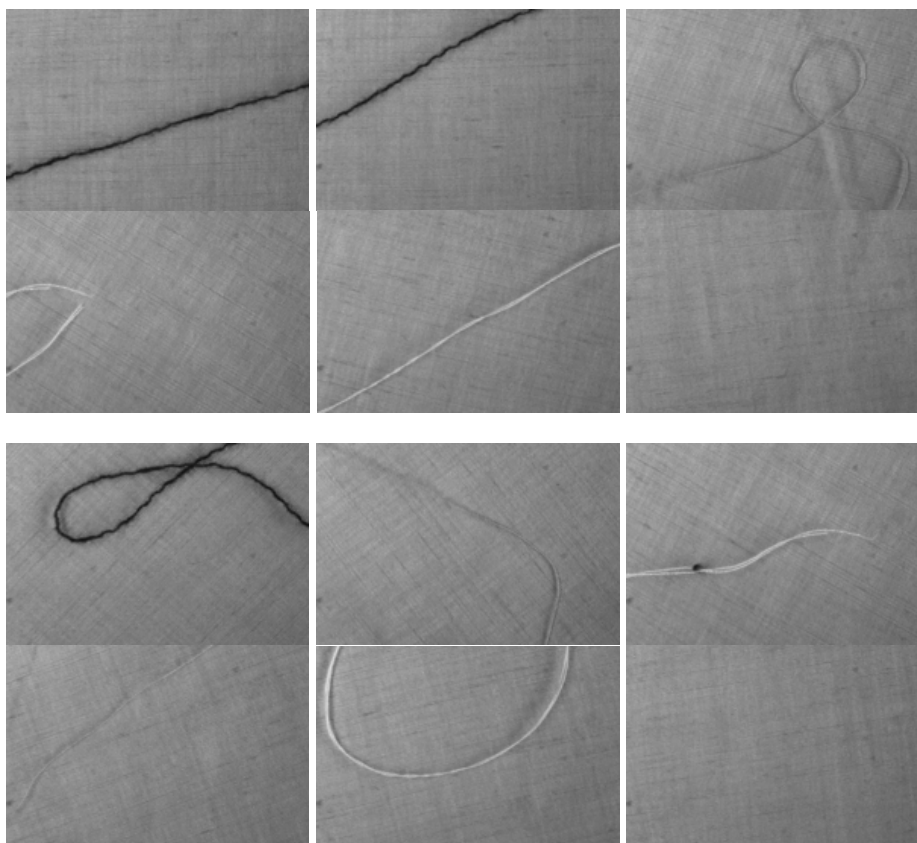


Abbildung 7-64: Trainings- (oben) und Testdaten (unten) für die TILDA-Klasse c1e4. Dargestellt sind jeweils die Bilder mit Defekt (Fremdkörper) und je eine Referenzbild.

Für den Entwurf des Systems wurden bei den TILDA-Daten je fünf Bilder mit und ohne Defekt verwendet. Zum Testen der optimierten BV-Lösung wurden 40 Bilder (5 mit Defekt und 40 Bilder ohne Defekt) verwendet (vgl. Abbildung 7-64). Bei dem DAGM-Datensatz wurden 20 (10/10) und 100 (50/50) Bilder für die Optimierung und das Testen verwendet (vgl. Tabelle 7-12). Die Optimierung wurde pro Datensatz fünf Mal wiederholt. Für die Optimierung werden die Parameter in Tabelle 7-13 verwendet:

Tabelle 7-12. Anzahl der verwendeten Trainings- und Testdaten

	Trainingsdaten	Testdaten
TILDA	10 (5/5)	40 (5/35)
DAGM-6	20 (10/10)	100 (50/50)
Leder	3 (3/0)	6 (6/0)

Tabelle 7-13. Parameter der Optimierung

	N_P	20 Individuen
GA/GP	Turnier-Selektion, Anzahl der Konkurrenten	2
	1-Punkt-Kreuzung, Kr	70
	Mutation, Mu (PM:MM:EAM)	50 (1/0/0)
	Bewertung	Klassifikation C-SVM
C-PSO	---	---

7.2.5 Ergebnisse und Diskussion

Tabelle 7-14 zeigt die durchschnittlich erreichte Fitness, d. h. die mittlere Klassifikationsrate bei einer Zweifach-Kreuzvalidierung von jeweils fünf Optimierungsläufen mit PSO bzw. GA für die Trainingsdaten und die Generalisierungsdaten.

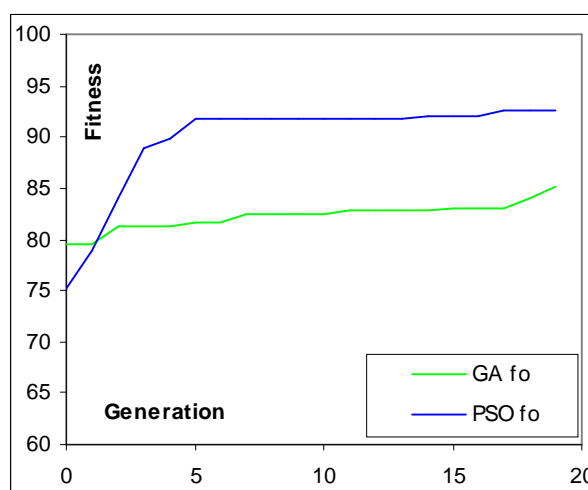


Abbildung 7-65: Korrekt klassifizierte Texturregionen während der Optimierung (2-fach Kreuzvalidierung) des TILDA-Datensatzes c1e4. Dargestellt ist die mittlere Klassifikationsrate aus fünf Optimierungsdurchläufen.

Tabelle 7-14. Durchschnittlich erreichte Fitness bei der Defekterkennung mit 2-fach Kreuzvalidierung

PSO	Training	Generalisierung
TILDA	92,49	87,26
Leder	95,28	93,11

GA	Training	Generalisierung
TILDA	85,20	84,73
Leder	92,43	89,83

Mit GA und PSO wurden hierbei vergleichbare Lösungen erzielt, wobei PSO etwas schneller zum Erfolg geführt hat als GA. Wird die Kreuzvalidierung während des Trainings auf eine 10-fache Kreuzvalidierung erhöht (vgl. Tabelle 7-15), steigt die Klassifikationsrate des Testdatensatzes.

Tabelle 7-15. Durchschnittlich erreichte Fitness bei der Defekterkennung mit 10-fach Kreuzvalidierung

PSO	Training	Generalisierung
TILDA	90,11	91,27
Leder	92,79	94,61

PSO	Training	Generalisierung
TILDA	82,32	85,52
Leder	90,65	92,34

Für die DAGM-Datensätze DAGM-4 bis DAGM-6 ergeben sich bei der Texturklassifikation die Konfusionsmatrizen gemäß Tabelle 7-16:

Tabelle 7-16. Durchschnittlich erreichte Fitness bei der Defekterkennung mit 10-fach Kreuzvalidierung

GA	Soll/Ist	Kein Defekt	Defekt	DAGM
	Kein Defekt	137	13	4
	Defekt	4	46	
	Kein Defekt	148	2	5
	Defekt	19	31	
	Kein Defekt	143	7	6
	Defekt	0	50	

GA	Soll/Ist	Kein Defekt	Defekt	DAGM
	Kein Defekt	146	4	4
	Defekt	5	45	
	Kein Defekt	142	8	5
	Defekt	14	36	
	Kein Defekt	149	1	6
	Defekt	0	50	

7.3 Mehrzieloptimierung mit PSO

Anhand des DAGM-6 Datensatzes wurde zusätzlich untersucht, ob eine Bestrafung von zeitintensiven Algorithmen über eine additive Fitnessbewertung die Auswahl der Algorithmen beeinflusst.

$$f_O = R_{Ok} - \frac{Penalty}{Normierung} \quad (7.41)$$

Der Bestrafungsterm kommt nur zum Einsatz, wenn ein vorgegebenes Kriterium – in diesem Fall eine maximale Rechenzeit – nicht eingehalten wird. Die Auswahl der Texturoperatoren für die Defektklassifikation wurde in diesem Fall auf HIST und COOC, und auf die Berechnung der daraus resultierender Merkmale beschränkt. Neben der Regionengröße wird in diesem Fall auch die Anzahl der verwendeten Grauwerte optimiert, da diese einen entscheidenden Einfluss auf die Rechenzeit haben können. Es hat sich herausgestellt, dass Aufgrund der Bestrafung „schnellere“ Lösungen bevorzugt verwendet werden. Diese sind unter Umständen jedoch suboptimal bzgl. der Klassifikationsrate.

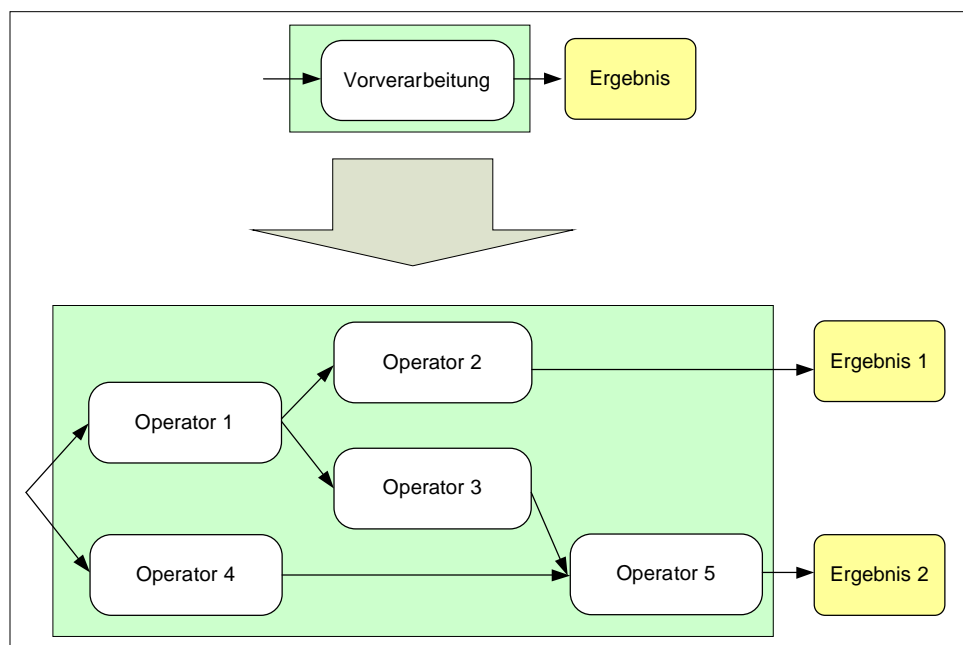


Abbildung 7-66: Ersetzen der Bildvorverarbeitung durch komplexe Vorverarbeitungsgraphen.

7.4 Automatische Defektdetektion mittels ikonischer Bildsegmentierung

Neben der Erkennung von Texturdefekten mittels Klassifikation können Defekte auch mittels ikonischer Bildvorverarbeitung, d. h. beispielsweise mit Schwellwert- oder Filteroperationen vom Hintergrund getrennt werden. In diesem Abschnitt wird die Bildvorverarbeitung durch komplexe Bildverarbeitungsgraphen ersetzt. Abbildung 7-66 zeigt exemplarisch einen

Vorverarbeitungsgraphen mit fünf Operatoren. Die Zusammensetzung eines Graphen ergibt sich gemäß Kapitel 6.6. Die Knoten eines Graphen sind Bildverarbeitungsmethoden, die Kanten repräsentieren den Datenverkehr zwischen den einzelnen Methoden. In diesem Fall handelt es sich bei den transferierten Daten um Bilder oder Bildstapel, d. h. die hier verwendeten ikonischen BV-Methoden benötigen mindestens ein Eingangsbild und liefern mindestens ein Ergebnisbild. Diese Ergebnisbilder können mehrfach von nachfolgenden Operatoren verwendet werden.

Ein BV-Graph kann ein oder mehrere Ergebnisbilder liefern. Diese Bilder werden gemäß Kapitel 4 über ein Ähnlichkeitsmaß mit den vorhandenen Sollbildern verglichen. Zum Einsatz kommen in diesem Fall *SQG* und *MI*, da diese sich als besonders günstig bezüglich der Handhabung herausgestellt haben.

Für die Strukturoptimierung des Bildvorverarbeitungsgraphen werden GA/GP eingesetzt. Für die Rekombination werden zwei Methoden eingesetzt. Zwei BV-Graphen werden an unterschiedlichen Positionen (Fall A) auseinander geschnitten und rekombiniert. Startet man mit einer geringen Anzahl an Knoten, können sich durch die Rekombination komplexe Graphen ergeben. Die maximale Knotenanzahl ist in diesem Fall nicht begrenzt. Der Vorteil dieser Methode besteht darin, dass zu Beginn der Optimierung die Rechenzeit aufgrund der geringen Anzahl von Knoten minimal ist. Im zweiten Fall haben alle BV-Graphen die gleiche Anzahl an Knoten (Fall B). Beide Graphen werden an der gleichen Position getrennt und anschließend neu zusammengefügt. Die Gesamtzahl an Knoten bleibt erhalten.

In beiden Fällen wird Mutation gemäß Kapitel 5.4.3 ausgeführt. Tabelle 7-17 zeigt die Parameter der Optimierung.

Tabelle 7-17. Parameter der Optimierung

	N_P	20 Individuen
	Fall A) Anzahl der Knoten zu Beginn	3
	Fall B) Maximale Anzahl der Knoten	10
	Maximale Anzahl von Generationen	20
GA/GP	Turnier-Selektion, Anzahl der Konkurrenten	2
	1-Punkt-Kreuzung, Kr	70
	Mutation, Mu (PM/MM/EAM)	50 (2/1/1)
	Bewertungsmaß	SQG / MI

7.4.1 Exemplarische Ergebnisse der Bildsegmentierung

Abbildung 7-67 zeigt das Ergebnis der Bildvorverarbeitung exemplarisch anhand von drei Beispielbildern für den DAGM-Datensatz DAGM-1. Als Bewertungsmaß wurde MI verwendet. Für die Rekombination wurde Fall A eingesetzt. Die Ergebnisbilder wurden letztendlich mittels eines morphologischen Gradienten und einer Schwellwertoperation erreicht. Der morphologische Gradient hebt die nahezu homogenen Bildbereiche hervor. Mittels der Schwellwertoperation werden diese anschließend vom Hintergrund getrennt. Menschliche Experten haben bestätigt, dass das Ergebnis der automatischen Segmentierung sinnvoll sein kann, diese Lösung sich anhand der Bilddaten jedoch nicht sofort als offensichtliche Lösung anbietet. Die Automatisierung kann also mit minimalem Aufwand geeignete Lösungsvorschläge anbieten. Die gefundene Lösung war nicht auf allen Generalisierungsdaten vollständig erfolgreich, in einzelnen Bildern wurden Teile des Hintergrundes ebenfalls segmentiert, die Lösung bietet sich aber auf alle Fälle als Vorwissen für weitere Optimierungsläufe an.

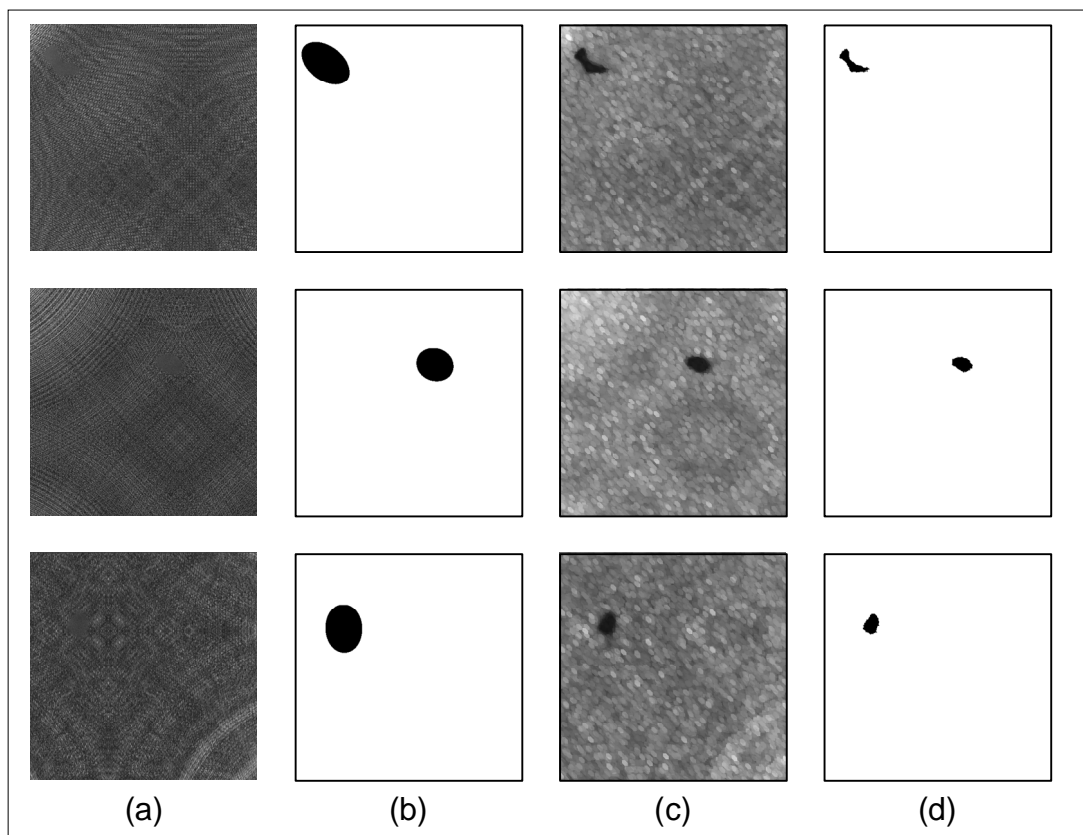


Abbildung 7-67: Erkennung von Texturfehlern in synthetischen Daten (DAGM-1). Auswahl der Originaldaten (a). Manuelle Segmentierung der Defekte mittels einer umschreibenden Ellipse (b). Anwendung des morphologischen Gradienten auf die Originaldaten (c). Ergebnis der Optimierung nach einer abschließenden Schwellwertoperation (d).

Als weiteres Beispiel folgt die Segmentierung der Texturfehler des DAGM-6 Datensatzes. Für die Rekombination wurde hier Fall B verwendet. Die Bildähnlichkeit zwischen den Ergebnisbildern des BV-Graphen und den Referenzbildern wurde in diese Fall über die Summe der quadratischen Grauwertdifferenzen SQG bestimmt. Die über alle Bilder gemittelte SQG wird dabei im Laufe der Optimierung minimiert. Abbildung 7-68 zeigt drei Beispiele der Segmentierung dieses Texturdefekts. Aufgrund des rauen Untergrundes bei der Segmentierung der groben Defekte des DAGM-Datensatzes treten hier deutlich mehr Fehlsegmentierungen in defektfreien Bildern auf. Diese Fehlsegmentierungen sind jedoch ausnahmslos sehr viel kleiner als die Defektsegmentierung, so dass sich über einen Flächenvergleich leicht zwischen Defekt und Hintergrund unterscheiden lässt.

