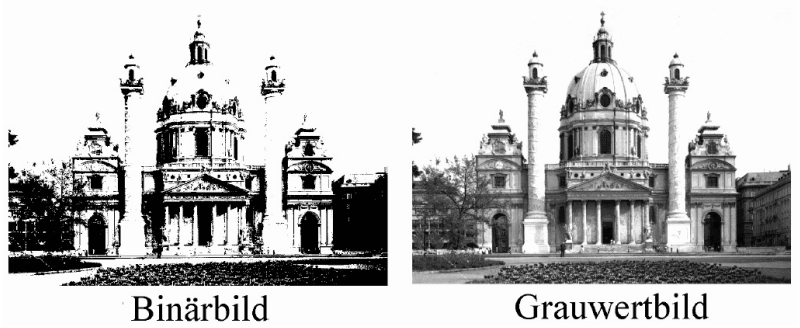
Einführung in Visual Computing
II: Grundlagen der Digitalen Bildverarbeitung
Skriptum zur Vorlesung

VU 186.822: Walter G. Kropatsch, N. Artner, M. Göbel
Institute of Computer Graphics and Algorithms 186/3 Vienna University of Technology
Pattern Recognition and Image Processing Group

SS2012 Dienstag 13:15 - 14:45
Hörsaal: Kuppelsaal / Prechtl-Saal

Grundlagen der Digitalen Bildverarbeitung
Skriptum zur Vorlesung

SS2012 Dienstag 13:15 - 14:45 Hörsaal: Kuppelsaal / Prechtl-Saal



Auflage vom 14. Mai 2012
Autoren: Walter G. Kropatsch, Nicole Artner, Max Göbel, Ines Janusch, Roxane Licandro

Die digitale Bildverarbeitung umfaßt die Methoden und Datenstrukturen, um aus digitalen Bildern Informationen zu extrahieren, die für eine Weiterverarbeitung durch Mensch oder Maschine von Bedeutung sind. Der sehr breit gestreute Anwendungsbereich (Textzeichen "lesen", Computertomographie, Fernerkundung, Überwachungsaufgaben, Robotersehen, ...) setzt die Kenntnis gemeinsamer Grundbegriffe voraus. Es werden daher die verschiedenen Arten von digitalen Bildern (S/W, Farbe, Satelliten-, Luftaufnahmen, usw.), die Formen der Darstellung von Objekten, die wichtigen Methoden zur Analyse dieser Bilder bis zur einfachen symbolischen Beschreibung des Bildinhaltes vorgestellt. Diese überarbeitete Version beinhaltet zusätzlich zahlreiche Übungsbeispiele zum vertieften Verständnis des Inhalts. Als weitere Neuerung werden einige wissenschaftliche Artikel auf der Web-Seite der Vorlesung zum Herunterladen bereitgestellt (siehe Anhang).

Auszug aus dem Studienplan Medieninformatik

Das Bachelorstudium Medieninformatik und Visual Computing vermittelt eine breite, wissenschaftlich und methodisch hochwertige, auf dauerhaftes Wissen ausgerichtete Grundausbildung, welche die Absolventinnen und Absolventen sowohl für eine Weiterqualifizierung im Rahmen eines facheinschlägigen Masterstudiums als auch für eine Beschäftigung in beispielsweise folgenden Tätigkeitsbereichen befähigt und international konkurrenzfähig macht:

- Anwendungsorientierte Forschung
- Entwicklung von Informations- und Kommunikationssystemen
- Gestaltung, Umsetzung und Evaluierung interaktiver Komponenten in Informations- und Kommunikationssystemen

...

Fachliche und methodische Kenntnisse

Das Studium vermittelt grundlegende Kenntnisse im Bereich der Informatik und ein kritisches Verständnis ihrer Theorien und Grundsätze:

- Algorithmen und Datenstrukturen

- Softwareentwicklung
- Verteilten Systemen
- Mathematik, Statistik und theoretische Informatik
- Design- und Evaluierungsverfahren

Darauf aufbauend vermittelt das Studium eine Einführung in:

- Computer Vision
- Computer Graphics
- Visualisierung,
- Augmented/Mixed/Virtual Reality
- Interaktion zwischen Menschen und Maschinen
- Aufbau von speziellen Computer- und Sensorsystemen
- Verständnis für Designdenken und Designpraxis
- Soziale, kognitive und kulturelle Grundlagen

Kognitive und praktische Fertigkeiten

Durch die praktische Auseinandersetzung mit aktuellen Technologien, Methoden und Werkzeugen (wie modernen Programmiersprachen und Entwicklungsumgebungen) werden folgende kognitiven Fertigkeiten vermittelt:

- Einsatz formaler Grundlagen und Methoden zur Modellbildung, Lösungsfindung und Evaluation
- Anwendung ausgewählter qualitativer, partizipativer und quantitativer sozialwissenschaftlicher Methoden
- Einfache interdisziplinäre und systemorientierte Herangehensweisen
- Methodisch fundierte Herangehensweise an Probleme, insbesondere im Umgang mit offenen/unspezifizierten Problemsituationen
- Entwurfs- und Implementierungsstrategien
- Kommunikation und technische Dokumentation
- Entwicklung und Umsetzung von Design-Konzepten

Soziale Kompetenzen, Innovationskompetenz und Kreativität

Der Schwerpunkt liegt hier einerseits auf der Ausbildung berufsnotwendiger Zusatzkompetenzen, und andererseits auf der besonderen Förderung hoher Kreativitäts- und Innovationspotentiale.

- Teamfähigkeit,
- Eigeninitiative und Neugierde,
- Selbstorganisation, Eigenverantwortlichkeit,
- Problemformulierungs- und Problemlösungskompetenz,

- Verantwortung in komplexen Projekten oder Tätigkeiten,
- Kenntnisse der eigenen Fähigkeiten und Grenzen, Kritikfähigkeit,
- Reflexion der eigenen Arbeit und ihrer Wechselwirkung mit dem gesellschaftlichen Kontext

Auszug aus dem Studienplan Medizinische Informatik

Das Bachelorstudium Medizinische Informatik vermittelt eine breite, wissenschaftlich und methodisch hochwertige, auf dauerhaftes Wissen ausgerichtete Grundausbildung, welche die Absolventinnen und Absolventen sowohl für eine Weiterqualifizierung im Rah- men eines facheinschlägigen Masterstudiums als auch für eine Beschäftigung in beispiels- weise folgenden Tätigkeitsbereichen befähigt und international konkurrenzfähig macht:

- Analyse und Entwicklung von Gesundheits-, Informations- und Kommunikations- systemen
- Analyse und Entwicklung von medizinischer Software
- Analyse und Entwicklung bzw. Adaption von klinischen Prozessen
- Aufbau und Management von IT-Systemen im Gesundheitswesen
- Anwendungsorientierte medizininformatische Forschung

...

Fachliche und methodische Kenntnisse

Das Studium vermittelt grundlegende Kenntnisse im Bereich der Informatik und ein kritisches Verständnis ihrer Theorien und Grundsätze:

- Algorithmen und Datenstrukturen
- Datenbanken, wissensbasierte Systeme
- Mathematik, Statistik und theoretische Informatik
- Projektmanagement
- Softwareentwicklung

Darauf aufbauend vermittelt das Studium eine Einführung in folgende Gebiete:

- Bioelektrische Signalverarbeitung
- Grundlagen der Medizin
- Life Sciences (Physik, Chemie, Biologie)

sowie

- Analyse komplexer Systeme (z.B. physiologischer Prozesse)
- Gesundheitstelematiken
- Integration neuer Technologien im Gesundheitsbereich (z.B. mobile Endgeräte)
- Qualitätsmanagement und Prozessoptimierung
- Usability und Interaktionsdesign

- Verantwortungsvoller Umgang mit Datenschutz und Datensicherheit
- Verwendung und Planung elektronischer Gesundheitsakten
- Visualisierung

Kognitive und praktische Fertigkeiten

Durch die praktische Auseinandersetzung mit aktuellen Technologien, Methoden und Werkzeugen (wie modernen Programmiersprachen und Entwicklungsumgebungen) werden folgende kognitiven Fertigkeiten vermittelt:

- Einsatz formaler Grundlagen und Methoden zur Modellbildung, Lösungsfindung und Evaluation
- Entwicklung und Umsetzung von Design-Konzepten
- Entwurfs- und Implementierungsstrategien
- Hochwertige Dokumentation und überzeugende Präsentation
- Interdisziplinäre, systemorientierte und flexible Denkweise
- Kritische Reflexion
- Methodisch fundierte Herangehensweise an Probleme, insbesondere im Umgang mit offenen/unspezifizierten Problemsituationen
- Verstehen medizinischer Prozesse

Soziale Kompetenzen, Innovationskompetenz und Kreativität

Der Schwerpunkt liegt hier einerseits auf der Ausbildung berufsnotwendiger Zusatzkompetenzen und andererseits auf der besonderen Förderung hoher Kreativitäts- und Innovationspotentiale.

- Eigeninitiative und Neugierde
- Innovationsfähigkeit durch fundiertes technisches und medizinisches Wissen
- Kenntnisse der eigenen Fähigkeiten und Grenzen, Kritikfähigkeit
- Problemformulierungs- und Problemlösungskompetenz
- Reflexion der eigenen Arbeit und ihrer Wechselwirkung mit dem gesellschaftlichen, sozialen und beruflichen Kontext
- Selbstorganisation, Eigenverantwortlichkeit
- Teamfähigkeit
- Verantwortung in komplexen Projekten oder Tätigkeiten
- Verantwortungsvoller Umgang mit Menschen, beruflichen und sozialen Gruppen in allen Tätigkeiten

Auszug aus dem Studienplan Software & Information Engineering

Das Bachelorstudium Software & Information Engineering vermittelt eine breite, wissenschaftlich und methodisch hochwertige, auf dauerhaftes Wissen ausgerichtete Grundausbildung, welche die Absolventinnen und Absolventen sowohl für eine Weiterqualifizierung im Rahmen eines facheinschlägigen Masterstudiums als auch für eine Beschäftigung in beispielsweise folgenden Tätigkeitsbereichen befähigt und international konkurrenzfähig macht:

- Entwicklung informationsverarbeitender Systeme sowohl als Experte im Team als auch in leitender Funktion
- Unterstützende Aufgaben in der Forschung

...

Fachliche und methodische Kenntnisse

Das Studium vermittelt fortgeschrittene Kenntnisse im Bereich der Informatik und ein kritisches Verständnis ihrer Theorien und Grundsätze. Aufbauend auf

- Mathematik, Statistik und theoretischer Informatik
- Algorithmen, Datenstrukturen und Programmierung
- Grundkenntnissen über den Aufbau von Computersystemen

werden die spezifischen Teilbereiche des Software- und Informationsengineering vermittelt:

- Softwareerstellung und -wartung
- Erhebung, Modellierung und Aufbereitung von Information
- Methoden der statistischen Datenanalyse
- Paradigmen und Konzepte von Programmiersprachen
- Datenbanken, wissensbasierte Systeme
- Betriebsysteme, Übersetzer und abstrakte Maschinen
- Verteilte Systeme, Internetcomputing
- Interaktion zwischen Menschen und Maschinen

Kognitive und praktische Fertigkeiten

Durch die praktische und theoretische Auseinandersetzung mit aktuellen Technologien, Methoden und Werkzeugen (wie Programmiersprachen und Entwicklungsumgebungen) werden folgende Fertigkeiten vermittelt:

- Modellbildung und Abstraktion
- Einsatz formaler Grundlagen und Methoden zur Modellbildung, Lösungsfindung und Evaluation
- interdisziplinäre, systemorientierte und flexible Denkweise
- methodisch fundierte Herangehensweise an Probleme, insbesondere im Umgang mit unspezifizierten Problemsituationen
- kritische Bewertung und Reflexion von Lösungen
- hochwertige Dokumentation und überzeugende Präsentation

Soziale Kompetenzen, Innovationskompetenz und Kreativität

Der Schwerpunkt liegt einerseits auf der Ausbildung berufsnotwendiger Zusatzkompetenzen, und andererseits auf der besonderen Förderung hoher Kreativitäts- und Innovationspotentiale.

- Selbstorganisation, Eigenverantwortlichkeit
- Teamfähigkeit im globalisierten Umfeld
- Eigeninitiative und Neugierde
- Finden kreativer Problemlösungen
- Entscheidungsverantwortung und Führungskompetenz in komplexen Projekten oder Tätigkeiten
- Kenntnisse der eigenen Fähigkeiten und Grenzen, Kritikfähigkeit
- Reflexion der eigenen Arbeit und ihrer Wechselwirkung mit dem gesellschaftlichen Kontext

Kapitel A

Einleitung

A.1 EVC Überblick

Die VU Einführung in Visual Computing ist die einzige Lehrveranstaltung des Moduls Visual Computing und wird im Sommersemester 2012 erstmals abgehalten, gemeinsam von drei Professoren der Informatik. Inhaltlich werden 3 Themenbereiche des Visual Computing getrennt aber koordiniert präsentiert:

1. Bildverarbeitung (W. Kropatsch)

2. Computergraphik (W. Purgathofer)

3. Computer Vision (R. Sablatnig)

Grundsätzlich ist die Lehrveranstaltung eine VU (Vorlesung mit Übung), das heißt, dass die Beurteilung sich aus dem Absolvieren von Übungsbeispielen und den Ergebnissen von Tests zusammensetzt. Auf Grund der besonderen Struktur dieser VU werden die 3 Themenbereiche erkennbar getrennt vorgetragen und auch beurteilt. Dies ändert aber nichts daran, dass die VU nur zu einer einzigen Gesamtnote führt, dass die einzelnen Teile nicht getrennt voneinander angerechnet werden (Ausnahmen gibt es in der Übergangsregelung zum alten Studienplan) und vor allem, dass die VU nur als Ganzes absolviert werden kann. Es gibt daher keine Beurteilungsmöglichkeit außerhalb des Semesters, in dem die Lehrveranstaltung stattfindet (normalerweise einmal im Jahr im Sommersemester).

A.2 Termine, Themen

Die folgenden Termine beziehen sich ausschließlich auf Teil I *Bildverarbeitung* des Kurses *Einführung in Visual Computing*, SS 2012 (ausschließlich der Prüfungstermine).

Datum	Thema		Seite
5.03.2012	Einleitung	A	7
6.03.2012	Digitales Bild, Bildqualität, Störungen	B	13
20.03.2012	Histogramm, Threshold, Qualitätsverbesserungen	C	27
27.03.2012	Bildoperationen, Filter	D	37
28.03.2012	Segmentation, Kanten erkennen, Hough Transformation	E	51
17.04.2012	Überblick Bildanalyse	F	71
25.04.2012	Prüfung EVC Test 1		
8.05.2012	Binäre Bilder, Math.Morphologie	G	85
15.05.2012	Bild- u. Objektdarstellung, Datenstrukturen	H	101
22.05.2012	Bildpyramide	I	111
5.06.2012	Bildbeschreibung, Eigenschaften	J	135
18.06.2012	Prüfung EVC Test 2		
27.06.2012	Prüfung EVC Test Wiederholung		

A.3 Übungen

Unterrichtsbegleitend werden das gesamte Semester hindurch Übungen angeboten (in den Labors der Favoritenstraße, für genaue Raumreservierung siehe TISS/TUWEL). Ziel der Übungen ist, das theoretische Wissen aus den Vorlesungen praktisch zu erfahren und zu vertiefen.

Die Übungen sind mit dem *Matlab Toolkit* durchzuführen, welches gegen eine geringe Lizenzgebühr vom Studenten Sofware Service (http://www.sss.tuwien.ac.at/sss/) bezogen werden kann. Außerdem ist die Software auf den Terminalcomputern der entsprechenden Labors vorinstalliert. Am 14.03.2012 findet eine Einführungsveranstaltung zu Matlab in den Informatiklabors statt.

A.4 Bewertung

Im Rahmen der VU sind 6 Übungsbeispiele abzugeben und 2 Tests zu absolvieren. Die Übungsbeispiele müssen alle positiv bewertet sein und die Punktesumme aus den beiden Tests muss mindestens 50% der erreichbaren Punkte ausmachen. Details zu den Übungsbeispielen erhalten Sie bei der Vorbesprechung und (anschließend) auf den oben genannten Webseiten der drei Vortragenden.

Die Tests finden am 25.04.2012 und am 18.06.2012 von 13 bis 15 Uhr statt, die Hörsaaleinteilung wird rechtzeitig bekannt gegeben. Am 27.06.2012 findet ein Ersatztest statt. Wenn man aus welchem Grund auch immer (Krankheit, Bundesheer, persönlichen Gründen, ...) einen Testtermin versäumt hat, dann kann man diesen beim Ersatztest nachholen. Aber auch wenn Sie beide Tests absolviert haben können Sie zum Ersatztest antreten wenn Sie sich verbessern wollen. Es wird dann auf jeden Fall der schlechtere der beiden Tests durch das Ersatztestergebnis ersetzt (auch wenn der Ersatztest schlechter ausfällt!). Details zur Anmeldung hierzu gibt es rechtzeitig vor dem Ersatztest.

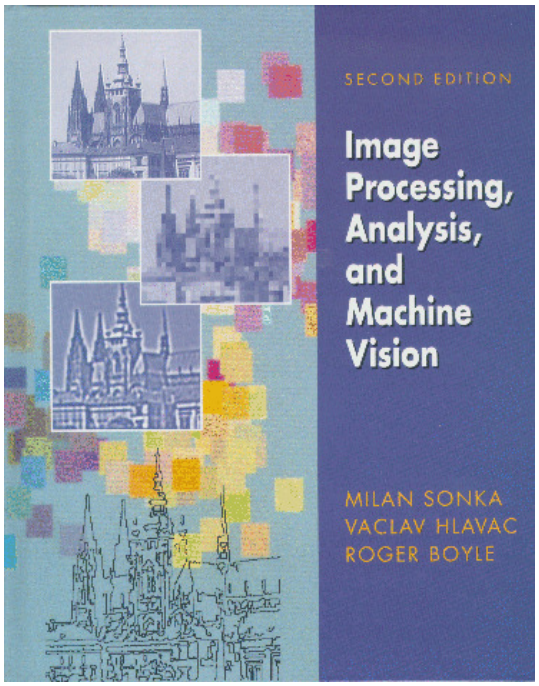
A.5 Errata

Leider sind Fehler in Skripten bei aller Sorgfalt nie ganz auszuschließen. Sollten inhaltliche Fehler gefunden werden, so bitten wir diese per Mail an `gdbv@prip.tuwien.ac.at` zu senden. Bei Korrektheit und Angabe einer Korrektur werden Zusatzpunkte vergeben, die sich positiv auf die Bewertung des Übungsteils auswirken.

A.6 Literatur

A.6.1 Bücher

Empfohlenes Buch



1993,
Chapman & Hall,
Computing Series,
London.

Bücher/Kapitel zur digitalen Bildverarbeitung

- A. Rosenfeld, A.C. Kak: Digital Picture Processing; Academic Press, 1982,Vol 1: Kap.6,Vol. 2: Kap.10, 11; mathematische Grundlagen.
- Rafael C. Gonzales and Richard E. Woods:*Digital Image Processing.*Prentice Hall, Inc., Third Edition, 2008.
- W.K. Pratt: Digital Image Processing; John Wiley, 1978, Part5.
- Klaus D. Tönnies:*Grundlagen der Bildverarbeitung.*Pearson Education, Inc., 2005. ISBN 3-8273-7155-4.
- Wilhelm Burger and Mark James Burge:*Digitale Bildverarbeitung, eine Einführung mit Java und ImageJ.*Springer Verlag, Berlin, Heidelberg, 2005, 2006. ISBN-10 3-540-30940-3.
- J. Serra: Image Analysis and Mathematical Morphology; Academic Press, 1982, Kap. II.

Weitere Bücher/Kapitel

- Reinhard Klette and Azriel Rosenfeld;*Digital Geometry, Geometric Methods for Digital Picture Analysis.*Morgan Kaufmann/Elsevier, 2004.
- Ballard, Brown: Computer Vision, http://homepages.inf.ed.ac.uk/rbf/BOOKS/BANDB/bandb.htm

- John R. Jensen:*Introductory Digital Image Processing.*Pearson, Prentice Hall, Inc., third edition, 2000.ISBN 0-13-145361-0.
- H. Bässmann, Ph.W. Besslich: Bildverarbeitung Ad Oculos;Springer-Verlag, Berlin, Heidelberg 1991; Überblick und Programmierung (C).
- Nick Efford:*Digital Image Processing-a practical introduction using JAVA.*Pearson, Addison Wesley, first edition, 2000.ISBN 0-201-59623-7.
- Rafael C. Gonzales and Richard E. Woods and Steven L. Eddins:*Digital Image Processing USING MATLAB.*Prentice Hall, Inc., first edition, 2002.

Alle hier aufgelisteten Bücher stehen in der institutseigenen Bibliothek zur Verfügung.

A.6.2 Journale

CVIU: Computer Vision and Image Understanding.

IP: IEEE Transations on Image Processing

PAMI: IEEE Transactions on Pattern Analysis and Machine Intelligence.

PAA: Pattern Analysis and Applications,

PR: Pattern Recognition, Pergamon Press.

PRL: Pattern Recognition Letters, North Holland.

A.6.3 Weblinks

Umfangreiche Liste+andere Kurse auf:

- http://www.prip.tuwien.ac.at/teaching/186822
- Keith Price's bibliography http://iris.usc.edu/Vision-Notes/bibliography/contents.html
- IAPR Education Committee http://homepages.inf.ed.ac.uk/rbf/IAPR/index.php

A.6.4 Computer Vision Tele-Course

Computer Vision Tele-Course:

Course Unit Titles and Outlines

- Module 1: Computer Vision, an introduction
- Module 2: Computer Vision, an eye for an eye?
- **Module 3: Computer Vision, image preprocessing**

- Module 4: Computer Vision, object shapes
- Module 5: Computer Vision, on the move
- Module 6: Computer Vision, what's in a frame?

Lecturers:
Prof.Dr.Ir.Luc Van Gool Dr.-Ing.Reinhard Koch

A.6.5 Englischen Begriffsdefinitionen (Glossaries)

findet man in:

- R.M. Haralick, L.G. Shapiro: Glossary of Computer Vision Terms. *Pattern Recognition*, Vol.24, No.1, pp.69–93, 1991.
- http://en.wikipedia.org/wiki/Main_Page

A.7 Lehrziel

- Verständnis der Grundbegriffe und
- der Zusammenhänge zwischen
  Kamera, Objekt und seinem Bild.
- Erlernen der Fähigkeit,
  einfache Fachartikel zu verstehen,
- Erkennungsalgorithmen auswählen,
- Objekte in einem digitalen Bild erkennen.

Kapitel B

Digitales Bild, Bildqualität, Störungen

B.1 Bildentstehung

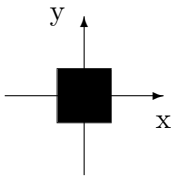
Ein von einer Lichtquelle bestrahltes Objekt reflektiert je nach dessen Oberflächenbeschaffenheit das Licht (diffuse Streuung), welches von einem Detektor detektiert werden kann (z.B. menschliches Auge, CCD Kamera, etc.). Es findet eine sogenannte perspektivische Projektion von der 3D Szene der realen Welt, in der sich das Objekt befindet, in ein 2D Bild, in welchem die reflektierten Lichtintensitäten abgebildet werden, statt. (Siehe Abbildung B.1 und B.2)

B.2 Bildentstehung: mathematisches Modell

- Einfallswinkel α = Winkel zwischen Oberflächennormale und Einfallsstrahl (siehe Abbildung B.3)
- Ausfallswinkel β = Winkel zwischen Oberflächennormale und Bildstrahl (siehe Abbildung B.3)
- Reflexionsgesetz: Einfallswinkel = Ausfallswinkel
- 1 Bildpunkt sammelt Licht (Photonen) aller Strahlen, die ihn innerhalb einer Zeiteinheit treffen.
- Der Einfluß der einfallenden Strahlen ist am Reflexionspunkt am stärksten und nimmt mit zunehmender Entfernung von diesem ab.
- Beim Auftreten einer perfekten Reflexion des Lichtes kann die Punktstreufunktion (PSF) als Delta Dirac Impulsfunktion $\delta()$ angesehen werden. Sie stellt, wenn man sie als Bild betrachtet, einen einzigen hellen Punkt (mit dem Wert 1 am Ursprung des Koordinatensystems) dar und ist von einer unendlichen, schwarzen Fläche umgeben.

  Wir beginnen mit der Definition der Rechtecksfunktion:

$$\text{rect}(x,y) = \begin{cases} \mathbf{1} & \text{für } |x| \leq \frac{1}{2} \text{ und } |y| \leq \frac{1}{2} \\ 0 & sonst \end{cases}$$



Für die Delta Dirac Funktion gilt

$$\int_0^\infty \int_0^\infty \text{rect}(x,y)dxdy = 1,$$

was interpretiert wird als immer kleinere Fenster (Rechtecke) bei gleichbleibender Gesamtintensität.

$$\delta_n(x,y) = n^2\text{rect}(nx,ny), \quad n = 1,2,\ldots.$$

Die Delta Dirac Funktion ergibt sich daher als Grenzwert von

$$\delta_n : \delta(x,y) = \lim_{n\to\infty} \delta_n(x,y)$$

mit und hat folgende Eigenschaften:

1. $\delta(x,y) = 0$ für $x \neq 0, y \neq 0$
2. $\delta(0,0) = \infty$
3. $\int_0^\infty \int_0^\infty \delta(x,y)dxdy = 1$

Die Delta-Dirac Funktion kann als Filterfunktion eines Bildes verwendet werden. Wird im Falle der perfekten Reflexion die PSF mit dem Bild gefaltet, erhält man wieder das unveränderte Bild. Im umgekehrten Fall wenn man als Input δ verwendet und mit einem beliebigen Filter H faltet, erhält man den Filter H (Kommutativität der Faltung).

$$H * \delta = \delta * H = H(H \ldots linearerFilter) \tag{B.1}$$

Dieses Verhalten kann genutzt werden um Filter (z.B. Punktstreufunktion (PSF)) mit unbekannten Eigenschaften oder Parametern zu messen, sofern es sich um einen linearen Filter handelt. Mittels anschließender Faltung können aus einem einzigen Impuls Informationen extrahiert werden. Das Resultat wird auch als Impulsantwort bezeichnet, welches in optischen Systemen als Lichtverteilung angesehen wird und als Punktstreufunktion bezeichnet wird. Bei Betrachtung der PSF im Falle nicht perfekter Reflexion wird folgendes Berechnungsmodell verwendet:

Voraussetzung: Wir bilden ein zweidimensionales Objekt $R(x',y')$ auf ein zweidimensionales Bild $B(x,y)$ ab.

$$B(x,y) = PSF * R(x',y') \tag{B.2}$$

Die Faltung der Terme PSF und R(x',y') in Integralschreibweise ergibt:

$$B(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} PSF(x-x',y-y')R(x',y')dxdy \tag{B.3}$$

- Die diffuse Streuung wird mit Gauss-ähnlicher Gewichtsfunktion modelliert.

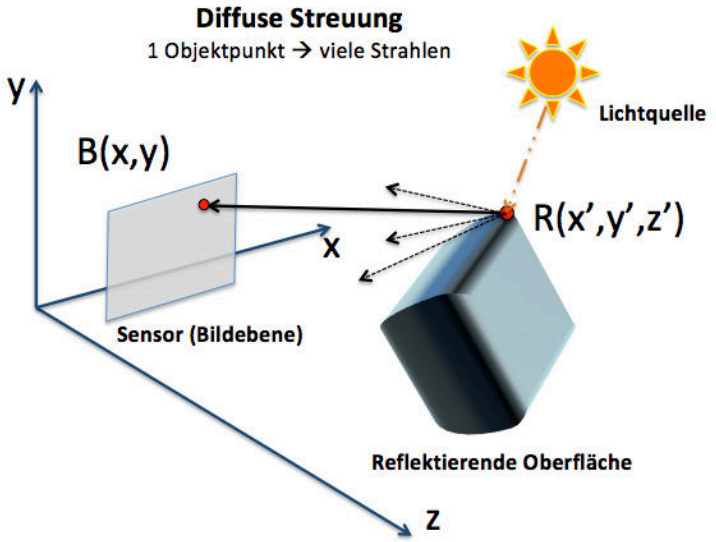


Abbildung B.1: In dieser Grafik sei $R(x', y', z')$ ein Punkt eines Objektes im dreidimensionalen Raum, mit den Koordinaten x', y' und z', welcher in den 2D Raum (Bildebene) projiziert wird, $B(x, y)$ und die Koordinaten x und y aufweist. Es tritt eine diffuse Streuung auf, welche aus einem Objektpunkt viele Bildpunkte entstehen lässt.

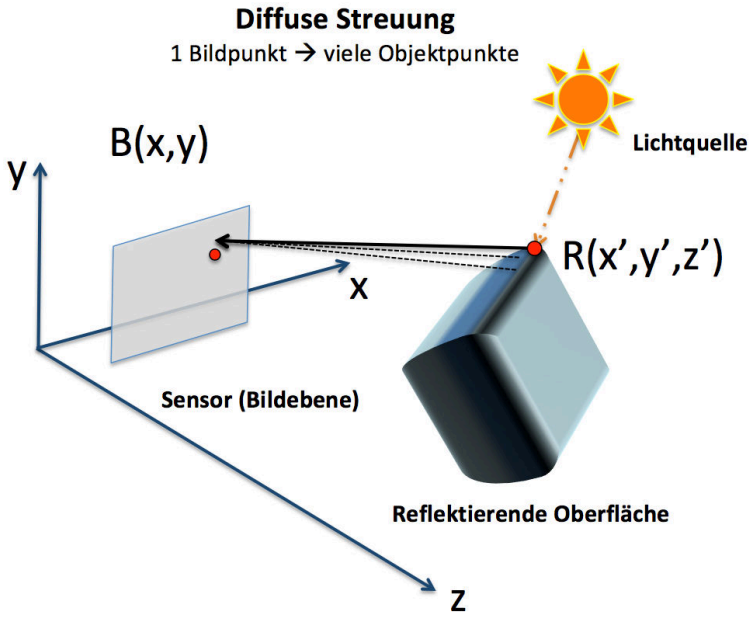


Abbildung B.2: Im Gegensatz zur Abbildung B.1 ist die diffuse Streuung von der Sicht der Bildebene gezeigt, hier entspricht ein Bildpunkt vielen Objektpunkte. Das Ausmaß dieses Verhaltens wird durch die Punktstreufunktion - PSF (engl. point spread function)beschrieben.

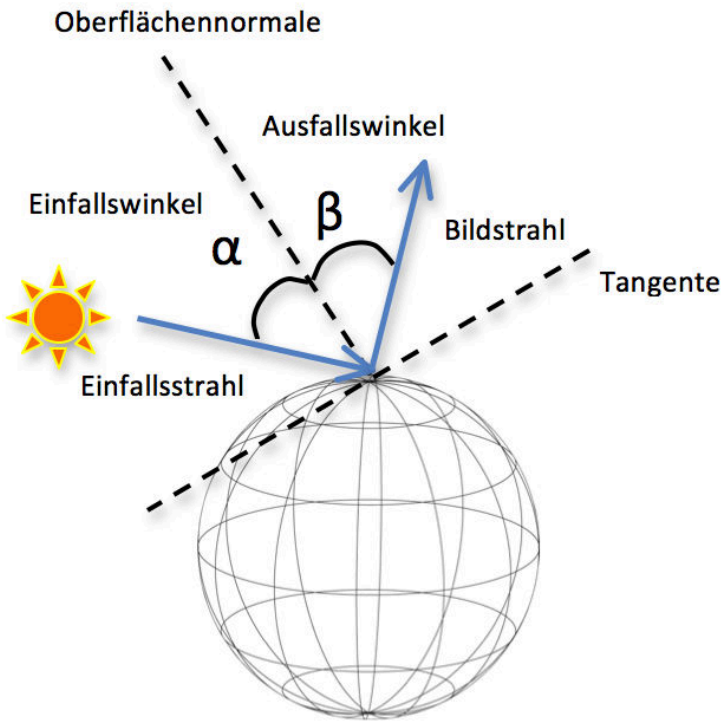


Abbildung B.3: Einfallswinkel α und Ausfallswinkel β bei Beleuchtung eines Objektes

B.3 Digitales Bild

- Der Trick bei digitaler Speicherung eines Bildes besteht darin, eine Darstellung zu erzeugen, die sehr nahe am analogen Original ist.
  - Wenn die Messwerte genau genug sind (z.B. Thermometer, das 37.4319 Grad anzeigt) und oft genug wiederholt werden, dann stellen sie die originale, analoge Information hinreichend genau dar.
- Ein digitales Bild ist eine Anordnung von Punkten (Pixel), wobei jeder Punkt einen Grau- oder Farbwert hat (siehe Abbildung:B.4).
  - Ähnlich dem Zeichnen eines Bildes durch Einfärben eines Gitters, wobei jedes Quadrat des Gitters nur mit einer einzigen Farbe gefüllt werden kann.
  - Wenn die Quadrate klein genug sind und eine große Auswahl von Farben verfügbar ist, ergibt die Zeichnung ein gutes Abbild der Realität.

B.4 Shannon'sches Abtasttheorem

Das analoge Originalbild wird durch eine kontinuierliche Bildfunktion f(x,y) beschrieben. Um ein Bild zu digitalisieren (Bildung einer diskreten Bildfunktion) verwendet man ein diskretes Abtastgitter, das an definierten Punkten das Bild abtastet:

$x = j\Delta x,\ j = 1, \ldots, m$
$y = k\Delta y,\ k = 1, \ldots, n$

Das Shannon'sche Abtasttheorem besagt:

Das Abtastintervall $(\Delta x, \Delta y)$ sollte mindestens halb so klein gewählt werden wie das kleinste interessante Bilddetail.

Wird dieses Theorem missachtet kommt es zu dem sogenannten Aliasing Effekt (siehe Abschnitt B.5)

B.5 Aliasingbeispiel

In Abbildung B.5 sind in den 3 Grafiken Sinuskurven verschiedener Frequenz aufgetragen. Jede Kurve wurde mit denselben 8 Intervallen abgetastet. Im linken Bild tritt kein Aliasing Effekt auf. Im mittleren Bild kommt der Aliasing Effekt zur Geltung: Durch die Auswahl einer zu niedrigen Abtastfrequenz werden sogenannte Scheinfrequenzen abgetastet - die zu einer Verfälschung des digitalen Signales führt. In der rechten Abbildung sieht man einen zum mittleren Bild gesteigerten Aliasing Effekt. Hier wird anstatt der tatsächlichen hochfrequenten Schwingung eine niederfrequente Schwingung als Signal registriert.

0	100	0	102	0	0	0	38	47	0	87	0	55
139	0	0	0	173	122	113	244	142	0	0	0	61
0	0	3	228	181	228	126	80	255	134	38	0	80
0	0	255	255	163	201	255	209	61	207	255	61	103
0	0	255	207	232	255	148	255	172	255	255	187	0
0	0	255	255	255	107	204	255	86	204	221	0	0
90	0	255	134	202	162	196	255	255	173	197	0	0
0	0	156	170	232	255	255	255	58	255	255	38	0
0	0	0	255	195	255	241	233	118	104	16	230	0
0	7	0	0	224	254	95	109	135	139	145	2	59
0	0	0	0	0	0	111	20	45	19	14	0	0
30	29	0	10	128	0	0	0	58	67	0	46	0

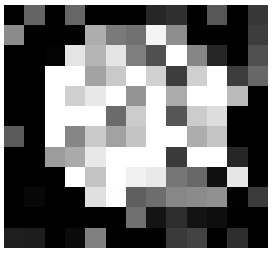
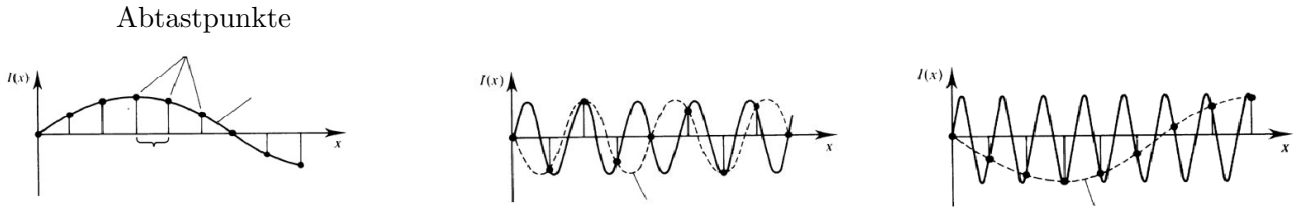


Abbildung B.4: Ein digitales Bild als eine Anordnung von Punkten (Pixel) am Beispiel eines Grauwertbildes



kein Aliasing,
3/4 Schwingung

Aliasing Kurve
5 Schwingungen

Aliasing Kurve
8 Schwingungen

Abbildung B.5: Auftreten von Aliasing bei Missachtung des Shannon-Theorems

B.6 Bildaufnahme

Die Bildaufnahme lässt sich in den Bereich der analogen und digitalen Bildaufnahme gliedern. Der Übergang zwischen dem analogen und digitalen Bereich erfolgt durch die Analog- /Digital- (A/D) Wandlung. Im analogen Bereich werden Lichtreize mittels eines Sensors detektiert, der in einer CCD Kamera, Digitalkamera oder einem mobilen Gerät liegt. Das gewonnene analoge Signal der CCD Kamera muss durch einen externen A/D - Wandler digitalisiert werden, Digitalkamera und Mobiltelefone haben diesen integriert. Die Digitalisierung ist notwendig um weitere Verarbeitungsschritte mittels eines Computers durchführen zu können. Das Ergebnis dessen kann durch Abspeicherung auf Medien verschiedener Art und Speichergröße gesichert werden. Details siehe Abbildung B.6

Abbildung B.6: Bildaufnahme

B.7 Die Natur des Lichtes [21]

Licht besitzt sowohl Teilchen- als auch Welleneigenschaften, welche in der Physik durch experimentelle Versuche erforscht wurden. Dieses Phänomen wird in der Quantenphysik "Welle - Teilchen - Dualismus" genannt. Das Teilchenmodell benennt Lichtteilchen als Photonen, welche keine Masse besitzen und sich mit Lichtgeschwindigkeit c (c = 299.792,46 km/s) unabhängig vom Medium und Bewegungszustand des Betrachters fortbewegen.
Das in der Technik und Natur vorkommende Licht besitzt unterschiedliche Wellenlängen, die sich im für den Menschen sichtbaren Bereich und unsichtbaren Bereich befinden. Das menschliche Auge kann Wellenlängen im Bereich 380 nm (rot) bis 790 nm (blau) wahrnehmen und ist im grünen Bereich (≈ 480 nm - 560 nm) am empfindlichsten.

Wellenlängen < 380 nm umfassen z.B. die Bereiche:

Ultraviolettestrahlung (1nm - 380 nm)

Röntgenstrahlung(1 pm -10 nm)

Gammastrahlung (< 5 pm)

Wellenlängen > 770 nm umfassen z.B. die Bereiche:

Infrarotstrahlung (780 nm - 1 mm)

Radar(1 m-10 m)

Radiowellen (1 m - 10 km)

Details siehe Abbildung B.7

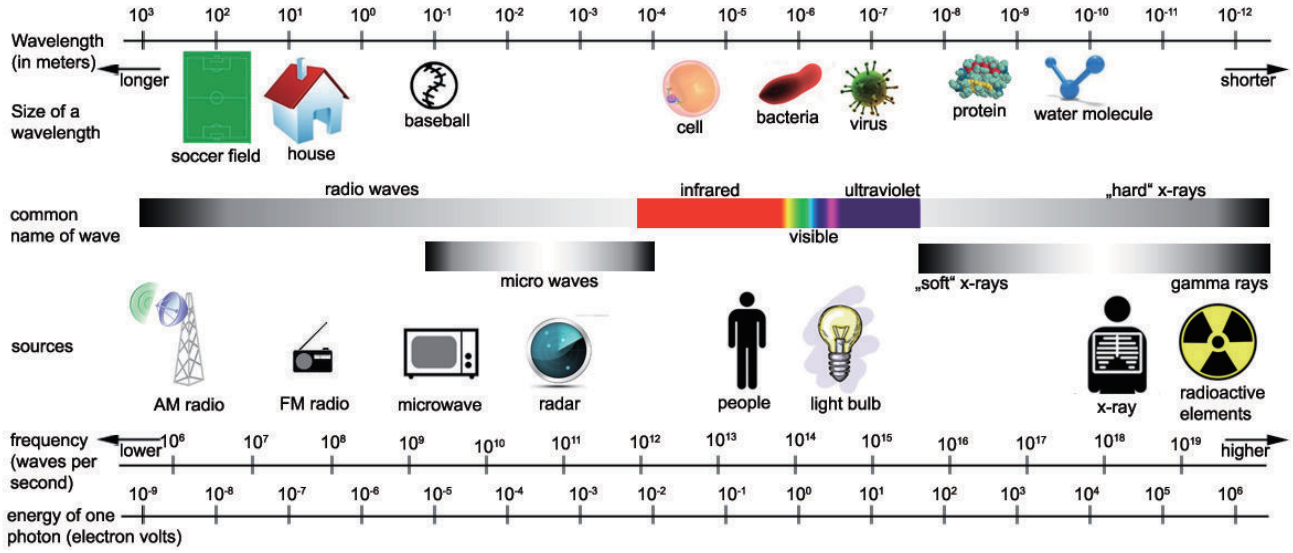


Abbildung B.7: Lichtspektrum

B.8 Sensortechnologie

Die Quantenschwankung kann als **Rauschen (noise)** betrachtet werden. Signale, die sich in einer Variation in Intensität und Wellenlänge auszeichnen, können von einem Rauschsignal überlagert werden,

welches die Qualität des Nutzsignals senkt. Das Maß für die Qualtität eines Signals wird als Signal-Rauschabstand (engl. signal to noise ratio - SNR) bezeichnet und definiert das Verhältnis der mittleren Leistung des Nutzsignals zur mittleren Rauschleistung des Störsignals.
Die Qualität des Signals kann durch Mittelung über größere Flächen verbessert werden, jedoch steht dieser der Verlust der räumlichen Auflösung gegenüber. Eine weitere Möglichkeit wäre die Mittelung über grössere Zeitintervalle, dem der Verlust der zeitlichen Auflösung gegenübersteht. Ziel ist es einen Kompromiss zwischen diesen beiden Aspekten zu finden. In Abbildung B.8 ist dieses zusammenfassend dargestellt.

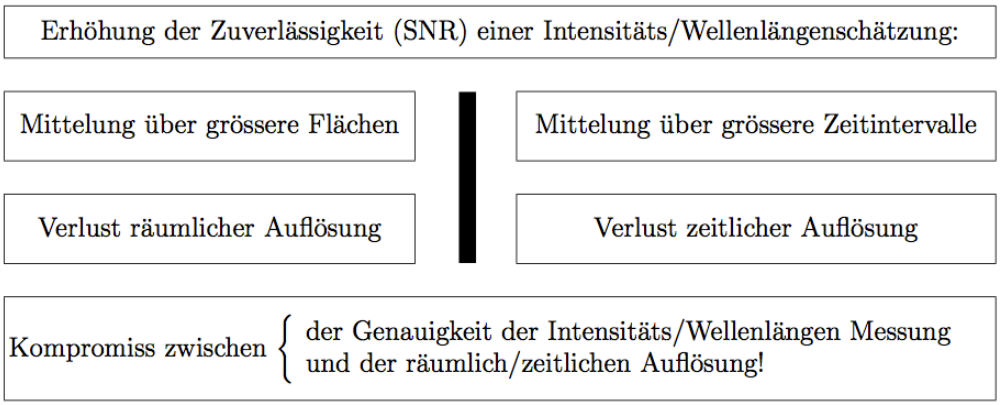


Abbildung B.8: Verbesserung der Qualität von Nutzsignalen

B.9 Farbsehen und Farbkameras

Das menschliche Auge besitzt auf der Netzhaut (Retina) Helligkeitsrezeptoren (Stäbchen) und 3 Arten von Farbrezeptor-Zellen (Zapfen). Bei den Zapfenarten ist jede einzelne gegenüber verschiedenen Lichtwellenlängen (Farben) empfindlich. Grob gesehen können Empfindlichkeiten der Zapfenzellen in die Farbbereiche rot, grün und blau eingeteilt werden.
Das Auge nimmt Farbe als Kombination der Primärfarben wahr, welche im Falle der additiven Mischung die Lichtfarben Rot, Grün und Blau sind. Für die sogenannte subtraktive Mischung sind als Primärfarben die Körperfarben Cyan, Gelb und Magenta zu verstehen.
TV-Geräte funktionieren durch Verwendung von 3 verschiedenen Elektronenstrahlen, welche beim Auftreffen auf bestimmte Phosphorarten die Emittierung von rotem, grünem und blauem Licht auslösen.
Das RGB-Format (Rot Grün Blau - Format / RGB-Farbraum), beinhaltet den Umfang der Farben, die durch additive Mischung der drei Grundfarben (Rot, Grün und Blau) nachgebildet werden können und ist Geräte abhängig. Computer speichern im allgemeinen Bilder im RGB-Format. (Andere Formate wären z.B. der CMYK - Farbraum (**C**yan **M**agenta **Y**ellow **K**ey plate)- subtraktives Farbmodell für den Vierfarbdruck).

Abbildung B.9 (links) zeigt die Aufnahme einer Bildszene mittels einer Kamera mit Farbfiltern, welche die aufgenommene Szene in drei Farbbereiche Rot Grün Blau aufteilt und diese als Schwarz/Weiß Bilder darstellt. Bei einer Projektion (rechts) dieser Bilder, wird durch entsprechende Farbfilter, eine additive Mischung der einzelnen Farbkomponenten des Lichtes durchgeführt, sodass der selbe Farbeindruck wie bei einer “ reinen“ Spektralfarbe entsteht.

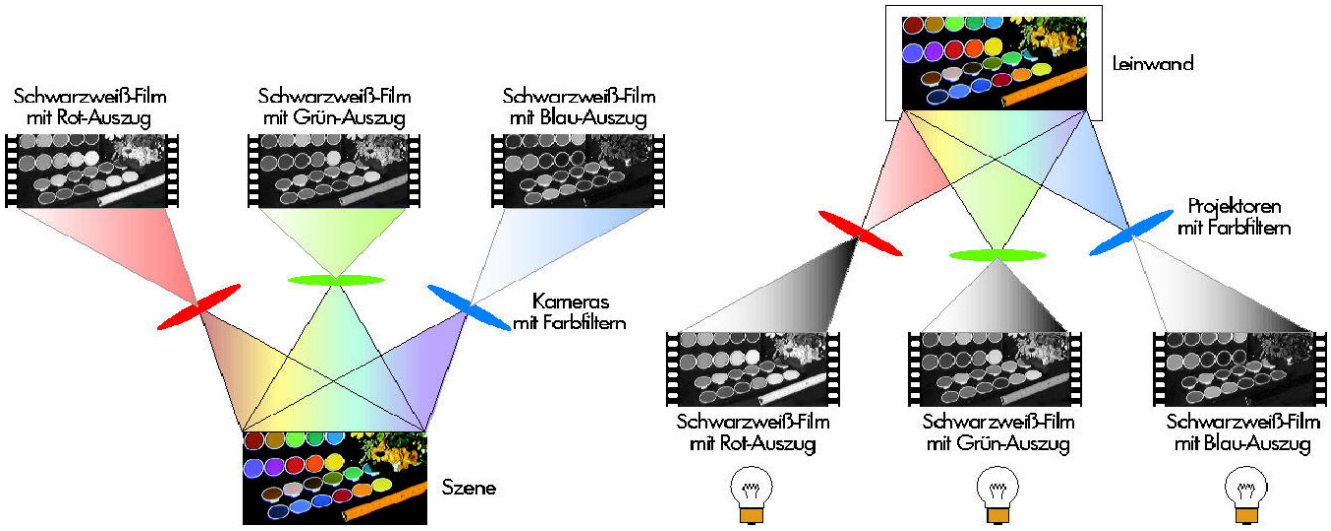


Abbildung B.9: Additiv gemischtes Licht aus veränderlichen Rot-, Grün- und Blau-Anteilen bewirkt im menschlichen Auge denselben Farbeindruck wie eine 'reine' Spektralfarbe.

