Dreidimensionale Bildverarbeitung

Sommersemester 2020

Dr. Christina Gillmann [1]

Institut für Informatik
Universität leipzig

Inhalt

Teil I

Einführung

> *„The term digital image processing generally refers to processing of a two-dimensional picture by a digital computer. In a broader context, it implies digital processing of any two-dimensional data.“*

[Jain, Fundamentals of Digital Image Processing, Prentice Hall, Englewood Cliffs, NJ, USA, 1989]

> *„The term digital image processing generally refers to processing of a two-dimensional picture by a digital computer. In a broader context, it implies digital processing of any two-dimensional data.“*

[Jain, Fundamentals of Digital Image Processing, Prentice Hall, Englewood Cliffs, NJ, USA, 1989]

> *„A three-dimensional digital image is a 3-D array whose elements (called volume elements or voxels) represent samples of a certain physical quantity over a usually rectangular 3-D grid.“*

[Nikolaidis, Pitas, 3-D Image Processing Algorithms, John Wiley & Sons, New York, NY, USA, 2001]

Abgrenzung

- Computergrafik:
  - Enger Zusammenhang zu Image-based Rendering und Point-based Rendering
  - Erklärung des Aliasing-Effekts durch Abtasttheorem
  - Geometriedaten aus Laserabtastung und Stereobildverarbeitung

Abgrenzung

- Computergrafik:
  - Enger Zusammenhang zu Image-based Rendering und Point-based Rendering
  - Erklärung des Aliasing-Effekts durch Abtasttheorem
  - Geometriedaten aus Laserabtastung und Stereobildverarbeitung
- Computer Vision:
  - Vorverarbeitung der Daten
  - Erkennung von Objektgrenzen/Objekträndern

Abgrenzung

- Computergrafik:
  - Enger Zusammenhang zu Image-based Rendering und Point-based Rendering
  - Erklärung des Aliasing-Effekts durch Abtasttheorem
  - Geometriedaten aus Laserabtastung und Stereobildverarbeitung
- Computer Vision:
  - Vorverarbeitung der Daten
  - Erkennung von Objektgrenzen/Objekträndern
- Visualisierung:
  - Hintergründe und Ideen zu texturbasierten Visualisierungsverfahren
  - Aufbereitung von Daten aus CT, MRT, Röntgenbildern, Laserabtastung, konfokale Mikroskopie
  - Enge Verknüpfung mit Merkmalsextraktion und Segmentierung in der Medizin oder in Strömungsdaten

Beziehung zur Computergrafik

- Die **Computergrafik** erzeugt vor allem photorealistische Bilder aus dreidimensionalen Szenen.
- Die **Bildverarbeitung**
  - rekonstruiert insbesondere dreidimensionale Szenen aus Kameraaufnahmen
  - kann Ausgangsdaten für Computergrafiken - etwa Texturen oder Geometrie - liefern

Beziehung zur Computergrafik

- Die **Computergrafik** erzeugt vor allem photorealistische Bilder aus dreidimensionalen Szenen.
- Die **Bildverarbeitung**
  - rekonstruiert insbesondere dreidimensionale Szenen aus Kameraaufnahmen
  - kann Ausgangsdaten für Computergrafiken - etwa Texturen oder Geometrie - liefern
- In diesem Sinne sind Bildverarbeitung und Computergrafik reziproke Ansätze.

Beziehung zur Computergrafik

- Die **Computergrafik** erzeugt vor allem photorealistische Bilder aus dreidimensionalen Szenen.
- Die **Bildverarbeitung**
  - rekonstruiert insbesondere dreidimensionale Szenen aus Kameraaufnahmen
  - kann Ausgangsdaten für Computergrafiken - etwa Texturen oder Geometrie - liefern
- In diesem Sinne sind Bildverarbeitung und Computergrafik reziproke Ansätze.
- Eine Integration von Bildverarbeitung und Computergrafik wird von vielen Experten angestrebt.
- Das gesamte Gebiet wird auch als Visual Computing bezeichnet und umfasst u.a.
  - Geometrisches Modellieren
  - Virtual Reality
  - Computer Vision
  - Visualisierung

Beziehung zur Visualisierung

- Es gibt eine starke Überlappung von
  - **dreidimensionaler Bildverarbeitung**,
  - **medizinischer Bildverarbeitung**,
  - **medizinischer Visualisierung** und
  - **Volumenvisualisierung**.
- Im Rahmen der dreidimensionalen Bildverarbeitung wird die Herkunft vieler Ideen und Verfahren aus der zweidimensionalen Bildverarbeitung besonders deutlich
  - Abtastung
  - Segmentierung
  - Filterung

Beziehung zur Visualisierung

- Es gibt eine starke Überlappung von
  - **dreidimensionaler Bildverarbeitung**,
  - **medizinischer Bildverarbeitung**,
  - **medizinischer Visualisierung** und
  - **Volumenvisualisierung**.
- Im Rahmen der dreidimensionalen Bildverarbeitung wird die Herkunft vieler Ideen und Verfahren aus der zweidimensionalen Bildverarbeitung besonders deutlich
  - Abtastung
  - Segmentierung
  - Filterung
- Die Verfahren zur **Datenvorverarbeitung** (Filterungsphase der Visualisierungspipeline) stammen aus der Bildverarbeitung
  - Glättungsfilter
  - Geometrische Korrektur von Verzerrungen
  - Kontrasterhöhung
- Beiden Disziplinen ist die Bedeutung des menschlichen visuellen Systems als finalem Verarbeitungsprozessor der Ergebnisse und beständiger Quelle neuer Ideen und Herausforderungen gemeinsam.

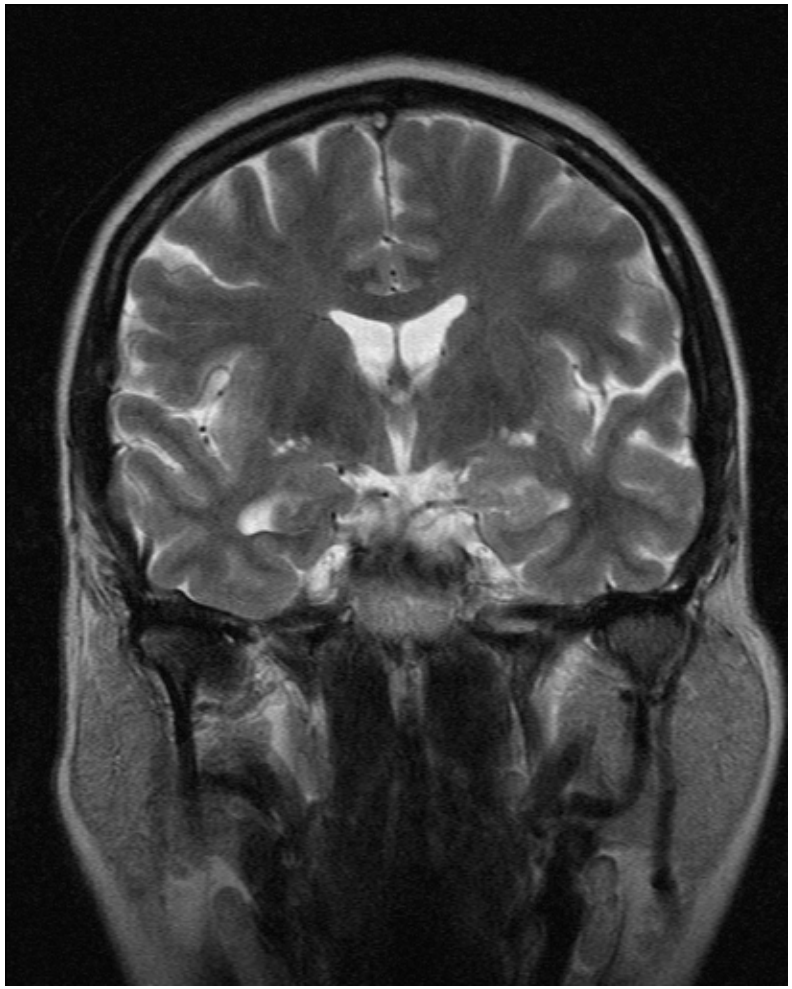


Abbildung: Medizinische Magnetresonanztomographieaufnahme. Schnitt durch ein 3D Bild in Graustufen eingefärbt.

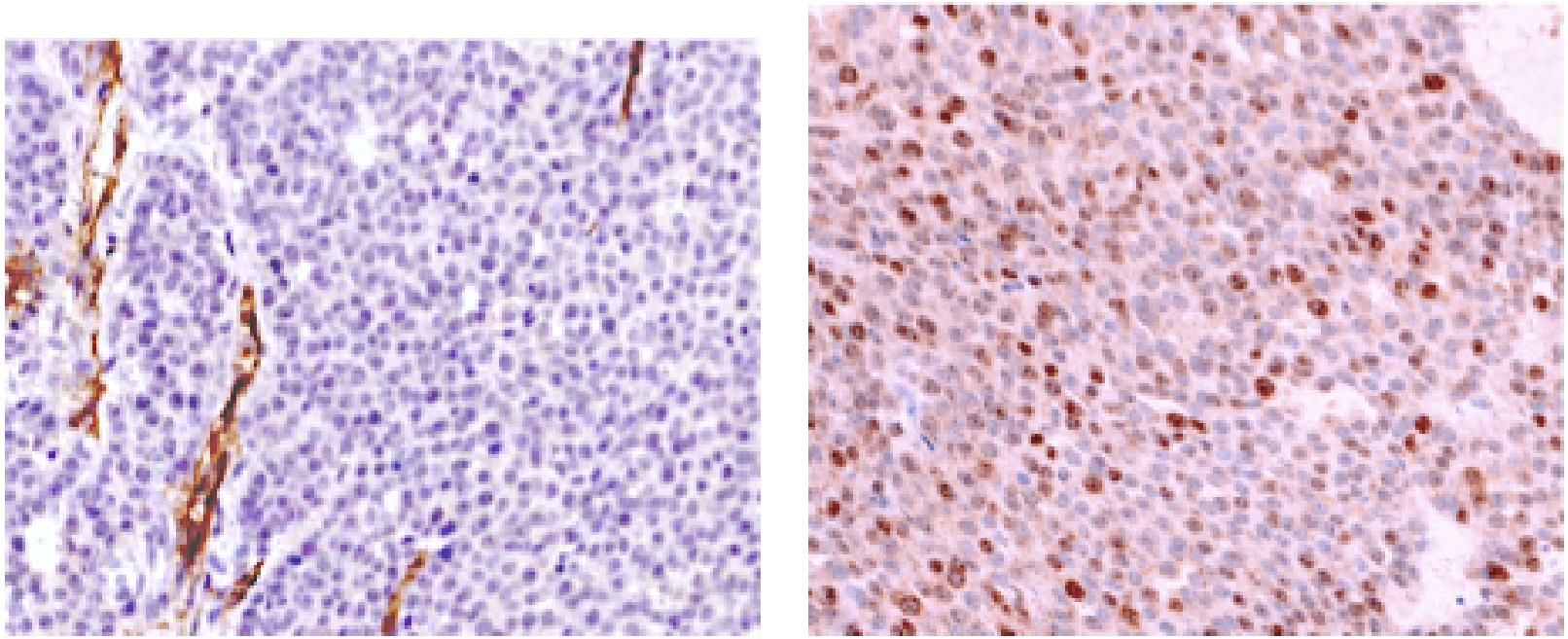


Abbildung: Mikroskopieaufnahme von Zellschichten. Etwas dunkler sind die geschnittenen Gefäße sichtbar. Eine Fragestellung: Wie kann man automatisiert die Durchblutung des Gewebes quantisieren?

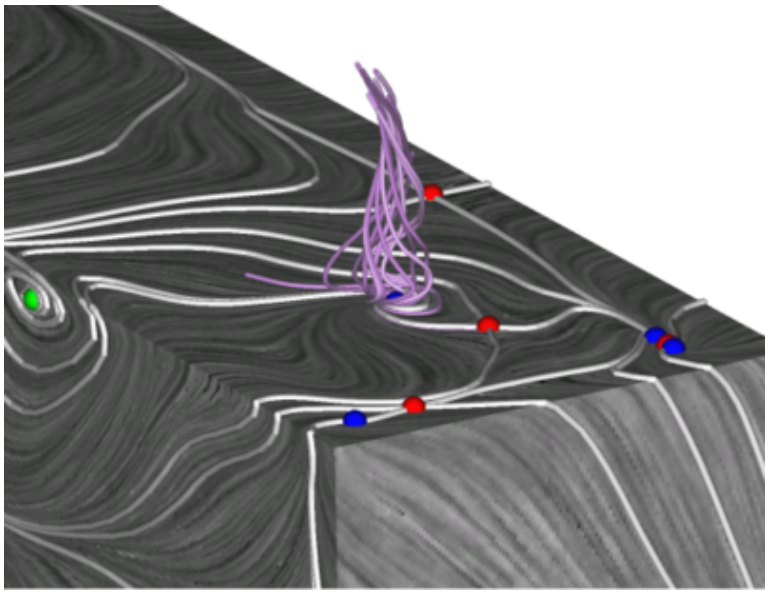


Abbildung: Die Linienintegralfaltung (Line Integral Convolution, LIC) ist ein Beispiel, bei dem Bildverarbeitungsoperatoren in der Visualisierung verwendet werden. Hier wird mittels Verschmieren einer Rauschtextur ein Strömungsfeld dargestellt. Zusätzlich werden die wesentlichen Strukturen über Punkte und Linien hervorgehoben.
[Cabral, Leedom: Imaging vector fields using line integral convolution, ACM SIGGRAPH, 1993]
[Stalling, Hege: Fast and resolution independent line integral convolution, ACM SIGGRAPH, 1995]

Bildverarbeitungsprozess

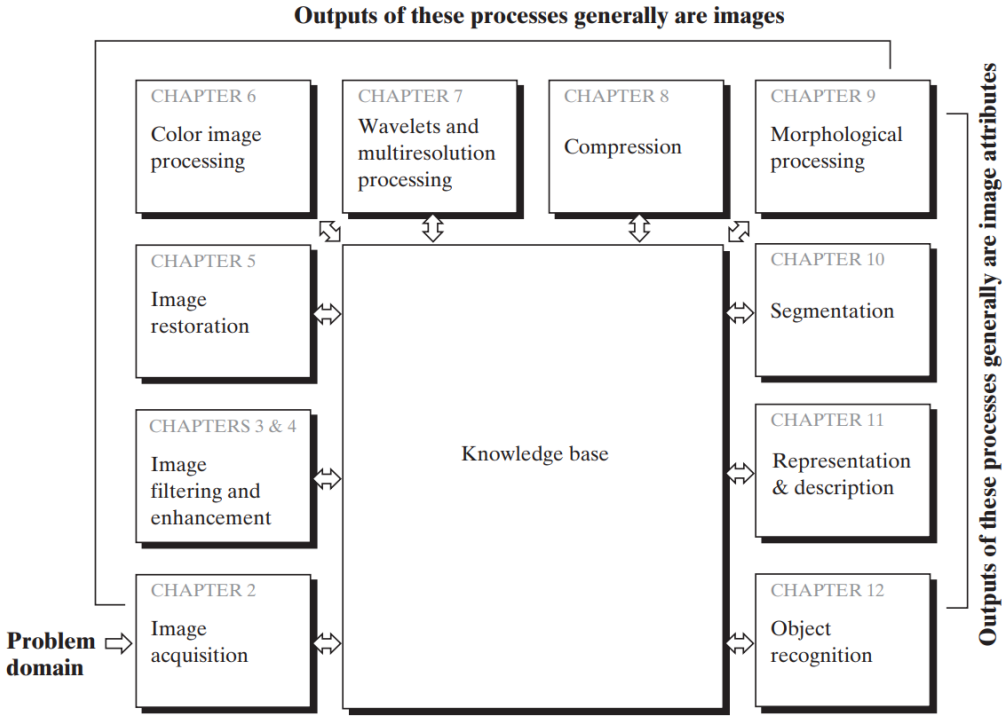


Abbildung: Die Bildverarbeitung umfasst eine reihe von Algorithmen, die entweder wiederum Bilder oder Bildattribute ausgeben. [Digital Image Processing, 3rd Edition, Gonzales and Woods]

Teil II

Bildrepräsentation

Bilder

- **Bilder** sind skalare Funktionen auf diskreten Rastern, die an jeder Stelle Abtastwerte einer physikalischen Größe als Mittel über einen kleinen Raumbereich beschreiben.

Bilder

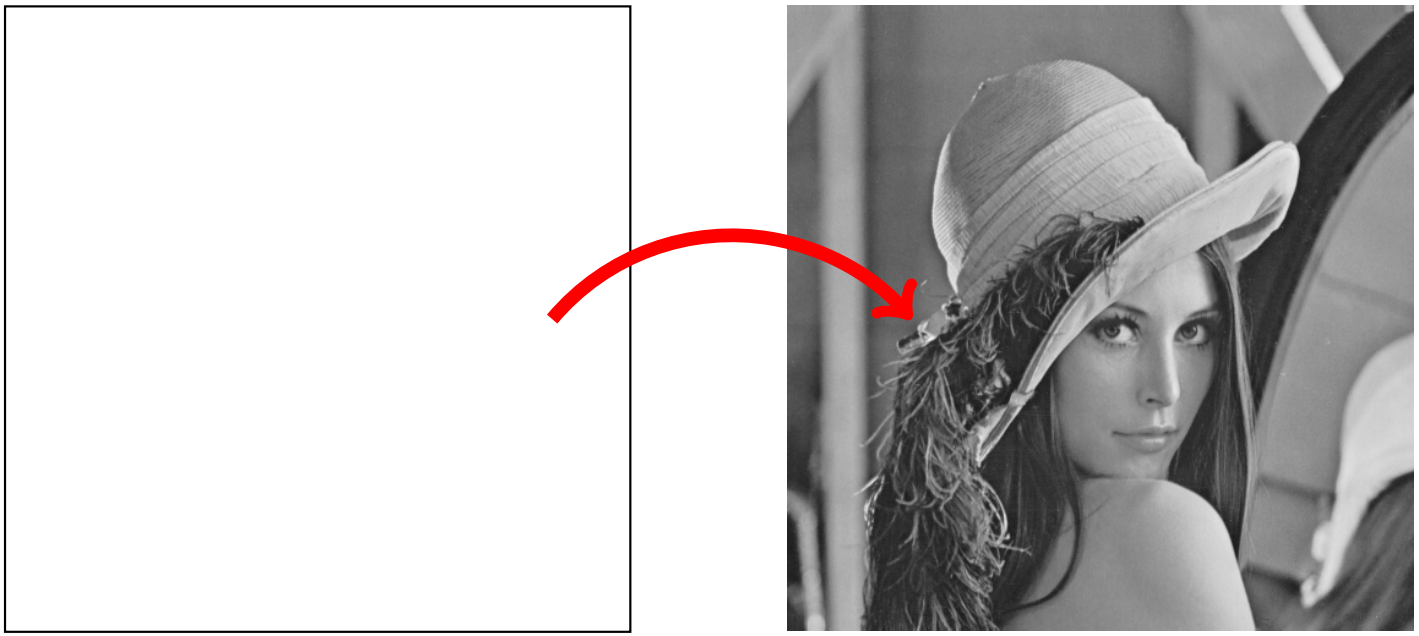
- **Bilder** sind skalare Funktionen auf diskreten Rastern, die an jeder Stelle Abtastwerte einer physikalischen Größe als Mittel über einen kleinen Raumbereich beschreiben.
- Ein **2D-Bild** kann aus Grauwerten über einem 2D-Punktraster bestehen, bei dem jeder Grauwert jene Intensität angibt, die zum Beispiel zum Zeitpunkt der Linsenöffnung einer Digitalkamera auf einen Photorezeptor getroffen ist.

Bilder

- **Bilder** sind skalare Funktionen auf diskreten Rastern, die an jeder Stelle Abtastwerte einer physikalischen Größe als Mittel über einen kleinen Raumbereich beschreiben.
- Ein **2D-Bild** kann aus Grauwerten über einem 2D-Punktraster bestehen, bei dem jeder Grauwert jene Intensität angibt, die zum Beispiel zum Zeitpunkt der Linsenöffnung einer Digitalkamera auf einen Photorezeptor getroffen ist.
- Ein **3D-Bild/Volumenbild** kann etwa aus Skalarwerten über einem 3D-Punktraster bestehen, bei dem jeder Wert die Dichte des Volumens angibt.
  - Computertomographie: die Werte geben die Absorption von Röntgenstrahlen an
  - Magnetresonanztomographiebilder: die gemittelte Abnahme der Spinsynchronisation von Wasserstoffatomen während einer MRI-Aufnahme (magnetic resonance imaging - Kernspintomographie) wird wiedergegeben

Als **analoges n-dimensionales Bild** bezeichnen wir ein n-dimensionales, kontinuierliches, reellwertiges Signal über einem n-dimensionalen Rechteck $R_n \subset \mathbb{R}^n$

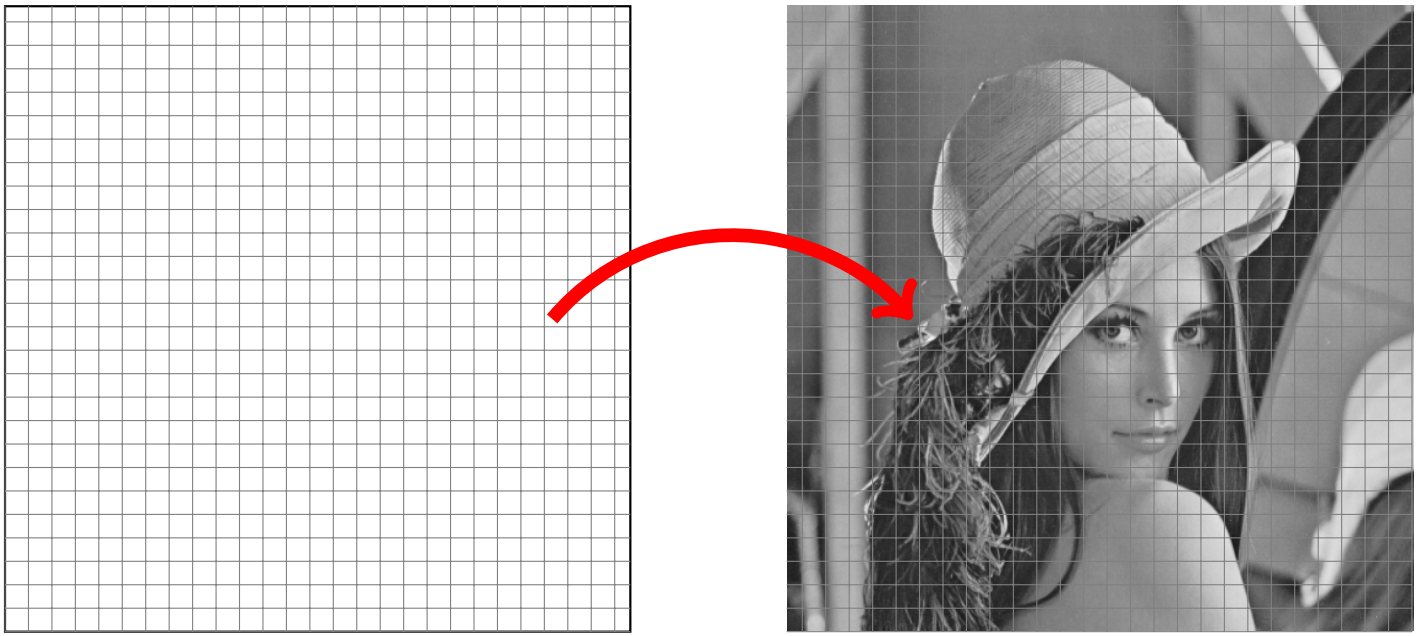
$$\begin{array}{rcl} R_n & \rightarrow & \mathbb{R} \\ x & \mapsto & g(x) \end{array}$$



Als n-dimensionales, **digitales Bild** bezeichnen wir ein diskretes, skalares Signal über einem Rechteckgitter mit Punktabständen $\Delta x_0, \ldots, \Delta x_n$ mit Werten einer in der Regel endlichen Wertemenge W.

Als n-dimensionales, **digitales Bild** bezeichnen wir ein diskretes, skalares Signal über einem Rechteckgitter mit Punktabständen $\Delta x_0, \ldots, \Delta x_n$ mit Werten einer in der Regel endlichen Wertemenge W.
Für Computerbilder ist hier die Auflösung oft durch die Quantisierung bestimmt (z.B. 8 Bit Quantisierung, Werte von 0 bis 255), wobei in der Theorie auch $\mathbb{R}$ und die komplexe Ebene $\mathbb{C}$ als Wertebereich sinnvoll sind.

Als n-dimensionales, **digitales Bild** bezeichnen wir ein diskretes, skalares Signal über einem Rechteckgitter mit Punktabständen $\Delta x_0, \ldots, \Delta x_n$ mit Werten einer in der Regel endlichen Wertemenge W.
Für Computerbilder ist hier die Auflösung oft durch die Quantisierung bestimmt (z.B. 8 Bit Quantisierung, Werte von 0 bis 255), wobei in der Theorie auch $\mathbb{R}$ und die komplexe Ebene $\mathbb{C}$ als Wertebereich sinnvoll sind.