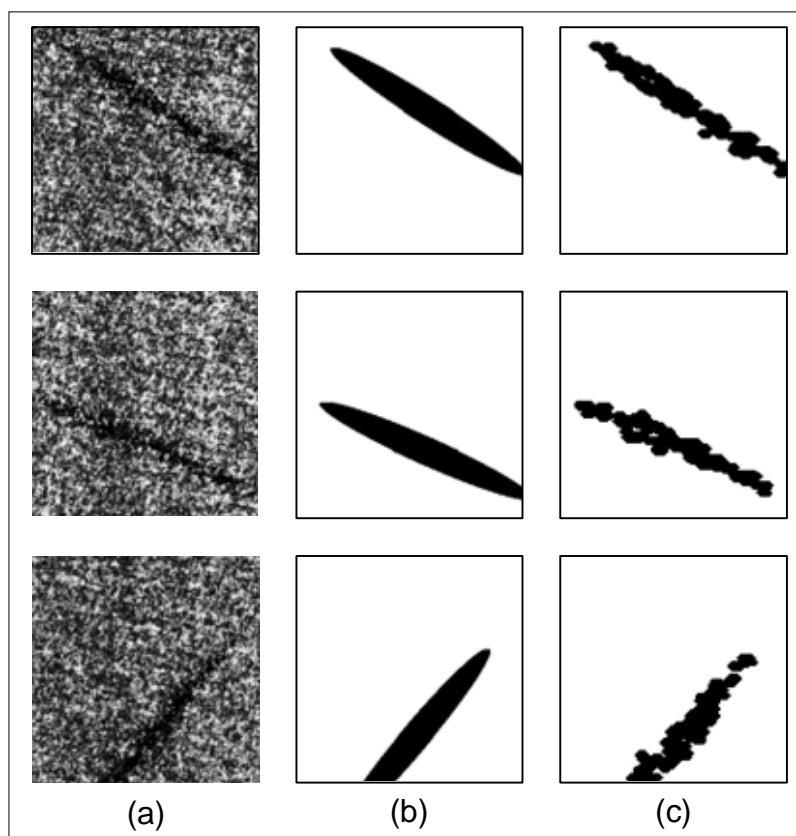


Abbildung 7-68: Erkennung von deutlichen Texturfehlern in synthetischen Daten (DAGM-6). Auswahl der Originaldaten (a). Manuelle Segmentierung der Defekte mittels einer umschreibenden Ellipse (b). Ergebnis der Optimierung (c).

Die Datenzusammensetzung und die resultierende Verlustmatrix sind in Tabelle 7-18 und Tabelle 7-19 dargestellt. Bezüglich der Klassifikationsergebnisse der Wettbewerbsteilnehmer hinsichtlich dieser Trainingsdaten gibt es keine Information. Bei den eigentlichen Wettbewerbsdaten gab es bei den Gewinnern bei 4600 Klassifikationen (6 Klassen) gerade mal drei Fehlsegmentation, womit sie in deutlichen Abstand vor ihren Konkurrenten mit hunderten

von Fehlsegmentierungen gewannen. Diese Information ist der Produktseite des aus dem Wettbewerb entstandenen patentgeschützten Werkzeugs DEFX entnommen. Genaue Zahlen hinsichtlich der Klassifikationsraten stehen leider nicht zur Verfügung.

Tabelle 7-18. Anzahl der verwendeten Trainings- und Testdaten

	Trainingsdaten	Testdaten
DAGM	20 (10/10)	100 (50/50)

Tabelle 7-19. Verlustmatrix DAGM-Datensatz 6 mit GA

	Defekt	Kein Defekt
Defekt	50	0
Kein Defekt	17	33

Als weiteres Beispiel wurden ausgewählte Bilder der Fehlerklasse $e4$ (Fremdkörper auf dem Textilstoff, siehe Abbildung 7-69) der ersten Textilstoffklasse $c1r1$ des TILDA-Datensatzes gewählt (vgl. Abbildung 7-69).

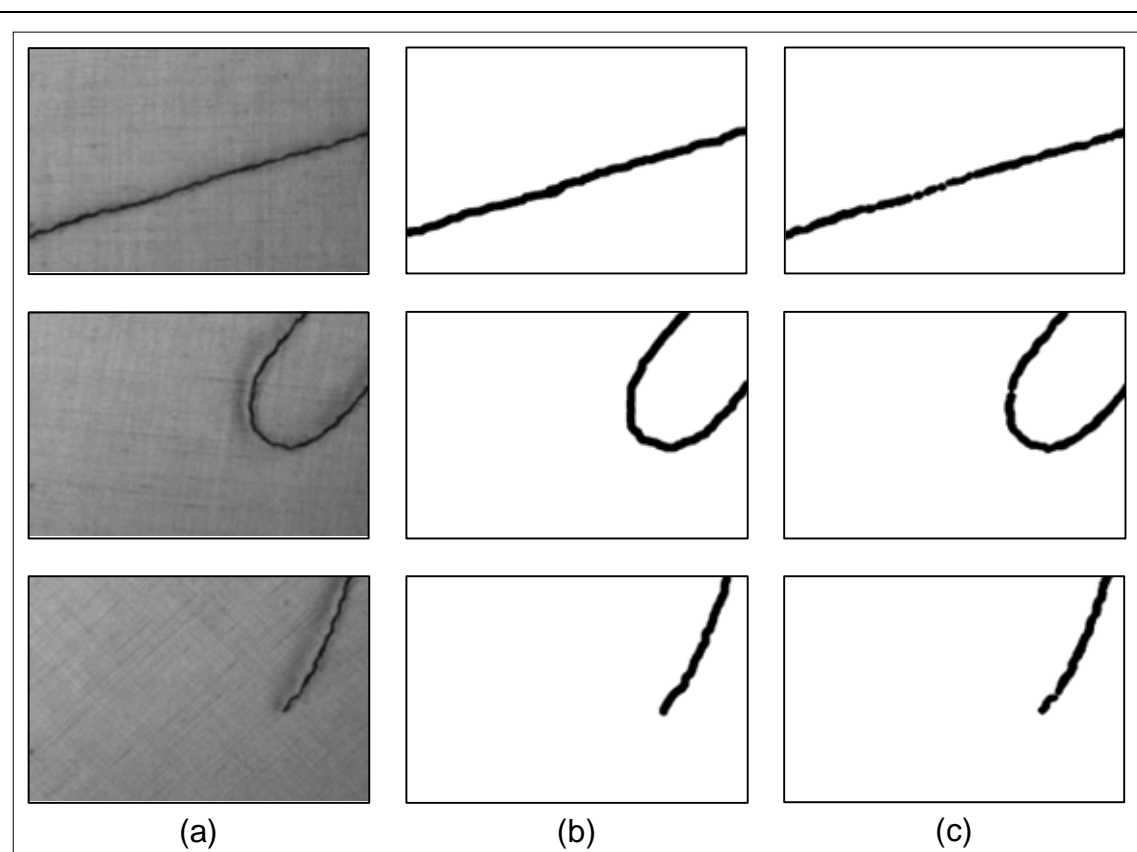


Abbildung 7-69: Erkennung von deutlichen Defekten auf Textilien. Auswahl der Originaldaten (a). Manuelle grobe Segmentierung der Defekte (b). Ergebnis der Optimierung (c). Für die automatische Segmentierung wurden Schwellwert- und Morphologische Operatoren verwendet.

Tabelle 7-20. Anzahl der verwendeten Trainings- und Testdaten

	Trainingsdaten	Testdaten
TILDA	10 (5/5)	40 (5/35)

Tabelle 7-21. Verlustmatrix TILDA-Daten mittels GA/GP

	Defekt	Kein Defekt
Defekt	5	0
Kein Defekt	1	34

Die folgenden Abbildungen (Abbildung 7-70 bis Abbildung 7-72) zeigen weitere exemplarisch Einsatzmöglichkeiten des automatischen Entwurfs eines ikonischen BV-Systems. In Abbildung 7-70 wurde das Gaußsche Rauschen innerhalb eines Bildes eliminiert, ohne die kreisförmige Objektstruktur maßgeblich zu beeinflussen.

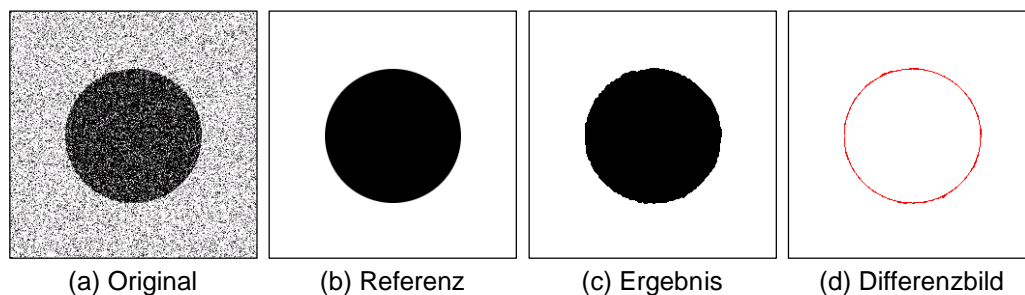


Abbildung 7-70: Eliminierung von Gaußschem Rauschen.

Das folgende Beispiel (vgl. Abbildung 7-71) zeigt die automatische Segmentierung von geprägten Zeichen auf einer metallischen Oberfläche. Die Zeichen, die im weitesten Sinne ebenfalls Oberflächenfehler darstellen, wurden unter verschiedenen Beleuchtungsbedingungen (Tageslicht, Dunkelfeld und Hellfeld) aufgenommen.

Die automatische Bildsegmentierung kann also je nach Beleuchtungssituation geeignete Algorithmen finden. In diesem Fall handelt es sich um drei in ihrer Struktur deutlich unterschiedliche Lösungen. Die hier gefundenen Einzellösungen können also nicht ohne weiteres durch Parameteradaption an die jeweils anderen Beleuchtungssituationen angepasst werden.

In Abbildung 7-72 wurden links laufende Fasern in einem gewebten Material nur mit Hilfe ikonischer BV-Methoden ohne zusätzliche Merkmalsberechnung und Klassifikation extrahiert. Ein solches Vorgehen kann gegenüber einer Objektanalyse und Merkmalsbestimmung zeitliche Vorteile bringen.

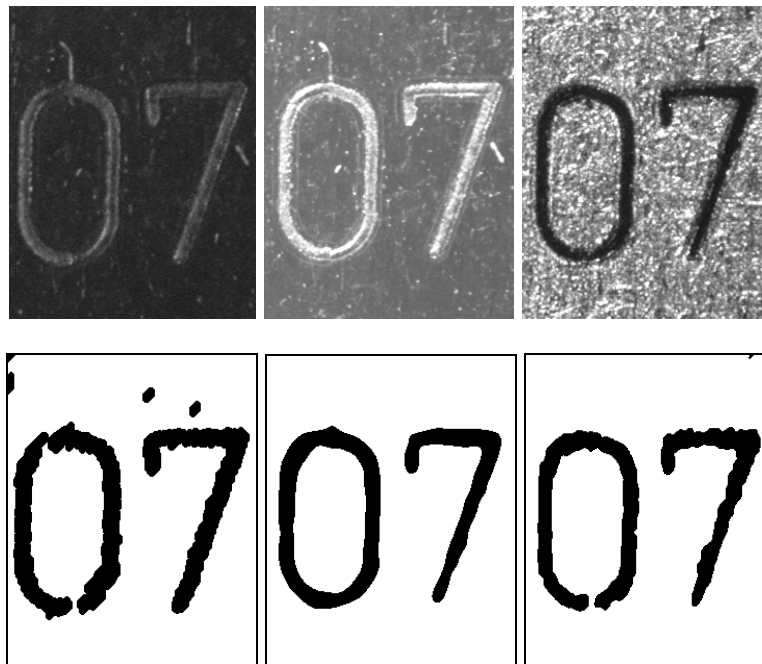


Abbildung 7-71: Geprägte Zeichen in Metall unter verschiedenen Beleuchtungssituationen. Obere Reihe von links nach rechts: undefiniertes Tageslicht, Dunkelfeld, Hellfeld. Untere Reihe: Automatische Segmentierung der geprägten Zeichen.

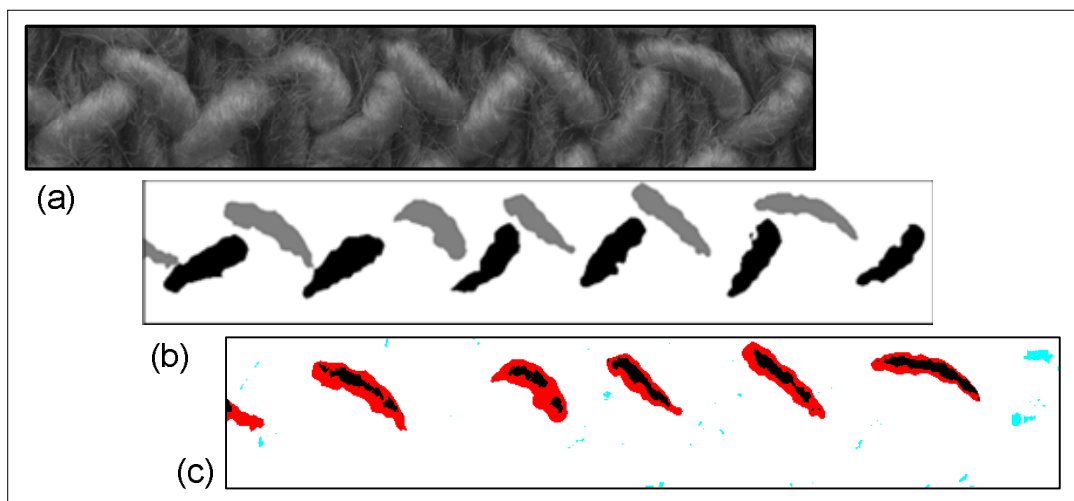


Abbildung 7-72: Segmentierung der links laufenden Kettfäden in einem gewebten Material. Originaldaten (a). Manuelle Segmentierung der links und rechts laufenden Kettfäden (b). Automatische Selektion der links laufenden Kettfäden. Die Differenz zur manuellen Segmentierung ist in Farbe dargestellt (c).

7.5 Optimierte Texturklassifikation

Als zusätzliches Beispiel für die Texturklassifikation wurde aus der OUTEX-Datenbank²⁵ [95] der Datensatz *TC00000* ausgewählt. Dieser Datensatz enthält 24 verschiedene Texturklassen der Größe 128×128 Bildpunkte (vgl. Abbildung 7-73). Pro Texturklasse stehen zwanzig Beispielbilder zur Verfügung. Der Bilddatensatz enthält hierbei neben stochastischen Texturen auch periodische Texturen.

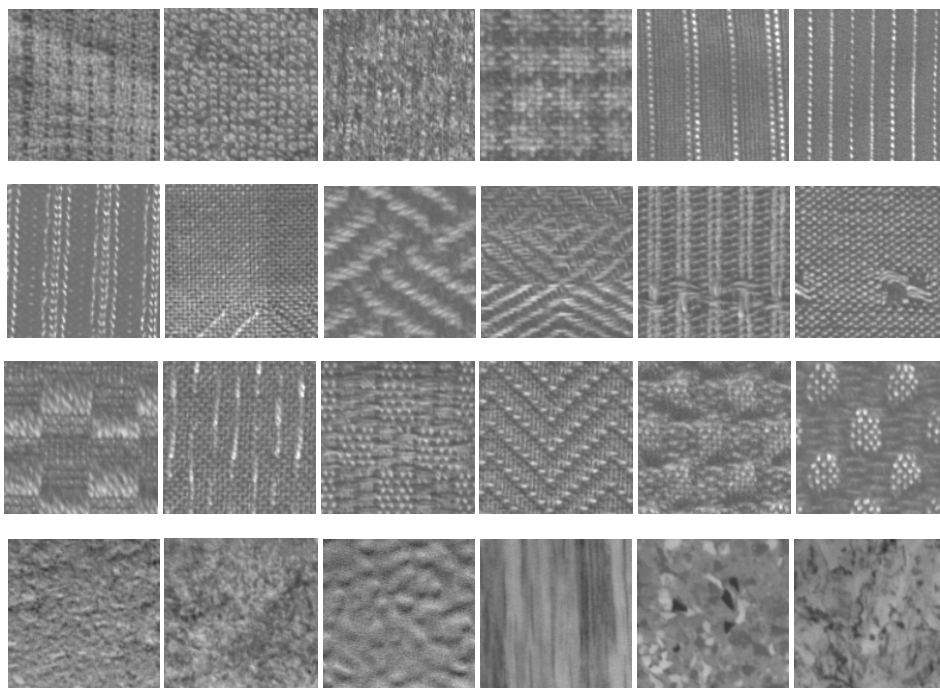


Abbildung 7-73: Texturklassen des OUTEX-*TC00000* Datensatzes.

Für die Problemstellung der Texturklassifikation werden die 480 Texturbilder in disjunkte Teilmengen unterteilt, wobei pro Texturklasse jeweils zehn Bilder für das Training und zehn Bilder für Testzwecke ausgewählt werden. Insgesamt stehen 100 verschiedene Konfigurationen der Bilddatenzusammensetzung zur Verfügung, um Texturklassifikationsalgorithmen zu testen.

7.5.1 Aufbau des Experiments

An dieser Stelle wird die Optimierung dazu verwendet ein Texturklassifikationssystem zu erstellen, das neben den Operatoren der Texturanalyse zusätzliche Bildvorverarbeitungsmethoden enthält (vgl. Abbildung 7-74). Anhand einer deutlich kleineren Untermenge der Texturen, in diesem Fall jeweils die ersten vier Texturen einer Klasse der Trainingsmenge der ersten Datenkonfiguration, wird das Texturklassifikationssystem über eine Zweifach-Kreuzvalidierung trainiert und optimiert. Zum Einsatz bei der Optimierung kommen die

²⁵ Outex Texture Database, <http://www.outex.oulu.fi> (letzter Zugriff: 30.11.2008)

bereits mehrfach aufgeführten Texturanalyseoperatoren NLOK, AK, LTEM, HIST, HISTS, COOC und HARA, Filtermethoden, morphologische Methoden, Histogrammtransformationen und Schwellwertoperatoren. Eine Auflistung der verwendeten Bildvorverarbeitungsmethoden und deren mögliche Parametrierung sind im Anhang zu finden. Gemäß den Testbedingungen wird die Regionengröße in diesem Fall nicht variiert, sie entspricht der Bildgröße von 128×128 Bildpunkten. Für die Optimierung wird Strukturoptimierung mittels GA eingesetzt. Maximal kann die erstellte Lösung zehn verschiedene Bildverarbeitungsmethoden enthalten. Die Texturklassifikationslösung wird über 30 Generationen hinweg trainiert und optimiert und die endgültige Lösung auf die 100 verschiedenen Datenkonfigurationen angewendet. Konform zu den Testbedingungen der OUTEX-Datenbank wird für den Test hierbei keine Kreuzvalidierung verwendet.