B.10 Bildqualität

Die Qualität eines Bildes kann durch folgende Parameter spezifiziert werden:

1. Schärfe (engl. acutance, edge sharpness)
2. Kontrast (engl. contrast)
3. Auflösung (engl. resolution)
4. Störungen/Rauschen (engl. noise)

B.10.1 Schärfe

Die Genauigkeit der Darstellung von Bilddetails wird als Schärfe bezeichnet. Die Schärfe kann am Kantenprofil eines Bildes abgelesen werden, je steiler der Anstieg des Profiles ist, desto schärfer ist die Darstellung der Kanten im Bild bzw. je flacher der Anstieg ist desto unschärfer ist das Bild.

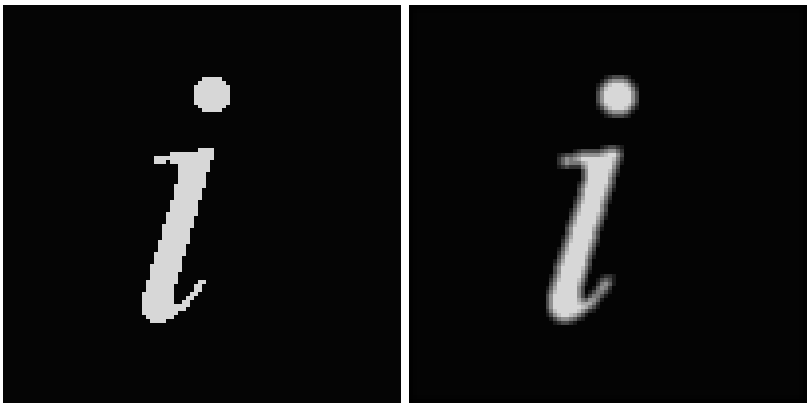
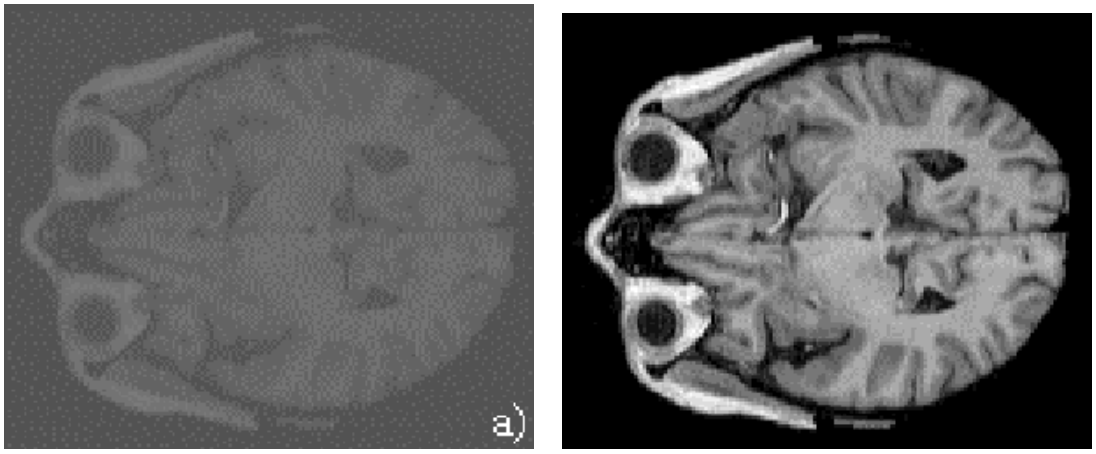


Abbildung B.10: Auf dem linken Bild wird der Buchstabe " i " scharf dargestellt, zum Vergleich ist ein unscharfes Bild des selben Motives auf der rechten Seite der Abbildung zu sehen (Kanten sind verschmiert, Details sind nicht genau erkennbar)

B.10.2 Kontrast

Als Kontrast wird die Differenz bezeichnet, die zwischen maximalen und minimalen Pixelwert eines Bildes auftritt. Der Kontrast ist auch als die lokale Änderung der Helligkeit zu verstehen - als das Verhältnis zwischen durchschnittlicher Helligkeit eines Objektes und der Helligkeit des Hintergrundes. Je höher der Kontrast ist desto effektiver wird der Bereich von Intensitätsstufen in einem Bild ausgenutzt. Er kann aus einem sogenannten Histogramm (siehe Abschnitt C.1) abgelesen werden.

Abbildung B.11: Schnittaufnahmen eines menschlichen Gehirnes



(a) Schnittaufnahme eines menschlichen Gehirnes mit niedrigem Kontrast (Wertebereich: 82 bis 255)

(b) Im Vergleich zur Abbildung B.12(a) ist ein kontrastreicheres Bild desselben Motives zu sehen (Objekt hebt sich deutlich vom Hintergrund ab - Wertebereich: 0 bis 255)

B.10.3 Auflösung

Als Auflösung eines Bildes bezeichnet man die tatsächliche Größe bzw. räumliche Ausdehnung eines Pixels in der realen Welt. Als Einheit der Auflösung wird die Anzahl der Bildelemente pro Längeneinheit angegeben. In der Literatur sind z.B. dpi (dots per inch), lpi (lines per inch) oder Pixel pro Kilometer (bei Satellitenfotos) zu finden.

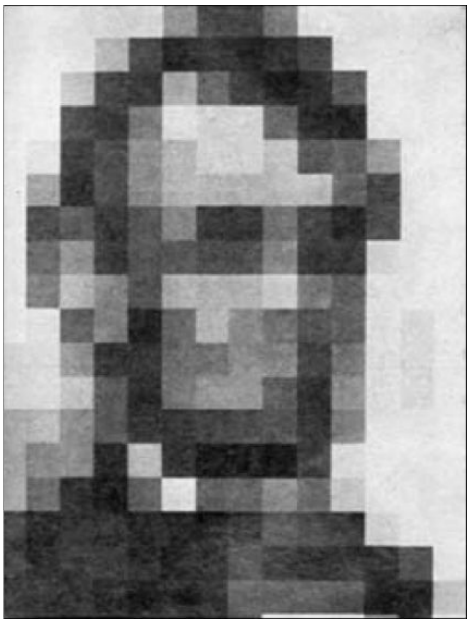


Abbildung B.12: Leon Harmon of Bell Labs: picture of Lincoln (252 pixels), The Recognition of Faces, Scientific American, (Nov. 1973).

Abtastung und Objektgrösse

Beim Anzeigen eines Bildes auf z.B. einem Display können sogenannte Bildartefakte entstehen. Dies geschieht, wenn Bild und Display eine unterschiedlich große Auflösung besitzen. In Abbildung B.13 soll ein Bild mit einer Pixelgröße von $1mm^2$ auf ein z.B. Display mit einer Pixelgröße von $9mm^2$ abgebildet werden. In der Tabelle B.1 ist oben eine LUT (Look up Table), welche angibt mit welchen Grauwerten ein Bild am Display dargestellt werden soll, d.h. jedem Grauwert wird ein Anzeigegrauwert zugewiesen. Da für das angeführte Beispiel die Pixelanzahl um einen **Faktor 3** reduziert werden muss (um eine korrekte Anpassung zu vollziehen), bildet man zunächst Mittelwerte jeder **Dreier**gruppe, welches in der Tabelle B.1 unten zu sehen ist. (In Abbildung B.13 entspräche dies einer Mittelwertbildung über 3×3 Pixel).
Durch die Mittelung erhält man einen Glättungseffekt - welcher Grauwerte zwischen 3 und 0 auftreten lässt. Diese Werte müssen auf die Ausgabe Werte angepasst werden, welches über die LUT erfolgt. Durch die Mittelung geht auch Bildinformation verloren, welches in Abbildung B.14 zu sehen ist (z.B. Striche verschwinden gänzlich oder werden verschmälert dargestellt).

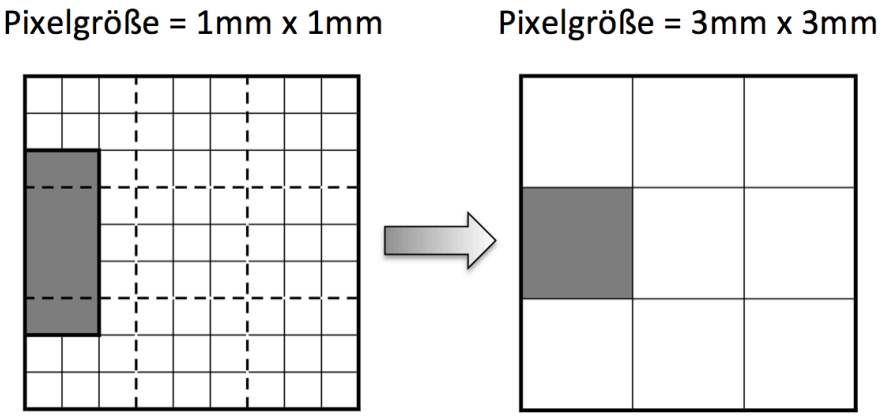


Abbildung B.13: Problematik der Anpassung von verschiedenen Auflösungen

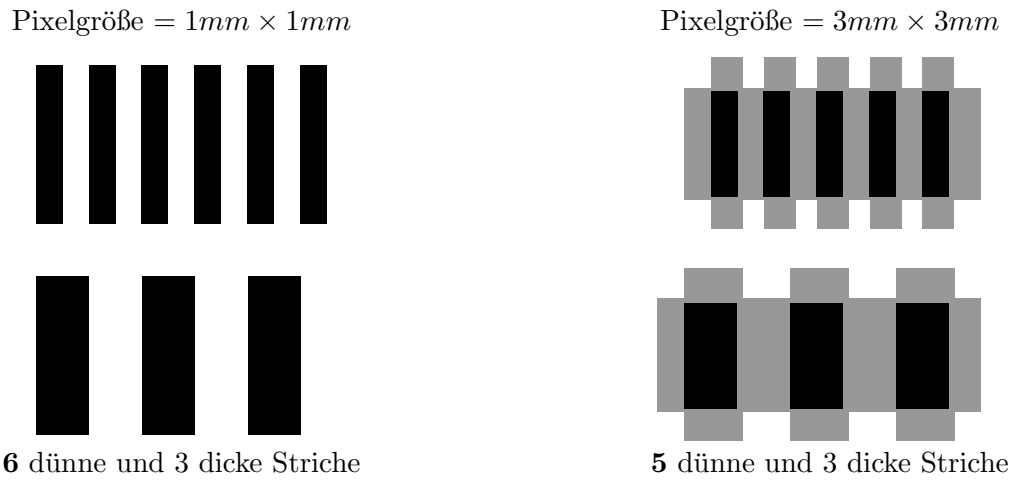


Abbildung B.14: Problematik der Anpassung von verschiedenen Auflösungen

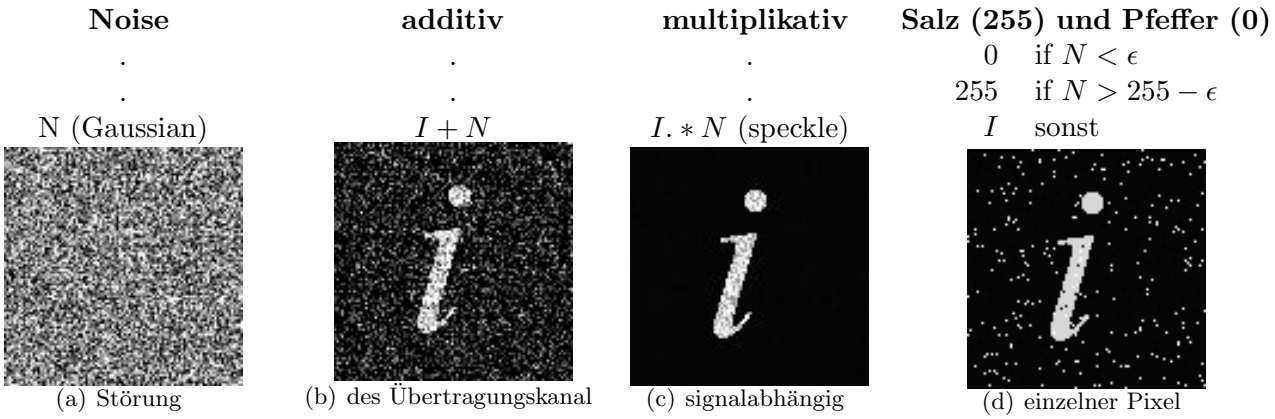
B.10.4 Störungen, Rauschen

Die Qualtität eines Bildes kann durch Störungen bzw. Rauschen herabgesetzt werden. Abbildung B.15(a) zeigt eine Störung durch **Gaußsches Rauschen**. **Additive Störungen** können bei einer fehlerhaften Übertragung über einen Übertragungskanal auftreten (siehe Abbildung B.15(b)), indem dem Bild additiv ein Störfaktor hinzugefügt wird. In Abbildung B.15(c) ist die **multiplikative Störung** abgebildet. Hier wird das Bild mit einem Störsignal im Bildbereich gefaltet, welches einer Multiplikation im Frequenzbereich gleichkommt (Faltungstheorem) - daher der Name multiplikative Störung. Sie tritt z.B. bei Messungen mittels Radar auf. Auch eine **Störung einzelner Pixel** ist möglich und im Bild B.15(d) abgebildet. Diese ist unter dem Namen Salz-Pfeffer-Rauschen bekannt.

Tabelle B.1: LUT (oben) Veranschaulichung der Kontraststreckung (unten)

LUT (Look up table)		
Farbe	Grauwerte	Gewünschte Ausgabegrauwerte
weiss	3	3
	2	3
	1	0
schwarz	0	0

3 Striche:					■		■		■				
Grauwerte:	3	3	3	3	0	3	0	3	0	3	3	3	3
Mittelwerte:	-	3	3	2	2	1	2	1	2	2	3	3	-
Kontrast gestreckt:	-	3	3	3	3	0	3	0	3	3	3	3	-
2 Striche:						■		■					



(a) Störung (b) des Übertragungskanal (c) signalabhängig (d) einzelner Pixel

Abbildung B.15: Arten der Bildstörungen bzw. des Bildrauschens

B.11 Zusammenfassung

- Bildentstehung: Licht - Reflexion - Sensor
- Digitales Bild: Grauwertraster, z.B. $512 \times 512 \times 256$
- Bildaufnahme: Kamera - Verarbeitung - Speicherung
- Mathematisches Modell: Integration über Ort und Zeit, Faltung
- Färbiges Licht: RGB - Auszüge
- Bildqualität: Schärfe, Kontrast, Auflösung
- Rauschen, Bildstörungen

Kapitel C

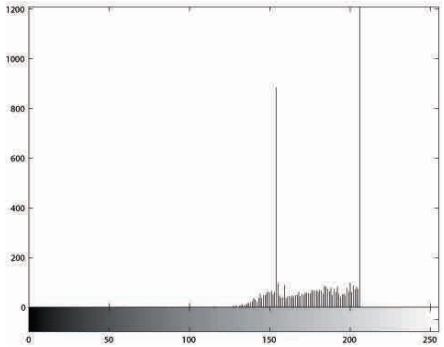
Histogramm, Threshold, Qualitätsverbesserungen

C.1 Histogramm

Ein Bild hat nur 256 Grauwerte. Wie oft kommt jeder Grauwert vor?

Bild I

Histogramm H von Bild I



$H(g) := \#\{I(x,y) = g\}, g \in [0, 255]$

verkleinerte Version (154×81) der gescannten Themenseite

Das Histogramm beschreibt die Häufigkeit des Auftretens eines Grauwertes in einem Bild.

C.2 Histogramm: Details

Ausschnitt des Histogramms von Grauwert 107 bis 206:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
100								0	1	0	0	0	0	0	1	4	0	0	0	1
120	0	0	0	1	0	0	1	5	3	6	0	5	7	12	6	10	17	17	19	24
140	37	30	21	37	52	35	48	48	57	63	59	65	51	60	883	96	47	38	40	88
160	35	41	46	42	47	41	48	49	60	43	53	50	58	57	57	55	66	68	66	67
180	62	69	68	55	86	81	74	63	78	48	73	61	81	54	41	52	55	47	77	62
200	99	61	86	71	82	74	7670													

Es treten nur Grauwerte im Bereich $[108, 206]$ auf.
Es gibt 2 hervorstechende Maxima:

H(154) = 883 ... (dunklere) Schriftzeichen
H(206) = 7670 ... (hellerer) Papierhintergrund

C.3 Histogramm, Eigenschaften

Einige Eigenschaften des Bildes lassen sich aus dem Histogramm bestimmen:
kleinster (analog: größter) Grauwert:
.... $min\{I(x,y)|x \in [1:n], y \in [1:m]\} = min\{g \in [0,255]|H(g) > 0\}$

Größe des Bildes = Anzahl der Pixel: .... $card(I) = \sum_{g=0}^{255} H(g)$

Mittelwert der Grauwerte: .... $\mu(I) = \sum_{g=0}^{255} g \cdot H(g)/card(I)$

Median der Grauwerte: .... med(I) =???

Varianz der Grauwerte: .... $\sigma^2(I)$ =??? (Wer kommt drauf?)

Der große Vorteil: 256 Operationen statt $n \times m$ (12 474) des Bildes!
1 Bild → 1 Histogramm ABER 1 Histogramm → viele Bilder !!

C.4 Histogrammäqualisierung, flattening

Ziel ist eine möglichst gleichmäßige Verteilung der Grauwerte
Die Histogrammäqualisierung führt zu einer Erhöhung des Kontrastes für Grauwerte nahe des Histogrammmaximums und zu einer Erniedrigung nahe des Minimums.

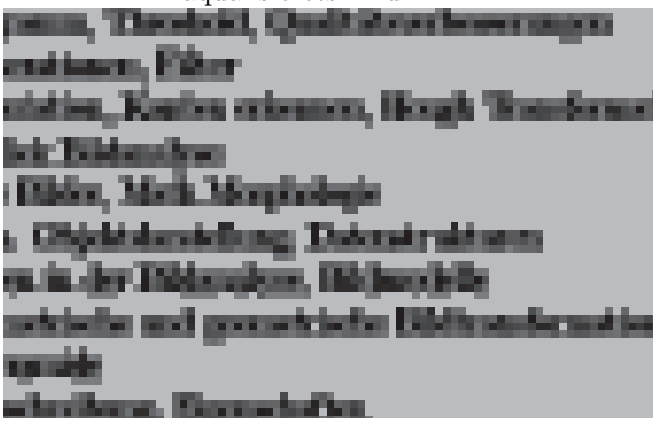
Operation: Transformation der Grauwerte $T : [0, 255] \mapsto [0, 255]$.

$$H_{soll} = H(T(g)) = nm/g_{max} \text{ für alle } g \in [0, 255]$$

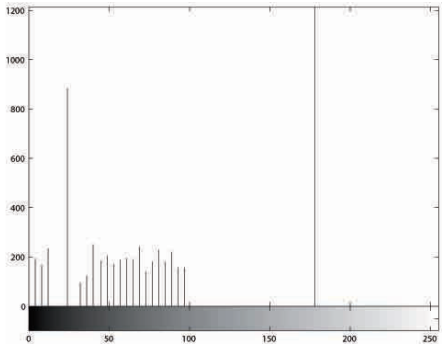
g_{max} ist die Anzahl der angestrebten Grauwerte, H_{soll} beschreibt das gewünschte Histogramm nach der Äqualisierung. Damit die Grauwerte gleichmäßig verteilt sind, wird die Gesamtzahl der Pixel durch die

Anzahl der neuen Grauwerte dividiert.

äqualisiertes Bild I



Histogramm H von äqualisierten Bild I



C.5 Algorithmus Histogrammäqualisierung

1. Finde kleinstes $g_0 \in [0, 255]$ mit $\sum_{g=0}^{g_0-1} H(g) \leq H_{soll}(0) < \sum_{g=0}^{g_0} H(g)$

   Setze $T(g) := \begin{cases} 0 & \text{für alle } g \in [0, g_0 - 1] \\ 0 & \text{für } H_{soll}(0) - \sum_{g=0}^{g_0-1} H(g) \text{ Pixel mit Grauwert } g_0 \end{cases}$

   Gilt das =-Zeichen in Schritt 1 nicht, so muß Grauwert g_0 auf die zwei Grauwerte 0 und 1 aufgeteilt oder gerundet werden.

2. Finde kleinstes $g_1 \in [g_0, 255]$ mit $\sum_{g=0}^{g_1-1} H(g) \leq H_{soll}(0) + H_{soll}(1) < \sum_{g=0}^{g_1} H(g)$

   Setze $T(g) := \begin{cases} 1 & \text{für } \max\{H_{soll}(1), \sum_{g=0}^{g_0} H(g) - H_{soll}(0)\} \text{ Pixel mit Grauwert } g_0 \\ 1 & \text{für alle } g \in [g_0 + 1, g_1 - 1] \\ 1 & \text{für } H_{soll}(0) + H_{soll}(1) - \sum_{g=0}^{g_1-1} H(g) \text{ Pixel mit Grauwert } g_1 \end{cases}$

   Gilt das =-Zeichen in Schritt 2 nicht, so muß Grauwert g_1 gerundet oder auf die zwei Grauwerte 1 und 2 aufgeteilt werden.

C.5.1 Beispiel

$g =$	0	1	2		3			4			5		6	7
$H(g) =$	1	7	21		35			35			21		7	1
split	1	7	8	13	3	16	16	16	16	3	13	8	7	1
$T(g) =$	0	0	0	1	1	2	3	4	5	6	6	7	7	7
$H_{soll}(g) =$	16		16		16		16	16		16		16		
runden	1	7	8	**13**	3	**16**	16	16	**16**	3	**13**	8	7	1
$T(g) =$	0	0	1	1	2	2	2	5	5	5	6	6	7	7
$H_{rund}(g) =$	8		21		35			35			21		8	

Für das Histogramm in Punkt C.2 bedeutet das: das kumulative Histogramm enthält bei Grauwert 133 47 Pixel, bei Grauwert 134 53 Pixel. Daher gilt $g_0 = 134$. Alle Pixel der ursprünglichen Grauwerte $[0, 133]$ werden daher auf 0 gesetzt. Da es sich hierbei aber um weniger als 49 Pixel ($H_{soll} = 48,7 \approx 49$) handelt, gilt hier das Zeichen = nicht, es werden daher auch noch zwei Pixel mit dem Grauwert 1 mitberücksichtigt.

C.6 Grauwerttransformation, Lookup Tabelle

Die Transformation des Bildes erfolgt dann mit der Lookup Tabelle $T(g)$:

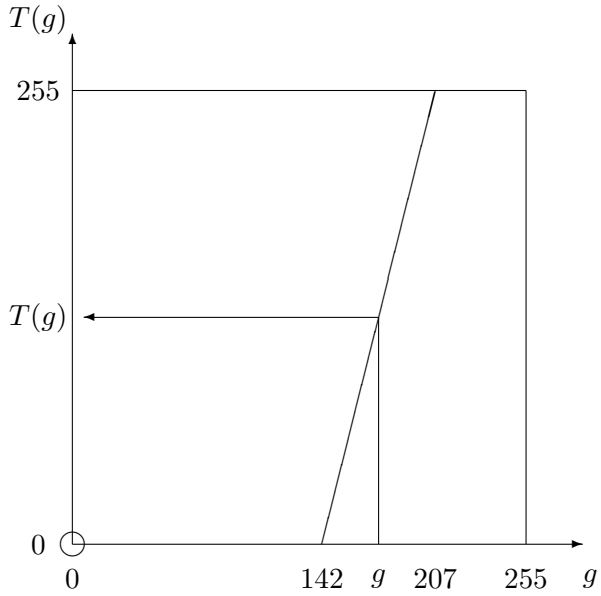
$$I_{soll}(x, y) = T(I(x, y)) \text{ für alle } x \in [1 : n], y \in [1 : m]$$

MATLAB [Teq,LUT]=histeq(Themen); LUT(107:206);

LUT	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
100								0	0	0	0	0	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
140	0	0	4	4	4	4	4	8	8	8	12	12	12	12	24	32	36	36	36	40
160	40	40	40	40	44	44	44	44	48	48	48	48	52	52	52	56	56	56	60	60
180	60	64	64	64	68	68	68	72	72	76	76	76	80	80	80	80	85	85	85	89
200	89	89	93	93	97	97	178	255												

Anmerkung: Diese LUT entspricht nicht der Berechnung der Histogrammäqualisierung des Histogramms aus Punkt C.2 anhand des besprochenen Algorithmus, sondern wurde bei der Berechnung skaliert.

C.7 Grauwerttransformation: grafische Darstellung



'Uniforme Grauwerttransformation', Punktoperation:
der Grauwert g jedes Pixels wird durch $T(g)$ ersetzt.

C.8 Uniforme Grauwerttransformationen

Kontrast strecken

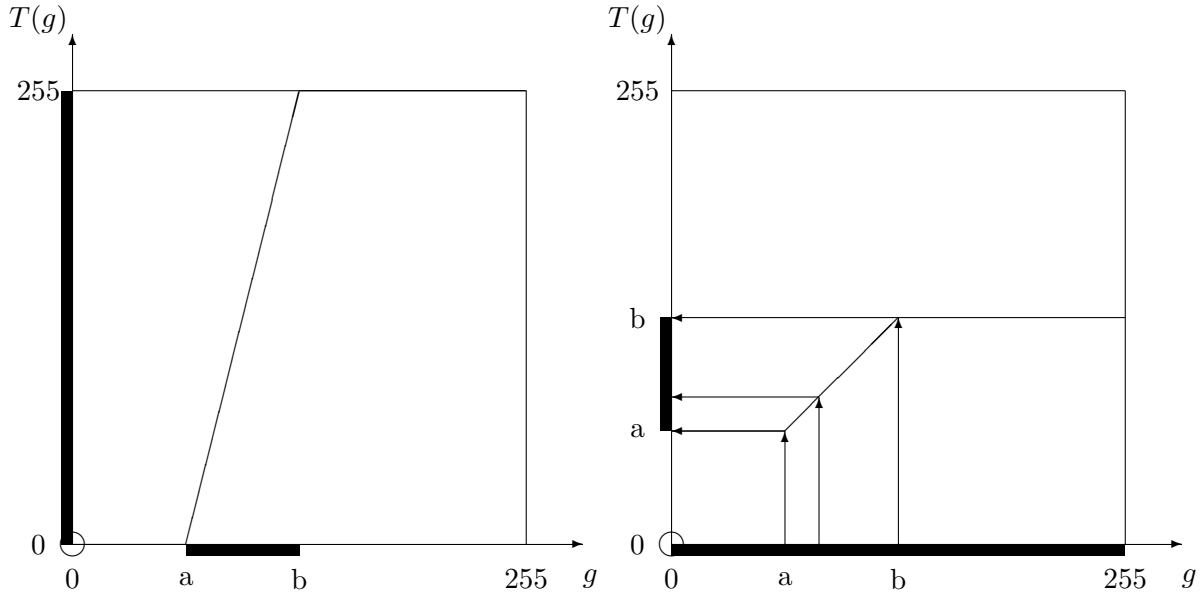
$T : [a, b] \mapsto [0, 255]$

$T(g) = \frac{255}{b-a}(g - a)$

Kompression, Auswahl

$T : [0, 255] \mapsto [a, b]$

$$T(g) = \begin{cases} a & g < a \\ g & a \leq g \leq b \\ b & b < g \end{cases}$$



Kombinationen ...

Bei der Kompression wird das Intervall der möglichen Farbwerte [0,255] in das Intervall [a,b] konvertiert. Dabei ändert sich die Häufigkeit der Grauwerte in diesem Intervall. Auswahl bedeutet, dass ein Intervall [a,b] ausgewählt wird, Grauwerte kleiner a werden auf a abgebildet, Grauwerterte gößer b auf b. Dadurch ändert sich die Häufigkeit der Grauwerte im Intervall [a,b] nur an den Stellen a und b, sonst aber nicht.

C.9 Die Anzahl der Ergebnisgrauwerte g_{max} und eine Verallgemeinerung

Eine geringere Anzahl der Grauwerte bei Histogrammäqualisierung gibt dem Algorithmus den Spielraum, die neuen Grauwerte der Gleichverteilung gut anzupassen.

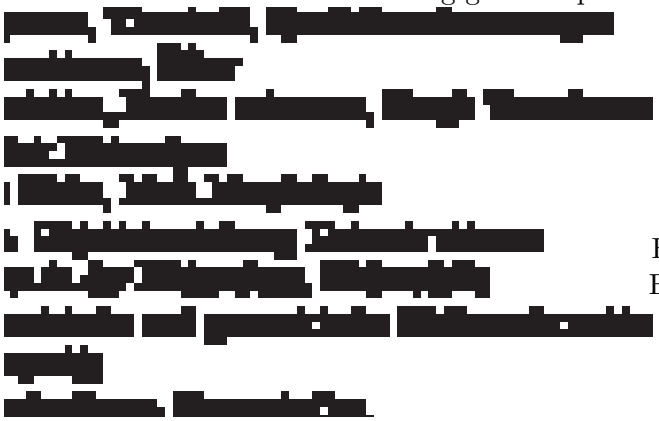


Bild = Äqualisierung auf 2 Werte
$g_{max} = 2$ ergibt ein Binärbild:
Alle Pixel außer den 7670
Hintergrundpixeln werden auf 0 gesetzt.
Eine erste Form der Schwellwertbildung.

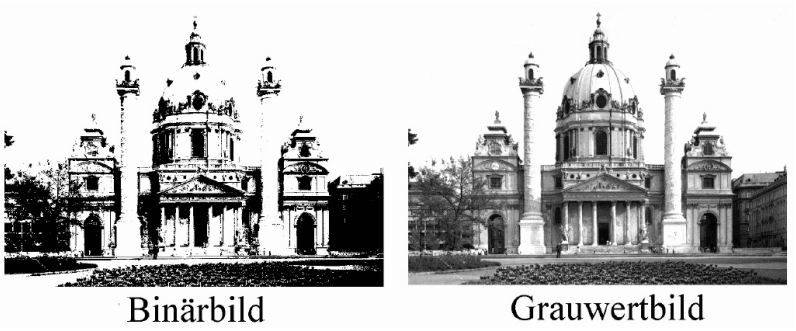
Als Sollverteilung H_{soll} im Algorithmus kann jede beliebige diskrete Verteilung gewählt werden. Dies kann vorteilhaft sein, wenn die Grauwertverteilung der betrachteten Bildklasse bekannt ist (z.B. Textdo-

kumente).

C.10 Threshold

Eine Grauwertschwelle ('threshold' t) erzeugt ein Binärbild:

$$g(x,y) = \begin{cases} 1 & \text{wenn } f(x,y) > t \\ 0 & \text{sonst} \end{cases}$$



C.11 Threshold aus Histogramm bestimmen

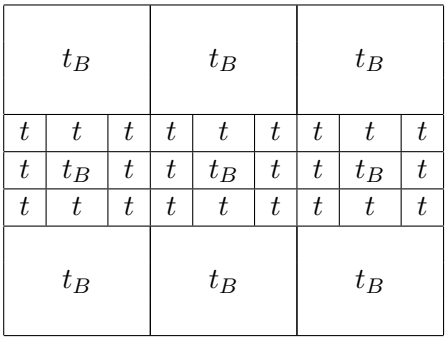
Wie wählt man den Parameter t?

1. probieren
2. gezieltes probieren: höherer Schwellwert t $\iff$ weniger Pixel mit $g(x,y) = 1$
   niedrigerer Schwellwert t $\iff$ mehr Pixel mit $g(x,y) = 1$
3. aus dem (bimodalen) Histogramm:
   - z.B. 2 Gipfel und ein Tal dazwischen: wähle t an der tiefsten Stelle des Tales.
   - mehr Gipfel? Glätten des Histogramms reduziert lokale Extrema.
4. aus Vorwissen, z.B. Verteilung von Schwarz $N_0(\mu_0, \sigma_0)$ und Weiß $N_1(\mu_1, \sigma_1)$
   schätzen der Verteilungsparameter $\mu_0, \sigma_0, \mu_1, \sigma_1$
   aus dem Histogramm $H = N_0 + N_1$,
   Schwellwert t soll Fehlpixel minimieren.

C.12 variable Schwellwerte, lokale Max.

Problem: globaler Schwellwert $\longrightarrow$ zu niedrig in einem Bildteil
zu hoch in anderem Bildteil

- Unterteile Bild in Blöcke
- bestimme Schwellwert t_B für jeden Block individuell
- interpoliere Schwellwert $t = T(x,y)$ für jeden Pixel
- $g(x,y) = 1 \iff f(x,y) > T(x,y)$



CV97, Module 3: Image Preprocessing/ Contrast Enhancement
Topics: Contrast Enhancement, Thresholding, Histogram Flattening.
unter: http://www.prip.tuwien.ac.at/sites/default/files/cv-b3-contrastenhancement.mpeg

Dieses Video bietet eine gute Zusammenfassung zu essentiellen Themen der Bildverarbeitung. Folgenden Themen werden behandelt:

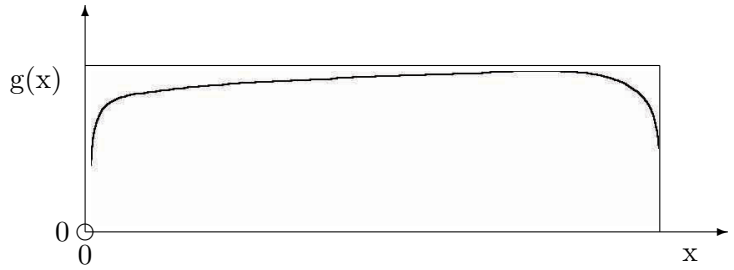
- Zusammenhang Bild - Histogramm
- Histogrammqualisierung (bei Über- / Unterbelichtung)
- Threshold - richtige Wahl des Thresholds, variabler Threshold

C.13 Qualitätsverbesserungen

- uniforme Grauwerttransformation (Kontrast strecken, Auswahl)
- ortsabhängige Grauwertkorrektur
- Störungen glätten $\longrightarrow$ Glättungsfilter (Seite 37)
- Verwischungen schärfen $\longrightarrow$ Kantenfilter (Seite 37)

C.14 ortsabhängige Grauwertkorrektur

Problem: Durch ungleichmäßige Beleuchtung und unterschiedliche Empfindlichkeit der Sensoren trotz homogener Objektoberfläche $\mapsto$ verschiedene Grauwerte.



$$g(x,y) = c(x,y) \cdot f(x,y)$$

Kalibrieren: 1) Scan von weißer Fläche liefert den Wert $g_1(x,y)$
2) Daraus berechnet sich der Korrekturfaktor: $c_1(x,y) = g_1(x,y)/255$

ortsabhängige Grauwertkorrektur der eigentlichen Aufnahme $g(x,y)$:

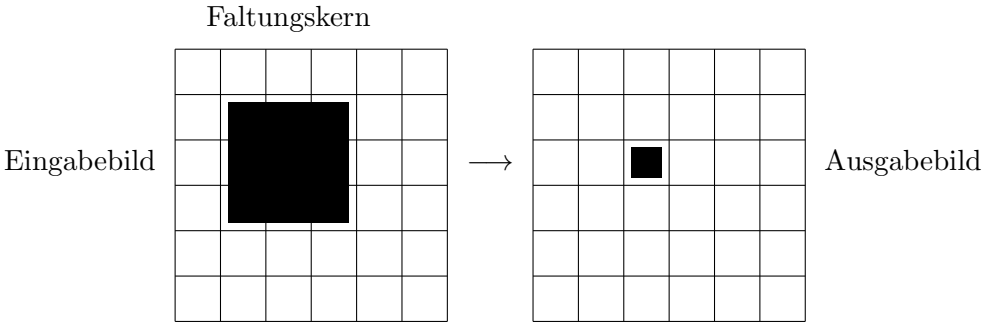
$$f(x,y) = g(x,y)/c_1(x,y)$$

C.15 Faltung

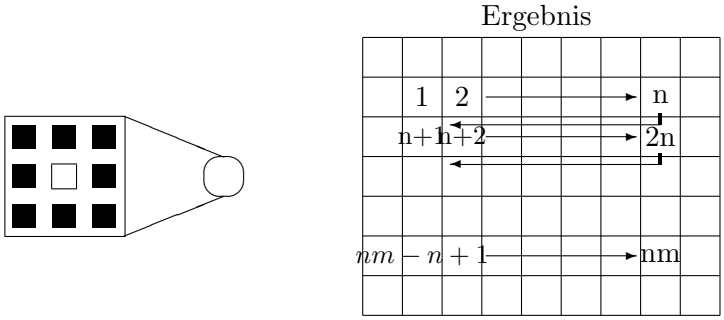
1 Objektpunkt $\longrightarrow$ mehrere Bildpunkte (Seite 13)
Integration $\longrightarrow$ gewichtete Mittelung (Seite 13)

Nachbarschaftsoperation

1 Pixel des Ausgabebildes wird aus mehreren Pixeln des Eingabebildes berechnet.



C.16 Faltung: zeilenweise Berechnung



$$B * F = \begin{array}{|ccccc|} \hline 0 & 1 & 0 & 1 & 1 \\ 3 & 1 & 2 & 1 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 \\ \hline \end{array} * \begin{array}{|ccc|} \hline 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ \hline \end{array} = \begin{array}{|ccc|} \hline 9 & 6 & 7 \\ 6 & 10 & 5 \\ 5 & 5 & 4 \\ \hline \end{array}$$

$$B(i,j) * F_{nm} = \sum_{k=-n}^{n} \sum_{l=-m}^{m} F(k,l) \cdot B(i+k, j+l)$$

Diese Formel dient der pixelweisen Berechnung der Faltung des Eingabebildes mit dem Faltungskern F (bei einem Faltungskern der Größe 3x3 wie er hier vorliegt, durchlaufen die Parameter n und m die Werte -1 bis 1).
Diese Formel kann nur für Faltungskerne von ungerader Größe verwendet werden. Natürlich ist grundsätzlich eine Verwendung von Faltungskernen gerader Größe ebenfalls möglich, hierzu müssen aber die Zählvariablen angepasst werden und es muss ein Referenzpixel zur Speicherung festgelegt werden (bei ungerader Größe wird, wenn nicht anders festgelegt, immer das zentrale Pixel als Referenzpixel gewählt).

C.17 Faltung: 3×3 Mittelwert

Mittelwertfilter: alle Gewichte des Faltungskerns haben denselben Wert.
Summe der Gewichte soll 1 ergeben.
Faltungskern:

$$* \begin{pmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{pmatrix} =$$

Originalbild + additive Störung

Ergebnis der Faltung

Anwendungen: Glättung, Kanten, lineare Filter

C.18 Zusammenfassung

- Histogramm = Häufigkeiten der Grauwerte eines Bildes
- das Histogramm enthält viele Bildeigenschaften trotz hoher Datenkompression
- uniforme Grauwerttransformation (LUT)
  Histogrammäqualisierung,
  Kontrast strecken,
  Grauwertbereich auswählen.
- Schwellwert transformiert Grauwertbild in Binärbild, nur 1 Parameter (threshold)
  globaler Threshold oder Threshold für Bildblöcke
- Qualitätsverbesserungen: ortsabhängige Grauwertkorrektur: Korrektur ansich gleicher aber unterschiedlich
  abgebildeter Grauwerte
- Faltung des Bildes mit einem Faltungskern: pixelweise Neuberechnung, zB.: 3x3-Mittelwertfilter

Kapitel D

Bildoperationen, Filter

D.1 Klassen von Operationen auf Bildern

1. **Punktoperationen**: $g(x,y) = O(f(x,y))$, $g(x,y) = O(f_1(x,y), \ldots f_n(x,y))$.
   Kontrast strecken, Grauwerttransformation (Seite 27).

2. **Lokale Operationen**: $g(x,y) = O(f(x,y), f(x-1,y), \ldots)$.
   Wichtigste Anwendungen:

   - Schärfen, Kanten erkennen (Seite 51).
   - Glätten
   - Merkmalserkennung

3. **Globale Operationen**: $g(x,y) = O(x, y, f(0,0), \ldots f(n,m))$.
   (siehe Vorlesungsteil CV)
   Fourier, Cosinus, Karhunen-Loeve (PCA)

4. **Geometrische Operationen**: Bild vergrössern, verschieben, drehen, Perspektive, Überlagern verschiedener Aufnahmen. (siehe Vorlesungsteil CG und CV)

D.2 Eigenschaften und Abhängigkeiten von Operationen

Lineare Operatoren: $O(\lambda B_1 + \mu B_2) = \lambda O(B_1) + \mu O(B_2)$

Lineare Operatoren berechnen den Resultatwert im Ausgabepixel als Linearkombination der Helligkeitswerte in der Nachbarschaft des jeweiligen Pixels im Eingabebild.

Das Ergebnis einer Bildoperation kann abhängen von:

- dem Ort (x, y), (z.B. Abdunklung am Bildrand)
- der Signalstärke (z.B. Multiplikation)
- der Verarbeitungsreihenfolge:
  - sequentiell,
  - iterativ (vom Vorgänger abhängig),
  - rekursiv (von mehreren Vorgängern abhängig),
  - parallel (von Vorgängern unabhängig, beliebige Reihenfolge).

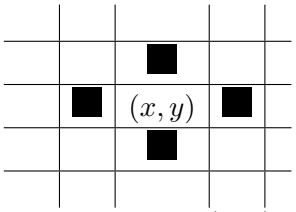
Unabhängigkeiten (ortsunabhängig, signalunabhängig) sind vorteilhaft, da sich in diesem Fall Operationen unabhängig vom Eingabebild immer gleich verhalten und "vorhersagbare" Ergebnisse liefern.

D.3 4-/8-Nachbarschaft

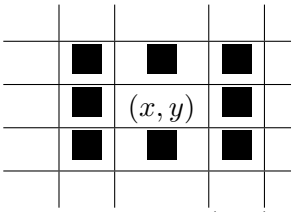
$(x-1, y-1)$	$(x, y-1)$	$(x+1, y-1)$
$(x-1, y)$	(x, y)	$(x+1, y)$
$(x-1, y+1)$	$(x, y+1)$	$(x+1, y+1)$

Nachbarkoordinaten

Es gibt zwei einfache Möglichkeiten die Nachbarschaft eines Pixels in einem 2-D Bild zu definieren: 4- und 8-Nachbarschaft. Je nach Definition werden dann auch diagonal angrenzende Pixel zur Nachbarschaft gezählt oder nicht.



4-Nachbarn:$\Gamma_4(x, y)$
City-Block Distanz

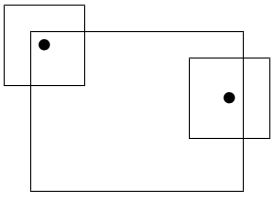


8-Nachbarn:$\Gamma_8(x, y)$
Schachbrett Distanz

D.4 Behandlung des Bildrandes

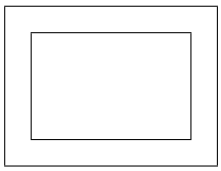
Beim praktischen Einsatz von Nachbarschaftsoperatoren tritt das Randproblem auf: bei Randpixel ragt die Nachbarschaftsregion über den Bildrand hinaus.

Problem der Operationen:



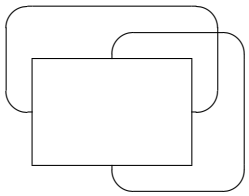
Lösungen:

1. Spezialfälle, algorithmische Berücksichtigung

2. die Randpixel werden nicht betrachtet: 

   Das Referenzpixel wird immer so am Bild plaziert, dass der gesamte Filterkern (= Referenzpixel + alle weiteren Pixel des Filterkerns) mit dem Bild überlappt und nicht darüber hinausstehen. D.h.: wenn zB.: ein 3×3 großer Filter mit zentralem Referenzpixel eingesetzt wird, wird die äuerßte Zeile bzw. Spalte von Pixeln, am Rand des Bildes nicht mit dem Filter behandelt.Dadurch ist das Ergebnisbild um genau diese nicht betrachteten Pixel kleiner als das Originalbild.

3. Annahme: Pixel außerhalb des Bildes 0, dadurch wird die Funktion der Operation am Rand beeinflusst. Bei lokaler Mittelung wird der Rand des Bildes dunkler.

4. zyklischer Abschluß, das Bild wird periodisch fortgesetzt: 

   Wird ein Bild zyklisch fortgesetzt, kann es (abhängig vom jeweiligen Bild und dem eingesetzten Filter) zu unerwarteten Strukturen im Ergebnisbild kommen. Diese enstehen wenn sich rechter und linker Bildrand, bzw. oberer und unterer Bildrand in ihrern Grauwerten stark unterscheiden. Diese „neuen“ Grauwerte, die durch den zyklischen Abschluss in die Nachbarschaft eines Pixels fallen, können sich bei Anwendung eines Filters auf eine solche Nachbarschaft auf den resultierenden Grauwert des Ausgabepixels auswirken.

D.5 Kontrast schärfen, diskrete Ableitungen

- Unschärfe wird durch Mittelung/Integration hervorgerufen
- Aufhebung durch Differentiation/Ableitungen, da der Gradient bei starken Intensitätsänderungen, wie diese an Kanten auftreten, besonders groß ist.

stetige Ableitung	diskrete Ableitung
$\frac{\partial f}{\partial x}$	$\Delta_H f(i,j) = f(i,j) - f(i-1,j)$
$\frac{\partial f}{\partial y}$	$\Delta_V f(i,j) = f(i,j) - f(i,j-1)$
in Richtung θ	$\Delta_\theta f(i,j) = \Delta_H f(i,j) \cos\theta + \Delta_V f(i,j) \sin\theta$
Gradient	$\sqrt{(\Delta_H f(i,j))^2 + (\Delta_V f(i,j))^2}$
Richtung	$\arctan \frac{\Delta_V f(i,j)}{\Delta_H f(i,j)}$

Diskrete Ableitungen werden zur Kantendetektion verwendet.

D.6 1. Ableitung

Die 1. Ableitung ist groß für Bereiche mit starker Steigung der Funktion (im Bild Bereiche mit großen Intensitätsunterschieden, zB.: Objektkanten), bei keiner bis wenig Steigung (im Bild homogene Flächen) ist die 1. Ableitung klein (siehe Abbildung D.1).

