

Wilhelm Burger · Mark James Burge

Digitale Bildverarbeitung

Eine algorithmische Einführung mit Java

3., überarbeitete und ergänzte Auflage
Mit 375 Abbildungen, 25 Tabellen und 60 Algorithmen

Springer

Berlin Heidelberg New York
Hongkong London
Mailand Paris Tokio

Vorwort

Dieses Buch ist eine Einführung in die digitale Bildverarbeitung, die sowohl als Referenz für den Praktiker wie auch als Grundlagentext für die Ausbildung gedacht ist. Es bietet eine Übersicht über die wichtigsten klassischen Techniken in moderner Form und damit einen grundlegenden „Werkzeugkasten“ für dieses spannende Fachgebiet. Das Buch sollte daher insbesondere für folgende drei Einsatzbereiche gut geeignet sein:

- Für Wissenschaftler und Techniker, die digitale Bildverarbeitung als Hilfsmittel für die eigene Arbeit einsetzen oder künftig einsetzen möchten und Interesse an der Realisierung eigener, maßgeschneiderter Verfahren haben.
- Als umfassende Grundlage zum Selbststudium für ausgebildete IT-Experten, die sich erste Kenntnisse im Bereich der digitalen Bildverarbeitung und der zugehörigen Programmiertechnik aneignen oder eine bestehende Grundausbildung vertiefen möchten.
- Als einführendes Lehrbuch für eine ein- bis zweisemestrige Lehrveranstaltung im ersten Studienabschnitt, etwa ab dem 3. Semester. Die meisten Kapitel sind auf das Format einer wöchentlichen Vorlesung ausgelegt, ergänzt durch Einzelaufgaben für begleitende Übungen.

Inhaltlich steht die praktische Anwendbarkeit und konkrete Umsetzung im Vordergrund, ohne dass dabei auf die notwendigen formalen Details verzichtet wird. Allerdings ist dies kein Rezeptbuch, sondern Lösungsansätze werden schrittweise in drei unterschiedlichen Formen entwickelt: (a) in mathematischer Schreibweise, (b) als abstrakte Algorithmen und (c) als konkrete Java-Programme. Die drei Formen ergänzen sich und sollen in Summe ein Maximum an Verständlichkeit sicherstellen.

Wir betrachten digitale Bildverarbeitung nicht vorrangig als mathematische Disziplin und haben daher die formalen Anforderungen in diesem Buch auf das Notwendigste reduziert – sie gehen über die im ersten Studienabschnitt üblichen Kenntnisse nicht hinaus. Als Einsteiger sollte man daher auch nicht beunruhigt sein, dass einige Kapitel auf den ersten Blick etwas mathematisch aussehen. Die durchgehende, einheitliche Notation und ergänzenden Informationen im Anhang tragen dazu bei, eventuell bestehende Schwierigkeiten leicht zu überwinden. Bezüglich der *Programmierung* setzt das Buch gewisse Grundkenntnisse voraus, idealerweise (aber nicht notwendigerweise) in **Java**. Elementare Datenstrukturen, prozedurale Konstrukte und die Grundkonzepte der objektorientierten Programmierung sollten dem Leser vertraut sein. Da Java mittlerweile in vielen Studienplänen als erste Programmiersprache unterrichtet wird, sollte der Einstieg in diesen Fällen problemlos sein. Aber auch Java-Neulinge mit etwas Programmiererfahrung in ähnlichen Sprachen (insbesondere C/C++) dürften sich rasch zurechtfinden.

Softwareseitig basiert dieses Buch auf **ImageJ**, einer komfortablen, frei verfügbaren Programmierumgebung, die von Wayne Rasband am U.S. National Institute of Health (NIH) entwickelt wird.¹ ImageJ ist vollständig in Java implementiert, läuft damit auf vielen Plattformen und kann durch eigene, kleine „Plugin“-Module leicht erweitert werden. Die meisten Programmbeispiele sind jedoch so gestaltet, dass sie problemlos in andere Umgebungen oder Programmiersprachen portiert werden können.

Einsatz in Forschung und Entwicklung

Dieses Buch ist einerseits für den Einsatz in der Lehre konzipiert, bietet andererseits jedoch an vielen Stellen grundlegende Informationen und Details, die in dieser Form nicht immer leicht zu finden sind. Es sollte daher für den interessierten Praktiker und Entwickler eine wertvolle Hilfe sein. Es ist aber nicht als umfassender Ausgangspunkt zur Forschung gedacht und erhebt vor allem auch keinen Anspruch auf wissenschaftliche Vollständigkeit. Im Gegenteil, es wurde versucht, die Fülle der möglichen Literaturangaben auf die wichtigsten und (auch für Studierende) leicht zugreifbaren Quellen zu beschränken. Darüber hinaus konnten einige weiterführende Techniken, wie etwa hierarchische Methoden, Wavelets, Eigenimages oder Bewegungsanalyse, aus Platzgründen nicht berücksichtigt werden. Auch Themenbereiche, die mit „Intelligenz“ zu tun haben, wie Objekterkennung oder Bildverstehen, wurden bewusst ausgespart, und Gleiches gilt für alle dreidimensionalen Problemstellungen aus dem Bereich „Computer Vision“. Die in diesem Buch gezeigten Verfahren sind durchweg „blind und dumm“, wobei wir aber glauben, dass gerade die

¹ <http://rsb.info.nih.gov/ij/>

technisch saubere Umsetzung dieser scheinbar einfachen Dinge eine essentielle Grundlage für den Erfolg aller weiterführenden (vielleicht sogar „intelligenteren“) Ansätze ist.