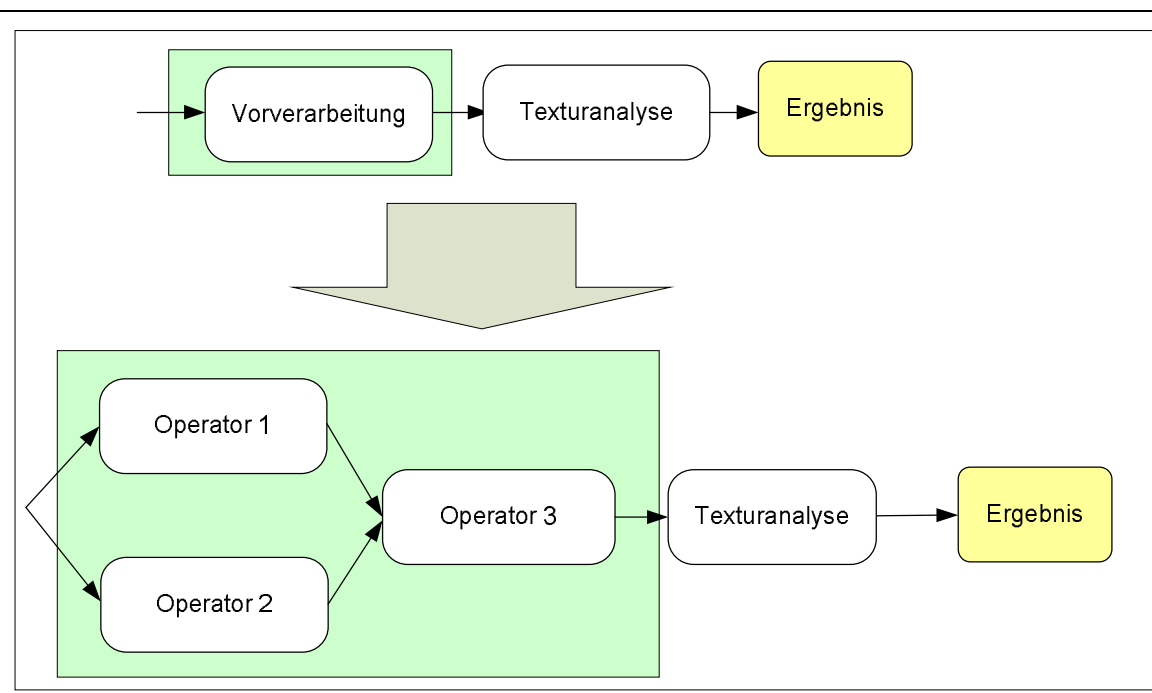


Abbildung 7-74: Texturklassifikation mit zusätzlicher komplexer Bildvorverarbeitung.

7.5.2 Ergebnisse und Diskussion

Gemäß der OUTEX-Internetpräsenz²⁶ liegt der derzeit höchste eingereichte Wert für die mittlere Klassifikationsrate (der 100 Testkonfigurationen) des Texturklassifikationsdatensatzes *TC00000* bei 99,5% bei Verwendung von speziellen Gaborfiltern. Mittels der Optimierung konnte eine mittlere Klassifikationsrate von 99,55% erreicht werden. Dieses Ergebnis wurde mittels eines Glättungsfilters, einer quadratischen Histogrammtransformation und eines

²⁶ <http://www.outex.oulu.fi> (letzter Zugriff: 30.11.2008)

Laplacefilters in Kombination mit AK ($p=q=13$) mit erzielt. In Abbildung 7-75 sind beispielhaft Eingangsbilder für den Texturanalyseoperator nach der Bildvorverarbeitung dargestellt.

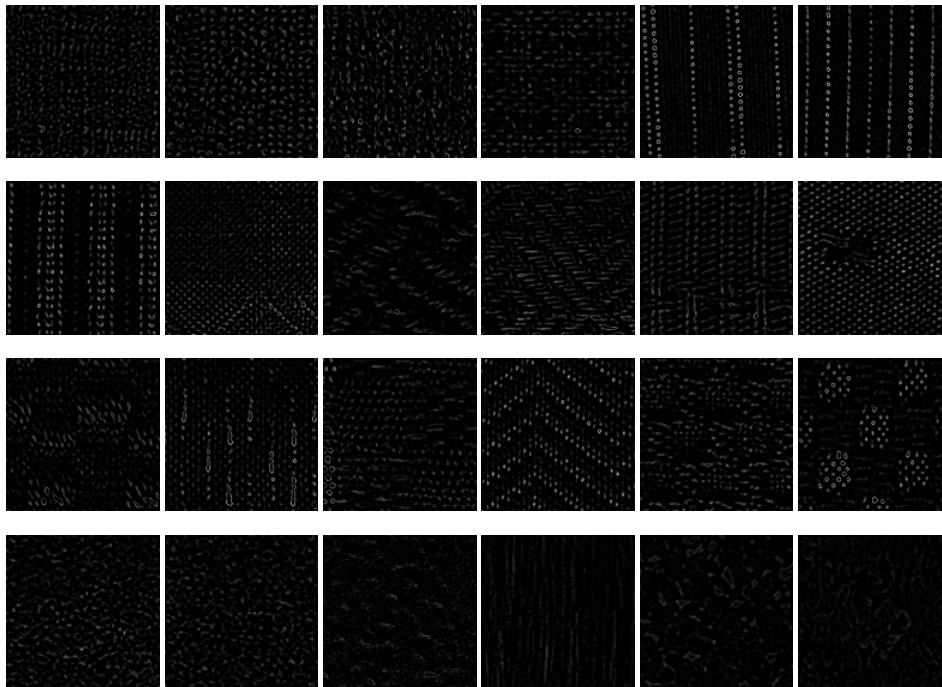


Abbildung 7-75: Texturklassen des OUTEX-TC0000 nach der Bildvorverarbeitung.

Bei den 24000 Klassifikationen gab es insgesamt 107 Fehlklassifikationen. Auffällig ist hierbei, dass sich die Fehlklassifikationen auf einige wenige Klassen beschränken. Allein Texturen der Klasse 16 wurden 44-mal der Klasse 17 zugeschrieben. Dies liegt wahrscheinlich an einer ähnlichen Periodizität der Hellen Texturpixel. Die Gesamtzahl der Fehlklassifikationen ist in Tabelle 7-22 aufgeführt. Die Zeile gibt die Soll-Klasse, die Spalte die durch die Klassifikation zugeordnete Ist-Klasse an. Wird der optimierte Texturoperator bei gleicher Konfiguration ohne Bildvorverarbeitung verwendet, ergibt sich eine mittlere Klassifikationsrate von 95,18% mit 1156 Fehlklassifikationen.

Bei einer wiederholten Optimierung von Vorverarbeitung und Texturoperator mittels GA ergaben sich unterschiedliche BV-Lösungen für die Bildvorverarbeitung. In drei von fünf Optimierungsläufen wurde die stochastische bzw. periodische Struktur innerhalb der Texturen durch die Bildverarbeitung hervorgehoben und AK als Texturoperator ausgewählt. Bei dem schlechtesten Trainingsergebnis nach der Optimierung der fünf Durchläufe wurden die Werte der Co-Occurrence-Matrix für die Klassifikation herangezogen. In diesem Fall liegt die mittlere Klassifikationsrate für alle 100 Testfälle mit ca. 97% aber immer noch über der Klassifikationsrate ohne zusätzliche Bildvorverarbeitung.

Tabelle 7-22. Fehlklassifikationen bei der OUTEX-TC00000 Texturklassifikation

Soll \ Ist	0	8	12	16	17	18	19	20	23	Σ
9	-	3	12	-	-	2	1	3	-	21
14	-	-	-	3	-	-	-	-	-	3
15	6	-	-	-	-	8	-	-	-	14
16	-	-	-	-	44	-	-	-	-	44
17	-	-	-	6	-	-	-	-	-	6
19	-	-	-	-	-	-	-	-	2	2
23	-	-	-	-	-	-	17	-	-	17
Σ	6	3	12	9	44	10	18	3	2	107

Die erreichte mittlere Klassifikationsrate zeigt, dass mittels Struktur- und Parameteroptimierung und konventionellen Standardmethoden der Bildverarbeitung ein Ergebnis erzielt werden kann, dass einem speziellen Algorithmus für die Texturklassifikation ebenbürtig ist.

Wird die Optimierung wiederholt, ergeben sich u.a. auch geringere Klassifikationsraten. Die Ergebnisse waren jedoch in allen Fällen einer Klassifikation ohne Bildvorverarbeitung überlegen.

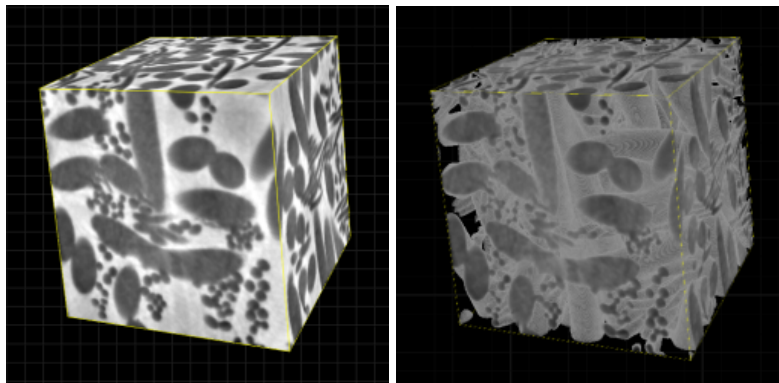


Abbildung 7-76: Computertomografie (CT)-Aufnahme eines Fasermaterials mit dünnen und dicken Fasern. Auflösung: 6.65 μm , 2563 Voxel. Rechts: Hervorgehobene Fasern.

7.6 Segmentierung von 3d-Volumendaten

Die Verarbeitung und Analyse von 3d-Volumendaten gewinnt zunehmend an Bedeutung. Aufgrund hoher Datenraten und begrenzter Rechenkapazität ergeben sich für die Bildverarbeitung jedoch lange Verarbeitungszeiten. Der Einsatz des Werkzeugs für den automatischen Entwurf und die Autokonfiguration wird im folgenden Experiment für die Segmentierung von Faserstrukturen verwendet. Zu diesem Zweck wurden grundlegende ikonische BV-Algorithmen, wie Filter-, Schwellwert- oder morphologische Operatoren, neben der Standard-Implementierung für zwei-dimensionale Daten für die Verarbeitung von 3d-

Volumendaten erweitert. Als Daten wurden computertomografische Aufnahmen von Fasermaterialien verwendet. Abbildung 7-76 zeigt eine Computertomografie (CT)-Aufnahme eines Fasermaterials mit dünnen und dicken Fasern, sowie eine grafische Hervorhebung der Faserdaten. Bildverarbeitungsgraphen erzeugt und folgende Parameter für eine GA/GP-Optimierung verwendet:

Tabelle 7-23. Parameter der Optimierung

	N_P	20 Individuen
	Anzahl der Knoten	10
	Maximale Anzahl von Generationen	20
	Abbruch der Optimierung beim Erreichen einer Fitness von:	96
GA/GP	Turnierselektion, Anzahl der Konkurrenten	2
	1-Punkt-Kreuzung, Kr	60
	Mutation, Mu (PM/MM/EAM)	100 (2/1/1)

Da bei den komplexen Bildverarbeitungsgraphen mehrere Ergebnisbilder erzeugt werden können, wurde jedes einzelne über einen Binärbildvergleich mit einem manuell erstellten Referenzbild mit der jeweiligen Segmentierung dünner bzw. dicker Fasern verglichen und bewertet. Als Gütemaß diente hierbei die Summe der Erkennungsraten für Objekt- und Hintergrundpixel.

Da ein ikonischer Bildverarbeitungsgraph nicht zwangsläufig ein Binärbild liefert, müssen die Ergebnisbilder ggf. für den Binärbildvergleich über einen Schwellwertoperator binarisiert werden. Da automatische Schwellwertverfahren nach Otsu, Pun und Rosenfeld [125] nicht die erwünschten Ergebnisse erbracht haben, wurde jedes Ergebnisbild zusätzlich anhand eines festen Sets von Grauwertschwellen T , $T=\{1, 32, 64, 96, 128, 224, 254\}$ binarisiert und jedes Binärbild für die Bewertung herangezogen. Die endgültige Güte eines BV-Graphen ergibt sich anhand des Maximums aller für diesen Graphen berechneten Gütewerte. Die Optimierung der Segmentierung der dünnen und dicken Fasern im Volumenbild wurde anhand von 2d-Volumenscheiben und 3d-Subvolumen durchgeführt. Als Trainingsbilder wurden hierfür Volumenausschnitte der Größe 128×128 (2d) bzw. $128 \times 128 \times 128$ verwendet. Die Generalisierungsfähigkeit der optimierten BV-Graphen wurde anhand von Testausschnitten mit doppelter Kantenlänge getestet. Da die Optimierung auf 3d-Volumendaten sehr zeitaufwendig ist und mehrere Stunden gedauert hat, wurde der Ergebnisgraph der 2d-Optimierung zusätzlich auf das 3d-Generalisierungsvolumen angewendet.

7.6.1 Ergebnisse der automatischen Segmentierung von dünnen und dicken Fasern

Die Optimierung der Segmentierung der dünnen und dicken Fasern im Volumenbild wurde anhand von 2d-Volumenscheiben und 3d-Subvolumen durchgeführt.

Bei typischen Optimierungsläufen wurden folgende Fitnesswerte für Training und Generalisierung für die Segmentierung von dünnen Fasern erreicht.

Tabelle 7-24. Erreichte Fitnesswerte bei der automatischen Segmentierung

	Training	Generalisierung
dünne Fasern, 2d-Optimierung	97,2	96,9
Anwendung des Ergebnisgraphen der 2d-Segmentierung auf das Volumenbild	---	92,8
dünne Fasern, 3d-Optimierung	96	93

Die Fitness bei der Anwendung der Ergebnisgraphen aus der 2d-Optimierung auf das Volumenbild ist nur unwesentlich schlechter als das Generalisierungsergebnis bei der 3d-Optimierung. Dies liegt hauptsächlich daran, dass sich die prinzipielle Bildstruktur der einzelnen Volumenscheiben in diesem Fall nur unwesentlich ändert.

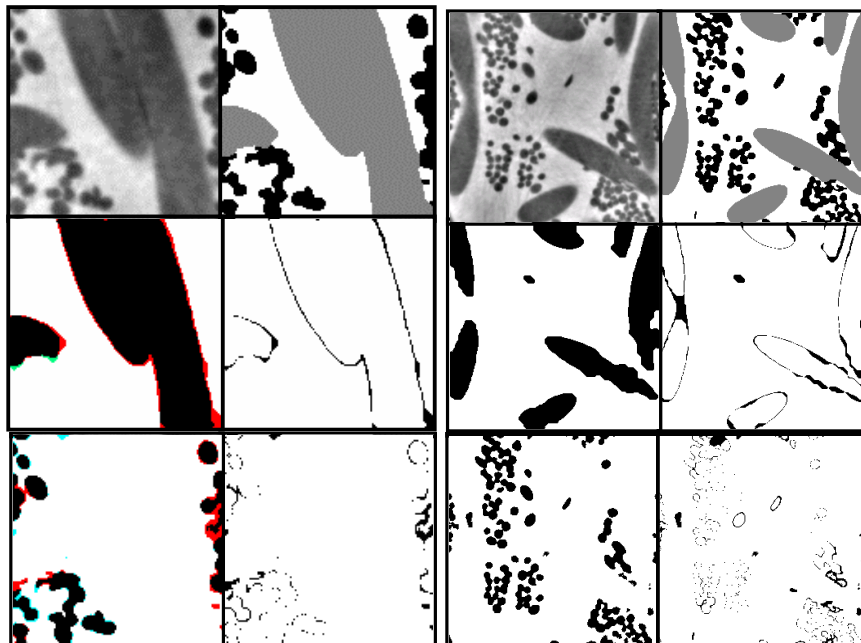


Abbildung 7-77: Manuelle und automatische Segmentierung der dünnen bzw. dicken Fasern. Links: Training, rechts: Generalisierung. Dargestellt sind jeweils das Originalbild, die manuelle Segmentierung der dünnen und dicken Fasern, das Ergebnis der automatischen Segmentierung und die absolute Differenz zwischen manueller und automatischer Segmentierung.

Abbildung 7-77 zeigt die beispielhaft die Ergebnisbilder der automatischen Segmentierung der dünnen bzw. dicken Fasern anhand der 2d-Volumenscheiben. Die linke Spalte zeigt die Segmentierungen des Trainingsbildes, die rechte Spalte zeigt die Generalisierungsfähigkeit. Dargestellt sind jeweils das Originalbild, die manuelle Segmentierung der dünnen und dicken Fasern, das Ergebnis der automatischen Segmentierung und die absolute Differenz zwischen manueller und automatischer Segmentierung.

Eine vollständige Übersicht über die Experimente mit Volumendaten ist in [104] zu finden.

7.7 Diskussion der Ergebnisse

Mittels PSO und GA/GP lassen sich bereits mit Standardmethoden der Bildvorverarbeitung und der Texturanalyse Systeme für die Segmentierung von texturierten Objekten und Hintergrundtextur oder die Texturklassifikation erzeugen und optimieren. Ist die Struktur des zu optimierenden Systems bekannt und soll nicht verändert werden, bietet sich für die automatische Optimierung PSO im Vergleich zur GA an. In der Regel werden mittels PSO schneller zufrieden stellende Lösungen generiert.

Bevor man zeitlich aufwendige Klassifikationssysteme (zumindest für das Training des Klassifikators) entwirft, kann man versuchen, das Problem mittels Standardverfahren der BV zu lösen oder eine bessere Ausgangslage für eine folgende Merkmalsextraktion und Klassifikation zu schaffen.

Nicht jeder Optimierungslauf führt zwangsläufig zu einem optimalen Ergebnis. In einigen Fällen wurden augenscheinlich gute Zwischenlösungen aufgrund des gewählten Fitnesskriteriums wieder verworfen. Im Fall der Verwendung von *SQG* können beispielsweise zufrieden stellende Binarisierungen im Laufe der Optimierung zugunsten von Graustufenbildern verworfen werden, bei denen die Binarisierung etwas kleiner als die Referenzmaske ist. Bei der Verwendung von *MI* als Ähnlichkeitsmaß und einem binären Sollbild werden bevorzugt homogene, d. h. nahezu einfarbige Ergebnisbilder generiert. Bei diesen ergibt sich in der Regel ein höherer Ähnlichkeitswert als bei Grauwertbildern. Gegebenenfalls kann in diesen Fällen ein grauwertiges Sollbild Abhilfe schaffen, bei dem die Detektion von Objekt- bzw. Defektzentren stärker und Objektränder geringer gewichtet werden (beispielsweise mittels eines Distanzbildes). Alternativ dazu kann ein erfahrener Systementwickler Zwischenlösungen einer Optimierung visuell begutachten und ggf. manuell zu verbessern.

Um für eine Problemstellung geeignete Lösungen generieren zu können, sollten die Optimierungsläufe nach Möglichkeit öfter wiederholt werden. Auch eine Optimierung mit verschiedenen Inseln kann sich unter Umständen anbieten.

Neben der Güte der angebotenen Lösungen spielt die Optimierungszeit eine entscheidende Rolle. Neben dem Ansatz, die vorhandenen Bilddaten optimal zum Entwurf einzusetzen, können Bestrafungselemente die Rechenzeit einer Lösung regulieren. Hierdurch reduziert sich nicht nur die benötigte Rechenzeit der eigentlichen BV-Lösung, sondern auch die Optimierungszeit.