Profil

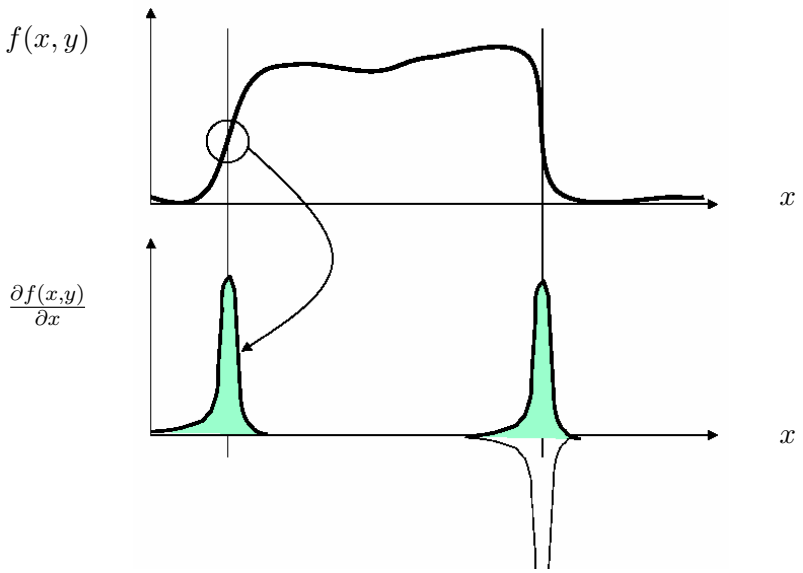
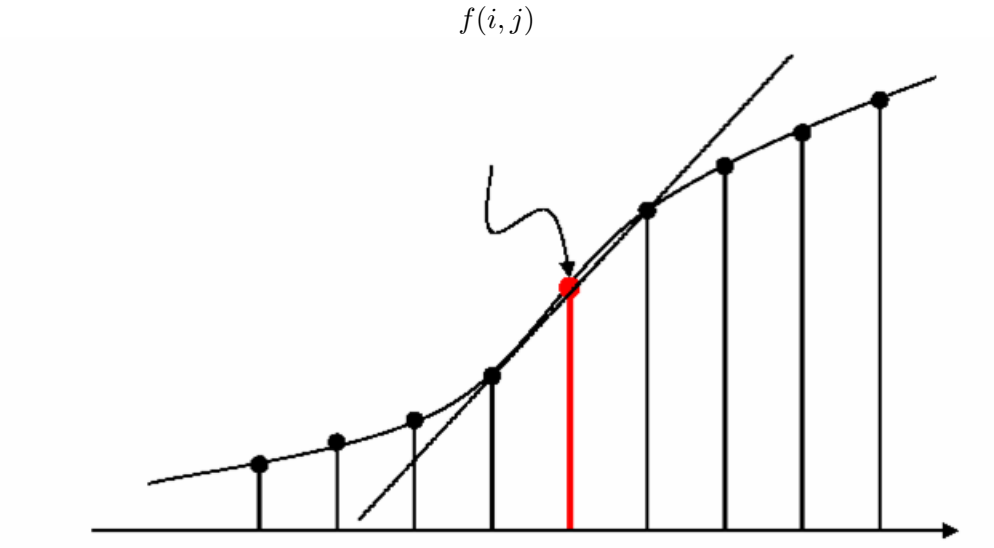


Abbildung D.1: Funktion und 1. Ableitung

D.7 Wie misst man die 1. Ableitung?

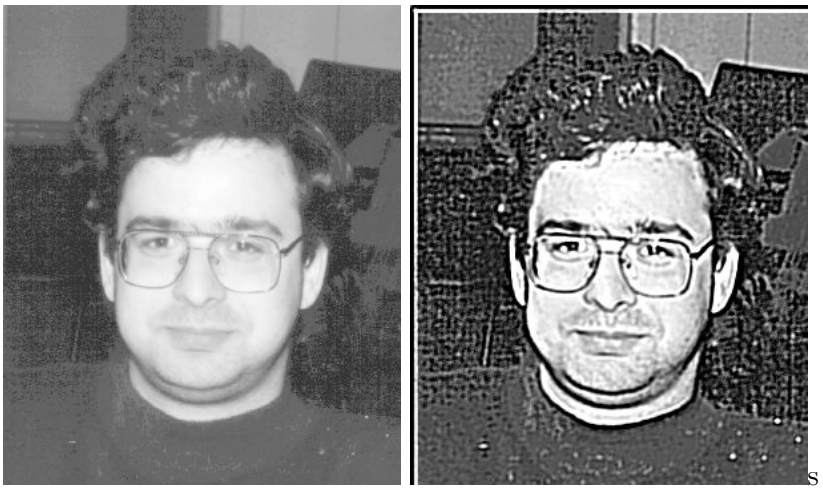


Approximation durch $\frac{\partial f(i,j)}{\partial i} \approx f(i,j) - f(i-1,j)$

D.8 Kontrast schärfen: 2. Ableitung, Laplace-Operator

stetige Funktion	diskrete Version
2. Ableitung $\frac{\partial^2 f}{\partial x^2}$	$\Delta_H^2 f(i,j) = \Delta_H f(i+1,j) - \Delta_H f(i,j)$ $= f(i-1,j) - 2f(i,j) + f(i+1,j) = f * (1,-2,1)$
	Laplace Operator
$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$	$(\Delta_H f(i,j))^2 + (\Delta_V f(i,j))^2$ $= f * \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$
	Kanten schärfen
$f + \nabla^2 f$ siehe Abbildung D.2	$f * \begin{pmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{pmatrix}$

D.9 Kanten schärfen



(a) Originalbild (b) Kanten geschärft

Abbildung D.2: Kanten schärfen

Nebeneffekt: Rauschen wird verstärkt!

D.10 CV97: Contrast Enhancement, Teil 2

Link: http://www.prip.tuwien.ac.at/sites/default/files/cv-b3-contrastenhancement.mpeg

Dieses Video bietet eine Zusammenfassung zu folgenden Themen:

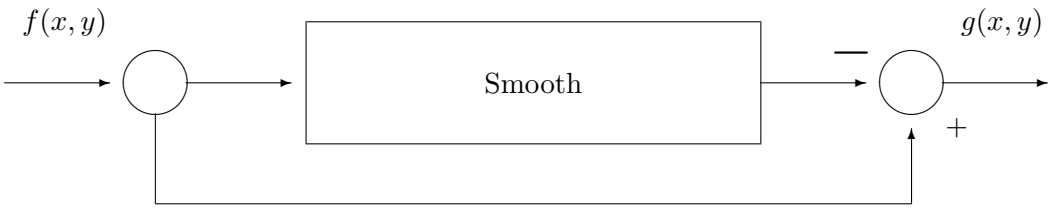
- Unsharp Masking
- Linear Smoothing

D.11 Unsharp Masking

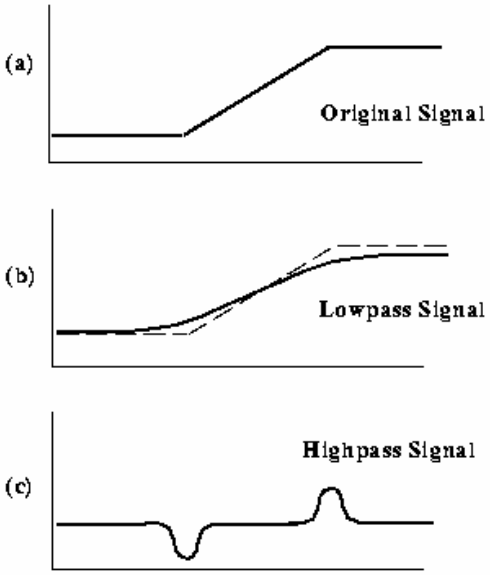
- Unsharp Masking erzeugt ein Kantenbild $g(x, y)$ aus einem Eingabebild $f(x, y)$ durch:

$$g(x, y) = f(x, y) - f_{smooth}(x, y)$$

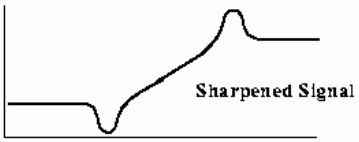
- $f_{smooth}(x, y)$ ist eine geglättete Version von $f(x, y)$



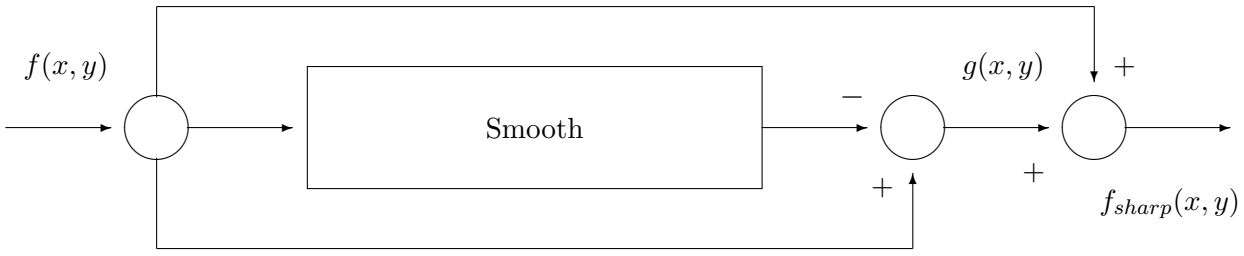
Subtraktion der Tiefpasskomponente eines Signals erzeugt ein Hochpass-Signal



Durch Addition der Hochpasskomponente zum Originalsignal kann eine Schärfung erreicht werden.

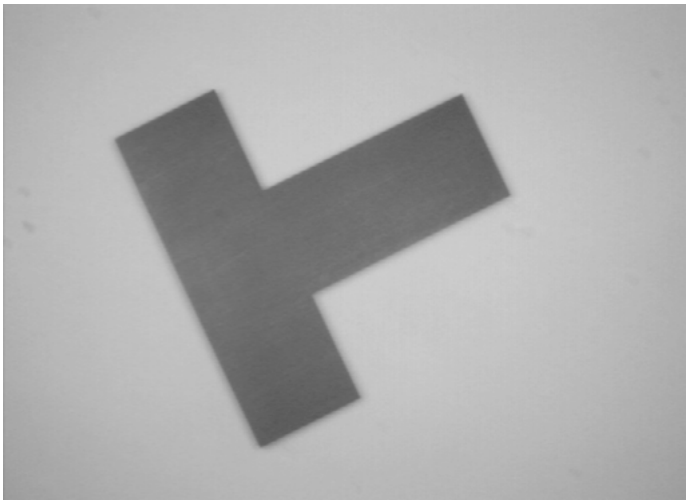


(siehe Abbildung D.3)

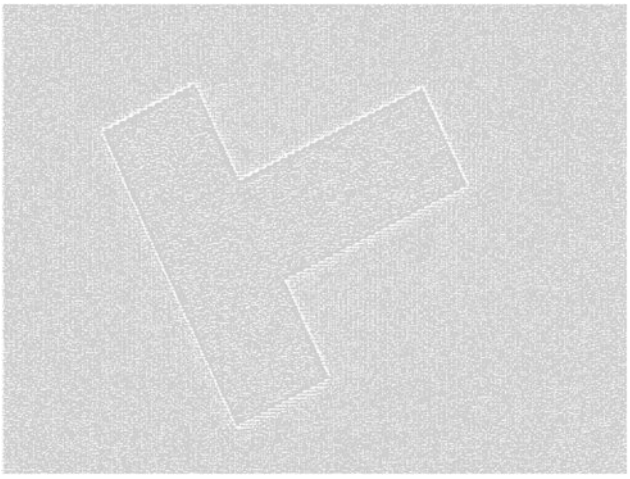


$$f_{sharp}(x, y) = f(x, y) + k \cdot (f(x, y) - f_{smooth}(x, y))$$

Der Faktor k ist eine Skalierungskonstante.



(a) Originalbild



(b) Bild hochpassgefiltert, für Darstellung normiert

(c) geschärftes Bild

Abbildung D.3: Unsharp Masking

D.12 Bild glätten

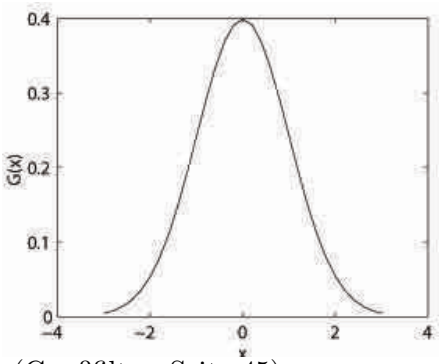
- Ziel: Störungen reduzieren, eliminieren
- Nebenwirkungen: Kanten verwischen, Bild wird unscharf

Mathematisch betrachtet gibt es folgende Möglichkeiten eine Glättung durchzuführen:

1. Mittelung M einer additiven Störung N: $R = (F + N) * M = F * M + N * M$
   da die Faltung ein linearer Operator ist. (Mittelwertfilter: Seite 44)

2. gewichtete Mittelung: Gauß'sche Glockenkurve in 1-D

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$



1-D Gaußverteilung mit Mittelwert 0 und $\sigma = 1$. (Gaußfilter: Seite 45)

3. Anwendung eines lokalen Medianwertes (Medianfilter: Seite 46)
4. Mittelung einzelner Bildzeilen (LANDSAT: Seite 47)
5. selektive Mittelung (Mittelungsvarianten: Seite 48)

D.12.1 Mittelwertfilter (Box-Filter)

Beim Mittelwertfilter entsprechen die Pixelwerte im Ausgabebild dem Mittelwert der Pixelwerte der Nachbarschaft des Pixels im Eingabebild. Somit haben alle Gewichte des Filterkerns den selben Wert. Die Summe der Gewichte ergibt 1.

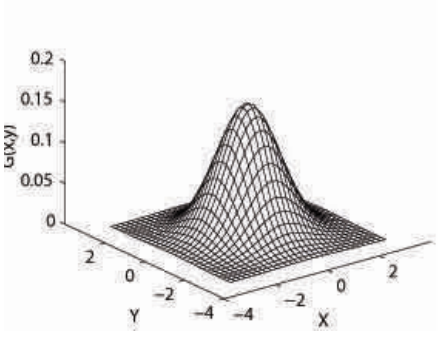
Originalbild + additive Störung

$$* \begin{pmatrix} \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{pmatrix} =$$

Ergebnis der Faltung mit einem 5x5 Mittelwertfilter

D.13 Gewichtete Mittelung

Gauß-Gewichte in 2-D: $$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



2-D Gaußverteilung mit Mittelwert 0 und $\sigma = 1$.

Filterkern: $\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$
Variante: Mittelung zeitlich aufeinanderfolgender Bilder einer Sequenz.

D.13.1 Gaußfilter

Der Filterkern eines Gaußfilters entpsricht einer diskreten, 2-D Gaußfunktion. Die benachbarten Pixel des Pixels im Eingabebildes wirken sich nicht wie beim Mittelwertfilter gleichwertig auf den Wert des Pixels im Ausgabebild aus, sondern werden entsprechend einer zweidimensionalen Gaußglocke gewichtet.

- Die Gaußgewichtung reduziert den Einfluß von weiter vom Filterzentrum entfernten Pixel.
- $G(\sigma_1) * G(\sigma_2) = G(\sqrt{\sigma_1^2 + \sigma_2^2})$ Die Gaußgewichtung ist invariant in Bezug auf die Faltung, daher ist die Summe zweier Gaußverteilungen wieder eine Gaußverteilung.

Beispiel eines 3x3 Gaußfilters: $\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$

Originalbild + additive Störung

Ergebnis der Faltung mit einem 3x3 Gaufilter mit $\sigma = 1$

D.14 Additive Störung glätten

Originalbild + additives Rauschen

3×3 Mittelwert

5×5 Gaußfilter

Additives Rauschen ist folgendermaßen gegeben: Für ein diskretes Bild I und Rauschen δ sind die

gemessen Daten D punktweise gegeben durch:

$D = I + \delta$

D.15 Salz und Pfeffer: lokaler Mittelwert, Gaußgewichtung

Originalbild, 1% Salz + Pfeffer

3×3 Mittelwert

5×5 Mittelwert

Salz und Pfeffer Rauschen zeichnet sich durch zufällig auftretende weiße (Salz) und schwarze (Pfeffer) Pixel aus.
Eine Andwendung der soweit bekannten Filter Mittelwert- und Gaußfilter haben auf dieses Rauschen folgende Auswirkung:

- falsche Pixel werden zwar ihrer Umgebung angepaßt
- verschwinden aber nur bei großer Glättung.

D.16 Medianfilter

Der Medianfilter ersetzt jedes Pixel durch den 'Median' seiner Nachbarschaft.

232	255	255	255	58
195	255	241	233	118
224	254	95	109	135
0	0	111	20	45
128	0	0	0	58

1. sortiere Grauwerte der Nachbarschaft: 0, 20, 95, 109, **111**, 233, 241, 254, 255.
2. wähle den mittleren Pixel (111) als Ergebnis

Der Medianfilter zählt zu den nichtlinearen Filtern und gehört der Klasse der Rangordnungsfiltern an. Beim Einsatz eines Medianfilters werden einzelne Pixel ersetzt, ohne dadurch eine Kantenglättung zu bewirken.

nicht-lineare Varianten: Rangordnungsfilter (rank order filter)
G. Heygster, „Rank filters in digital image processing," Computer Vision, Graphics and Image Processing, vol. 19 pp. 148-164, 1982

D.17 Salz und Pfeffer: Medianfilter

Durch die Ersetzung jedes Pixels durch den Median der Pixel in der Nachbarschaft, werden Ausreißer (d.h.: Pixel die sich in ihrem Grauwert stark von ihrer Nachbarschaft unterscheiden und möglicherweise Rauschen darstellen) eliminiert. Zur Verringerung / Eliminierung von Salz und Pfeffer Rauschen eignet sich daher der Medianfilter.

Originalbild, 1% Salz + Pfeffer

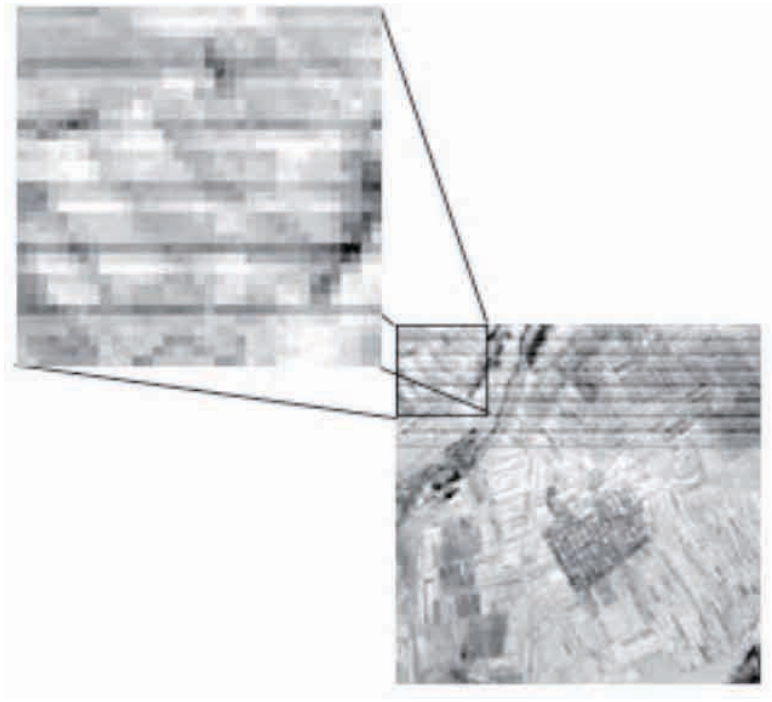
3×3 Medianfilter

3×3 Mittelwert

Der Medianfilter hat folgende Auswirkungen:

- einzelne Ausreißer haben keinen Einfluß auf das Ergebnis
- im Ergebnisbild entstehen keine neuen Grauwerte

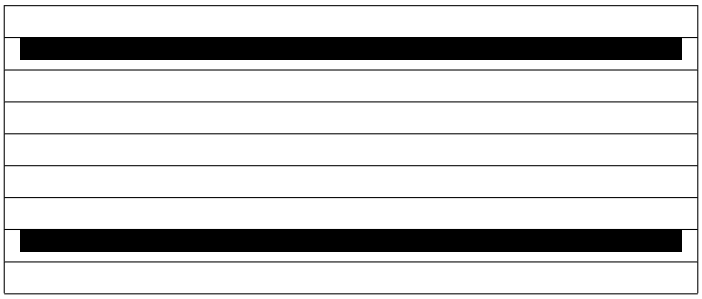
D.18 LANDSAT: Streifen-Phänomen



Aufnahme erfolgt über 6 Zeilensensoren in Serie, unterschiedliche Empfindlichkeit (gegeben durch unterschiedlichen Verschleiß der Sensoren) lässt störende Streifen entstehen.

D.19 Weitere Glättungsvarianten: Zeilen mitteln

Landsat: 6 Zeilensensoren in Serie $\longrightarrow$ Streifen

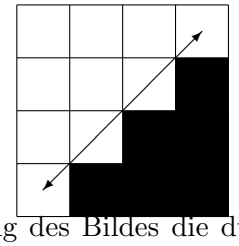


$M_1 = 255$	$\bar{M}_1 = 220$	-35
$M_2 = 45$	$\bar{M}_2 = 220$	$+175$
$M_3 = 255$	$\bar{M}_3 = 220$	-35
$M_4 = 255$	$\bar{M}_4 = 220$	-35
$M_5 = 255$	$\bar{M}_5 = 220$	-35
$M_6 = 255$	$\bar{M}_6 = 220$	-35
$M_7 = 255$	$\bar{M}_7 = 220$	-35
$M_8 = 45$	$\bar{M}_8 = 220$	$+175$
$M_9 = 255$	$\bar{M}_9 = 220$	-35

1. M_j = Mittelwert einer Bildzeile
2. $\bar{M}_j = (M_{j-3} + M_{j-2} + \cdots + M_{j+2})/6$ Mittelwert benachbarter Zeilen
3. Korrektur der einzelnen Pixelwerte z_{ij}: $z'_{ij} = z_{ij} + \bar{M}_j - M_j$

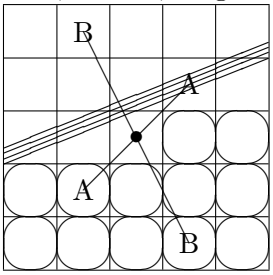
D.20 Selektiv Mitteln

1. nur in Kantenrichtung mitteln



Ziel dieser Methode ist es bei Glättung des Bildes die durch die Glättung entstehnde Unschärfe nur an Kanten zu verhindern. Die Kantenrichtung lässt sich mittels des Gradientens bestimmen. Die Glättung findet dann nur in Richtung der Kante(n) statt und verringert somit Unschärfe an Kanten.

2. Symmetric Nearest Neighbour Filter (SNN, PRL 6, August 1987):



SNN Filter glätten ein Bild und unterdrücken somit Rauschen, ohne dabei Kanten und Strukturen im Bild zu verwischen. Zur Glättung eines Bildes wird der neue Grawert eines zentralen Pixels eines quadratischen Fensters wie folgt berechnet:
Für jedes Paar von Pixeln, die sich am zentralen Pixel des Fensters symmetrisch gegenüberliegen, bestimme das Pixel, dessen Grauwert sich weniger von dem des zentralen Pixels unterscheidet.

Das heißt: für ein $(2n+1) \times (2n+1)$ Fenster mit zentralem Pixel (x, y), wähle für jedes Pixelpaar $\{(x+i, y+j), (x-i, y-j)\}$ mit $-n \leq i, j \leq +n$

entweder $(x+i, y+j)$ wenn $|g(x,y) - g(x+i, y+j)| < |g(x,y) - g(x-i, y-j)|$

oder $(x-i, y-j)$ wenn $|g(x,y) - g(x+i, y+j)| > |g(x,y) - g(x-i, y-j)|$

So werden 50% der Pixel selektiert, die sich auf einer Seite einer geraden Kante durch den Mittelpunkt des Fensters befinden. Dies trifft für jede Kante unabhängig von ihrer Richtung zu, solange die Grauwerteverteilungen der Grauwerte auf beiden Seiten der Kante gut separierbar sind.
Der neue Grauwert des zentralen Pixels wird anschließend als Mittelwert oder Median dieser ausgewählten Grauwerte berechnet.

Ergebnis = Mittelwert/Median jener Pixel, die dem Zentrum ähnlicher sind.

D.21 Zusammenfassung

- Klassen von Operationen $\longleftrightarrow$ Abhängigkeiten bei der Berechnung
- Lokale Operationen: unterschiedliche Definitionen der Nachbarschaft von Pixel: 4-/8-Nachbarschaft
- Mittelung/Glättung (Integration) verursacht Unschärfe
- Kantendetektion und Kontrastschärfung durch diskrete Differentiation (detektiert Grauwertsprünge)
- unsharp masking: Kanten schärfen durch Addition der Hochpasskomponente (wird durch Subtraktion eines geglätteten Bildes vom Originalbild gewonnen) zum Originalsignal
- Mittelwert-, Gaußfilter reduzieren additive Störungen
- Medianfilter eliminiert Salz & Peffer Rauschen

Kapitel E

Segmentation, Kanten erkennen, Hough Transformation

E.1 Segmentation und Bildanalyse

BILD $\mapsto$ BILD - Folgende Gebiete der Bildverarbeitung liefern als Ergebnis, ausgehend von einem Bild, wieder ein Bild:

- Bildverbesserung (Filterung, Glättung, etc.)
- Bildoperation (Faltung, Addition, Subtraktion, etc.)
- Bildtransformation (Fouriertransformation, Radontransformation, etc.)

BILD $\mapsto$ BESCHREIBUNG - Folgende Gebiete der Bildverarbeitung liefern als Ergebnis, ausgehend von einem Bild, eine Beschreibung dessen:

- Bildanalyse
- Computer Vision
- Szenenanalyse

E.2 Segmentierung

Der 1. Schritt vom Bild zur Beschreibung ist die Segmentierung, wo Pixel zu Regionen zusammengefasst werden.

E.2.1 Segmentation Beispiel

In folgendem Beispiel sollen schwarze Buchstaben auf einem weißen Papier segmentiert werden. Die Segmentierung der eingescannten Textseite liefert als Ergebnis die Regionen für die Buchstaben von A bis Z in Form einer Beschreibung von Bildeigenschaften oder Beziehung zwischen einzelnen Bildbestandteilen (siehe Abbildung E.1 und E.2).

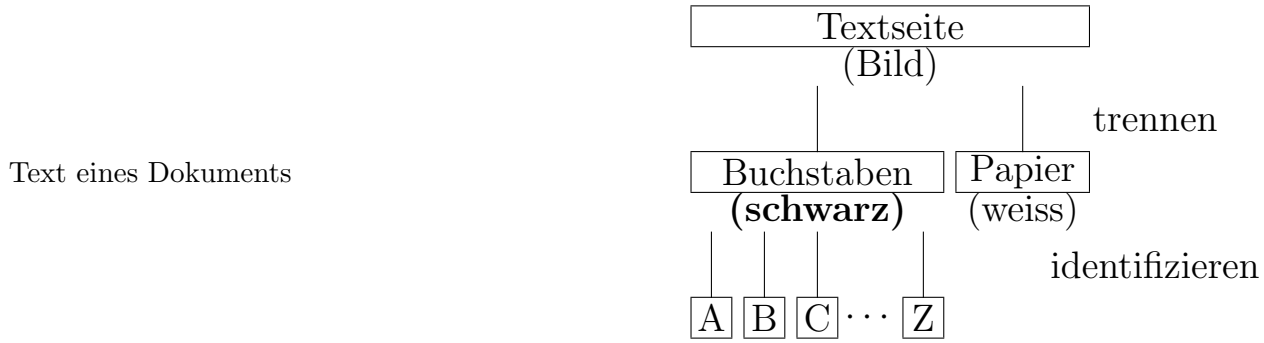


Abbildung E.1: Beispiel Segmentierung

Abbildung E.2: Segmentierung

E.2.2 Segmentation

Zusammengehörige Bildteile werden als "homogen" bezeichnet

- **Homogenitätskriterien**:
  - gleicher/ähnlicher Grauwert (z.B. $H(R) = \sigma(R)$)
  - gleiche/ähnliche Farbe
  - gleiche/ähnliche Textur
  - bilden eine bekannte Form (z.B. Buchstaben)
- Regionenbasiert: Pixel klassifizieren (z.B. Threshold, Filter)
- Kantenbasiert: detektieren von trennenden Kanten und Linien.

E.3 Kanten

E.3.1 Marr's Motivation

- Physikalische Objekte hängen zusammen und sind i.A. undurchsichtig.
- Entfernung entlang einer Oberfläche ändert sich kontinuierlich.

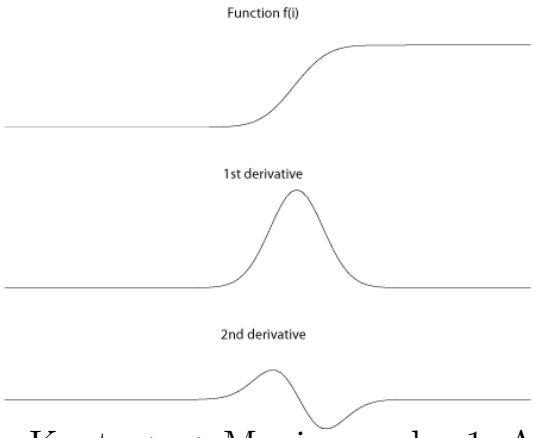
- Grauwerte entlang einer Oberfläche ändern sich nicht abrupt.
- Grauwerte an Objekträndern ändern sich abrupt.
- **Objektrand $\longleftrightarrow$ Unstetigkeit**
- bestimme Unstetigkeiten im Bild $\longrightarrow$ Hypothesen für Objektrand

Marr82 David Marr. *Vision.* Freeman, W.H., San Francisco, 1982.

E.3.2 Kantenprofil

Ein Kantenprofil gibt Auskunft über die Schärfe von Kanten in einem Bild. Je steiler ein Profil ansteigt desto schärfer d.h. ausgeprägter ist die Kante. Um diese Kanteninformationen zu detektieren, kann man in der 1. Ableitung der Kantenprofilfunktion das Maximum und in der 2. Ableitung des Kantenprofils die Nullstellen aufsuchen, welche einem steilen Kantenprofil im Bild entsprechen (siehe Abbildung E.3.2)

Kantenerkennung: 1 x differenzieren + Maximum suchen
2 x differenzieren + Nullstelle suchen



Kantenprofil,

1. Ableitung

2. Ableitung

Kante $\longleftrightarrow$ Maximum der 1. Ableitung $\longleftrightarrow$ Nullstelle der 2. Ableitung

Abbildung E.3: Kantenprofil

E.3.3 Kanten im Bild

In einem zweidimensionalen Bild können Kanten durch Bildung des Gradienten detektiert werden. Dieser ist richtungsbezogen und extrahiert im diskreten Wertebereich den größten Grauwertunterschied in einem Bild.

Ableitung in x-Richtung:

$$G_x = \frac{\partial f}{\partial x} \tag{E.1}$$

Ableitung in y-Richtung

$$G_y = \frac{\partial f}{\partial y} \tag{E.2}$$

Kanteninformation setzt sich zusammen aus der Kantenhöhe + Kantenrichtung θ. Die Kantenhöhe K kann folgend definiert werden:

$$K = ||(G_x, G_y)|| \tag{E.3}$$

Es gibt verschiedene Möglichkeiten für die Bestimmung des Gradienten:

$$||(G_x, G_y)|| = \begin{cases} \sqrt{G_x^2 + G_y^2} \\ |G_x| + |G_y| \\ \max\{G_x, G_y\} \end{cases} \tag{E.4}$$

Die Kantenrichtung θ kann wie folgt bestimmt werden:

$$\tan\theta = \frac{G_y}{G_x} \tag{E.5}$$

E.4 Kantendetektoren - Filter zur Kantendetektion

In diesem Abschnitt werden folgende Kantendetektoren (Filter) vorgestellt:

- Roberts Kantendetektor
- Prewitt Kantendetektor
- Sobel Kantendetektor
- Canny Kantendetektor

E.4.1 Roberts Kantendetektor [22]

Der Roberts Kantendetektor ist einer der ältesten Kantenoperatoren und benutzt zwei kleine (2×2) Filter, die den Gradienten der beiden Diagonalen schätzen können. Der Robertskantendetektor ist sensibel auf diagonal verlaufende Kanten, jedoch ist dieser Filter gering richtungsselektiv. (siehe Abbildung E.4)

Bildfenster	Kantenhöhe = Betrag	tan(Kantenrichtung θ)
$I = \begin{pmatrix} \ddots & & \\ & A & B \\ & C & D \\ & & \ddots \end{pmatrix}$	$\sqrt{(A-D)^2 + (B-C)^2}$	$\frac{A-D}{B-C}$
Bestimmung mit 2 Filtern:		
$G_x = I * \begin{pmatrix} +1 & 0 \\ 0 & -1 \end{pmatrix}$ $G_y = I * \begin{pmatrix} 0 & +1 \\ -1 & 0 \end{pmatrix}$	$\|\|(G_x, G_y)\|\|$	$\frac{G_y}{G_x}$

Abbildung E.4: Robertskantendetektor

E.4.2 Prewitt Kantendetektor

Der Prewitt Kantendetektor verwendet einen 3×3 Filter (siehe Abbildung E.5), der jeweils eine Mittelung über drei benachbarte Zeilen bzw. Spalten durchführt, welches einer sogenannten Glättung entspricht, die vor oder nach der Berechnung des Gradienten durchgeführt werden kann (Kommutativität der Faltung). Der Prewittkantendetektor ist sensibel auf vertikale und horizontale Kanten.

Faltungskern	Kantenhöhe	tan(Kantenrichtung θ)
$G_x = F * \frac{1}{3}\begin{pmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{pmatrix}$	$\|\|(G_x, G_y)\|\|$	$\frac{G_y}{G_x}$
$G_y = F * \frac{1}{3}\begin{pmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$		

Abbildung E.5: Prewitt Kantendetektor

E.4.3 Sobel Kantendetektor

Der Sobelkantendetektor ist dem Prewitt Kantendetektor sehr ähnlich. Der Unterschied besteht darin, dass bei der Glättung mehr Gewicht auf die zentrale Zeile bzw. Spalte gelegt wird (siehe Abbildung E.6)

Faltungskern	Kantenhöhe	tan(Kantenrichtung θ)
$G_x = F * \frac{1}{4}\begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix}$	$\|\|(G_x, G_y)\|\|$	$\frac{G_y}{G_x}$
$G_y = F * \frac{1}{4}\begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$		

Abbildung E.6: Sobel Kantendetektor

E.4.4 Kompass Kantendetektor

Die starke Richtungsabhängigkeit beim Erkennen von Kanten durch Filter d.h. die Kombination von G_x, G_y ist problematisch (z.B. ungleiches Gewicht für $\theta = 45^o$). Daher wurde beim Kompass Kantendetektor die Verwendung von Filtern für mehrere Richtungen angestrebt.
Die Arbeitsweise des Filters zeichnet sich durch die Bildung der 1. Ableitung G_θ in Abhängigkeit von mehreren Richtungen aus (z.B. $\theta \in \{0^o, 45^o, 90^o, 135^o, 180^o, 225^o, 270^o, 315^o\}$. Anschließend erfolgt durch die Kombination der Ergebnisse aller Richtungen eine gewichtete Mittelung in die jeweilige Normalrichtung:
Kantenhöhe $= \max\{G_\theta\}$
Kantenrichtung $\phi = \underset{\theta}{argmax}\{G_\theta | \theta \in \{0^o, 45^o, 90^o, 135^o, 180^o, 225^o, 270^o, 315^o\}\}$

Einige Beispiele für Kompass Kantendetektoren sind in Abbildung E.7 zu sehen.

G_{0^o}	G_{45^o}	G_{90^o}	G_{135^o}
$\frac{1}{5}\begin{pmatrix} -1 & +1 & +1 \\ -1 & -2 & +1 \\ -1 & +1 & +1 \end{pmatrix}$	$\frac{1}{5}\begin{pmatrix} +1 & +1 & +1 \\ -1 & -2 & +1 \\ -1 & -1 & +1 \end{pmatrix}$	$\frac{1}{5}\begin{pmatrix} +1 & +1 & +1 \\ +1 & -2 & +1 \\ -1 & -1 & -1 \end{pmatrix}$	$\frac{1}{5}\begin{pmatrix} +1 & +1 & +1 \\ +1 & -2 & -1 \\ +1 & -1 & -1 \end{pmatrix}$ ...

Abbildung E.7: Kompass Kantendetektoren

E.4.5 Canny- Kantendetektor [4]

Canny hat seinen Operator als optimalen Kantendetektor entworfen.

optimal $\longrightarrow$ Kriterien

— es gibt auch andere, die optimal bezüglich leicht anderer Kriterien sind.

- lokal stärkste Intensitätsänderung
- Störkanten glätten
- Parallelkanten vermeiden
- Kantensegmente zu Kantenfolgen zusammenfassen

Input: **Grauwertbild**,
Output: **Bild, das die Intensitätssprünge zeigt.**

Funktionsweise des Canny Operators

1. Glättung des ganzen Bildes mit einem **Gaußfilter** .
2. Hervorhebung starker Grauwertschwankungen durch einen 2D Gradientenoperator (1.Ableitung z.B. Roberts),
   d.h. Kanten werden zu Graten im Bild des Gradientenbetrags.
3. Verfolgen der Grate und Null setzen aller Pixel, die nicht am Grat liegen:
   $\longrightarrow$ dünne Linie (**non-maxima suppression**).

Der Verfolgungsprozess wird von 2 Schwellwerten kontrolliert: *T1* und *T2*, mit *T1* > *T2*.
Die Verfolgung beginnt an einem Gratpunkt höher als *T1*. Sie wird in beide Richtungen fortgesetzt bis die Höhe des Grates unter *T2* fällt. Dieser Vorgang ('Hysterese') soll das Zerfallen einer gestörten Kante in viele kleine Segmente verhindern.

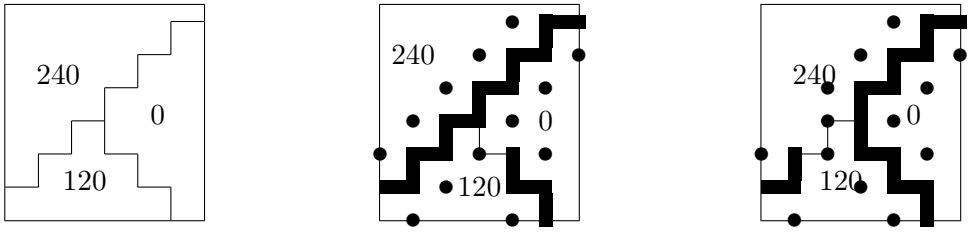
Canny Parameter (σ, T1, T2)

Der Canny Operator wird von drei Parametern bestimmt:

1. **die Breite des Gausskerns** σ (Schritt 1 Glättung)
2. den **oberen Schwellwert T1** und
3. den **unteren Schwellwert T2** für die Verfolgung der Grate.

	Vorteile	Nachteile
größeres σ	weniger Noise abhängig	weniger Details, ungenauere Kanten
T1 hoch	nur 'starke' Kanten	Kanten ohne starken Kontrast fehlen
T1 zu niedrig	auch 'schwache' Kanten	oft unerwünscht
T2 niedrig	wenig Unterbrechungen	
T2 zu hoch		gestörte Kanten können zerbrechen

Problem Y-Kreuzungen



Durch z.B. teilweise Verdeckung von Objekten im Bild können sich an einer Y-Kreuzung drei Grate treffen. Canny verbindet zwei dieser drei Grate zu einer Kante und lässt den dritten Grat in der Nähe enden.

Anwendung des Canny Operators mit den Parametern (1.0, 255, 1)

Durch den Canny Operator werden die meisten wichtigen Kanten erkannt und viele Details können wiedergegeben werden. Die Frage ist nur werden ALLE gebraucht?. In Abbildung E.8 ist im linken Bild E.8(a) in der linken unteren Spiegelecke eine Y-Kreuzung. In Bild E.8(b) sieht man die Problematik der y-Kreuzungen nach Anwendung des Canny Operators.

(a) Grauwertbild eines sich schminkenden Clowns

(b) Grauwertbild eines sich schminkenden Clowns nach der Anwendung von Canny (1.0,255,1)

Abbildung E.8: Anwendung des Canny Operators mit den Parametern (1.0, 255, 1)

Anwendung des Canny Operators mit den Parametern (1.0, 255, 220)

Die Erhöhung des T_2 Wertes bewirkt, dass die Kanten in mehrere Fragmente zerfallen, welches schlecht für die Weiterverarbeitung ist. Die vertikalen Kanten an der Wand sind somit unvollständig detektiert (siehe Abbildung E.9)

(a) Grauwertbild eines sich schminkenden Clowns (b) Grauwertbild eines sich schminkenden Clowns nach der Anwendung von Canny(1.0,255,220)

Abbildung E.9: Anwendung des Canny Operators mit den Parametern (1.0, 255, 220)

Anwendung des Canny Operators mit den Parametern (1.0, 128, 1)

Die Anwendung des Canny Operators mit diesen Parametern auf das Bild E.10(a) zeigt zusätzliche Kantenfragmente, z. T. auch Störungen, das Haar des Clowns jedoch hat nun Struktur (siehe AbbildungE.10(b)).

(a) Grauwertbild eines sich schminkenden Clowns (b) Grauwertbild eines sich schminkenden Clowns nach der Anwendung von Canny(1.0, 128, 1)

Abbildung E.10: Anwendung des Canny Operators mit den Parametern (1.0, 128, 1)

Anwendung des Canny Operators mit den Parametern (2.0, 128, 1)

Durch den erhöhten σ-Parameter erfolgt eine Glättung, welche in einer Reduktion des Kontrastes resultiert. (Abbildung E.11)

(a) Grauwertbild eines sich schminkenden Clowns (b) Grauwertbild eines sich schminkenden Clowns nach der Anwendung von Canny(2.0, 128, 1)

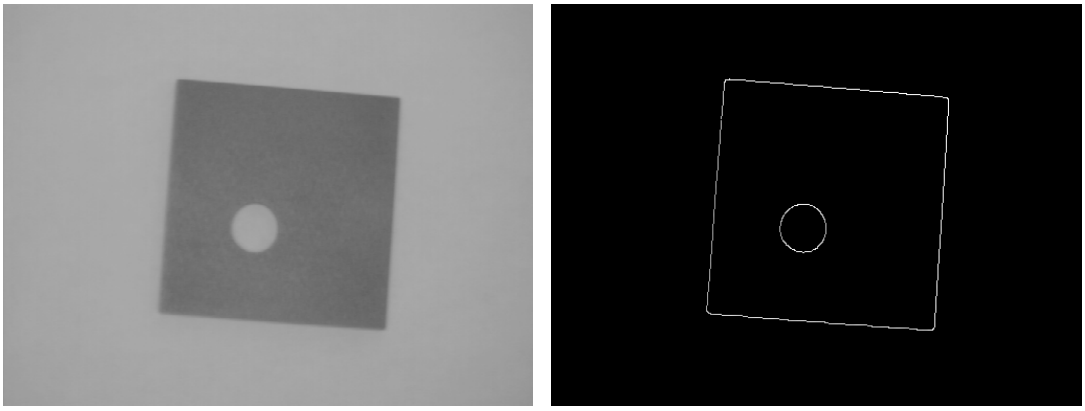
Abbildung E.11: Anwendung des Canny Operators mit den Parametern (2.0, 128, 1)

Künstliche Szenen - natürliche Szenen

Kanten in künstlichen Szenen sind oft schärfer und weniger komplex als in natürlichen Szenen, welches die Kantendetektion erleichtert (siehe Abbildung E.12). Die Gaussglättung kontrolliert den Detailreichtum und unterdrückt Störungen.

Objektkanten ohne Störungen ?

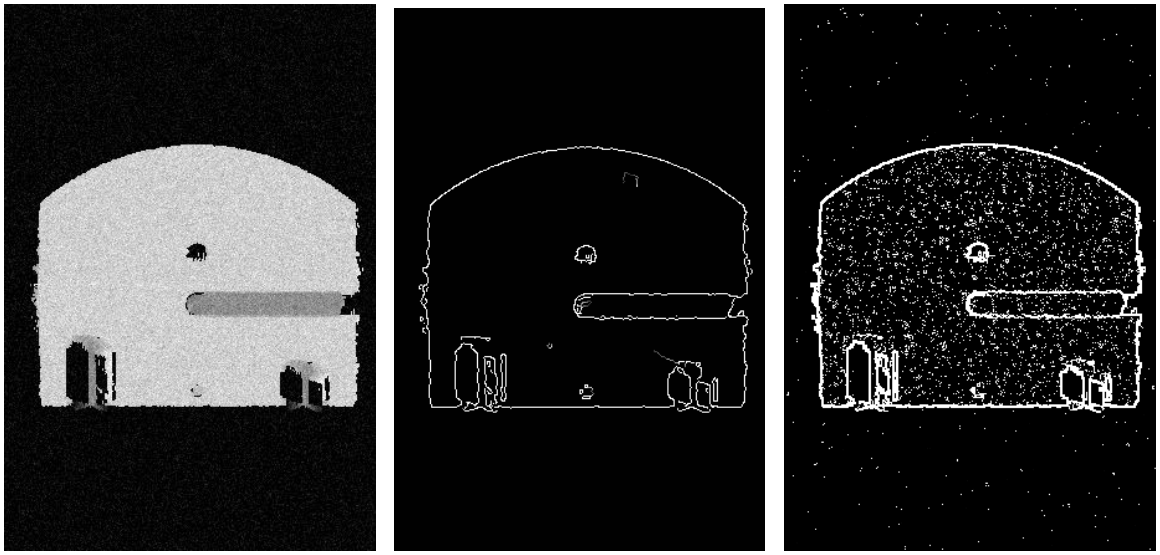
In Abbildung E.13 ist die Problematik der Kantendetektierung bei verrauschten Bildern visualisiert. Weder Roberts noch Sobel detektieren alle Objektkanten und eliminieren alle Störungen E.13(c). Hingegen



(a) Künstliche Szene (b) Künstliche Szene nach der Anwendung von Canny

Abbildung E.12: Anwendung des Canny Operators auf eine künstliche Szene

liefert der Canny Operator ein fast rauschfreies Kantenbild (Abbildung E.13(b)), d.h. dieser Kantendetektor ist weniger empfindlicher gegenüber Störungen, als der Sobel- oder Robertsdetektor.