Im Sinne der Signalverarbeitung handelt es sich bei einem digitalen Bild also um ein Signal mit den Eigenschaften

- **diskret**:
  - räumlich abgetastet
  - diskreter Definitionsbereich
- **quantisiert**: diskreter Wertebereich

Im Sinne der Signalverarbeitung handelt es sich bei einem digitalen Bild also um ein Signal mit den Eigenschaften

- **diskret**:
  - räumlich abgetastet
  - diskreter Definitionsbereich
- **quantisiert**: diskreter Wertebereich

Ein n-dimensionales, digitales Bild ist dann definiert als

$$p: \quad \begin{array}{l} \{0, \Delta x_0, 2 \cdot \Delta x_0, \ldots, (N_0 - 1) \cdot \Delta x_0\} \quad \times \\ \ldots \quad \times \\ \{0, \Delta x_n, 2 \cdot \Delta x_n, \ldots, (N_n - 1) \cdot \Delta x_n\} \end{array} \quad \rightarrow \quad W$$

Anstelle der Funktionsschreibweise wird bei digitalen Bildern oft die abkürzende Indexnotation verwendet, so schreibt man $p_{ij} := p(r_{ij})$, wobei $r_{ij} := \begin{pmatrix} i \cdot \Delta x_0 \\ j \cdot \Delta x_1 \end{pmatrix}$

Anstelle der Funktionsschreibweise wird bei digitalen Bildern oft die abkürzende Indexnotation verwendet, so schreibt man $p_{ij} := p(r_{ij})$, wobei $r_{ij} := \begin{pmatrix} i \cdot \Delta x_0 \\ j \cdot \Delta x_1 \end{pmatrix}$

Ein zweidimensionales Bild lässt sich somit vollständig durch das Tupel

$$P = (N_1, N_2, \Delta x_0, \Delta x_1, W, (p_{ij}))$$

beschreiben.

Anstelle der Funktionsschreibweise wird bei digitalen Bildern oft die abkürzende Indexnotation verwendet, so schreibt man $p_{ij} := p(r_{ij})$, wobei $r_{ij} := \begin{pmatrix} i \cdot \Delta x_0 \\ j \cdot \Delta x_1 \end{pmatrix}$

Ein zweidimensionales Bild lässt sich somit vollständig durch das Tupel

$$P = (N_1, N_2, \Delta x_0, \Delta x_1, W, (p_{ij}))$$

beschreiben.
Zur Betonung der räumlichen Ausdehnung einzelner Bildpunkte werden diese oft auch

- bei 2D Bildern als **Pixel** („Picture Element“),
- bei 3D Bildern als **Voxel** („Volume Element“)

bezeichnet.

Beispiel einer 2D Bilddatei im PPM ASCII Format

```
P3  Dateityp
# bild.ppm
9 9  N1 , N2
10  W = {0, 1, 2, . . . , 10}
 0  5  0    0  5  0    0  5  0      pij
 0  5  0    0  5  0    0  5  0
 0  5  0    0  5  0    0  5  0
 9  9  0    0  5  0    0  5  0

 0  5  0    0  5  0    9  9  0
 9  9  0    9  9  0    0  5  0
 0  5  0    9  9  0    9  9  0
 0  0  0    9  9  0    9  9  0

 0  5  0    9  9  0    9  9  0
 9  9  0    9  9  0    9  9  0
 0  5  0    9  9  0    0  0  0
 9  9  0    0  0  0    9  9  0
```

RGB Bilder

Abbildung: Pixel können auch aus mehreren Kanälen bestehen. Das Beispiel zeigt ein Farbbild mit Rot, Grün und Blau Kanal.

Wir werden in den folgenden Kapiteln in der Regel mit digitalen Bildern arbeiten.
Trotzdem spielen kontinuierliche Bilder bei der Analyse der angewandten Verfahren oft eine wesentliche Rolle.

Nachbarschaften

Von großer Bedeutung sind Nachbarschaften in 2D- und 3D-Bildern, für die es jedoch verschiedene Definitionen gibt.
Generell unterscheidet man zwischen

- **Punktnachbarschaft**: benachbarte Pixel haben gemeinsame Punkte
- **Kantennachbarschaft**: benachbarte Pixel haben gemeinsame Kanten
- **Flächennachbarschaft**: benachbarte Pixel haben gemeinsame Flächen

Nachbarschaften

Von großer Bedeutung sind Nachbarschaften in 2D- und 3D-Bildern, für die es jedoch verschiedene Definitionen gibt.
Generell unterscheidet man zwischen

- **Punktnachbarschaft**: benachbarte Pixel haben gemeinsame Punkte
- **Kantennachbarschaft**: benachbarte Pixel haben gemeinsame Kanten
- **Flächennachbarschaft**: benachbarte Pixel haben gemeinsame Flächen
- Da Pixel mit gemeinsamen Kanten auch automatisch gemeinsame Punkte besitzen, lässt sich hiermit eine Hierarchie aufbauen.
- Gleichzeitig ist offensichtlich, dass Flächennachbarschaft (eine Fläche ist ein zweidimensionales Objekt) nur im dreidimensionalen Raum sinnvoll ist.

  Nachbarschaften können also immer nur in einer Dimension, welche niedriger als die Dimension des einbettenden Raumes ist, definiert sein

2D-Nachbarschaften

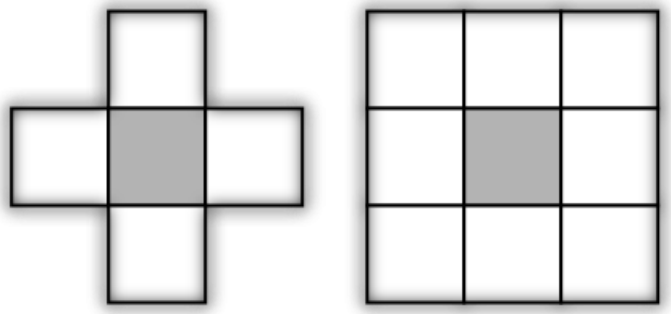
Für reguläre Gitter im zweidimensionalen Raum gilt:

- Zählen nur Pixel mit gemeinsamen Kanten als Nachbarn, so hat jedes innere Pixel im Datensatz genau vier Nachbarn. Man spricht auch von einer **Vierernachbarschaft**.

2D-Nachbarschaften

Für reguläre Gitter im zweidimensionalen Raum gilt:

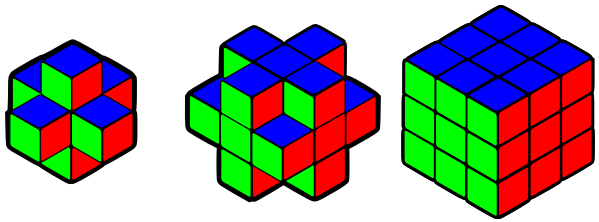
- Zählen nur Pixel mit gemeinsamen Kanten als Nachbarn, so hat jedes innere Pixel im Datensatz genau vier Nachbarn. Man spricht auch von einer **Vierernachbarschaft**.
- Zählen alle Pixel mit gemeinsamen Punkten als Nachbarn, so erhält man acht Nachbarn (die vier Kantennachbarn sind darin enthalten) und spricht von einer **Achternachbarschaft**.



3D-Nachbarschaften

Im dreidimensionalen Raum werden analog dazu eingeführt

- Die Flächennachbarschaft als 6er-Nachbarschaft
- Die Kantennachbarschaft als 18er-Nachbarschaft
- Die Punktnachbarschaft als 26er-Nachbarschaft



Wie aus der Mathematik bekannt, kann der Abstand zweier Punkte über

$$d(r_{i,j}, r_{i',j'}) = \sqrt[p]{|i - i'|^p + |j - j'|^p}$$

definiert werden.

Wie aus der Mathematik bekannt, kann der Abstand zweier Punkte über

$$d(r_{i,j}, r_{i',j'}) = \sqrt[p]{|i-i'|^p + |j-j'|^p}$$

definiert werden.
Drei Mögliche Werte für p sind 1, 2 und ∞.

- $d(r_{i,j}, r_{i',j'}) = |i-i'| + |j-j'|$: **Manhattendistanz**, **Blockdistanz** (Summe der Distanz in x- und in y-Richtung)
- $d(r_{i,j}, r_{i',j'}) = \sqrt[2]{(i-i')^2 + (j-j')^2}$: **Euklidische Distanz**; abgekürzt mit $\|\cdot\|_2 = \|\cdot\|$
- $d(r_{i,j}, r_{i',j'}) = \max\{|i-i'|, |j-j'|\}$: **Schachbrettdistanz**, **Maximumsnorm** (die größere der Entfernungen entlang der x- bzw. der y-Achse)

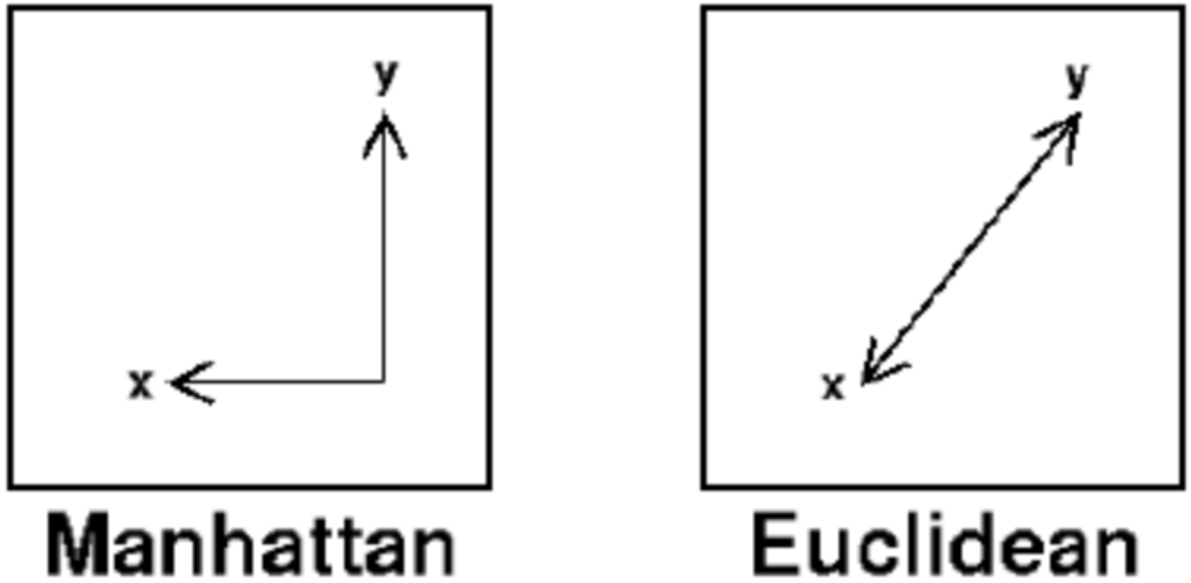


Abbildung: Euklidische Distanz vs. Manhatten Distanz

Randbehandlung bei Bildern

Um überall eine Nachbarschaft bestimmen zu können, ist eine Fortsetzung des Bildes am Rand nötig. Hierfür gibt es verschiedene Methoden, wovon vier hier vorgestellt werden sollen.

- Konstanter Wert:
  - Vorteil: Einfache Ermittlung der Pixelwerte.
  - Nachteil: Kein Bezug zum eigentlichen Bild.

Randbehandlung bei Bildern

Um überall eine Nachbarschaft bestimmen zu können, ist eine Fortsetzung des Bildes am Rand nötig. Hierfür gibt es verschiedene Methoden, wovon vier hier vorgestellt werden sollen.

- Konstanter Wert:
  - Vorteil: Einfache Ermittlung der Pixelwerte.
  - Nachteil: Kein Bezug zum eigentlichen Bild.
- Nächstes Randpixel:
  - Vorteil: Bild außerhalb hat lokale Nähe zum Bildinhalt.
  - Nachteil: Randpunkte werden stark gewichtet.

Randbehandlung bei Bildern

Um überall eine Nachbarschaft bestimmen zu können, ist eine Fortsetzung des Bildes am Rand nötig. Hierfür gibt es verschiedene Methoden, wovon vier hier vorgestellt werden sollen.

- Konstanter Wert:
  - Vorteil: Einfache Ermittlung der Pixelwerte.
  - Nachteil: Kein Bezug zum eigentlichen Bild.
- Nächstes Randpixel:
  - Vorteil: Bild außerhalb hat lokale Nähe zum Bildinhalt.
  - Nachteil: Randpunkte werden stark gewichtet.
- Zyklisch verschoben:
  - Vorteil: Einfache Ermittlung der Pixel durch Modulo-Operation. Die Behandlung ist kompatibel zur später beschriebenen Fouriertransformation.
  - Nachteil: Ermittelte Werte haben keinen Bezug zu den Randwerten, da sie vom gegenüberliegenden Rand stammen.

Randbehandlung bei Bildern

Um überall eine Nachbarschaft bestimmen zu können, ist eine Fortsetzung des Bildes am Rand nötig. Hierfür gibt es verschiedene Methoden, wovon vier hier vorgestellt werden sollen.

- Konstanter Wert:
  - Vorteil: Einfache Ermittlung der Pixelwerte.
  - Nachteil: Kein Bezug zum eigentlichen Bild.
- Nächstes Randpixel:
  - Vorteil: Bild außerhalb hat lokale Nähe zum Bildinhalt.
  - Nachteil: Randpunkte werden stark gewichtet.
- Zyklisch verschoben:
  - Vorteil: Einfache Ermittlung der Pixel durch Modulo-Operation. Die Behandlung ist kompatibel zur später beschriebenen Fouriertransformation.
  - Nachteil: Ermittelte Werte haben keinen Bezug zu den Randwerten, da sie vom gegenüberliegenden Rand stammen.
- Zyklisch gespiegelt:
  - Vorteil: Lokaler Bezug der Randpixel; Randwerte werden nur moderat gewichtet.
  - Nachteil: Erhöhter Berechnungsaufwand der Positionsermittlung.

Die unterschiedlichen Möglichkeiten der Randbehandlung graphisch dargestellt:

k	k	k	k	k	k	k	k
k	k	k	k	k	k	k	k
k	k	[gray]0.811	[gray]0.812	[gray]0.813	[gray]0.814	k	k
k	k	[gray]0.821	[gray]0.822	[gray]0.823	[gray]0.824	k	k
k	k	[gray]0.831	[gray]0.832	[gray]0.833	[gray]0.834	k	k
k	k	k	k	k	k	k	k
k	k	k	k	k	k	k	k

konstanter Wert k

Die unterschiedlichen Möglichkeiten der Randbehandlung graphisch dargestellt:

k	k	k	k	k	k	k	k
k	k	k	k	k	k	k	k
k	k	[gray]0.811	[gray]0.812	[gray]0.813	[gray]0.814	k	k
k	k	[gray]0.821	[gray]0.822	[gray]0.823	[gray]0.824	k	k
k	k	[gray]0.831	[gray]0.832	[gray]0.833	[gray]0.834	k	k
k	k	k	k	k	k	k	k
k	k	k	k	k	k	k	k

konstanter Wert k

11	11	11	12	13	14	14	14
11	11	11	12	13	14	14	14
11	11	[gray]0.811	[gray]0.812	[gray]0.813	[gray]0.814	14	14
21	21	[gray]0.821	[gray]0.822	[gray]0.823	[gray]0.824	24	24
31	31	[gray]0.831	[gray]0.832	[gray]0.833	[gray]0.834	34	34
31	31	31	32	33	34	34	34
31	31	31	32	33	34	34	34

nächstes Pixel

Die unterschiedlichen Möglichkeiten der Randbehandlung graphisch dargestellt:

k	k	k	k	k	k	k	k
k	k	k	k	k	k	k	k
k	k	[gray]0.811	[gray]0.812	[gray]0.813	[gray]0.814	k	k
k	k	[gray]0.821	[gray]0.822	[gray]0.823	[gray]0.824	k	k
k	k	[gray]0.831	[gray]0.832	[gray]0.833	[gray]0.834	k	k
k	k	k	k	k	k	k	k
k	k	k	k	k	k	k	k

konstanter Wert k

11	11	11	12	13	14	14	14
11	11	11	12	13	14	14	14
11	11	[gray]0.811	[gray]0.812	[gray]0.813	[gray]0.814	14	14
21	21	[gray]0.821	[gray]0.822	[gray]0.823	[gray]0.824	24	24
31	31	[gray]0.831	[gray]0.832	[gray]0.833	[gray]0.834	34	34
31	31	31	32	33	34	34	34
31	31	31	32	33	34	34	34

nächstes Pixel

Die unterschiedlichen Möglichkeiten der Randbehandlung graphisch dargestellt:

23	24	21	22	23	24	21	22
33	34	31	32	33	34	31	32
13	14	[gray]0.811	[gray]0.812	[gray]0.813	[gray]0.814	11	12
23	24	[gray]0.821	[gray]0.822	[gray]0.823	[gray]0.824	21	22
33	34	[gray]0.831	[gray]0.832	[gray]0.833	[gray]0.834	31	32
13	14	11	12	13	14	11	12
23	24	21	22	23	24	21	22

zyklisch

Die unterschiedlichen Möglichkeiten der Randbehandlung graphisch dargestellt:

23	24	21	22	23	24	21	22
33	34	31	32	33	34	31	32
13	14	[gray]0.811	[gray]0.812	[gray]0.813	[gray]0.814	11	12
23	24	[gray]0.821	[gray]0.822	[gray]0.823	[gray]0.824	21	22
33	34	[gray]0.831	[gray]0.832	[gray]0.833	[gray]0.834	31	32
13	14	11	12	13	14	11	12
23	24	21	22	23	24	21	22

zyklisch

22	12	21	22	23	24	24	23
12	11	11	12	13	14	14	13
12	11	[gray]0.811	[gray]0.812	[gray]0.813	[gray]0.814	14	13
22	21	[gray]0.821	[gray]0.822	[gray]0.823	[gray]0.824	24	23
32	31	[gray]0.831	[gray]0.832	[gray]0.833	[gray]0.834	34	33
32	31	31	32	33	34	34	33
22	21	21	22	23	24	24	23

zyklisch gespiegelt

Ein Vektorraum über einem Körper $(K, +, \cdot)$ oder kurz K-Vektorraum ist eine additive abelsche Gruppe $(V, +)$, auf der die Multiplikation mit einem Skalar aus K definiert ist

$$\cdot : K \times V \to V$$

wobei für alle $u, v \in V$ und $\alpha, \beta \in K$ gilt

- $\alpha \cdot (\beta \cdot v) = (\alpha \cdot \beta) \cdot v$
- $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$
- $(\alpha + \beta) \cdot v = \alpha \cdot v + \beta \cdot v$
- $1_K \cdot v = v$

Bilder als Elemente von Vektorräumen

Man kann 2D-Bilder als $N_1 \times N_2$-dimensionale reell-wertige oder komplex-wertige Vektorräume (für $K = \mathbb{R}$ oder $K = \mathbb{C}$) auffassen.
Wir definieren $N_1 \times N_2$ Basisbilder ${}^{m,n}P$ durch

$$ {}^{m,n}P_{i,j} := \begin{cases} 1 & i = m \wedge j = n \\ 0 & \text{sonst} \end{cases} $$

Dies sind die Bilder, welche genau einen Bildpunkt mit dem Wert Eins und sonst ausschließlich Punkte mit dem Wert Null enthalten.

Man kann 2D-Bilder als $N_1 \times N_2$-dimensionale reell-wertige oder komplex-wertige Vektorräume (für $K = \mathbb{R}$ oder $K = \mathbb{C}$) auffassen.
Wir definieren $N_1 \times N_2$ Basisbilder ${}^{m,n}P$ durch

$$ {}^{m,n}P_{i,j} := \begin{cases} 1 & i = m \wedge j = n \\ 0 & \text{sonst} \end{cases} $$

Dies sind die Bilder, welche genau einen Bildpunkt mit dem Wert Eins und sonst ausschließlich Punkte mit dem Wert Null enthalten.
Definiert man nun das Skalarprodukt auf zwei Bildern über

$$ \langle P, Q \rangle = \sum_{i=0}^{N_0-1} \sum_{j=0}^{N_1-1} p_{i,j} \cdot q_{i,j} $$

so ist offensichtlich, dass die genannte Basis eine Orthonormalbasis ist, d.h., das Skalarprodukt zweier Basisvektoren ist eins gdw. die beiden Vektoren gleich sind, sonst null.

Gleichzeitig ist offensichtlich, dass sich jedes Bild als Linearkombination der genannten Basisvektoren darstellen lässt, denn es gilt:

$$V_{i,j} = \sum_{m=0}^{N_0-1} \sum_{n=0}^{N_1-1} {}^{m,n}P_{i,j} \cdot V_{i,j}$$

Teil III

Farben

Bildverarbeitungsprozess

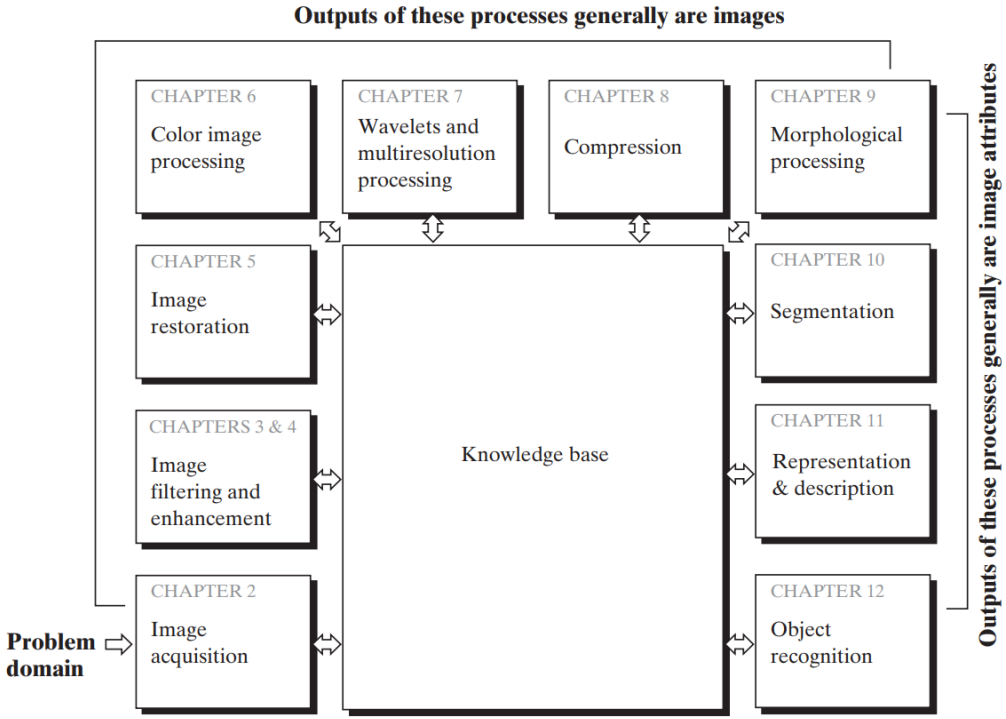


Abbildung: Die Bildverarbeitung umfasst eine reihe von Algorithmen, die entweder wiederum Bilder oder Bildattribute ausgeben. [Digital Image Processing, 3rd Edition, Gonzales and Woods]

Diese Vorlesung: Farben in Bildern

Farbsysteme

- Aristoteles: Experimente mit farbigen Gläsern. Theorie, dass Farben ein Gemisch aus schwarz und weiß sind.
- Isaak Newton: Experimente mit Prismen
- Thomas Young (1801) + Hermann von Helmholtz: Dreifarbentheorie
- Maxwell, Helmholtz, Grassman: Grundlagen der Colormetrie
- Grassmann (1853): Regeln über das Mischen von farbigem Licht

Für die menschliche Farbwahrnehmung gilt:

- Farben, die durch additive Mischung entstehen, werden vollständig durch drei Farbkomponenten beschrieben und sind unabhängig von ihrer Position im Spektrum des weißen Lichtes. Voraussetzung dafür ist es, dass keine dieser drei Farben durch Kombination der beiden anderen gebildet werden kann.

Für die menschliche Farbwahrnehmung gilt:

- Farben, die durch additive Mischung entstehen, werden vollständig durch drei Farbkomponenten beschrieben und sind unabhängig von ihrer Position im Spektrum des weißen Lichtes. Voraussetzung dafür ist es, dass keine dieser drei Farben durch Kombination der beiden anderen gebildet werden kann.
- Die drei Farbkomponenten sind hinreichend, um eine beliebige Farbe darzustellen. Die Kombination zweier beliebiger Farben lässt sich linear aus den zu jeder dieser Farben korrespondierenden Basisfarben erstellen.