Die, in der frühen Arbeit über eine automatische Segmentierung von Volumendaten erzielten Ergebnisse eröffnen zusätzlich Perspektiven für eine zeiteffiziente Optimierungsstrategie von multispektralen oder multidimensionalen Daten.

8 Diskussion und Zusammenfassung der Ergebnisse, Ausblick

In dieser Arbeit wurde auf Basis vorangegangener Arbeiten ein effizienteres Entwurfs- und Konfigurationswerkzeug für Bildverarbeitungssysteme mit neuen und verbesserten Eigenschaften geschaffen. In diesem Zusammenhang wurden aktuelle Einsatzgebiete und die Module eines Bildverarbeitungssystems für die industrielle Oberflächeninspektion analysiert und aktuelle Arbeiten bzw. Potenzial für eine automatische Konfiguration und Optimierung dieser Module aufgeführt. Basierend auf den bereits geschaffenen Grundlagen für die automatische bzw. anwenderunterstützte Entwicklung eines BV-Systems mit dem Programm QuickCog in den Bereichen der Automatisierung und Optimierung der Merkmalsauswahl, Klassifikation und effektiven Visualisierung der klassifizierten Merkmalsräume, wurden in dieser Arbeit insbesondere die Bereiche der Struktur- und Parameteroptimierung eines komplexen Bildverarbeitungsgraphen betrachtet. Aus dem Bedürfnis für eine schnelle, automatische Generierung von Bildverarbeitungssystemen wurden die Grundlagen, das Konzept und die Architektur einer Umgebung für den automatischen Entwurf der Struktur und der anschließenden Konfiguration von Bildverarbeitungssystemen entwickelt, in der Systeme für die überwachte Bildsegmentierung, -analyse und -klassifikation mittels bewährter Methoden der Bildvorverarbeitung und der Texturanalyse zum Einsatz kommen. Ein modulares Konzept erlaubt eine einfache Erweiterung der Umgebung hinsichtlich neuer Algorithmen, Klassifikatoren oder Datentypen. Das Einbringen von ebenfalls modularen Bewertungsmethoden erlaubt eine flexible datengestützte Optimierung des BV-Systems für verschiedene Aufgabenbereiche.

Für den Entwurf und die Konfiguration des Systems wurden Genetische Algorithmen, Genetische Programmierung und Partikelschwarmoptimierung eingesetzt. Insbesondere bei der Konfiguration von Systemen mit einer begrenzten Anzahl von Modulen und Parametern hat sich die kanonische PSO als besser als GA/GP erwiesen. Es werden schneller bessere Resultate erzielt als mit GA/GP. Wenn ein Nutzer der Umgebung für den Entwurf und die Konfiguration eine „Ein-Klick-Lösung“ gegenüber einer sinnvollen Variation der Optimierungsparameter bevorzugt, bietet sich die kanonische PSO aufgrund der geringen Konfigurationsmöglichkeiten im Vergleich zu GA/GP an. Allerdings ist die Struktur- und Datenrepräsentation bei Verwendung von GA/GP bei der Verwendung einer Vielzahl von möglichen BV-Modulen und daraus resultierend einer Vielzahl von möglichen Parametern für jede Position innerhalb einer BV-Kette oder eines BV-Graphen übersichtlicher, da für jedes Modul nur ein Parametersatz gespeichert werden muss.

Eine Einschränkung des Suchraums bezüglich der Methoden- und Parameterauswahl durch das Einbringen von Vorwissen führt in der Regel zu schnelleren Ergebnissen. Allerdings werden dabei ggf. ebenso erfolgreiche andere Lösungsansätze nicht berücksichtigt. Eine parallele Optimierung konkurrierender, auf Vorwissen basierender Verfahren und einer Suche im kompletten Suchraum kann hierbei ggf. helfen, Lösungen zu finden, die nicht in dem zur Verfügung stehenden Wissensspeicher abgedeckt sind.

Bei der Klassifikation hat sich die Verwendung von Support-Vektor-Maschinen gegenüber klassischer kNN-Klassifikation als vorteilhaft bezüglich der erreichten Klassifikationsergebnisse herausgestellt. Allerdings wird für das Training des Klassifikators unter Umständen sehr viel Speicherplatz und Rechenzeit benötigt. Eine sinnvolle Eingrenzung der Trainingsdatenmenge kann hier ggf. Abhilfe schaffen, bleibt jedoch zukünftiger Arbeit vorbehalten. Neben der Optimierung von Systemen für die 2d-Bildanalyse wurde eine Problemstellung aus dem Bereich der 3d-Bildsegmentierung untersucht.

Anhand der erreichten Verbesserung bei den betrachteten repräsentativen Beispielen ist eine Produktivitätssteigerung in industriellen BV-Lösungen zu extrapolieren. Das in dieser Arbeit geschaffene Werkzeug wird in der nächsten Phase in eine geeignete Oberfläche integriert und für einen komfortablen praxisnahen Einsatz für den Entwurf industrieller BV-Systeme nutzbar gemacht.

Die Generalisierungsfähigkeit und Stabilität der generierten Systeme ist noch ein offener Punkt für die Zukunft. In diesem Rahmen sollen u.a. die entworfenen Systeme auch auf multiple Instanzen abgebildet und Drifteffekte untersucht werden. Die Übertragung auf mehrere Instanzen und die dafür ggf. erforderlichen Systemanpassungen [30] werden derzeit – neben weiteren Forschungsaktivitäten im Bereich des automatischen Entwurfs, der Optimierung und der Autokonfiguration von Sensorsignalsystemen – an dem Lehrstuhl, an dem diese Arbeit entstanden ist, in einer zu dieser Arbeit parallelen laufenden Dissertation betrachtet.

Abkürzungen und Formelzeichen

Allgemeine Abkürzungen

2d, 3d, nd	zwei-, drei-, n -dimensional
nd -Bild	Bild der Dimension n
3GL	Programmiersprachen der 3. Generation
S	Soll- oder Referenzbild
B	Ist-Bild
BV	Bildverarbeitung
CT	Computertomografie
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
LED	<i>Light Emitting Diode</i> , Leuchtdiode
Pixel	<i>Picture Element</i> , Bildpunkt eines Bildes
ROI	<i>Region of Interest</i> , Arbeitsbereich innerhalb eines Bildes
TDE	Texturdefekterkennung
Texel	<i>Texture Element</i> , Kleinster sich wiederholender Bereich einer Textur
TSP	<i>Traveling-Salesman-Problem</i> , Problem des Handlungsreisenden
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V.
Voxel	<i>Volume Element</i> , Bildpunkt eines 3d-Bildes

Evolutionäre Algorithmen und Schwarmintelligenz

ACO	<i>Ant colony optimization</i> , Ameisenalgorithmus
11-PSO	Partikelschwarmoptimierung mit festem Trägheitsgewicht
C-PSO	Kanonische Partikelschwarmoptimierung
CV-PSO	Kanonische PSO mit Reinitialisierung der Geschwindigkeit
EA	Evolutionäre Algorithmen
EAM	Eingangs- und Ausgangsmutation
ES	Evolutionsstrategien
GA	Genetische Algorithmen
GP	Genetische Programmierung
MM	Methodenmutation
PM	Parametermutation
PSO	Partikelschwarmoptimierung

Bewertung und Klassifikation

CART	<i>Classification and Regression Trees</i> , Klassifikations- und Regressionsbäume
CV	<i>Cross Validation</i> , Kreuzvalidierung
kNN	k-Nächste-Nachbar-Klassifikator
NNK	Nächste-Nachbar-Klassifikator
RBF	Radiale Basisfunktion
SVM	Support-Vektor-Maschine

Implementierte Texturoperatoren und ihre Merkmale

AK	Autokorrelation:	
		Akk_{ff} Autokorrelationskoeffizienten
		$p_{max} \ q_{min}$ Maximale Verschiebung in x- bzw. y-Richtung
COOC	Histogrammeinträge der Co-Occurrence-Matrix	
HARA1	Fünf Haralick-Merkmale:	
		Con Kontrast der Co-Occurrence-Matrix
		Cor Korrelation
		Eng Energie der Co-Occurrence-Matrix
		Ent Entropie der Co-Occurrence-Matrix
		Hom Homogenität
HARA2	14 Haralick Merkmale	
HIST	Histogrammeinträge erster Ordnung	
HISTS	Merkmale des Histogramms erster Ordnung:	
		con Kontrast
		ent Entropie
		kur Wölbung oder Kurtosis
		sk Schiefe
		$qsum$ Quadratsumme
LTEM	Laws' Texture and Energie Measures, Laws' Textureenergien	
NLOK	Nichtlineare und orientierte Kerne	
NLOK-HIST	Histogramm der dominanten Kerne	
NLOK-COOC	Histogramm 2ter Ordnung der dominanten Kerne	
NLOK-HARA1	Fünf Haralick-Merkmale aus NLOK-COOC	
NLOK-HARA2	14 Haralick-Merkmale aus NLOK-COOC	

Weitere Texturoperatoren

LAC	<i>Local Autocorrelation</i> , lokale Autokorrelation
LBP	<i>Local Binary Pattern</i> , lokale Binärmuster
LOC	<i>Local Orientation Coding</i> , Kodierung der lokalen Orientierung

Allgemeine Formelzeichen

a, b	Halbachsen einer Ellipse
α	Winkel zwischen der Hauptachse einer Ellipse und der positiven x-Achse
n	Dimension
A	Steilheit einer Funktion
M, N	Breite und Höhe eines Bildes bzw. einer Bildregion
T	Tiefe eines 3d-Bildes bzw. einer Bildregion
Z	Anzahl der Bildpunkte in einer Bildregion der Größe $M \times N$
x, y, z	Position eines Pixels im Bild oder einer Bildregion
$\Delta x, \Delta y, \Delta z$	Abstand zweier Bildpunkte
p, q	Verschiebung in x - bzw. y -Richtung
d	Verschiebungsvektor
$d = [p, q]^T$	Verschiebungsvektor 2d
G	Anzahl der Grauwerte in einem Bild
g	Grauwert, $0 \leq g < G$
$g(x, y)$	Grauwert des Bildpunktes an der Position x, y
$P(g)$	relative Grauerthäufigkeit
$P_{ij} = P(g_i, g_j)$	relative Häufigkeit des Grauwertpaares g_i, g_j
$H(g)$	absolute Grauerthäufigkeit
$H(g_i, g_j)$	absolute Häufigkeit des Grauwertpaares g_i, g_j
Q	Anzahl der Grauwerte nach einer Grauwertquantelung
RAS	Rastrigrin-Funktion
SPH	Sphere-Funktion
GRI	Griewank-Funktion

Evolutionäre Algorithmen und Schwarmintelligenz

N_D	Diversität, Anzahl neuer Individuen
N_E	Elitismus, Anzahl der besten Individuen, die unverändert in die nächste Generation übernommen werden
N_P	Populationsgröße, d. h. Anzahl der Individuen in einer Generation
Mu	Mutationsrate bei GA und GP
Kr	Kreuzungsrate bei GA und GP
\mathbf{x}_i^t	Positionsvektor des i -ten Partikel im Suchraum zum Zeitpunkt t
\mathbf{v}_i^t	Geschwindigkeitsvektor des i -ten Partikel zum Zeitpunkt t
v_{max}	obere Grenze der Geschwindigkeit
\mathbf{lb}	Lokal beste Position im Suchraum, Positionsvektor
\mathbf{gb}	Global beste Position im Suchraum, Positionsvektor
ϕ_1	„Soziale“ Konstante PSO
ϕ_2	„Persönliche“ Konstante PSO
τ	Trägheitsgewicht PSO
χ	Begrenzungsfaktor PSO

Bewertung und Klassifikation

$s=s(x,y)$	Grauwert des Bildpunktes an der Position x, y des Sollbildes
$b=b(x,y)$	Grauwert des Bildpunktes an der Position x, y des Ist-Bildes
$VBB(s,b)$	Anzahl der übereinstimmenden Bildpunkte
$SQG(s,b)$	Summe der quadratischen Grauwertdifferenzen
$KLD()$	Kullback-Leibler-Distanz
$MI()$	<i>Mutual Information</i> . Gegenseitige Information.

\mathbf{m}	Merkmalsvektor
L	Anzahl der Klassen
ω_i	Klassenzugehörigkeit eines Merkmalsvektors, $1 \leq i \leq L$
\mathbf{r}_i	Prototyp oder Referenzmerkmalsvektor der Klasse ω_i
$kn(\mathbf{m}, \mathbf{m}')$	Kernfunktion der Support-Vektor-Maschine
C	Regulierungsfaktor SVM
ξ	Schlupfvariable SVM
R	Bildregion
R_{OK}	Klassifikationsrate korrekt klassifizierten Bildregionen R
R_i	Anzahl aller Regionen der Klasse ω_i
$R_{i,OK}$	Klassifikationsrate korrekt klassifizierten Bildregionen R der Klasse ω_i

Industrielle Bildverarbeitung: Software, Firmen und Messen

Frei verfügbare Programmierbibliotheken

ITK Insight Segmentation and Registration Toolkit,
<http://www.itk.org>

libSVM A Library for Support Vector Machines,
Chih-Chung Chang and Chih-Jen Lin,
<http://www.csie.ntu.edu.tw/~cjlin/libsvm>

OpenCV Open Computer Vision Library,
<http://sourceforge.net/projects/opencvlibrary>

Der letzte Zugriff auf die Internetseiten erfolgte jeweils am 30.11.2008.

Kommerzielle Bildverarbeitungssoftware

Common Vision Blocks Stemmer Imaging GmbH,
Minos, Manto <http://www.stemmer-imaging.de>

HDevelop, HALCON MVTec Software GmbH,
<http://www.mvtec.com>

Heurisko AEON Verlag & Studio Walter H. Dorn e.K.
<http://www.heurisko.de>

Khoros/Cantata Khoral Inc.,
<http://www.khoral.com>

NeuroCheck NeuroCheck GmbH
<http://www.neurocheck.de>

QuickCog Andreas König, Michael Eberhardt und Robert Wenzel,
<http://www.quickcog.de>
Vertrieb: PHYTEC Messtechnik GmbH
<http://www.phytec.de>

Der letzte Zugriff auf die Internetseiten erfolgte jeweils am 30.11.2008.

Ausgewählte Bildverarbeitungsfirmen

AEON Verlag & Studio Walter H. Dorn e.K., 63452 Hanau

Cognex Germany, 76131 Karlsruhe

MVTec Software GmbH, Neherstraße 1, 81675 München

NeuroCheck GmbH, Neckarstraße 76/1, 71686 Remseck

Stemmer Imaging GmbH, Gutenbergstraße 9-11, 82178 Puchheim

Ausgewählte Bildverarbeitungsmessen

Automatica, (2-jährl.), München, <http://www.automatica-muenchen.de>

Control (jährlich), Stuttgart, <http://www.messe-control.de>

Vision (jährlich), Stuttgart, <http://www.vision-messe.de>

Hannover Messe, Hannover, <http://www.hannovermesse.de>

Der letzte Zugriff auf die Internetseiten erfolgte jeweils am 30.11.2008.

Anhang

A1 Implementierung in das Programmpaket Genesis

Das im Rahmen dieser Arbeit entstandenen Programmpaket „Genesis – GENetic programming for Surface Inspection Systems“ umfasst zu dem jetzigen Zeitpunkt eine Entwicklungsumgebung für Visual C++ (Microsoft), in die sämtliche Ablaufs-, Bildverarbeitungs- und Optimierungs-Algorithmen in verschiedenen Unterprojekten integriert sind, ein dazu gehörendes Kommandozeilenprogramm ohne eigene Oberfläche für die Ausführung der Optimierung und eine interaktive Applikation mit Oberfläche für eine komfortable manuelle Segmentierung von 2d- und 3d-Bilddaten (vgl. Abbildung A1-1).

Hauptaufgabengebiete von Genesis sind die automatische Segmentierung von 2d- und 3d-Bilddaten, eine reine Texturklassifikation mit oder ohne Bildvorverarbeitung von 2d-Bilddaten sowie eine kombinierte Texturklassifikation und Segmentierung für die Erkennung von Defekten oder Strukturen in einem Bild. Die Bewertung der Ergebnisbilder findet mittels eines Bildähnlichkeitsmaßes (Filter-Methode, Ähnlichkeitsmaße gemäß Kapitel 4.2) oder anhand der Ergebnisse der Texturklassifikation (Wrapper-Methode) statt. Für die Klassifikation können Nächste-Nachbar-Klassifikatoren oder Support-Vektor-Maschinen eingesetzt werden.

In die Umgebung sind Fraunhofer ITWM-interne Bibliotheken für die Handhabung eines modularen Bildverarbeitungsgraphen und für die Verwendung verschiedener Datenstrukturen, wie beispielsweise einem speziellen Bilddatentyp für 2d- und 3d-Grauwertbilder eingebunden. Die zusätzlich erbrachte Eigenleistung an C++-Methoden umfasst circa 60.000 Zeilen Quellcode.

Im Folgenden wird kurz auf die Struktur und die Funktionalität des Programmpakets Genesis eingegangen. Das Kommandozeilenprogramm wird hierbei mit GenesisOpt (kurz für Genesis-Optimierung) bezeichnet, die interaktive Oberflächenapplikation trägt den Namen GenesisMSeg (Genesis-Manuelle-Segmentierung).