(a) Grauwertbild I mit einem normalverteilten Störanteil $N(\sigma = 15)$

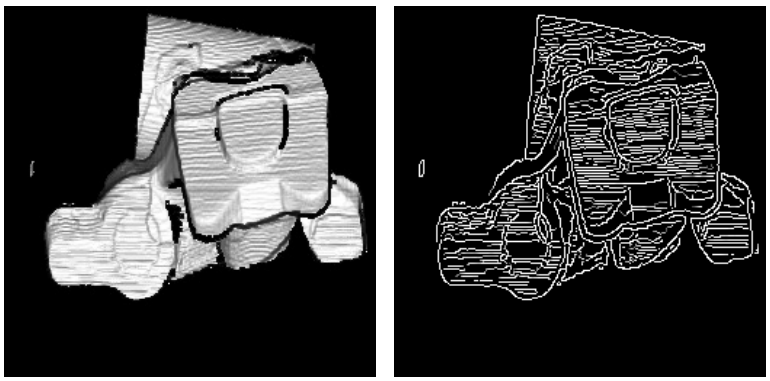
(b) Grauwertbild nach der Anwendung von Canny(σ=1.0)

(c) Grauwertbild nach der Anwendung von Sobel > 150.

Abbildung E.13: Vergleich des Canny Operators mit dem Robert bzw. Sobel Operator in Hinblick auf Empfindlichkeit bei verrauschten Bildern

Kontrolle des Detailreichtums

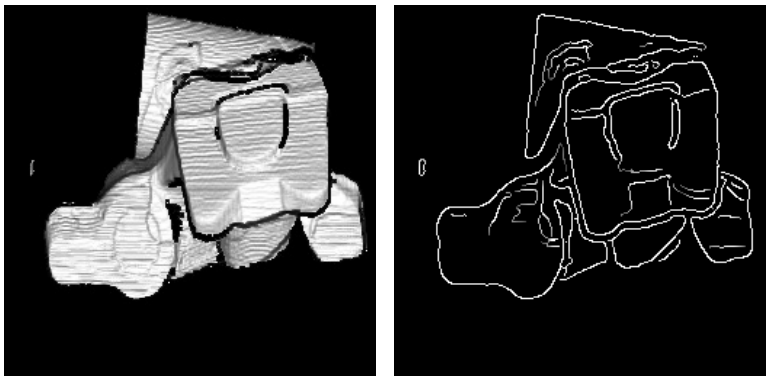
Viele Details - Wie kontrollieren wir den Detailreichtum?
Bei einer automatischen Erkennung durch Canny (1.0, 255, 1) werden alle Ränder detektiert auch jene von unebenen Oberflächen (siehe Abbildung E.14)
Durch richtige Parameterisierung kann dieser Effekt minimiert werden. Zum Beispiel werden bei Canny (1.8, 255,1) nicht mehr alle Kanten der unebenen Oberfläche erkannt, einige bleiben erhalten. Sie sind zwar schwächer, die Pixelwerte sind aber wegen der Sättigung dieselben (siehe Abbildung E.15). Um alle Objektgrenzen zu erhalten und alle anderen Kanten zu eliminieren kann man z.B. vor Anwendung des Canny Operators das Bild glätten und anschließend den Canny Operator mit den Parametern (1.8, 255, 1) anwenden (Abbildung E.16).



(a) Grauwertbild

(b) Grauwertbild nach der Anwendung von Canny(1.0,255,1)

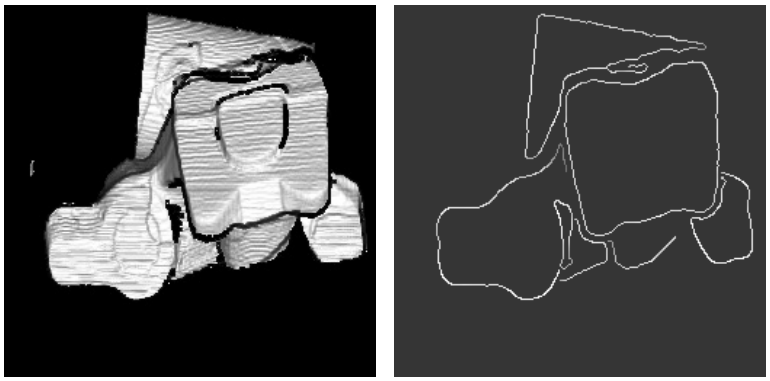
Abbildung E.14: Detailreiches Bild mittels Canny Operator



(a) Grauwertbild

(b) Grauwertbild nach der Anwendung von Canny(1.8,255,1)

Abbildung E.15: Detailarmes Bild mittels Canny Operator



(a) Grauwertbild

(b) Grauwertbild nach der Anwendung von Canny(1.8,255,1) $(0.25 * Bild)$

Abbildung E.16: Ausgewogene Anzahl an Details im Bild mittels Canny Operator

Interpretation des Resultats

Canny erkennt Intensitätssprünge, aber nicht jede Canny-Kante entspricht einer Objektkante (Abbildung E.17)

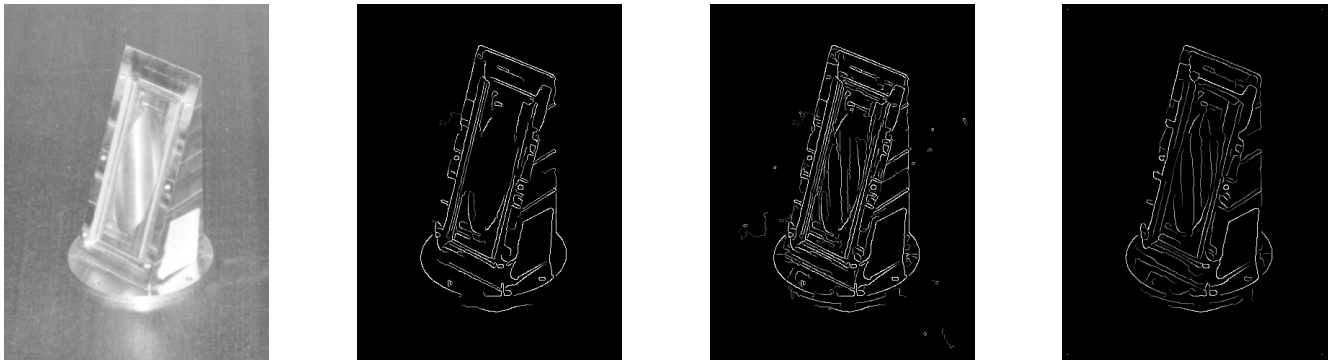
Referenzen

R. Boyle and R. Thomas *Computer Vision: A First Course*, Blackwell Scientific Publications, 1988, p 52.

J. Canny *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, Nov. 1986.

E. Davies *Machine Vision: Theory, Algorithms and Practicalities*, Academic Press, 1990, Chap. 5.

R. Gonzalez and R. Woods *Digital Image Processing*, Addison-Wesley Publishing Company, 1992, Chap. 4.



Original **Canny(1.0,255,1)** **...(1.0,150,1)** **...(2.0,100,1)**

Abbildung E.17: Interpretation des Canny Detektors

E.5 Linien

	Kante	Linie
Profil		
1. Ableitung Faltungskern	Maximum $(-1,+1)$	0-Durchgang $\frac{1}{2}(-1,+2,-1)$

Zur Unterscheidung zwischen Linie und isoliertem Punkt (Spot):
Nicht-lineare Zusatzbedingung:

$$G(x,y) = \begin{cases} B(x,y) * \text{Filter} & \text{wenn} \quad \begin{array}{|ccc|} \hline A & B & C \\ \wedge & \wedge & \wedge \\ D & E & F \\ \vee & \vee & \vee \\ G & H & I \\ \hline \end{array} \\ 0 & \text{sonst} \end{cases}$$

E.6 Hough-Transformation, Radon-Transformation [6]

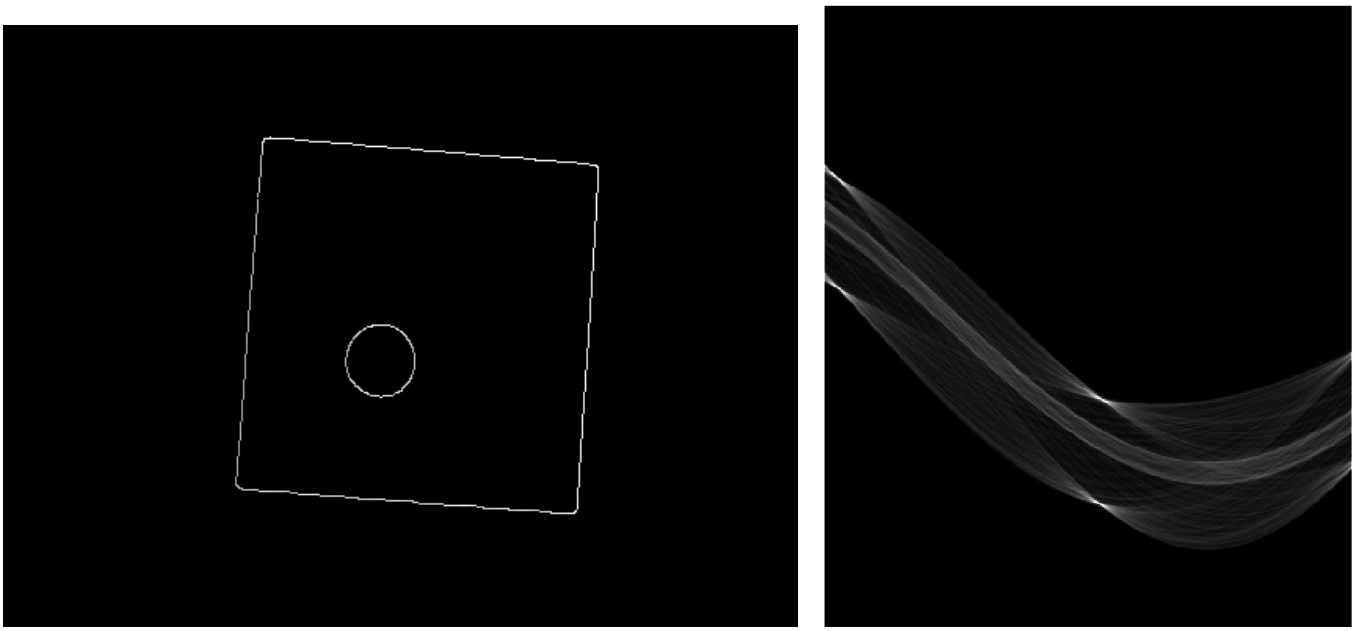
Die Hough-Transformation wurde von Paul.V.C. Hough entwickelt und ursprünglich als US-Patent 1962 publiziert (http://www.uspto.gov/patft/index.html). Sie dient zur Detektion von geometrischen Formen, die parameterisierbar sind (z.B. Geraden (-Segmente), Kreise, Ellipsen, etc.). Am Beispiel einer Geraden soll die Houghtransformation erläutert werden. Laut Definition ist eine Gerade eine unendlich lange, unendlich dünne und in beide Richtung unbegrenzte Linie, die parametrisierbar ist, d.h. sie kann durch **2 Parameter** beschrieben werden:

- **Radius r**, dies ist der Abstand der Normalen der Geraden vom Ursprung und kann in Hesse'scher Normalform angeschrieben werden:

$$r = x\cos\theta + y\sin\theta \tag{E.6}$$

- **Winkel** θ, dies ist jener Winkel der zwischen der Normalen der Geraden und der x-Achse aufgespannt wird.

Die Parameter (θ, r) können in einem Parameterraum dargestellt werden, auch Houghraum genannt, wo auf der x-Achse θ und auf der y-Achse r aufgetragen wird. Jeder Punkt im Houghraum entspricht einem



(a) Bildraum (b) quantisierter Houghraum

Abbildung E.18: Houghtransformation

Parameterpärchen (θ, r), welches im Bildraum einer Geraden entspricht (siehe Abbildung E.19)
Im Houghraum bilden Punkte eine Sinuskurve, wenn sich die entsprechenden Geraden im Bildraum in einem Punkt schneiden. Ein Schnittpunkt der Sinuskurven im Houghraum entspricht jenen Punkten im Bildraum, die dort auf einer Geraden liegen, d.h. wenn sich **N** Sinus-Kurven an einer Stelle (θ_k, r_k) im Houghraum schneiden, dann liegen auf der dazugehörigen Geraden im Bildraum **N** Bildpunkte.
Ziel der Houghtransformation ist es Koordinaten jener Punkte im Parameterraum zu finden, in denen sich die meisten Sinus-Kurven schneiden, um eine Gerade im Bildraum zu detektieren. In Abbildung E.18 ist in E.18(a) der Bildraum dargestellt und in E.18(b) der dazugehörige quantisierte Houghraum.

Die Hough-Transformation stellt einen Spezialfall der **Radontransformation** dar [6]. Die Radontransformation findet Anwendung in der Computertomographie. Dort wird diese für die Röntgenabsorption, die für jedes Gewebe im menschlichen Körper verschieden ist, bestimmt und aus diesen Werten wird ein zweidimensionales CT - Bild gewonnen.

E.7 Algorithmus der Hough-Transformation

Durch folgende Schritte kann die Houghtransformation durchgeführt werden:

1. Zuerst erfolgt eine **Kanten/Linien Detektion** durch z.B. Filter oder einen Threshold. Diese liefert Koordinaten (x_k, y_k) der Punkte, mit hohem Kontrast.

2. **Houghraum quantisieren:** Durch Quantisierung des kontinuierlichen Houghraums wird bestimmt, in welchem Abstand Δr und in welchem Winkelabstand $\Delta\theta$ Punkte im Bildraum betrachtet werden:

   $\theta_i = i\Delta\theta \ , \ i = 0, 1, \ldots (n-1)$ n-1, da i bei 0 beginnt

   $r_j = j\Delta r \ , \ j = 0, 1, ..., (m-1)$ m-1, da j bei 0 beginnt

   $\theta_{max} = max(\theta_i)$

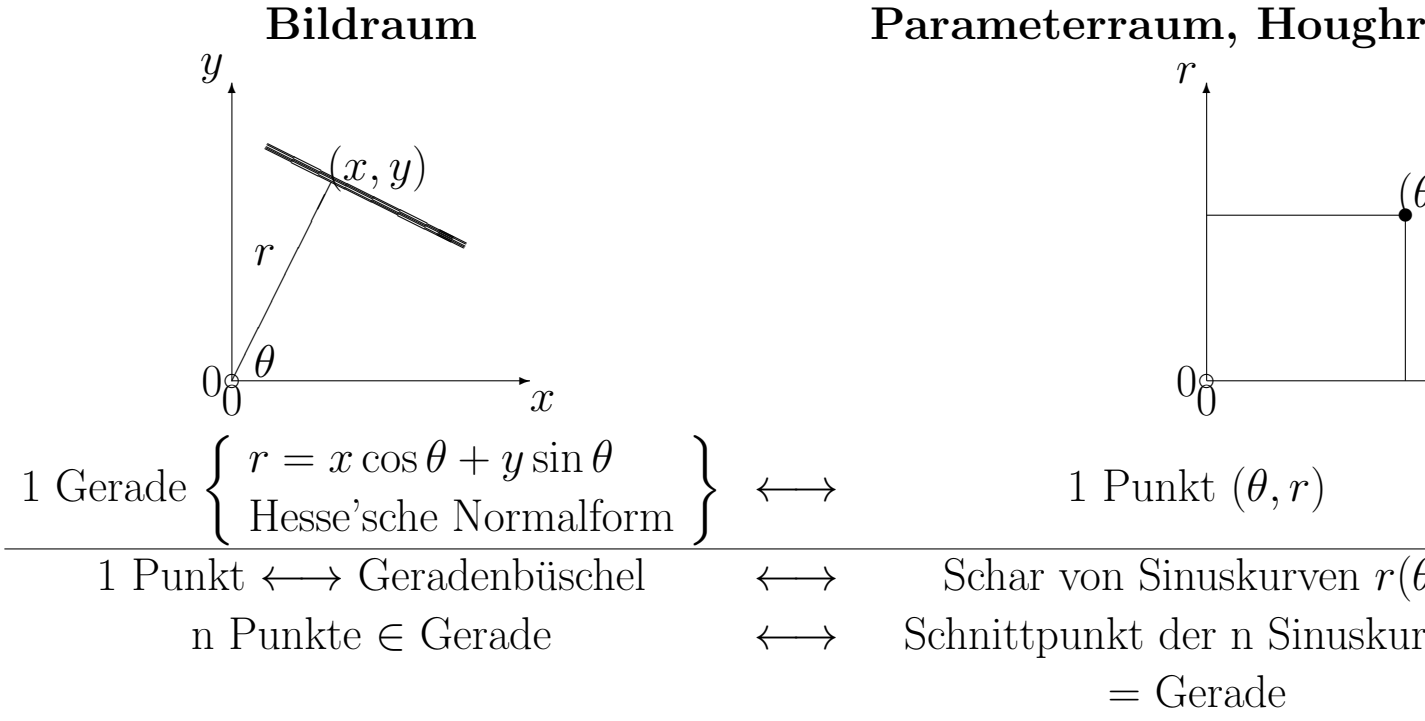


Abbildung E.19: Beziehung zwischen Bildraum und Parameterraum, Houghraum

$r_{max} = max(r_i)$

$n = \frac{\theta_{max}}{\Delta\theta}$

$m = \frac{r_{max}}{\Delta r}$

3. Zur Berechnung der Überschneidungen im Parameterraum wird ein n × m Akkumulator-Array benötigt und folgend initialisiert: $H(i, j) = 0$. Das Array stellt den diskretisierten Houghraum dar.

4. Für jeden Punkt (x_k, y_k), wird im Akkumulator-Array (diskretisierter Houghraum) der zugehörige Sinus gezeichnet , indem die sinuskurvendarstellenden Felder **inkrementiert** (Erhöhung des vorhandenen Wertes um eins) werden. Dies gilt nur bei der Betrachtung der Parameter θ und r (Hesse'sche NF). Werden z.B. die Parameter k und d unter Betrachtung der Geradengleichung: $y = k * x + d$ für die Houghtransformation verwendet, wird z.B. anstatt des Sinus eine Gerade gezeichnet).

5. Suche Maximum in $H(i, j) \mapsto$ man erhält Werte $H(i_{max}, j_{max})$ für i und j, welche jene Arrays adressieren, welche einen maximalen Wert aufweisen (entspricht der maximalen Anzahl an Sinus-Kurven die sich an diesem Punkt schneiden).

6. Gerade im Bildraum finden durch: $j_{max}\Delta r = x \cos\theta_{i_{max}} + y \sin\theta_{i_{max}}$

7. Tracking: suche Anfangs- und Endpunkte des Geradensegments im Bild.

E.7.1 Beispiel Houghtransformation

1.Schritt - Kantendetektion
Folgende N Punkte $P_k = (x_k, y_k)$ wurden im Bildraum (siehe Abbildung E.20(a))durch die Kantendetektion aufgefunden, die zu einer Geraden gehören (k=1,...,N):N=5;

- $x_1 = 3$, $y_1 = 7$,
- $x_2 = 4$, $y_2 = 6$,
- $x_3 = 5$, $y_3 = 5$,
- $x_4 = 6$, $y_4 = 4$,
- $x_5 = 7$, $y_5 = 3$,

$\theta_1 = \theta_2 = ... = \theta_6 = 45^0$ da alle Punkte auf der gleichen Geraden liegen

2.Schritt - Quantisierung des Houghraumes
Winkelabstand: $\Delta\theta = 3^0$
Abstand: $\Delta r = 2$

Berechnung von n und m:

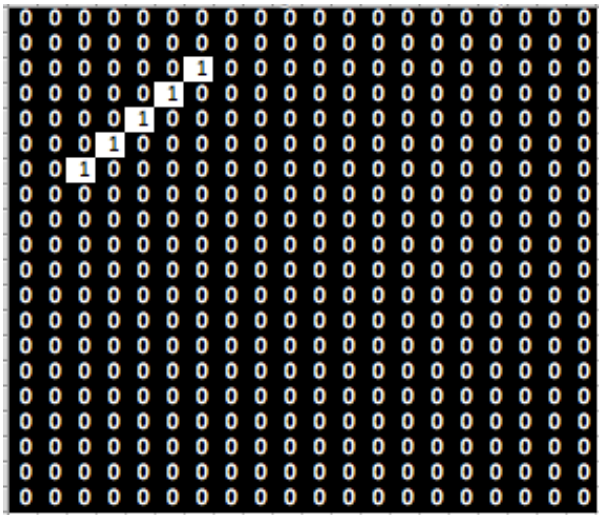
$i = 0, 1, ..., (n-1)$
Es werden nur Winkel im Bereich von 0^0 bis 180^0 (d.h. $\theta_i = 0, .., 180$) betrachtet, daher ergibt sich für n: n = $\frac{\theta_{max}}{\Delta\theta} = \frac{180}{\Delta\theta} = \frac{180}{3} = 60$ d.h. $\theta_i = 0, .., 180 \rightarrow i = 0, 1, ..., 59$

$j = 0, 1, ..., (m-1)$
Der am maximal möglich vorkommende Radius ist bei diesem Bildraum der Größe 20×20 Pixel: $r_{max} = \sqrt{20^2 + 20^2} = 28.2843 \approx 28$. In Abbildung E.20 sind auf der y-Achse 29 Pixel aufgetragen, welches sich erklären lässt, dass bei einem Δr von 2 wir genau 29 Positionen auswerten müssen um 28 Schritte für alle Radien von $-r_{max}$ bis $+r_{max}$ zu betrachten. Diese 29 Positionen entsprechen den 29 Pixeln. 28 ist die Anzahl der Zwischenräume (entspricht den 28 Schritten) zwischen den Pixeln.
Wir betrachten einen Winkel θ von 0^0 bis 180^0. Der Radius varriert dabei von $-r_{max}$ bis $+r_{max}$. Somit werden insgesamt 360^0 betrachtet. $m = \frac{2*r_{max}}{\Delta r} = 28 \rightarrow j = 0, 1, ..., 27$

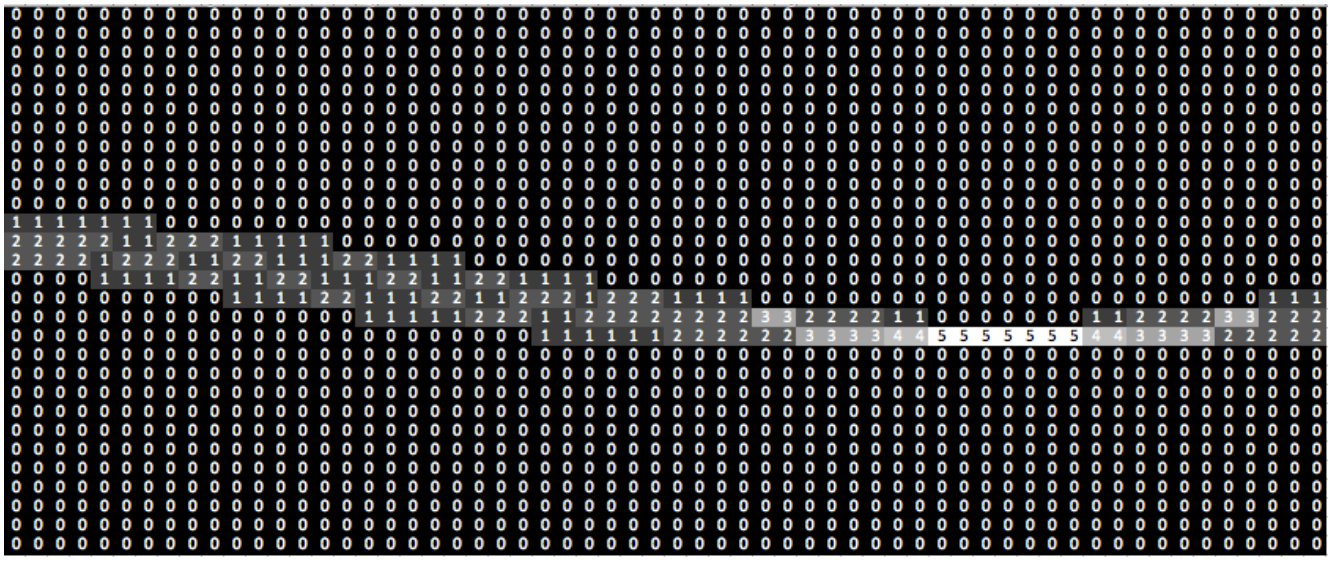
3.Schritt
Der Akkumulator wird mit 0 initialisiert: H(i,j) = 0;

4.Schritt
Für jeden Punkt (x_k, y_k) wird im Akkumulator-Array der zugehörige Sinus gezeichnet, indem die sinuskurvendarstellenden Felder inkrementiert werden (siehe Abbildung E.20) **5. Schritt**
Häufungspunkt: $j_{max} = -3$, $i_{max} = 16$
H(16, -3) = 5, d.h. 5 Sinuskurven schneiden sich in diesem Punkt.
In Abbildung E.20 sind 5 Häufungspunkte zu erkennen, welches sich auf einen Quantisierungsfehler zurück führen lässt. Als Häufungspunkt wurde jener von den 7 gewählt, der in der Mitte der Gruppe liegt.

6. Schritt Gerade im Bildraum finden durch Berechnung von r und θ
$\theta_{i_{max}} = i_{max} * \Delta\theta = 48^0$
Aufgrund der im 5. Schritt besprochenen Quantisierungsfehler, stellen 48^0 eine gute Näherung an die ursprünglich gewählten 45^0 dar.



(a) Bildraum: Bild wurde in MATLAB dargestellt daher liegt der Ursprung links oben



(b) diskretisierter Houghraum H: Dimension der x-Achse -90^0 bis $+90^0$, Dimension der y-Achse $-r_{max}$ bis $+r_{max}$

Abbildung E.20: Hough-Algorithmus Schritt 4 des Beispiels E.7.1

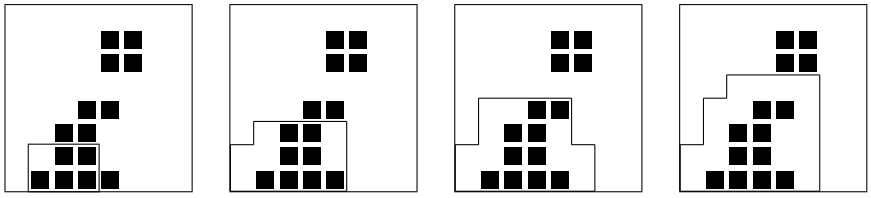
$r = j_{max} * \Delta r = -6$
Einen Punkt der Gerade kann man sich berechnen durch folgende Umrechnung:
$x = r * cos(\theta_{i_{max}})$
$y = r * sin(\theta_{i_{max}})$
Richtungsvektor des Normalvektors $\vec{n} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3,8 \\ 4,6 \end{pmatrix}$
Richtungsvektor der Geraden $\vec{g} = \begin{pmatrix} -y \\ x \end{pmatrix} = \begin{pmatrix} -4,6 \\ 3,8 \end{pmatrix}$

E.8 Region growing (Regionenwachstum)

- Threshold $\mapsto$ oft mehr als eine zusammenhängende Region

- Region Growing ist ein sequentieller Segmentationsprozeß

1. $S \in$ Bild: Saatpixel, Saatregion; $\mathbf{R}:= S$;
   Eine Saatregion bzw. ein Saatpixel R wird definiert.
2. while $S \neq \emptyset$ do
3. $S' = \boxed{\Gamma(R)} \setminus R$;
   Betrachtung der Nachbarn von R und Abspeicherung dieser in der Variablen S'
4. $\mathbf{S} = \text{homogen}(S')$;
   Vergleich der Nachbarn mit R anhand eines definierten Homogenitätskriteriums (z.B. ein bestimmter Grauwertebereich) und Abspeicherung in S von jenen, die das Kriterium erfüllen.
5. $R = S \cup R$
   Vereinigung der ausgewählten Nachbarn S mit der Saatregion zu einer neuen Region.



(a) 1. Iteration (b) 2. Iteration (c) 3. Iteration (d) 4. Iteration

Abbildung E.21: Ablauf Region growing

E.9 Split and Merge

1. Startregion = ganzes Bild
2. **Split:** Zerlege JEDE inhomogene Region, d.h. wenn ein Homogenitätskriterium mit Grenzwerten zwischen 0 (entspricht homogen) und 1(entspricht inhomogen) über einem Schwellwert liegt, $(H(R) > t)$ in Teilregionen.
3. **Merge:** Vereinige benachbarte Regionen (R_1, R_2), wenn diese ein Homogenitätskriterium erfüllen, d.h. Die Homogenität der vereinten Regionen ist größer als die grösste auftretende Homogenität der einzelnen Regionen: $H(R_1 \cup R_2) > \max(H(R_1), H(R_2))$
4. iteriere Split & Merge bis keine Änderung mehr auftritt.

E.10 Relaxation

Pixel in Bild $\mapsto m$ Klassen

1. Bestimme Maß für Klassenzugehörigkeit (für ALLE Pixel und ALLE Klassen !).
2. Prüfe Kompatibilität der Zuordnungen aller Nachbarn
3. Verbessere/verschlechtere Klassenzugehörigkeit je nach Kompatibilität.
4. Wiederhole Schritte 2 und 3 einige Male.

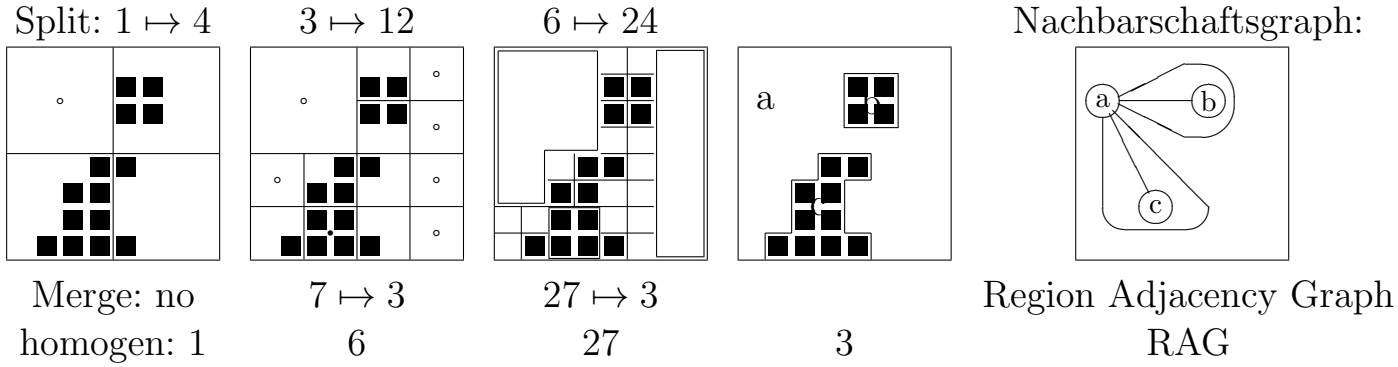


Abbildung E.22: Ablauf Split and Merge

E.10.1 Beispiel Relaxation

Durch Definition eines Schwellwertes können die Pixel im Bild in zwei Klassen eingeteilt werden (z.B. HELL, DUNKEL). Es stellt sich nur die Frage: Wie erhält man den optimalen Schwellwert? Die lokalen Schwellwerte können durch die Methode der Relaxation in den folgenden Punkten geschätzt werden:

1. Initialisierung: Wahrscheinlichkeiten für jedes Pixel werden im ersten Iterationsschritt (i=0) bestimmt: $P_H^{(i=0)}(X) = \frac{g_X}{g_{max}}$, $P_D^{(i=0)}(X) = 1 - P_H^{(i=0)} \qquad \in [0,1]$
   i...zeigt die Anzahl der Iterationsschritte an, die schon gemacht wurden und fängt bei 0 an.
   $P_H^{(0)}$...Wahrscheinlichkeit, dass das Pixel X **H**ell ist
   $P_D^{(0)}$...Wahrscheinlichkeit, dass das Pixel X **D**unkel ist
   g_X...Grauwert des Pixels X, g_{max}...maximale im Bild vorkommende Grauwert

2. Um die Wahrscheinlichkeit $P_H(X)$ d.h. Klassenzugehörigkeit eines Pixels X zu verbessern, wird nun auch jeweils ein Nachbarpixel dessen betrachtet. Es wird eine paarweise Berechnung von Koeffizienten der Überganswahrscheinlichkeitens (Wahrscheinlichkeit, dass das betrachtete Pixel in diesem Iterationsschritt der Klasse des Nachbarpixels zugeteilt wird - unabhängig von seiner Klasse) für diese 2 benachbarten Pixel (z.B. P und Q) durchgeführt : | P | Q |
   c_{PQ}...Koeffizient der die Übergangswahrscheinlichkeit von Pixel P zum Nachbarpixel Q angibt und dem in Abhängigkeit von der Pixeleigenschaften (Hell oder Dunkel) ein Wahrscheinlichkeitsparameter α oder β zugewiesen wird (siehe Tabelle E.1).

Tabelle E.1: Koeffizienten der Überganswahrscheinlichkeiten für ein Pixel P zu dessen Nachbarpixel Q

c_{PQ}	Q hell	Q dunkel
P hell	$c_{HH} = \alpha$	$c_{HD} = -\beta$
P dunkel	$c_{DH} = -\beta$	$c_{DD} = \alpha$

Parameter $\alpha, \beta \in [0,1]$

3. Die in Schritt 2 berechneten Koeffizienten werden nun verwendet um paarweise (Hell und Dunkel) Inkremente (=Betrag der Änderung eines Grauwertes)
   $\Delta g_H(A), \Delta g_D(A), \ldots, \Delta g_H(D), \Delta g_D(D)$ für alle 4 Nachbarn zu bilden:

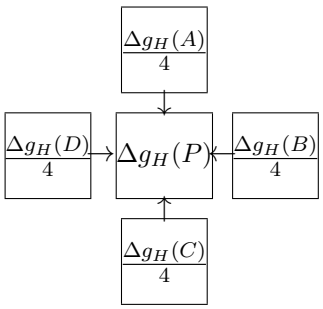
	A	
D	P	B
	C	

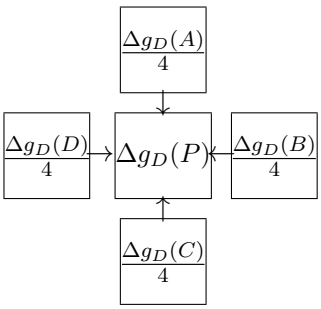
$$\Delta g_H(A) = c_{HH}P_H(A) + c_{DH}P_D(A) = \alpha P_H(A) - \beta(1 - P_H(A)) = P_H(A) \cdot (\alpha + \beta) - \beta$$

$$\Delta g_D(A) = c_{HD}P_H(A) + c_{DD}P_D(A) = -\beta P_H(A) + \alpha(1 - P_H(A)) = \alpha - P_D(A) \cdot (\alpha + \beta)$$

$\Delta g_H(A)$... gibt den Betrag der Änderung des Grauwertes an, wenn Pixel A (unabhängig ob hell oder dunkel) zu einem hellen Pixel übergeführt wird.
$\Delta g_D(A)$... gibt den Betrag der Änderung des Grauwertes an, wenn Pixel A (unabhängig ob hell oder dunkel) zu einem dunklen Pixel übergeführt wird.

4. Inkrement $q_H, q_D, \ldots$ aus allen 4 Nachbarn:





$$\Delta g_H^{(i)}(P) = \frac{g_a+g_b+g_c+g_d}{4 \cdot g_{max}}(\alpha + \beta) - \beta \qquad \Delta g_D^{(i)}(P) = \alpha - \frac{g_a+g_b+g_c+g_d}{4 \cdot g_{max}}(\alpha + \beta)$$

$$\Delta g_H^{(i)}(P) = \frac{g_a+g_b+g_c+g_d}{4 \cdot g_{max}}(\alpha + \beta) - \beta = \frac{1}{4} \cdot (P_H(A) + P_H(B) + P_H(C) + P_H(D))(\alpha + \beta) - \beta$$
$$\Delta g_D^{(i)}(P) = \alpha - \frac{g_a+g_b+g_c+g_d}{4 \cdot g_{max}}(\alpha + \beta) = \alpha - \frac{1}{4} \cdot (P_H(A) + P_H(B) + P_H(C) + P_H(D))(\alpha + \beta)$$

5. Verbessere Klassenzugehörigkeit eines Pixels durch Neuberechnung dessen Klassenzugehörigkeitswahrscheinlichkeit, solange, bis ein Abbruchkriterium (Schwellwert) für eine der Wahrscheinlichkeiten erreicht wird:

$$P'_H(P) = P_H^{(i)}(P) \cdot (1 + \Delta g_H^{(i)}(P)) \qquad P'_D(P) = P_D^{(i)}(P) \cdot (1 + \Delta g_D^{(i)}(P))$$
$$P_H^{(i+1)}(P) = \frac{P'_H(P)}{P'_H(P)+P'_D(P)} \qquad P_D^{(i+1)}(P) = \frac{P'_D(P)}{P'_H(P)+P'_D(P)}$$

E.11 Zusammenfassung

- Die Segmentation zerlegt ein Bild in homogene Teilregionen.
- Bestimmung homogener Regionen: Pixelklassifikation, Region Growing, Split & Merge.
- Kanten trennen homogene Regionen, haben eine Höhe und eine Richtung.
- einige Kantendetektoren: Roberts, Sobel, Prewitt, Canny.
- Die Houghtransformation akkumuliert Geradenparameter im Parameterraum.
- Die Relaxation verbessert iterativ die Kompatibilität der Klassenzugehörigkeiten.

Kapitel F

Überblick Bildanalyse

F.1 Griff in die Kiste

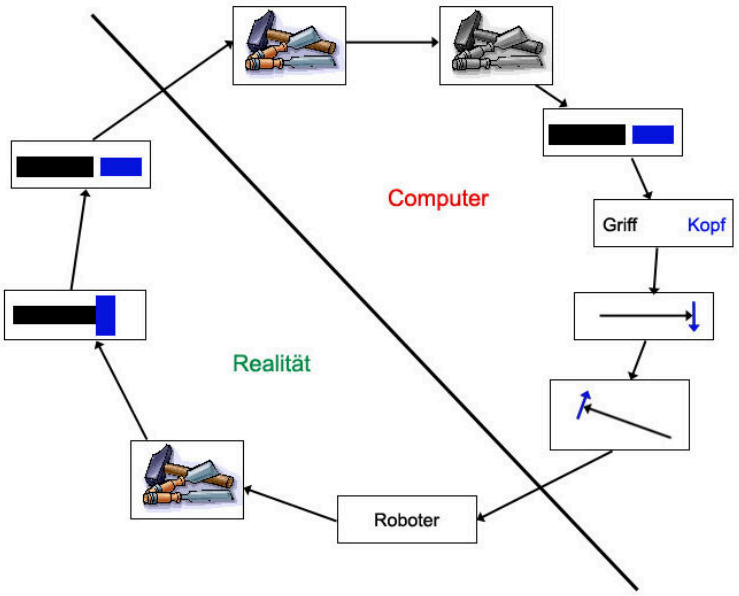


Abbildung F.1: Bildanalysekreis 1

Die menschliche Wahrnehmung unterscheidet sich von Computer Vision, wie in Abbildung F.1 dargestellt, grundlegend. Die dem Roboter übertragene Aufgabe greife den Hammer, ergibt sich von einem Betrachter aus gesehen folgendermaßen: der Roboter erkennt den Hammer und greift ihn. Was der Betrachter allerdings nicht sieht: Um den Auftrag auszuführen muss der Roboter erst einmal den Hammer unter den anderen Werkzeugen identifizieren. Dazu muss der Roboter in einer Aufnahme der Umge-

bung (Abbildung der Werkzeuge) den Hammer erkennen. Dem Roboter sind Eigenschaften des Hammers (Form, Anzahl der Einzelteile, etc.) bekannt. Unter Berücksichtigung einer gedrehten Lage des Hammers in der Realität im Vergleich zu dem, dem Roboter bekannten Modell des Hammers, detektiert der Roboter so das gewünschte Werkzeug.

F.2 Der BILDANALYSEKREIS (2)

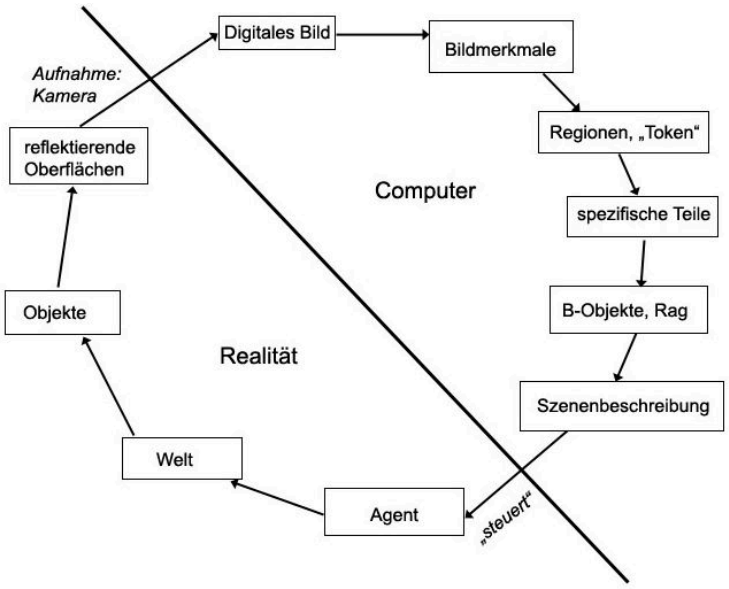


Abbildung F.2: Bildanalysekreis 2

Die reale Welt besteht aus vielen Objekten. Jedes Objekt besitzt eine reflektierende Oberfläche. Diese kann auf einem digital Bild (Aufnahme mittels Kamera) festgehalten werden. Der Bildanalysekreis in Abbildung F.2 zeigt die Verarbeitung eines solchen digitalen Bildes. Dieses lässt sich beschreiben durch Bildmerkmale, welche das Bild in Regionen teilen. Spezifische Teile eines Bildes setzen sich aus Bildregionen zusammen und lassen sich wiederum zu Bildobjekten zusammensetzen. Auf Grund der Objekte des Bildes kann dann eine Szenenbeschreibung gebildet werden.

F.3 Der BILDANALYSEKREIS (3)

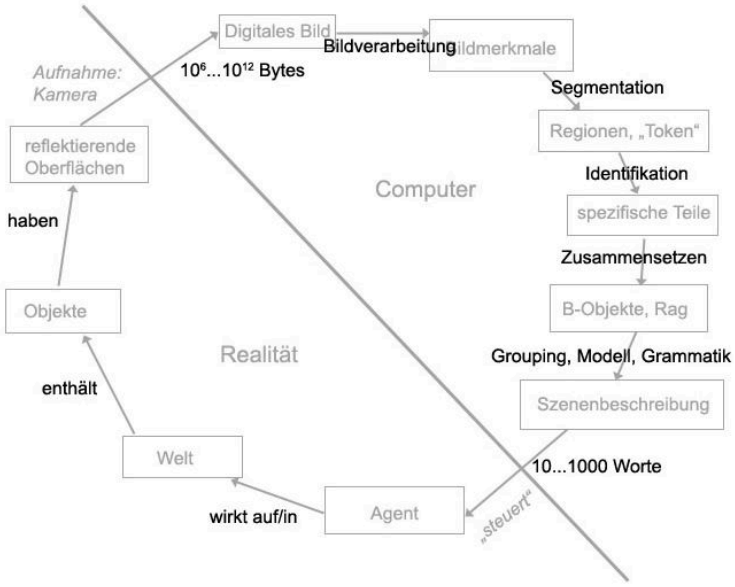


Abbildung F.3: Bildanalysekreis 3

Abbildung F.3 zeigt die zur Gewinnung der zuvor beschriebenen Bildelemente notwendigen Operationen. Mittels Bildverarbeitung werden Bildmerkmale eines Bildes gewonnen. Durch Segmentation kann das Bild dann in Bildregionen eingeteilt werden. Anschlieende Identifikation und entsprechendes Zusammensetzen liefert Bildobjekte. Diese können mit Hilfe von Gruppierungen, Grammatiken und vorhandenen Modellen beschrieben werden.

F.4 BILDANALYSE Methoden (4)

Zur Bildanalyse werden im Rahmen der Bildverarbeitung zB.: folgende bekannte Methoden verwendet: Glätten (siehe Abschnitt D.12, Seite 43) und Schrfen (siehe Abschnitt D.5, Seite 39). Bei der Segmentation finden folgende besprochene Methoden ihre Anwendung: Hough Transformation (siehe Abschnitt E.5, Seite 62), Region Growing (siehe Abschnitt E.8, Seite 66) und Relaxation (siehe Abschnitt E.10, Seite 67). Folgende Verfahren werden im Rahmen dieses Skriptums noch besprochen: Morphologie (siehe Abschnitt G.17, Seite 95), Connected-Component-Labeling (siehe Abschnitt G.15, Seite 93) und Thinning (siehe Abschnitt G.24, Seite 99).