Man wird auch enttäuscht sein, falls man sich ein Programmierhandbuch für ImageJ oder Java erwartet – dafür gibt es wesentlich bessere Quellen. Die Programmiersprache selbst steht auch nie im Mittelpunkt, sondern dient uns vorrangig als Instrument zur Verdeutlichung, Präzisierung und – praktischerweise – auch zur Umsetzung der gezeigten Verfahren.

Einsatz in der Ausbildung

An vielen Ausbildungseinrichtungen ist der Themenbereich digitale Signal- und Bildverarbeitung seit Langem in den Studienplänen integriert, speziell im Bereich der Informatik und Kommunikationstechnik, aber auch in anderen technischen Studienrichtungen mit entsprechenden formalen Grundlagen und oft auch erst in höheren („graduate“) Studiensemestern.

Immer häufiger finden sich jedoch auch bereits in der Grundausbildung einführende Lehrveranstaltungen zu diesem Thema, vor allem in neueren Studienrichtungen der Informatik und Softwaretechnik, Mechatronik oder Medientechnik. Ein Problem dabei ist das weitgehende Fehlen von geeigneter Literatur, die bezüglich der Voraussetzungen und der Inhalte diesen Anforderungen entspricht. Die klassische Fachliteratur ist häufig zu formal für Anfänger, während oft gleichzeitig manche populäre, praktische Methode nicht ausreichend genau beschrieben ist. So ist es auch für die Lektoren schwierig, für eine solche Lehrveranstaltung ein einzelnes Textbuch oder zumindest eine kompakte Sammlung von Literatur zu finden und den Studierenden empfehlen zu können. Das Buch soll dazu beitragen, diese Lücke zu schließen.

Die Inhalte der nachfolgenden Kapitel sind für eine Lehrveranstaltung von 1–2 Semestern in einer Folge aufgebaut, die sich in der praktischen Ausbildung gut bewährt hat. Die Kapitel sind meist in sich so abgeschlossen, dass ihre Abfolge relativ flexibel gestaltet werden kann. Der inhaltliche Schwerpunkt liegt dabei auf den klassischen Techniken im Bildraum, wie sie in der heutigen Praxis im Vordergrund stehen. Die Kapitel 18–20 zum Thema Spektraltechniken sind hingegen als grundlegende Einführung gedacht und bewusst im hinteren Teil des Buchs platziert. Sie können bei Bedarf leicht reduziert oder überhaupt weggelassen werden. Die nachfolgende Übersicht zeigt eine mögliche Aufteilung der Kursinhalte über zwei Semester.

„Road Map“ für 1- und 2-Semester-Kurse		1. Semester	2. Semester
1. Digitale Bilder	■	□
2. ImageJ	■	□
3. Histogramme	■	□
4. Punktoperationen	■	□
5. Filter	■	□
6. Kanten und Konturen	■	□
7. Auffinden von Eckpunkten	□	■
8. Detektion einfacher Kurven	□	■
9. Morphologische Filter	■	□
10. Regionen in Binärbildern	■	□
11. Automatische Schwellwertoperationen	□	■
12. Farbbilder	■	□
13. Farbquantisierung	□	■
14. Colorimetrische Farbräume	□	■
15. Filter für Farbbilder	□	■
16. Kanten in Farbbildern	□	■
17. Kantenerhaltende Glättungsfilter	□	■
18. Einführung in Spektraltechniken	□	■
19. Diskrete Fouriertransformation in 2D	□	■
20. Diskrete Kosinustransformation (DCT)	□	■
21. Geometrische Bildoperationen	■	□
22. Interpolation	■	□
23. Bildvergleich	■	□
24. Elastischer Bildvergleich	□	■
25. Skaleninvariante, lokale Bildmerkmale (SIFT)	□	■

Ergänzungen zur aktuellen 3. Auflage

Die vorliegende dritte Auflage dieses Buchs ist nicht nur eine durchgehende Überarbeitung sondern enthält einige wichtige zusätzliche Kapitel, die bisher nur in englischer Sprache publiziert wurden. Gänzlich neu sind Themen wie *automatische Schwellwertoperationen* (Kap. 11), *Filter und Kantendetektoren für Farbbilder* (Kap. 15–16), *kantenerhaltende Glättungsfilter* (Kap. 17) und *elastischer Bildvergleich* (Kap. 24). Diese Ergänzungen knüpfen direkt an die einführenden Kapitel an und enthalten zum Teil sehr detailliertes und vertiefendes Material, das eine unmittelbare Umsetzung der Verfahren ermöglicht.