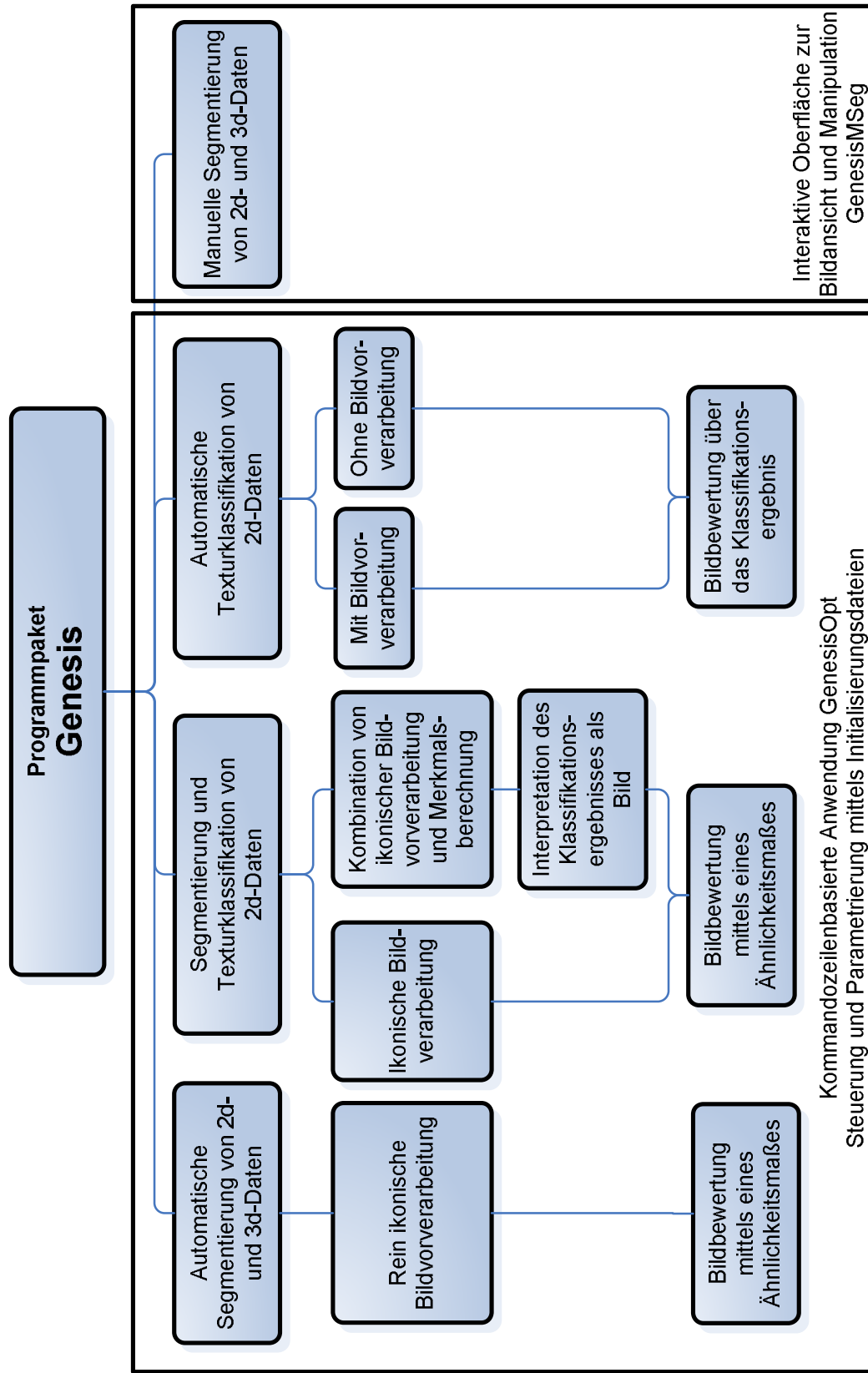


Abbildung A1-0: Aufteilung und Hauptaufgabengebiete des im Rahmen dieser Arbeit entstandenen Programmpaketes "Genesis".

A1.1 GenesisOpt

Mit der Kommandozeilen-basierten Applikation GenesisOpt lässt sich die Optimierung der überwachten Segmentierung von multispektralen oder multidimensionalen Daten mittels genetischer Algorithmen bzw. Schwarmintelligenz realisieren. Im Folgenden werden u.a. die verschiedenen implementierten Optimierungsstrategien mit oder ohne Verwendung von Vorwissen, die Optimierungsmethoden, die verwendeten BV-Methoden sowie einige Beispiele für die Initialisierung von GenesisOpt aufgeführt.

A1.1.1 Optimierung

Die prinzipiellen Möglichkeiten der Optimierung mit GenesisOpt gliedern sich in drei Bereiche, die Optimierung ohne zusätzliches Vorwissen, Optimierung mit Vorwissen und das Testen einer bestehenden Lösung an unbekanntem Daten (vgl. Abbildung A1-1).

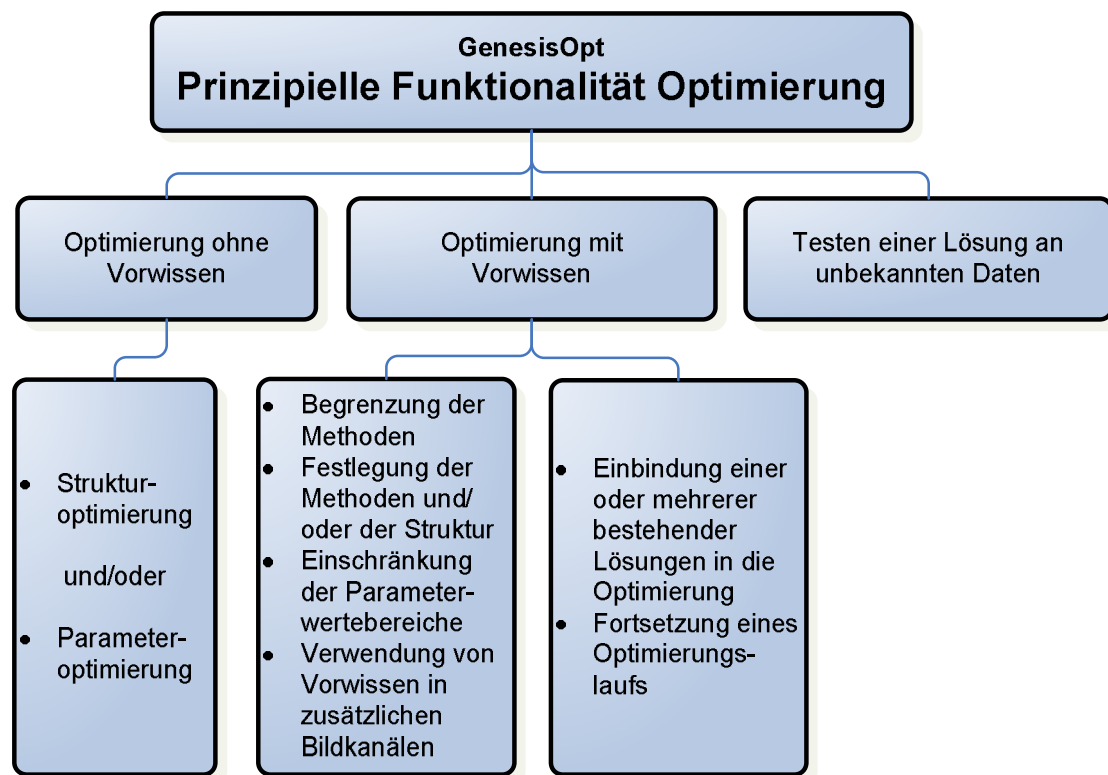


Abbildung A1-1: Prinzipielle Funktionalität des Kommandozeilenprogramms GenesisOpt bezüglich der Optimierung.

Bei der Optimierung ohne Vorwissen werden komplett zufällige Lösungen, d. h. Bildverarbeitungsgraphen, generiert. Die Parametrierung der einzelnen Module erfolgt innerhalb empirisch ermittelter, sinnvoller Parameterbereiche.

Je nach Festlegung der Optimierungsparameter kann es sich hierbei sowohl um Struktur- als auch um eine reine Parameteroptimierung handeln.

Für eine Beschränkung des Suchraums kann Vorwissen in die Optimierung eingebracht werden, beispielsweise über eine Unterauswahl der Bildverarbeitungsoperatoren, eine Variation der Parameterbereiche einzelner Methoden oder über die Festlegung gezielter Strukturen. Zusätzlich kann Vorwissen oder weitere Bildinformation in Form von zusätzlichen Bildkanälen verwendet werden. Hierbei kann es sich beispielsweise um Farbinformationen (r-, g-, b-Farbkanäle) oder um manuell segmentierte Bildregionen handeln. Bei der Generierung eines BV-Graphen können die BV-Methoden dann auf diese Bildkanäle zugreifen.

Neben der Auswahl von Bildkanälen, BV-Methoden und Parameterbereichen können einzelne Ergebnisse früherer Optimierungsläufe oder manuell erstellte BV-Graphen in die neue Optimierung einfließen oder eine bereits abgeschlossene Optimierung fortgesetzt werden.

Vorhandene Lösungen können zusätzlich an unbekanntem Daten getestet werden.

Abbildung A1-2 gibt einen Einblick in die in GenesisOpt implementierten Optimierungsmethoden gemäß Kapitel 5.

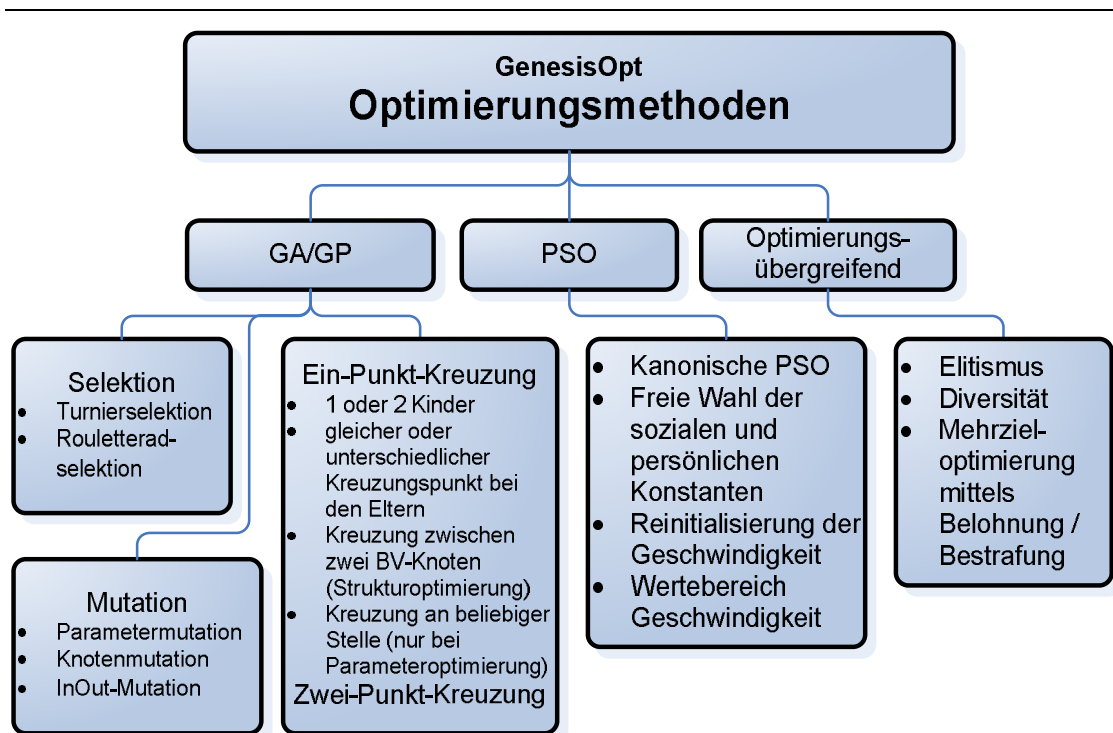


Abbildung A1-2: Optimierungsmethoden GenesisOpt.

A1.1.2 Verarbeitungsmethoden

Jede Methode die im Laufe der Optimierung optimiert werden soll, benötigt eine konsistente Beschreibung ihrer Eingangs- und Ausgangsparameter. Zu dieser Beschreibung gehören unter anderem der Name der Methode, eine Kurzbeschreibung der Funktionalität, die Anzahl der Übergabeparameter, ihre Namen und ihre Wertebereiche, sowie die Anzahl und Art der Eingangs- und Ausgangssignale. Diese Daten werden in einem sogenannten Map-Kontainer (vgl. Abschnitt Datenstrukturen) gespeichert und sind während des Systementwurfs, der Autokonfiguration und Optimierung verfügbar. Die prinzipielle Beschreibung einer Methode wird in Abbildung A1-3 für einen Glättungsfilter schematisch dargestellt.

<i>Name</i>	<i>FilterSmooth</i>
<i>Beschreibung</i>	<i>Mittelwertfilter zur Bildglättung</i>
<i>Typ</i>	<i>Bildverarbeitung</i>
<i>Anzahl der benötigten Eingangsparameter</i>	<i>2</i>
<i>Eingangsparameter 1</i>	<i>Name: SizeX Beschreibung: Filtergröße in x-Richtung Typ: Long Kleinsten Wert: 3, maximaler Wert: 25 Schrittweite: 2, Default-Wert: 3</i>
<i>Eingangsparameter 2</i>	<i>Name: SizeY Beschreibung: Filtergröße in y-Richtung Typ: Long Minimaler Wert: 3, maximaler Wert: 25 Schrittweite: 2, Default-Wert: 3</i>
<i>Anzahl der benötigten Eingangssignale</i>	<i>1</i>
<i>Eingangssignal 1</i>	<i>Typ: Bild Dimension: 2d oder 3d Bittiefe: 8-bit, 16-bit oder 32-bit</i>
<i>Anzahl der Ausgangssignale</i>	<i>1</i>
<i>Ausgangssignal 1</i>	<i>Typ: Bild Bildtyp, Dimension und Bittiefe entsprechen dem Eingangssignal</i>

Abbildung A1-3: Beispielhafte Beschreibung der Methodenparameter eines Mittelwertfilters.

Abbildung A1-4 zeigt einen Teil der in GenesisOpt verfügbaren Methoden der ikonischen Bildverarbeitung und Merkmalsberechnung. Neben den hier dargestellten Methoden werden u.a. adaptive Morphologiemethoden mit einem zusätzlichen Maskierungsbild verwendet. Zum jetzigen Zeitpunkt sind die Bildverarbeitungsalgorithmen auf eine Verarbeitung von ganzzahligen 8-bit Grauwertbildern begrenzt. Viele der ikonischen BV-Algorithmen laufen

sowohl auf 2d- als auch auf 3d-Bilder. Die Texturmerkmale können zum jetzigen Zeitpunkt nur aus 2d-Bildern berechnet werden.

Die wählbaren Parameter der ikonischen Bildverarbeitungsmethoden sind beispielsweise die Größe und Art des Strukturelementes bei der Morphologie, Schwellwerte und Transformationsarten (z. B. linear) der dazugehörigen Grauwertbereiche bei den Schwellwertoperatoren sowie Filtergrößen und Filtermasken bei den Filteroperatoren. Die Konfigurationsparameter der Methoden zu Texturmerkmalsberechnung sind in Tabelle 3-3 in Kapitel 3.8 aufgeführt.

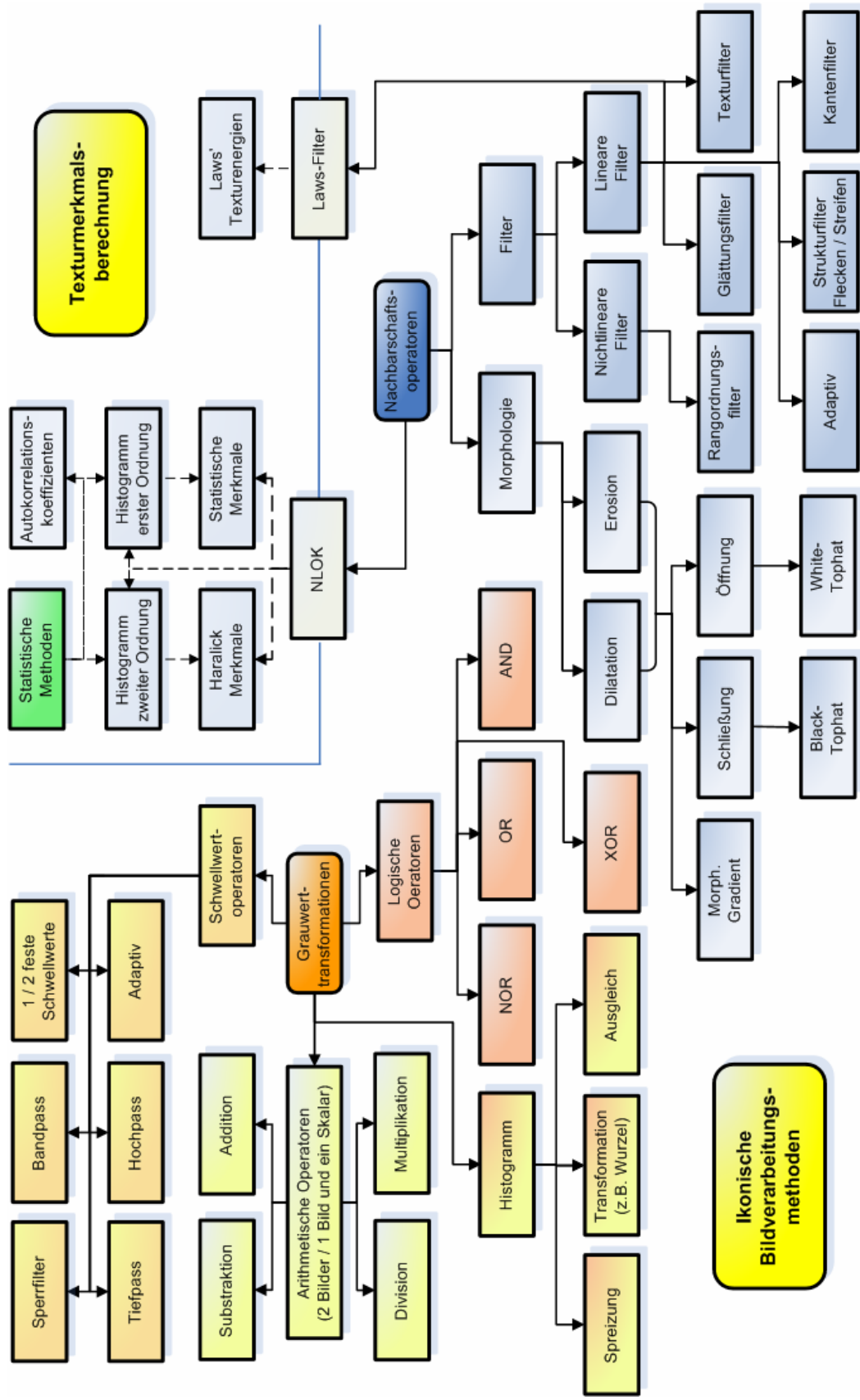


Abbildung A1-4: Implementierte Methoden für die ikonische Bildvorverarbeitung und die Berechnung von Texturmerkmalen. Gestrichelte Pfeile weisen auf die Transformation vom Bildraum in den Merkmalsraum hin.

A1.1.3 Datenstrukturen

Um einen reibungslosen Datentransfer zwischen den einzelnen Verarbeitungsalgorithmen zu gewährleisten, wird ein temporärer globaler Datenspeicher für die Optimierung verwendet, der viele verschiedene Datenstrukturen beinhalten kann. Die verschiedenen Strukturen sind in Abbildung 1-5 aufgeführt.

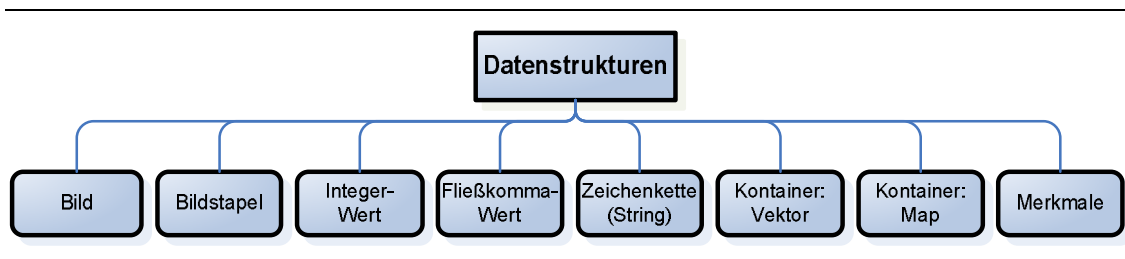


Abbildung A1-5: Verwendete Datenstrukturen in GenesisOpt.

Bei dem Bild handelt es sich um eine 2d-Flächen- oder 3d-Volumenmatrix der Größe $M \times N$ bzw. $M \times N \times T$. Die Bildintensitäten sind hierbei als ganzzahlige 8bit-, 16bit- oder 32bit- Werte gespeichert bzw. als 32bit-Fließkommazahlen.