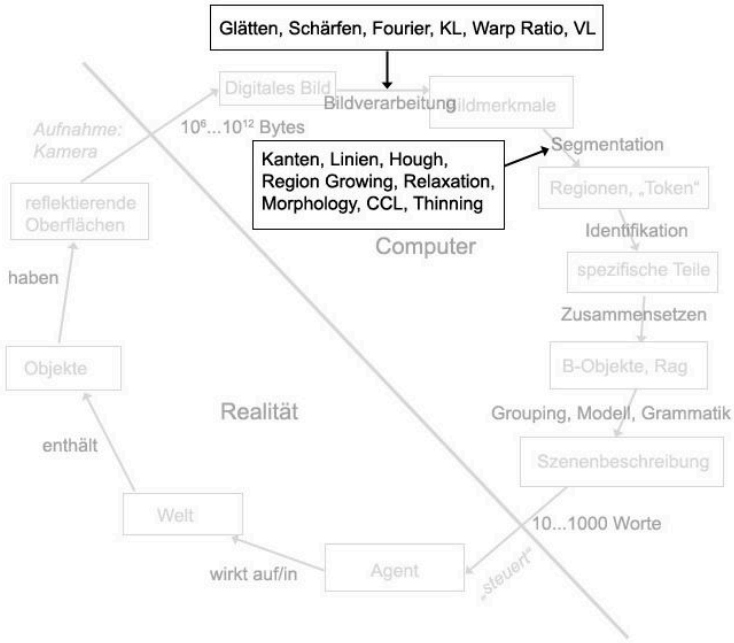


Abbildung F.4: Bildanalysemethoden

F.5 Zusammenspiel High-level-Low-level, Einsatz von Wissen

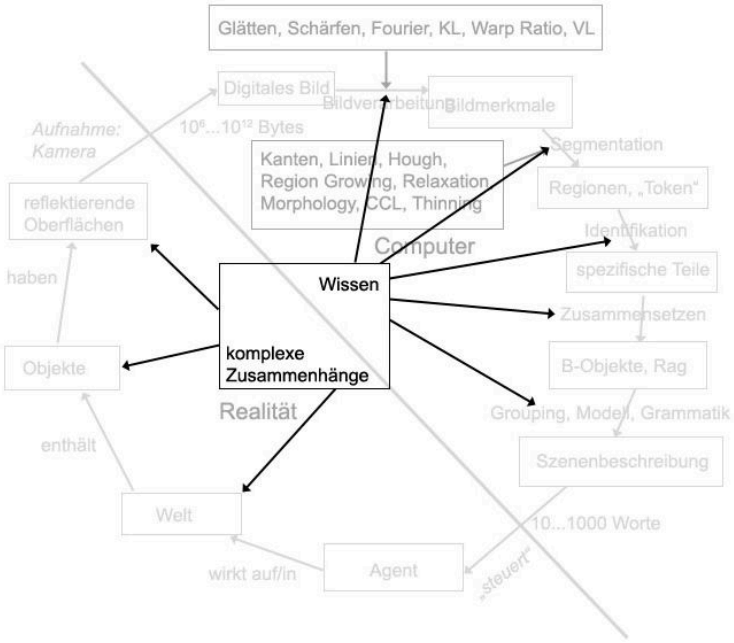


Abbildung F.5: Zusammenspiel

F.6 Charakteristika digitaler Bilder, wichtige Probleme

- viele MB an Daten

- hohe Redundanz der Daten
- komplexe Zusammenhänge

Wichtige PROBLEME

- Rauschen (Noise)
- verlässliche Erkennung
- unabhängig von Veränderungen in der Lage,
der Orientierung,
der Größe
- teilweise Verdeckungen
- hohe Rechenkomplexität

F.7 einige Fakten zur Geschichte

ab 1960 Digitale Bildverarbeitung mit Computer

1968 1. Zeitschrift: *Pattern Recognition* (Pergamon)

1969 1. Textbuch: *Picture Processing by Computer* (A. Rosenfeld)

1970 1. International Conference on Pattern Recognition (*ICPR*)

ab 1977 *CVPR*

1978 International Association for Pattern Recognition (*IAPR*, 43 Länder)

1979 IEEE Transactions on Pattern Analysis and Machine Intelligence (*PAMI*)

1981/1987 Österreichische Arbeitsgemeinschaft für Mustererkennung (*ÖAGM*, 70 Mitglieder)

1984,1994,2005,2012 *DAGM* Symposien in Österreich, 200 Teilnehmer

1987 1. International Conference on Computer Vision (*ICCV*)

1996 *13. ICPR* in Wien, 1000 Teilnehmer

F.8 Early CV @ TU Graz

1974 Franz LEBERL - NASA JPL

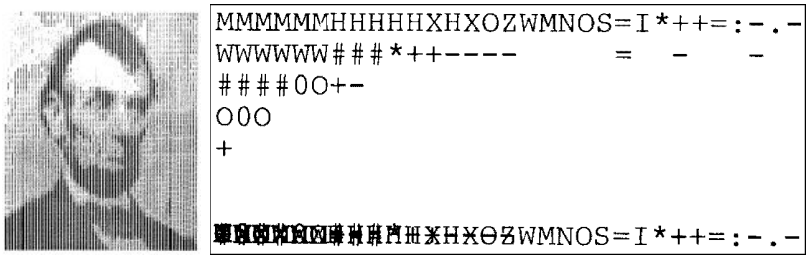
1975 Johannes G. MOIK - NASA Goddard SFC

1978/79 Leberl & Kropatsch: DIBAG @ JR
CV @ TU Graz

F.9 DiBaG around 1980

Digitale **B**ild **A**uswertung **G**raz
Erste BV Software von Joachim Wiesel (Karlsruhe): DIDAK
transformiert/erweitert zu DIBAG

Input	Output
künstlich erzeugte Matrizen	Arrays von Zahlen
Magnetbänder von LANDSAT MSS	Überdruck einiger m^2
kleine gespendete Bilder	4-fach Ausdruck, Diapositive

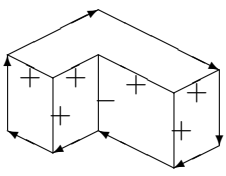


F.10 CV Systems (Linda G. Shapiro, 1983)

1965 Roberts' PhD@MIT: erstes CV System

1968 Guzman's PhD@MIT: **SEE**, blocks world

1971-77 Binford, Agin, Nevatia, Marr: **generalized cylinders**



1971-78 Huffman, Clowes, Waltz label line drawings of 3D objects

1976 Rosenfeld, Hummel, Zucker: **MSYS**, relaxation

1974-78 Hanson, Riseman **VISIONS**, 'schema'

1981 Brooks' PhD@Stanford: **ACRONYM**, enthält Graphen für:
Objekte, Einschränkungen, Vorhersagen, Beobachtungen, Interpretationen.

1982 Haralick, Shapiro@VPI: **GIPSY**, rel. Datenstrukturen, spatial reasoning

F.11 CV in Austria

1980 DIBAG @ JOANNEUM RESEARCH: Digitale Bildauswertung Graz (10...30 researchers)

1981/1987 Österreichische Arbeitsgemeinschaft für Mustererkennung (*ÖAGM*, 10...100 members)

1984,1994,2005,2012 *DAGM* Symposia in Österreich, 200 participants

1990 PRIP @ TU Wien: Pattern Recognition and Image Processing group

1994-2000 FWF FSP S70: Image Processing

1996 *13. ICPR* in Wien, 1000 participants

1992 ICG @ TU Graz: Computer Graphics and Vision

2000-2007 K-plus Zentrum ACV: Advanced Computer Vision

2004-2009 FWF NFN S91: Cognitive Vision

F.12 ICPR + K.S.Fu Award Gewinner 2010-2000

(a) H. Bunke

(b) A. K. Jain

(c) J. Kittler

(d) J. K. Aggarwal

(e) T. S. Huang

Abbildung F.6: ICPR und K.S.Fu Award Gewinner 2000 bis 2010

2010 Istanbul: Horst Bunke, Towards the Unification of Structural and Statistical Pattern Recognition

2008 Tampa: Anil K. Jain, Data Clustering: 50 Years Beyond K Means

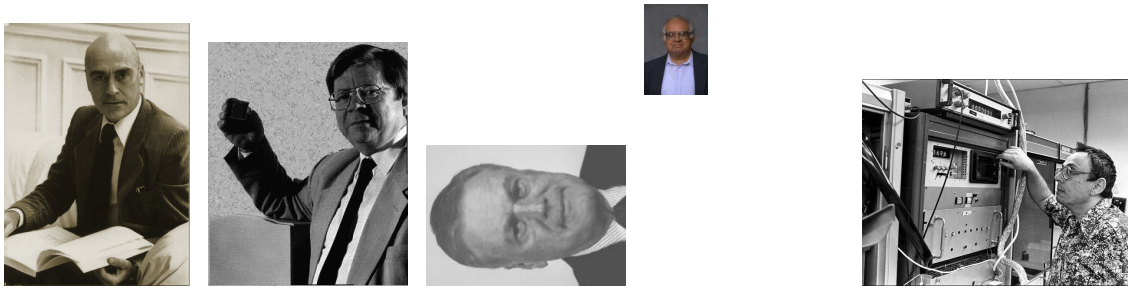
2006 Hong Kong: Josef Kittler, On Context, Modelling, Dimensionality ...

2004 Cambridge: J. K. Aggarwal, Structure and Motion from Image Sequences

2002 Quebec City: Thomas S. Huang, 3D Motion

2000 Barcelona: Theo Pavlidis, Structural Pattern Recognition
Vortrag verfügbar auf: http://www.theopavlidis.com/technology/KSFuLecture.htm

F.13 ICPR + K.S.Fu Award Gewinner 1998-1988



(a) J. C. Simon (b) T. Kohonen (c) H. Freeman (d) Rangasami L. Kashyap (e) A. Rosenfeld

Abbildung F.7: ICPR und K.S. Fu Award Gewinner von 1988 bis 1998

1998 Brisbane: Jean-Claude Simon, Automated Recognition of Handwritten Words

1996 Vienna: Teuvo Kohonen, Self-Organizing Map

1994 Jerusalem: Herbert Freeman, Chain Code; Automated Cartographic Text Placement

1992 The Hague: Levin Kanal

1990 Atlantic City: Rangasami L. Kashyap

1988 Rome: Azriel Rosenfeld, Computer Image Analysis, Digital Geometry and Topology

F.14 Meilensteine der Bildverarbeitung

Azriel Rosenfeld verfasste 1998 für den IAPR Newsletter 22 (1) einen technischen Report (CAR-TR-892) [23] über die Meilensteine der Bildverarbeitung. Folgender Abschnitt fasst diesen zusammen.

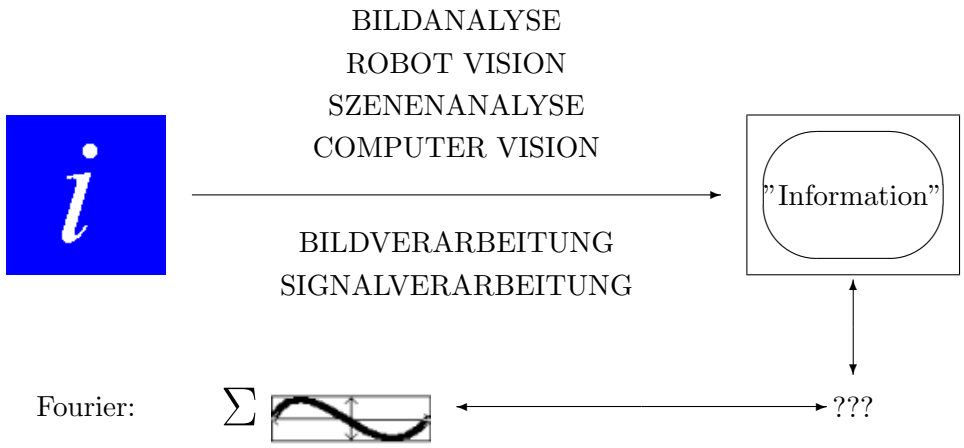
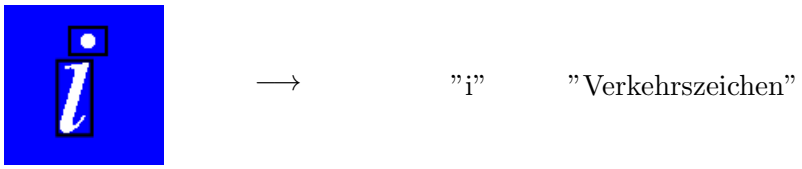


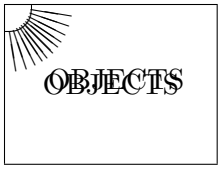
Abbildung F.8: Meilensteine der Bildverarbeitung 1950 - 2000

F.14.1 Bildanalyse



unterscheide	messe	klassifiziere	beschreibe
	Eigenschaften		
Bildteile		Regionen	Bild
	Relationen		

F.14.2 Robot Vision / Szenenanalyse / Computer Vision

3D SZENE	Projektion	
	←X→	image region

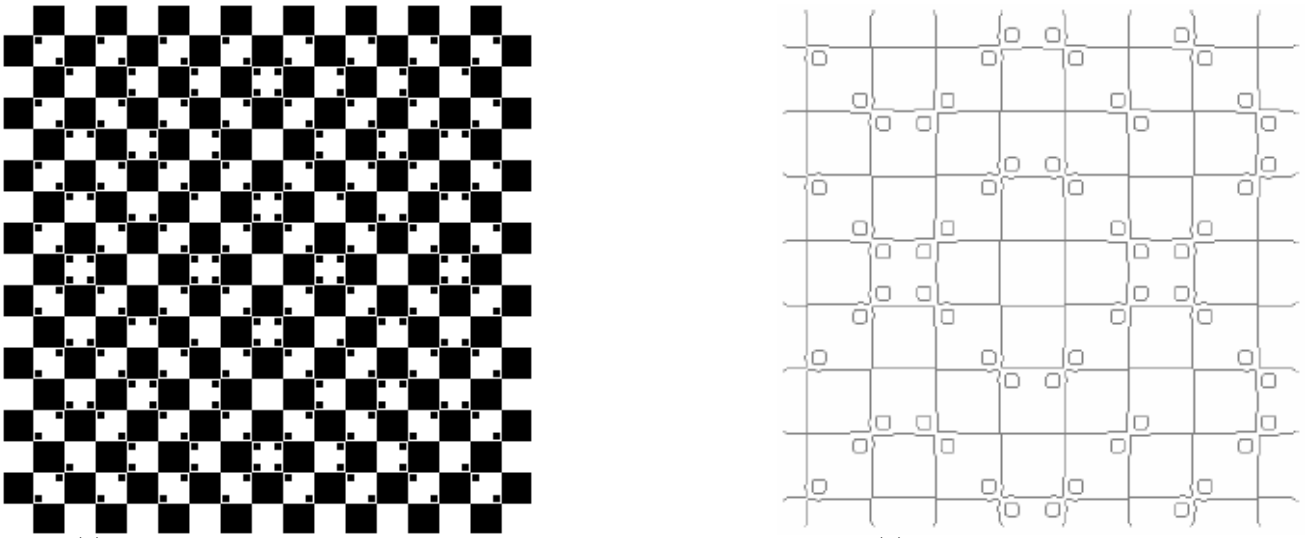
NEUE ZIELE:		VERDECKUNGEN
MANIPULATION		SCHATTEN
NAVIGATION		BELEUCHTUNG
HINDERNIS VERMEIDEN		VERZERRUNGEN
TIEFENBILD	⟵	STEREO
		TRIANGULATION
IDENTITÄT	⟵	MODELLE

F.14.3 Ausblick

Die meisten VISION PROBLEME	sind	MATHEMATISCH 'ILL DEFINED' FALSCH GESTELLT mit COMPUTER UNDURCHFÜHRBAR
Heutige SZENENMODELLE	sind	UNREALISTISCH
Heutige CV-SYSTEME	benutzen	unausgereifte MODELLE, *ad hoc* und *'brute force'* METHODEN
	arbeiten	in eingeschränkten ANWENDUNGEN
∃ CV LÖSUNGEN	⟸	Tier, Mensch
In der BIOLOGIE: REDUNDANZ	entdeckt	INKONSISTENZEN
	korrigiert	FEHLER
STEIGENDE RECHENLEISTUNG	⟹	MEHR REDUNDANZ
	⟹	GRÖSSERE LEISTUNG

F.15 Haben Computer geometrische optische Illusionen?

Uncertainty in Visual Processes Predicts Geometrical Optical Illusions
Cornelia Fermüller, Henrik Malm und Yiannis Aloimonos [7]



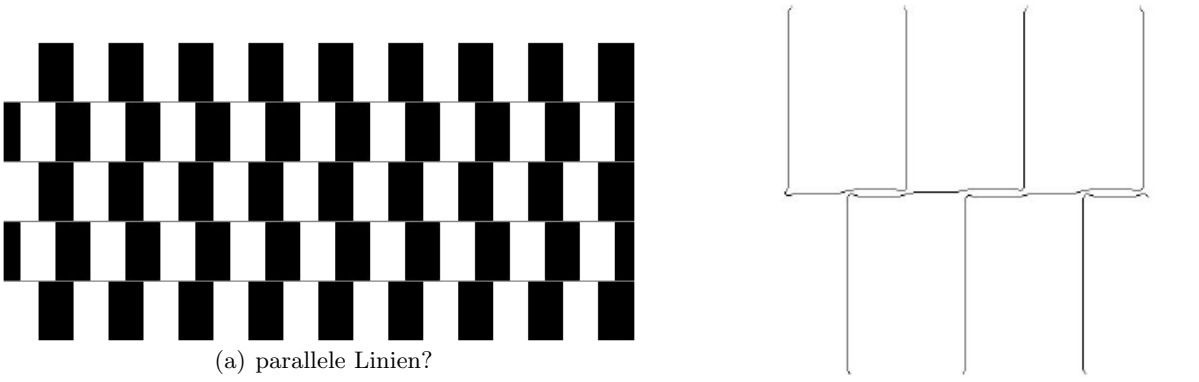
(a) Gerade oder gekrümmte Linien?

(b) Kanten im geglätteten Bild

Abbildung F.9: Optische Illusion

Die meisten Theorien, die optische Illusionen erklären, basieren auf Mechanismen des menschlichen Sehens, bzw. der Verarbeitung des Gesehenen im menschlichen Gehirn. Es ist aber auch möglich, optische Illusionen anhand mathematischer Methoden basierend auf Geometrie und Statistik zu erklären. Diese Theorien gelten somit gleichermaßen für alle Vision Systeme, egal ob biologischer oder künstlicher Natur. Diese Theorie erklärt auch die in Abbildung F.9 Bild F.9(b) sichtbaren leicht gekrümmten Linien.

F.15.1 Cafewall



(a) parallele Linien?

(b) Kanten im geglätteten Bild

Abbildung F.10: optische Illusion

Das in Abbildung F.10(a) dargestellte Muster beinhaltet ausschließlich parallele Geraden. Die einzelnen schwarzen und weißen Felder grenzen allerdings nicht ganz aneinander, zwischen diesen Feldern ist ein grau gefärbter Spalt vorhanden. Dieser ist bei benachbarten farblich unterschiedlichen Feldern weniger auffällig, da der graue Farbton zwischen weiß und schwarz liegt, bei gleich gefärbten benachbarten

Feldern ist er auffälliger. Dies lässt den Effekt gekrümmter Linien enstehen. Auch ein Kantendetektionsalgorithmus interpretiert die graue Linie abhängig von den Farben der angrenzenden Felder unterschiedlich (siehe Abbildung F.10(b), mehr darüber: http://www.cfar.umd.edu/~fer/optical/).

F.15.2 Rotierendes Rad?

Fixieren Sie die Mitte und verändern Sie Ihre Entfernung...

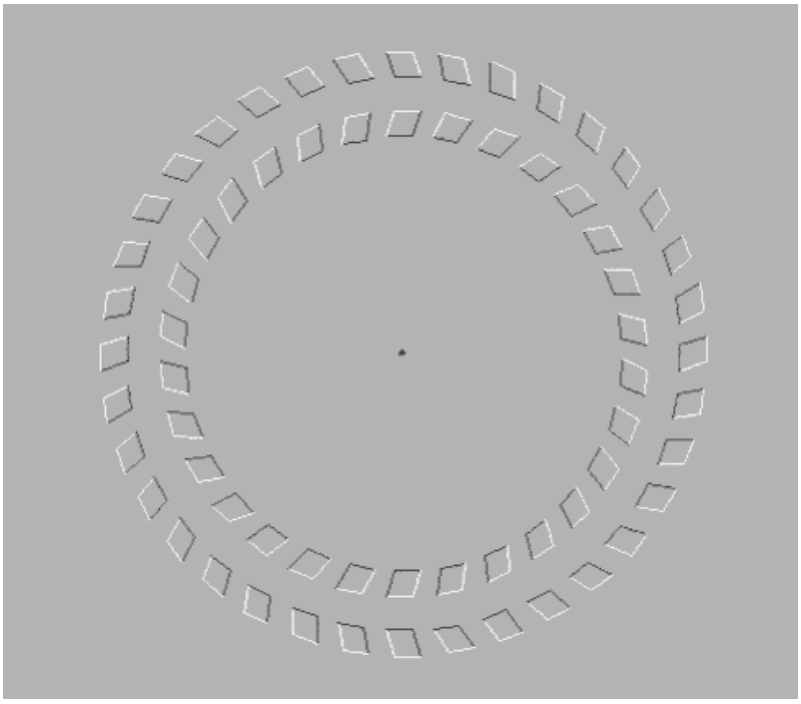
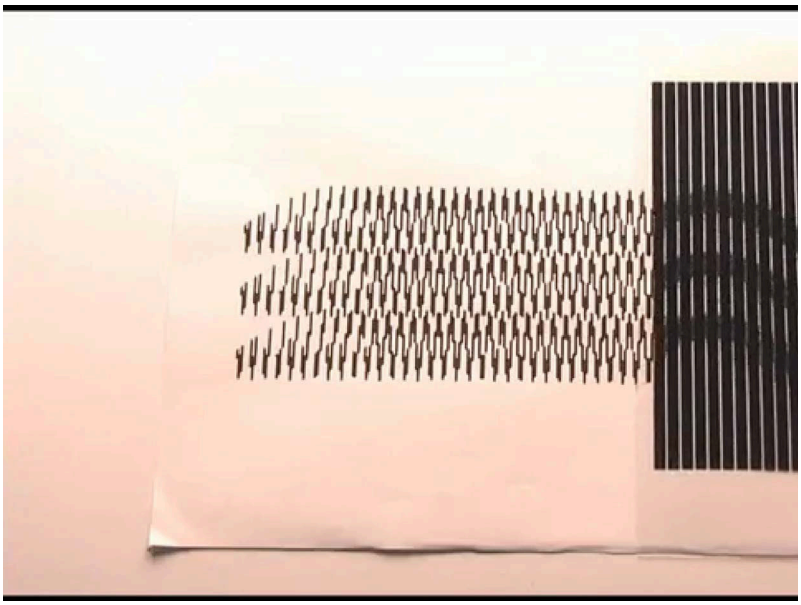


Abbildung F.11: optische Illusion

Diese Kreise bestehen aus kleinen, einzelnen, hellgrauen Elementen auf hellgrauem Hintergrund. Diese Elemente sind allerdings begrenzt von weißen und dunkelgrauen Linien. Die entgegengesetzte Bewegung der Kreise, die bei Veränderung der Distanz zwischen Betrachter und Bild für den Betrachter scheinbar sichtbar ist, entsteht auf Grund der Helligkeitsunterschied der Begrenzungslinien des äußeren und des inneren Kreises des Rades.

F.16 Amazing Animated Optical Illusions!



mehr darüber: http://www.youtube.com/watch?v=lvvcRdwNhGM

F.17 Zusammenfassung

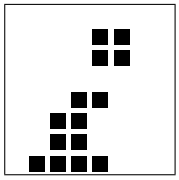
- (Bottom-up) Bildanalyse beginnt beim digitalen Bild und...
- ...leitet schrittweise eine Szenenbeschreibung her.
- Dabei werden viele MB an Daten auf 'wenige' Worte komprimiert.
- Den Dateneinheiten der verschiedenen Verarbeitungsstufen entsprechen Teile in der Realität.
- Wissen kann für die Top-Down Bildanalyse verwendet werden.
- Digitale Bildverarbeitung ist noch ein junges Fachgebiet,...
- ...dessen spezielle Probleme es von den meisten anderen Aufgaben der Informatik unterscheiden.

Kapitel G

Binäre Bilder, Math.Morphologie

G.1 Regionenrepräsentation

Binärbild:

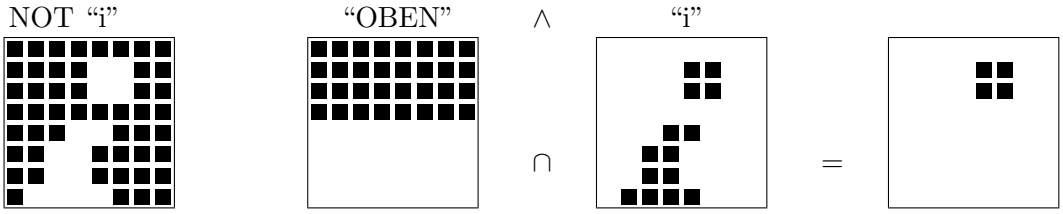


- Bildpunkt $g(x,y) = \begin{cases} 0 \\ 1 \end{cases}$
- Speicherbedarf: 1 bit/Pixel
- Binärbilder erhält man als Ergebnis von Threshold oder Region Growing

G.2 Masken: logische Operationen, Mengenoperationen

- Maske zur Darstellung einer Region R: $g(x,y) = \begin{cases} 1 & \text{wenn } (x,y) \in R \\ 0 & \text{sonst} \end{cases}$
- Maskieren mit Binärmaske: punktweise Multiplikation mit Bild.
- "punktweise":

Logische Operationen		entsprechen	Mengenoperationen:
NOT	$\neg R$	Komplement	$\bar{R}$
AND	$A \wedge B$	Durchschnitt	$A \cap B$
OR	$A \vee B$	Vereinigung	$A \cup B$



G.3 Zusammenhang, Weg, einfach zusammenhängend

- "lokal": **diskrete/digitale Geometrie**
- Nachbar (Γ_4  , Γ_8 )
- Weg = Folge von benachbarten Pixeln 
- eine Region R hängt zusammen, wenn alle Pixelpaare in R durch einen Weg verbunden sind.
- Eine nicht zusammenhängende Region zerfällt in Zusammenhangskomponenten, das sind Teilregionen, die maximal sind und zusammenhängen.
- Ein Loch(Insel) L einer Region R ist eine Zusammenhangskomponente des Komplements $\bar{R}$ der Region, die nicht Hintergrund ist.
- Eine zusammenhängende Region ohne Loch heißt "einfach zusammenhängend".

G.4 Zusammenhang: 4 / 8 Paradoxa

Die Wahl der Nachbarschaft $N \in \{4, 8\}$ spielt eine wichtige Rolle für einen N-Weg (N-Kurve) und für die Zusammenhangskomponenten $Z_N(R)$ einer Region R.

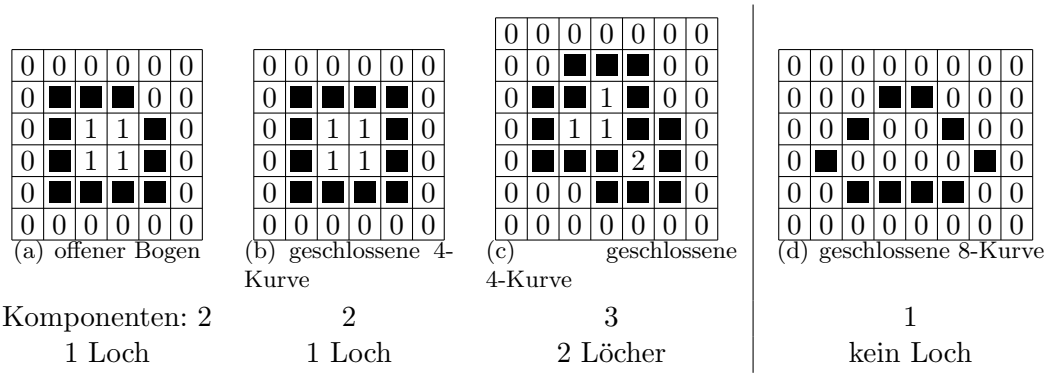


Abbildung G.1: Zusammenhang

Abbildung G.1(d) zeigt eine 8-Kurve basierend auf einer 8-verbundenen Fläche. Dadurch entsteht kein Loch, weil auch die innerhalb der Kurve liegenden Pixel mit den außerhalb liegenden auf Grund der 8-Nachbarschaft nach wie vor verbunden sind. Möchte man hier mit einem einfachen Algorithmus alle Punkte außerhalb der Kurve einfärben, so würden auch die Pixel innerhlab der Kurve gefärbt. Durch eine 4-Kurve auf einer 4-verbundenen Fläche kann diese Fläche in mehrere Zusammenhangskomponenten zerfallen - siehe Abbildung G.1(c). Die innerhalb der Kurve liegenden Pixel zerfallen in zwei Zusammenhangskomponeten. Möchte man in diesem Fall, wie zuvor erwähnt, die Pixel innerhalb der Kurve färben, so ist dies nicht möglich, weil nur Pixel, die zur gleichen Zusammenhangskomponente gehören, gefärbt werden.

G.5 Diskretes JORDAN'sches Kurventheorem

- Jede einfach geschlossene Kurve teilt die Ebene in 2 zusammenhängende Regionen. (JORDAN'sches Kurventheorem im Kontinuierlichen)
- Stimmt nicht immer im diskreten Raum
  - z.B. für $Z_4(\overline{\text{4-Kurve}})$ oder $Z_8(\overline{\text{8-Kurve}})$
  - Siehe dazu zB.: Abbildung G.1(c): hier teilt eine 4-zusammenhängende Kurve eine 4-zusammenhängende Fläche, dadurch entstehen nicht ein sondern zwei Löcher.
- Aber:
  Ist die Kurve 4-zusammenhängenden und die Fläche 8-zusammenhängend, bzw. umgekehrt, trifft das Jordan'sche Kurventheorem auch im diskreten Raum immer zu.

$$Z_8(\overline{\text{4-Kurve}}) = 2$$
$$Z_4(\overline{\text{8-Kurve}}) = 2$$

Das besprochene Problem tritt nicht auf, wenn das Bild *wohlgeformt* ist [16]:

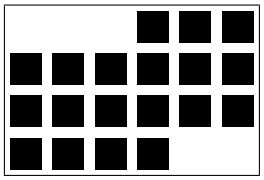
Das heißt, keine dieser beiden Formen:  oder tritt auf.

G.6 Rand und Inneres einer Region

- Der **Rand** einer Region R kann wie folgt definiert werden:

  1. $\text{Rand}_N(R) = R \cap \Gamma_N(\bar{R}), N \in \{4, 8\}$
  2. $P \in 4 - \text{Rand} \quad \begin{array}{|c|c|c|}\hline & N & \\ \hline W & \mathbf{P} & O \\ \hline & S & \\ \hline \end{array} \Leftrightarrow P \wedge (\neg N \vee \neg O \vee \neg S \vee \neg W)$

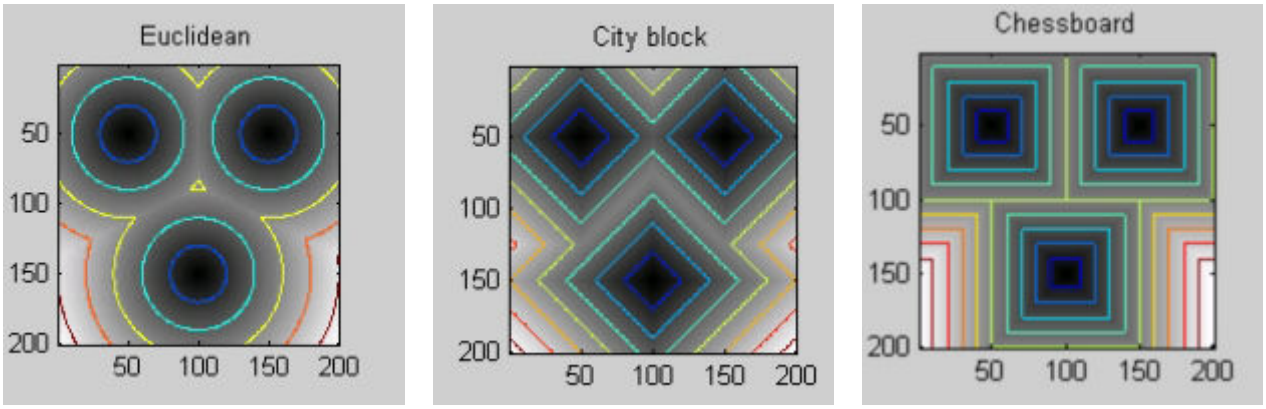
- Das **Innere** einer Region, $\textbf{Inneres}(R) = R \cap \overline{\text{Rand}(R)}$



G.7 Distanz: Euklid, City-block, Chessboard

- Bestimme die Distanz $d(P, Q)$ zwischen 2 Punkten $P = \begin{pmatrix} x \\ y \end{pmatrix}$ und $Q = \begin{pmatrix} u \\ v \end{pmatrix}$.

EUCLID	$d_e(P,Q) = \sqrt{(x-u)^2 + (y-v)^2}$
CITY BLOCK	$d_4(P,Q) = \lvert x-u \rvert + \lvert y-v \rvert$
Schachbrett	$d_8(P,Q) = \max(\lvert x-u \rvert, \lvert y-v \rvert)$



(a) Euklidische Distanz (b) City Block Distanz (c) Schachbrett Distanz

Abbildung G.2: Vergleich unterschiedlicher Distanzen

- $d(P, Q)$ ist eine **Metrik**:

1.	$d(P,Q) \geq 0$	positiv definite
2.	$d(P,Q) = 0 \Leftrightarrow P = Q$	
3.	$d(P,Q) = d(Q,P)$	symmetrisch
4.	$d(P,R) \leq d(P,Q) + d(Q,R)$	Dreiecksungleichung

- $d(P, X) \leq t$ bestimmt eine Form:
  vergleiche Abbildung G.2

G.8 Distanztransformation, paralleler Algorithmus

- Bestimme den Abstand $d_R(p)$ jedes Pixels p einer Region R von ihrem Rand!
- parallele Berechnung:
  1. initialisiere $d_R(p) = \begin{cases} \infty & p \in R \quad (Foreground) \\ 0 & p \notin R \quad (Background) \end{cases}$
  2. iteriere $d_R(p) = \min\{d_R(q), d(p,q) | q \in \Gamma(p)\}$

  Wobei d einer der in G.7 besprochenen Distanzfunktionen entspricht.
- Komplexität:

Speicher	$\mathcal{O}$(Bildgrösse)
Prozessoren	$\mathcal{O}$(Bildgrösse)
Iterationen	$\mathcal{O}$(Dicke(R))

G.9 Distanztransformation, sequentieller Algorithmus

- Bestimme den Abstand $d_R(p)$ jedes Pixels p einer Region R von ihrem Rand!
- sequentielle Berechnung (für d_4):
  1. initialisiere $d_R(p) = \begin{cases} \infty & p \in R \\ 0 & p \notin R \end{cases}$
  2. Vorwärtsdurchlauf : $d_R(p) = \min \left\{ \begin{array}{|c|c|} \hline & d_R(a) \\ \hline d_R(b) & \mathbf{d_R(p)} \\ \hline \end{array} + \begin{array}{|c|c|} \hline & 1 \\ \hline 1 & 0 \\ \hline \end{array} \right\}$
  3. Rückwärtsdurchlauf  : $d_R(p) = \min \left\{ \begin{array}{|c|c|} \hline d_R(p) & d_R(a) \\ \hline d_R(b) & \\ \hline \end{array} + \begin{array}{|c|c|} \hline 0 & 1 \\ \hline 1 & \\ \hline \end{array} \right\}$

G.10 Distanztransformation: Initialisierung

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	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												0
0												0
0												0
0	0	0	0	0	0	0	0	0	0	0	0	0

Startwerte: $0, \infty$ (leere Kästchen)

G.11 Vorwärtsdurchlauf und Rückwärtsdurchlauf

0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	2	2	2	2	2	2	2	2	2	2	0
0	1	2	3	3	3	3	3	3	3	3	3	0
0	1	2	3	0	0	0	0	0	0	0	0	0
0	1	2	3	0	1	1	1	1	1	1	1	0
0	1	2	3	0	1	2	2	2	2	2	2	0
0	1	2	3	0	1	2	3	3	3	3	3	0
0	1	2	3	0	1	2	3	0	1	2	3	0
0	1	2	3	0	0	0	0	0	1	2	3	0
0	1	2	3	1	1	1	1	1	2	3	4	0
0	1	2	3	2	2	2	2	2	3	4	5	0
0	1	2	3	3	3	3	3	3	4	5	6	0
0	0	0	0	0	0	0	0	0	0	0	0	0

nach **Vorwärtsdurchlauf** startet **Rückwärtsdurchlauf**

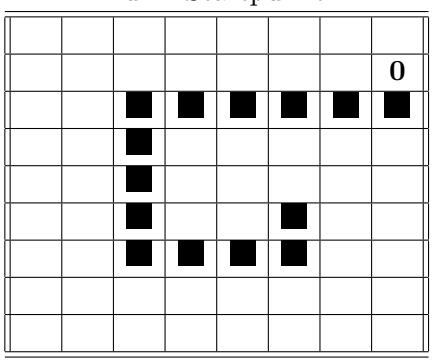
G.12 Vorwärtsdurchlauf und Rückwärtsdurchlauf

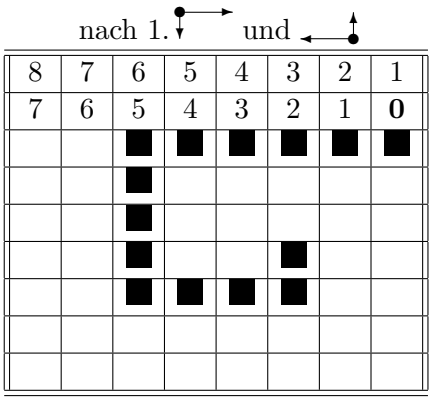
0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	2	2	2	2	2	2	2	2	2	**1**	0
0	1	2	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0
0	1	2	**1**	0	0	0	0	0	0	0	0	0
0	1	2	**1**	0	1	1	1	1	1	1	1	0
0	1	2	**1**	0	1	2	2	2	2	2	**1**	0
0	1	2	**1**	0	1	2	**2**	**1**	**2**	**2**	**1**	0
0	1	2	**1**	0	1	**1**	**1**	0	1	2	**1**	0
0	1	2	**1**	0	0	0	0	0	1	2	**1**	0
0	1	2	**2**	1	1	1	1	1	2	**2**	**1**	0
0	1	2	**2**	2	2	2	2	2	**2**	**2**	**1**	0
0	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Ergebnis nach Rückwärtsdurchlauf

G.13 Distanzen mit Hindernissen

Manchmal sollen Distanzen nur **innerhalb** einer Region bestimmt werden. In diesen Fällen stoppt die Ausbreitung der Distanzen an den Grenzen der Region:





nach 2. und 

8	7	6	5	4	3	2	1
7	6	5	4	3	2	1	**0**
8	7	■	■	■	■	■	■
9	8	■	24	23	22	21	22
10	9	■	23	22	21	20	21
11	10	■			■	19	20
12	11	■	■	■	■	18	19
13	12	13	14	15	16	17	18
14	13	14	15	16	17	18	19

nach Konvergenz

8	7	6	5	4	3	2	1
7	6	5	4	3	2	1	**0**
8	7	■	■	■	■	■	■
9	8	■	24	23	22	21	22
10	9	■	23	22	21	20	21
11	10	■	24	23	■	19	20
12	11	■	■	■	■	18	19
13	12	13	14	15	16	17	18
14	13	14	15	16	17	18	19

In diesen Fällen erfordert die sequentielle Durchführung mehrere Iterationen!

G.14 Distanztransformation: Masken und Komplexität

- Alternativmasken:

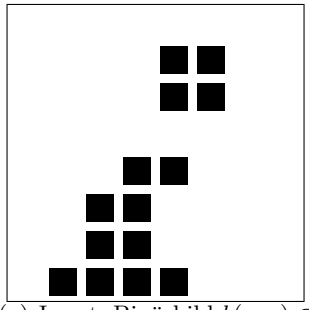
	d_4	d_8	d_{23}
vorwärts			
	1	1 1 1	3 2 3
	1 0	1 0	2 0
rückwärts			
	0 1	0 1	0 2
	1	1 1 1	3 2 3
	4-Metrik	8-Metrik	2/3-Metrik

- Komlexität:

Speicher	$\mathcal{O}$(Bildgrösse)
Prozessoren	1
Iterationen	$\mathcal{O}(2 \cdot$ Bildgrösse$)$

G.15 Zusammenhangskomponenten bestimmen

Engl. *Connected Component Labeling (CCL)*



(a) Input: Binärbild $b(x, y) \in \{0, 1\}$

0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	2	2	0	0	0
0	0	2	2	0	0	0	0
0	0	2	2	0	0	0	0
0	2	2	2	2	0	0	0

(b) Output: Labelbild $z(x, y) \in \{0, 1, \ldots m\}$

Abbildung G.3: Anwendung des CCL-Algorithmus

G.15.1 CCL-Algorithmus

1. Initialisierung:
   - $z(p) = b(p)$ (Initialsiere z(p) mit den Werten des zu behandelten Binärbilds),
   - setze Labelvariabel: $new = 2$,
   - Initialisier LUT des Labelings: $\text{LUT}(g) = g, g = 1, \ldots m$. m ist die Anzahl der bestimmten Komponenten nach Punkt 2 und ist daher erst nach vollständigem Durchlauf dieses Punktes bekannt. Die LUT wird während des Ablaufs des Algorithmus erweitert, wenn neue Komponenten dazukommen.
2. Vorwärtsdurchlauf ↓→:

IF				
		$*$	0	a
	z(p)	$*$ **0**	0 **1**	a **1** $\quad$ 0 **1** $\quad$ a **1**	b **1**
setze $z(p) =$		0	new	a	$\min(a, b)$
$new =$			$new + 1$		
$\text{LUT}(\max(a, b)) =$					$\min(a, b)$

Das Bild wird zeilenweise jeweils von links nach rechts und von oben nach unten durchlaufen. Das aktuelle Pixel wird dabei immer mit seinem linken Nachbarn und seinem oberen Nachbarn verglichen. Ist das aktuelle Pixel 0 (Pixel mit Wert 0 entsprechen Hintergrundpixel), so sind keine Vergleiche notwendig, da dieses auf 0 gesetzt bleib. Handelt es sich bei dem Pixel um ein Vordergrundpixel (Pixelwert 1) so wird entsprechend der obenstehenden Tabelle ein Labelwert gesetzt. Sind beide Nachbarn ungleich 0 aber unterschiedlichen Komponenten ($a \neq b$) zugeteilt, so bekommt das aktuelle Pixel das Label der Komponete mit niedrigerem Labelwert. In diesem Fall wird aber auch in der LUT für das größere Label der enstsprechende Wert auf den Labelwert des kleineren Labels gesetzt.

Vorläufige Komponten nach vollständigem Durchlauf von Punkt 2 angewandt auf Abbildung G.3(a):

0	0	0	0	0	0	0	0
0	0	0	0	2	2	0	0
0	0	0	0	2	2	0	0
0	0	0	0	0	0	0	0
0	0	0	3	3	0	0	0
0	0	4	3	0	0	0	0
0	0	4	3	0	0	0	0
0	5	4	3	3	0	0	0

Abbildung G.4: CCL Schritt 2

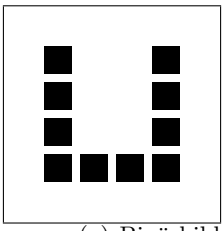
3. bestimme Äquivalenzklassen
   $Kl(g) = \min\{\text{LUT}(x) | x = g, \text{LUT}(g), \text{LUT}(\text{LUT}(g)), \ldots\} - 1$

Tabelle G.1: Äquivalenzklassen für Abbildung G.4

g	LUT(g)	$Kl(g)$
2	2	1
3	3	2
4	3	2
5	4	2

4. Scan Bild: $z(p) = Kl(z(p))$ für alle Pixel p.
   Ergebnis siehe Abbildung G.3(b)

G.15.2 CCL-Beispiel "U" und CCL_8



(a) Binärbild

0	0	0	0	0	0
0	2	0	0	3	0
0	2	0	0	3	0
0	2	0	0	3	0
0	2	2	2	2	0
0	0	0	0	0	0

(b) nach 1. Durchlauf

g	LUT(g)	$Kl(g)$
1	1	0
2	2	1
3	2	1

(c) LUT

0	0	0	0	0	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	1	1	1	0
0	0	0	0	0	0

(d) Resultat

Abbildung G.5: CCL-Beispiel "U"

CCL-Maske für Γ_8 :

NW	N	NO
W	1	

G.16 Beispiel: NURIKABE Regeln

Färben Sie die Felder des Diagramms hell oder dunkel nach folgenden Regeln:

- Ein Feld mit einer Zahl ist immer hell. Die Zahl in einem Feld gibt an, wie viele Felder einen hellen Bereich bilden. Alle Felder eines hellen Bereichs müssen waagrecht oder senkrecht miteinander verbunden sein. Zu jedem hellen Bereich gehört genau ein Feld mit einer Zahl.
- Alle hellen Bereiche müssen durch dunkle Felder voneinander getrennt sein. Helle Bereiche dürfen sich nicht berühren, weder horizontal noch vertikal (wohl aber diagonal).
- Alle dunklen Felder müssen einen einzigen dunklen Bereich bilden. Alle Felder des dunklen Bereichs müssen waagrecht oder senkrecht miteinander verbunden sein. Es gibt keinen dunklen Teilbereich der Größe 2×2.

	1					
2				3		3
			1			
				2		
4		3				2

G.16.1 NURIKABE, ein Binärbild

Lösung:

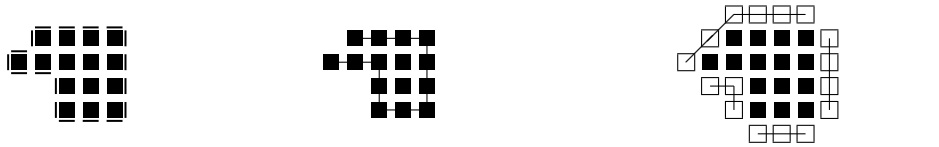
■	1	■	■	■	■	
■	■	■			■	
2		■	■	3	■	3
■	■	■	1	■	■	■
		■	■	2		■
	■	■		■	■	■
4	■	3		■		2

- Die Zahl eines Pixels gibt die Größe der weissen 4-Zusammenhangskomponente an, in der er liegt. Die Anzahl aller 4-Zusammenhangskomponenten entspricht genau der Anzahl an bezifferten Pixel.
- Es gibt genau eine schwarze 4-Zusammenhangskomponente.
- Es gibt keinen schwarzen Teilbereich der Größe 2×2.

G.17 Stereologie von Regionen

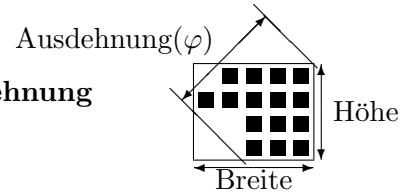
Stereologie bestimmt Maßzahlen fü r Regionen, die diese Regionen beschreiben und hat keinen Zusammenhang mit Stereoaufnahmen. Die meisten Maßzahlen sind invariant gegenüber Operationen wie (Rotation, Translation, etc.).

1. **Fläche(R)** = Anzahl Pixel von R
2. **Umfang**:



a) # cracks = 18 b) # $\text{Rand}_8(R) = 13$ c) # $\text{Rand}_4(\bar{R}) = 16$

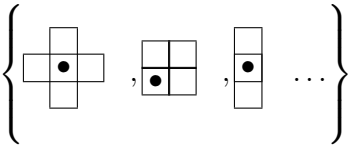
3. **Höhe, Breite, Ausdehnung**



4. **Durchmesser**$(R) = \max\{\text{Ausdehnung}(\varphi) | \varphi \in [0^0, 180^0)\}$
5. **Dicke**$(R) = \max\{d_R(p) | p \in R\}$
6. **Kompaktheit**$(R) = \frac{\text{Durchmesser}^2(R)}{\text{Fläche}(R)}$

G.18 Morphologische Filter

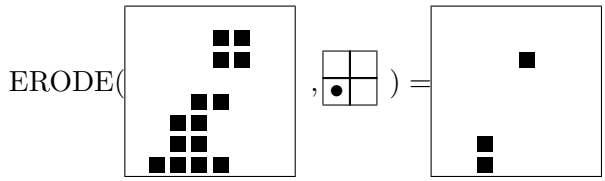
Morphologische Filter sind in der Lage die Struktur von Bildern gezielt zu beeinflussen und sind für Binärbilder gedacht(es gibt allerdings auch Erweiterung für Grauwert- und sogar Farbbilder). Es gibt zwei grundlegende morphologische Operationen „Schrumpfen“ (Erosion) und „Wachsen“ (Dilatation). Beide Operationen basieren auf Nachbarschaftsoperationen. Beim Einsatz morphologischer Filter werden Strukturelemente verwendet. Diese sind Matritzen und ähnlich zu Filterkernen, enthalten aber, so wie das zu bearbeitende Binärbild, nur die Werte 0 und 1. Wie bei einem Filterkern wird auch bei Strukturelementen ein Referenzpixel (=hot spot) festgelegt. Häufig verwendete Strukturelement in 2D sind: disk (Kreis), Rechteck, Quadrat, Linie, diamond (Raute).