Eine Sonderrolle nimmt das abschließende Kapitel über *skaleninvariante, lokale Bildmerkmale* (Kap. 25) ein, das eine sehr ausführliche Darstellung des klassischen SIFT-Verfahrens enthält, nicht zuletzt auch um beispielhaft zu zeigen, welche diffizilen Aspekte bei der Realisierung eines funktionierenden Verfahrens dieser Art in der Praxis zu berücksichtigen sind. Einige weitere Kapitel wurden zur besseren Übersichtlichkeit

und für den einfacheren Einsatz in der Lehre neu strukturiert oder auf mehrere Kapitel aufgeteilt. Die mathematische Notation und die Programmbeispiele wurden durchgehend überarbeitet und fast alle Abbildungen wurden an die Möglichkeiten des Farbdrucks angepasst oder neu erstellt. Der Anhang wurde vor allem durch zusätzliche mathematische Grundlagen beträchtlich erweitert.

Um den vorgegebenen Seitenumfang des Buchs nicht zu sprengen, wurde andererseits auf die bisher enthaltene ImageJ-Kurzreferenz sowie auf die Einbindung von Java-Quellcode im Anhang verzichtet. Beides ist weiterhin in der jeweils aktuellen Fassung auf der Website des Buchs online verfügbar.

Online-Materialien

Auf der Website zu diesem Buch,

`www.imagingbook.com`,

stehen zusätzliche Materialien in elektronischer Form frei zur Verfügung, u. a. Testbilder in Originalgröße, Java-Quellcode zu den angeführten Beispielen, Links, aktuelle Ergänzungen, und allfällige Korrekturen. Kommentare, Fragen, Anregungen und Korrekturen sind willkommen und sollten adressiert werden an

`imagingbook@gmail.com`.

Übungsaufgaben und Lösungen

Dieses Buch enthält zu jedem Kapitel beispielhafte Übungsaufgaben, die vor allem Lehrenden die Erstellung eigener Aufgaben erleichtern sollen. Die meisten dieser Aufgaben sind nach dem Studium des zugehörigen Kapitels einfach zu lösen, andere wiederum erfordern etwas mehr Denkarbeit oder sind experimenteller Natur. Wir gehen davon aus, dass Lehrende Umfang und Schwierigkeit der einzelnen Aufgaben in Relation zum Ausbildungsstand ihrer Studierenden am Besten beurteilen und auch selbst entsprechende Adaptierungen vornehmen können. Nicht zuletzt aus diesem Grund stellen wir auch keine expliziten Lösungen zu den Aufgaben zur Verfügung, helfen jedoch auf persönliche Anfrage gerne aus, falls einzelne Aufgaben unklar erscheinen oder sich einer einfachen Lösung entziehen.

Ein Dankeschön

Dieses Buch wäre nicht entstanden ohne das Verständnis und die Unterstützung unserer Ehepartner und Familien, und zwar über einen wesentlich längeren Zeitraum hinweg, als für dieses Projekt ursprünglich veranschlagt war. Unser Dank geht auch an Wayne Rasband am NIH

für die unermüdliche (Weiter-)Entwicklung von ImageJ und sein anhaltendes Engagement innerhalb der Community. Die Verwendung von Open-Source Software birgt immer ein gewisses Risiko, da die langfristige Akzeptanz und Kontinuität nur schwer einschätzbar ist. ImageJ als Softwarebasis für dieses Buch zu wählen, war nachträglich gesehen eine gute Entscheidung, und wir würden uns glücklich schätzen, mit diesem Buch indirekt auch ein wenig zum Erfolg dieser Software beigetragen zu haben.

Ein herzlicher Dank geht auch an die zahlreichen aufmerksamen Leser der bisherigen Auflagen für ihre positiven Kommentare, Korrekturen und konstruktiven Verbesserungsvorschläge. Alle Planungen bezüglich des erforderlichen Zeitaufwands für diese Neuauflage erwiesen sich letztlich als viel zu optimistisch, wodurch sich die Fertigstellung des Manuskripts leider immer wieder verzögerte. Dem Produktionsteam des Springer-Verlags und insbesondere Frau Dorothea Glaunsinger und Herrn Hermann Engesser sei daher unser besonderer Dank für ihr Verständnis, ihre unendliche Geduld und die moralische Unterstützung versichert, ohne die dieses Projekt mit Sicherheit gescheitert wäre. Ausdrücklich möchten wir uns auch nochmals beim Verlag für den durchgängigen Farbdruck, die Verwendung von hochwertigem Papier und die großzügig eingeräumten Freiheiten in der Gestaltung des Layouts bedanken.

Hagenberg / Washington D.C.
Februar 2015

Inhaltsverzeichnis

1	Digitale Bilder	1
1.1	Programmieren mit Bildern	2
1.2	Bildanalyse und „intelligente“ Verfahren	3
1.3	Arten von digitalen Bildern	4
1.4	Bildaufnahme	4
1.4.1	Das Modell der Lochkamera	4
1.4.2	Die „dünne“ Linse	7
1.4.3	Übergang zum Digitalbild	8
1.4.4	Bildgröße und Auflösung	9
1.4.5	Bildkoordinaten	10
1.4.6	Pixelwerte	10
1.5	Dateiformate für Bilder	13
1.5.1	Raster- vs. Vektordaten	14
1.5.2	Tagged Image File Format (TIFF)	14
1.5.3	Graphics Interchange Format (GIF)	15
1.5.4	Portable Network Graphics (PNG)	16
1.5.5	JPEG	16
1.5.6	Windows Bitmap (BMP)	20
1.5.7	Portable Bitmap Format (PBM)	20
1.5.8	Weitere Dateiformate	20
1.5.9	Bits und Bytes	21
1.6	Aufgaben	23
2	ImageJ	25
2.1	Software für digitale Bilder	26
2.1.1	Software zur Bildbearbeitung	26
2.1.2	Software zur Bildverarbeitung	26
2.2	Eigenschaften von ImageJ	26
2.2.1	Features	28