Für die menschliche Farbwahrnehmung gilt:

- Farben, die durch additive Mischung entstehen, werden vollständig durch drei Farbkomponenten beschrieben und sind unabhängig von ihrer Position im Spektrum des weißen Lichtes. Voraussetzung dafür ist es, dass keine dieser drei Farben durch Kombination der beiden anderen gebildet werden kann.
- Die drei Farbkomponenten sind hinreichend, um eine beliebige Farbe darzustellen. Die Kombination zweier beliebiger Farben lässt sich linear aus den zu jeder dieser Farben korrespondierenden Basisfarben erstellen.
- Die Farbwahrnehmung hängt nicht von der Intensität des Lichtes ab. Das trifft allerdings nicht für sehr niedrige Lichtintensitäten zu, bei denen die Wahrnehmung durch die Stäbchen in der Netzhaut dominiert wird.

Um den kompletten Farbraum darzustellen, wäre eine kontinuierliche Darstellung aller Frequenzen nötig.

Um den kompletten Farbraum darzustellen, wäre eine kontinuierliche Darstellung aller Frequenzen nötig.
Da einerseits der Mensch nur die drei Farbkanäle zur Wahrnehmung zur Verfügung hat, andererseits die technischen Möglichkeiten die Farbdarstellung begrenzen, beschränkt man sich oft auf vereinfachte Farbsysteme.

Geräteabhängige Farbsysteme

Die bekanntesten Farbsysteme für bestimmte Ausgabegeräte sind

- RGB-System
  - Farben **R**ot, **G**rün und **B**lau
  - Wird in der Regel für Bildschirme verwendet
  - Additive Farbmischung

Geräteabhängige Farbsysteme

Die bekanntesten Farbsysteme für bestimmte Ausgabegeräte sind

- RGB-System
  - Farben **R**ot, **G**rün und **B**lau
  - Wird in der Regel für Bildschirme verwendet
  - Additive Farbmischung
- CMY-System
  - Farben **C**yan (Cyanblau), **M**agenta (Purpur) und **Y**ellow (Gelb)
  - Wird in der Regel für die Druckausgabe verwendet
  - Subtraktive Farbmischung

Geräteabhängige Farbsysteme

Die bekanntesten Farbsysteme für bestimmte Ausgabegeräte sind

- RGB-System
  - Farben **R**ot, **G**rün und **B**lau
  - Wird in der Regel für Bildschirme verwendet
  - Additive Farbmischung
- CMY-System
  - Farben **C**yan (Cyanblau), **M**agenta (Purpur) und **Y**ellow (Gelb)
  - Wird in der Regel für die Druckausgabe verwendet
  - Subtraktive Farbmischung
- Das CMYK-Farbsystem hat eine zusätzliche Komponente für schwarz (blac**k**).

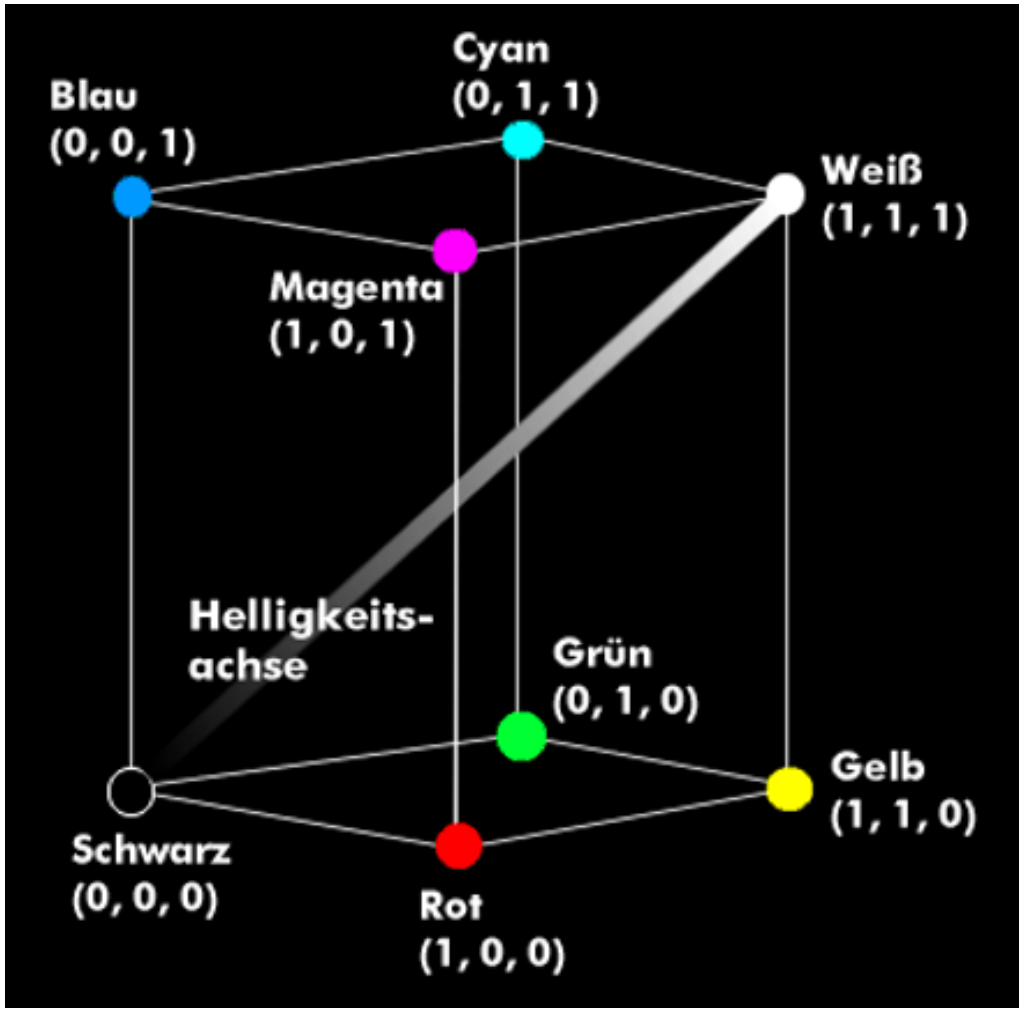


Abbildung: Darstellung des RGB Raums als Würfel

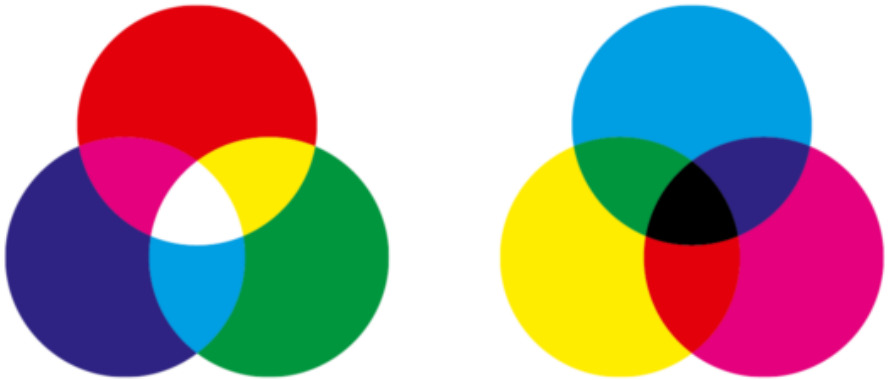


Abbildung: Additive vs. Subtraktive Farbmischung

RGB und CMY-Farbsystem

Die Umrechnung von RGB- auf das CMY-Farbsystem ist relativ einfach, da die Farben jeweils komplementär sind.
Aus dem RGB-Tripel (r,g,b) ergibt sich das CMY-Tripel
(c,m,y) durch

$$\begin{aligned} c &= 1 - r \\ m &= 1 - g \\ y &= 1 - b \end{aligned}$$

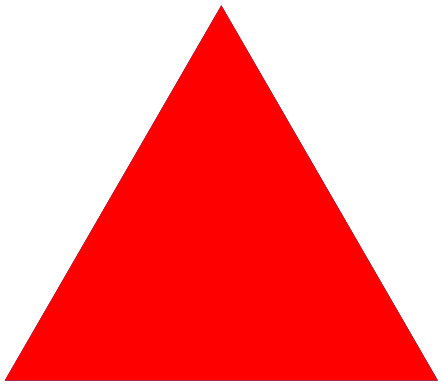
RGB und CMY-Farbsystem

Die Umrechnung von RGB- auf das CMY-Farbsystem ist relativ einfach, da die Farben jeweils komplementär sind.
Aus dem RGB-Tripel (r,g,b) ergibt sich das CMY-Tripel (c,m,y) durch

$$\begin{aligned} c &= 1-r \\ m &= 1-g \\ y &= 1-b \end{aligned}$$

Aus dem CMY-Tripel (c,m,y) ergibt sich das RGB-Tripel (r,g,b) durch

$$\begin{aligned} r &= 1-c \\ g &= 1-m \\ b &= 1-y. \end{aligned}$$



Aus dem CMY-Tripel (c', m', y') ergibt sich das CMYK-Quadrupel (c, m, y, k) durch

$$
\begin{aligned}
k &= \min(c', m', y') \\
c &= \frac{c' - k}{1 - k} \\
m &= \frac{m' - k}{1 - k} \\
y &= \frac{y' - k}{1 - k}
\end{aligned}
$$

Aus dem CMY-Tripel (c', m', y') ergibt sich das CMYK-Quadrupel (c, m, y, k) durch

$$\begin{aligned} k &= \min(c', m', y') \\ c &= \frac{c' - k}{1 - k} \\ m &= \frac{m' - k}{1 - k} \\ y &= \frac{y' - k}{1 - k} \end{aligned}$$

Aus dem CMYK-Quadrupel (c, m, y, k) ergibt sich das CMY-Tripel (c', m', y') durch

$$\begin{aligned} c' &= \min(1, c \cdot (1 - k) + k) \\ m' &= \min(1, m \cdot (1 - k) + k) \\ y' &= \min(1, y \cdot (1 - k) + k) \end{aligned}$$

Wahrnehmungsabhängige Farbsysteme

Die einzige Möglichkeit, eine geräteunabhängige Farbangabe zu machen, besteht darin, das Auge als Referenz zu benutzen.
Die Bestimmung der Farbempfindlichkeit des Auges beruht auf statistischen Tests der Farbwahrnehmung von Testpersonen. Eine allgemein gültige Aussage, wie eine Person ein Farbspektrum wahrnimmt, ist jedoch nicht möglich.

Abbildung: Fruchtkorb als Originalbild (links) und simuliert als Deuteranopie (Grünblindheit, 5% der Männer, Mitte) und Protanopie (Rotblindheit, 2% der Männer, rechts).

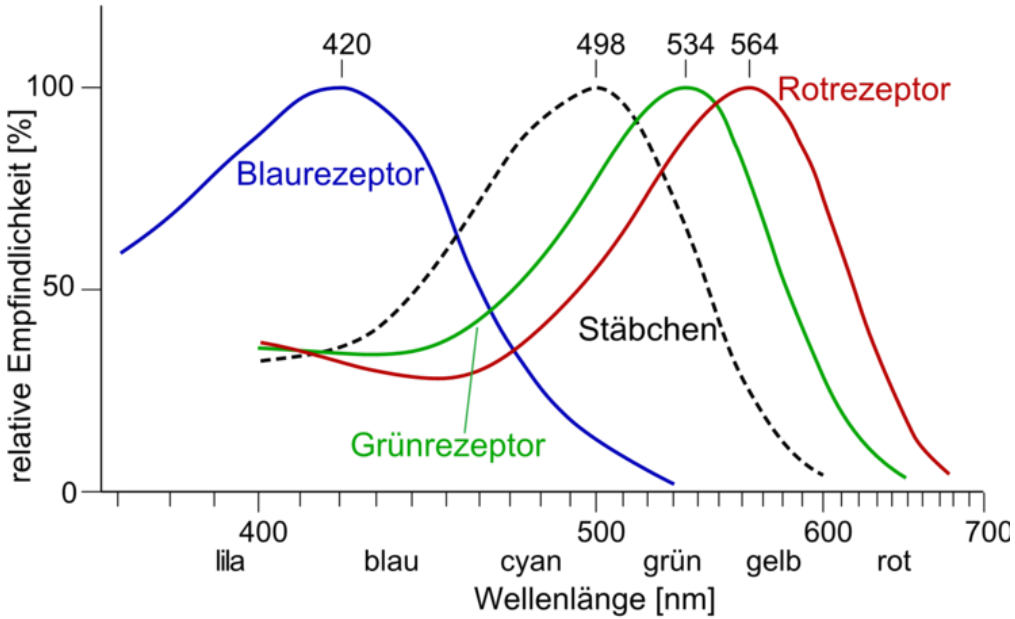


Abbildung: Rezeptoren und ihre Wahrnehmung beim Menschen

Auf dieser Grundlage ist von der International Commission on Illumination 1913 das CIE (**C**ommission **I**nternationale de l'**E**claire) entwickelt worden. Soll eine farbtreue Ausgabe auf verschiedenen Geräten erreicht werden, so muss jeweils aus den Eigenschaften des Gerätes (z.B. unterschiedliche Luminizenzschichten auf einem Monitor, die Farben des verwendeten Druckers) in das CIE-System umgerechnet werden und für das jeweilige Ausgabegerät eine weitere Umrechnung erfolgen.

CIE Farbsystem

Beim CIE-System werden die vom Menschen unterscheidbaren Farbvalenzen als Grundlage benutzt. Die Empfindlichkeiten der drei Komponenten $x(\lambda)$, $y(\lambda)$ und $z(\lambda)$ entspricht dabei grob der Empfindlichkeit für das Unterscheiden von Gelb, Grün und Blau.

CIE Farbsystem

Beim CIE-System werden die vom Menschen unterscheidbaren Farbvalenzen als Grundlage benutzt. Die Empfindlichkeiten der drei Komponenten $x(\lambda)$, $y(\lambda)$ und $z(\lambda)$ entspricht dabei grob der Empfindlichkeit für das Unterscheiden von Gelb, Grün und Blau.

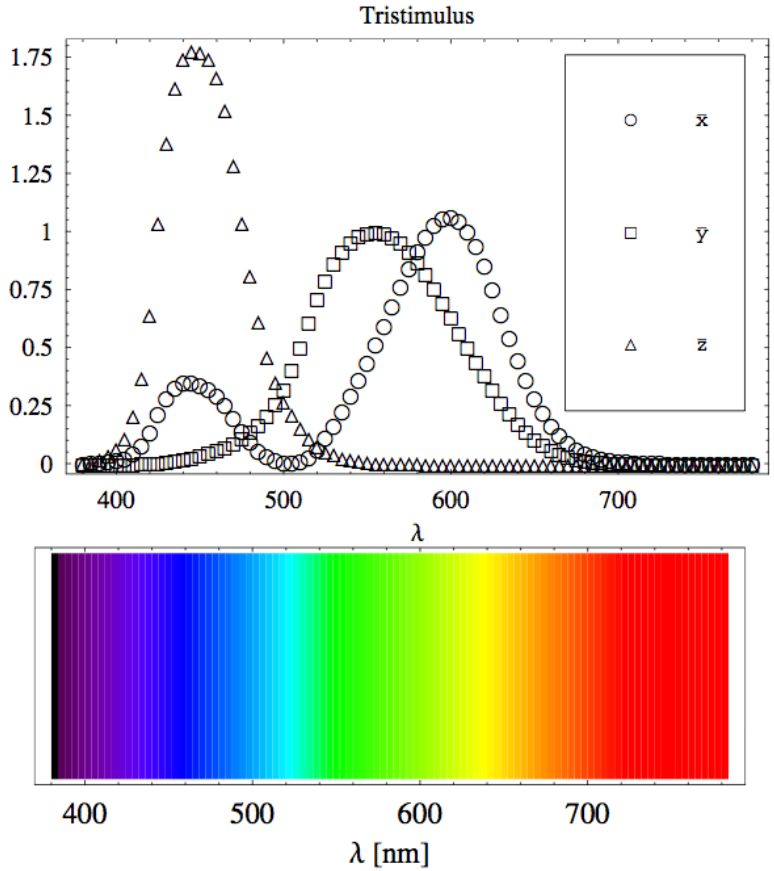


Abbildung: Normalspektralkurven $x(\lambda)$, $y(\lambda)$ und $z(\lambda)$ im CIE-System. Darunter sind die Wellenlängen der korrespondierenden Spektralfarben dargestellt.

Für die Farben des Lichtes ergibt sich die Zerlegung in Normalfarbwerte entsprechend der Abbildung.

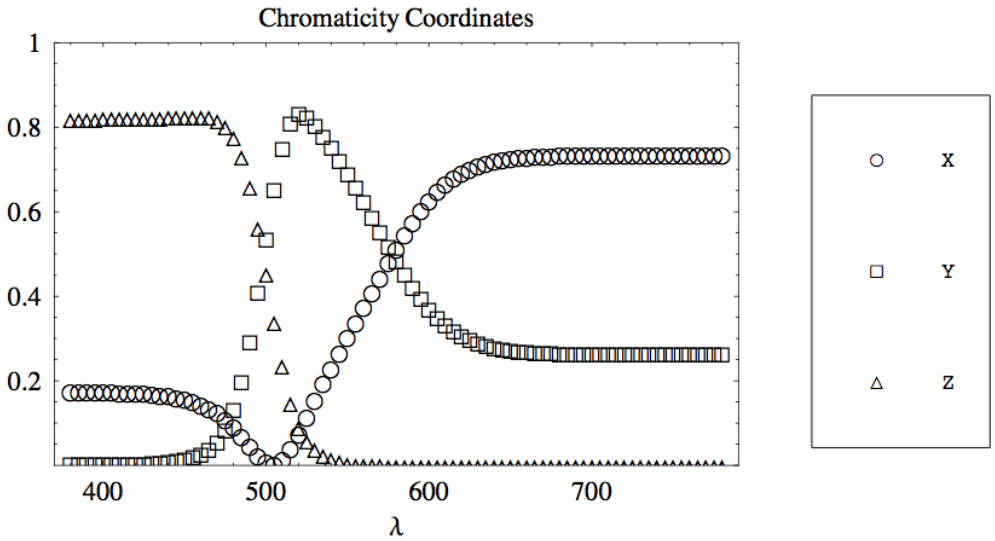


Abbildung: Die Zerlegung der Spektralfarben in Normalfarbwerte X, Y und Z.

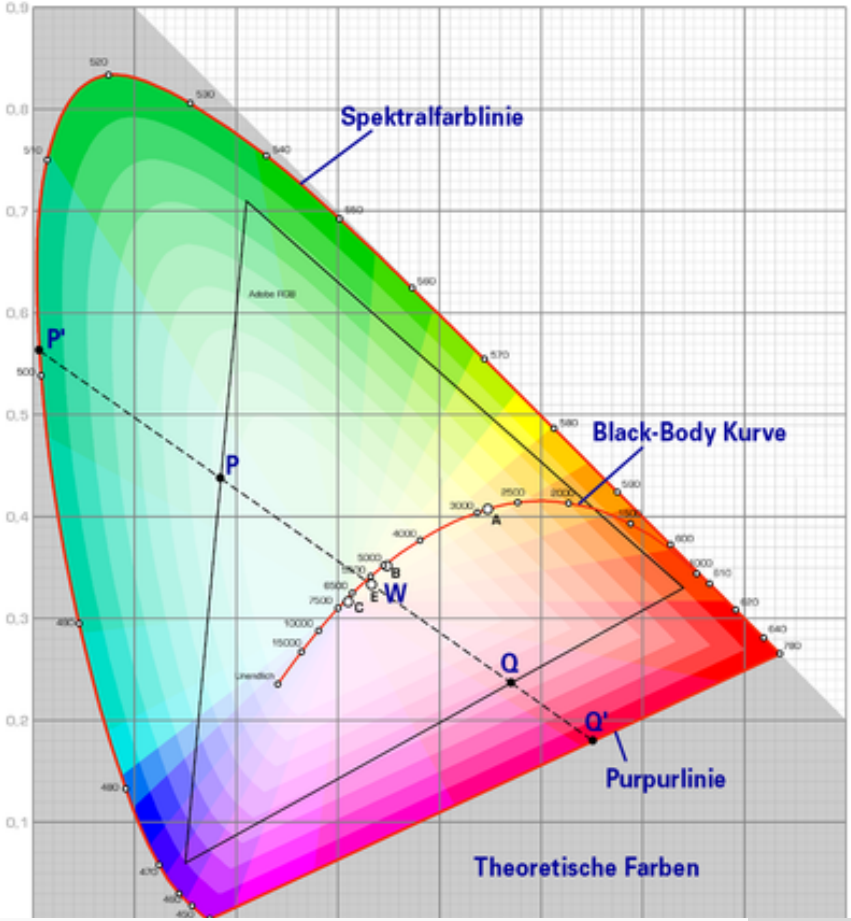
Aufgrund der Normierung ist eine der Koordinaten überflüssig und man verwendet die Koordinaten (x, y)

$$
\begin{aligned}
x &= \frac{X}{X+Y+Z} \\
y &= \frac{Y}{X+Y+Z} \\
z &= \frac{Z}{X+Y+Z} = 1-x-y
\end{aligned}
$$

Insbesondere bei der Transformation zu RGB-Farben ist zu beachten, dass negative Werte der Farbkomponenten zu Null gesetzt werden und Werte größer als Eins gekappt werden.
Damit ergibt sich die Normalfarbtafel nach CIE für den Normalbeobachter (Abbildung rechts). Das schwarze Polygon entspricht der parametrischen Darstellung $(x(\lambda); y(\lambda))$.

CIE Farbsystem

Insbesondere bei der Transformation zu RGB-Farben ist zu beachten, dass negative Werte der Farbkomponenten zu Null gesetzt werden und Werte größer als Eins gekappt werden.
Damit ergibt sich die Normalfarbtafel nach CIE für den Normalbeobachter (Abbildung rechts). Das schwarze Polygon entspricht der parametrischen Darstellung $(x(\lambda); y(\lambda))$.



Den Farben können Merkmale wie Helligkeit, Farbsättigung und Farbton zugewiesen werden.
HSV-Farbmodell:

- Farbton (Hue) als Farbwinkel auf dem Farbkreis
- Sättigung (Saturation) in Prozent
  - 0% = Grau
  - 100% = reine Farbe
- Hellwert (Value) als Prozentwert
  - 0%= keine Helligkeit
  - 100%= volle Helligkeit

Den Farben können Merkmale wie Helligkeit, Farbsättigung und Farbton zugewiesen werden.
HSV-Farbmodell:

- Farbton (Hue) als Farbwinkel auf dem Farbkreis
- Sättigung (Saturation) in Prozent
  - 0% = Grau
  - 100% = reine Farbe
- Hellwert (Value) als Prozentwert
  - 0%= keine Helligkeit
  - 100%= volle Helligkeit

HSL-Farbmodell

- Farbton (Hue) als Farbwinkel auf dem Farbkreis
- Sättigung (Saturation) in Prozent
  - 0% = Grau
  - 100% = reine Farbe
- Helligkeit (Lightness)
  - 0% = schwarz
  - 100% = weiß

Oft sind die vorgestellten Modelle jedoch nicht ausreichend. Um ein weiteres Spektrum an Farben abbilden zu können, bedient man sich spezieller Tricks, um die Farben lokal anzupassen. Ein Beispiel dafür sind Bilder mit besonders hohem Dynamikumfang (High Dynamic Range images, HDR).

Teil IV

Lineare Filter

Bildverarbeitungsprozess

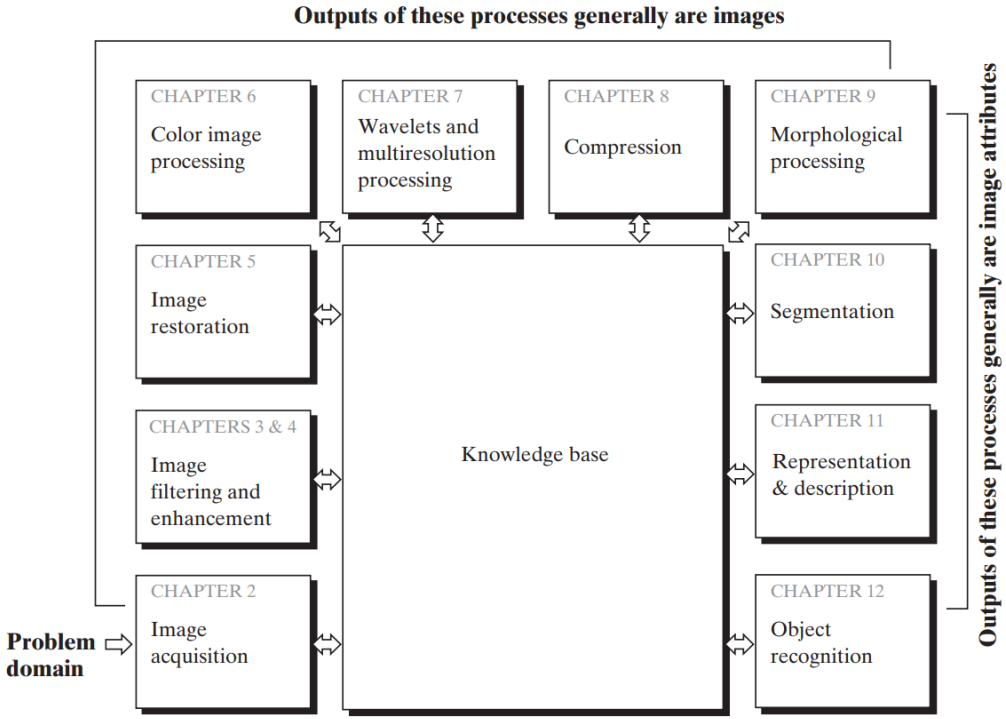


Abbildung: Die Bildverarbeitung umfasst eine reihe von Algorithmen, die entweder wiederum Bilder oder Bildattribute ausgeben. [Digital Image Processing, 3rd Edition, Gonzales and Woods]

Diese Vorlesung: Lineare Filter

Filter

- In der Bildverarbeitung ist es üblich, Operationen auf Bilder anzuwenden, die erneut Bilder erzeugen.
- In Anlehnung an die Signalverarbeitung wird dieser Prozess allgemein als Filtern bezeichnet.
- Der angewandte Operator wird als **Filter** bezeichnet.
  - Duden: das Filter (neutrum)
  - ingenieursmäßiger Gebrauch: der Filter (maskulin)

- In der Bildverarbeitung ist es üblich, Operationen auf Bilder anzuwenden, die erneut Bilder erzeugen.
- In Anlehnung an die Signalverarbeitung wird dieser Prozess allgemein als Filtern bezeichnet.
- Der angewandte Operator wird als **Filter** bezeichnet.
  - Duden: das Filter (neutrum)
  - ingenieursmäßiger Gebrauch: der Filter (maskulin)

Ein **Filter** auf Bildern der Größe

$$N_1 \times N_2 \times \ldots \times N_d$$

mit einem Wertebereich

$$W$$

ist eine Abbildung

$$F : W^{N_1 \times N_2 \times \ldots \times N_d} \rightarrow W^{N_1 \times N_2 \times \ldots \times N_d}$$

welche aus einem d-dimensionalen Bild wieder ein d-dimensionales Bild gleicher Größe erzeugt.