Strukturelemente $S \in$ 

G.19 Mathematische Morphologie: Erosion

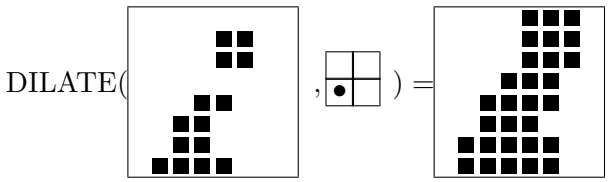
- Schrumpfen einer Region R: $R \setminus \text{Rand}(R)$
- gleichwertig mit der Frage:

  Paßt die Nachbarschaft  vollständig in R?
- als Mengenoperation: $ERODE(R,S) = R \ominus S = \{(x+i, y+j) \in R, \forall (i,j) \in S\}$
- EROSION: Paßt Strukturelement S vollständig in R, sonst entferne Referenzpixel aus R.



G.20 Mathematische Morphologie: Dilatation

- Ausdehnen einer Region R: $R \cup \text{Rand}(\bar{R})$
- gleichwertig mit: Ersetze jeden Pixel $p \in R$ durch seine Nachbarschaft 
- als Mengenoperation: $DILATE(R,S) = R \oplus S = \{(x,y) = (u+i, v+j) | (u,v) \in R, (i,j) \in S\}$
- DILATATION: Ersetze $p \in R$ durch $S(p)$.

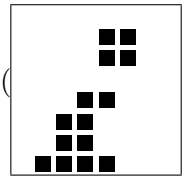
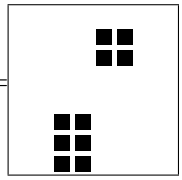


G.21 Eigenschaften von Dilatation und Erosion

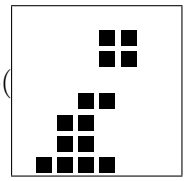
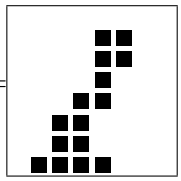
Dilatation und Erosion sind nicht zu einander inverse Operationen (man kann im Allgemeinen die Auswirkung einer Erosion nicht durch Anwenden einer Dilatation rückgängig machen und umgekehrt). Aber die beiden sind dual: eine Dilatation des Vordergrundes kann durch eine Erosion des Hintergrundes ersetzt werden. Genauso umgekehrt.
Die Dilatation ist kommutativ und assoziativ. Die Erosion ist allerdings nicht kommutativ.

- nicht invers: $(R \ominus S) \oplus S \neq R$
- dual: $\overline{(R \ominus S)} = \overline{R} \oplus \overline{S}$
- Erosion verkleinert: $(R \ominus S) \subset R$, Dilatation vergrössert: $R \subset R \oplus S)$
- kommutativ: $(R \oplus S) = (S \oplus R)$ aber: $(R \ominus S) \neq (S \ominus R)$
- assoziativ: $(R \oplus S) \oplus S' = R \oplus (S \oplus S')$ aber: $(R \ominus S) \ominus S' = R \ominus (S \oplus S')$
- Erosion: 1 Strukturelement erzeugt 1 Pixel
- Dilatation: 1 Pixel setzt alle Pixel des Strukturelements.

G.22 Mathematische Morphologie: Öffnen/Opening

- Opening: Erosion gefolgt von einer Dilatation (durchgeführt mit dem selben Strukturelement)
- Öffnen einer Region R: $OPEN(R, S) = DILATE(ERODE(R, S), S) = R \circ S$
- OPEN( , ) = 
- Eigenschaften (mit scheibenförmigem Strukturelement):
  glättet Kontur,
  unterbricht Engstellen,
  eliminiert kleine Regionen (Inseln) und Halbinseln,
  die kleiner sind als das verwendete Strukturelement.
  idempotent: $(R \circ S) \circ S = R \circ S$

G.23 Mathematische Morphologie: Schliessen/Closing

- Closing: Dilatation gefolgt von einer Erosion (durchgeführt mit dem selben Strukturelement)
- Schliessen einer Region R:
  $CLOSE(R, S) = ERODE(DILATE(R, S), S) = R \bullet S$
- CLOSE( , ) = 

- Eigenschaften (mit scheibenförmigem Strukturelement)
  glättet Kontur,
  schließt Engstellen und schmale Einbuchtungen,
  eliminiert kleine Löcher und
  schließt kleine Löcher der Kontur,
  die kleiner sind als das verwendete Strukturelement.
  idempotent: $(R \bullet S) \bullet S = R \bullet S$

G.24 Verdünnen/Thinning, einfacher Punkt

- Kurve = 'längliche' Region
- Verdünnen soll Kurve auf 1 Pixel Breite schrumpfen, **ohne ihre Länge zu reduzieren** und **ohne ihren Zusammenhang zu zerstören**.
- Wird ein "einfacher Randpunkt" weggenommen, so bleibt der Zusammenhang der Region(en) erhalten:

(a) (b) (c)

Abbildung G.6: unterschiedliche Kurven bzw. Zusammenhangskomponenten

Die nachfolgende Tabelle zeigt die Anzahl der Zusammenhangskomponenten der in Abbildung G.6 sichtbaren Fälle für 4- bzw- 8-Zusammenhangskomponenten:

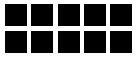
Grafik	G.6(a)	G.6(b)	G.6(c)
$Z_4(\blacksquare \cup \mathbb{P})$	2	1	1
$Z_4(\blacksquare)$	2	2	2
$\mathbb{P}$	4-einfach	nicht 4-einfach	
$Z_8(\blacksquare \cup \mathbb{P})$	1	1	1
$Z_8(\blacksquare)$	2	2	1
$\mathbb{P}$	nicht 8-einfach		8-einfach

Ist $\mathbb{P}$ 4- bzw. 8-einfach, so kann dieses Pixel für eine 4- bzw. 8-zusammenhängende Kurve gelöscht werden, ohne dass diese Kurve zerfällt oder verkürzt wird, da es dann einen einfachen Randpunkt darstellt.

- Ein Endpunkt hat nur 1 Nachbarn in R.

G.24.1 Verdünnen/Thinning Algorithmus

- Der in Punkt G.24 besprochene Ansatz bringt allerdings folgendes Problem bei gleichzeitiger Prüfung aller Bedingungen vor dem Löschen einfacher Punkte:

  R =  verschwindet vollständig!

- Daher wird folgender paralleler/sequentieller Algorithmus verwendet:

  Für jeden Punkt P: $\Gamma(P) =$ 

  1. lösche alle einfachen N-Randpunkte, die nicht Endpunkte sind.
  2. lösche alle einfachen O-Randpunkte, die nicht Endpunkte sind.
  3. lösche alle einfachen S-Randpunkte, die nicht Endpunkte sind.
  4. lösche alle einfachen W-Randpunkte, die nicht Endpunkte sind.
  5. wiederhole 1.-4. bis keine Änderung mehr autritt

Das Ergebnis der Anwendung dieses Algorithmus auf eine Kurve ist in Abbildung G.7 zu sehen. Die mit 1 markierten Pixel werden nach durchlaufen von Schritt 1 des Algorithmus gelöscht, die mit 2 markierten Pixel werden nach Schritt 2 gelöscht, usw. Die schwarz markierten Pixel zeigen das Endresultat, nach Anwendung des Algorithmus, sie stellen die auf 1 Pixel Breite verdünnte Kurve dar.

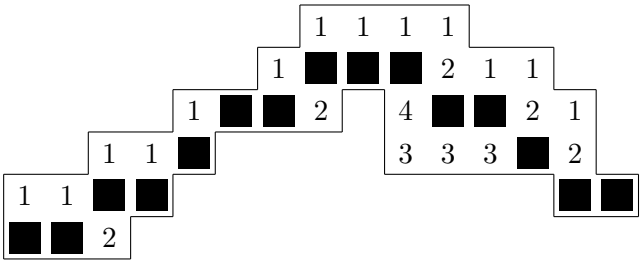


Abbildung G.7: Ergebnis nach Anwendung des parallelen Thinning-Algorithmus

G.25 Zusammenfassung

- Ein Binärbild kann Regionen darstellen.
- Ein solches Binärbild von Regionen kann daher als Maske dienen.
- Logische Operationen $(\wedge, \vee, \neg)$ verknüpfen Binärbilder.
- Die diskrete Topologie definiert Nachbarschaften: Γ_4, Γ_8, Wege, Löcher, Rand und das diskrete Jordan'sche Kurventheorem.
- Distanzen können auf verschiedene Weise definiert sein (Euklid, City-block, Chessboard)
- Die Algorithmen zur Bestimmung der Distanz und der Zusammenhangskomponenten sind ähnlich aufgebaut.
- Die Stereologie bestimmt Eigenschaften einer Region.
- Morphologische Operatoren (ERODE, DILATE, OPEN, CLOSE, THIN) verändern ein Binärbild.

Kapitel H

Bild- u. Objektdarstellung, Datenstrukturen

Exakte vs. Inexakte Darstellung

Die Bewertungskriterien für die Exaktheit einer Darstellung sind der Speicherbedarf und die Aussagekraft. Zu den **exakten** Darstellungen zählen **vollständige / verlustfreie (lossless)** Darstellungen, welche die Rekonstruktion der Originaldaten ermöglichen. Die **approximativen (lossy)** Darstellungen nähern die Daten an und reduzieren die Störungen. **3D** Darstellungen können Daten im dreidimensionalen Raum (3 Dimensionen) beschreiben. Unter dem Begriff **Bildmodelle** versteht man die Beschreibung ganzer Klassen von Bildern. In Tabelle H.1 werden Datenstrukturen für verschiedene Darstellungsmöglichkeiten angegeben.

Tabelle H.1: Datenstrukturen für verschiedene Darstellungen

Darstellung von	Datenstruktur
Bild	Matrix
Bildzeile	Ketten (Chain)
Bildblock	topologische Datenstruktur
Rand 2D	Zahlencodes, Symbolcodes
Rand 3D	Flächen
Bildregion	hierarchische Datenstruktur

H.1 Darstellung von Bildern

Die **Matrix** ist die häufigst verwendete Datenstrukutr für digitale Bilder. Ihr Vorteil ist der Direktzugriff auf Pixel durch Zeile und Spalte und hat einen Speicherbedarf von $n \times m \times b$, wobei n für die Anzahl der

Zeilen, m für die Anzahl der Spalten und b für die Bits pro Pixel steht.
Stapel von Matrizen können verwendet werden um **Multiresolutionspyramiden** zu formieren, welche im Kapitel I näher betrachtet werden.
Binärmatrizen werden zur Darstellung und Verarbeitung von Binärbildern und Regionen verwendet, welche nur 1 Bit zur Farbcodierung eines Pixels benötigen.
3D Matrizen B(x, y, z) können entweder eine räumliche Anordnungen von Volumselementen (Voxel) enthalten (z.B. Anwendung in der Computertomographie (CT) oder Kernspinresonanzspektroskopie (NMR)), oder die Zeit als dritte Dimension verwenden, d.h. Darstellung von Bildsequenzen, Videos.

H.2 Darstellung von Bildzeilen

Die Darstellung von Bildzeilen kann durch eine Lauflängencodierung erfolgen. Als Beispiel dafür sind der Run Length Code und der Binärbaum in den nächsten Abschnitten erklärt.

H.2.1 Lauflängencodierung mittels Run Length Code

Als Lauf bezeichnet man ein wiederholtes Auftreten desselben Wertes (z.B. eines Pixel). Um eine kompakte Darstellung eines Bildes zu erlangen, kann man die sogenannte Lauflängenodierung (engl.: Run Length Code) anwenden, welche die Anzahl der Wiederholungen eines Pixels desselben Wertes und den Wert selbst codiert. Hiermit kann die Anzahl der benötigten Zahlen für die kompakte Darstellung eines Bildes minimiert werden. Im folgenden Beispiel wird dieses Verfahren veranschaulicht:

Beispiel Run Length Code eines Grauwertbildes:

- Der Anfang von Zeile 50 der Abbildung H.1 hat folgende 49 Werte:
  5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 215, 215, 215, 215, 215, 215, 215, 5, 5, 5, 5, 5, 5, 5, 5, 5, 0, 215, 215, 215, 215, 215, 215, 215, 215, 215, 215, 215, 215 (49 Zahlen)
- Die kompakte Darstellung mittels Run Length Code ergibt nur mehr 10 Werte:
  $19 \times 5, 7 \times 215, 9 \times 5, 1 \times 0, 13 \times 215$ (10 Zahlen)

Abbildung H.1: Grauwertbild des Buchstaben “i“

Bei Binärbildern wechselt der Code bei jedem Lauf, daher kann nach Definition des Wertes der Pixel des ersten Laufes, dieser Wert weggelassen werden, welches die Anzahl der benötigten Zahlen für die kompakte Darstellung eines Binärbildes reduziert.

Beispiel Vereinfachung des Run Length Code bei Binärbildern:
$19 \times 0, 7 \times 1, 10 \times 0, 13 \times 1, \ldots$ (8 Zahlen) $= 19, 7, 10, 13, \ldots$ (4 Zahlen)

H.2.2 Lauflängencodierung mittels Binärbaum

Bei der Lauflängencodierung mittels Binärbaum werden die Läufe der Länge 2^n jeweils zusammengefasst, um eine kompakte Darstellung eines Bildes zu erhalten. Die Anzahl der Blätter des Binärbaumes entspricht der Anzahl der Pixel in einem Bild. Der Wert der Pixel eines Laufes wird auf der Ebene n der

zugehörigen zweier Potenz 2^n gespeichert, wobei n = 0 die Ebene der Blätter darstellt (siehe Abbildung H.2)

- Zusammenfassen von Läufen der Länge 2^n:
  [5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,] [5, 5,] [5,] [215,] [215, 215, 215, 215,] [215, 215,] [5, 5,]
  [5, 5, 5, 5,] [5, 5,] [5,] [0,] [215, 215, 215, 215,] [215, 215, 215, 215, 215, 215, 215, 215,]
- Binärbaum:

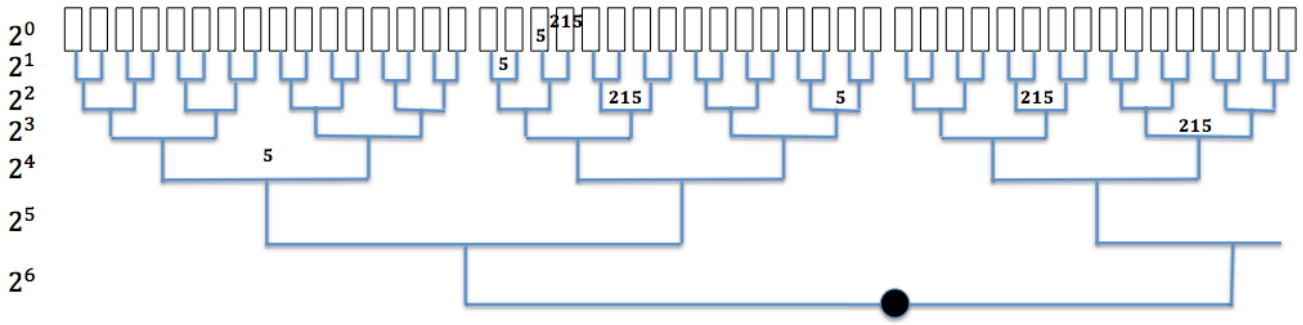


Abbildung H.2: Binärbaum einer Lauflängencodierung

- Darstellung als Klammerausdruck (z.B. LISP):
  $((5,(((5,(5,215)),215),((215,5),5))),((((5,(5,0)),215),215),\ldots$

H.3 Darstellung von Bildblöcken in 2D

H.3.1 Medialachse, Symmetrieachse

Ein **Block** in einem Bild kann als ein quadratisches Fenster mit gleichen Pixelwerten (vgl. Lauf) angesehen werden und wird durch einen Referenzpunkt (z.B. Zentrum), seine Größe und seinen Wert beschrieben. Eine **Region** stellt eine Vereinigung von Blöcken dar. **Maximale Blöcke** werden angestrebt, d.h. das Bild wird in größtmögliche Blöcke gleicher Pixelwerte unterteilt (siehe Abbildung H.3). Die Medialachse bzw. Symmetrieachse stellt die Zentren maximal eingeschriebener Blöcke/ Kreise dar (siehe Abbildung H.4)

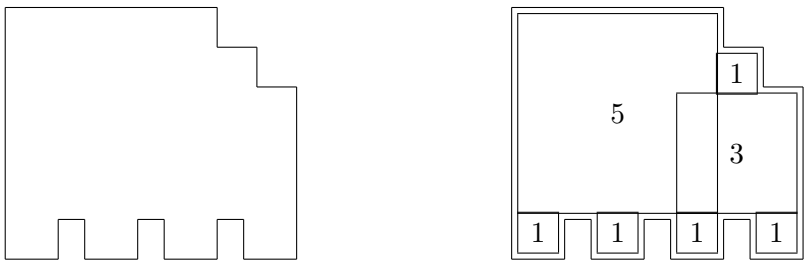


Abbildung H.3: Bildung maximaler Blöcke

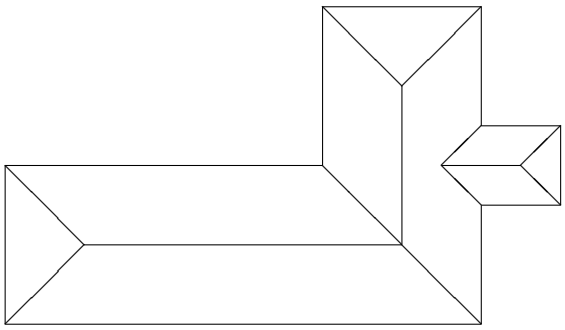


Abbildung H.4: Medialachse, Symmetrieachse

H.3.2 Quadtree

Die Darstellung von Bildblöcken kann auch in der Form eines Quadtree's erfolgen. Hierzu wird das zweidimensionale Bild in Quadranten zerlegt. Jeder Quadrant wird wiederum in neue Quadranten zerlegt, solange bis jeder Quadrant einen Block bildet, d.h. ein quadratisches Fenster mit gleichen Pixelwerten (siehe Abbildung H.5). Das Ergebnis der Zerlegung kann in einem Quadtree abgespeichert werden. (siehe Abbildung H.6).
Die Blockgröße beträgt bei insgesamt n Ebenen pro Ebene: $2^0 \times 2^0, 2^1 \times 2^1, \ldots, 2^n \times 2^n$.

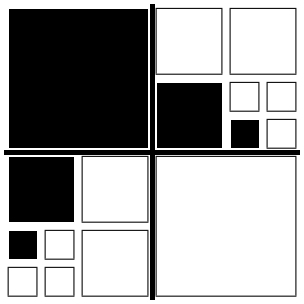


Abbildung H.5: Zerlegung des Bildes in homogene Quadtranten

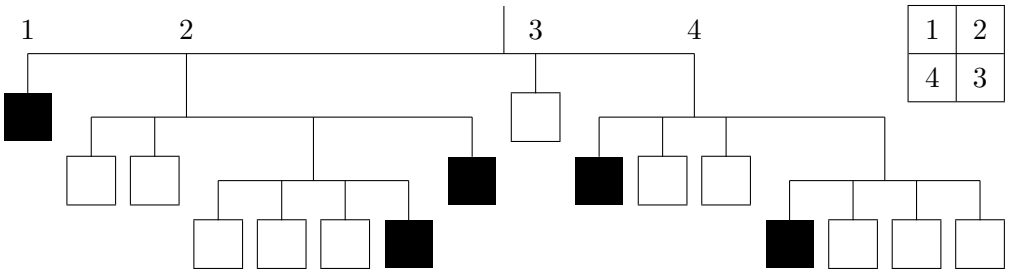


Abbildung H.6: Quadtree

H.4 Darstellung von Bildblöcken in 3D

H.4.1 Octree

Die Darstellung von Bildblöcken im dreidimensionalen Raum kann in der Form eines Octree's erfolgen. Hierzu wird das Bildvolumen W in jeweils 8 Würfel zerlegt:

$W \to W_{+++}W_{++-}W_{+--}W_{+-+}W_{-++}W_{-+-}W_{---}W_{--+}$ (siehe Abbildung H.7)
Dies erfolgt solange, bis ein Würfel homogen ist, d.h. gleiche Voxelwerte enthält.

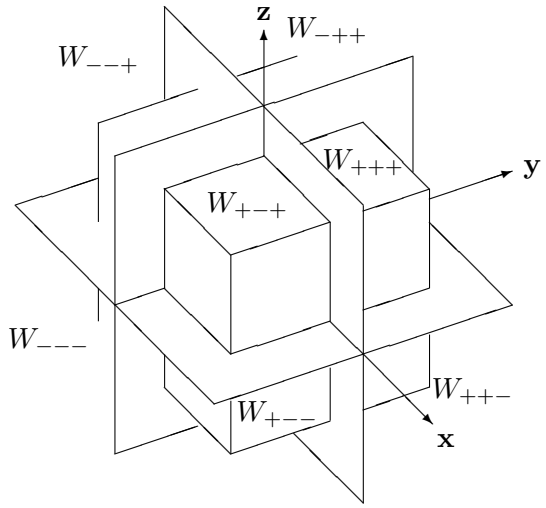


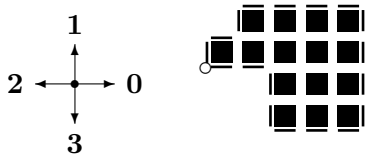
Abbildung H.7: Octree

H.5 Darstellung von Rändern in 2D

Als Rand bezeichnet man einen Teil einer Objektgrenze. In den folgenden vorgestellten Randcodierungen beschreiben Symbole (z.B. beim RULI chain Code) oder Zahlen eines Codes (z.B. beim Freeman Chain Code oder Crack Code) einen Einzelschritt.

H.5.1 Crack-Code

Beim Crackcode wird eine Iterierung entlang des Randes durch Zahlen codiert, indem jedem Schritt abhängig von der Richtung (N,S,W,O), eine unterschiedliche Zahl zugewiesen wird. In Abbildung H.8 ist die Funktionsweise des Crack-Codes dargestellt. Hier wurde z.B. der Nordrichtung die Zahl 1 zugewiesen, der Ostrichtung 0 usw.. Der Anfangspunkt für die Randcodierung wird hier mit einem Kreis dargestellt.

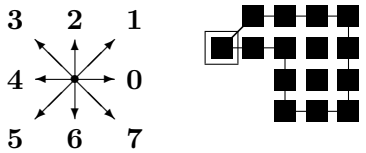


1, 0, 1, 0, 0, 0, 0, 3, 3, 3, 3, 2, 2, 2, 1, 1, 2, 2.

Abbildung H.8: Beispiel Crack-Code

H.5.2 Freeman Chain Code (1962)

Im Vergleich zum Crack-Code sind beim Freeman Chain Code zusätzlich Querrichtungen (NO, OS, SW, NW) bei der Iterierung entlang des Randes zugelassen. Somit erhält man 8 mögliche Richtungen, welche durch 3 Bits codiert werden. Siehe hierzu Abbildung H.9

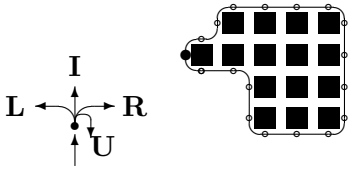


1, 0, 0, 0, 6, 6, 6, 4, 4, 2, 2, 4, 4

Abbildung H.9: Beispiel Freeman Chain Code

H.5.3 RULI chain code

Beim RULI Chain Code werden wie beim Crack-Code auch die 4 Himmelsrichtungen mit den Symbolen **R** (rechts), **U** (Umkehr) **L** (links) **I** (in die selbe Richtung) codiert, jedoch mit dem Unterschied, dass sich die Blickrichtung nach jedem Iterierungsschritt, abhängig von der Fortbewegungsrichtung, in diese neu orientiert (vgl. Autofahren; Abbildung H.10)



RLRIIIRIIIRIIIRILIR

Abbildung H.10: Beispiel RULI Chain Code

H.6 Darstellung von Rändern in 3D

H.6.1 Begrenzungsflächen

Ränder im dreidimensionalen Bereich werden als Begrenzungsflächen bezeichnet. Diese können in Form von **quadratischen Flächen** (siehe Abbildung H.11) oder **Dreiecksnetzen** (siehe Abbildung H.12) dargestellt werden.

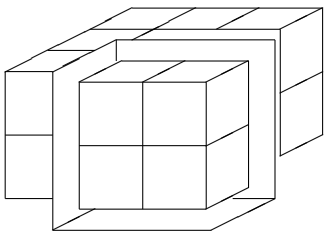


Abbildung H.11: Begerenzungsfläche quadratischer Form

Eine weiter Möglichkeit Begrenzungsflächen darzustellen, ist das **Verschieben eines allgemeinen Querprofils** (z.B. eines Kreises oder Ellipse) **entlang einer Achse** eines dreidimensionalen Objektes. Diese kann eine gerade oder gekrümmte Form haben. Auch die Form oder Größe des Querprofils (z.B. der Radius bei einem Kreis) kann während des Verschiebens verändert werden. In Abbildung H.13 ist die Darstellung der Begrenzungsfläche eines allgemeinen Kegels skizziert. Die Achse dieses dreidimensionalen Objektes ist gerade und als Querprofil wurde eine Ellipse verwendet. Zusätzlich wird nach jedem Verschiebungsschritt t die horizontale Ausdehnung (Hauptachse r) der Ellipse verändert, um die

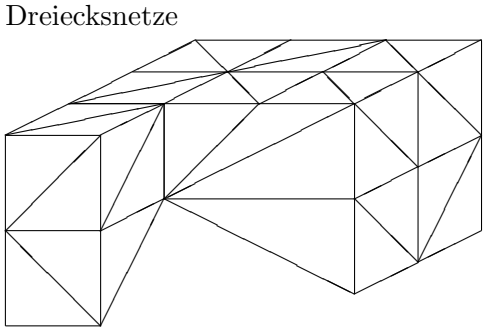


Abbildung H.12: Begrenzungsfläche bestehend aus einem Dreiecksnetz

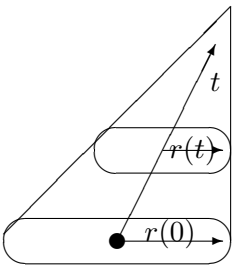


Abbildung H.13: Bestimmung der Begrenzungsfläche eines allgemeinen Kegels

Begrenzungsfläche des Kegels darzustellen.
In Abbildung H.14 ist die Darstellung der Begrenzungsfläche eines allgemeinen gebogenen Zylinders skizziert. Dieser besitzt eine gekrümmte Achse, entlang derer ein ellipsenförmiges Querprofil verschoben wird, dessen Form oder Größe sich nicht in den darauffolgenden Verschiebungsschritten ändert, um die Begrenzungsfläche des Zylinders darzustellen.

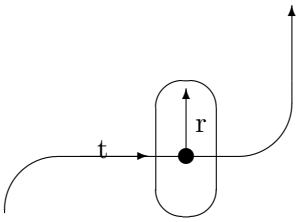


Abbildung H.14: Bestimmung der Begrenzungsfläche eines gekrümmten allgemeinen Zylinders

H.7 Darstellung von Bildregionen

H.7.1 Bildmodelle

Bildmodelle dienen zur Beschreibung einer ganzen Klasse von Bildern und zählen zum Vokabular der Bildanalyse. Zu diesem zählt auch der Begriff der Beschreibungsebenen welche sich in folgende gliedern:

- Ebene A: Arrays, Matrix: z.B. Originalbild, verarbeitete Bilder

- Ebene B: REGION, Geometrie und Form: z.B. RLC, MAT, maximale Blöcke

Tabelle H.2: Modelltypen von Bildmodellen

DEKLARATIV Menge von Bedingungen für Eigenschaften und Beziehungen auf einer Bildebene	PROZEDURAL Prozeß erkennt oder erzeugt Bilder, formale Grammatiken	
DETERMINISTISCH eindeutig Eingabe $\longrightarrow$ Ausgabe	PROBABILISTISCH Verteilungen	FUZZY Zugehörigkeitsmaß

- Ebene C: Beziehungen, Relationen: z.B. Graph G(V,E)

Bei den Bildmodellen kann zwischen 5 Modelltypen unterschieden werden. Siehe hierzu Tabelle H.2

Deklarativer Modelltyp

Für jede Bildbeschreibungsebene gibt es deklarative Modelltypen, welche in Tabelle H.3 mit Beispielen aufgelistet sind.

Tabelle H.3: Deklarative Modeltypen der Bildbeschreibungsebenen A,B und C

Ebene	Bedingungen	Beispiele
A:	**Pixeleigenschaften**	hell, dunkel, rot, in Intervall
B:	**Form**	Zusammenhang, Glattheit, Konvexität, Länglichkeit, Anstieg, Krümmung, ...
	zwischen **Regionen**	Überlagerung, Verdeckung, gemeinsame Grenze, Einschluß, ...
C:	**Relationenklassen**	part-whole, is-a, oberhalb, gehören zusammen: Gruppierung, grösser als, dunkler, ...

Prozeduraler Modelltyp

Zu dem prozeduralen Modelltyp zählen die (Array-)Grammatiken. Formale Grammatiken G werden durch ein 4er-Tupel beschrieben:

$\mathbf{G} = (V_T, V_N, S, P)$

- V_T enthält Grauwerte, die für die Terminalsymbole stehen
- V_N enthält die Non-Terminalsymbole
- S ist das Startsymbol
- P enthält die Produktionsregeln P = $\{P_1, ..., P_N\}$ =$\{L_1 \rightarrow R_1, ..., L_N \rightarrow R_N\}$ die angeben, wie die linke Seite L durch die rechte Seite R substituiert wird, d.h. wie das Bild erzeugt wird. Durch **Matchen** erhält man die linke Seite L, welche durch die rechte Seite R ersetzt wird.

Beispiel Arraygrammatik G_R

$G_R = (V_T, V_N, S, P)$

- $V_T = \{\#, a\}$

- $V_N = \{S, A, B, C, D\}$
- $S = \{S\}$
- P:

$P_1 = \boxed{S}\boxed{\#} \rightarrow \boxed{a}\boxed{S}$

$P_2 = S \rightarrow A|a$

$P_3 = \begin{array}{|c|}\hline A \\ \hline \# \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline a \\ \hline B \\ \hline \end{array}$

$P_4 = \begin{array}{|c|c|}\hline a & \\ \hline \# & B \\ \hline \end{array} \rightarrow \begin{array}{|c|c|}\hline a & \\ \hline B & a \\ \hline \end{array}$

$P_5 = \begin{array}{|c|c|}\hline \# & \\ \hline \# & B \\ \hline \end{array} \rightarrow \begin{array}{|c|c|}\hline \# & \\ \hline \# & a \\ \hline \end{array} \mid \begin{array}{|c|c|}\hline \# & \\ \hline \# & C \\ \hline \end{array}$

$P_6 = \begin{array}{|c|}\hline C \\ \hline \# \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline a \\ \hline D \\ \hline \end{array}$

$P_7 = \begin{array}{|c|c|}\hline & a \\ \hline D & \# \\ \hline \end{array} \rightarrow \begin{array}{|c|c|}\hline & a \\ \hline a & D \\ \hline \end{array}$

$P_8 = \begin{array}{|c|c|}\hline & \# \\ \hline D & \# \\ \hline \end{array} \rightarrow \begin{array}{|c|c|}\hline & \# \\ \hline a & \# \\ \hline \end{array} \mid \begin{array}{|c|c|}\hline & \# \\ \hline A & \# \\ \hline \end{array}$

Nun stellt sich die Frage welches Bild die Grammatik G_R erzeugt. Hierzu analysiert man die Produktionsregeln:

Analyse der Produktionsregeln P_1 und P_2 anhand eines Beispiels:

$P_1 \begin{array}{|c|c|}\hline S & \# \\ \hline \end{array} \rightarrow \begin{array}{|c|c|}\hline a & S \\ \hline \end{array}$, $P_2 = S \rightarrow A|a$

$$\begin{array}{|c|c|c|c|}\hline S & \# & \cdots & \# \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|}\hline a & \cdots & a & \mathbf{S} \\ \hline \end{array} \rightarrow \begin{cases} \begin{array}{|c|c|c|c|}\hline a & \cdots & a & \mathbf{a} \\ \hline \end{array} & S \rightarrow a \\ \begin{array}{|c|c|c|c|}\hline a & \cdots & a & \mathbf{A} \\ \hline \end{array} & S \rightarrow A \end{cases}$$

Analyse der Produktionsregeln P_3, P_4, P_5 anhand eines Beispiels:

$$P_3 = \begin{array}{|c|}\hline A \\ \hline \# \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline a \\ \hline B \\ \hline \end{array}, P_4 = \begin{array}{|c|c|}\hline a & \\ \hline \# & B \\ \hline \end{array} \rightarrow \begin{array}{|c|c|}\hline a & \\ \hline B & a \\ \hline \end{array}, P_5 = \begin{array}{|c|c|}\hline \# & \\ \hline \# & B \\ \hline \end{array} \rightarrow \begin{array}{|c|c|}\hline \# & \\ \hline \# & a \\ \hline \end{array} \mid \begin{array}{|c|c|}\hline \# & \\ \hline \# & C \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|}\hline \cdots & a & \mathbf{A} \\ \hline \cdots & \# & \# \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline \cdots & \mathbf{a} & \mathbf{a} \\ \hline \cdots & \# & \mathbf{B} \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline \cdots & \mathbf{a} & a \\ \hline \cdots & \mathbf{B} & a \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|}\hline \# & a & \cdots & a \\ \hline \# & \mathbf{B} & \cdots & a \\ \hline \end{array} \rightarrow \begin{cases} \begin{array}{|c|c|c|c|c|}\hline \# & a & \cdots & a & a \\ \hline \# & \mathbf{a} & \cdots & a & a \\ \hline \end{array} \\ \begin{array}{|c|c|c|c|c|}\hline \# & a & a & \cdots & a \\ \hline \# & \mathbf{C} & a & \cdots & a \\ \hline \end{array} \end{cases}$$

Analyse der Produktionsregeln P_6, P_7, P_8 anhand eines Beispiels:

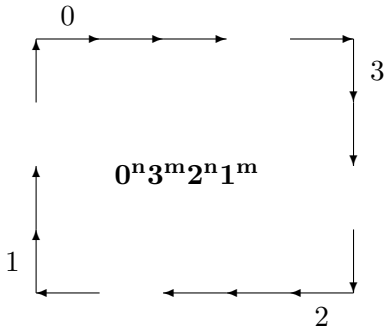
$$P_6 = \begin{array}{|c|}\hline C \\ \hline \# \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline a \\ \hline D \\ \hline \end{array}, P_7 = \begin{array}{|c|c|}\hline & a \\ \hline D & \# \\ \hline \end{array} \rightarrow \begin{array}{|c|c|}\hline & a \\ \hline a & D \\ \hline \end{array}, P_8 = \begin{array}{|c|c|}\hline & \# \\ \hline D & \# \\ \hline \end{array} \rightarrow \begin{array}{|c|c|}\hline & \# \\ \hline a & \# \\ \hline \end{array} \mid \begin{array}{|c|c|}\hline & \# \\ \hline A & \# \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline \# & \mathbf{C} & a & \cdots \\ \hline \# & \# & \# & \cdots \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline \mathbf{a} & \mathbf{a} & \cdots \\ \hline \mathbf{D} & \# & \cdots \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|}\hline a & \mathbf{a} & \mathbf{a} & \cdots \\ \hline a & \mathbf{D} & \# & \cdots \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|}\hline a & \cdots & a & \# \\ \hline a & \cdots & \mathbf{D} & \# \\ \hline \end{array} \rightarrow \begin{cases} \begin{array}{|c|c|c|c|}\hline a & \cdots & a & \# \\ \hline a & \cdots & \mathbf{a} & \# \\ \hline \end{array} \\ \begin{array}{|c|c|c|c|}\hline a & \cdots & a & \# \\ \hline a & \cdots & \mathbf{A} & \# \\ \hline \end{array} \end{cases}$$

Aufgrund der Analyse der Produktionsregeln von G_R, kann festgestellt werden, dass diese Grammatik ein achsparalleles Rechteck, aus 'a' bestehend, erzeugt bzw. erkennt.

$$\begin{array}{ccccc} \# & \# & \cdots & \# & \# \\ \# & \mathbf{a} & \cdots & \mathbf{a} & \# \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \# & \mathbf{a} & \cdots & \mathbf{a} & \# \\ \# & \# & \cdots & \# & \# \end{array}$$

Beispiel achsparalleles Rechteck im Crack Code



1. Ziel

2. Start mit $(\mathbf{AB})^{\mathbf{n}}(\mathbf{CD})^{\mathbf{m}}$
3. $BC \to CB$ bis $\mathbf{A^nC^mB^nD^m}$
4. dann $A \to 0$, $B \to 2$, $C \to 3$, $D \to 1$.

H.8 Zusammenfassung

- Darstellungen beschreiben Bilder und Bildinhalt durch Datenstrukturen.
- Bildmodelle beschreiben ganze Klassen von Bildern.
- Einheiten zur Bilddarstellung: Bild, Zeile, Block, Region, Rand.
- Datenstrukturen: Matrix, Lauflängen, (Binär-)Baum, Medialachse, Quadtree, Octree, Chaincode, Dreiecksnetze, allgemeine Kegel + Zylinder.
- Arraygrammatiken sind prozedurale Bildmodelle.

Kapitel I

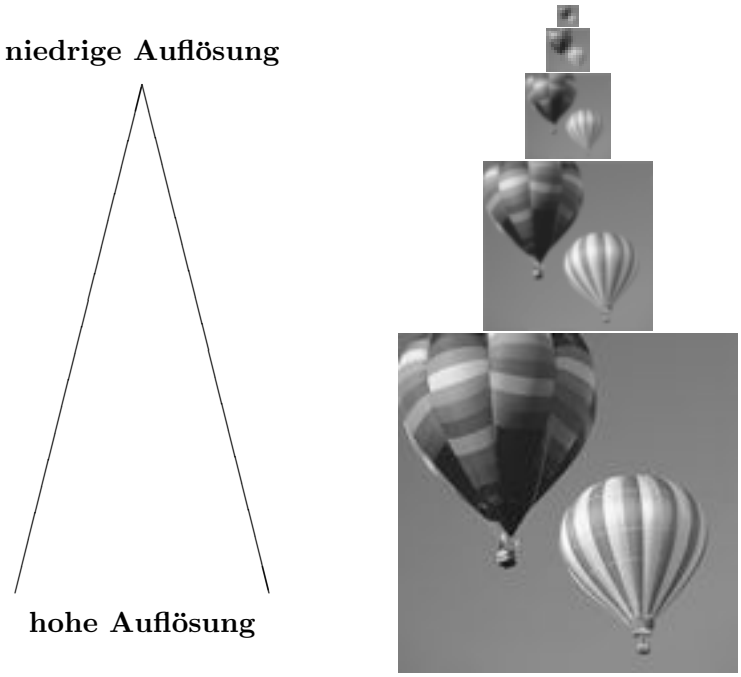
Bildpyramide

I.1 Sehen

Biologisch	**Digital**
Mensch, Tier	Computer
Gehirn $\approx$ 12.000.000.000 Neuronen	$1, 2, \ldots, 64, \ldots$? Prozessoren
Zykluszeit $\approx 1 - 2$ msec	$<<$ 1 nsec !
Wahrnehmung	**Bild-, Mustererkennung**
A) einfacher Szenenausschnitt: $\approx$ 30 msec	
B) komplexe Szene: $\approx$ 1 sec	
Verarbeitungstiefe	
für A) $\approx 15 - 30$ Schritte	massive Parallelität, Hierarchie
für B) einige $\approx$ 100 Schritte	

Optische Wahrnehmung findet bei Menschen und Tieren im Gehirn statt. Dort werden von den Augen empfangene Signale weiterverarbeitet. An der Verarbeitung im Gehirn sind eine Vielzahl an Neuronen beteiligt; in einem Computer wird die Verarbeitung von Bildern von ein oder mehreren Porzessoren erledigt. Die benötigte Zeit zur Verarbeitung von Bildern ist für einen Computer deutlich geringer verglichen mit dem Gehirn (die Zeit für die Aktivierung aller beteiligten Neuronen beträgt 1-2 msec).
Der Mensch übertrifft aber den Computer in der Wahrnehmung einer Szene: dies kann je nach Komplexität der Szene in 30msec - 1sec erfolgen. Um den Inhalt eines Bildes durch einen Computer analysieren zu lassen, bedarf es komplexer Algorithmen der Mustererkennung, die oft sehr viel mehr Zeit als ein Mensch zur Analsyse einer Szene benötigen.

I.2 Bildpyramide: Multiskalenanalyse von Burt und Adelson [1]



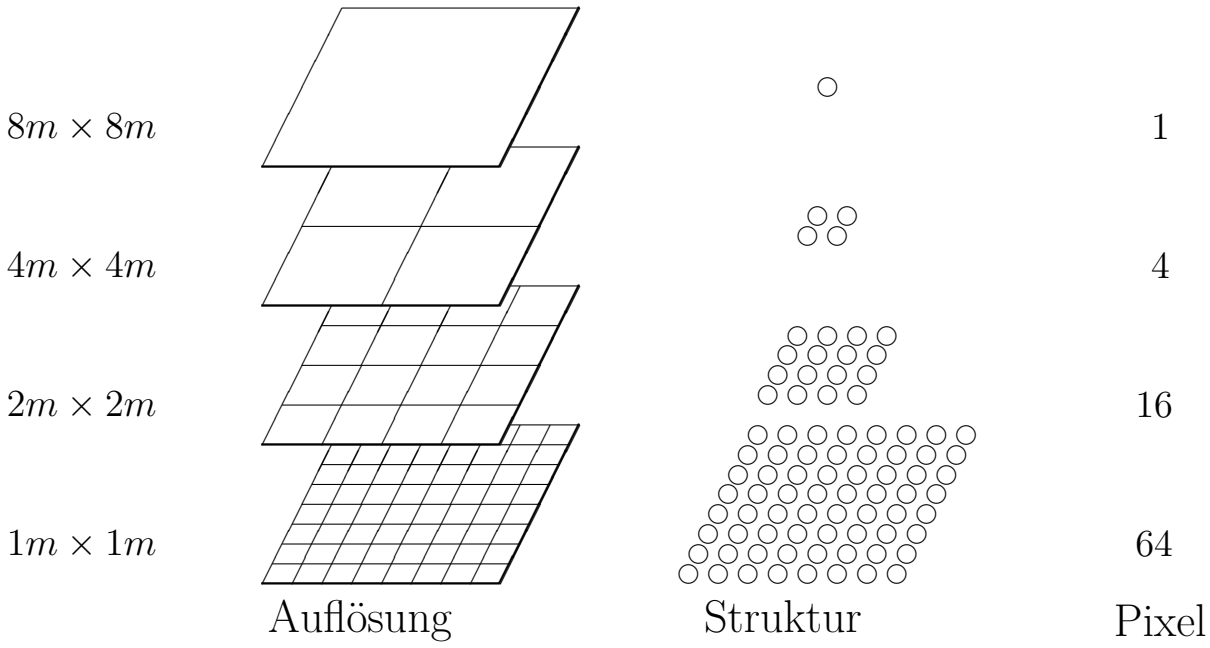
Zerlegung des Bildes in verschiedene Auflösungsebenen: Grundlagen der Multiresolutionsanalyse.

I.3 Multiresolution

	Auflösung / Maßstab FEIN	GROB
Datenmenge	groß	klein
Speicherbedarf	groß	klein
Rechenzeit	sehr lange	rel. kurz
Detailreichtum	groß	gering
Überblick	gering	gut
Genauigkeit	hoch	rel.gering

Um die Vorteile unterschiedlicher Auflösungen /Maßstäbe zu nützen, werden Bildpyramiden verwendet. Die Multiskalenanalyse (engl.: multiresolution analysis) wird in der Signalverarbeitung, der Sprachverarbeitung und in der Bildverarbeitung verwendet. Die in der Bildverarbeitung verwendeten Pyramiden-Methoden stellen einen Vorgänger der Multiskalenanalyse dar. Für die Pyramiden-Darstellung wird ein Signal bzw. Bild wiederholtem Glätten und Subsampling unterzogen.
Eine Matrixpyramide ist eine Folge $\{M_L, M_{L-1}, \ldots, M_0\}$ von Bildern, wobei Bild M_L die gleichen Dimensionen wie das Originalbild hat und M_{i-1} von M_i ausgehend durch Verkleinerung gebildet wird, als Reduktion der Auflösung um zB.: den Faktor 1/2.

I.4 Die klassische $2 \times 2/4$ Pyramide

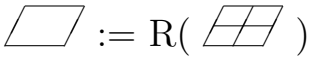


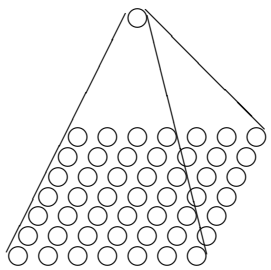
Bei der klassischen $2 \times 2/4$ Pyramide wird die nächst kleiner Auflösung jeweils durch Reduktion der Größe des Bildes, um den Faktor 1/2 an beiden Seiten, gewonnen. Dadurch wird die Anzahl der Pixel im Bild geviertelt.

I.4.1 Die Parameter der $2 \times 2/4$ Pyramide

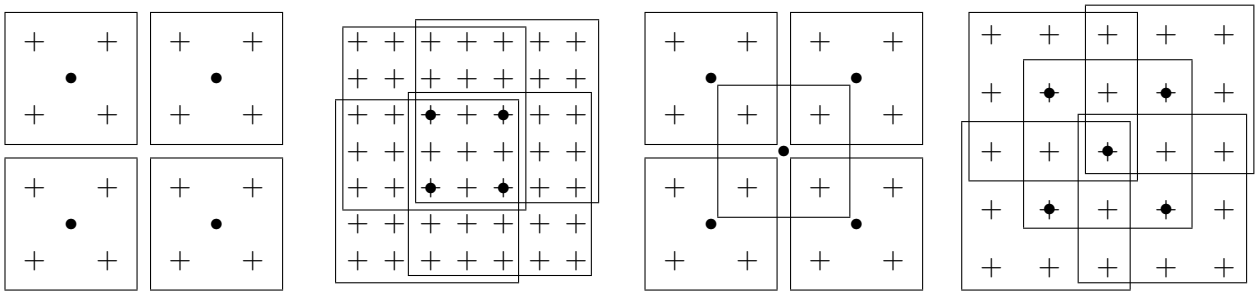
Reduktionsfenster: 2×2

Reduktionsfaktor: 4

Reduktionsfunktion: 

Rezeptives Feld: 

I.5 Andere regelmässige Pyramidenstrukturen



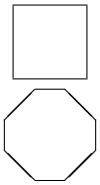
(a) $2 \times 2/4$ (b) $5 \times 5/4$ (c) $2 \times 2/2$ (d) $3 \times 3/2$

Abbildung I.1: regelmäßige Pyramiden

Bei diesen Strukturen handelt es sich um populäre Formen regelmäßiger Pyramiden. Ein + kennzeichnet Pixel der Ebene k, ein • Pixel der Ebene k+1 und Rechtecke/Rahmen kennzeichen die Reduktionsfenster der Pyramiden.