2.2.2	Interaktive Werkzeuge	28
2.2.3	ImageJ-Plugins	29
2.2.4	Beispiel-Plugin: „inverter“	30
2.2.5	Plugin <code>My_Inverter_A</code>	31
2.2.6	Plugin <code>My_Inverter_B</code>	33
2.2.7	<code>PlugIn</code> oder <code>PlugInFilter</code> ?	33
2.2.8	Ausführen von ImageJ „ <i>Commands</i> “	35
2.3	Weitere Informationen zu ImageJ und Java	37
2.3.1	Ressourcen für ImageJ	38
2.3.2	Programmieren mit Java	38
2.4	Aufgaben	38
3	Histogramme	41
3.1	Was ist ein Histogramm?	42
3.2	Was ist aus Histogrammen abzulesen?	43
3.2.1	Eigenschaften der Bildaufnahme	44
3.2.2	Bildfehler	46
3.3	Berechnung von Histogrammen	47
3.4	Histogramme für Bilder mit mehr als 8 Bit	50
3.4.1	Binning	50
3.4.2	Beispiel	50
3.4.3	Implementierung	51
3.5	Histogramme von Farbbildern	52
3.5.1	Luminanzhistogramm	52
3.5.2	Histogramme der Farbkomponenten	52
3.5.3	Kombinierte Farbhistogramme	52
3.6	Das kumulative Histogramm	54
3.7	Statistische Informationen aus dem Histogramm	54
3.7.1	Mittelwert und Varianz	55
3.7.2	Median	56
3.8	Aufgaben	57
4	Punktoperationen	59
4.1	Änderung der Bildintensität	60
4.1.1	Kontrast und Helligkeit	60
4.1.2	Beschränkung der Ergebniswerte (<i>clamping</i>)	60
4.1.3	Invertieren von Bildern	61
4.1.4	Schwellwertoperation (<i>thresholding</i>)	61
4.2	Punktoperationen und Histogramme	62
4.3	Automatische Kontrastanpassung	63
4.4	Modifizierte Auto-Kontrast-Funktion	64
4.5	Linearer Histogrammausgleich	66
4.6	Histogrammanpassung	69
4.6.1	Häufigkeiten und Wahrscheinlichkeiten	69
4.6.2	Prinzip der Histogrammanpassung	71
4.6.3	Stückweise lineare Referenzverteilung	71
4.6.4	Anpassung an ein spezifisches Histogramm	73

4.6.5	Beispiel 1: Stückweise lineare Verteilungsfunktion .	75
4.6.6	Beispiel 2: Gaußförmiges Referenzhistogramm	77
4.6.7	Beispiel 3: Histogrammanpassung an ein zweites Bild	77
4.7	Gammakorrektur	78
4.7.1	Warum Gamma?	79
4.7.2	Mathematische Definition	80
4.7.3	Reale Gammawerte	80
4.7.4	Anwendung der Gammakorrektur	82
4.7.5	Implementierung	83
4.7.6	Modifizierte Gammakorrektur	83
4.8	Punktoperationen in ImageJ	86
4.8.1	Punktoperationen mit Lookup-Tabellen	86
4.8.2	Arithmetische Standardoperationen	87
4.8.3	Punktoperationen mit mehreren Bildern	87
4.8.4	ImageJ-Plugins für mehrere Bilder	89
4.9	Aufgaben	92
5	Filter	93
5.1	Was ist ein Filter?	93
5.2	Lineare Filter	95
5.2.1	Die Filtermatrix	95
5.2.2	Anwendung des Filters	96
5.2.3	Berechnung der Filteroperation	97
5.2.4	Beispiele für Filter-Plugins	98
5.2.5	Ganzzahlige Koeffizienten	99
5.2.6	Filter beliebiger Größe	101
5.2.7	Arten von linearen Filtern	102
5.3	Formale Eigenschaften linearer Filter	105
5.3.1	Lineare Faltung	105
5.3.2	Eigenschaften der linearen Faltung	106
5.3.3	Separierbarkeit von Filtern	108
5.3.4	Impulsantwort eines linearen Filters	110
5.4	Nichtlineare Filter	111
5.4.1	Minimum- und Maximum-Filter	112
5.4.2	Medianfilter	113
5.4.3	Gewichtetes Medianfilter	114
5.4.4	Andere nichtlineare Filter	117
5.5	Implementierung von Filtern	118
5.5.1	Effizienz von Filterprogrammen	118
5.5.2	Behandlung der Bildränder	118
5.5.3	Debugging von Filterprogrammen	119
5.6	Filteroperationen in ImageJ	120
5.6.1	Lineare Filter	121
5.6.2	Gaußfilter	121
5.6.3	Nichtlineare Filter	122
5.7	Aufgaben	122