Schreibt man ein Bild als eindimensionalen Vektor $\vec{v}$, so kann man eine **lineare Operation** auf dem Bild über eine Matrix M mit

$$\begin{array}{lcccc} M & : & W^{N_1 \times N_2 \times \ldots \times N_d} & \rightarrow & W^{N_1 \times N_2 \times \ldots \times N_d} \\ & & \vec{v} & \mapsto & M \cdot \vec{v} \end{array}$$

beschreiben.

Glättungs- und Mittelungsoperatoren

Operationen zur Glättung und Mittelung von Bildern werden der eigentlichen Merkmalsextraktion oft vorgeschaltet, um bessere Resultate zu erzielen.

Glättungsfilter

Eine wichtige Gruppe der Glättungsoperationen basiert auf LSI Filtern. Zur Herleitung der Eigenschaften kann man Bedingungen an die Filter formulieren.

Glättungsfilter: Bedingungen

- Ein Glättungsfilter sollte Merkmalspositionen nicht verändern. Insbesondere sollte er bei keiner Wellenlänge eine Phasenverschiebung induzieren (nullphasiger Filter).
  - Da nach dem Faltungstheorem aus der Faltung eine Multiplikation im Fourierraum wird und eine Phasenverschiebung durch den imaginären Anteil beschrieben wird, muss die Transferfunktion eines Glättungsfilters reell sein.
  - Dies wiederum bedeutet, dass der Filter h selbst symmetrisch sein muss.

Glättungsfilter: Bedingungen

- Ein Glättungsfilter sollte Merkmalspositionen nicht verändern. Insbesondere sollte er bei keiner Wellenlänge eine Phasenverschiebung induzieren (nullphasiger Filter).
  - Da nach dem Faltungstheorem aus der Faltung eine Multiplikation im Fourierraum wird und eine Phasenverschiebung durch den imaginären Anteil beschrieben wird, muss die Transferfunktion eines Glättungsfilters reell sein.
  - Dies wiederum bedeutet, dass der Filter h selbst symmetrisch sein muss.
- Ein Glättungsfilter h sollte den Mittelwert erhalten.
  - Dies bedeutet, dass $\hat{h}(0) = 1$ sein muss.
  - Umgekehrt kann man schlussfolgern, dass sich die einzelnen Filtereinträge zu eins aufaddieren müssen.

Glättungsfilter: Bedingungen

Ein Glättungsfilter sollte feinere Strukturen, also kleinere Wellenlängen, stärker dämpfen als große Wellenlängen:

$$\hat{h}(k_2) \leq \hat{h}(k_1) \quad \text{falls} \quad k_2 > k_1$$

Glättungsfilter: Bedingungen

Ein Glättungsfilter sollte feinere Strukturen, also kleinere Wellenlängen, stärker dämpfen als große Wellenlängen:

$$\hat{h}(k_2) \leq \hat{h}(k_1) \quad \text{falls} \quad k_2 > k_1$$

Für unsere endlichen Bilder gibt es eine größte Wellenzahl, für welche die Transferfunktion 0 sein sollte.

Glättungsfilter: Bedingungen

Ein Glättungsfilter sollte feinere Strukturen, also kleinere Wellenlängen, stärker dämpfen als große Wellenlängen:

$$\hat{h}(k_2) \leq \hat{h}(k_1) \quad \text{falls} \quad k_2 > k_1$$

Für unsere endlichen Bilder gibt es eine größte Wellenzahl, für welche die Transferfunktion 0 sein sollte.
Ferner liefert die zuvor definierte Erhaltung des Mittelwerts $\hat{h}(k) \leq 1$ für alle k.

Glättungsfilter: Bedingungen

Ein Glättungsfilter sollte feinere Strukturen, also kleinere Wellenlängen, stärker dämpfen als große Wellenlängen:

$$\hat{h}(k_2) \leq \hat{h}(k_1) \quad \text{falls} \quad k_2 > k_1$$

Für unsere endlichen Bilder gibt es eine größte Wellenzahl, für welche die Transferfunktion 0 sein sollte.
Ferner liefert die zuvor definierte Erhaltung des Mittelwerts $\hat{h}(k) \leq 1$ für alle k.
Bei mehrdimensionalen Bildern wird auch eine Unabhängigkeit der Glättung von der Richtung angestrebt (Isotropie).

Glättungsfilter: Bedingungen

Ein Glättungsfilter sollte feinere Strukturen, also kleinere Wellenlängen, stärker dämpfen als große Wellenlängen:

$$\hat{h}(k_2) \leq \hat{h}(k_1) \quad \text{falls} \quad k_2 > k_1$$

Für unsere endlichen Bilder gibt es eine größte Wellenzahl, für welche die Transferfunktion 0 sein sollte.
Ferner liefert die zuvor definierte Erhaltung des Mittelwerts $\hat{h}(k) \leq 1$ für alle k.
Bei mehrdimensionalen Bildern wird auch eine Unabhängigkeit der Glättung von der Richtung angestrebt (Isotropie).
Dies bedeutet im Objektraum und im Fourierraum, dass der Wert nur radial vom Ursprung abhängen sollte

- Objektraum: $h(x) = h_r(|x|)$
- Frequenzraum: $\hat{h}(k) = \hat{h}_r(|k|)$

Dies ist für diskrete Filter aufgrund der Rasterisierung des Definitionsbereiches jedoch nur annähernd möglich.

Rechtecksfilter

Das einfachste „Glättungsfilter“ ist das Rechtecksfilter.
Wir betrachten zunächst das Filter der Größe 1×3:

$$R = \frac{1}{3}[1\ \underline{1}\ 1]$$

Rechtecksfilter

Das einfachste „Glättungsfilter“ ist das Rechtecksfilter.
Wir betrachten zunächst das Filter der Größe 1×3:

$$R = \frac{1}{3}[1\ \underline{1}\ 1]$$

Die Berechnung der Transferfunktion des Filters ergibt:

$$\hat{R}(k) = \frac{1}{3} + \frac{2}{3}\cos(\pi k)\ \forall k \in [0,1]$$

Rechtecksfilter

Das einfachste „Glättungsfilter“ ist das Rechtecksfilter.
Wir betrachten zunächst das Filter der Größe 1×3:

$$R = \frac{1}{3}[1\ \underline{1}\ 1]$$

Die Berechnung der Transferfunktion des Filters ergibt:

$$\hat{R}(k) = \frac{1}{3} + \frac{2}{3}\cos(\pi k)\ \forall k \in [0,1]$$

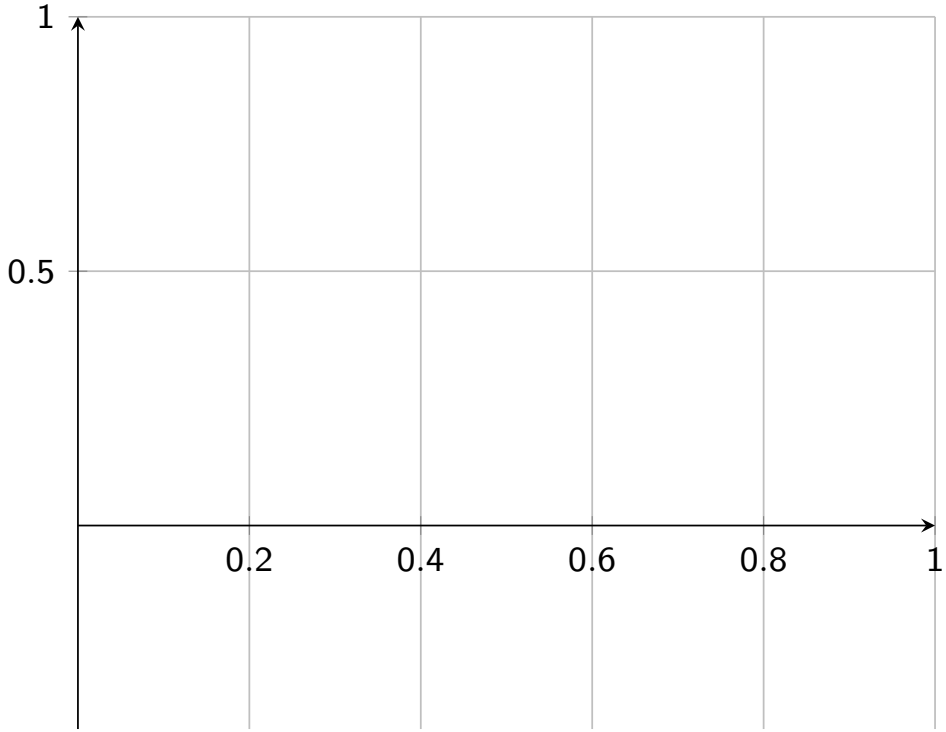
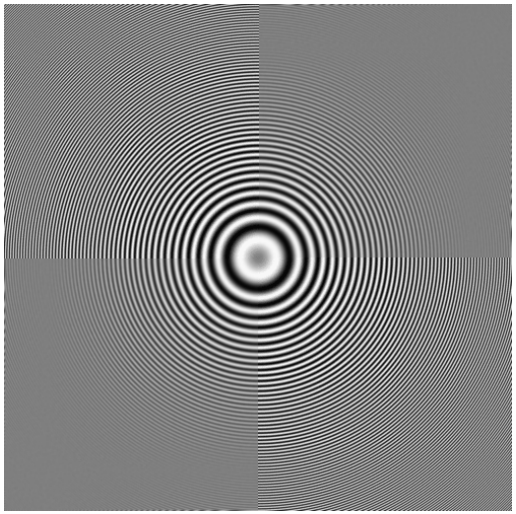
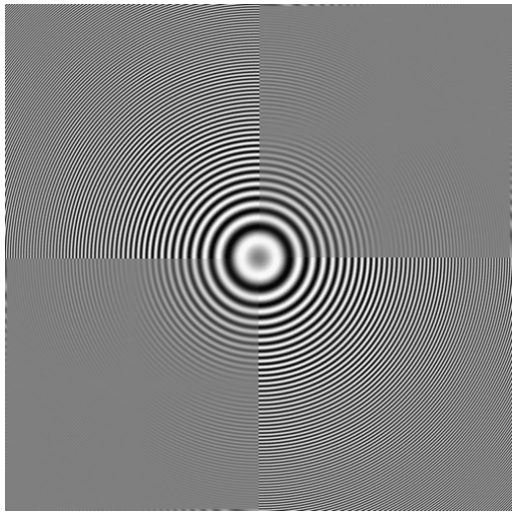
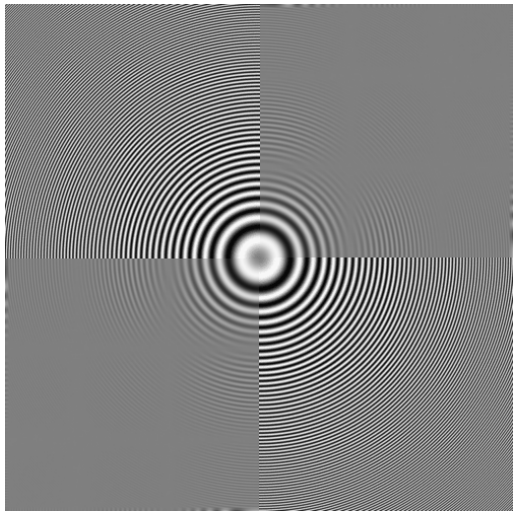


Abbildung: Transferfunktion des Rechteckfilters.

Aus dem Graphen entnehmen wir:

- Bei $k = \frac{2}{3}$ eine Nullstelle liegt.
  Es handelt sich um die Wellenlänge, bei der das Signal, das zweimal pro Wellenlänge abgetastet wird, bei der Anwendung des Filters vollständig eliminiert wird.
- Die kleinste Wellenlänge ($k = 1$) hingegen wird lediglich um Faktor $\frac{1}{3}$ gedämpft und negiert, also Minimum und Maximum vertauscht.
  Es ergibt sich also keine monoton steigende Dämpfung der hohen Wellenzahlen.

Rechteckfilter der Größen 3, 5 und 7:

Der Vorteil des Rechtecksfilters liegt in der Einfachheit. In 1D lässt sich der Filter unabhängig von seiner Größe mit nur 3 Operationen auf ein weiteres Pixel anwenden:

$$g'_m = g'_{m-1} + \frac{1}{2r+1}(g_{m+r} - g_{m-r-1})$$

Binomialfilter

Eine Analyse des Rechteckfilters ergibt, dass die Werte zur besseren Filterung zum Rand hin abfallen müssen. Binomialfilter erfüllen diese Bedingung:

Binomialfilter

Eine Analyse des Rechteckfilters ergibt, dass die Werte zur besseren Filterung zum Rand hin abfallen müssen. Binomialfilter erfüllen diese Bedingung:
Man beginnt in 1D mit

$$B = \frac{1}{2}[1\ 1]$$

und betrachtet die p-fache Faltung des Filters mit sich selbst:

$$B^p = \frac{1}{2^p}[1\ 1] * [1\ 1] * \ldots * [1\ 1]$$

Binomialfilter

Eine Analyse des Rechteckfilters ergibt, dass die Werte zur besseren Filterung zum Rand hin abfallen müssen. Binomialfilter erfüllen diese Bedingung:
Man beginnt in 1D mit

$$B = \frac{1}{2}[1\ 1]$$

und betrachtet die p-fache Faltung des Filters mit sich selbst:

$$B^p = \frac{1}{2^p}[1\ 1] * [1\ 1] * \ldots * [1\ 1]$$

Es ergibt sich:

$$\begin{aligned} B^2 &= \frac{1}{4}[1\ 2\ 1] \\ B^3 &= \frac{1}{8}[1\ 3\ 3\ 1] \\ B^4 &= \frac{1}{16}[1\ 4\ 6\ 4\ 1] \end{aligned}$$

Binomialfilter

Eine Analyse des Rechteckfilters ergibt, dass die Werte zur besseren Filterung zum Rand hin abfallen müssen. Binomialfilter erfüllen diese Bedingung:
Man beginnt in 1D mit

$$B = \frac{1}{2}[1\ 1]$$

und betrachtet die p-fache Faltung des Filters mit sich selbst:

$$B^p = \frac{1}{2^p}[1\ 1] * [1\ 1] * \ldots * [1\ 1]$$

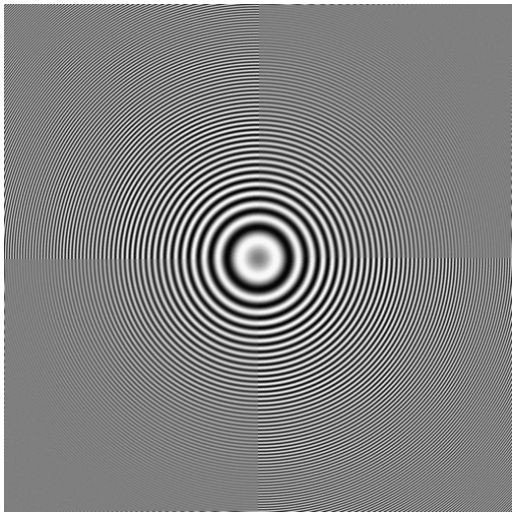
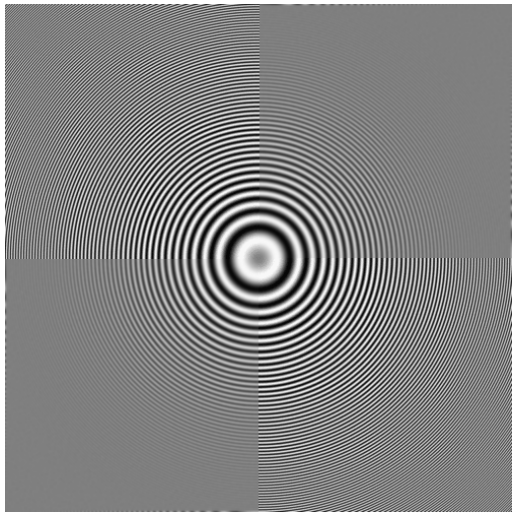
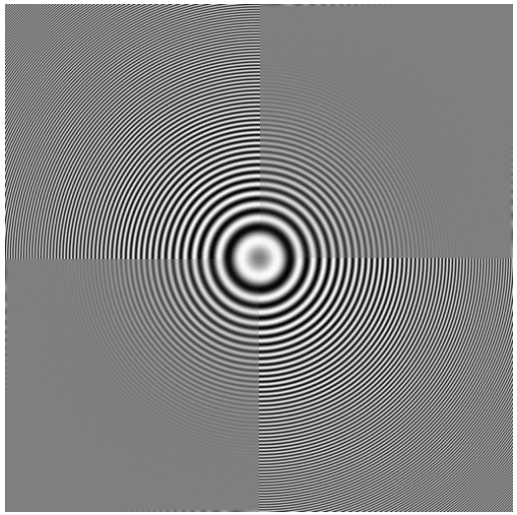
Es ergibt sich:

$$\begin{aligned} B^2 &= \frac{1}{4}[1\ 2\ 1] \\ B^3 &= \frac{1}{8}[1\ 3\ 3\ 1] \\ B^4 &= \frac{1}{16}[1\ 4\ 6\ 4\ 1] \end{aligned}$$

Um keine Verschiebung zu erhalten, faltet man abwechselnd mit

$$\begin{aligned} B_l^0 &= [\underline{1}\ 1] \text{ und} \\ B_r^0 &= [1\ \underline{1}] \end{aligned}$$

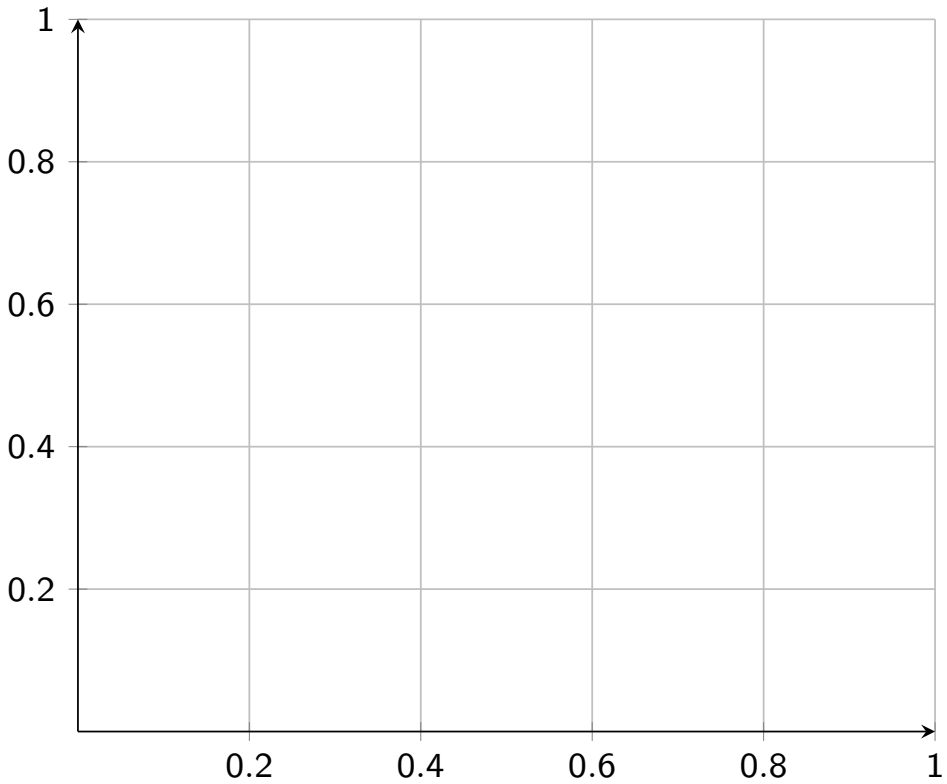
Binomialfilter der Breite 3 ($B^2 = \frac{1}{4}[1\ \underline{2}\ 1]$), 5 ($B^4 = \frac{1}{16}[1\ 4\ \underline{6}\ 4\ 1]$) und 7 ($B^6 = \frac{1}{64}[1\ 6\ 15\ \underline{20}\ 15\ 6\ 1]$):

Die Fouriertransformierte ist:

$$\hat{B}^p = \cos^p\left(\pi \cdot \frac{k}{2}\right)$$

Farben	p
Rot	1
Grün	2
Blau	3
Schwarz	4



Die mehrdimensionalen Binomialfilter ergeben sich aufgrund der Separabilität aus den 1D-Filtern:

$$
\begin{aligned}
B^2 &= B_x^2 * B_y^2 \\
&= \frac{1}{4}[1\ 2\ 1] * \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \\
&= \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}
\end{aligned}
$$

Die mehrdimensionalen Binomialfilter ergeben sich aufgrund der Separabilität aus den 1D-Filtern:

$$\begin{aligned} B^2 &= B_x^2 * B_y^2 \\ &= \frac{1}{4}[1\,2\,1] * \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \\ &= \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \end{aligned}$$

Für die Transferfunktion ergibt sich dann:

$$\begin{aligned} \hat{B} &= \hat{B}_x \hat{B}_y \\ &= \cos^p\left(\pi \cdot \frac{k_x}{2}\right) \cdot \cos^p\left(\pi \cdot \frac{k_y}{2}\right) \end{aligned}$$

Mehrschrittmittelung

Zur schnelleren Mittelung nimmt man manchmal auch größere, dünn besetzte Masken, wie z.B.:[a]

$$B_{x+y} = \frac{1}{4}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad B_{x-y} = \frac{1}{4}\begin{bmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

oder auch

$$B_{2x} = \frac{1}{4}[1\,0\,2\,0\,1], \ldots$$

[a] Jähne, Digitale Bildverarbeitung, 5. Aufl., Springer, 2001, Kap 11.6

Eigenschaften linearer Glättungsfilter

Vorteile linearer Filter:

- Sie dämpfen weißes Rauschen.

Eigenschaften linearer Glättungsfilter

Vorteile linearer Filter:

- Sie dämpfen weißes Rauschen.

Nachteile linearer Filter:

- Sie sind anfällig für binäres Rauschen (z.B. falsche Pixel).
- Sie verwischen Objektkanten.
- Sie erzeugen vorher nicht vorhandene Werte (Mittelung).

Rangordnungsfilter betrachten lediglich die Werte innerhalb der Maske und wählen einen Wert aufgrund einer **Sortierung** aus.

Rangordnungsfilter - Medianfilter

Rangordnungsfilter betrachten lediglich die Werte innerhalb der Maske und wählen einen Wert aufgrund einer **Sortierung** aus.
Der **Medianfilter** wählt innerhalb der Maske den mittleren Wert aus. (Nicht den Mittelwert!)

Definitionen:

- Eine **konstante Nachbarschaft** ist ein Bereich von $N+1$ gleichen Grauwerten.
- Eine **Kante** ist ein Bereich monoton steigender oder fallender Grauwerte zwischen zwei konstanten Nachbarschaften.
- Ein **Impuls** ist ein Bereich von höchstens N Punkten, der rechts und links von einer konstanten Nachbarschaft berandet ist.
- Ein **Fixpunkt** ist ein Grauwertsignal, das sich unter dem Medianfilter nicht ändert.

Medianfilter

Definitionen:

- Eine **konstante Nachbarschaft** ist ein Bereich von $N+1$ gleichen Grauwerten.
- Eine **Kante** ist ein Bereich monoton steigender oder fallender Grauwerte zwischen zwei konstanten Nachbarschaften.
- Ein **Impuls** ist ein Bereich von höchstens N Punkten, der rechts und links von einer konstanten Nachbarschaft berandet ist.
- Ein **Fixpunkt** ist ein Grauwertsignal, das sich unter dem Medianfilter nicht ändert.