Form der rezeptiven Felder:

$k \times k/4$



$k \times k/2$

Überlappende rezeptive Felder: $k \times k/f > 1$
Eine Zelle (+) hat im Schnitt k^2/f Eltern(•)
Pyramid (Re-)Linking $\longrightarrow$ Segmentation

I.6 Zellinhalt eines Knotens

1. Zelle = Pixel (T. Hong [11], 1982)

   - Pyramide = Stapel von Bildern
   - Zellinhalt = mittlerer Grauwert des rezeptiven Feldes
   - $\rightarrow$ Gaußpyramide (P. Burt [2], 1981, J. Koenderink [15], 1984),
   - $\rightarrow$ Verkettete Pyramide (P. Burt [3], 1981, P. Nacken [20], 1995)

2. Zelle = Parameter eines Modells des Zellinhaltes (R.L. Hartley, 1984)

   - Zellinhalt = Best Fit des rezeptiven Feldes (kleinste Quadrate)
   - $\rightarrow$ Merkmalspyramide

Zellmerkmal	Autor(en)	Jahr
Kante	M. Shneier [26]	1981
codierte Kurve	G. Hartmann [10]	1984
Kurve, Ecke	R.L. Hartley [9]	1984
Gipfel, Täler ($2 \times 2/2$-Pyr.)	J. Crowley [5]	1984

I.7 Numerische Berechnungen in einer Pyramide

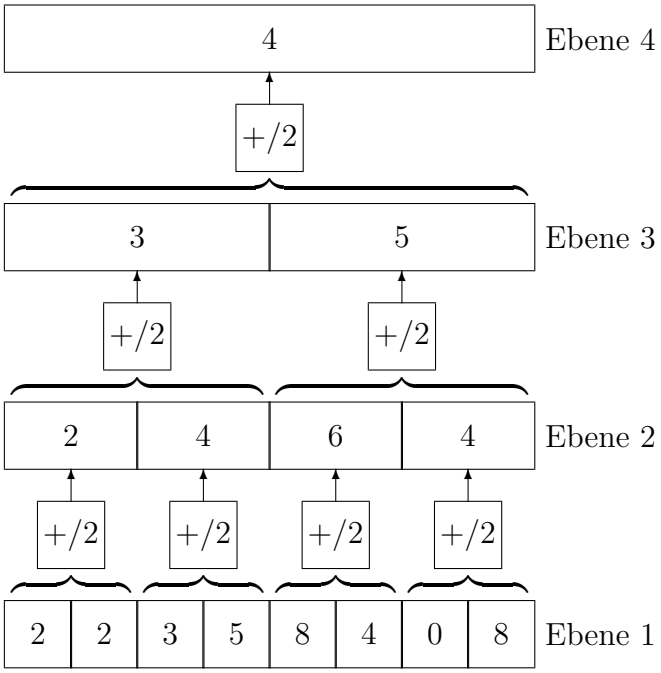
Berechnung des Mittelwerts μ in einer Pyramide
Reduktionsfunktion:

$$\mu_i^{(j+1)} = \frac{\mu_{2i}^{(j)} + \mu_{2i+1}^{(j)}}{2}$$

Wobei $j = \{1, \ldots, n\}$ die jeweilige Ebene kennzeichnet und $i = \{0, \ldots, m\}$ die Zelle.

Ähnlich: Zählen
Summe
Varianz
Momente
kleinste Quadrate
integrale Bildeigenschaften

Parallele Komplexität: $\mathcal{O}(\log Durchmesser)$



I.8 Fehlerfortpflanzung in einer Pyramide

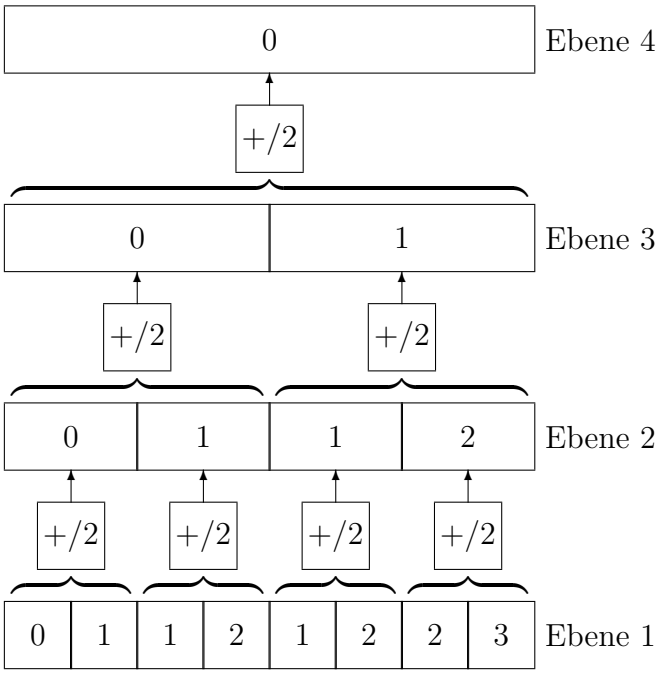
Rundungsfehler
bei ganzzahliger Abspeicherung des Mittelwerts μ.
Maximaler Fehler pro Abspeicherung: 0.5

Top-Down Konstruktion der maximalen Fehlerpyramide, berechneter Wert in der Spitze sei $\mu^{\text{Top}} = 0$

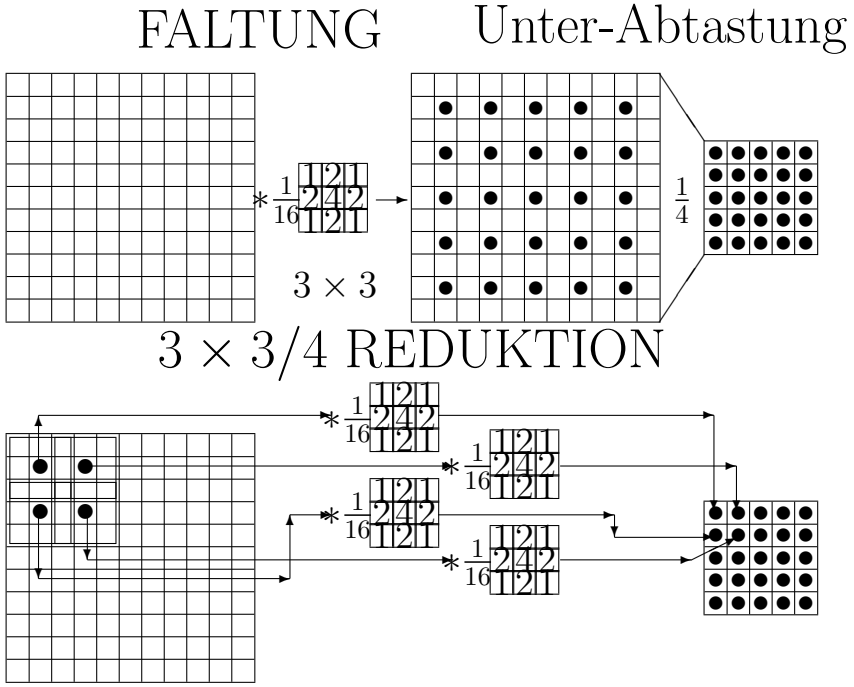
Sei n die Ebene unter der Spitze

$\mu^{\text{Basis}} = \frac{\text{Summe}}{\text{Anzahl}} = \frac{\sum_{i=0}^{n} \binom{n}{i} \cdot i}{2^n} =$
$\frac{n \cdot \sum_{i=1}^{n} \binom{n-1}{i-1}}{2^n} = \frac{n \cdot 2^{n-1}}{2^n} = \frac{n}{2}$

In diesem Fall: $\mu^{\text{Basis}} = 1.5$

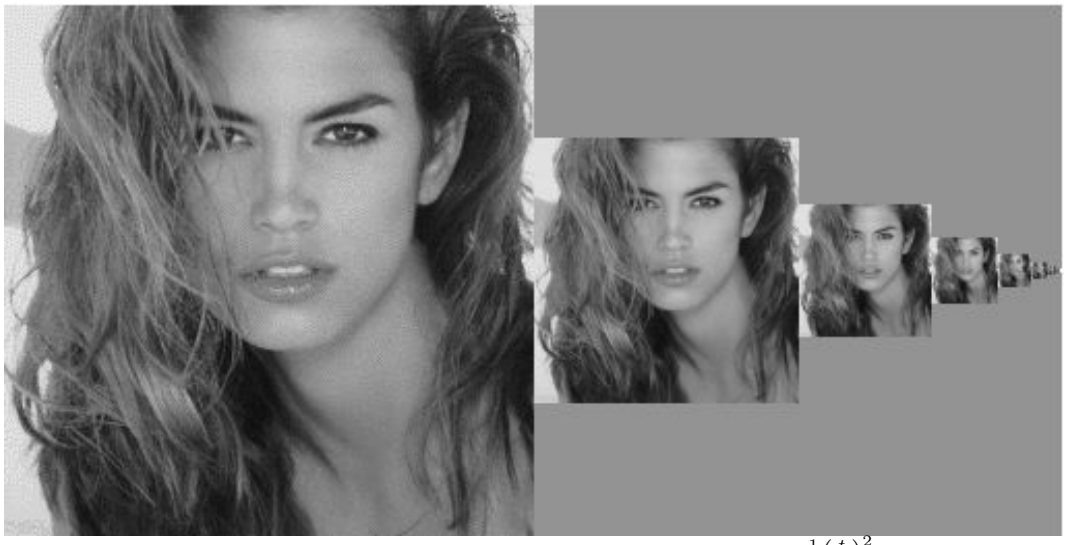


I.8.1 Berechnung einer $3 \times 3/4$ Reduktion



Eine $3\times3/4$ Reduktion wird berechnet, in dem das Originalbild mit einem 3×3 Filterkern (hier Gaußfilter) gefalten wird. Es werden anschließend nur die berechneten Werte jedes zweiten Pixels (von sowohl Spalten und Zeilen des Bildes) zu dem Ergebnisbild zusammengesetzt, dadurch ensteht ein Bild, von 1/4 der Größe des Originalbildes.

I.9 Gaußpyramide

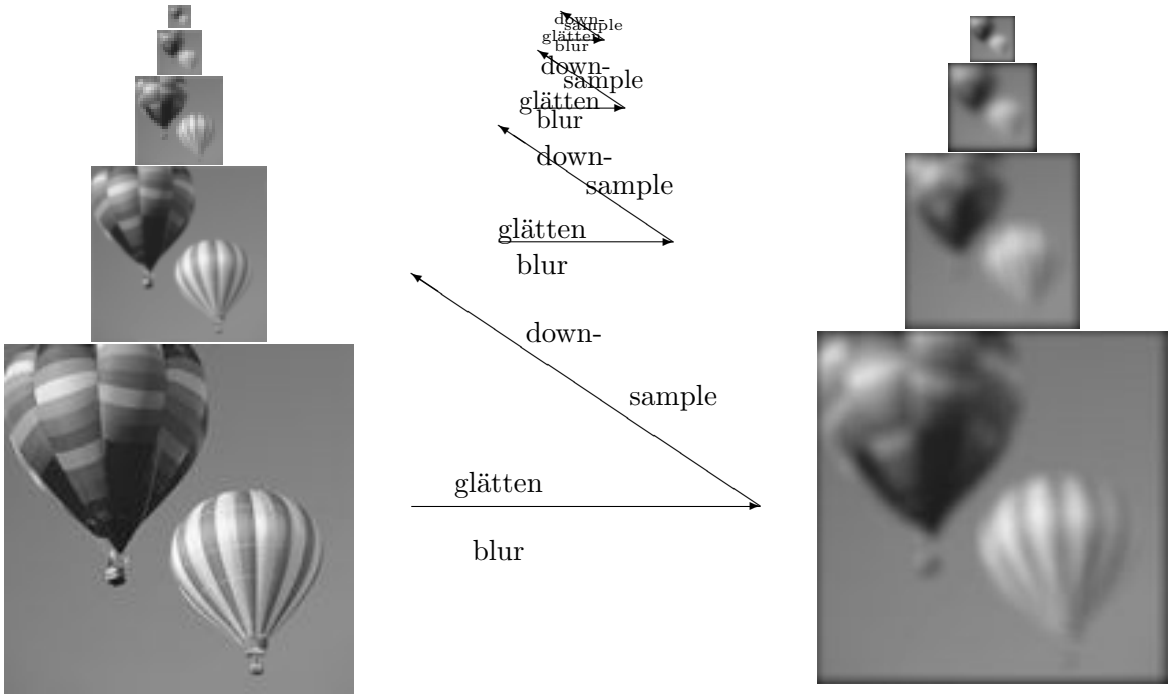


Faltungskern = Gaußfunktion $g(t,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2}$

$t \ldots$ Abstand zum Zentrum, Reduktionsfenster: $g(t,\sigma) > \epsilon$

Die Gaußpyramide wird durch wiederholtes Glätten durch Anwendung eines Gaußfilters (siehe Kapitel D.12, Seite 43) und anschließendes Downsampling gebildet.

I.9.1 Aufbau der Gaußpyramide



Die Gaußpyramide zerlegt ein Bild in verschiedene Glättungsstufen, ohne den Berechnungsaufwand exponentiell ansteigen zu lassen. Hier wird das Abtasttheorem nach Nyquist konsequent ausgenutzt: filtert man die Hälfte des Frequenzspektrums eines Bildes mit einem Tiefpaßfilter heraus, so lässt sich das entstehende Bild ohne Verlust an Information mit der Hälfte der Stützstellen (Bildpunkte) darstellen. Das Abtasttheorem stellt hierbei sicher, dass das Bild mit höherer Auflösung aus dem mit niedrigerer Auflösung wiederhergestellt werden kann.

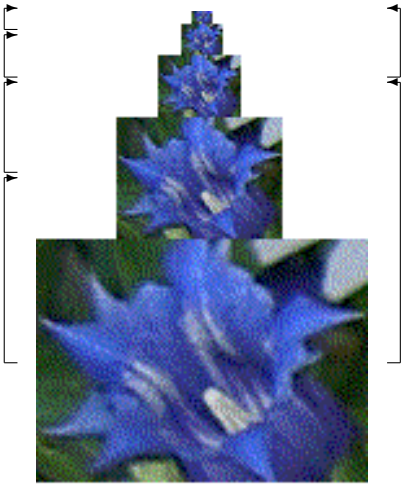
I.9.2 Difference of Gaussians (DOG)

Eine Anwendung findet die Gaußpyramide im Difference of Gaussians (DOG) Verfahren. In diesem Verfahren wird eine geglättete Version eines Bildes von einer anderen weniger stark geglätteten Version dieses Bildes subtrahiert. Das entspricht der Subtraktion aufeinanderfolgender Ebenen einer Gaußpyramide. Durch Anwendung von DOG erhält man ein Kantenbild (siehe Abbildung I.2(c)). Das in I.2(c) gezeigte Bild erhält man durch Subtraktion von Bild I.2(b) von I.2(a). Zur bessern Darstellbarkeit zeigt Abbildung I.2(c) allerdings das Komplementärbild des Ergebnisses dieser Subtraktion.

(a) Originalbild (b) geglättetes Bild (c) DOG

Abbildung I.2: Difference of Gaussians

I.10 Äquivalente Gewichtsfunktion



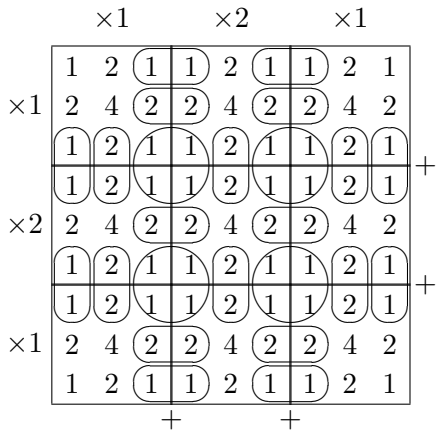
- Wiederholte Gaußfilterung
  $\boxed{G(\sigma) * (G(\sigma) * B) = G(\sigma\sqrt{2}) * B}$
- die Fläche unter der Gaußkurve ist konstant $\Rightarrow$
- grösseres σ bewirkt breitere Kurve und grösseres Reduktionsfenster.

Die iterative Pyramidenbrechnung ist äquivalent zu einer Faltung des Originalbildes mit einer Menge äquivalenter Gewichtsfunktionen. Die Breite dieser äquivalenten Gewichtsfunktionen erhöht sich dabei bei jeder neuen Ebene. Die einzelnen Ebenen einer Pyramide können daher mit zwei unterschiedlichen Ansätzen berechnet weden:

- iterativ: Ebene für Ebene bottom to top
- direkt aus dem Originalbild unter Verwendung der äquivalenten Gewichtsfunktion und entsprechend höheren Abtastung

I.10.1 Äquivalente Gewichtsfunktion der $3 \times 3/4$ Gaußpyramide

$$\left(\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \right)^2 \equiv \frac{1}{256} \begin{pmatrix} 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ 2 & 4 & 6 & 8 & 6 & 4 & 2 \\ 3 & 6 & 9 & 12 & 9 & 6 & 3 \\ 4 & 8 & 12 & 16 & 12 & 8 & 4 \\ 3 & 6 & 9 & 12 & 9 & 6 & 3 \\ 2 & 4 & 6 & 8 & 6 & 4 & 2 \\ 1 & 2 & 3 & 4 & 3 & 2 & 1 \end{pmatrix}$$



Die Berechnung der Äquivalenten Gewichtsfunktion der $3 \times 3/4$ Gaußpyramide kann ausgehend von einem 3×3 Gaußfilter auch folgendermaßen berechnet werden:
Der Filterkern des 3×3 Gaußfilters wird $3\times$ nebeneinander und $3\times$ untereinander geschrieben, dies ergibt eine 9×9 Matrix. Die Werte der jeweils mittleren 2 Spalten bzw. 2 Zeilen erscheinen dabei doppelt. Anschließend werden 3. und 4 Spalte, bzw. Zeile sowie 6. und 7. Spalte bzw. Zeile gewichtet addiert. So erhält man die äquivalente Gewichtsfunktion der $3 \times 3/4$ Gaußpyramide.

I.11 Pyramidenmaschinen

Name	Wo?	Wann?	Wer?	Anz.Proz.	Links	Proz.Typ
PCLIP	Wash.,USA	1980	Tanimoto		13	1-bit
PAPIA	Italien	1981	Cantoni, Levialdi	128×128	5	1-bit
GAM	G.Mason,USA	1984	Schaefer	16×16	9	MPP-chip
HCL	Wash.,USA	1984	Tanimoto	16×16	13	NMOS-VLSI
EGPA	Erlangen,D	1984		4×4		32-bit
SPHINX	Paris,F	1985	Merigot	16×16	7	
PVM	D.Sarnoff,USA	1985	Burt etal	2		Pipeline
SFU2D	S.Fraser,Can	1993	Li, Zhang	$512, 63, 1$		AIS-4000, T-800, Sun-4

I.11.1 Störungsreduktion, Rechenkomplexität

Störungsreduktion:

- normalverteilte Störung $S(\tau) = G(\tau)$
- Filterung einer additive Störung (siehe Kapitel D.14) $F * B = F * (I + S(\tau)) = F * I + F * G(\tau)$
- $F * I$ glättet Ränder $\longrightarrow$ unscharf $F * G(\tau)$ reduziert die Störung

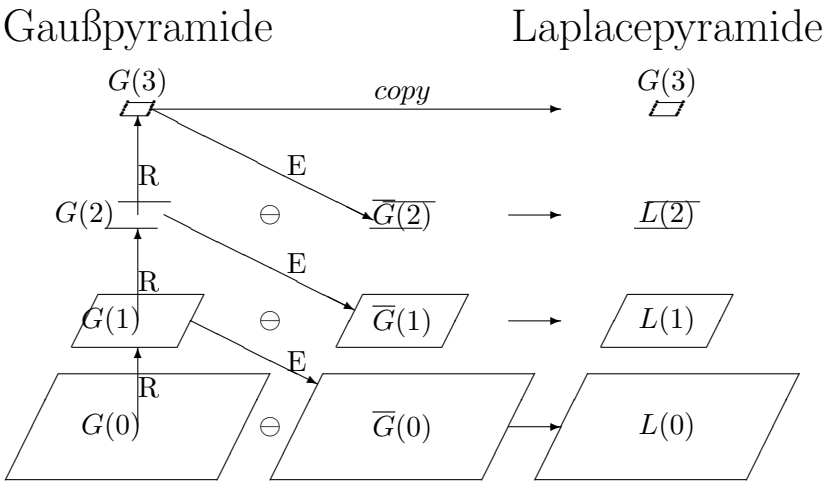
Rechenkomplexität:

- äquivalente Fenstergrössen $(3 \times 3/4)$: $3 \times 3, 7 \times 7, 15 \times 15, \ldots, (2^{k+1} - 1) \times (2^{k+1} - 1)$
- bei direkter Berechnung der Ebene k: $\frac{n^2}{4^k}(2^{k+1} - 1)^2$
- Ebene für Ebene (schrittweise): $\frac{n^2}{4}3^2 + \frac{n^2}{4^2}3^2 + \ldots + \frac{n^2}{4^k}3^2 = 9n^2\left(1 - \frac{1}{4^k}\right)/3$

Ebene k	direkt	schrittweise
1	$9/4n^2 = 2.25n^2$	$9/4n^2 = 2.25n^2$
2	$49/16n^2 = 3.06n^2$	$45/16n^2 = 2.81n^2$
3	$225/64n^2 = 3.52n^2$	$189/64n^2 = 2.95n^2$

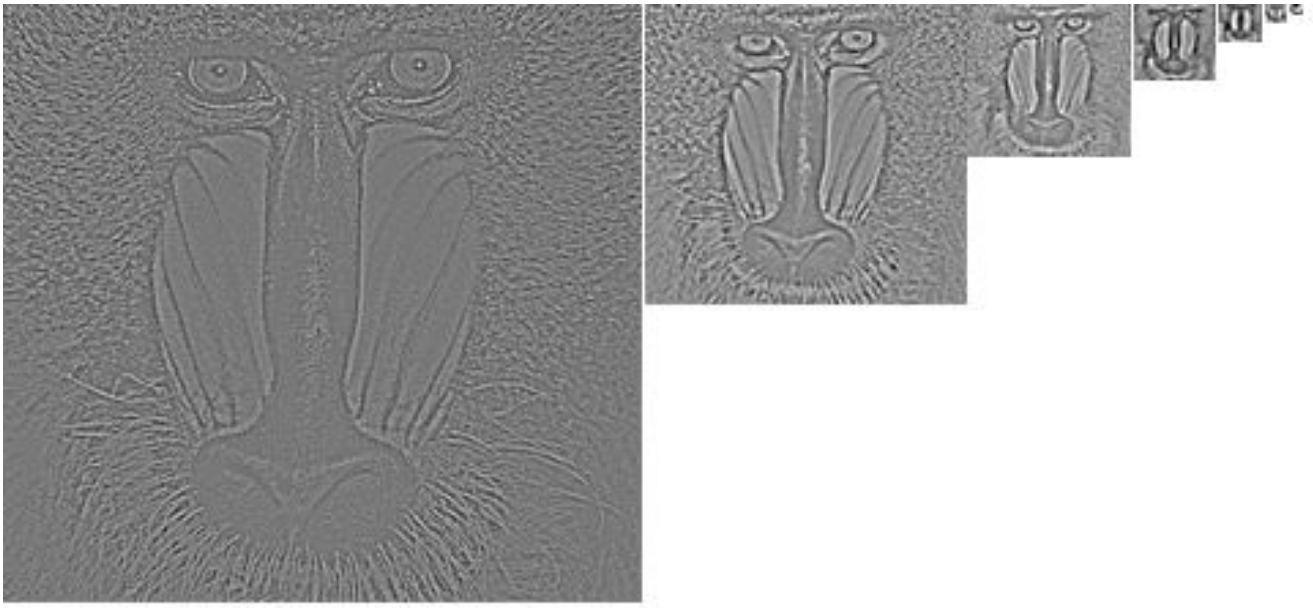
I.12 Aufbau der Laplacepyramide

Um eine Laplacepyramide zu konstruieren, muss zuerst eine Gaußpyramide konstruiert werden. Eine Laplacepyramidenebene wird durch Subtraktion zweier aufeinanderfolgender Ebenen der Gaußpyramide erzielt. Dabei ist ein Resampling nötig, da diese beiden Ebenen die gleiche Größe aufweisen müssen. So erhält man bandpaßgefilterte Ergebnisbilder. Ein Bandpaßfilter kann durch die Differenzbildung der Filterantworten von zwei Glättungsmasken mit unterschiedlicher Standardabweichung erzeugt werden.



Reduktionsfunktion $G(i+1) = R(G(i))$
Expansionsfunktion $\overline{G}(i-1) = E(G(i))$
Laplaceebene $L(i) = G(i) - E(R(G(i)))$

I.12.1 Laplacepyramide



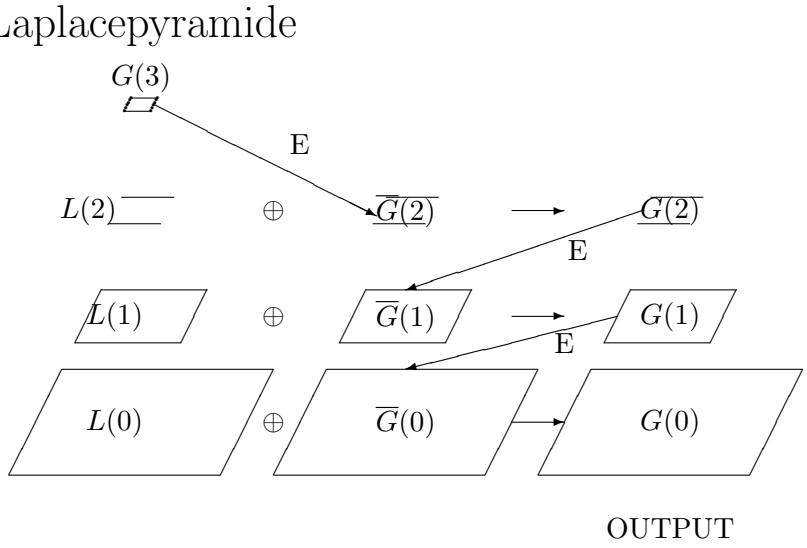
Viele kleine Werte (grau) Quantisierung Kompression

Ein beliebter Anwendungsbereich der Laplacepyramide ist die Datenkompression. Hierfür werden hohe Frequenzen gefiltert, da diese den geringsten Informationsanteil darstellen. Dazu werden die höchsten Laplacepyramidenebenen weggelassen.

I.12.2 Bildrekonstruktion aus der Laplacepyramide

Zur Rekonstruktion der Gaußpyramide aus der Laplacepyramide wird folgendermaßen berechnet:
Die Berechnung erfolgt top down. Die höchste Ebene der Gaußpyramide ist bekannt, da diese bei der Berechnung der Laplacepyramide ohne Reduktion kopiert wurde. Die nachfolgenden Ebenen werden

brechnet, in dem die nächst höhere Ebene der Gaußpyramide expandiert wird und zur zu rekonstruierenden Ebene der Laplacepyramide addiert wird.



Ohne Quantisierung exakte Rekonstruktion von $G(0)$

mit Quantisierung $\Longrightarrow$ 20:1 KOMPRESSION

I.13 Mallat's Waveletpyramide

$$A^d_{2^{j+1}} f \to \begin{cases} \boxed{\tilde{G}_{Zeilen}} \to \boxed{2 \downarrow 1} \to \begin{cases} \boxed{\tilde{G}_{Spalten}} \to \boxed{1 \downarrow 2} \to D^3_{2^j} f & \text{Ecken} \\ \boxed{\tilde{H}_{Spalten}} \to \boxed{1 \downarrow 2} \to D^2_{2^j} f & \text{vertikale Kanten} \end{cases} \\ \boxed{\tilde{H}_{Zeilen}} \to \boxed{2 \downarrow 1} \to \begin{cases} \boxed{\tilde{G}_{Spalten}} \to \boxed{1 \downarrow 2} \to D^1_{2^j} f & \text{horizontale Kanten} \\ \boxed{\tilde{H}_{Spalten}} \to \boxed{1 \downarrow 2} \to A^d_{2^j} f & \text{Rekursion} \end{cases} \end{cases}$$

$\boxed{X}$... 1D Faltung (Zeilen oder Spalten) mit Filterkern X
$\boxed{2 \downarrow 1}$... nur jede 2. Spalte, jede Zeile
$\boxed{1 \downarrow 2}$... jede Spalte, nur jede 2. Zeile
G ... Hochpassfilter
H ... Tiefpassfilter

Die Zeilen des Bildes werden sowohl Hochpass als auch Tiefpass gefiltert und dabei in ihrer Länge um 2 unterabgetastet (Downsampling: nur jede 2. Spalte bleibt bestehen). Anschließend werden die Spalten hochpass und tiefpass gefiltert und um 2 unterabgetastet (Downsampling: nur jede 2. Zeile bleibt bestehen).

I.13.1 Beispiel: 1D Haar-Wavelet

Das Haar-Wavelet ist das erste in der Literatur bekannt gewordene Wavelet. Es ist das einfachste Wavelet und kann aus der Kombination zweier Rechteckfunktionen gebildet werden. Das Haar-Wavelet lässt sich sehr einfach implementieren, da es aber aus der Kombination zweier Rechtecksfunktionen gebildet wird, ist es unstetig und daher auch nicht differenzierbar.

1. bestimme paarweise Mittelwerte, $\frac{a+b}{2}$;

2. bestimme paarweise Differenzen zwischen den Originalwerten und den Mittelwerten, $a - \frac{a+b}{2} = \frac{a-b}{2} = \frac{a+b}{2} - b$;
3. fülle die erste Hälfte des Arrays mit den Mittelwerten und
4. die zweite Hälfte mit den Differenzen.
5. Wiederhole Punkt 3 und 4 für jeweils die erste Hälfte der vorangegangen Ebene, so lange möglich.

$I_0 =$ | 1 | 3 | 4 | 8 | 4 | 0 | 2 | 2 |

Mittelwerte: Differenzen:

$(1+3)/2 = 2$ $1 - 2 = 2 - 3 = -1$

$(4+8)/2 = 6$ $4 - 6 = 6 - 8 = -2$

$(4+0)/2 = 2$ $4 - 2 = 2 - 0 = 2$

$(2+2)/2 = 2$ $2 - 2 = 2 - 2 = 0$

$I_1 =$

2	6	2	2	−1	−2	2	0
$(I_0 * \tilde{H}) \downarrow 2$				$(I_0 * \tilde{G}) \downarrow 2$			

$I_1 =$ | 2 | 6 | 2 | 2 | −1 | −2 | 2 | 0 |

Mittelwerte: Differenzen:

$(2+6)/2 = 4$ $2 - 4 = 4 - 6 = -2$

$(2+2)/2 = 2$ $2 - 2 = 2 - 2 = 0$

$I_2 =$

4	2	−2	0	−1	−2	2	0
$(I_1 * \tilde{H}) \downarrow 2$		$(I_1 * \tilde{G}) \downarrow 2$					

$I_2 =$ | 4 | 2 | −2 | 0 | −1 | −2 | 2 | 0 |

Mittelwerte: Differenzen:

$(4+2)/2 = 3$ $4 - 3 = 3 - 2 = 1$

$I_3 =$

3	1	−2	0	−1	−2	2	0
$(I_2 * \tilde{H}) \downarrow 2$	$(I_2 * \tilde{G}) \downarrow 2$						

I.13.2 Rekonstruktion

Grundidee: $\frac{a+b}{2} + \frac{a-b}{2} = a$ und $\frac{a+b}{2} - \frac{a-b}{2} = b$

$I_3 =$

3	1	−2	0	−1	−2	2	0

$I_2 =$

3 + 1	3 − 1	−2	0	−1	−2	2	0
$(I_3 \uparrow 2) * H$	$(I_3 \uparrow 2) * G$						

Zum Verschmelzen mit $\uparrow 2$ werden dazwischen 0 eingefügt und zweilenweise berechnet:

$I_3 \uparrow 2 =$	0	3	0	$*H =$	1.5	1.5
+	0	1	0	$*G =$	0.5	−0.5
$I_2 =$				$*2 =$	4	2

Das Ergebnis erhält man durch anschließendes Addieren der Spalten und Verdoppeln des Ergebnisses.

$I_2 =$ | 4 | 2 | −2 | 0 | −1 | −2 | 2 | 0 |

Die Werte für H und G sind durch das 1D Haar-Wavelet gegeben:

$\Longrightarrow$ 1D Haar-Wavelet:

$$\boxed{\begin{array}{l}\tilde{H} = (0.5, 0.5)\\ \tilde{G} = (-0.5, 0.5)\\ H = (0.5, 0.5)\\ G = (0.5, -0.5)\end{array}}$$

$I_2 =$	4	2	-2	0	-1	-2	2	0
$I_1 =$	$4+(-2)$	$4-(-2)$	$2+0$	$2-0$	-1	-2	2	0
	$(I_2 \uparrow 2) * H$	$(I_2 \uparrow 2) * G$						

$$I_1 = \boxed{\begin{array}{|c|c|c|c||c|c|c|c|}\hline 2 & 6 & 2 & 2 & -1 & -2 & 2 & 0\\ \hline\end{array}}$$

$I_1 =$	2	6	2	2	-1	-2	2	0
$I_0 =$	$2+(-1)$	$2-(-1)$	$6+(-2)$	$6-(-2)$	$2+2$	$2-2$	$2+0$	$2-0$
	$(I_2 \uparrow 2) * H$	$(I_2 \uparrow 2) * G$						

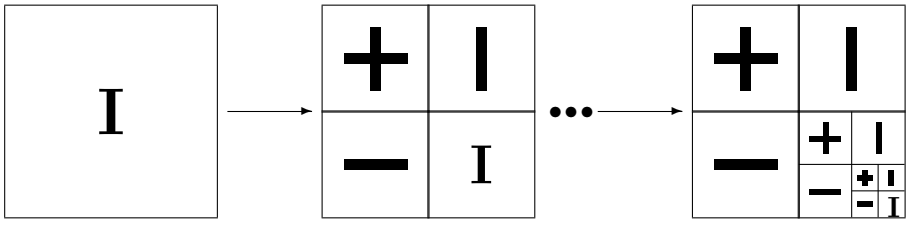
$$I_0 = \boxed{\begin{array}{|c|c|c|c|c|c|c|c|}\hline 1 & 3 & 4 & 8 & 4 & 0 & 2 & 2\\ \hline\end{array}}$$

I.13.3 Rekonstruktion aus der Waveletpyramide

$$\left.\begin{array}{l} D^3_{2^j} f \to \boxed{2\uparrow 1} \to \boxed{G_{Zeilen}} \\ D^2_{2^j} f \to \boxed{2\uparrow 1} \to \boxed{H_{Zeilen}} \end{array}\right\} \oplus \to \boxed{1\uparrow 2} \to \boxed{G_{Spalten}}$$
$$\left.\begin{array}{l} D^1_{2^j} f \to \boxed{2\uparrow 1} \to \boxed{G_{Zeilen}} \\ A^d_{2^j} f \to \boxed{2\uparrow 1} \to \boxed{H_{Zeilen}} \end{array}\right\} \oplus \to \boxed{1\uparrow 2} \to \boxed{H_{Spalten}}$$
$$\left.\begin{array}{l} \\ \\ \end{array}\right\} \oplus \to \boxed{*4} \to A^d_{2^{j+1}} f$$

$\boxed{X}$... 1D Faltung (Zeilen oder Spalten) mit Filterkern X
$\boxed{1\uparrow 2}$... füge je eine 0-Zeile ein
$\boxed{2\uparrow 1}$... füge je eine 0-Spalte ein

I.13.4 Wavelet-Pyramide

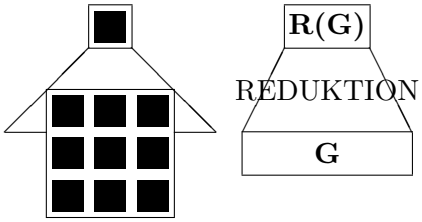


(1-dim.)Berechnungsprinzip nach Mallat:
$I = 2 * ((((I * \tilde{H}) \downarrow 2) \uparrow 2) * H + (((I * \tilde{G}) \downarrow 2) \uparrow 2) * G)$

I	...	(1-dim.) Bildsignal
$H, \tilde{H}$	...	Tiefpaßfilter
$G, \tilde{G}$	...	Hochpaßfilter
$I * G$	...	Faltung
$(2,2,4,8,0,4) \downarrow 2$	=	$(2,4,0)$
$(2,4,0) \uparrow 2$	=	$(0,2,0,4,0,0)$

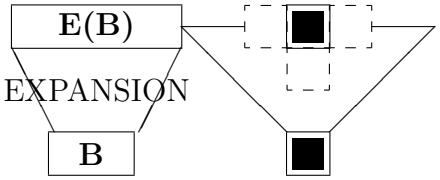
I.14 Zusammenfassung Pyramiden-(Re-)Konstruktion

KONSTRUKTION



- Selektion: Mittelpixel, Maximum, Minimum, ...
- Filter: μ, Gauß, Closing

REKONSTRUKTION



- Projektion
- Interpolation: Filter

Zur Konstruktion einer Pyramide, wird ausgehend vom Originalbild, jede Ebene als Reduktion der vorhergehenden Ebene berechnet. Zur Rekonstruktion einer Ebene wird die nächst kleinere Ebene expandiert.

I.15 Die verschiedenen Differenzbildungen

- **RE-Pyramide** (Reduce - Expand, Burt)
  $D = I - E(R(I))$
- **Ratio-Pyramide** (Toet [28])
  $D = I/E(R(I))$
- **FSD-Pyramide** (Filter - Subtract - Decimate, Burt)
  (siehe DOG Seite 117)

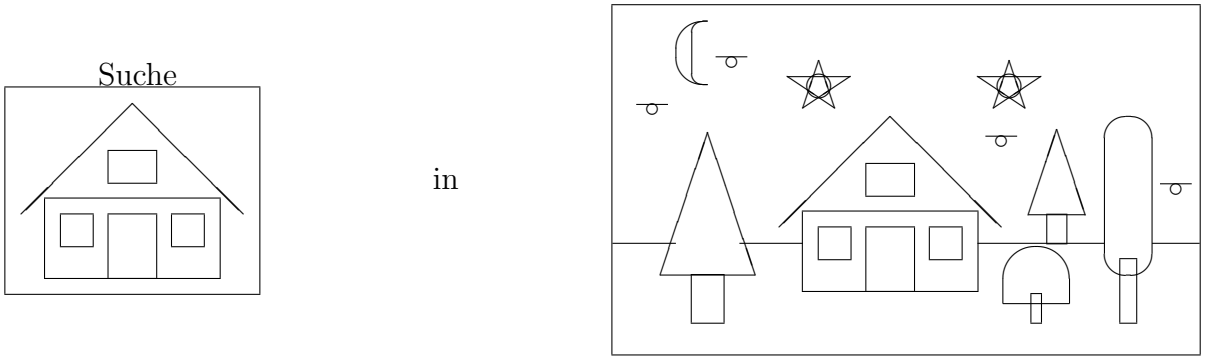
$$\begin{array}{ccc} I & \rightarrow & H * I \\ \downarrow \ominus & \swarrow & \downarrow 4 \\ D & & (H * I) \downarrow 4 \end{array}$$

- **Wavelet** (Mallat [19])
  $D = (I * \tilde{G}) \downarrow 2$

I.16 Anwendungen

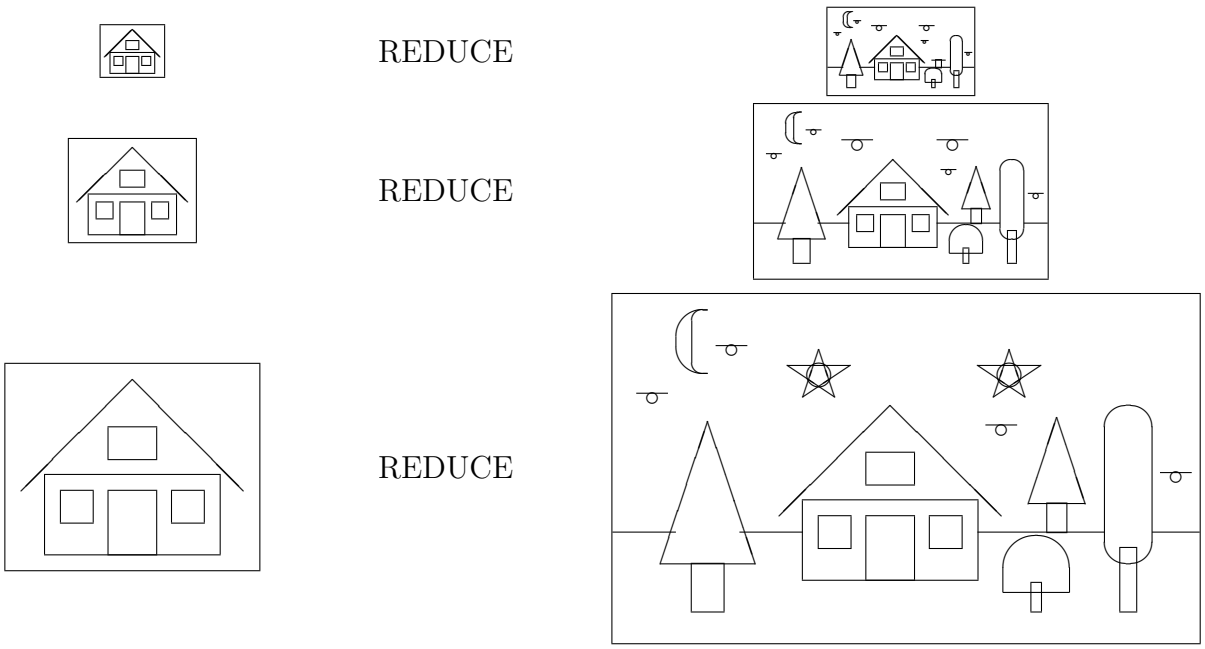
- Gaußpyramide
  - Schätzung integraler Bildeigenschaften
  - Pattern Matching
  - Grob-Fein-Suche
  - Segmentation (Pyramid Linking)
- Laplacepyramide, Wavelet
  - Datenkompression
  - Bildmosaike
  - Alarmsysteme
  - Objektverfolgung (Tracking)

I.17 Schnelles Pattern Matching

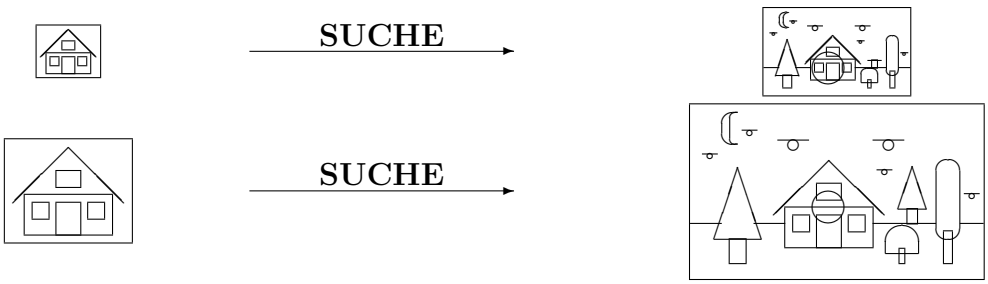


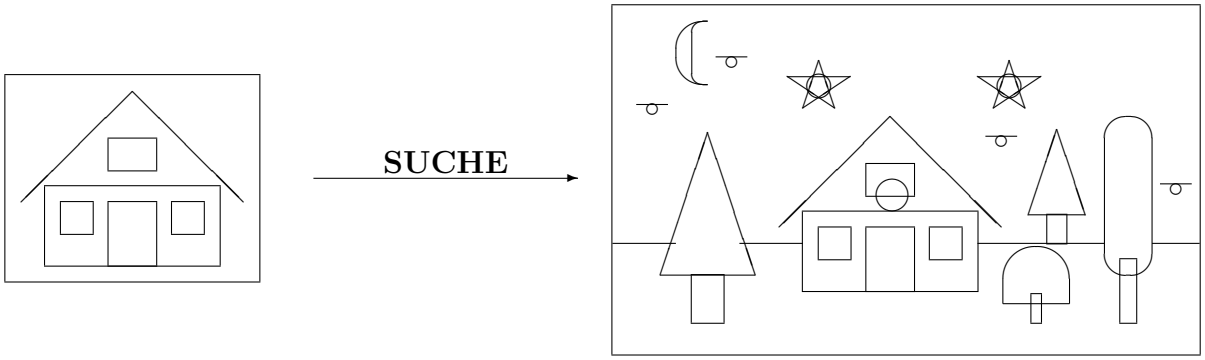
Bildpyramiden finden auch Einsatz im Rahmen von Pattern Matching. Dazu werden Bildpyramiden von sowohl dem zu findenden Objekt, als auch dem Bild, in dem gesucht werden soll erstell (Reduction). Danach wird in einer kleinen Auflösung nach dem Objekt gesucht und dessen Position so grob bestimmt. In den größeren Auflösungen muss nun nicht mehr das gesamte Bild durchsucht werden, sondern nur eine Umgebung um die zuvor grob bestimmte Position, so wird die Position schrittweise genauer (Grob-Fein Suche).

I.17.1 Schnelles Pattern Matching: Reduction



I.17.2 Schnelles Pattern Matching: Grob-Fein Suche





I.18 Keypoint Detektoren

Keypoints finden in der Bildverarbeitung häufig Anwendung: sie werden beim Matching von Stereobildern, beim Tracking von Objekten oder auch für Image Retrieval eingesetzt.

I.18.1 Lowe's SIFT Operator [17], [18]

SIFT (Scale-Invariant Feature Transform) ist ein von David G. Lowe 1999 entwickelter Algorithmus (2004 wurde eine aktualisierte Version publiziert) zur Erkennung und Beschreibung lokaler Bildmerkmale aus Abbildungen (siehe Abbildung I.3). Dadurch wird das ganze Bild auf relativ wenige Keypoints reduziert.

(a) zu findende Objekte

(b) Objekte in Szene

(c) detektierte Objekte

Abbildung I.3: Objekterkennung durch SIFT-Algorithmus; Abbildungen von [17], Seite 1154

SIFT Algorithmus:

1. Merkmalserkennung (feature detection)

   - Ein Eingabebild wird in eine Menge an Merkmalsvektoren umgewandelt (diese sind invariant gegenüber Translation, Skalierung und Rotation und bis zu einem gewissen Grad auch gegenüber Änderungen in der Beleuchtung sowie lokaler geometrischer Verzerrung).
   - Key locations (Keypoints) sind gegeben als Maxima und Minima des Ergebnisses von DOG (siehe Seite 117) durchgeführt im scale-space. Bei dem scale-space handelt es sich um eine Bildfolge entstanden durch wiederholtes Glätten und Resampling von Bildern (= Gaußpyramide siehe Seite 116).