Ein Bildstapel besteht aus mehreren Bildern oder Bildregionen. Zu jedem Bild werden zusätzliche Informationen, wie die Bildgröße und bei einer Region deren Position und die Identität des Ursprungsbildes gespeichert.

Bei dem Integer-Wert handelt es sich um eine ganzzahlige 32bit-Zahl, bei dem Fließkomma-Wert um eine 32bit-Fließkommazahl. Neben Zahlen können auch Zeichenketten verwendet werden.

Der Vektor-Kontainer kann die Datentypen Integer-Wert, Fließkomma-Wert, Zeichenkette, Vektor und Map beinhalten. Die einzelnen Elemente werden sequenziell ohne zusätzliche Benennung gespeichert. Der Map-Kontainer kann dieselben Datenstrukturen wie der Vektor beinhalten, die einzelnen Elemente werden jedoch über einen eigenen Namen angesprochen und gespeichert.

Die Bildmerkmale werden in einer gesonderten Datenstruktur gespeichert. Neben den eigentlichen Merkmalen gehören hierzu Informationen über das dazugehörige Bild, die Bildregion und die Klassenzugehörigkeit.

A1.1.4 Parametrierung der Optimierung

Die Struktur- bzw. Parameteroptimierung mit GenesisOpt wird über Initialisierungsdateien parametrisiert. Die Form entspricht xml-Dateien, in denen Parameter der oben aufgeführten Datenstrukturen mit den Ausnahmen Bild, Bildstapel und Bildmerkmale direkt, d. h. mit Datentyp, an die Optimierung übergeben werden können.

Für eine bessere Übersicht werden die Initialisierungsdateien hierfür in verschiedene Unterdokumente unterteilt:

- „Init“ – Allgemeine Optimierungsparameter, wie beispielsweise die Anzahl der Individuen oder die Anzahl der temporären Bildspeicher
- „Data“ – Eingabe der Trainings-, Test- und Generalisierungsdaten
- „Evolution“ – Optimierungsmethode (GA/PSO) und Optimierungsparameter
- „Evaluation“ – Bewertungsmethode für einen einzelnen Graphen
- „Save“ – Speicherung von Zwischen- und Endergebnissen

A1.1.5 Anwendungsbeispiele für die Initialisierung von GenesisOpt

Die folgenden Ausschnitte aus den Initialisierungsdateien zeigen beispielhaft, wie GenesisOpt parametrisiert wird. Die entsprechenden Variablennamen und die Kurzdokumentation sind hierbei in englischer Sprache verfasst.

Ein Variableneintrag beginnt stets mit dem Datentyp, beinhaltet den Variablennamen und endet erneut mit dem jeweiligen Datentyp. Kommentare werden folgendermaßen gekennzeichnet:
 <!-- Kommentar -->.

Initialisierung der allgemeinen Parameter der Optimierung (Init.xml)

Initialisierung der Populationsgröße (20), Elitismus (3), Diversität (3), Anzahl der Knoten pro BV-Graph (10) und die maximale Anzahl der ausgeführten Generationen (30):

```
<long id="NumberOfGraphs">20</long>
<long id="NumberOfBestGraphs">3</long> <!-- Elitism -->
<long id="NumberOfNewGraphs">3</long> <!-- Diversity -->
<long id="NumberOfNodes">10</long> <!-- Image processing nodes -->
<long id="NumberOfGenerations">30</long> <!-- max. number of generations -->
```

Initialisierung der Optimierungsmethode und deren Parameter (Evolution.xml)

GA/GP mit Turnierselektion (2 Teilnehmer), Kreuzung mit Unterschiedlichen Kreuzungspunkten bei den Eltern (Kreuzungsrate 70%) und Mutation (70%, Verhältnis Parametermutation:InOutMutation:Knotenmutation = 2:1:1)

```

<!-- _____ Help Info Evolution _____
EvolutionType
0: eGP
100: ePSO
_____ -->

<long id="EvolutionType">0</long>

<!-- _____ Help Info Evolution GP _____
EvolutionType (eGP      = 0):
SelectionType:
    eNoSelection      = 0,
    eRandom           = 1,
    eTournament       = 2
Tournament:
    Number of Parents
CrossOverType:
    eCloning          = 0,
    eSinglePointFixedLengthOneChild = 1,
    eSinglePointVariousLengthOneChild = 2,
    eParamOneChild   = 3
If CrossOverType == eParamOneChild: PMutationParam = 100!
PCrossOver:          0 - 100
PMutation:           0 - 100
PMutationParam + PMutationNode + PMutationInOut      = 100
Modus:
    1: Reduce graph to best algorithm chain
Should only be used with CrossOverType = eSinglePointVariousLengthOneChild!
_____ -->

<long id="Modus">0</long>
<long id="SelectionType">2</long>
<long id="Tournament">2</long>
<long id="CrossOverType">2</long>
<long id="PCrossOver">70</long>
<long id="PMutation">70</long>
<long id="PMutationParam">50</long>
<long id="PMutationInOut">25</long>
<long id="PMutationNode">25</long>

```

Laden der Bilddaten (Data.xml)

Laden der Originalbilder (Bildserie) für Training und Test der Optimierung (Data_Ref), der dazugehörigen manuellen Segmentierungen/Klasseninformationen (Data_Ref_Ref) bzw. der Generalisierungsbilder (Data_Test und Data_Test_Ref).

Angegeben wird jeweils der Pfad und Dateiname der Bilddaten. Ein großes „X“ ist ein Platzhalter für eine in dem Bildnamen integrierte Bildnummer, eine Ziffer vor dem „X“ gibt optional die Anzahl der Stellen der Bildnummer mit führenden Nullen an. Die Integer (long)-Werte in dem Vektor führen zu der Verwendung der Bilder Texture_set1_c00.pgm, Texture_set1_c01.pgm, und so weiter. Bei mehrkanaligen Bildern oder der Verwendung von Vorwissen in Form zusätzlicher Eingangsbilder werden mehrere Vektoren angegeben. Pro Mehrkanalbild wird jedoch nur eine manuelle Segmentierung/Klasseninformation verwendet:

```

<string id="Data_Type_Ref">Image_Series</string>
<vector id="Data_Ref">
  <!-- Channel 1 -->
  <vector>
    <string>..\..\images\Textures\Set1\Texture_set1_c2X.pgm</string>
    <vec_long>
      0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
    </vec_long>
  </vector>
  <!-- Channel 2 -->
  <vector>
    <string>..\..\images\Textures\Set1\Texture_set2_c2X.pgm</string>
    <vec_long>
      0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
    </vec_long>
  </vector>
</vector>

<string id="Data_Type_Ref_Ref">Image_Series</string>
<vector id="Data_Ref_Ref">
  <vector>
    <string>..\..\images\Textures\class\Texture_class_cX.pgm</string>
    <vec_long>
      0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
    </vec_long>
  </vector>
</vector>

```

SVM-Klassifikator und Kreuzvalidierung für die Bewertung (Evaluation.xml)

Initialisierung der Verwendung der C-SVM für die Bewertung und eine 10-fach Kreuzvalidierung. Die Bildregionen verschiedener Eingangsbilder werden für die Validierung zufällig gezogen:

```

<!-- Classifier Information -->
<string id="Classifier">SVM</string>
<string id="Plugin">FeatureComputationPlugind::ComputeFitnessFeature
</string>
<!-- _____ Help Info _____
CrossValidation:
< 0:          no crossvalidation
              Divide data into CrossValidation percent
              training data and 100-CrossValidation
              percent test data,
              e.g. -50: 50% train, 50% test
0:           no crossvalidation
              Feature set is already divided in
              training and test sets (this is for example
              internally used for generalization tests
> 0:        CrossValidation-crossvalidation
              Normalization of complete data set to [0-1]
              Division into n-datasets (last m datasets
              may have -1 members). Computation of the mean values.
CrossvalidationROI
0:           All ROIs of one image are in one data set

```

```
1:          ROIs are randomly (independent of origin image) divided into
          datasets
```

```
-->
```

```
<!-- Data Handling -->
<long id="CrossValidation">10</long>
<long id="CrossValidationRoi">1</long>
```

Bestrafung bzw. Belohnung (Evaluation.xml)

Initialisierung der Verwendung eines akkumulativen Bestrafungselements, wenn die Verarbeitungszeit eines BV-Graphen 0,01 ms überschreitet. In diesem Fall werden zwei Prozent der maximal erreichbaren Fitness von dem jeweiligen Fitnesswert abgezogen:

```
<!-- Penalty or Reward -->
<!-- _____ Help Info Penalty _____
      Param: ComputationTime, NumberOfNodes
      Condition:
      Lower than = -1
      Equal      = 0
      Greater than = 1
      ConditionArg (Depending on the chosen Param):
      e.g. 0.02 msec ComputationTime = 0.02
      Penalty:                                          0
- 1 _____
-->
<long id="FactorPenalty">1</long>
<map id="Penalty">
  <string id="Param">ComputationTime</string>
  <long id="Condition">1</long>
  <double id="ConditionArg">0.01</double>
  <double id="Penalty">0.02</double>
</map>
```

A1.1.6 Grafische Oberfläche für GenesisOpt

Zurzeit wird ein Programm mit grafischer Oberfläche für die Initialisierung und Ausführung der Optimierung entwickelt. Dieses soll dann sowohl als Einzellösung mit dem in dieser Arbeit beschriebenen Funktionsumfang nutzbar sein, zusätzlich jedoch auch in die interne Bildverarbeitungssoftware des Fraunhofer ITWM integriert werden, mit dem Ziel, beliebige dort erstellte Bildverarbeitungsgraphen zu optimieren.

A1.2 GenesisMSeg

Neben der Entwicklung der Optimierungsumgebung entstand während dieser Arbeit zusätzlich ein interaktives Werkzeug für die Bildsegmentierung, mit dem Ziel, möglichst einfach und komfortabel sowohl 2d- als auch 3d-Referenzbilder für die Optimierung schaffen zu können. Neben herkömmlichen Malwerkzeugen können hierbei automatische Segmentierungsmethoden, wie Regionenwachstums- oder Schwellwertoperatoren für die Generierung der segmentierten Ergebnisbilder verwendet werden. Die so erstellte Klasseninformation wird in

einer halbtransparenten Farbschicht über das Originalbild gelegt. Für eine Zwei-Klassen-Segmentierung entspricht die Farbe Grün der Klasse „Wahr“ (bzw. dem Grauwert 255 im Ergebnisbild), die Farbe Rot der Klasse „Falsch“ (Grauwert 0). Insgesamt können bis zu 10 Klassen farblich markiert werden. Nicht markierte Bereiche werden mit einem gesonderten Grauton gespeichert. Bereits erstellte Segmentierungen können zusammen mit dem Originalbild zu einem späteren Zeitpunkt erneut in das Programm geladen und verändert werden. Für die gezielte Manipulation einzelner Bildpunkte wird ein variabler Bildausschnitt zusätzlich vergrößert dargestellt.

Die folgenden Abbildungen zeigen drei Ansichten des Programms GenesisMSeg: die Menükarte mit Bildvorschau für das Laden der 2d- oder 3d-Bilder (Abbildung A1-6), eine Ansicht der Segmentierung einer Deckenplatte mit einem Kreis-Malwerkzeug und eine unvollständige Beispielsegmentierung mit dem Regionenwachstumsoperator (Abbildung A1-7).

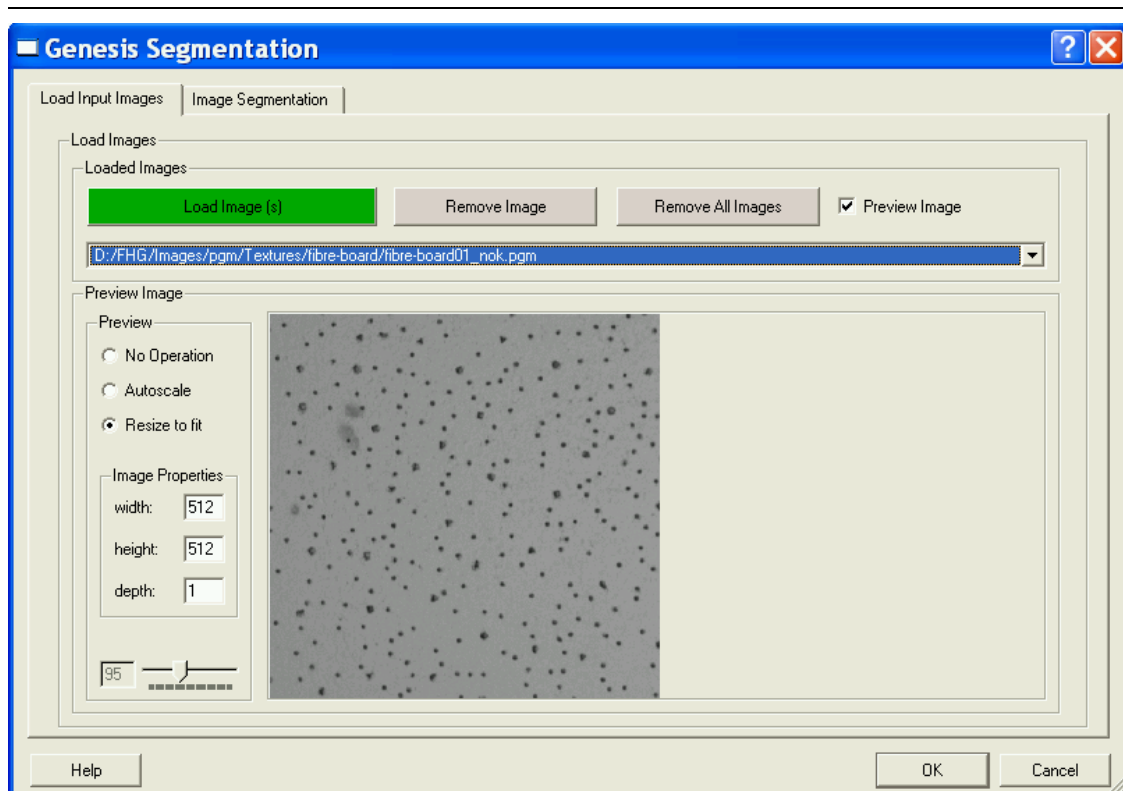


Abbildung A1-6: Menükarte des Programms GenesisMSeg mit Bildvorschau für das Laden von 2d- oder 3d-Bildern .

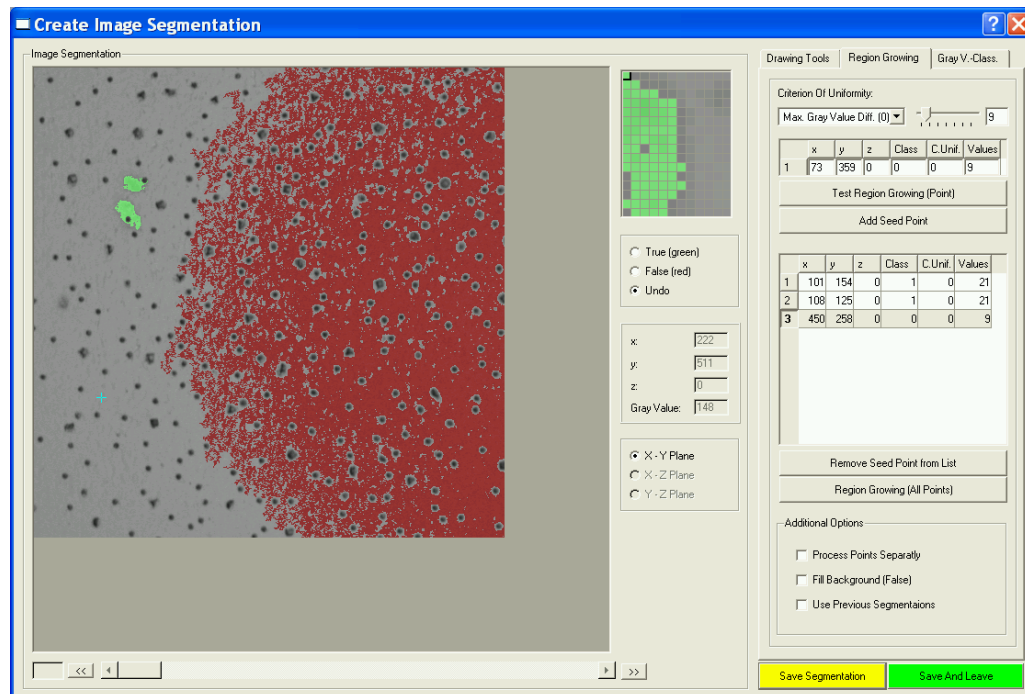
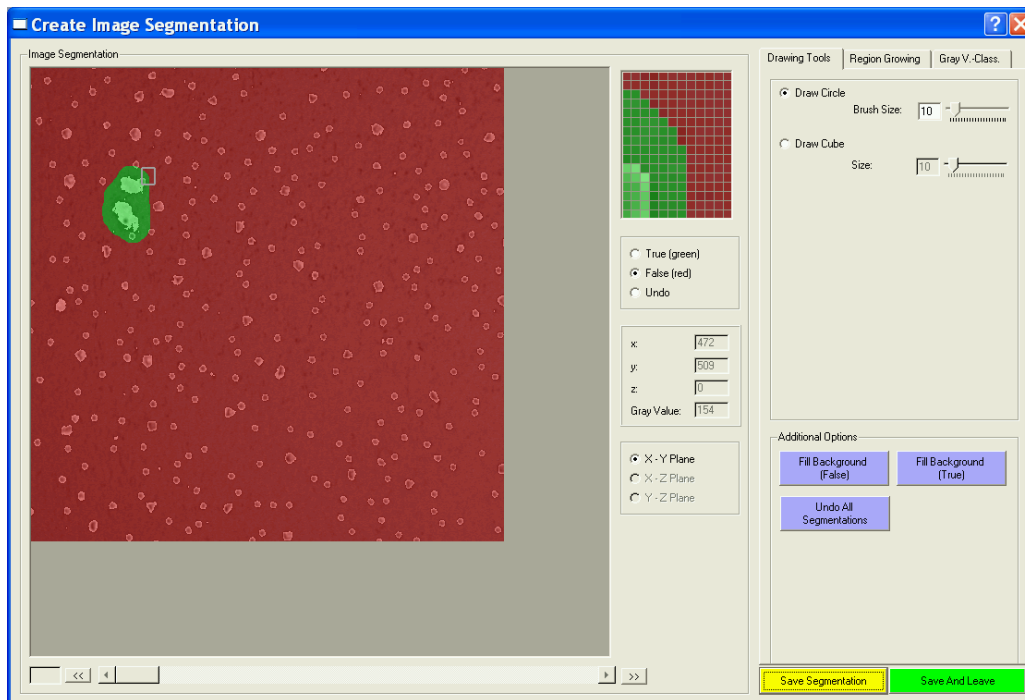


Abbildung A1-7: Oben: Manuelle Segmentierung einer Deckenplatte mit dem Kreis-Malwerkzeug des Programms GenesisMSeg. Unten: Beispielhafte halbautomatische Segmentierung mit dem Regionenwachstumsoperator.