6	Kanten und Konturen	125
6.1	Wie entsteht eine Kante?	125
6.2	Gradienten-basierte Kantendetektion	126
6.2.1	Partielle Ableitung und Gradient	127
6.2.2	Ableitungsfilter	127
6.3	Einfache Kantenoperatoren	129
6.3.1	Prewitt- und Sobel-Operator	129
6.3.2	Roberts-Operator	131
6.3.3	Kompass-Operatoren	131
6.3.4	Kantenoperatoren in ImageJ	134
6.4	Weitere Kantenoperatoren	135
6.4.1	Kantendetektion mit zweiten Ableitungen	135
6.4.2	Kanten auf verschiedenen Skalenebenen	135
6.4.3	Von Kanten zu Konturen	136
6.5	Der Canny-Kantenoperator	137
6.5.1	Vorverarbeitung	138
6.5.2	Lokalisierung der Kanten	140
6.5.3	Kantenverfolgung mit Hysterese-Schwellwert	141
6.5.4	Weitere Informationen zum Canny-Operator	143
6.5.5	Implementierung	145
6.6	Kantenschärfung	145
6.6.1	Kantenschärfung mit dem Laplace-Filter	145
6.6.2	Unschärfe Maskierung (<i>unsharp masking</i>)	149
6.7	Aufgaben	152
7	Auffinden von Eckpunkten	155
7.1	„Points of interest“	155
7.2	Harris-Detektor	156
7.2.1	Lokale Strukturmatrix	156
7.2.2	<i>Corner Response Function</i> (CRF)	157
7.2.3	Bestimmung der Eckpunkte	158
7.2.4	Beispiele	158
7.3	Implementierung	158
7.3.1	Schritt 1 – Berechnung der <i>corner response function</i>	160
7.3.2	Schritt 2 – Bestimmung der Eckpunkte	164
7.3.3	Schritt 3: Aufräumen	166
7.3.4	Zusammenfassung	167
7.4	Aufgaben	168
8	Detektion einfacher Kurven	169
8.1	Auffällige Strukturen	169
8.2	Hough-Transformation	170
8.2.1	Parameterraum	171
8.2.2	Akkumulator	173
8.2.3	Eine bessere Geradenparametrisierung	173
8.3	Hough-Algorithmus	175

8.3.1	Auswertung des Akkumulators	176
8.3.2	Erweiterungen der Hough-Transformation	180
8.3.3	Schnittpunkte von Geraden	182
8.4	Implementierung	184
8.5	Hough-Transformation für konische Kurven	186
8.5.1	Kreise und Kreisbögen	186
8.5.2	Ellipsen	188
8.6	Aufgaben	189
9	Morphologische Filter	191
9.1	Schrumpfen und wachsen lassen	192
9.1.1	Nachbarschaft von Bildelementen	193
9.2	Morphologische Grundoperationen	193
9.2.1	Das Strukturelement	194
9.2.2	Punkt <i>mengen</i>	194
9.2.3	Dilation	195
9.2.4	Erosion	196
9.2.5	Formale Eigenschaften von Dilation und Erosion	197
9.2.6	Design morphologischer Filter	198
9.2.7	Anwendungsbeispiel: <i>Outline</i>	202
9.3	Zusammengesetzte morphologische Operationen	203
9.3.1	Opening	203
9.3.2	Closing	204
9.3.3	Eigenschaften von Opening und Closing	204
9.4	Verdünnung – <i>Thinning</i>	205
9.4.1	Thinning-Algorithmus von Zhang und Suen	206
9.4.2	Schneller Algorithmus (<i>fast thinning</i>)	209
9.4.3	Implementierung	209
9.4.4	Morphologische Operationen in ImageJ	213
9.5	Morphologische Filter für Grauwertbilder	215
9.5.1	Strukturelemente	215
9.5.2	Dilation und Erosion	216
9.5.3	Opening und Closing	216
9.5.4	Implementierung	217
9.6	Aufgaben	219
10	Regionen in Binärbildern	223
10.1	Auffinden von Bildregionen	224
10.1.1	Regionenmarkierung durch <i>Flood Filling</i>	224
10.1.2	Sequentielle Regionenmarkierung	228
10.1.3	Regionenmarkierung – Zusammenfassung	233
10.2	Konturen von Regionen	234
10.2.1	Äußere und innere Konturen	235
10.2.2	Kombinierte Regionenmarkierung und Konturfindung	236
10.3	Repräsentation einzelner Bildregionen	240
10.3.1	Matrix-Repräsentation	240

10.3.2	Lauf­längen­kodierung	242
10.3.3	<i>Chain Codes</i>	243
10.4	Eigenschaften binärer Bildregionen	246
10.4.1	Formmerkmale (<i>Features</i>)	246
10.4.2	Geometrische Eigenschaften	247
10.4.3	Statistische Formeigenschaften	250
10.4.4	Momentenbasierte geometrische Merkmale	253
10.4.5	Projektionen	259
10.4.6	Topologische Merkmale	260
10.5	Implementierung	261
10.6	Aufgaben	262
11	Automatische Schwellwertoperationen	267
11.1	Globale, histogrammbasierte Schwellwertoperationen	268
11.1.1	Einfache Verfahren zur Bestimmung des Schwellwerts	271
11.1.2	Iterative Schwellwertbestimmung (Isodata- Algorithmus)	273
11.1.3	Methode von Otsu	274
11.1.4	Maximale-Entropie-Methode	279
11.1.5	Minimum-Error-Methode	283
11.2	Lokale, adaptive Schwellwertbestimmung	291
11.2.1	Methode von Bernsen	291
11.2.2	Adaptive Schwellwertmethode von Niblack	295
11.3	Java-Implementierung	304
11.3.1	Globale Schwellwertoperationen	304
11.3.2	Adaptive Schwellwertoperationen	304
11.4	Zusammenfassung und weitere Quellen	307
11.5	Aufgaben	308
12	Farbbilder	309
12.1	RGB-Farbbilder	309
12.1.1	Aufbau von Farbbildern	311
12.1.2	Farbbilder in ImageJ	314
12.2	Farbräume und Farbkonversion	322
12.2.1	Umwandlung in Grauwertbilder	323
12.2.2	HSV/HSB- und HLS-Farbraum	325
12.2.3	TV-Komponentenfarbräume: YUV, YIQ und YCbCr	337
12.2.4	Farbräume für den Druck: CMY und CMYK	340
12.3	Statistiken von Farbbildern	345
12.3.1	Wie viele Farben enthält ein Bild überhaupt?	345
12.3.2	Histogramme	346
12.4	Aufgaben	347