2. Orientation Assignment

   Jedem Keypoint wird, basierend auf lokalen Bildeigenschaften, eine Richtung zugewiesen. Dadurch kann jeder Keypoint Descriptor relativ zu seiner Ausrichtung beschrieben werden und ist somit invariant gegenüber Bild- bzw. Objekt-Rotationen. Die Berechnung der Richtung eines Keypoint Descriptors wird folgendermaßen durchgeführt: Ein orientation histogram (Histogramm siehe Seite 27) wird für Gradientenrichtungen von sample points in einer Umgebung um den Keypoint erstellt. Maxima in diesem Histogramm entsprechen dominanten Richtungen lokaler Gradienten. Der höchste Wert im Histogramm wird bestimmt. Entsprechend diesem werden lokale Maxima detektiert, die mindestens 80% des globalen Maximum ausmachen. Diese entsprechen dann ebenfalls einem Keypoint mit diesem Gradienten. Das heißt, für Positionen mit mehreren Maxima (ähnlichen Ausmaßes) werden an der gleichen Position und mit gleicher Skalierung mehrere Keypoints mit unterschiedlichen Richtungen erzeugt.

3. Matching

   - Nearest Neighbour
     Für das Matching werden berechnete Keypoints verglichen. Für jeden Keypoint des Eingabebildes wird sein nächster Nachbar (nearest neighbour) aus den Keypoints, der in der Datenbank gespeicherten Trainingsbilder, berechnet. Der nächste Nachbar ist definiert als der Keypoint, dessen Vektor die geringste Euklidische Distanz (siehe Seite 88) zum ortsunabhängigen Vektor des gegebenen Keypoints hat. Um die Wahrscheinlichkeit, dass Keypoints korrekt gematcht wurden, zu erhöhen, wird die Distanz zum nächsten Nachbarn mit der zum zweit nächsten Nachbarn verglichen. Lowe's SIFT Algorithmus verwirft Matches bei denen das Verhältnis dieser zwei Distanzen größer als 0.8 ist. Dies eliminiert 90% der falschen Matches und verwirft dabei ca. 5% korrekter Matches.
   - Cluster Erkennung
     Zur Cluster Erkennung wird die Hough Transformation (siehe Seite 62) verwendet um Cluster von Merkmalen (Keypoints) zu finden, die einer bestimmten model pose (= Kombination aus Position und Orientierung) entsprechen.
   - Modell Verifizierung
     Jedes so identifizierte Cluster wird nun eines Verifikationsprozesses unterzogen. Dazu wird das Modell auf das Bild bezogen und die, für diese Transformation nötigen Parameter, der Methode der kleinsten Quardate (linear least square) unterzogen.
   - Detektion von Ausreißern
     Ausreißer können detektiert werden, in dem die Übereinsteimmung zwischen Modell und Bildeigenschaften überprüft wird. Bleiben nach Verwerfung dieser Ausreißer weniger als 3 Keypoints übrig, wird der Match verworfen.

I.18.2 SFOP Keypoint Detector [8]

SFOP (Scale-invariant Feature Operator) ist ein von Förstner et al. 2009 publizierter, skalierungsinvarianter Keypoint Detector. Dieser Ansatz bietet allerdings nur eine Detektion der Keypoints, zur Beschreibung dieser wird der von David G. Lowe im SIFT Algorithmus publizierte Ansatz verwendet.

Der SFOP Ansatz verwendet zwei verschiedene Arten von Keypoints:

1. point-like keypoints: beschreiben einen spezifischen Punkt in einem Bild (z.B.: eine Ecke oder den Mittelpunkt einer runden Fläche) (dunkle Ringe in Abbildung I.4)
2. blob-like keypoints: beschreiben kleine Regionen (nicht notwendigerweise rund) im Bild, in denen keine spezifischen Punkte im Bild indetifiziert werden müssen (helle bzw. umrandete Ringe in Abbildung I.4)

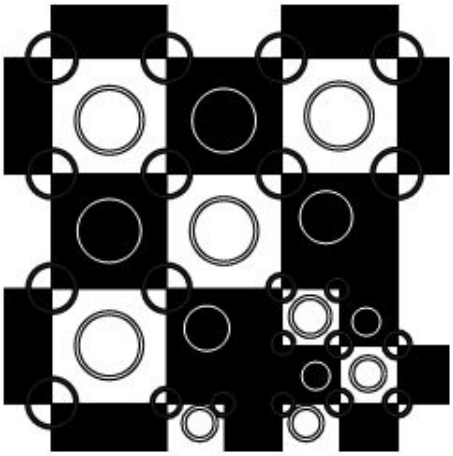
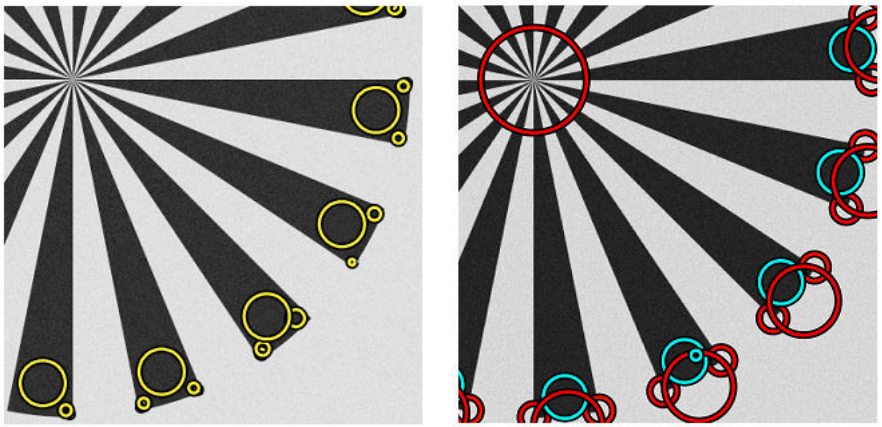


Abbildung I.4: Keypoints detektiert mittels SFOP; Abbildung von [8], Seite 2258

Der SFOP Keypoint Detector basiert auf dem spiral model nach Bigün. Zur Anwendung dieses spiral models wird ein Bild nicht in kartesischen Koordinaten ($f(x, y)$) beschrieben sondern in Polarkoordinaten ($f(r, \phi)$). F wird dann bezüglich r und ϕ Fourier transformiert. Die gesuchten Keypoints sind im Fourier-Raum als Linien durch den Ursprung dargestellt. Abhängig vom Winkel dieser Linien, können Keypoints verschiedener Arten erkannt werden. Ein Wert $\alpha = 0°$ weist auf eine Ecke / Verzweigung hin, $\alpha = 90°$ zeigt kreisförmige Strukturen an.
Der SFOP Keypoint Detector sucht nach Keypoints $[p, \sigma] = [x, y, \sigma]$ in einem geglätteten Bild $g(p, \tau)$. τ ist der Glättungsparameter, p die Position einer Spiral Structur mit Winkel α in einem Fenster von Skalierungsfaktor σ. Als Ergebnis liefter der SFOP Algorithmus Keypoints $[p, \sigma]_{opt} = argmax_{p,\alpha,\tau,\sigma} w(p, \alpha, \tau, \sigma)$.

Abbildung I.5 zeigt einen Vergleich des SFOP Keypoint Detectors (Abbildung I.5(b)) mit dem SIFT Keypoint Detector (Abbildung I.5(a)). SIFT erkennt wie auch SFOP die Strahlen als blob-like keypoints und die Eckpunkte jedes Strahls als point-like keypoints. Zusätzlich dazu wird von SFOP aber auch die Kante am Ende jedes Strahls, sowie der Mittelpunkt der sternförmigen Struktur als Keypoint detektiert. Die in Abbildung I.5(b) mit hellen Ringen gekennzeichneten Keypoints zeigen blob-like Keypoints an, die dunklen Ringe point-like Keypoints.



(a) keypoint detection mit Lowe's SIFT (b) keypoint detection mit SFOP

Abbildung I.5: Vergleich der beiden Keypoint Detectors SIFT und SFOP; Abbildungen von [8], Seite 2261

I.19 Pyramid Linking, Pyramidenwachstum

Im ersten Schritt des Algorithmus wird eine pyramidenartige Datenstruktur aus mehreren Ebenen erstellt, deren unterste Ebene das Eingangsbild ist. Zur Vereinfachung wird das Verfahren nur für ein eindimensionales Bild (Bildzeile) erläutert. Alle Ebenen über dem Originalbild werden gebildet, indem jedes Pixel einer oberen Ebene aus dem Durchschnittswert von vier benachbarten Pixeln aus der darunter liegenden Ebene berechnet wird (siehe Abbildung I.6). Jedes Pixel trägt dabei zu maximal vier Pixeln bei (zwei im eindimensionalen Beispiel).

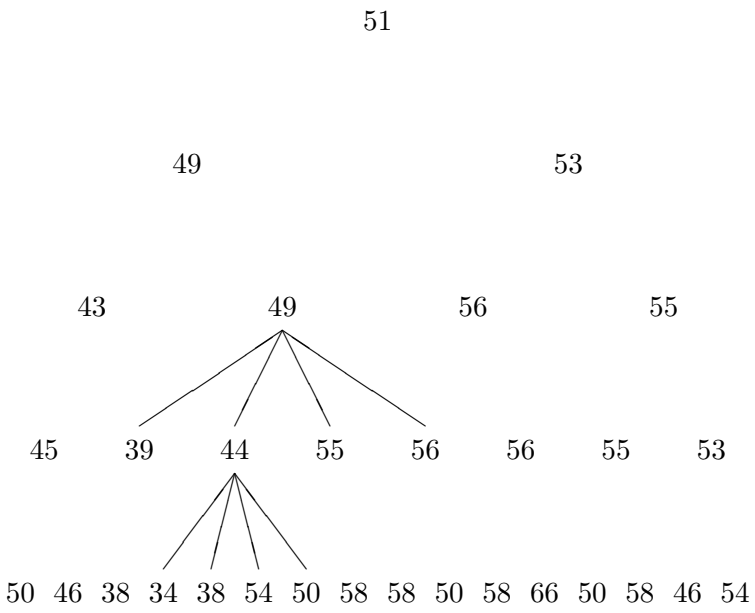


Abbildung I.6: Pyramid Linking - Berechnung der Pyramide

Dadurch reduziert sich die Anzahl der Pixel/Datenmenge von jeder Schicht auf die nächste um die Hälfte. Wenn die Pyramide bis zu einem Pixel in der obersten Ebene gebildet ist, werden alle Pixel aus höheren Ebenen (ausgenommen die Spitze) mit einem von max. zwei benachbarten

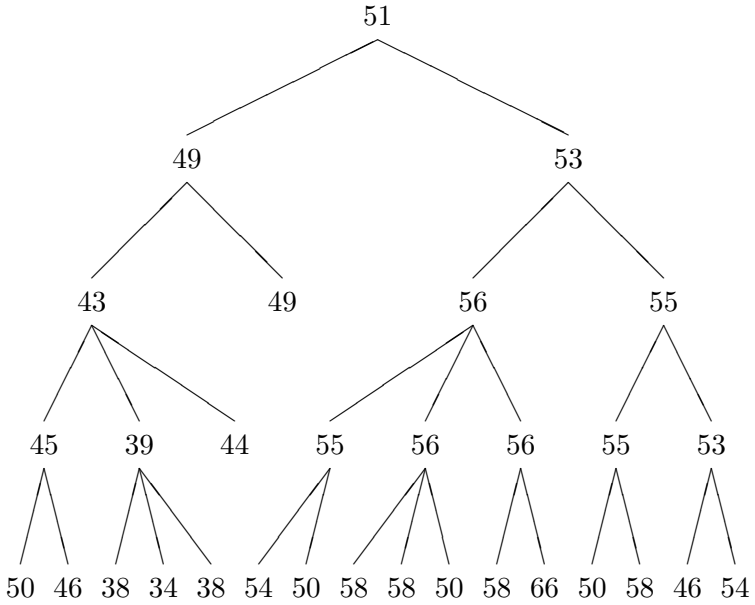


Abbildung I.7: Pyramid Linking - Verbindungen in der Pyramide

Pixel in der nächsthöheren Ebene verbunden. Jedes Pixel wird mit dem Pixel der nächsten Ebene verbunden, zu dessen Grauwert es beigetragen hat und zu dem es den geringsten Unterschied hat (siehe Abbildung I.7).

Nach diesem Initialisierungsschritt wird der Grauwert der Pixel aller oberen Ebenen aktualisiert, indem der Grauwert aus den verbundenen Pixeln der darunterliegenden Ebene neu berechnet wird. Hat ein Pixel der oberen Ebenen (d.h. mit Ausnahme der untersten Ebene) keine Verbindungen zu einem Pixel der nächsten unteren Ebene, so wird der Grauwert auf Null gesetzt und erhält keine weiteren Verbindungen von einer unteren Ebene (siehe Abbildung I.8).

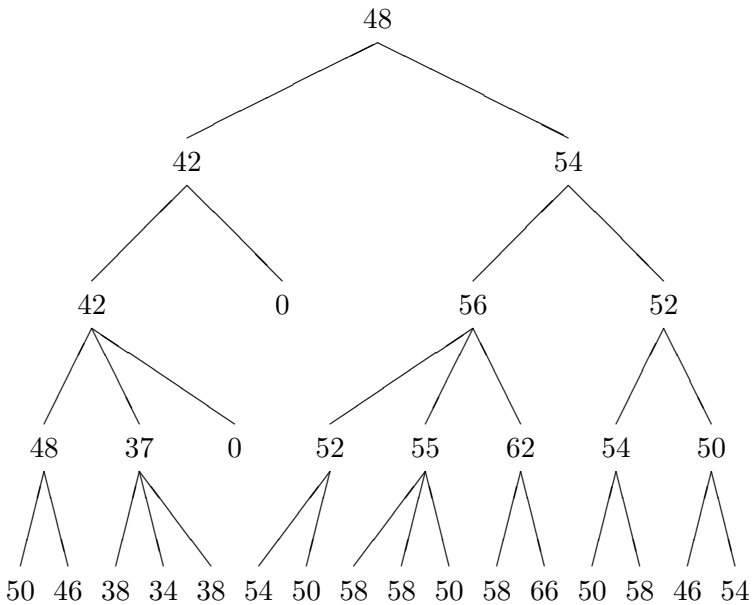


Abbildung I.8: Pyramid Linking - Neuberechnung der Grauwerte

Danach werden die Verbindungen wie zuvor neu gesetzt und die Pixel neu berechnet. Dieser Vorgang wird sooft wiederholt, bis sich die Verknüpfungen nicht mehr ändern. Das Ergebnis ist ein Verknüpfungsbaum, dessen Wurzel die Pyramidenspitze ist von der aus über Verzweigungen Pfade zu jedem Pixel der Grundebene, dem Originalbild, führen.

Der Vorteil dieses Algorithmus im Vergleich zu anderen Segmentierungsverfahren ist, dass sofort eine unterschiedliche Anzahl von Klassifizierungen verfügbar ist. Jede Ebene enthält eine Anzahl von Pixeln, die die obersten Elemente von Verknüpfungsbäumen bilden, dessen Verknüpfungen in die unterste Schicht reichen und so mehrere benachbarte Pixel auf dem Originalbild zu einer Klasse zusammenfassen (siehe Abbildung I.9). Dieses Verfahren kommt sehr gut mit Rauschen zurecht, da jedes Pixel des Eingangsbildes einer benachbarten Klasse zugeordnet wird. Darüber hinaus ist es vergleichsweise schnell und braucht für die Datenstruktur nur 1/7 des Speicherplatzes des Originalbildes.

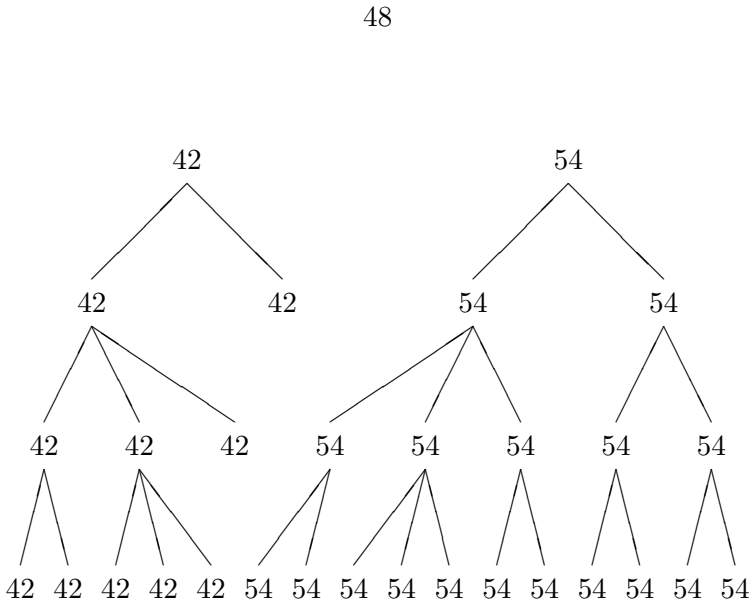
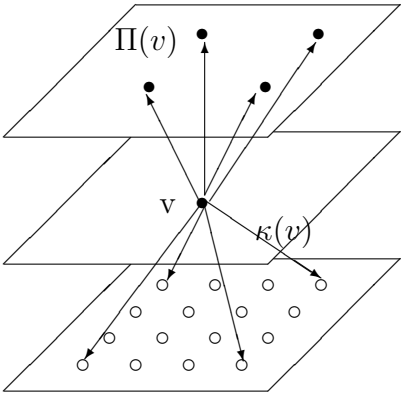


Abbildung I.9: Pyramid Linking - Segmentierung anhand der zweit höchsten Ebene

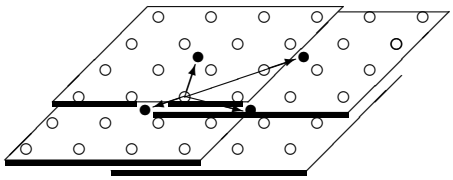
I.19.1 Linking Algorithmus

1. baue PYRAMIDE auf (mögliche Verfahren z.B.: $2 \times 2/4$ Mittelwert oder $4 \times 4/4$ Gauss)
2. für ALLE ZELLEN unter der SPITZE wähle BESTEN Elternteil:
   $\pi_0(v) := argmin\{d(g(v), g(p))|p \in \Pi(v)\}$
3. für ALLE ZELLEN über der BASIS berechne ihre Werte neu:
   $g(v) := \sum_{k \in \kappa_0(v)} g(k)/\#\kappa_0(v)$
4. wiederhole Schritte 2 und 3 bis zur Konvergenz
5. projiziere Ebene n in die Basis $\rightarrow$ SEGMENTATION / DELINEATION



Parameter: R in Schritt 1 und 3;
d(. , .) und $\Pi(v)$ in Schritt 2;
TOP LEVEL n in Schritt 5.

Probleme: Eltern OHNE Kinder
nicht zusammenhängendes rezeptives Feld



I.20 Vorteile von Bildpyramiden

Bildpyramiden:

- reduzieren den Einfluß des Rauschens
- ermöglichen Auflösungsunabhängige Verarbeitung ('Multi-Resolution')
- ermöglichen die Abbildung globaler Merkmale als lokale Merkmale auf höheren Ebenen
- ermöglichen effiziente Berechnung durch DIVIDE & CONQUER:
  - Histogramm
  - "grobe" Merkmalserkennung:
    grobe Operatoren in der Basis $\leftrightarrow$ feine Operatoren auf höheren Ebenen
  - numerische Berechnungen: Anzahl, Summe, Mittelwert, $\sigma \ldots$
- ermöglichen Top-Down Suche:
  1. finde Kanten auf reduzierter Auflösung
  2. suche Verfeinerung in der nächstfeineren Auflösung (Kelly [13])

I.21 Zusammenfassung

- Pyramiden berechnen nichts 'Neues': Glättung und Subsampling.
- Vorteile:
  - Massiv parallele Verarbeitung
  - Globale Operatoren $\longrightarrow$ parallele lokale Operatoren
  - Parallele Rechenkomplexität $\mathcal{O}(\log Durchmesser)$
- Unterschiedliche Arten von Pyramiden: Gaußpyramide, Laplacepyramide, Waveletpyramide
- Keypoint Detektoren: SIFT, SFOP
- Pyramid Linking

Kapitel J

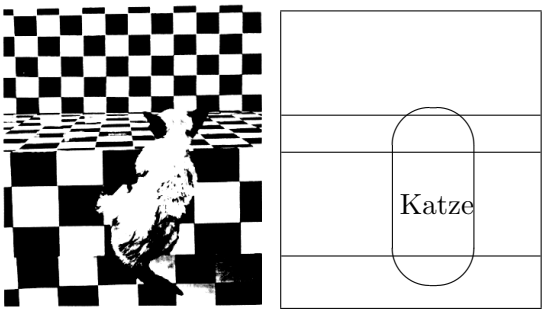
Bildbeschreibung, Eigenschaften

J.1 Bildbeschreibung

Eine Bildbeschreibung hat 3 Bestandteile:

1. Bildteile
2. Eigenschaften der Bildteile
3. Beziehung zwischen den Bildteilen

Bildteile im Bild entsprechen einer Region. Werden Bildteile mittels eines RAGs visualisiert, entsprechen diese den Knoten, deren Eigenschaften den Attributen eines Knotens und die Beziehungen zwischen den Bildteilen den Kanten des Graphens. Kantenattribute stellen eine qualitative oder quantitative Bewertung der Beziehungen dar. Ein Beispiel für eine Bildbeschreibung ist in Abbildung J.1 zu sehen.



(a) Bild einer Katze auf Schachbrettmuster

(b) Beschreibung des Bildes einer Katze auf Schachbrettmuster

Abbildung J.1: Beispiel der Beschreibung eines Bildes

J.1.1 Beschreibung einer Objektform (engl. Shape) [29]

Eine detaillierte Beschreibung von Shapes kann bei

> Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, January 2004

gefunden werden. Zusammenfassend werden Objektformen in konturbasierte (engl. contour-based; siehe Tabelle J.1) und regionenbasierte (engl. region-based; siehe Tabelle J.2) unterteilt.

Tabelle J.1: Konturbasierte Formdeskriptoren

Strukturell	Global
Chain Code	**Umfang**
Polygon	**Kompaktheit**
B-spline	Exzentrizität
Invarianten	Shape Signature
	Hausdorff Distanz
	Fourier Deskriptoren
	Wavelet Descriptoren
	Scale Space
	Autoregression
	Elastisches Matching

Tabelle J.2: Regionenbasierte Formdeskriptoren

Strukturell	Global
Konvexe Hülle	**Fläche**
Medial Achse	**Euler Zahl**
Core	Extentrizität
	Geometrische Momente
	Zernike Momente
	Pseudo-Zernike Momente
	Legendre Momente
	Generic Fourier Descriptor
	Grid Methode
	Shape Matrix

J.2 Bildeigenschaften

Bildeigenschaften dienen zur Erkennung, Identifikation und Klassifikation von Bildern. Man kann zwischen folgenden Eigenschaften unterscheiden (formale Definition siehe Tabelle J.3).

- geometrische Eigenschaft (hängt nicht direkt von den Grauwerten ab)
- lineare Eigenschaft
- lineare und verschiebungsinvariante Eigenschaft
- lineare und begrenzte Eigenschaft (Riesz Theorem)
- lokale Eigenschaft (d.h. wenige Pixel oder kleiner Durchmesser)

- invariante Eigenschaft
- statistische Eigenschaft

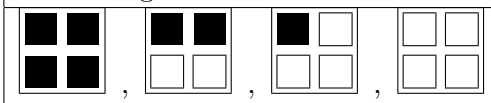
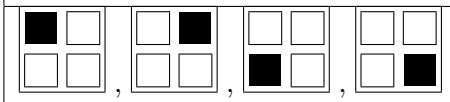
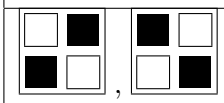
Tabelle J.3: Bildeigenschaften

Begriff	**Formale Definition**
Bild	$B : D \mapsto G, \quad D \subset \mathbb{R}^2$ oder $\mathbb{Z}^2$
Eigenschaft	$\mathcal{P} : B \mapsto \mathbb{R}$ oder $\mathbb{C}$ oder $\mathbb{Z}$
Prädikat	$\mathcal{L} : B \mapsto L = \{\text{true}, \text{false}\} \leftrightarrow \{1, 0\}$
Support Menge	$\Leftrightarrow S_\mathcal{P} \subset B$ von der $\mathcal{P}$ abhängt
geometrische Eigenschaft	$\mathcal{P}[B] \equiv \mathcal{P}[T], \quad T \subset D$
lineare Eigenschaft	$\Leftrightarrow \mathcal{P}[aB_1 + bB_2] = a\mathcal{P}[B_1] + b\mathcal{P}[B_2]$
linear und verschiebungsinvariant	$\Leftrightarrow \exists h : \mathcal{P}[B] = h * B$
linear und begrenzt	$\Leftrightarrow \exists h : \mathcal{P}[B] = \iint h \cdot B$
lokale Eigenschaft	$\lvert S_\mathcal{P} \rvert$ klein

J.2.1 Lokale Eigenschaft

Anhand des Vorkommens bestimmter lokaler Eigenschaften (z.B. Muster) in einem Binärbild, können Rückschlüsse auf das Auftreten bestimmter Strukturen im Bild geschlossen werden (siehe Tabelle J.4).

Tabelle J.4: Lokale Eigenschaften in Binärbildern

Lokale Eigenschaften	Vorkommen und Struktur im Bild
 $+90^0$ Rotationen	Wenn dies die einzigen 2x2 auftretenden Muster sind, enthält das Binärbild nur achsparalle separierte Rechtecke
	Wenn diese 2x2 Muster je einmal auftreten, enthält das Bild genau 1 achsparalleles Rechteck
	Wenn diese 2x2 Muster nicht in einem Bild vorkommen, wird dieses als wohlgeformt (siehe G.5 Seite 87) bezeichnet, da das 4-/8-Paradoxon nicht auftreten kann.

J.2.2 Invariante Eigenschaften

Zum Verständis der invarianten Eigenschaften wird im folgenden Abschnitt der Begriff der Invarianz und Normalisierung definiert.

Invarianz, Normalisierung

Invarianz erlaubt die Bestimmung der Eigenschaft vor oder nach einer Bildoperation.

- Sei $O : B \mapsto B$ eine Bildoperation.
  Dann ist die Eigenschaft $\mathcal{P}$ **invariant** bezüglich O, wenn $\mathcal{P}[O[B]] \equiv \mathcal{P}[B]$, d.h. die Eigenschaft $\mathcal{P}$ verändert sich durch O nicht.
- Eine Eigenschaft kann auch bezüglich einer ganzen Menge von Operationen $O \in \mathcal{S}$ invariant sein.
- ein normalisiertes Bild $\mathcal{N}[B]$, $\mathcal{N} : B \mapsto B$ ist invariant bez. aller Operationen $O \in \mathcal{S}$: $\mathcal{N}[O[B]] \equiv \mathcal{N}[B]$.

Beispiel monotone Grauwerttransformationen

Als Bildoperation O sei eine Normalisierung des Bildes folgend gegeben: O $\boxed{\mathcal{N}[B] = \frac{B-\mu[B]}{\sigma[B]}}$.
Das normalisiertes Bild $\mathcal{N}[B]$ hat daher laut Definition der Bildoperation O in diesem Beispiel den Mittelwert $\mu[\mathcal{N}[B]] = 0$ und die Varianz $\sigma[\mathcal{N}[B]] = 1$.

Beispiel normierte Filter

Gegeben sei ein normierter Filter $W = (w_{ij}) : \sum_{ij} w_{ij} = 1$

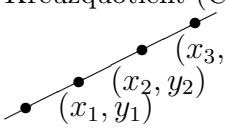
$$\boxed{G = W * B \Longrightarrow \sum_{ij} g_{ij} = \sum_{ij} b_{ij} \text{ bei zyklischem Abschluss}}$$

Beweis:

$$\begin{aligned}
\sum_{ij} g_{ij} &= \sum_{ij}\sum_{kl} w_{kl} b_{i+k,j+l} && \text{Vertauschen der Summen} \\
&= \sum_{kl}\sum_{ij} w_{kl} b_{i+k,j+l} && \text{Gewichte vor die ij-Summe} \\
&= \sum_{kl} w_{kl} \sum_{ij} b_{i+k,j+l} && \text{zyklischer Abschluss: Summe von Permutation unabhängig} \\
&= \sum_{kl} w_{kl} \sum_{ij} b_{i,j} && \text{Summen unabhängig} \\
&= \left(\sum_{kl} w_{kl}\right)\left(\sum_{ij} b_{i,j}\right) = \sum_{ij} b_{i,j}
\end{aligned}$$

In Tabelle J.5 sind fünf invariante Eigenschaften und deren Bezug aufgelistet.

Tabelle J.5: Invariante Bildeigenschaften

	Eigenschaft	**ist invariant bezüglich**
1.	Anstieg, Neigung (Slope), Chaincode	Verschiebung
2.	Fläche, Umfang, Distanz, Dicke, Krümmung	... und Rotation
3.	Quotienten aus Größen, Winkeln, Formeigenschaften (Länglichkeit, Konvexität)	... und Vergrösserung
4.	Kreuzquotient (Cross ratio)  $CR = \frac{(x_3-x_1)(x_4-x_2)}{(x_3-x_2)(x_4-x_1)}$	perspektive Projektion
5.	topologische Eigenschaften (Anzahl Komponenten, Löcher, Eulerzahl)	'Rubber Sheet' geom. Verzerrungen

J.2.3 Statistische Eigenschaften

In diesem Abschnitt werden Projektionen, Momente, etc. definiert und deren Anwendung als statistische Eigenschaft aufgezeigt. Die Definition der Eigenschaften sind in Tabelle J.6 zusam-

mengefasst dargestellt.

Tabelle J.6: Zusammenfassung der Definition für Moment, zentraler Moment, Schwerpunkt

Moment m_{ij} der Ordnung i, j	$m_{ij} = \sum_x \sum_y x^i y^j B(x, y)$
Moment m_{00} (z.B. Fläche einer Region) $\frac{m_{10}}{m_{01}}$	invariant bez. Rotation um Ursprung invariant bez. Skalierung
Schwerpunkt S(S_x,S_y)	$S = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right)$
Normalisierung bez. Verschiebung	$B_N = \mathcal{N}[B] = B(x - S_x(B), y - S_y(B))$
Zentrale Momente $\overline{m_{ij}}$ Daher: $\overline{m_{00}} = m_{00}$	$(i, j)-$ Momente von B_N $\overline{m_{10}} = \overline{m_{01}} = 0$

Momente

Für Punktverteilungen kann man mittels statistischer Analyse die **zentralen Momente** der Verteilung berechnen. Diese geben Auskunft über die charakteristischen Eigenschaften der Mittelpunkte bzw. der Schwerpunkte der Verteilung. Das gewöhnliche **Moment $\mathbf{m}_{ij}$ der Ordnung i,j** für ein Grauwertbild (oder Region eines Grauwertbildes) B, welches durch eine diskrete Bildfunktion B(x,y) beschrieben werden kann, wird folgend definiert:

$$m_{ij} = \sum_x \sum_y x^i y^j B(x, y)$$

Schwerpunkt

Ein spezieller Fall der Momente bildet der Schwerpunkt S =(S_x,S_y) einer binären Region B, der durch den arithmetischen Mittelwert der Koordinaten in x- und y- Richtung berechnet werden kann. Die Fläche einer Region $|B|$ kann durch das Moment m_{00} der Ordnung i=0, j=0 berechnet werden und ist invariant bzgl. der Rotation um den Ursprung.

$$S_x = \frac{1}{|B|} * \sum_x \sum_y x^1 y^0 B(x, y) = \frac{1}{m_{00}} * m_{10}$$

$$S_y = \frac{1}{|B|} * \sum_x \sum_y x^0 y^1 B(x, y) = \frac{1}{m_{00}} * m_{01}$$

Zentrale Moment

Der **zentrale Moment** unterscheidet sich vom gewöhnlichen Moment dahingehend, dass die Punktverteilung anhand derer er berechnet wird, durch den arithmetischen Mittelwert $\mu = (\mu_x, \mu_y)$ normalisiert wird. Da der Schwerpunkt den arithmetischen Mittelwert darstellt, können wir die behandelte Punktverteilung durch den Schwerpunkt normalisieren, d.h. der Ursprung des Koordinatensystems wird an den Schwerpunkt verschoben. Mit B_N wird das/die normalisierte Bild/Region bezeichnet.

$$B_N = \mathcal{N}[B] = B(x - \mu_x, y - \mu_y) = B(x - S_x[B], y - S_y[B]) = B(\overline{x}, \overline{y})$$

Der zentrale Moment $\overline{m}_{ij}$ der Ordnung i,j, lässt sich folgend beschreiben:

$$\overline{m}_{ij} = \sum_x \sum_y (x - S_x[B])^i (y - S_y[B])^j B(x, y) = \sum_x \sum_y \overline{x}^i \overline{y}^j B(x, y)$$

Durch Verwendung des Schwerpunktes als Referenz, können Merkmale von Regionen unabhängig von deren Lage (invariant) berechnet werden. Das Verschieben einer Regionen verändert den Bezugspunkt, aber die Fläche derer nicht, daher kann $\overline{m}_{00}$ gleich m_{00} gesetzt werden.

Projektion

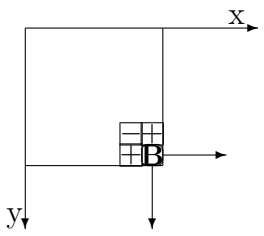
Die statistische Eigenschaft *Projektion* ist eine eindimensionale Abbildung von zweidimensionalen Bilddaten parallel zu den Koordinatenachsen. Daraus ergeben sich x-, y- Projektionen und Projektionen entlang einer Kurvenschar. In Tabelle J.7 sind diese näher definiert. Index H steht hier für horizontale Projektion und V für vertikale Projektion. Zum Beispiel wird durch die Aufsummierung der Pixelwerte der **Bildzeile** x die **horizontale** Projektion der Länge N gebildet und umgekehrt wird durch Aufsummierung der Pixelwerte der **Bildspalte** y die **vertikale** Projektion der Länge M gebildet. Allgemein können Projektion entlang jeder Geraden beliebiger Orientierung gemacht werden, z.B. entlang einer Hauptachse einer gerichteten Bildregion oder entlang einer Kurvenschar.

x-Projektion	$P_H(y) = \sum_x B(x,y)$ für $0 \le y \le N$
y-Projektion	$P_V(x) = \sum_y B(x,y)$ für $0 \le x \le M$
entlang Kurvenschar $(x(s,t), y(s,t))$	$P_K(s) = \sum_t B(x(s,t), y(s,t))$
Hauptachse $y = x \tan\theta$	$\Leftrightarrow \sum_x \sum_y (x\sin\theta - y\cos\theta)^2 B_N(x,y) \longrightarrow$ MIN
Ableitung nach $\theta = 0$ liefert	$\tan^2\theta + \frac{\overline{m}_{20} - \overline{m}_{02}}{\overline{m}_{11}} \tan\theta - 1 = 0$

Tabelle J.7: Definition von Projektionen

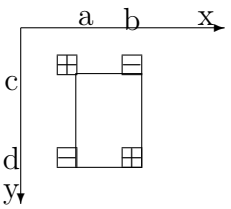
J.2.4 Integral Image

- Ziel: Summe der Grauwerte $B(x,y)$ in beliebigem Bildfenster $[a,b] \times [c,d]$
- Zwischenschritt: Bestimmung des Integral Image $I(x,y) := \sum_{i=0}^{x} \sum_{j=0}^{y} B(i,j)$



$I(0,0) = B(0,0)$
$I(x+1,0) = B(x+1,0) + I(x,0)$
$I(x+1,y+1) = B(x+1,y+1) + I(x+1,y) + I(x,y+1) - I(x,y)$
$y = 0,1,2,...; x = 0,1,...$

- Ergebnis mit 4 Zugriffen auf $I(x,y)$:



$$\sum_{x=a}^{b} \sum_{y=c}^{d} B(x,y) = I(b,d) - I(a,d) - I(b,c) + I(a,c)$$

J.2.5 Co-occurence Matrix

Die Co-occurence Matrix, bestimmt die Häufigkeit des gleichzeitigen Auftretens verschiedener Werte. Für deren Bildung werden ein kombiniertes Histogramm und ein zusätzlicher Verschiebungsvektor benötigt, die im folgenden Abschnitt erklärt werden:
Zur Wiederholung: Anhand eines Histogrammes $h_B(g)$ eines Bildes B mit Grauwerten g, können

folgende Parameter bestimmt werden:

Anzahl der Pixel: $n = \sum_g h_B(g)$

Mittelwert μ: $\mu[B] \equiv \frac{1}{n} \sum_g g h_B(g)$

Varianz $\sigma^2[B] \equiv \frac{1}{n} \sum_g (g - \mu[B])^2 h_B(g) = \frac{1}{n} \sum_g g^2 h_B(g) - \mu^2[B]$,

Standardabweichung $\sigma[B]$

Kombinierte Histogramme M(g_i, g_j) stellen die Anzahl der Grauwerte g_i und g_j jener Pixel P_i und P_j dar, die in einer definierten Relation bzw. Beziehung zueinander stehen. Beispiele für Beziehungen sind:

g_j ist der rechte Nachbar von g_i
g_j ist der zweite linke Nachbar von g_i
g_j liegt unterhalb von g_i
usw. ...

Die Beziehung zwischen den Pixeln (auch Verschiebung genannt) kann durch die Entfernung Δx in x-Richtung und Δy in y-Richtung von P_i zu P_j mit dem Verschiebungsvektor $\delta = (\Delta x, \Delta y)$ definiert werden und ist in Abbildung J.2 dargestellt. Die Menge der möglichen betrachteten Pixel

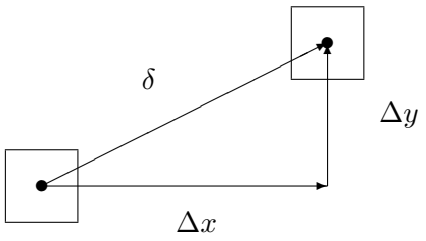


Abbildung J.2: Verschiebung δ eines Pixels

kann abhängig von der ausgewählten Beziehung (d.h. die Definition des Vektors δ) variieren und ist folgend definiert:

$$N_\delta = \{(x, y) \in D | (x + \Delta x, y + \Delta y) \in D\}$$

Das kombinierte Histogramm für die Verschiebung δ kann dann folgend definiert werden:

$$M_\delta(g_i, g_j) = |\{(x, y) \in N_\delta | B(x, y) = g_i, B(x + \Delta x, y + \Delta y) = g_j\}|$$

Die Co-occurence Matrix $P_\delta[B]$ kann dann folgend gebildet werden:

$$P_\delta[B] = M_\delta / |N_\delta|$$

Unter der Verwendung einer Co-occurence Matrix kann Textur erkannt und beschrieben werden. Das **Brodatz Album**, welches in Abbildung J.3 zu sehen ist, kann als Beispiel für erkennbare Strukturen gesehen werden. Texturen können klassifiziert werden, indem die Maße bzw. Summen über die Co-occurence Matrix $P_\delta[B]$ gebildet werden, d.h. durch die Berechnung der Energie bzw.

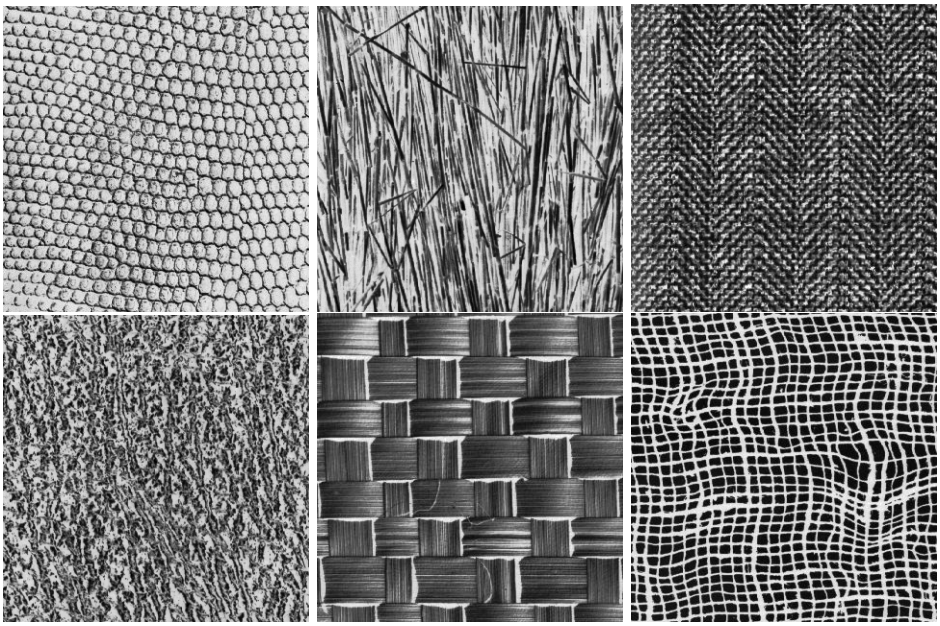


Abbildung J.3: Texturbeispiel Brodatz-Album

Entropie:

$$\text{Energie:} \sum_{\delta} P_\delta[B]$$

$$\text{Entropie:} \sum_{\delta} P_\delta[B] \log P_\delta[B]$$

Die Charakterisierung einer Textur kann anhand der Berechnung der Autokorrelation erfolgen:

$$R_B(\delta) = \mu(B(x,y) \cdot B(x + \Delta x, y + \Delta y))$$

$R_B((0,0))$ stellt ein Maximum dar. Der Abfall der Werte mit steigendem Abstand von $(0,0)$ charakterisiert die Textur.

J.3 Zusammenfassung

- Ein Bild wird durch seine Bildteile und deren Beziehungen beschrieben.
- Bildteile werden durch ihre Eigenschaften bestimmt.
- Eigenschaften sind Funktionen, die linear und lokal sein können.
- Invarianz erlaubt die Bestimmung der Eigenschaft vor oder nach einer Operation.
- Statistische Eigenschaften: Projektion, Momente, Schwerpunkt
- Textur kann durch Co-occurence Matrizen erkannt und beschrieben werden.

Anhang A

Sechs wissenschaftlichen Beiträge [12, 14, 24, 25, 27, 29] können über die Webseite der Vorlesung

http://prip.tuwien.ac.at/teaching/186822

heruntergeladen werden. Ihre Lektüre sollte mit den Inhalten der Lehrveranstaltung zu verstehen sein. Sie dient daher auch sehr gut als Vorbereitung auf die Prüfung.

Index

Abkürzungen

CCL Connected Component Labeling

CV Computer Vision

DIBAG Digitale Bild Auswertung Graz

DOG Difference of Gaussians

FCC Freeman Chain Code

LISP List Processing, Listen-Verarbeitung

LUT Lookup Table

NF Normalform

PSF Punktstreufunktion, engl. point spread function

RAG Region Adjacency Graph

SFOP Scale-invariant Feature Operator

SIFT Scale-Invariant Feature Transform

SNN Symmetric Nearest Neighbour

SNR Signal/Rauschverhältnis, engl. signal noise ratio

TR Technischer Report

Literaturverzeichnis

[1] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, April 1983.

[2] P. J. Burt. Fast filter transform for image processing. *Computer Graphics and Image Processing*, 16(1):20–51, May 1981.

[3] P. J. Burt, T.-H. Hong, and Azriel Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11(No.12):pp.802–809, December 1981.

[4] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, nov 1986.

[5] J. L. Crowley and A. Parker. A representation of shape based on peaks and ridges in the difference of low–pass transform. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI–6:pp.156–170, 1984.

[6] Stanley R Deans. Hough transform from the radon transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(2):185–188, March 1981.

[7] C. Fermüller and H. Malm. Uncertainty in visual processes predicts geometrical optical illusions. *Vision Research*, 44(7):727–749, March 2004.

[8] W. Förstner. Detecting interpretable and accurate scale-invariant keypoints. *2009 IEEE 12th International Conference on Computer Vision*, pages 2256–2263, September 2009.

[9] R. L. Hartley. *Multi–Scale Models in Image Analysis.* PhD thesis, University of Maryland, Computer Science Center, 1984.

[10] G. Hartmann. Principles and strategies of hierarchical contour coding. In *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 1087–1089, Montreal, Canada, 1984.

[11] T.-H. Hong. *Pyramid Methods in Image Analysis.* PhD thesis, University of Maryland, Computer Science Center, 1982.

[12] Rangachar Kasturi, Lawrence OGorman, and Venu Govindaraju. Document image analysis: A primer. *Sadhana*, 27:3–22, February 2002.

[13] M. D. Kelly. Edge detection in pictures by computer using planning. In B. Meltzer and D. Michie, editors, *Machine Intelligence 6*, pages 397–409, Edinburgh, Scotland, 1971. Edinburgh University Press.

[14] Yakov Keselman and Sven Dickinson. Generic model abstraction from examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1141–1156, 2005.

[15] J. Koenderink, K. Huys, and L. Toet. Resolution dependent structure of images. *Pre-print*, 1984.

[16] L. Latecki, U. Eckhardt, and Rosenfeld A. Well-composed sets. *Computer Vision and Image Understanding*, 61:70–83, 1995.

[17] D.G. Lowe. Object recognition from local scale-invariant features. *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2:1150–1157, September 1999.

[18] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, January 2004.

[19] Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI–11(No. 7):pp. 674–693, July 1989.

[20] P. F. M. Nacken. Image segmentation by connectivity preserving relinking in hierarchical graph structures. *Pattern Recognition*, 28(6):907–920, June 1995.

[21] Frank Pedrotti, Leno Pedrotti, Werner Bausch, and Hartmut Schmidt. Erzeugung und messung von licht. In *Optik für Ingenieure*, pages 9–39. Springer Berlin Heidelberg, 2005.

[22] Lawrence G. Roberts. Machine perception of three-dimensional solids. http://dspace.mit.edu/handle/1721.1/11589?show=full, 1963. Thesis (Ph. D.)–Massachusetts Institute of Technology, Dept. of Electrical Engineering, 1963.

[23] Azriel Rosenfeld. Image analysis and computer vision: Motives, methods, and milestones, 1955-1979. Technical Report CAR-TR-892, CS-TR-3920R, University of Maryland, Center for Automation Research, Computer Science Center, july 1998.

[24] Azriel Rosenfeld. Image analysis and computer vision: Motives, methods, and milestones, 1955-1979. *Computer Vision and Image Understanding*, 74(1):36–95, April 1999.

[25] Paul L. Rosin. Measuring the orientability of shapes. In Walter G. Kropatsch, Martin Kampel, and Allan Hanbury, editors, *Computer Analysis of Images and Patterns*, volume 4673, pages 620–627. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[26] M. Shneier. Two hierarchical linear feature representations: Edge pyramids and edge quadtrees. *Computer Graphics and Image Processing*, Vol. 17:pp.221–224, 1981.

[27] Kaleem Siddiqi, Ali Shokoufandeh, Sven J. Dickinson, and Steven W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999. 10.1023/A:1008102926703.

[28] A. Toet. Multiscale color image enhancement. *Pattern Recognition Letters*, 13(3):167–174, March 1992.

[29] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, January 2004.