A2 Laws' Filtermasken der Größe 5×5

$L5L5=L5^T*L5$					$L5E5=L5^T*E5$					$L5S5=L5^T*S5$					$L5R=L5^T*W5$					$L5=L5^T*R5$				
1	4	6	4	1	-1	-2	0	2	1	-1	0	2	0	-1	-1	2	0	-2	1	1	-4	6	-4	1
4	16	24	16	4	-4	-8	0	8	4	-4	0	8	0	-4	-4	8	0	-8	4	4	-16	24	-16	4
6	24	36	24	6	-6	-12	0	12	6	-6	0	12	0	-6	-6	12	0	-12	6	6	-24	36	-24	6
4	16	24	16	4	-4	-8	0	8	4	-4	0	8	0	-4	-4	8	0	-8	4	4	-16	24	-16	4
1	4	6	4	1	-1	-2	0	2	1	-1	0	2	0	-1	-1	2	0	-2	1	1	-4	6	-4	1
$E5L5$					$E5E5$					$E5S5$					$E5W5$					$E5R5$				
-1	-4	-6	-4	-1	1	2	0	-2	-1	1	0	-2	0	1	1	-2	0	2	-1	-1	4	-6	4	-1
-2	-8	-12	-8	-2	2	4	0	-4	-2	2	0	-4	0	2	2	-4	0	4	-2	-2	8	-12	8	-2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	8	12	8	2	-2	-4	0	4	2	-2	0	4	0	-2	-2	4	0	-4	2	2	-8	12	-8	2
1	4	6	4	1	-1	-2	0	2	1	-1	0	2	0	-1	-1	2	0	-2	1	1	-4	6	-4	1
$S5L5$					$S5E5$					$S5S5$					$S5W5$					$S5R5$				
-1	-4	-6	-4	-1	1	2	0	-2	-1	1	0	-2	0	1	1	-2	0	2	-1	-1	4	-6	4	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	8	12	8	2	-2	-4	0	4	2	-2	0	4	0	-2	-2	4	0	-4	2	2	-8	12	-8	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	-4	-6	-4	-1	1	2	0	-2	-1	1	0	-2	0	1	1	-2	0	-2	1	1	4	-6	4	-1
$W5L5$					$W5E5$					$W5S5$					$W5W5$					$W5R5$				
-1	-4	-6	-4	-1	1	2	0	-2	-1	1	0	-2	0	1	1	-2	0	2	-1	-1	4	-6	4	-1
2	8	12	8	2	-2	-4	0	4	2	-2	0	4	0	-2	-2	4	0	-4	2	2	-8	12	-8	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-2	-8	-12	-8	-2	2	4	0	-4	-2	2	0	-4	0	2	2	-4	0	4	-2	-2	8	-12	8	-2
1	4	6	4	1	-1	-2	0	2	1	-1	0	2	0	-1	-1	2	0	-2	1	1	-4	6	-4	1
$R5L5$					$R5E5$					$R5S5$					$R5W5$					$R5R5$				
1	4	6	4	1	-1	-2	0	2	1	-1	0	2	0	-1	-1	2	0	-2	1	1	-4	6	-4	1
-4	-16	-24	-16	-4	4	8	0	-8	-4	4	0	-8	0	4	4	-8	0	8	-4	-4	16	-24	16	-4
6	24	36	24	6	-6	-12	0	12	6	-6	0	12	0	-6	-6	-12	0	12	6	6	-24	36	-24	6
-4	-16	-24	-16	-4	4	8	0	-8	-4	4	0	-8	0	4	4	-8	0	8	-4	-4	16	-24	16	-4
1	4	6	4	1	-1	-2	0	2	1	1	0	2	0	1	-1	2	0	-2	1	1	-4	6	-4	1

Abbildung A2-1 Laws' Filtermasken der Größe 5×5.

Literaturverzeichnis

- [1] Abouelela, A.; Abbas, H.M.; Eldeeb, H.; Wahdan, A.A.; Nassar, S.M.: Automated vision system for localizing structural defects in textile fabrics. In: Pattern Recognition Letters, Vol. 26, S. 1435-1443, 2005.
- [2] Ahmadian, A.; Mostafa, A.; Abolhassani, M.D.; Salimpour, Y.: A texture classification method for diffused liver diseases using Gabor wavelets. In: Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society, IEEE-EMBS 2005, Shanghai, China, 2005, S. 1567-1570.
- [3] Ballerini, L.; Bocchi, L.; Johansson, C.B.: Image segmentation by a genetic fuzzy c-means algorithm using color and spatial information. In: Proceedings of the EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoISAP, EvoMUSART, and EvoSTOC, Coimbra, Portugal, S. 260-269, 2004.
- [4] Ballerini, L.; Franzen, L.: Genetic optimization of morphological filters with applications in breast cancer detection. In: Proceedings of the EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoISAP, EvoMUSART, and EvoSTOC, Coimbra, Portugal, S. 250-259, 2004.
- [5] Bartels, M.: Vorlesungsskript: Grundwissen der Elektrotechnik. Kapitel 10: Vergleich CMOS - CCD, SS07, Universität Kassel, 2007
- [6] Bauer, N. (Hrsg.): Handbuch zur Industriellen Bildverarbeitung: Qualitätssicherung in der Praxis, Fraunhofer IRB-Verlag, Stuttgart, 978-3-8167-7386-3, 2007.
- [7] Bishop, C.M.: Pattern recognition and machine learning. Springer, New York, ISBN 9780387310732, 2006.
- [8] Blackwell, T.; Branke, J.: Multi-swarm optimization in dynamic environments. In: Proceedings of the EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoISAP, EvoMUSART, and EvoSTOC, Coimbra, Portugal, S. 489-500, 2004.
- [9] Boer, B.: Classifier systems. A useful approach to machine learning? M.Sc. Thesis, Universität Leiden, 1994.
- [10] Bolle, R.M.; Jain, A.K.; Karu, K.: Is there any texture in the image? In: Pattern Recognition, Vol. 29, Nr. 9, S. 1437-1446, 1996.
- [11] Bresch, M.: Optimizing filter banks for supervised texture recognition. In: Pattern Recognition Letters, Vol. 35, S. 783-790, 2002.
- [12] Brinkmann, J.: Effizientes Lösen komplexer Probleme durch natürlich inspirierte Schwarm-Algorithmen, Studienarbeit, Universität Paderborn, 2004.
- [13] Cagnoni, S.; Lutton, E.; Gustavo, G. (Hrsg.): Genetic and evolutionary computation for image processing and analysis, EURASIP Book Series on Signal Processing and Communications, Hindawi Publishing Corporation, ISBN 9789774540011, Vol. 8, 2007.
- [14] Chan, C.-C.; Lin, C.-J.: LIBSVM: A library for support vector machines, 2001, Software: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, (letzter Zugriff: 30.11.2008).
- [15] Chen, C.-C.; Chen, C.-C.: Texture synthesis: A review and experiments. In: Journal of Information Science and Engineering, Vol. 19, S. 371-380, 2003.

- [16] Chen, C.-C.; Huang, C.L.: Markov random fields for texture classification. In: Pattern Recognition Letters, Vol. 14, S. 907-914, 1993.
- [17] Chetverikov, D.; Hanbury, A.: Finding defects in texture using regularity and local orientation. In: Pattern Recognition, Vol. 35, Nr. 10, 2002, S. 2165-2180.
- [18] Chetverikov, D.: Texture imperfections. In: Pattern Recognition Letters, Vol. 6, S. 45-50, 1987.
- [19] Chronis, G.: Vorlesungsfolien Herbst 2003: Evolutionary computation. An introductory computation, <http://www.cs.missouri.edu/~skubicm/375/GA1.pdf> (letzter Zugriff: 30.11.2008)
- [20] Clausi, D.A.; Jernigan, M.E.: A fast method to determine co-occurrence texture features. In: IEEE Transactions on Geosciences and Remote Sensing, Vol. 36, Nr. 1, S. 298-300, 1998.
- [21] Clerc, M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation, CEC 1999, Vol. 3, S. 1951-1957, 1999.
- [22] Cohen, F.S.; Fan, Z.: Automated inspection of textile fabrics using textural models, In: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, S. 803-808, 1991.
- [23] Copper, C.; Sloper, C.; Moscato, P.: Evolutionary search of thresholds for robust feature set selection: application to the analysis of microarray data. In: Proceedings of the EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoISAP, EvoMUSART, and EvoSTOC, Shanghai, China, S. 21-30, 2004..
- [24] Crosier, M.; Griffin, L.D.: Texture classification with a dictionary of basic image features, In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, CVPR 2008, Anchorage, Alaska, S. 1-7, 2008.
- [25] Dash, M.; Liu, H.: Feature selection for classification. In: Intelligent Data Analysis, Vol. 1, S. 131-156, 1997.
- [26] De Ves, E.; Benavent, X.; Ayala, G.; Domingo, J.: Selecting the structuring element for morphological texture classification. In: Pattern Analysis and Applications, Vol. 9, Nr. 1, 2006, S. 48-57.
- [27] Demant, C.; Streicher-Abel, B.; Waszkewitz, P.: Industrielle Bildverarbeitung: Wie optische Qualitätskontrolle wirklich funktioniert, 2. Auflage, Springer, Berlin, ISBN 3540419772, 2002.
- [28] Destruels, A.; Clauss, S.: Kann GigE-Vision seine Versprechen halten. In: INSPECT, GIT Verlag GmbH & Co. KG, Vol. 9, Nr. 1, S. 4-7, 2008.
- [29] Drews, K.: Vortrag: Bildverarbeitung als integraler Bestandteil der Automatisierungslandschaft. VDMA Infotag: Bildverarbeitung erfolgreich anwenden, 2008.
- [30] Eberhardt, M.; Roth, S.; König, A.: Industrial application of machine-in-the-loop-learning for a medical robot vision system – Concept and comprehensive field study. In: Special issue on Advances on Computer-based Biological Signal Processing, of the Journal Computers and Electrical Engineering, Elsevier, 2008.
- [31] Eberhart, R.C.; Kennedy, J.: Particle swarm optimization. In: Proceeding of the IEEE International Conference on Neural Networks, 1995, Piscataway, NJ, USA, S.1942-1948, 1995.
- [32] Eberhart, R.C.; Shi, Y.: Comparing inertia weights and constriction factors in particle

- swarm optimization. In: Proceedings of the Congress on Evolutionary Computation, Vol. 1, S. 84–88, 2000.
- [33] Eberhart, R.C.; Shi, Y.: Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of the 2001 Congress on Evolutionary Computation 2001, Vol. 1, S. 94-100, 2001.
- [34] Ellenrieder, M.M.; Komoto, H.: Model-based automatic calculation and evaluation of camera position for industrial machine vision. In: Proceedings of the SPIE Electronic Imaging Conference 2005, Vol. 5674, S. 467-478, 2005.
- [35] Ellenrieder, M.M.; Wöhler, C.; d'Angelo, P.: Reflectivity function based illumination and sensor planning for industrial inspection. Osten, W.; Gorecki, C.; Novak, E.L. (Hrsg.): SPIE Optical Measurement Systems for Industrial Inspection IV, Vol. 5856, S. 89-98, 2005.
- [36] Engel, M.: Intelligence, not size. Structure and uses of smart cameras. In: INSPECT, GIT Verlag GmbH & Co. KG, Vol. 7, Nr. 3, S. 30-32, 2006.
- [37] Freud, Y.; Schapire, R.E.: A short introduction to boosting. In: Journal of the Japanese Society for Artificial Intelligence, Vol. 14, Nr. 5, S. 771-780, 1999.
- [38] Fühner, T.; Erdmann, A.; Farkas, R.; Tollkühn, B.; Kókai, G.: Genetic algorithms to improve mask and illumination geometries in lithographic imaging systems. In: Proceedings of the EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoSAP, EvoMUSART, and EvoSTOC, Shanghai, China, S. 208-218, 2004.
- [39] Garnier, S.; Gautrais, J.; Theraulaz, G.: The biological principles of swarm intelligence. In: Swarm Intelligence, Vol. 1, Nr. 1, S. 3-31, 2007.
- [40] Göhring, D.: Seminararbeit: CCD vs. CMOS, Humboldt Universität Berlin, 2002.
- [41] Goldberg, G.A.: Genetic algorithms in search, optimization and machine learning, Addison Wesley, 1989.
- [42] Gonzales, R.; Woods, R.: Digital image processing (3rd Edition), Prentice Hall International, ISBN 9780131687288, 2007.
- [43] Görick, C.: Local orientation coding and adaptive thresholding for real time early vision. Interner Bericht IRINI 94-05, Institut für Neuroinformatik, Ruhr-Universität Bochum, 1994.
- [44] Görick, C.; Brauckmann, M.: Local orientation coding and neural network classifier with an application to real time car detection and tracking. In: Proceedings of the 16th Symposium of the DAGM and the 18th Workshop of the ÖAGM, Vienna, Österreich, 1994.
- [45] Haindl, M.; Grim, J.; Mikes, S.: Texture defect detection. In: Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns, CAIP 2007, Wien, Österreich, S. 987-994, 2007
- [46] Handl, J.; Meyer, B.: Ant-based and swarm-based clustering. In: Swarm Intelligence, Vol. 1, Nr. 2, S. 95-113. 2007
- [47] Haralick, R.; Shapiro, L.G.: Image segmentation techniques. In: Computer Vision, Graphics, and Image Processing, Vol. 29, Nr. 1, S. 100-132, 1985.
- [48] Haralick, R.; Shanmugan, K.; Dinstein, I.: Texture features for image classification. In: IEEE Transactions on Systems, Man, and Cybernetics, Vol. 3, Nr. 6, S. 610-621, 1973.

- [49] Harvey, N.R.; Marshall, S.: The use of genetic algorithms in morphological filter design. In: Signal Processing: Image Communication, Vol. 8, Nr. 1, S. 55-71, 1996.
- [50] Harvey, N.R.; Theiler, J.; Brumby, S.P.; Perkins, S.; Symanski, J.J.; Bloch, J.J.: Comparison of GENIE and conventional supervised classifiers for multispectral image feature extraction. Benediktsson, J.E. (Hrsg.): IEEE Transactions on Geosciences and Remote Sensing, Vol. 40, Nr. 2, S. 393-404, 2002.
- [51] Heckenkamp, C.: Das magische Auge. In: INSPECT, GIT Verlag GmbH & Co. KG, Vol. 9, Nr. 1, S. 25-28, 2008.
- [52] Heckenkamp, C.: Grundlagen der Bildverarbeitung. Kompendium der Fachbeiträge aus der INSPECT, GIT VERLAG, 2007.
- [53] Hermann, H.: Ein System zur schnellen Entwicklung von Bildverarbeitungsalgorithmen. Dissertation, Universität Mannheim, 2004.
- [54] Hickinbotham, S.; Austin, J.: Novelty detection in airframe strain data. In: Proceedings of the 15th International Conference on Pattern Recognition, 2000, Vol. 2, S. 536-539, 2000.
- [55] InnoNet: 3d-Wahrnehmung mit Smart-Pixel-Sensor (SmartVision). Projektbeschreibung, 2006.
- [56] Jähne, B.; Massen, R.; Nickolay, B.; Scharfenberg, H.: Technische Bildverarbeitung - Maschinelles Sehen, Springer, 1995.
- [57] Jähne, B.: Digitale Bildverarbeitung, 6. Auflage, Springer, ISBN 9783540249993, 2005.
- [58] Jain, A.K.; Duin, R. P.W.; Mao, J.: Statistical pattern recognition: A review. Kriegman, D.J. (Hrsg.): IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, Nr. 1, S. 4-37, 2000.
- [59] Jain, A.K.; Farrokhnia, F.: Unsupervised texture segmentation using Gabor filters. In: Pattern Recognition, Vol. 23, Nr. 12, S. 1167-1186, 1991.
- [60] Janson, S.; Middendorf, M.: A hierarchical particle swarm optimizer for dynamic optimization problems, In: Proceedings of the EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoISAP, EvoMUSART, and EvoSTOC, Coimbra, Portugal, S. 513-524, 2004.
- [61] Kanungo, P.; Nanda, P.K.; Samal, U.C.: Image segmentation using thresholding and genetic algorithm. In: Proceedings of the Conference on Soft Computing Technique for Engineering Applications, SCT 2006, Rourkela, India, 2006.
- [62] Kim, S.-M.; Kim, W.: An Algorithm for Segmenting Gaseous Objects on Images. In: Proceedings of the EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoISAP, EvoMUSART, and EvoSTOC, Coimbra, Portugal, S. 322-328, 2004
- [63] Koch, M.: Terahertz-Technologie: Quo-Vadis. In: Photonik, AT-Fachverlag GmbH, Nr. 4, S. 68-71, 2005
- [64] Kohavi, R.; John, G.H.: Wrappers for feature subset selection. In: Artificial Intelligence, Vol. 97, Nr. 1-2, S. 273-324, 1997.
- [65] König, A.; Mayr, C.; Bormann, T.; Klug, C.: Dedicated implementation of embedded vision systems employing low-power massively parallel feature computation. In: Proceedings of the 3rd VIVA-Workshop, Chemnitz, Germany, S. 1-8, 2002
- [66] Köppen, M.; Teunis, M.: Adaptation of morphological masks by genetic algorithm. In:

- Proceedings of the International Conference on Artificial Neural Networks and Expert Systems 1995, Dunedin, Neuseeland, S. 398-403, 1995.
- [67] Köppen, M.; Teunis, M.; Nickolay, B.: A framework for the evolutionary generation of 2d-lookup based texture filters. In: Proceedings of the 5th International Conference on Soft Computing, S. 965-970, 1998.
- [68] Köppen, M.; Nickolay, B.: Genetic programming based texture filtering framework, Pattern recognition in soft computing paradigm, World Scientific Publishing Co., S. 275-304, 2001.
- [69] Kovtun, I.: Texture segmentation of images on the basis of markov random fields, Technischer Rep.: TUD-FI03, Technische Universität Dresden, 2005.
- [70] Koza, J.R.: Genetic programming, MIT Press, 1992.
- [71] Kumar, A.; Pang, G.K.H.: Defect detection in textured materials using optimized filters. In: IEEE Transactions on Systems, Man, and Cybernetics, Vol. 32, Nr. 5, S. 553-570, 2002.
- [72] Kurita, T.; Otsu, N.; Sato, T.: A face recognition method using higher order local autocorrelation and multivariate analysis. In: Proceedings of the 11th IAPR International Conference on Pattern Recognition, The Hague, Niederlande, Vol. 2, S. 213, 1992.
- [73] Lades, M.; Vorbrueggen, J.C.; Buhmann, J.; Lange, J.; vd. Malsburg, C.; Wuertz, R.P.: Distortion invariant object recognition in the dynamic link architecture. Lombardi, F. (Hrsg.): IEEE Transactions on Computers, Vol. 42, Nr. 3, S. 300-311, 1993.
- [74] Laws, K.: Textured image segmentation. Dissertation, 1980
- [75] Leeser, M.; Miller, S.; Yu, H.: Smart camera based on reconfigurable hardware enables diverse real-time applications. In: Proceedings of the 12th IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM, Napa, Kanada, S. 147-155, 2004
- [76] Liedtke, C.; Münkel, H.; Rost, U.: SOLUTION for a learning configuration system for image processing, tasks and methods in applied artificial intelligence, In: Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Benicàssim, Spanien, S. 437-447, 1998.
- [77] Liedtke, C.-E.; Schnier, T.; Bloemer, A.: Automated learning of rules using genetic operators. In: Proceedings of the 5th International Conference on Computer Analysis of Images and Patterns, Budapest, Ungarn, S. 327-334, 1993.
- [78] Lin, Y.; Bhanu, B.: Learning features for object recognition. In: Proceedings of the Genetic and Evolutionary Computation Conference 2003, S. 2227
- [79] Liu, X.; Wang, D.; Member, S.: Texture classification using spectral histograms. In: IEEE Transactions on Image Processing, Vol. 12, Nr. 2003, S. 661
- [80] Mähler, K.: Zeile für Zeile Qualität. In: INSPECT, GIT Verlag GmbH &Co. KG, Vol. 8, Nr. 2, S. 22
- [81] Maillard, P.: Comparing texture analysis methods through classification, In: Photogrammetric Engineering and Remote Sensing, S. 357-368, 2003
- [82] Malamas, E. N.; Petrakis, E.-G.M.; Zervakis, M.; Pettit, L.; Legat, J.-D.: A survey on industrial vision systems, applications and tools. In: Image and Vision Computing,