Für Medianfilter gilt:

- Konstante Nachbarschaften und Kanten sind Fixpunkte.
- Impulse werden eliminiert.

Beispiel: Ableitungsfilter

- Aufgrund der Diskretisierung des Bildes ist eine Bestimmung von $\frac{\partial}{\partial x} g(x)$ direkt nicht möglich.
- Nimmt man jedoch die Herleitung der Ableitungsoperation, so erhält man für die rechtsseitige Ableitung

$$\frac{\partial}{\partial x} g(x) = lim_{\epsilon \to 0} \frac{g(x+\epsilon) - g(x)}{\epsilon}$$

- Zur Annäherung setzt man nun ϵ auf einen hinreichend kleinen Wert.

Beispiel: Ableitungsfilter

- Aufgrund der Diskretisierung des Bildes ist eine Bestimmung von $\frac{\partial}{\partial x} g(x)$ direkt nicht möglich.
- Nimmt man jedoch die Herleitung der Ableitungsoperation, so erhält man für die rechtsseitige Ableitung

$$\frac{\partial}{\partial x} g(x) = lim_{\epsilon \to 0} \frac{g(x+\epsilon) - g(x)}{\epsilon}$$

- Zur Annäherung setzt man nun ϵ auf einen hinreichend kleinen Wert.
- Im Falle digitaler Bilder ist der sinnvoll zu wählende Abstand der Punktabstand Δp.
- Damit erhält man als eine Möglichkeit, eine diskrete Ableitung zu berechnen:

$$\frac{\partial}{\partial x} g(x) = \frac{g(x+\Delta p) - g(x)}{\Delta p}$$

Beispiel: Ableitungsfilter

Betrachtet man die Operation genauer, so stellt man fest, dass es sich hierbei um die gewichtete Differenz zweier Pixelwerte handelt. Diesen Zusammenhang können wir auch in einer Kurznotation schreiben:

$$[1 \ \underline{-1}] \quad \text{oder} \quad [1 \ -1.]$$

Diese wird bezeichnet als

- Filtermaske
- Faltungsmaske

Beispiel: Ableitungsfilter

Betrachtet man die Operation genauer, so stellt man fest, dass es sich hierbei um die gewichtete Differenz zweier Pixelwerte handelt. Diesen Zusammenhang können wir auch in einer Kurznotation schreiben:

$$[1 \ \underline{-1}] \quad \text{oder} \quad [1 \ -1.]$$

Diese wird bezeichnet als

- Filtermaske
- Faltungsmaske

Die Position, an welcher das Ergebnis abgespeichert wird definiert durch:

- der Punkt nach der Zahl
- die Unterstreichung

Die Faktoren stehen relativ zu diesem Pixel in den benachbarten Feldern.

Verschiebungsinvarianz

Wir betrachten ein Bild, dass **periodisch** in alle Raumrichtungen ist

$$\forall a_1, a_2, \ldots, a_d \in \mathbb{Z}:$$

$$g_{i_1+a_1 N_1, i_2+a_2 N_2, \ldots, i_d+a_d N_d} = g_{i_1, \ldots, i_d}$$

Eine Verschiebung sei gegeben als

$$(S_{r_1, \ldots, r_d}(g))_{i_1, i_2, \ldots, i_d} = g_{i_1-r_1, i_2-r_2, \ldots, i_d-r_d}$$

Verschiebungsinvarianz

Wir betrachten ein Bild, dass **periodisch** in alle Raumrichtungen ist

$$\forall a_1, a_2, \ldots, a_d \in \mathbb{Z}:$$

$$g_{i_1+a_1 N_1, i_2+a_2 N_2, \ldots, i_d+a_d N_d} = g_{i_1,\ldots,i_d}$$

Eine Verschiebung sei gegeben als

$$(S_{r_1,\ldots,r_d}(g))_{i_1,i_2,\ldots,i_d} = g_{i_1-r_1,i_2-r_2,\ldots,i_d-r_d}$$

Dann gilt für einen **verschiebungsinvarianten Filter** $F : \mathbb{Z}^d \to W$:

$$(S_{r_1,\ldots,r_d}(F(g)))_{i_1,i_2,\ldots,i_d} = F(g_{i_1-r_1,i_2-r_2,\ldots,i_d-r_d})$$

Verschiebungsinvarianz

Wir betrachten ein Bild, dass **periodisch** in alle Raumrichtungen ist

$$\forall a_1, a_2, \ldots, a_d \in \mathbb{Z}:$$

$$g_{i_1+a_1 N_1, i_2+a_2 N_2, \ldots, i_d+a_d N_d} = g_{i_1,\ldots,i_d}$$

Eine Verschiebung sei gegeben als

$$(S_{r_1,\ldots,r_d}(g))_{i_1,i_2,\ldots,i_d} = g_{i_1-r_1,i_2-r_2,\ldots,i_d-r_d}$$

Dann gilt für einen **verschiebungsinvarianten Filter** $F : \mathbb{Z}^d \to W$:

$$(S_{r_1,\ldots,r_d}(F(g)))_{i_1,i_2,\ldots,i_d} = F(g_{i_1-r_1,i_2-r_2,\ldots,i_d-r_d})$$

Dies ist gleichbedeutend mit der Eigenschaft, dass der Filter an jedem Bildpunkt die gleiche Operation in Abhängigkeit der Nachbarn ausführt.
Verschiebungsinvariante Filter lassen sich also recht einfach durch eine Rechenregel relativ zu einem Referenzpunkt beschreiben, ohne dass sie die Position des Punktes berücksichtigen.

Lineare Filter

Wie jede lineare Abbildung nennen wir einen Filter **linear**, wenn sich die Abbildung durch lineare Kombination der Basisvektoren definieren lässt.

Wie jede lineare Abbildung nennen wir einen Filter **linear**, wenn sich die Abbildung durch lineare Kombination der Basisvektoren definieren lässt.
Für **lineare Filter** gilt dann unter anderem:

$$F\{ax[n] + by[n]\} = aF\{x[n]\} + bF\{y[n]\}$$

Verschiebungsinvarianz

Wenden wir also zum Beispiel im Eindimensionalen auf das Bild

$$[0\ 0\ 0\ 0\ \underline{1}\ 0\ 0\ 0\ 0]$$

ein lineares verschiebungsinvariantes Filter an und erhalten die „Antwort“

$$[0\ 1\ 2\ 3\ \underline{4}\ 1\ 1\ 1\ 0]$$

Verschiebungsinvarianz

Wenden wir also zum Beispiel im Eindimensionalen auf das Bild

$$[0\ 0\ 0\ 0\ \underline{1}\ 0\ 0\ 0\ 0]$$

ein lineares verschiebungsinvariantes Filter an und erhalten die „Antwort“

$$[0\ 1\ 2\ 3\ \underline{4}\ 1\ 1\ 1\ 0]$$

so können wir aus der Verschiebungsinvarianz schließen, dass die Antwort auf

$$[0\ 0\ 0\ 0\ \underline{0}\ 1\ 0\ 0\ 0]$$

Wenden wir also zum Beispiel im Eindimensionalen auf das Bild

$$[0\ 0\ 0\ 0\ \underline{1}\ 0\ 0\ 0\ 0]$$

ein lineares verschiebungsinvariantes Filter an und erhalten die „Antwort“

$$[0\ 1\ 2\ 3\ \underline{4}\ 1\ 1\ 1\ 0]$$

so können wir aus der Verschiebungsinvarianz schließen, dass die Antwort auf

$$[0\ 0\ 0\ 0\ \underline{0}\ 1\ 0\ 0\ 0]$$

die um einen Pixel verschobene Antwort

$$[0\ 0\ 1\ 2\ \underline{3}\ 4\ 1\ 1\ 1]$$

ist.

Gleichzeitig können wir aufgrund der Linearität schließen, dass sich die Antwort auf

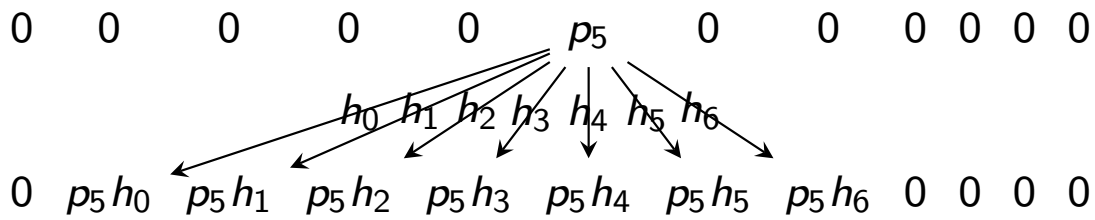
$$[0\ 0\ 0\ 0\ \underline{2}\ -1\ 0\ 0\ 0\ ]$$

aus der Linearkombination der beiden Antworten von oben ergibt, also

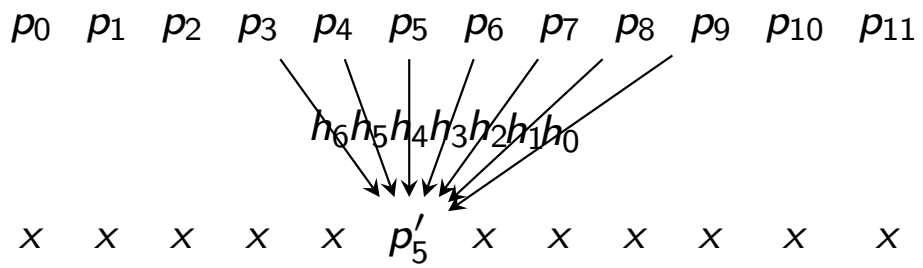
$$\begin{array}{lll} & 2 & \cdot\ [0\ 1\ 2\ 3\ \underline{4}\ 1\ 1\ 1\ 0] \\ + & (-1) & \cdot\ [0\ 0\ 1\ 2\ \underline{3}\ 4\ 1\ 1\ 1] \\ = & & [0\ 2\ 3\ 4\ \underline{5}\ -2\ 1\ 1\ -1] \end{array}$$

ist.

Man kann also den Einfluss von Pixeln des Originalbilds auf das Ergebnisbild mit der Filterantwort [h_0 h_1 h_2 h_3 $\underline{h_4}$ h_5 h_6] wie folgt zusammenfassen:



Umgekehrt ist der Einfluss auf ein Ergebnispixel darstellbar als:



Somit lässt sich ein linearer verschiebungsinvarianter Filter (linear shift-invariant filter, LSI-Filter) **eineindeutig** über seine Punktantwort beschreiben. Die Punktantwort ist dabei das Ergebnis der Anwendung des Filters auf den Dirac-Impuls bzw. das Kronecker-Delta.

- Kronecker-Delta (diskret)

$$\delta : x \mapsto \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases}$$

Somit lässt sich ein linearer verschiebungsinvarianter Filter (linear shift-invariant filter, LSI-Filter) **eineindeutig** über seine Punktantwort beschreiben. Die Punktantwort ist dabei das Ergebnis der Anwendung des Filters auf den Dirac-Impuls bzw. das Kronecker-Delta.

- Kronecker-Delta (diskret)

$$\delta : x \mapsto \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases}$$

- Dirac-Delta (kontinuierlich):

$$\begin{array}{rcccl} \delta & : & \mathbb{R} & \rightarrow & \mathbb{R} \cup \{\infty\} \\ \delta & : & x & \mapsto & \begin{cases} 0 & x \neq 0 \\ \infty & x = 0 \end{cases} \end{array}$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

Somit lässt sich ein linearer verschiebungsinvarianter Filter (linear shift-invariant filter, LSI-Filter) **eineindeutig** über seine Punktantwort beschreiben. Die Punktantwort ist dabei das Ergebnis der Anwendung des Filters auf den Dirac-Impuls bzw. das Kronecker-Delta.

- Kronecker-Delta (diskret)

$$\delta : x \mapsto \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases}$$

- Dirac-Delta (kontinuierlich):

$$\begin{array}{cccc} \delta & : & \mathbb{R} & \rightarrow & \mathbb{R} \cup \{\infty\} \\ \delta & : & x & \mapsto & \begin{cases} 0 & x \neq 0 \\ \infty & x = 0 \end{cases} \end{array}$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

- Physiker:

$$\int_{-\infty}^{\infty} \delta(x) \cdot f(x) dx = f(0)$$

Faltung zweier Bilder

Sei g das Ausgangsbild und h die Punkt-antwort. Dann schreibt man

$$(g * h)(x) = \sum_t g(t)h(x-t)$$

und

$$(g * h)(x) = \int_{-\infty}^{\infty} g(t)h(x-t)dt$$

wobei $*$ der Faltungsoperator ist. Die Punktantwort wird dementsprechend auch Faltungskern genannt.

In der Bildverarbeitung ist jedoch häufig schon alleine aus Gründen der Anschaulichkeit eine andere Darstellung üblich. Man verwendet hierbei die gespiegelte Punktantwort $h'(x) = h(-x)$ und definiert die **Korrelation** von g und h' als

$$(g \star h')(x) = \sum_t g(t)h'(x+t)$$

wobei $\star$ der Korrelationsoperator ist. Im Kontinuierlichen analog dazu

$$(g \star h')(x) = \int_{-\infty}^{\infty} g(t)h'(x+t)dt$$

In der Bildverarbeitung ist jedoch häufig schon alleine aus Gründen der Anschaulichkeit eine andere Darstellung üblich. Man verwendet hierbei die gespiegelte Punktantwort $h'(x) = h(-x)$ und definiert die **Korrelation** von g und h' als

$$(g \star h')(x) = \sum_t g(t)h'(x+t)$$

wobei $\star$ der Korrelationsoperator ist. Im Kontinuierlichen analog dazu

$$(g \star h')(x) = \int_{-\infty}^{\infty} g(t)h'(x+t)dt$$

Beispiel Berechnung des 5ten Pixels:

$$\begin{array}{rcccccccccc} g & = & [0 & 0 & 0 & 0 & 2 & -1 & 0 & 0 & 0] \\ h' & = & [0 & 1 & 1 & 1 & \underline{4} & 3 & 2 & 1 & 0] \\ & & [0 & 0 & 0 & 0 & 8 & -3 & 0 & 0 & 0] \end{array}$$

$$(g \star h')(5) = 5$$

Analog dazu ergibt sich für das sechste Pixel:

$$\begin{array}{rcccccccccc} g & = & [0 & 0 & 0 & 0 & 2 & -1 & 0 & 0 & 0] \\ h' & = & [0 & 0 & 1 & 1 & 1 & \underline{4} & 3 & 2 & 1] \\ & & [0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0] \end{array}$$

$$(g \star h')(6) = -2$$

2D Lineare Filter

Ein linearer Filter ist gegeben durch seinen „Filterkern“

- Dies ist eine kleine Matrix aus Koeffizienten
- Filterkerne haben prinzipiell eine beliebige Größe
- Aus Symmetriegründen werden sie in der Regel mit ungerader Kantenlänge und quadratisch angesetzt.
- Operation eines Linearen Filters in 2D:
- Man dreht den Filterkern h(x,y) um 180 Grad und legt ihn über das Originalbild g(x,y)
- Man multipliziert die Werte im Filterkern mit den Pixeln
- Man addiert alle diese Ergebnisse, dies ist der neue Grauwert für das zentrale Pixel
- Der Filterkern wird verschoben und alle anderen neuen Pixelwerte genauso berechnet
- Diese Operation heißt Faltung des Bildes g mit dem Kern h
- $g * h$

Häufige Aufgabe: Glätten eines Bildes, also entfernen von hochfrequenten Störungen.

- Entsprechende Filter werden Tiefpass-Filter genannt
- Beispiel Mittelwertfilter (Box Filter):
- Der Filterkern besteht aus Einsen mit einem Normierungsfaktor multipliziert.
- Allgemeine Beobachtung bei linearen Tiefpass-Filtern: Rauschen wird eliminiert, aber Kanten werden verwaschen. Je größer der Filterkern ist, desto ausgeprägter ist der Effekt

$$\frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Abbildung: 5x5 Boxplot Filter

Nachteil Mittelwertfilter: zentraler Pixel spielt gleiche Rolle wie alle anderen Pixel

- Pixel sollten weniger wichtig sein wenn sie weiter vom zentralen Pixel entfernt liegen
- Gauß-Filter, basiert auf Gaußfunktion (wird auch Normalverteilung genannt)
- 1D Gaußfunktion: $F(x) = \frac{1}{\sqrt{2\pi * \sigma^2}} e^{-\frac{x-\Sigma}{2\sigma^2}}$
- Σ entscheidet über die Position der Amplitude
- σ entscheidet über die Breite der Kurve

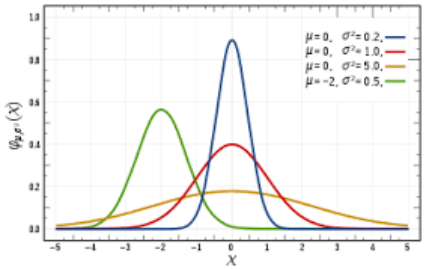


Abbildung: Gausfunktionen mit verschiedenen σ

Approximation des Gaußfilters durch Pacalsches Dreieck

- je nach größe des Gaußfilters muss die entsprechende Zeile des Pascalschen Dreiecks herangezogen werden

$$
\begin{array}{ccccccc}
 & & & 1 & & & \\
 & & 1 & & 1 & & \\
 & & 1 & 2 & 1 & & \\
 & 1 & 3 & 3 & 1 & & \\
 & 1 & 4 & 6 & 4 & 1 & \\
 & 1 & 5 & 10 & 10 & 5 & 1 \\
1 & 6 & 15 & 20 & 15 & 6 & 1
\end{array}
\quad 1/64 \quad
\begin{bmatrix} 1 \\ 6 \\ 15 \\ 20 \\ 15 \\ 6 \\ 1 \end{bmatrix}
\ * \
\begin{bmatrix} 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \end{bmatrix}
\quad 1/64
$$

$$
\begin{bmatrix}
1 & 6 & 15 & 20 & 15 & 6 & 1 \\
6 & 36 & 90 & 120 & 90 & 36 & 6 \\
15 & 90 & 225 & 300 & 225 & 90 & 15 \\
20 & 120 & 300 & 400 & 300 & 120 & 20 \\
15 & 90 & 225 & 300 & 225 & 90 & 15 \\
6 & 36 & 90 & 120 & 90 & 36 & 6 \\
1 & 6 & 15 & 20 & 15 & 6 & 1
\end{bmatrix}
\quad 1/4096
$$

Abbildung: Gausfilter durch Pascalsches Dreieck

Beis

- je nach größe des Gaußfilters muss die entsprechende Zeile des Pascalschen Dreiecks herangezogen werden

$$
\begin{array}{ccccccc}
 & & & 1 & & & \\
 & & 1 & & 1 & & \\
 & & 1 & 2 & 1 & & \\
 & 1 & 3 & 3 & 1 & & \\
 & 1 & 4 & 6 & 4 & 1 & \\
 & 1 & 5 & 10 & 10 & 5 & 1 \\
1 & 6 & 15 & 20 & 15 & 6 & 1
\end{array}
\quad 1/64 \quad
\begin{bmatrix} 1 \\ 6 \\ 15 \\ 20 \\ 15 \\ 6 \\ 1 \end{bmatrix}
\ast
\begin{bmatrix} 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \end{bmatrix}
\quad 1/64
$$

$$
\begin{bmatrix}
1 & 6 & 15 & 20 & 15 & 6 & 1 \\
6 & 36 & 90 & 120 & 90 & 36 & 6 \\
15 & 90 & 225 & 300 & 225 & 90 & 15 \\
20 & 120 & 300 & 400 & 300 & 120 & 20 \\
15 & 90 & 225 & 300 & 225 & 90 & 15 \\
6 & 36 & 90 & 120 & 90 & 36 & 6 \\
1 & 6 & 15 & 20 & 15 & 6 & 1
\end{bmatrix}
\quad 1/4096
$$

Abbildung: Gausfilter durch Pascalsches Dreieck

Teil V

Fourier Transformation

Ein Vektorraum über einem Körper $(K, +, \cdot)$ oder kurz K-Vektorraum ist eine additive abelsche Gruppe $(V, +)$, auf der die Multiplikation mit einem Skalar aus K definiert ist

$$\cdot : K \times V \to V$$

wobei für alle $u, v \in V$ und $\alpha, \beta \in K$ gilt

- $\alpha \cdot (\beta \cdot v) = (\alpha \cdot \beta) \cdot v$
- $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$
- $(\alpha + \beta) \cdot v = \alpha \cdot v + \beta \cdot v$
- $1_K \cdot v = v$

Bilder als Elemente von Vektorräumen

Man kann 2D-Bilder als $N_1 \times N_2$-dimensionale reell-wertige oder komplex-wertige Vektorräume (für $K = \mathbb{R}$ oder $K = \mathbb{C}$) auffassen.
Wir definieren $N_1 \times N_2$ Basisbilder ${}^{m,n}P$ durch

$$ {}^{m,n}P_{i,j} := \begin{cases} 1 & i = m \wedge j = n \\ 0 & \text{sonst} \end{cases} $$

Dies sind die Bilder, welche genau einen Bildpunkt mit dem Wert Eins und sonst ausschließlich Punkte mit dem Wert Null enthalten.

Man kann 2D-Bilder als $N_1 \times N_2$-dimensionale reell-wertige oder komplex-wertige Vektorräume (für $K = \mathbb{R}$ oder $K = \mathbb{C}$) auffassen.
Wir definieren $N_1 \times N_2$ Basisbilder ${}^{m,n}P$ durch

$$ {}^{m,n}P_{i,j} := \begin{cases} 1 & i = m \wedge j = n \\ 0 & \text{sonst} \end{cases} $$

Dies sind die Bilder, welche genau einen Bildpunkt mit dem Wert Eins und sonst ausschließlich Punkte mit dem Wert Null enthalten.
Definiert man nun das Skalarprodukt auf zwei Bildern über

$$ \langle P, Q \rangle = \sum_{i=0}^{N_0-1} \sum_{j=0}^{N_1-1} p_{i,j} \cdot q_{i,j} $$

so ist offensichtlich, dass die genannte Basis eine Orthonormalbasis ist, d.h., das Skalarprodukt zweier Basisvektoren ist eins gdw. die beiden Vektoren gleich sind, sonst null.

Gleichzeitig ist offensichtlich, dass sich jedes Bild als Linearkombination der genannten Basisvektoren darstellen lässt, denn es gilt:

$$V_{i,j} = \sum_{m=0}^{N_0-1} \sum_{n=0}^{N_1-1} {}^{m,n}P_{i,j} \cdot V_{i,j}$$

Da man die Basis prinzipiell beliebig wählen kann, kann man darüber nachdenken, die Basis so zu wählen, dass sie die Lösung bestimmter Probleme vereinfacht. Eine solche Basis ist die Basis der **periodischen Muster**.

Da man die Basis prinzipiell beliebig wählen kann, kann man darüber nachdenken, die Basis so zu wählen, dass sie die Lösung bestimmter Probleme vereinfacht. Eine solche Basis ist die Basis der **periodischen Muster**.
Die Transformation in die neue Basis verhält sich analog zur Transformation von Vektoren auf eine neue Basis: Man misst mit Hilfe des Skalarproduktes den Beitrag, den ein Vektor in „Richtung“ des neuen Basisvektors hat.

Zuerst definieren wir uns hierzu eine neue Basis über die Beschreibung ihrer Basisvektoren. Wir würden gerne Basisvektoren der Form $\cos(2\pi\omega \cdot x - \varphi)$ betrachten, mit

- Frequenz: $2\pi\omega$
- Phasenverschiebung: φ

Zuerst definieren wir uns hierzu eine neue Basis über die Beschreibung ihrer Basisvektoren. Wir würden gerne Basisvektoren der Form $\cos(2\pi\omega \cdot x - \varphi)$ betrachten, mit

- Frequenz: $2\pi\omega$
- Phasenverschiebung: φ

Um die Phasenverschiebung einfacher ausdrücken zu können, benutzen wir die komplexen Exponentialfunktionen, für die gilt

$$\begin{aligned} e^{i2\pi\omega} &= \cos(2\pi\omega) + i\sin(2\pi\omega) \\ e^{-i2\pi\omega} &= \cos(2\pi\omega) - i\sin(2\pi\omega) \end{aligned}$$

und folglich

$$\begin{aligned} e^{i2\pi\omega} + e^{-i2\pi\omega} &= 2 \cdot \cos(2\pi\omega) \\ &\Updownarrow \\ \frac{1}{2}\left(e^{i2\pi\omega} + e^{-i2\pi\omega}\right) &= \cos(2\pi\omega) \end{aligned}$$

Die Anzahl der Schwingungen pro Einheitslänge $2 \cdot \pi$ wird **Wellenzahl**

$$|k| = \frac{2 \cdot \pi}{\lambda}$$

genannt.
Hierbei ist λ die Wellenlänge.