13	Farbquantisierung	351
13.1	Skalare Farbquantisierung	352
13.2	Vektorquantisierung	353
13.2.1	Populosity-Algorithmus	354
13.2.2	Median-Cut-Algorithmus	354
13.2.3	Octree-Algorithmus	355
13.2.4	Weitere Methoden zur Vektorquantisierung	359
13.2.5	Implementierung	360
13.3	Aufgaben	362
14	Colorimetrische Farbräume	363
14.1	CIE-Farbräume	363
14.1.1	Der CIEXYZ-Farbraum	364
14.1.2	xy -Chromazitätsdiagramm	364
14.1.3	Normbeleuchtung	367
14.1.4	Gamut	367
14.1.5	Varianten des CIE-Farbraums	368
14.1.6	CIELAB	369
14.1.7	CIELUV	370
14.1.8	Berechnung von Farbdifferenzen	373
14.2	Standard-RGB (sRGB)	373
14.2.1	Lineare vs. nichtlineare Farbwerte	374
14.2.2	Transformation CIEXYZ \rightarrow sRGB	375
14.2.3	Transformation sRGB \rightarrow CIEXYZ	375
14.2.4	Rechnen mit sRGB-Werten	376
14.3	Adobe RGB-Farbraum	377
14.4	Chromatische Adaptierung	378
14.4.1	XYZ-Skalierung	379
14.4.2	Bradford-Adaptierung	379
14.5	Colorimetrische Farbräume in Java	380
14.5.1	<i>Profile Connection Space</i> (PCS)	381
14.5.2	Relevante Java-Klassen	384
14.5.3	Implementierung des CIELAB-Farbraums (Beispiel)	385
14.5.4	ICC-Profile	386
14.6	Aufgaben	389
15	Filter für Farbbilder	391
15.1	Lineare Filter	391
15.1.1	Monochromatische Anwendung linearer Filter	392
15.1.2	Einfluss des verwendeten Farbraums	395
15.1.3	Lineare Filteroperationen bei zyklischen Komponenten	399
15.2	Nichtlineare Filter für Farbbilder	403
15.2.1	Skalares Medianfilter	403
15.2.2	Vektor-Medianfilter	404
15.2.3	Schärfendes Vektor-Medianfilter	408

15.3	Java-Implementation	412
15.4	Weiterführende Literatur	415
15.5	Aufgaben	416
16	Kanten in Farbbildern	417
16.1	Monochromatische Methoden	418
16.2	Kanten aus vektorwertigen Bilddaten	423
16.2.1	Mehrdimensionale Gradienten	424
16.2.2	Die Jacobi-Matrix	424
16.2.3	Quadratischer lokaler Kontrast	425
16.2.4	Stärke von Farbkanten	426
16.2.5	Orientierung von Farbkanten	428
16.2.6	Grauwertgradient als Spezialfall	431
16.3	Canny-Operator für Farbbilder	433
16.4	Andere Farbkantenoperatoren	437
16.5	Java-Implementierung	438
17	Kantenerhaltende Glättungsfilter	441
17.1	Kuwahara-Filter	442
17.1.1	Anwendung auf Farbbilder	446
17.2	Bilaterales Filter	449
17.2.1	Domain-Filter	450
17.2.2	Range-Filter	450
17.2.3	Bilaterales Filter (allgemein)	451
17.2.4	Bilaterales Filter mit gaußförmigen Kernen	452
17.2.5	Anwendung auf Farbbilder	453
17.2.6	Effiziente Implementierung durch x/y -Separierung	459
17.2.7	Weitere Informationen	462
17.3	Anisotrope Diffusionsfilter	463
17.3.1	Homogene Diffusion und Wärmeleitungsgleichung	465
17.3.2	Das Perona-Malik-Filter	467
17.3.3	Perona-Malik-Filter für Farbbilder	470
17.4	Implementierung	475
17.5	Aufgaben	476
18	Einführung in Spektraltechniken	479
18.1	Fouriertransformation	480
18.1.1	Sinus- und Kosinusfunktionen	480
18.1.2	Fourierreihen zur Darstellung periodischer Funktionen	483
18.1.3	Fourierintegral	484
18.1.4	Fourierspektrum und -transformation	485
18.1.5	Fourier-Transformationspaare	486
18.1.6	Wichtige Eigenschaften der Fouriertransformation	487
18.2	Übergang zu diskreten Signalen	491
18.2.1	Abtastung	491
18.2.2	Diskrete und periodische Funktionen	497