- Vol. 21, Nr. 2, S. 171-188, 2003
- [83] Marcelli, A.; De Stefano, C.: Evolutionary algorithms for pattern recognition. Tutorial Note. In: Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006, Hongkong, 2006
- [84] Materka, A.; Strzelecki, M.: Texture analysis methods. CostB11 Report, 1998
- [85] Mayr, C.; König, A.: Low-power massively parallel feature computation for application-specific integrated vision systems. In: Proceedings of the 6th World Online Conference on Soft Computing WSC6, 2001
- [86] Melendez, J.; Garcia, M.; Puig, D.: Efficient distance-based per-pixel texture classification with Gabor wavelet filters. In: Pattern Analysis & Applications, Vol. 11, Nr. 3, S. 365-372, 2008.
- [87] McLaughlin, J.A.; Raviv, J.: Nth order autocorrelations in pattern recognition. In: Information and Control, Vol. 12, Nr. 2, S. 121-142, 1968.
- [88] Miyamoto, E.; Merryman, T. Jr: Fast calculation of Haralick texture features. Technischer Report, Carnegie Mellon Universität, <http://www.ece.cmu.edu/pueschel/teaching/18-799B-CMU/spring05/material/eizantad.pdf>, 2005 (letzter Zugriff 30.11.2008).
- [89] Monadjemi, A.: Towards efficient texture classification and abnormality Detection. Dissertation, University of Bristol, 2004
- [90] Mühlenbein, H.; Schomisch, D.; Born, J.: The parallel genetic algorithm as function optimizer, In: Parallel computing, Vol. 17, Nr. 6-7, S. 619-632, 1991
- [91] Munkelt, O.: VDMA Folienvortrag: Deutsche Bildverarbeitungsindustrie, 2007
- [92] Murino, V.; Bicego, M.; Rossi, I.A.: Statistical classification of raw textile defects. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, 2004.
- [93] Ojala, T.; Mäenpää, T.; Pietikäinen, M.; Viertola, J.; Kyllönen, J.; Huovinen, S.: Outex – new framework for empirical evaluation of texture analysis algorithms. In: Proceedings of the 16th International Conference on Pattern Recognition, Quebec, Canada, S. 701 - 706, 2002.
- [94] Ojala, T.; Pietikäinen, M.: Unsupervised texture segmentation using feature distributions. In: Pattern Recognition, Vol. 32, S. 477-486, 1999.
- [95] Ojala, T.; Pietikäinen, M.; Mäenpää, T.: Gray scale and rotation invariant texture classification with local binary patterns, In: Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland, S. 404-420, 2000.
- [96] Otsu, N.; Kurita, T.: A new scheme for practical flexible and intelligent vision systems. In: Proceedings of the IAPR Workshop on Computer Vision - Special Hardware and Industrial Applications, Tokio, Japan, Vol. 2, S. 431-435, 1988.
- [97] Pal, N.R.; Pal, S.: A review on image segmentation techniques. In: Pattern Recognition, Vol. 26, S. 1277-1294, 1993.
- [98] Pannekamp, J.: Adaptive Verfahren zur Bewertung texturierter Oberflächen. Dissertation, Jost-Jetter, Heimsheim, ISBN 3936947538, 2005.
- [99] Perkins, S.; Edlung, K.; Esch-Moscher, D.; Eads, D.; Harvey, N.R.; Brumby, S.P.: GENIE Pro: Robust image classification using shape, texture, and spectral information. Shen, S.S.; Lewis, P.L. (Hrsg.): Proceedings of the SPIE, S. 139-148,

- 2005.
- [100] Perkins, S.; Theiler, J.; Harvey, N.R.; Porter, R.B.; Bloch, J.J.; Symanski, J.J.: GENIE: A hybrid genetic algorithm for feature classification in multi-spectral images. Bosacchi, B.; Fogel, D.B.; Bezdek, J.C. (Hrsg.): Proceedings of the SPIE, S. 52-62, 2000.
 - [101] Peters, S.; Koenig, A.: A hybrid texture analysis system based on non-linear & oriented kernels, particle swarm optimization and kNN vs. support vector machines. In: Proceedings of the 7th International Conference on Hybrid Intelligent Systems, HIS'07, Kaiserslautern, Deutschland, 2007
 - [102] Peters, S.; Koenig, A.: Optimized texture operators for the automated design of image analysis systems: Non-linear and oriented kernels vs. gray value co-occurrence matrices. In: International Journal of Hybrid Intelligent Systems, Vol. 4, Nr. 3, S. 185-202, 2007.
 - [103] Peters, S.; Koenig, A.: A contribution to automatic design of image analysis systems - Breeding optimized non-linear and oriented kernels for texture analysis, In: Proceedings of the 6th International Conference on Hybrid Intelligent Systems, HIS 2006, Auckland, Neuseeland, 2006.
 - [104] Peters, S.; Koenig, A.: A contribution to automatic design of image analysis systems - segmentation of thin and thick fibers in 2d and 3d images, In: Proceedings of the International Conference on Instrumentation, Communication and Information Technology, ICICI 2005, Bandung, Indonesia, 2005.
 - [105] Petrou, M.; Garcia Sevilla, P.: Image processing: dealing with texture. Wiley, Chichester, 9780470026281, 2006.
 - [106] Pothos, V.; Theoharatos, C.; Zygoris, E.; Economou, G.: Distributional-based texture classification using non-parametric statistics. In: Pattern Analysis & Applications, Vol. 11, Nr. 2, S. 117-129, 2008.
 - [107] Poli, R.: Genetic programming for image analysis, MIT Press, S. 363-368, 1996.
 - [108] Poli, R.; Langdon, W.B.; Holland, O.: Extending particle swarm optimisation via genetic programming. In: Proceedings of the 8th European Conference on Genetic Programming, S. 291-300, 2005
 - [109] Poli, R.; Kennedy, J.; Blackwell, T.: Particle swarm optimization. An overview. In: Swarm intelligence, Springer, Vol. 1, Nr. 1, S. 33-57, 2007.
 - [110] Poli, R.: An analysis of publications on particle swarm optimisation applications. Technical Report CSM-469, 2007.
 - [111] Popovici, V.; Thiran, J.-P.: Pattern recognition using higher-order local autocorrelation coefficients. In: Pattern Recognition Letters, Vol. 25, S. 1107-1113, 2004.
 - [112] Puig, D.: Determining optimal window size for texture feature extraction methods. In: Proceedings of the IX Spanish Symposium on Pattern Recognition and Image Analysis, Benicasim, Spanien, Vol. 2, S. 237-242, 2001
 - [113] Puig, D.; Garcia, M.A.: Automatic texture feature selection for image pixel classification. In: Pattern Recognition Society, Elsevier, S. 1996-2009, 2006.
 - [114] Quintana, M.I.; Nava, M.: Morphological algorithm design for binary images using genetic programming. Banzhaf, W. (Hrsg.): Genetic Programming and Evolvable Machines, Springer, Vol. 7, Nr. 1, S. 81-102, 2006

- [115] Quintana, M.I.; Poli, R.; Claridge, E.: On two approaches to image processing algorithm design for binary images using GP. In: Proceedings of the EvoWorkshops 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM, Essex, UK, S. 422-431, 2003.
- [116] Randle, V.; Engler, O.: Introduction to texture analysis: macrotexture, microtexture and orientation mapping. Digital printing 2003, Taylor & Francis, London, 9056992244, 2003.
- [117] Randen, T.; Husoy, J.H.: Filtering for texture classification: A comparative study. Kriegman, D.J. (Hrsg.): IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, Nr. 4, S. 291-310, 1999
- [118] Rauhut, M.: Folienvortrag Messe Control 2007, 2007
- [119] Rechenberg, I.: Evolutionsstrategie '94. Frommann-Holzboog, Stuttgart, ISBN 3772816428, 1994.
- [120] Rechenberg, I.: Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart, ISBN 3772803733, 1973.
- [121] Reed, T.R.; Du Buf, J.M. H.: A review of recent texture segmentation and feature extraction techniques. In: Computer Vision, Graphics, and Image Processing: Image Understanding, Vol. 57, Nr. 3, S. 359-372, 1993.
- [122] Roberts, M.E.; Claridge, E.: An artificially evolved vision system for segmenting skin lesion images. In: Proceedings of the 6th International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI 2003, S. 655-662, 2003.
- [123] Rohrmus, D.R.: Invariant and adaptive geometrical texture features for defect detection and classification. In: Pattern Recognition Letters, Vol. 38, S. 1546-1559, 2005.
- [124] Rost, U.: Knowledge-based configuration of image processing algorithms. In: Proceedings of the International Conference on Computational Intelligence, ICCIMA98, Churchill, Australien, 1998.
- [125] Sankur, B.; Sezgin, M.: Image thresholding techniques: A survey over categories. Bogazii University, Electric--Electronic Engineering Department, Istanbul, Türkei
- [126] Sari-Sarraf, H.; Goddard, J.S.: Vision system for on-loom fabric inspection. Speck, C.E. (Hrsg.): IEEE Transactions on Industry Applications, Vol. 35, Nr. 6, S. 1252-1259, 1999.
- [127] Schael, M.: Methoden zur Konstruktion invarianter Merkmale für die Texturanalyse. Dissertation, Universität Freiburg, 2005
- [128] Schael, M.: Texture defect detection using invariant textural features. In: Proceedings of the 23rd DAGM-Symposium on Pattern Recognition, DAGM 2001, München, S. 17-24, 2001.
- [129] Schulz-Mirbach H.: Ein Referenzdatensatz zur Evaluierung von Sicht Prüfungsverfahren für Textil Oberflächen. Interner Bericht //96, Technische Universität Hamburg-Harburg, 1996, <http://lmb.informatik.uni-freiburg.de/research/dfg-texture/tilda/index.de.html> (letzter Zugriff: 30.11.2008)
- [130] Schwarzkopf, P.: VDMA: Deutsche Bildverarbeitungsindustrie, IBV-04-07-2007, 2007
- [131] Schwarzkopf, P.: VDMA Folienvortrag: Der Markt für die industrielle Bildverarbeitung

- in Deutschland, 2007
- [132] Sezgin, M.; Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. In: *Journal of Electronic Imaging*, Vol. 13, Nr. 1, S. 146-165, 2004
 - [133] Sharma, M.: Performance evaluation of image segmentation and texture extraction methods in scene analysis. Dissertation, Universität von Exeter, 2001.
 - [134] Shi, Y.: Particle swarm optimization. Feature article. Yen, G.G. (Hrsg.): *The Newsletter of the IEEE Neural Networks Society*, Vol. 2, Nr. 1, S. 1-13, 2004.
 - [135] Singh, S.; Markou, M.: An approach to novelty detection applied to the classification of image regions. Wu, X. (Hrsg.): *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, Nr. 4, S. 396-407, 2004.
 - [136] Singh, M.; Singh, S.: Spatial texture analysis: a comparative study. In: *Proceedings of the 16th International Conference on Pattern Recognition, ICPR 2002*, Quebec, Kanada, Nr. 1, S. 676-679, 2002.
 - [137] Sobral, J.L.: Optimized filter for texture defect detection. In: *Proceedings of the International Conference on Image Processing 2005, ICIP 2005*, Genua, Italien, Nr. 3, S. 565–568, 2005
 - [138] Soille, P.: *Morphological image analysis: principles and applications*, 2. Auflage, Springer, Berlin [u.a.], 3540429883, 2004.
 - [139] Stotz, M.: Finde den Fehler: Smart Kamera mit intuitiver Software zur automatisierten Sichtprüfung. In: *INSPECT*, GIT Verlag GmbH &Co. KG, Vol. 7, Nr. 2, S. 38-39, 2006.
 - [140] Struck, G.; Vieth, K.-U.: *Moderne Sichtprüfsysteme. Schütt mal, wir sortieren!*, Rheintal Druckerei und Verlag AG, Nr. 3, S. 24-26, 2008.
 - [141] Sun, K.; Yang, Y.; Ye, L.; Zhang, Q.: Image restoration using piecewise iterative curve fitting and texture synthesis, In: *Proceedings of the 4th International Conference on Intelligent Computing, ICIC 2008*, Shanghai, China, 2008, S. 1056-1063.
 - [142] Tangermann, M.W.: Feature selection for brain-computer interfaces. Dissertation, Universität Tübingen, 2007
 - [143] Tatari, S.: Auswahl von Texturanalyseverfahren zur Automatisierung industrieller Sichtprüfungen. Reihe 10: Informatik/Kommunikationstechnik, VDI Verlag, Nr. 67, 1986.
 - [144] Tawdross, P.M.M.: Bio-inspired circuit sizing and trimming methods for dynamically reconfigurable sensor electronics in industrial embedded systems. Dissertation, KLUEDO, Kaiserslauterer uniweiter elektronischer Dokumentenserver, 2007.
 - [145] Trelea, I.C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. In: *Information Processing Letters*, Vol. 85, Nr. 6, S. 317-325, 2003.
 - [146] Tsai, D.-M.; Wu, S.-K.: Automated surface inspection using Gabor filters. In: *The International Journal of Advanced Manufacturing Technology*, Vol. 16, S. 474-482, 2000.
 - [147] Van Gool, L.J.; Dewaele, P.; Oosterlinck, A.: Texture analysis anno 1983. In: *Computer Vision, Graphics, and Image Processing*, Vol. 29, Nr. 3, S. 336-357, 1985.

- [148] Varma, M.; Zisserman, A.: A statistical approach to texture classification from single images. In: International Journal of Computer Vision, Vol. 62, Nr. 1, 2005, S. 61-81.
- [149] VDI/VDE: VDI/VDE Richtlinie Nr. 2628. Automatisierte Sichtprüfung. Beschreibung der Prüfaufgabe, VDI-Verlag, 1989.
- [150] VDMA (Hrsg.): Industrielle Bildverarbeitung 2007/08. Anwendungen – Produkte – Bezugsquellen, VDMA Verlag GmbH, 2007
- [151] VDMA: Datenbank: Bildverarbeitung in Aktion. <http://www.vdma.org/wsp/portal/Home/de/Branchen/I/IBV/Anwenderberichte> (letzter Zugriff: 31.11.2008)
- [152] Viola, P.A.; Wells III, W.M.: Alignment by maximization of mutual information. In: International Journal of Computer Vision, Vol. 24, Nr. 2, S. 137-154, 1997.
- [153] Wagner, T.: Automatische Konfiguration von Bildverarbeitungssystemen. Habilitation, Shaker Verlag GmbH, 2000,
- [154] Weicker, K.: Evolutionäre Algorithmen. 2. Auflage, Teubner, Wiesbaden, 9783835102194, 2007.
- [155] Wie, L.-Y.; Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, New Orleans, USA, S. 479-488, 2001.
- [156] Xie, X.: A review of recent advances in surface defect detection using texture analysis techniques. In: Electronic Letters on Computer Vision and Image Analysis, Vol. 7, Nr. 3, S. 1-22, 2008.
- [157] Yang, L.; Albregtsen, F.; Taxt, T.: Fast computation of 3-D geometric moments using a discrete gauss' theorem. In: Proceedings of the 6th International Conference on Computer Analysis of Images and Patterns, CAIP 1995, Prag, Tschechien, S. 649-654, 1995.
- [158] Yardley, E.: Künstliche Intelligenz. Einsatz von neuronalen Netzwerken in der Bildverarbeitung. In: INSPECT, GIT Verlag GmbH & Co. KG, Vol. 8, Nr. 1, S. 30-32, 2007.
- [159] Yin, Q.; Kim, J.; Moon, K.-S.: Rotation invariant texture classification using Gabor wavelets, In: Proceedings of the 7th International Conference Computational Science, ICCS 2007, Beijing, China, 2007, S. 10-17.
- [160] Young, M.; Argiro, D.; Kubica, S.: Cantata: Visual programming environment for the Khoros system. In: Computer Graphics, Vol. 29, Nr. 2, S. 22-24, 1995.
- [161] Yu, M.; Eua-anant, N.; Saudagar, A.; Udpa, L: Genetic algorithm approach to image segmentation using morphological operations. In: Proceedings of the IEEE International Conference on Image Processing, ICIP 1998, Chicago, USA, 1998.
- [162] Zhang, J.G.; Tan, T.N.: Brief review of invariant texture analysis methods. In: Pattern Recognition Letters, Vol. 35, S. 735-747, 2002.
- [163] Zhang, X.; Du, Y.; Qin, Z.; Qin, G.; Lu, J.: A modified particle swarm optimizer for tracking dynamic systems. In: Proceedings of the 1st International Conference on Natural Computation, Lecture Notes in Computer Science, Springer, S. 592-601, 2005 Zorn, S.: Embedded Systems vs. Smart Camera. In: INSPECT, GIT Verlag GmbH & Co. KG, Vol. 8, Nr. 3, S. 16-17, 2007.
- [164] Zorn, S.: Embedded systems vs. smart Camera. In: INSPECT, GIT Verlag GmbH & Co. KG, Vol. 8, Nr. 3, S. 16-17, 2007.

Lebenslauf

Persönliche Daten

Name: Stefanie Peters
Geburtsdatum: 28. Mai 1977
Geburtsort: Lima/Peru
Nationalität: deutsch

Schule

1984-1996 Grundschule, Förderstufe und Gymnasium, Darmstadt

Studium

10/96-04/99 Studium der Physik,
Technische Universität Darmstadt

09/99-09/03 Studium der Optotechnik und Bildverarbeitung,
Fachhochschule Darmstadt

08/01-12/01 Praxissemester in Louisville (USA)

02/03-08/03 Diplomarbeit: Klarschriftlesung (OCR/OCV)
mit einer intelligenten Kamera

08/04-02/07 Promotionseignungsprüfung,
Technische Universität Kaiserslautern

03/04-03/09 Stipendiatin am Fraunhofer Institut für Techno- und
Wirtschaftsmathematik (ITWM), Abteilung Bildverarbeitung

03/07-03/10 Promotion,
Technische Universität Kaiserslautern