Die Anzahl der Schwingungen pro Einheitslänge $2 \cdot \pi$ wird **Wellenzahl**

$$|k| = \frac{2 \cdot \pi}{\lambda}$$

genannt.
Hierbei ist λ die Wellenlänge.
Sei $\vec{k}$ der zugehörige **komplexe Wellenzahlvektor**.
Dann kann man zweidimensionale komplex-wertige Bilder mit Wellenmuster über

$$p(x) = \mathrm{e}^{i2\pi\langle k,x\rangle}$$

definieren.

Sei

- φ die Phasenverschiebung
- r die Amplitude

Dann erhält man für reell-wertige Bilder

$$\begin{aligned} p(x) &= \frac{r}{2}\left(\mathrm{e}^{i(2\pi\langle k,x\rangle-\varphi)}+\mathrm{e}^{-i(2\pi\langle k,x\rangle-\varphi)}\right) \\ &= r\cdot\cos(2\pi\langle k,x\rangle-\varphi) \end{aligned}$$

Dadurch entstehen Muster der Wellenlänge $\lambda = \frac{1}{|k|}$ deren Wellenkämme senkrecht zu k verlaufen.

Beispiel

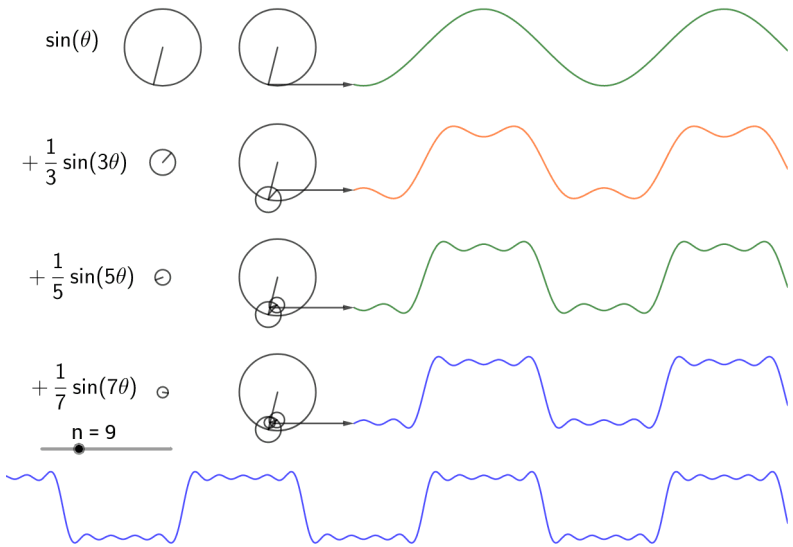


Abbildung: Rekonstruktion eines Signals mit Hilfe von Sinuswellen

Sei $g : \mathbb{R} \to \mathbb{C}$ ein quadratintegrables analoges Bild (Signal), d.h.

$$\int_{-\infty}^{\infty} |g(x)^2| dx < \infty$$

Dann ist die Fouriertransformierte von g die Funktion $\hat{g} : \mathbb{R} \to \mathbb{C}$ mit

$$\hat{g}(k) = \int_{-\infty}^{\infty} g(x) \mathrm{e}^{-i \cdot 2\pi kx} dx$$

$g \to \hat{g}$ bildet den Vektorraum der quadratintegrablen Funktionen auf sich selbst ab.

Sei $g : \mathbb{R} \to \mathbb{C}$ ein quadratintegrables analoges Bild (Signal), d.h.

$$\int_{-\infty}^{\infty} |g(x)^2| dx < \infty$$

Dann ist die Fouriertransformierte von g die Funktion $\hat{g} : \mathbb{R} \to \mathbb{C}$ mit

$$\hat{g}(k) = \int_{-\infty}^{\infty} g(x) \mathrm{e}^{-i \cdot 2\pi kx} dx$$

$g \to \hat{g}$ bildet den Vektorraum der quadratintegrablen Funktionen auf sich selbst ab.
Die inverse Transformation lautet

$$g(x) = \int_{-\infty}^{\infty} \hat{g}(k) \mathrm{e}^{i \cdot 2\pi kx} dk$$

Fouriertransformation als Skalarprodukt

Sei $\bar{g}$ die komplex-konjugierte Version von g. Das Skalarprodukt zweier komplexer Funktionen wird definiert als

$$\langle g(x), h(x) \rangle := \int_{-\infty}^{\infty} \bar{g}(x) h(x) dx$$

Setzt man nun $w = \mathrm{e}^{-i \cdot 2\pi}$, so kann man die Fouriertransformation als Skalarprodukt schreiben

$$\begin{aligned} \hat{g}(k) &= \left\langle w^{kx}, g(x) \right\rangle \\ g(x) &= \left\langle w^{-kx}, \hat{g}(k) \right\rangle \end{aligned}$$

Fouriertransformation als Skalarprodukt

Sei $\bar{g}$ die komplex-konjugierte Version von g. Das Skalarprodukt zweier komplexer Funktionen wird definiert als

$$\langle g(x), h(x) \rangle := \int_{-\infty}^{\infty} \bar{g}(x) h(x) dx$$

Setzt man nun $w = \mathrm{e}^{-i \cdot 2\pi}$, so kann man die Fouriertransformation als Skalarprodukt schreiben

$$\begin{aligned} \hat{g}(k) &= \left\langle w^{kx}, g(x) \right\rangle \\ g(x) &= \left\langle w^{-kx}, \hat{g}(k) \right\rangle \end{aligned}$$

In der Literatur finden sich auch die Schreibweisen

$$g(x) \circ\!\!-\!\!- \hat{g}(k)$$

oder

$$g(x) \circ\!\!-\!\!\bullet \hat{g}(k)$$

wobei letztere manchmal auch für die Laplacetransformation bezeichnet.

Da wir diskrete Bilder bearbeiten wollen, benötigen wird die **diskrete Fouriertransformation**.
Wir nehmen also ein komplexwertiges 1D Bild, gegeben durch den Vektor

$$g = (g_0, g_1, \ldots, g_{N-1})^T$$

Setzt man das Bild periodisch fort

$$g = (\ldots, g_{N-2}, g_{N-1}, g_0, g_1, \ldots, g_{N-1}, g_0, g_1, \ldots)^T$$

so sind nur periodische Basisfunktionen von Interesse.

Von allen möglichen Basisfunktionen

$$\begin{aligned} b_\omega(t) &= e^{-i2\pi\omega t} \\ &= \cos(2\pi\omega t) - i\sin(2\pi\omega t) \end{aligned}$$

sind also nur die interessant, welche auf dem gewählten Bild periodisch sind.

Von allen möglichen Basisfunktionen

$$\begin{aligned} b_\omega(t) &= e^{-i2\pi\omega t} \\ &= \cos(2\pi\omega t) - i\sin(2\pi\omega t) \end{aligned}$$

sind also nur die interessant, welche auf dem gewählten Bild periodisch sind.
Nimmt man $x = 1$ als Periode des Bildes, so interessieren nur die Basisfunktionen mit $\omega \in \mathbb{Z}$.

Die **diskrete, eindimensionale Fouriertransformation (1D-DFT)** bildet das **periodische, diskrete, eindimensionale Bild**

$$g = (g_0, g_1, \ldots, g_{N-1})^T$$

auf den Vektor $\hat{g}$ der gleichen Dimension N ab:

$$\mathrm{DFT}_N : g \mapsto \hat{g}$$

$$\hat{g}_k = \frac{1}{N} \sum_{n=0}^{N-1} \mathrm{e}^{-i \cdot 2\pi \frac{nk}{N}} g_n$$

$$k = 0, \ldots, N-1$$

Die **diskrete, eindimensionale Fouriertransformation (1D-DFT)** bildet das **periodische, diskrete, eindimensionale Bild**

$$g = (g_0, g_1, \ldots, g_{N-1})^T$$

auf den Vektor $\hat{g}$ der gleichen Dimension N ab:

$$\mathrm{DFT}_N : g \mapsto \hat{g}$$

$$\hat{g}_k = \frac{1}{N} \sum_{n=0}^{N-1} \mathrm{e}^{-i \cdot 2\pi \frac{nk}{N}} g_n$$

$$k = 0, \ldots, N-1$$

Es ergibt sich die Rücktransformation

$$g_n = \sum_{k=0}^{N-1} \mathrm{e}^{i \cdot 2\pi \frac{nk}{N}} \hat{g}_k, \quad n = 0, \ldots, N-1$$

Die **diskrete, eindimensionale Fouriertransformation (1D-DFT)** bildet das **periodische, diskrete, eindimensionale Bild**

$$g = (g_0, g_1, \ldots, g_{N-1})^T$$

auf den Vektor $\hat{g}$ der gleichen Dimension N ab:

$$\mathrm{DFT}_N : g \mapsto \hat{g}$$

$$\hat{g}_k = \frac{1}{N} \sum_{n=0}^{N-1} \mathrm{e}^{-i \cdot 2\pi \frac{nk}{N}} g_n$$

$$k = 0, \ldots, N-1$$

Mit

$$w_N = w^{\frac{1}{N}} = \mathrm{e}^{\frac{i \cdot 2\pi}{N}}$$

und den Basisvektoren

$$b_k = \frac{1}{N} \left[w_N^0, w_N^k, w_N^{2k}, \ldots, w_N^{(N-1)k} \right]$$

sowie dem üblichen Skalarprodukt ergibt sich die DFT als Skalarprodukt zu

$$\hat{g}_k = \langle b_k, g \rangle$$

Die 1D Bilder b_k für $k = 0, \ldots, N-1$ bilden hierbei eine orthogonale Basis des Vektorraums.
Ferner ist der Basisvektor

$$b_0 = \frac{1}{N}[1, 1, 1, \ldots, 1]$$

und daher der 0-te Koeffizient

$$\hat{g}_0 = \langle b_0, g \rangle$$

der Mittelwert des 1D Bildes g.

Die 1D Bilder b_k für $k = 0, \ldots, N-1$ bilden hierbei eine orthogonale Basis des Vektorraums.
Ferner ist der Basisvektor

$$b_0 = \frac{1}{N}[1, 1, 1, \ldots, 1]$$

und daher der 0-te Koeffizient

$$\hat{g}_0 = \langle b_0, g \rangle$$

der Mittelwert des 1D Bildes g.
Als Unterscheidung zur kontinuierlichen Fouriertransformation gibt es hierzu die Kurzschreibweise

$$g_i \overset{N}{\longmapsto} \hat{g}_k$$

Die mehrdimensionale Fouriertransformation für eine quadratintegrable Funktion $g : \mathbb{R}^d \to \mathbb{C}$ ist analog zur eindimensionalen Formulierung definiert.
Sei

$$\int_{\mathbb{R}^d} |g(x)^2| dx = \|g\|^2 < \infty$$

Die mehrdimensionale Fouriertransformation für eine quadratintegrable Funktion $g : \mathbb{R}^d \to \mathbb{C}$ ist analog zur eindimensionalen Formulierung definiert.
Sei

$$\int_{\mathbb{R}^d} |g(x)^2| dx = \|g\|^2 < \infty$$

Dann besitzt $g(x)$ die Fouriertransformierte

$$\hat{g} : \mathbb{R}^d \to \mathbb{C}$$

mit

$$\hat{g}(k) = \int_{\mathbb{R}^d} g(x) e^{-i \cdot 2\pi \langle k,x \rangle} dx = \left\langle w^{\langle k,x \rangle}, g(x) \right\rangle$$

und der inversen Transformation

$$g(x) = \int_{\mathbb{R}^d} \hat{g}(k) e^{i \cdot 2\pi \langle x,k \rangle} dk = \left\langle w^{-\langle k,x \rangle}, \hat{g}(k) \right\rangle$$

Beispiel: 2D-DFT

Für zweidimensionale, diskrete Bilder der Größe $N_1 \times N_2$ erhält man die 2D-DFT

$$\hat{p}_{\mu \cdot \nu} := \frac{1}{N_1 N_2} \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} p_{mn} \cdot e^{-i \cdot 2\pi \left(\frac{m \cdot \mu}{N_1} + \frac{n \cdot \nu}{N_2} \right)}$$

Beispiel: 2D-DFT

Für zweidimensionale, diskrete Bilder der Größe $N_1 \times N_2$ erhält man die 2D-DFT

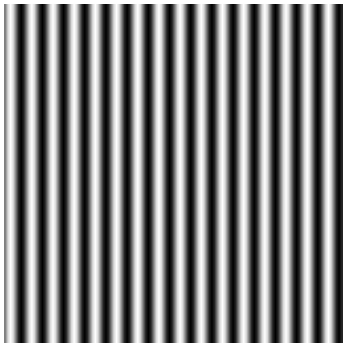
$$\begin{aligned} \hat{p}_{\mu \cdot \nu} &:= \frac{1}{N_1 N_2} \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} p_{mn} \cdot e^{-i \cdot 2\pi \left(\frac{m \cdot \mu}{N_1} + \frac{n \cdot \nu}{N_2} \right)} \\ &= \frac{1}{N_1 N_2} \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} p_{mn} \cdot e^{-i \cdot 2\pi \frac{m \cdot \mu}{N_1}} \cdot e^{-i \cdot 2\pi \frac{n \cdot \nu}{N_2}} \end{aligned}$$

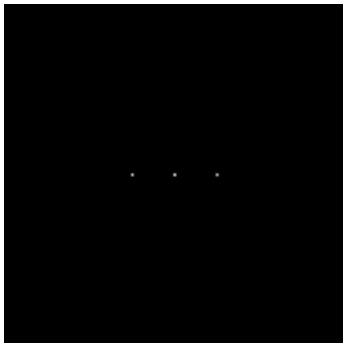
Für zweidimensionale, diskrete Bilder der Größe $N_1 \times N_2$ erhält man die 2D-DFT

$$
\begin{aligned}
\hat{p}_{\mu\cdot\nu} &:= \frac{1}{N_1 N_2} \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} p_{mn} \cdot e^{-i\cdot 2\pi \left(\frac{m\cdot\mu}{N_1} + \frac{n\cdot\nu}{N_2} \right)} \\
&= \frac{1}{N_1 N_2} \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} p_{mn} \cdot e^{-i\cdot 2\pi \frac{m\cdot\mu}{N_1}} \cdot e^{-i\cdot 2\pi \frac{n\cdot\nu}{N_2}} \\
&= \frac{1}{N_1} \sum_{m=0}^{N_1-1} \left(\frac{1}{N_2} \sum_{n=0}^{N_2-1} p_{mn} \cdot e^{-i\cdot 2\pi \frac{n\cdot\nu}{N_2}} \right) \cdot e^{-i\cdot 2\pi \frac{m\cdot\mu}{N_1}}
\end{aligned}
$$

Beispiele Fouriertransformation

Sinus

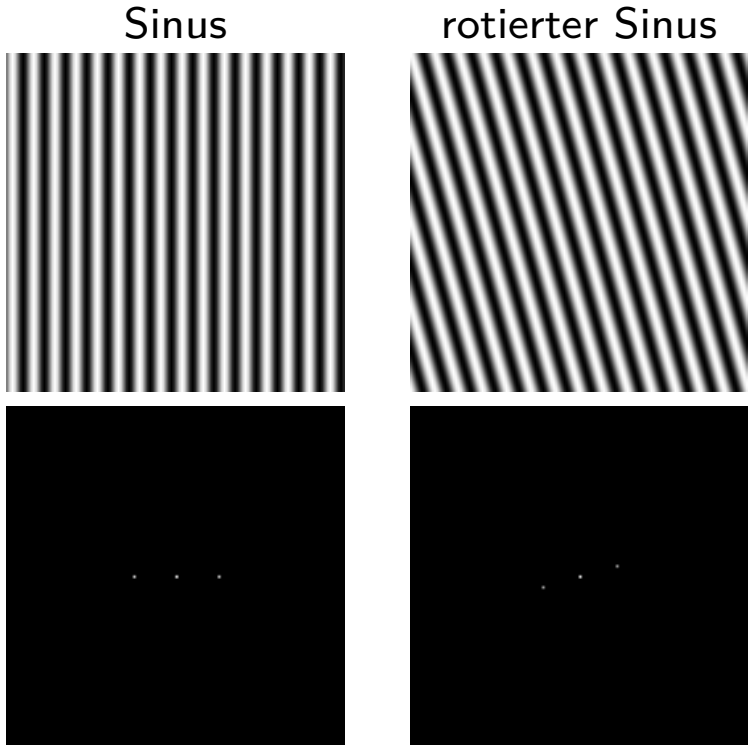




Fourier-Transformierte

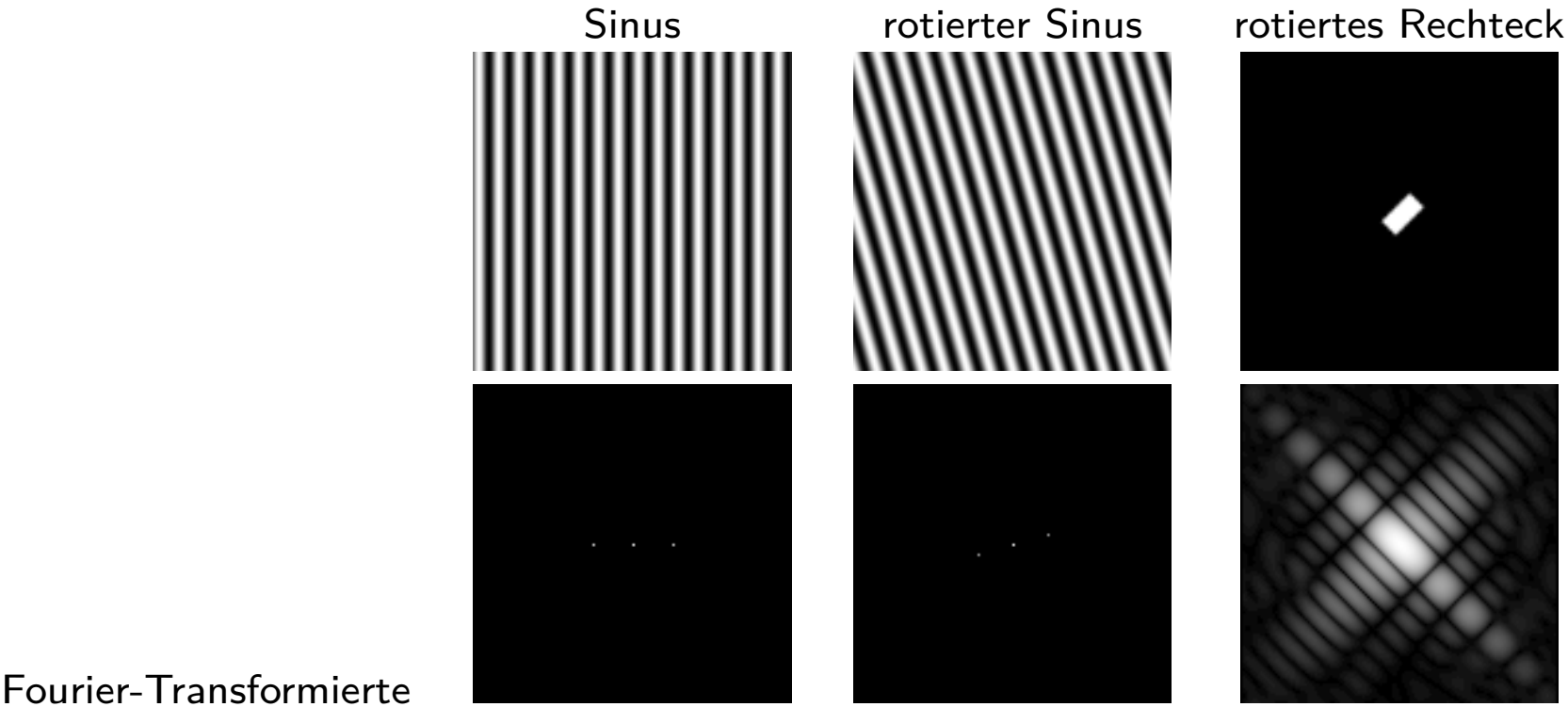
(`http://www.imagemagick.org/Usage/fourier/`)

Beispiele Fouriertransformation



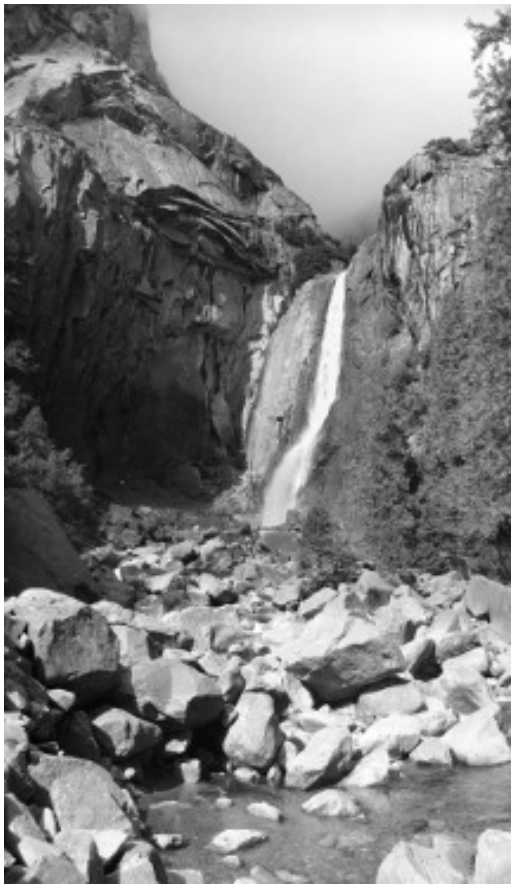
(http://www.imagemagick.org/Usage/fourier/)

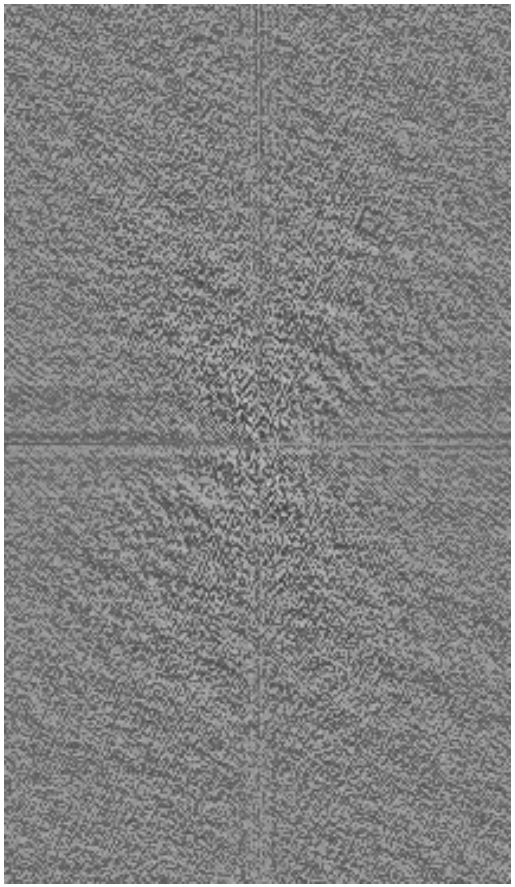
Beispiele Fouriertransformation



(`http://www.imagemagick.org/Usage/fourier/`)