18.3	Die diskrete Fouriertransformation (DFT)	498
18.3.1	Definition der DFT	498
18.3.2	Diskrete Basisfunktionen	501
18.3.3	Schon wieder Aliasing!	501
18.3.4	Einheiten im Orts- und Spektralraum	502
18.3.5	Das Leistungsspektrum	506
18.4	Implementierung der DFT	507
18.4.1	Direkte Implementierung	507
18.4.2	Fast Fourier Transform (FFT)	509
18.5	Aufgaben	509
19	Diskrete Fouriertransformation in 2D	511
19.1	Definition der 2D-DFT	511
19.1.1	2D-Basisfunktionen	512
19.1.2	Implementierung der zweidimensionalen DFT	512
19.2	Darstellung der Fouriertransformierten in 2D	513
19.2.1	Wertebereich	516
19.2.2	Zentrierte Darstellung	516
19.3	Frequenzen und Orientierung in 2D	516
19.3.1	Effektive Frequenz	517
19.3.2	Frequenzlimits und Aliasing in 2D	518
19.3.3	Orientierung	519
19.3.4	Geometrische Normalisierung des 2D-Spektrums ..	519
19.3.5	Auswirkungen der Periodizität	520
19.3.6	<i>Windowing</i>	520
19.3.7	Gängige Fensterfunktionen	522
19.4	Beispiele für Fouriertransformierte in 2D	527
19.4.1	Skalierung	527
19.4.2	Periodische Bildmuster	527
19.4.3	Drehung	527
19.4.4	Gerichtete, längliche Strukturen	527
19.4.5	Natürliche Bilder	527
19.4.6	Druckraster	527
19.5	Anwendungen der DFT	531
19.5.1	Lineare Filteroperationen im Spektralraum	531
19.5.2	Lineare Faltung und Korrelation	532
19.5.3	Inverse Filter	533
19.6	Aufgaben	534
20	Diskrete Kosinustransformation (DCT)	535
20.1	Eindimensionale DCT	535
20.1.1	Basisfunktionen der DCT	536
20.1.2	Implementierung der eindimensionalen DCT	536
20.2	Zweidimensionale DCT	538
20.2.1	Beispiele	539
20.2.2	Separierbarkeit	539
20.3	Implementierung	542

20.4	Weitere Spektraltransformationen	543
20.5	Aufgaben	543
21	Geometrische Bildoperationen	545
21.1	2D-Koordinatentransformation	546
21.1.1	Einfache geometrische Abbildungen	547
21.1.2	Homogene Koordinaten	548
21.1.3	Affine Abbildung (Dreipunkt-Abbildung)	548
21.1.4	Projektive Abbildung (Vierpunkt-Abbildung)	551
21.1.5	Bilineare Abbildung	556
21.1.6	Weitere nichtlineare Bildtransformationen	557
21.1.7	Lokale Transformationen	559
21.2	Resampling	560
21.2.1	<i>Source-to-Target Mapping</i>	561
21.2.2	<i>Target-to-Source Mapping</i>	562
21.3	Java-Implementierung	563
21.3.1	Lineare Abbildungen	564
21.3.2	Nichtlineare Abbildungen	564
21.3.3	Anwendungsbeispiele	564
21.4	Aufgaben	565
22	Interpolation	569
22.1	Einfache Interpolationsverfahren	569
22.1.1	„Ideale“ Interpolation	570
22.2	Interpolation als Faltung	574
22.3	Kubische Interpolation	574
22.4	Spline-Interpolation	576
22.4.1	Catmull-Rom-Interpolation	577
22.4.2	Kubische B-Spline-Interpolation	578
22.4.3	Mitchell-Netravali-Approximation	578
22.5	Lanczos-Interpolation	579
22.6	Interpolation in 2D	582
22.6.1	Nearest-Neighbor-Interpolation in 2D	582
22.6.2	Bilineare Interpolation	583
22.6.3	Bikubische und Spline-Interpolation in 2D	584
22.6.4	Lanczos-Interpolation in 2D	586
22.6.5	Beispiele und Diskussion	587
22.7	Aliasing	588
22.7.1	Abtastung der rekonstruierten Bildfunktion	589
22.7.2	Tiefpassfilter	591
22.8	Java-Implementierung	592
22.9	Aufgaben	594

23	Bildvergleich	597
23.1	Template Matching in Intensitätsbildern	598
23.1.1	Abstand zwischen Bildmustern	599
23.1.2	Umgang mit Drehungen und Größenänderungen	607
23.1.3	Implementierung	607
23.2	Vergleich von Binärbildern	607
23.2.1	Direkter Vergleich von Binärbildern	609
23.2.2	Die Distanztransformation	610
23.2.3	<i>Chamfer Matching</i>	613
23.2.4	Implementierung	617
23.3	Aufgaben	617
24	Elastischer Bildvergleich	619
24.1	Das Lucas-Kanade-Verfahren	619
24.1.1	Registrierung in 1D	620
24.1.2	Erweiterung auf mehrdimensionale Funktionen	621
24.2	Lucas-Kanade-Algorithmus	622
24.2.1	Zusammenfassung des Algorithmus	626
24.3	<i>Inverse-Compositional</i> -Algorithmus	627
24.4	Parametereinstellungen für verschiedene lineare Transformationen	630
24.4.1	Translation	630
24.4.2	Affine Transformation	633
24.4.3	Projektive Transformation	634
24.4.4	Verkettung linearer Transformationen	635
24.5	Beispiel	636
24.6	Java-Implementierung	637
24.6.1	Anwendungsbeispiel	640
24.7	Aufgaben	640
25	Skaleninvariante Bildmerkmale (SIFT)	643
25.1	Merkmalspunkte auf verschiedenen Skalenebenen	644
25.1.1	Das Laplace-Gauß-Filter (LoG)	645
25.1.2	Approximation der LoG-Funktion durch die Differenz zweier Gaußfunktionen (DoG)	648
25.1.3	Der Gauß-Skalenraum	650
25.1.4	LoG/DoG-Skalenraum	655
25.1.5	Hierarchischer Skalenraum	656
25.1.6	Der Skalenraum im SIFT-Verfahren	662
25.2	Lokalisierung von Merkmalspunkten	664
25.2.1	Detektion von Extremwerten im DoG-Skalenraum	666
25.2.2	Verfeinerung der Position	670
25.2.3	Unterdrückung kantenartiger Bildstrukturen	673
25.3	Berechnung der lokalen Deskriptoren	675
25.3.1	Bestimmung der dominanten Orientierungen	675
25.3.2	Konstruktion des SIFT-Descriptors	680
25.4	SIFT-Algorithmus – Zusammenfassung	688

25.5	Vergleich von SIFT-Merkmalen	698
25.5.1	Bestimmung der Ähnlichkeit von Merkmalen	698
25.5.2	Beispiele	699
25.6	Effiziente Zuordnung von Merkmalen	704
25.7	SIFT-Implementierung in Java	706
25.7.1	Detektion von SIFT-Merkmalen	706
25.7.2	Zuordnung von SIFT-Merkmalen	707
25.8	Aufgaben	708
A	Mathematische Notation	711
A.1	Symbole	711
A.2	Operatoren für Mengen	715
A.3	Komplexe Zahlen	716
A.4	Algorithmische Komplexität und \mathcal{O} -Notation	717
B	Ergänzungen zur Algebra	719
B.1	Vektoren und Matrizen	719
B.1.1	Spalten- und Zeilenvektoren	720
B.1.2	Länge (Norm) eines Vektors	720
B.2	Matrix-Multiplikation	721
B.2.1	Multiplikation mit einem Skalarwert	721
B.2.2	Produkt zweier Matrizen	721
B.2.3	Matrix-Vektor-Produkt	721
B.3	Vektor-Produkte	722
B.3.1	Skalarprodukt	722
B.3.2	Äußeres Product	723
B.3.3	Kreuzprodukt	723
B.4	Eigenvektoren und Eigenwerte	724
B.4.1	Berechnung von Eigenwerten	725
C	Ergänzungen zur Analysis	729
C.1	Quadratische Interpolation (1D)	729
C.1.1	Parabolische Funktion durch drei Stützstellen	729
C.1.2	Extrempunkte durch quadratische Interpolation	730
C.2	Skalar- und Vektorfelder	732
C.2.1	Jacobi-Matrix	732
C.2.2	Gradient	733
C.2.3	Divergenz	734
C.2.4	Laplace-Operator	734
C.2.5	Hesse-Matrix	735
C.3	Operationen auf mehrdimensionale, skalarwertige Funktionen (skalare Felder)	736
C.3.1	Ableitungen einer diskreten Funktion	736
C.3.2	Taylorentwicklung von Funktionen	737
C.3.3	Bestimmung lokaler Extrema von mehrdimensionalen Funktionen	741

D	Ergänzungen zur Statistik	747
D.1	Mittelwert, Varianz und Kovarianz	747
D.2	Kovarianzmatrix	748
D.3	Die Normal- oder Gaußverteilung	750
D.3.1	Maximum-Likelihood-Schätzung	751
D.3.2	Gaußsche Mischmodelle	752
D.3.3	Erzeugung von gaußverteiletem Rauschen	753
E	Gaußfilter	755
E.1	Kaskadierung von Gaußfiltern	755
E.2	Gaußfilter und Skalenraum	756
E.3	Auswirkungen des Gaußfilters im Spektralraum	756
E.4	LoG/DoG-Approximation	757
F	Java-Notizen	759
F.1	Arithmetik	759
F.1.1	Ganzzahlige Division	759
F.1.2	Modulo-Operator	761
F.1.3	Unsigned Bytes	761
F.1.4	Mathematische Funktionen (Math -Klasse)	762
F.1.5	Numerisches Runden	763
F.1.6	Inverse Tangensfunktion	764
F.1.7	Float und Double (Klassen)	764
F.2	Arrays in Java	764
F.2.1	Arrays erzeugen	764
F.2.2	Größe von Arrays	765
F.2.3	Zugriff auf Array-Elemente	765
F.2.4	Zweidimensionale Arrays	766
F.2.5	Arrays von Objekten	769
F.2.6	Sortieren von Arrays	769
	Literaturverzeichnis	771
	Sachverzeichnis	785