Beispiele Fouriertransformation

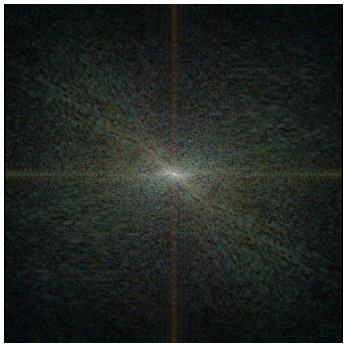




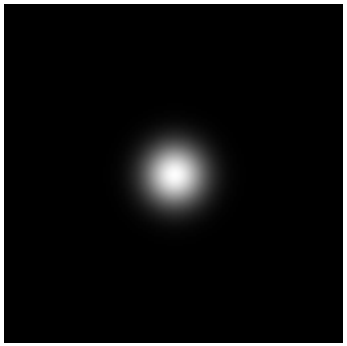
Beispiele Fouriertransformation

Original

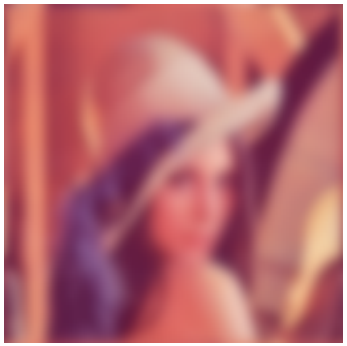
Spektrum



Tiefpass



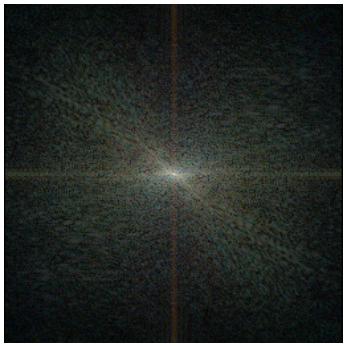
inv DFT: Gaussian blur



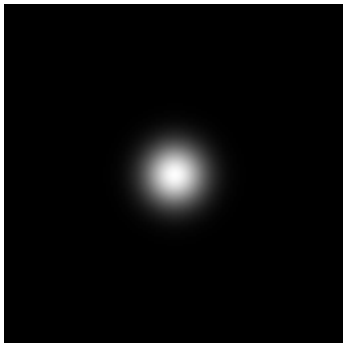
Beispiele Fouriertransformation

Original

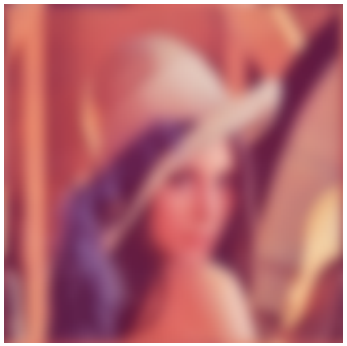
Spektrum



Tiefpass

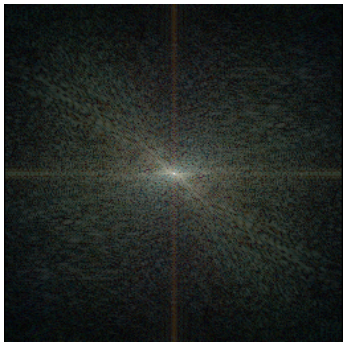


inv DFT: Gaussian blur



Original

Spektrum

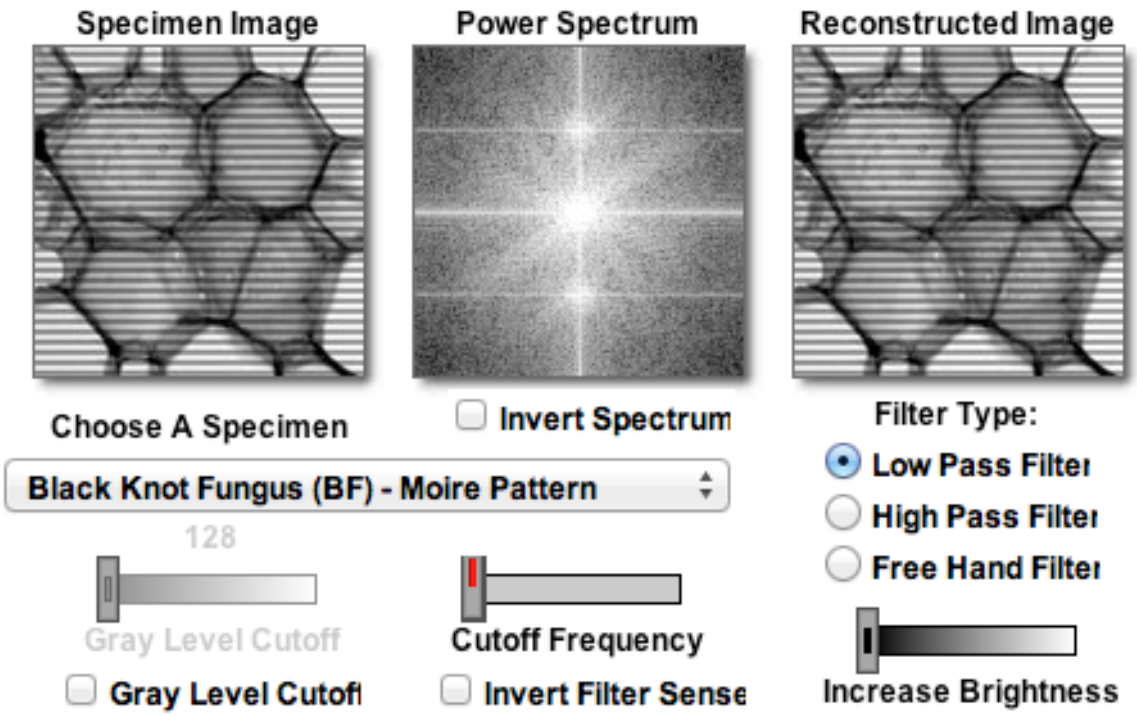


Hochpass

Kantenhervorhebung

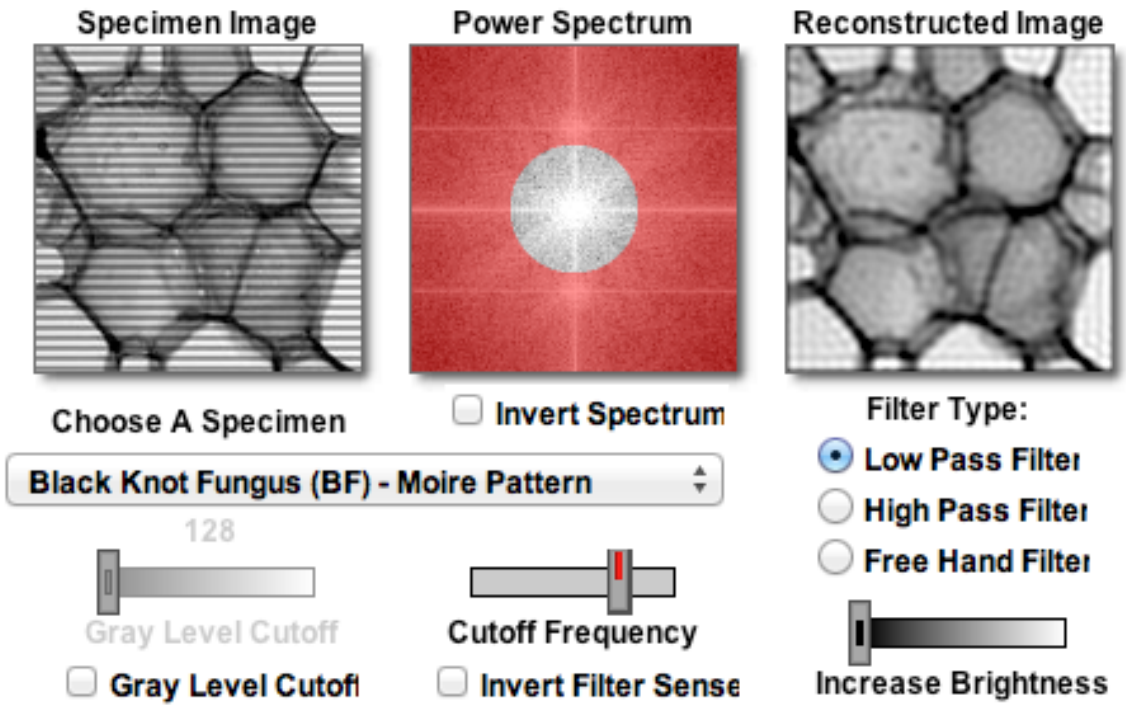
Beispiel: Filterung im Fourierbereich[a]

[a] http://micro.magnet.fsu.edu/primer/java/digitalimaging/processing/fouriertransform/index.html



Beispiel: Filterung im Fourierbereich[a]

[a] http://micro.magnet.fsu.edu/primer/java/digitalimaging/processing/fouriertransform/index.html



Fouriertransformation: Eigenschaften

Die Fouriertransformation hat einige wichtige Eigenschaften, die es erleichtern, sich schnell und oft mittels einfacher Grafiken ein Bild von komplexen Zusammenhängen zu machen.

Fouriertransformation: Eigenschaften

Zunächst irritiert bei der Fouriertransformation von reellwertigen Bildern der doppelte Speicherbedarf, da die Fouriertransformierte komplexwertig ist. Hier hilft folgende Eigenschaft weiter:

Zunächst irritiert bei der Fouriertransformation von reellwertigen Bildern der doppelte Speicherbedarf, da die Fouriertransformierte komplexwertig ist. Hier hilft folgende Eigenschaft weiter:

Die DFT einer reellen, diskreten Funktion ist **hermitsch**, d.h.

$$g_n^* = g_{N-n}$$

Zunächst irritiert bei der Fouriertransformation von reellwertigen Bildern der doppelte Speicherbedarf, da die Fouriertransformierte komplexwertig ist. Hier hilft folgende Eigenschaft weiter:

Die DFT einer reellen, diskreten Funktion ist **hermitsch**, d.h.

$$g_n^* = g_{N-n}$$

Dementsprechend reduziert sich für reellwertige Bilder der benötigte Speicheraufwand und man kann die Berechnung so umstellen, dass sie auf den reduzierten Daten direkt arbeitet.

Zunächst irritiert bei der Fouriertransformation von reellwertigen Bildern der doppelte Speicherbedarf, da die Fouriertransformierte komplexwertig ist. Hier hilft folgende Eigenschaft weiter:

Die DFT einer reellen, diskreten Funktion ist **hermitsch**, d.h.

$$g_n^* = g_{N-n}$$

Dementsprechend reduziert sich für reellwertige Bilder der benötigte Speicheraufwand und man kann die Berechnung so umstellen, dass sie auf den reduzierten Daten direkt arbeitet.

Da die Kerne der mehrdimensionalen Fouriertransformation **separabel** sind (d.h., dass die Kerne für höherdimensionale Funktionen dem Tensorprodukt der niederdimensionalen Kerne entstammen) gilt dies auch für die Transformation selbst.

Skalierung

Verändert man die räumliche Ausdehnung eines Signals, d.h., skaliert man den Raum um den Faktor $a \neq 0$, so ändert sich auch die Skalierung im Fourierbereich und man erhält

$$g(a \cdot x) \circ\!\!-\!\!- \frac{1}{|a|} \cdot \hat{g}\left(\frac{k}{a}\right)$$

Skalierung
Verändert man die räumliche Ausdehnung eines Signals, d.h., skaliert man den Raum um den Faktor $a \neq 0$, so ändert sich auch die Skalierung im Fourierbereich und man erhält

$$g(a \cdot x) \circ\!\!-\!\!- \frac{1}{|a|} \cdot \hat{g}\left(\frac{k}{a}\right)$$

Phasenverschiebung
Hat g die DFT $\hat{g}$, so hat das verschobene Bild g' mit $g'(x) = g(x - x_0)$ die DFT

$$\mathrm{e}^{-i \cdot 2\pi \langle x_0, k \rangle} \hat{g}(k)$$

Skalierung
Verändert man die räumliche Ausdehnung eines Signals, d.h., skaliert man den Raum um den Faktor $a \neq 0$, so ändert sich auch die Skalierung im Fourierbereich und man erhält

$$g(a \cdot x) \circ\!\!-\!\!- \frac{1}{|a|} \cdot \hat{g}\left(\frac{k}{a}\right)$$

Phasenverschiebung
Hat g die DFT $\hat{g}$, so hat das verschobene Bild g' mit $g'(x) = g(x - x_0)$ die DFT

$$\mathrm{e}^{-i \cdot 2\pi \langle x_0, k \rangle} \hat{g}(k)$$

Hat man also einmal die Fouriertransformierte berechnet, ergibt sich eine verschobene Version durch die (komplexe) Skalierung der Koeffizienten. Dies entspricht einer Phasenverschiebung jedes Basisbildes um den Verschiebungsvektor.

Faltung
Seien $f(x)$ und $g(x)$ zwei komplex-wertige Funktionen.
Die **Faltung** von f, g ist definiert als

$$(f * g)(x) := \int_{-\infty}^{\infty} f(x-t) \cdot g(t) dt$$

Faltung
Seien $f(x)$ und $g(x)$ zwei komplex-wertige Funktionen.
Die **Faltung** von f, g ist definiert als

$$(f * g)(x) := \int_{-\infty}^{\infty} f(x-t) \cdot g(t) dt$$

Es gilt:

$$(f * g)(x) \circ\!\!-\!\!- \hat{f}(x) \cdot \hat{g}(x)$$

Faltung
Seien $f(x)$ und $g(x)$ zwei komplex-wertige Funktionen.
Die **Faltung** von f, g ist definiert als

$$(f * g)(x) := \int_{-\infty}^{\infty} f(x-t) \cdot g(t) dt$$

Es gilt:

$$(f * g)(x) \circ\!\!-\!\!- \hat{f}(x) \cdot \hat{g}(x)$$

sowie:

$$(f \cdot g)(x) \circ\!\!-\!\!- \hat{f}(x) * \hat{g}(x)$$

Regeln der DFT

	$f(n)$	$\hat{f}(k)$
Linearität	$\alpha f_1(n) + \beta f_2(n)$	$\alpha \hat{f}_1(k) + \beta \hat{f}_2(k)$
Skalierung	$f(a \cdot n)$	$\frac{1}{\lvert a \rvert} \cdot \hat{f}(\frac{k}{a}), \quad a \neq 0$
(Zeit-)verschiebung	$f(n - n_0)$	$\hat{f}(k) \cdot e^{-i \cdot 2\pi \frac{n_0 \cdot k}{N}}$
(Frequenz-)verschiebung	$f(n) \cdot e^{i \cdot 2\pi \frac{k_0 \cdot k}{N}}$	$\hat{f}(k - k_0)$
Multiplikation	$f(n) \cdot g(n)$	$\hat{f}(n) * \hat{g}(n)$
Faltung	$f(n) * g(n)$	$\hat{f}(k) \cdot \hat{g}(k)$

Die schnelle Fouriertransformation

Die Berechnung der Fouriertransformation in der bisher vorgestellten Version ist recht rechenintensiv.
Die **schnelle Fouriertransformation** (**fast fourier transformation**, FFT) nutzt geeignete Umformungen, um möglichst viele zuvor berechnete Zwischenergebnisse wiederverwerten zu können.

Teil VI

Kantendetektion

Bildverarbeitungsprozess

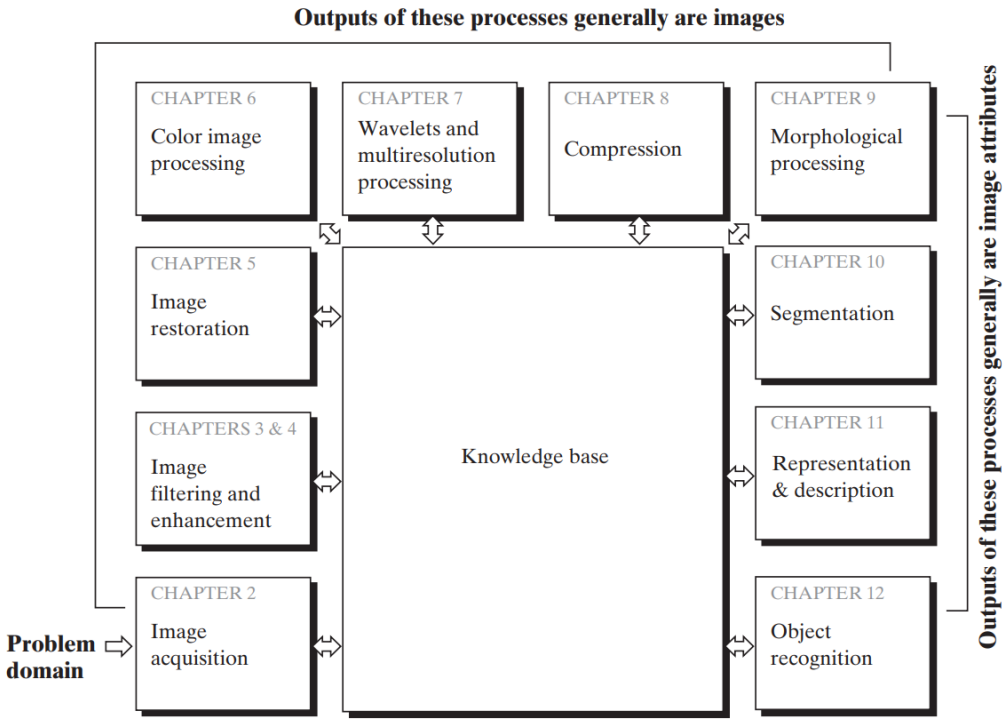


Abbildung: Die Bildverarbeitung umfasst eine reihe von Algorithmen, die entweder wiederum Bilder oder Bildattribute ausgeben. [Digital Image Processing, 3rd Edition, Gonzales and Woods]

Diese Vorlesung: Lineare Filter (Kantendetektion)

Abbildung: Was bedeutet eine Kante in einem Bild?

Kantendetektion

Will man Objekte identifizieren, so ist dies oft über die Beschreibung des Randes möglich. Daher ist ein wesentlicher Schritt in der Bildverarbeitung das Auffinden von Kanten.

Kantendetektion

Will man Objekte identifizieren, so ist dies oft über die Beschreibung des Randes möglich. Daher ist ein wesentlicher Schritt in der Bildverarbeitung das Auffinden von Kanten.

- Kanten sind Diskontinuitäten im Bild
  - Eine Kante in 2D
  - Eine Randfläche in 3D
- Kantendetektion verwendet Nachbarschaftsoperationen, um Kanten zu finden
- Diskretisierung und Rauschen erschweren Definition und Suche nach Kanten

Kantendetektion

Will man Objekte identifizieren, so ist dies oft über die Beschreibung des Randes möglich. Daher ist ein wesentlicher Schritt in der Bildverarbeitung das Auffinden von Kanten.

- Kanten sind Diskontinuitäten im Bild
  - Eine Kante in 2D
  - Eine Randfläche in 3D
- Kantendetektion verwendet Nachbarschaftsoperationen, um Kanten zu finden
- Diskretisierung und Rauschen erschweren Definition und Suche nach Kanten

Grundidee: Bildung der Ableitung des Bildes zum Erkennen von Kanten (Sprüngen in der Bildfunktion)

Eigenschaften des Kantendetektors

Ein Kantendetektor sollte folgende Eigenschaften haben:

- Ein Ableitungsoperator soll feine Strukturen stärker als grobe Strukturen verstärken und idealerweise nahe an der Fouriertransformation

$$k \to (2\pi ik)^p$$

für die p-te Ableitung sein.

Eigenschaften des Kantendetektors

Ein Kantendetektor sollte folgende Eigenschaften haben:

- Ein Ableitungsoperator soll feine Strukturen stärker als grobe Strukturen verstärken und idealerweise nahe an der Fouriertransformation

$$k \to (2\pi ik)^p$$

  für die p-te Ableitung sein.
- Vor der Kantendetektion werden jedoch häufig hohe Wellenzahlen entfernt (in der Regel Rauschen).

Eigenschaften des Kantendetektors

Ein Kantendetektor sollte folgende Eigenschaften haben:

- Ein Ableitungsoperator soll feine Strukturen stärker als grobe Strukturen verstärken und idealerweise nahe an der Fouriertransformation

$$k \to (2\pi i k)^p$$

  für die p-te Ableitung sein.
- Vor der Kantendetektion werden jedoch häufig hohe Wellenzahlen entfernt (in der Regel Rauschen).
  - Die Bedingung oben wird auf eine bestimmte maximale Wellenzahl k_{max} eingeschränkt.
  - Forderung: $\hat{h}(k > k_{\text{max}}) = 0$.

Eigenschaften des Kantendetektors

Ein Kantendetektor sollte folgende Eigenschaften haben:

- Ein Ableitungsoperator soll feine Strukturen stärker als grobe Strukturen verstärken und idealerweise nahe an der Fouriertransformation

$$k \to (2\pi ik)^p$$

  für die p-te Ableitung sein.
- Vor der Kantendetektion werden jedoch häufig hohe Wellenzahlen entfernt (in der Regel Rauschen).
  - Die Bedingung oben wird auf eine bestimmte maximale Wellenzahl k_{max} eingeschränkt.
  - Forderung: $\hat{h}(k > k_{\text{max}}) = 0$.
- Der Kantendetektor soll isotrop sein, d.h., dass der Operator soll richtungsunabhängig sein.

$$\hat{h}(k) = \hat{h'}(|k|)$$

Betrachten wir den Gradienten

$$D = \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix}$$

Dann ist der Gradientenfilter richtungsinvariant

$$|D| = \sqrt{D_x \cdot D_x + D_y \cdot D_y + D_z \cdot D_z}$$

Betrachten wir den Gradienten

$$D = \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix}$$

Dann ist der Gradientenfilter richtungsinvariant

$$|D| = \sqrt{D_x \cdot D_x + D_y \cdot D_y + D_z \cdot D_z}$$

Zur Berechnung wird der folgende Ablauf verwendet:

1. Das Bild wird mit allen drei Richtungsableitungen D_x, D_y, and D_z gefiltert
2. Die Ergebniswerte werden quadriert
3. Die quadrierten Werte an der selben Position im Bild werden addiert
4. Die Wurzel aus der jeweiligen Summe wird berechnet

Aufgrund der anscheinend hohen Komplexität der Berechnung des Gradientenfilters wurde folgende Approximation vorgeschlagen

$$|D| \approx |D_x| + |D_y| + |D_z|$$

Kantenerkennung mittels Gradientenfilter

Aufgrund der anscheinend hohen Komplexität der Berechnung des Gradientenfilters wurde folgende Approximation vorgeschlagen

$$|D| \approx |D_x| + |D_y| + |D_z|$$

Diese Approximation ist jedoch auch für kleine Wellenzahlen anisotrop, da die Diagonalen um den Faktor $\sqrt{3}$ stärker gewichtet werden.

Differenzen erster Ordnung

Die einfachsten diskreten Ableitungen sind die **Differenzen erster Ordnung**

Rückwärtsdifferenz	$\frac{g(x)-g(x-\Delta x)}{\Delta x}$	$D_x^- = [1. \; -1]$
Vorwärtsdifferenz	$\frac{g(x+\Delta x)-g(x)}{\Delta x}$	$D_x^+ = [1 \; -1.]$
Zentrale Differenz	$\frac{g(x+\Delta x)-g(x-\Delta x)}{2\Delta x}$	$D_{2x} = \frac{1}{2}[1 \; 0. \; -1]$

Kantenerkennung über zweite Ableitungen

- Kanten führen zu Nulldurchgängen in der zweiten Ableitung.
- Zweifache diskrete Ableitung mit D_x^- und D_x^+

$\rightarrow$ **Laplacefilter**

$$D_x^- * D_x^+ = D_x^2$$

$$[\underline{1}\ -1] * [1\ \underline{-1}] = [1\ \underline{-2}\ 1]$$

was der diskreten Näherung

$$\frac{\partial^2 g}{\partial x^2} \approx \frac{g(x+\Delta x) - 2g(x) + g(x-\Delta x)}{\Delta x^2}$$

entspricht.

Kantenerkennung über zweite Ableitungen

In 2D erhält man $L = D_x^2 + D_y^2$

$$L = [1\ -2\ 1] + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

In 3D entsprechend

$$L = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}_z$$

mit den 3 Ebenen

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & -6 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

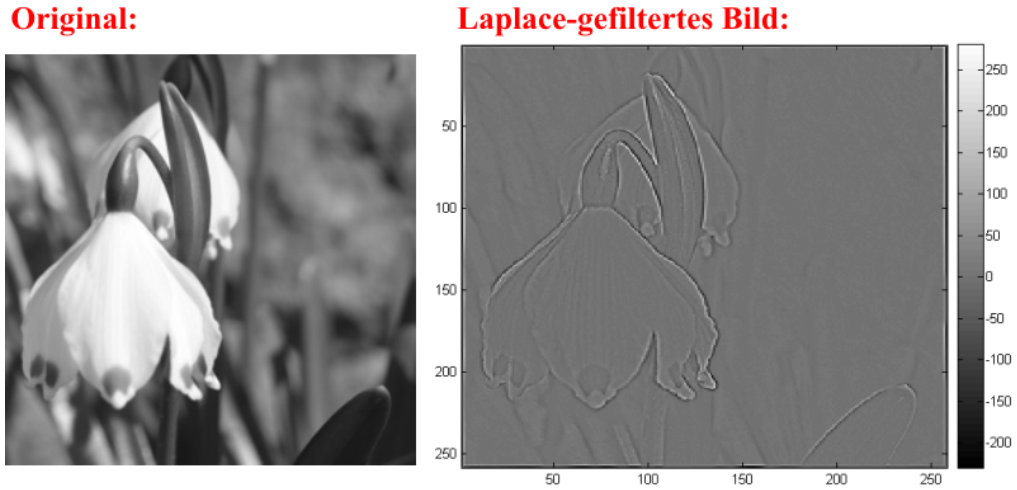


Abbildung: Beispiel der Laplace Kantendetektion

Es gibt weitere Wege der Approximation von

$$\Delta g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2},$$

so zum Beispiel über Binomialfilter. Man nutzt Filter der Form $B^p - I$ und erhält zum Beispiel

$$L' = 4(B^2 - I) = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} - 4\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Regularisierte Kantendetektoren

- Kantendetektion erwünscht
- Rauschen verhindert das Auffinden
- Glättung entfernt Kanten

Regularisierte Kantendetektoren

- Kantendetektion erwünscht
- Rauschen verhindert das Auffinden
- Glättung entfernt Kanten
- Lösung: Kombination von Glättung entlang der Kante und Kantendetektion senkrecht dazu

$$D_x * B_y = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$D_y * B_x = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Regularisierte Kantendetektoren

- Kantendetektion erwünscht
- Rauschen verhindert das Auffinden
- Glättung entfernt Kanten
- Lösung: Kombination von Glättung entlang der Kante und Kantendetektion senkrecht dazu

$$D_x * B_y = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$D_y * B_x = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

- Nachteil: Filter verschiebt Bild um $\frac{1}{2}$ Pixel
  $\rightarrow$ kein idealer Kantendetektor

Um das Verschieben des Bildes bei Filtern der Länge zwei zu vermeiden, nutzt man zentrale Differenzen und Binomialfilter der Breite drei

$$D_{2x} * B_y^2 = \frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$D_{2y} * B_x^2 = \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Diese Operation wird Sobel-Operator genannt[a].

[a] Irwin Sobel, `http://www.hpl.hp.com/personal/Irwin_Sobel/`

Ableitung der Gaußfunktion

- Da die Ableitungsoperation unter der Faltung erhalten bleibt, kann man beliebige geglättete Operatoren zur Ableitungsbestimmung benutzen.
- Man kann hier z.B. den besprochenen Filter $\begin{bmatrix} -1 & 1 \end{bmatrix}$ als Ableitung der Identität [1] auffassen.
- Leitet man Glättungsfilter ab, so erhält man eine geglättete Kantenerkennung.

Bildverarbeitungsprozess

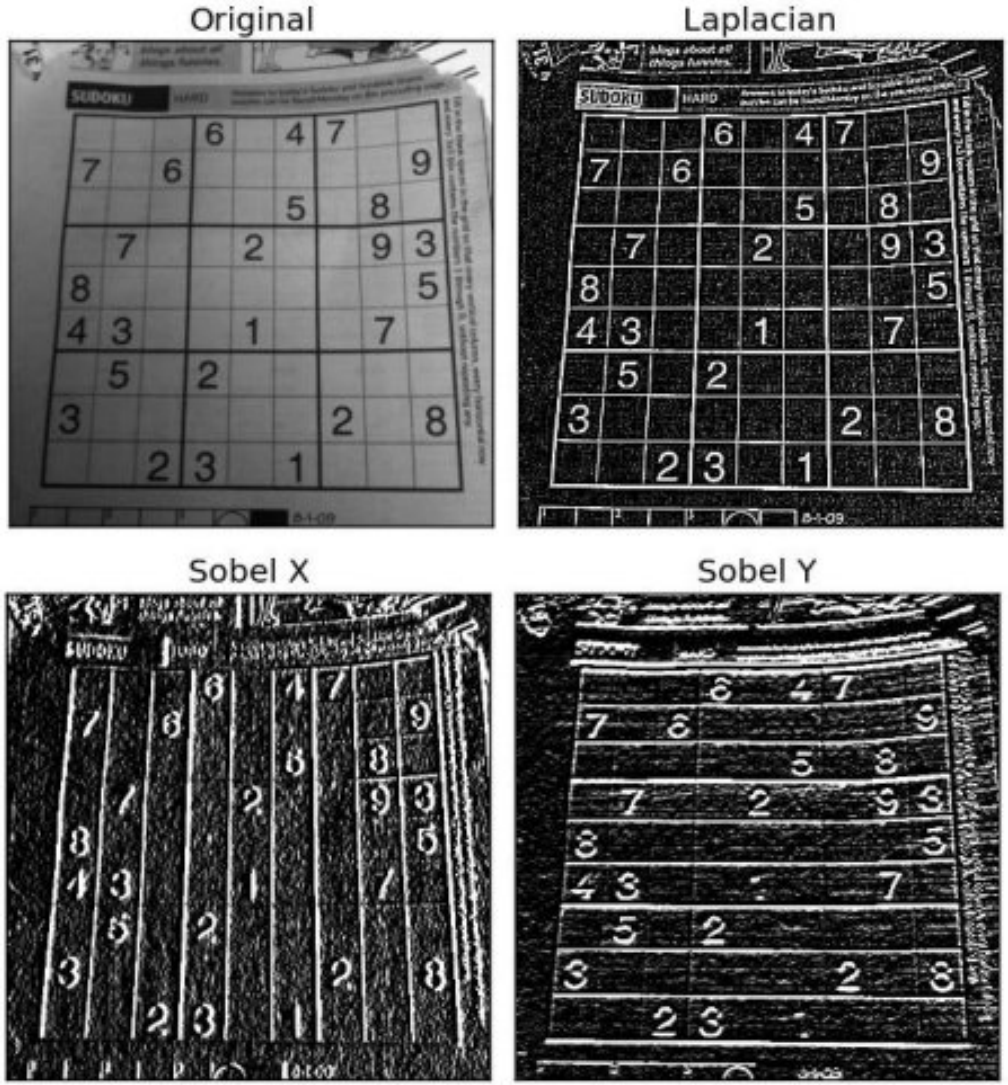


Abbildung: Beispiel des Sobel Operators im Vergeil zu Laplace

Ein Beispiel der Ableitung des Gaußkerns ist der **Canny edge detector**[2])
Als Approximation dieser Idee werden oft folgende Filter verwendet:

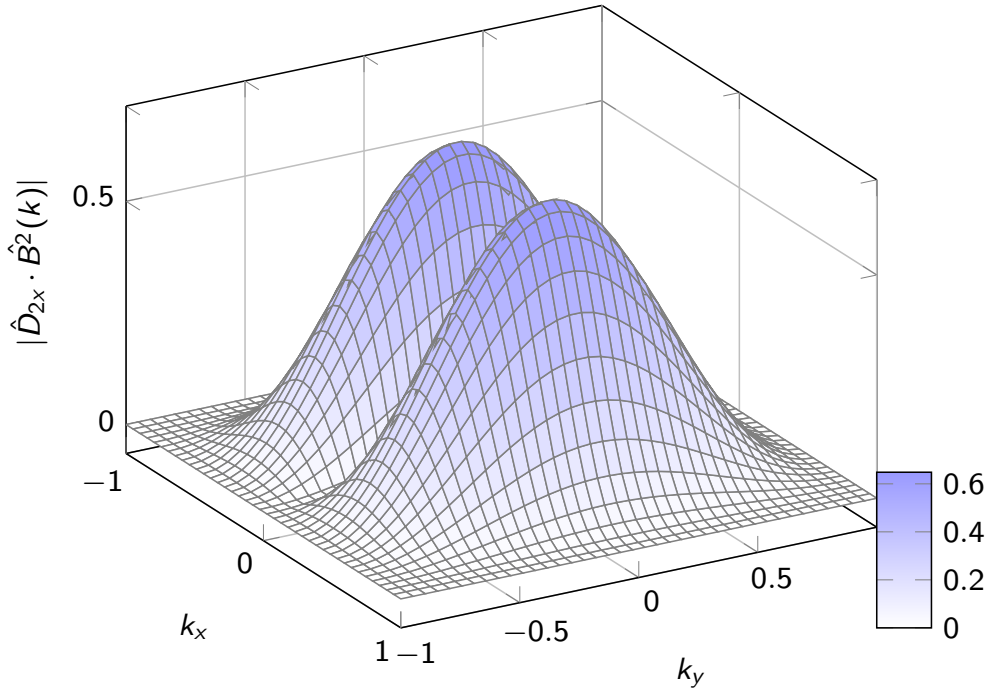
$$D_{2x} * B^2 = \frac{1}{2}[1\,0\,-1] * \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} = \frac{1}{32}\begin{bmatrix}1 & 2 & 0 & -2 & -1\\ 2 & 4 & 0 & -4 & -2\\ 1 & 2 & 0 & -2 & -1\end{bmatrix}$$

$$D_{2y} * B^2 = \frac{1}{2}\begin{bmatrix}1\\ 0\\ -1\end{bmatrix} * \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} = \frac{1}{32}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 0 & 0 & 0\\ -2 & -4 & -2\\ -1 & -2 & -1\end{bmatrix}$$

Canny Edge Detector wird häufig und gerne verwendet. Verwendung sollte man eventuell überdenken

[2] J. F. Canny. A computational apprach to edge detection. PAMI, 8:679–698, 1986

Trotz der in den 90er Jahren großen Beliebtheit des **Canny Edge Detectors** zeigt eine Fehlerberechnung die gleichen Anisotropiewerte wie D_{2x}. Größere Masken bringen keine wesentliche Verbesserung.



Canny-Edge-Detector

Abbildung: Canny Edge Detection

Da die Ableitung der Gaußfunktion keine optimalen Ergebnisse liefert, wurde der Filter von Jähne et al.[3] optimiert (**Optimierter Sobel-Operator**). Die folgenden Filter haben minimale Winkelfehler:

$$\frac{1}{4} D_{2x}(3B_y^2 + I) = \frac{1}{32} \begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix}$$

$$\frac{1}{4} D_{2y}(3B_x^2 + I) = \frac{1}{32} \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix}$$

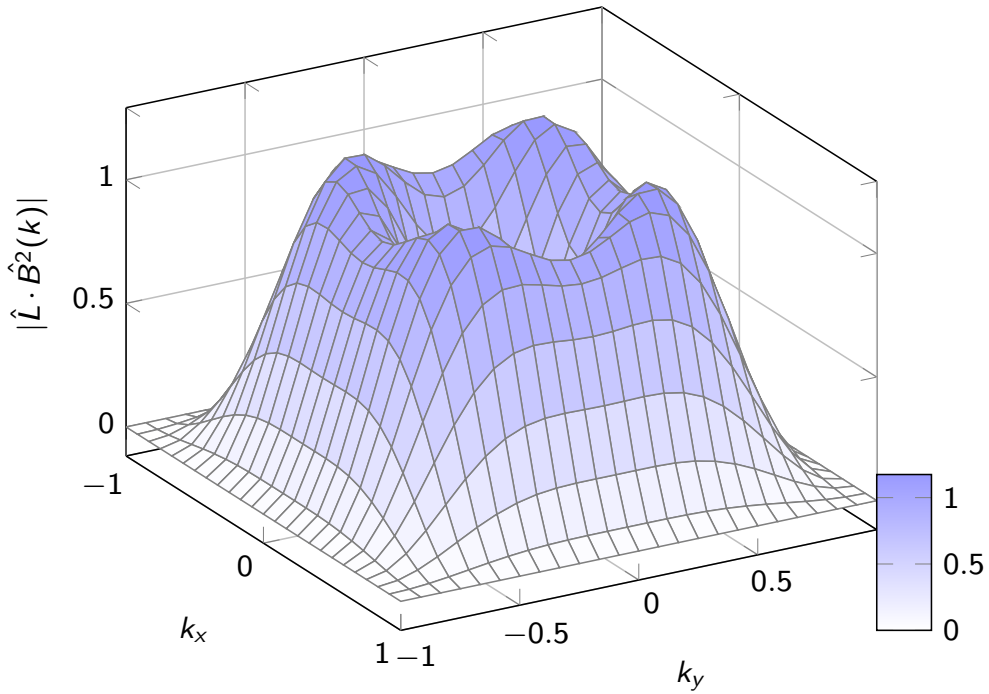
[3] B. Jähne, H. Scharr, S. Körgel. Principles of filter design, Computer Vision and Applications Vol 2, Signal Processing and Pattern Recognition, Chapter 6, pp. 125-151. Academic Press, San Diego, 1999

Laplacian of Gaussian/Difference of Gaussian

- Laplacefilter erhöhen das Rauschen
  $\rightarrow$ Glättung erwünscht
- Glättung des Bildes mit Binomialfiltermaske, z.B.
  $[1\ 2\ 1]$
- Anschließend diskreter Laplacefilter $[1\ -2\ 1]$
  oder Differenzenfilter

Der **Laplace of Gaussian** oder **Marr-Hildreth-Operator** genannte Filter ist dann

$$
\begin{aligned}
L * B^2 &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \\
&= \frac{1}{16} \begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 0 & -2 & 0 & 1 \\ 2 & -2 & -8 & -2 & 2 \\ 1 & 0 & -2 & 0 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}
\end{aligned}
$$


$|\hat{L} \cdot \hat{B}^2(k)|$
1
0.5
0
−1
0
1
k_x
−1
−0.5
0
0.5
k_y
1
0.5
0

Eine leichte Verbesserung ergibt sich über Differenzen $G - I$ des Gaußfilters (sog. DoG-Filter)

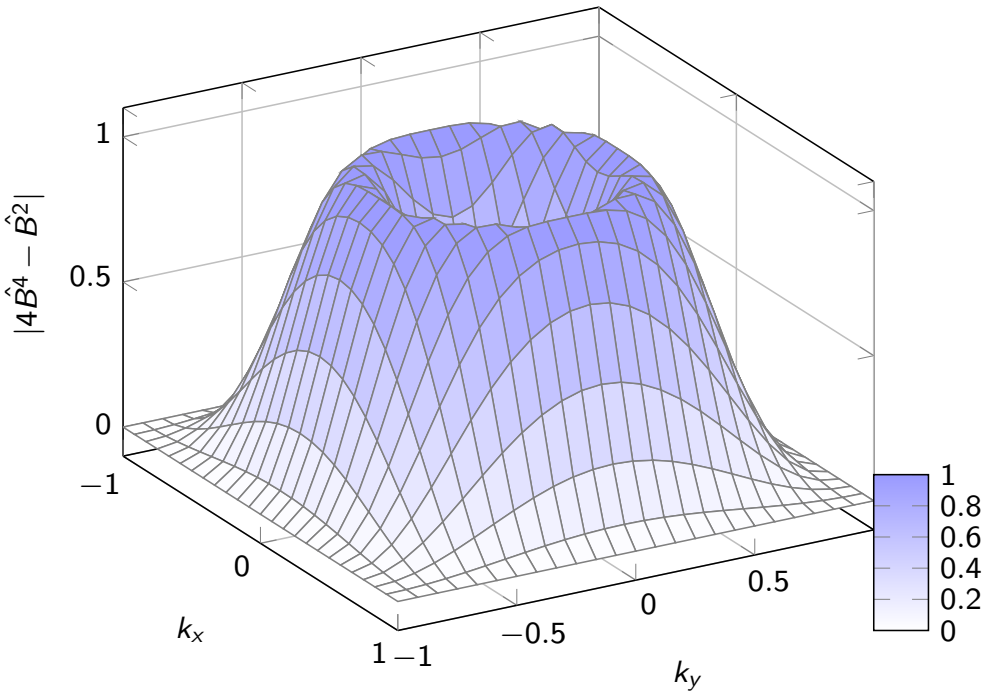
$$4(B^q - I)B^p = 4(B^{p+q'} - B^p)$$

wodurch sich für $q = 2$ und $p = 2$ der Filter

$$\begin{aligned} 4(B^2 - I) * B^2 &= 4\left(\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} - \begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\right) * \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} \\ &= \begin{bmatrix}1 & 4 & 6 & 4 & 1\\ 4 & 0 & -8 & 0 & 4\\ 6 & -8 & -28 & -8 & 6\\ 4 & 0 & -8 & 0 & 4\\ 1 & 4 & 6 & 4 & 1\end{bmatrix} \end{aligned}$$

ergibt.

DoG-Filter



Skalen

- Bisherige Operationen dienen nur zur Erkennung kleiner Objekte.
- Will man große Objekte erkennen können, so ist eine Reduktion der Bildgröße zur effizienten Bearbeitung notwendig.

Gauß- und Laplace-Pyramide

Ziel: Abtastung der Bilder auf unterschiedlichen Auflösungsstufen.

Gauß- und Laplace-Pyramide

Ziel: Abtastung der Bilder auf unterschiedlichen Auflösungsstufen.

- Bei diskreten Bildern erhält man das Problem der Artefakte, wenn die Abtastung geändert wird.

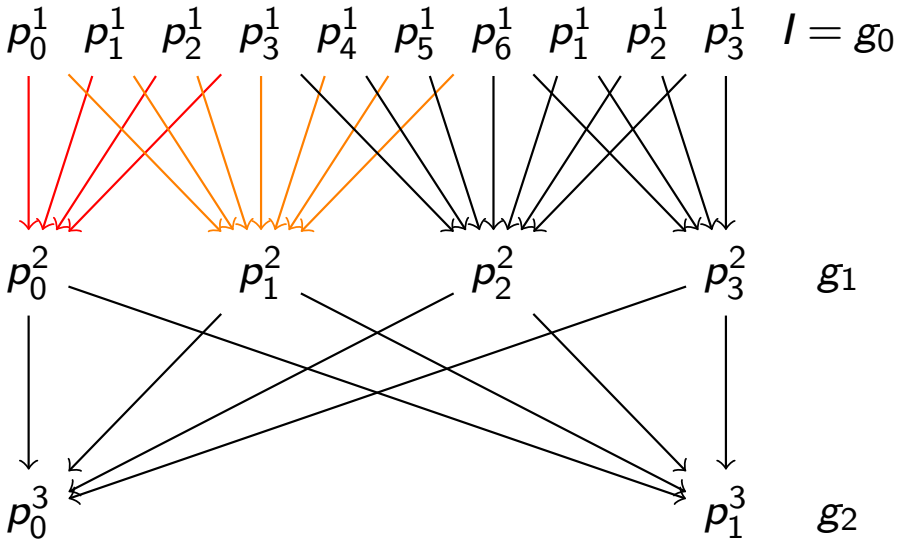
Gauß- und Laplace-Pyramide

Ziel: Abtastung der Bilder auf unterschiedlichen Auflösungsstufen.

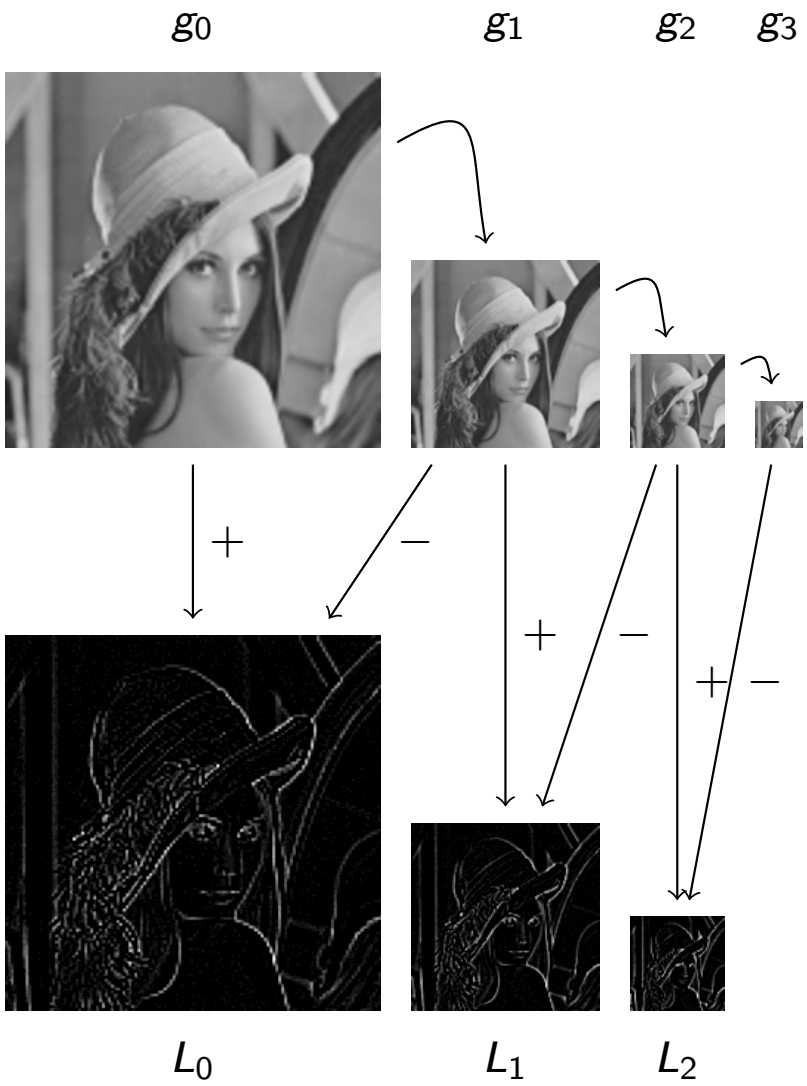
- Bei diskreten Bildern erhält man das Problem der Artefakte, wenn die Abtastung geändert wird.
- Da die Artefakte durch hohe Frequenzen auftreten, muss bei einer Abtastschrittweite r sichergestellt werden, dass keine Frequenzen über $k \geq \frac{1}{r}$ im Bild vorhanden sind. [a]

[a] The Laplacian Pyramid as a Compact Image Code, PJ. Burt and E H Adelson, https://forums.cs.tau.ac.il/~hezy/Vision%20Seminar/pyramid83.pdf

Man filtert hierfür das Bild mit einem Tiefpass und tastet das Bild ab.

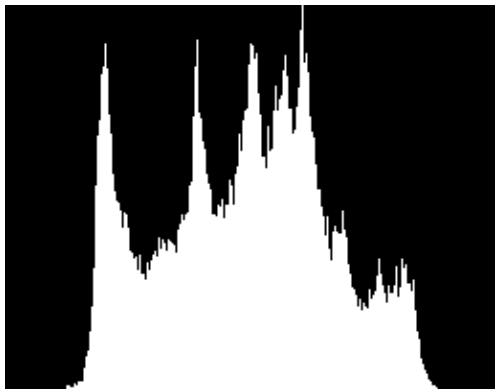


Gauß- und Laplace-Pyramide



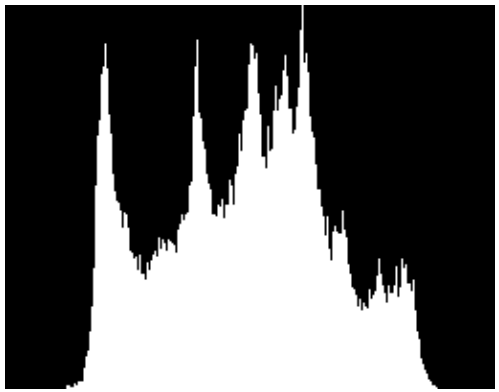
- Eigenschaften der Bilder auf unterschiedlichen Auflösungsstufen
- Kleinere Filter können für grobe Strukturen verwendet werden
- Gauß-Pyramide für Glättung auf unterschiedlichen Skalen
- Laplace-Pyramide für Kantenerkennung auf unterschiedlichen Skalen

- Das Originalbild lässt sich aufgrund des Histogramms schlecht komprimieren

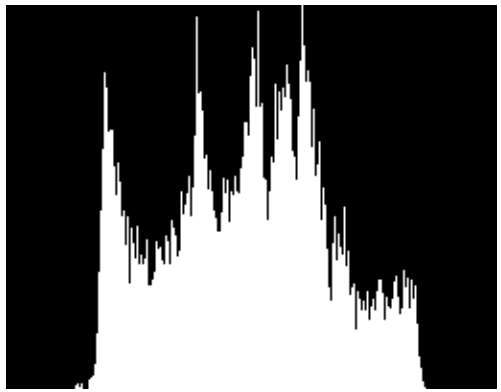


Original

- Das Originalbild lässt sich aufgrund des Histogramms schlecht komprimieren
- Die geglätteten Bilder verhalten sich ähnlich

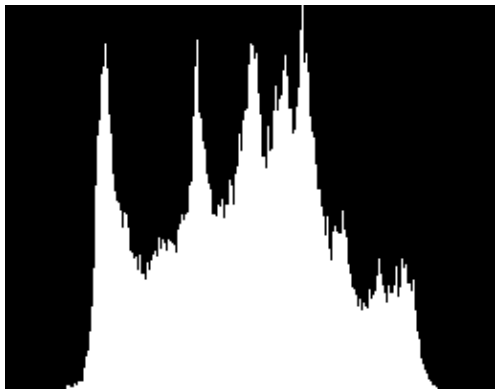


Original

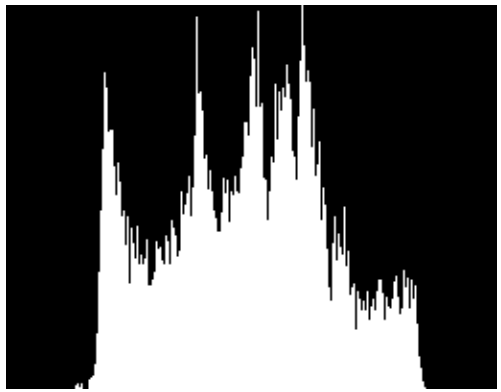


g_2

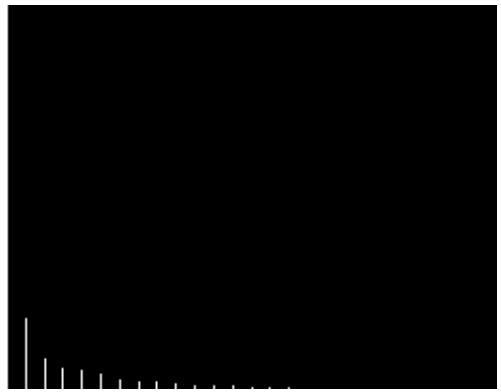
- Das Originalbild lässt sich aufgrund des Histogramms schlecht komprimieren
- Die geglätteten Bilder verhalten sich ähnlich
- Die Laplacebilder haben ein schmaleres Histogramm
  $\rightarrow$ weniger Werte
  $\rightarrow$ bessere Kompression



Original



g_2



L_2

Kontinuierliche Skalenräume

- Gauß- und Laplace-Pyramiden bieten Skalenansatz auf mehreren Gittern (Multigrid).
- Sie sind jedoch durch die feste Abtastung beschränkt.
- Dagegen ist manchmal eine kontinuierliche Darstellung erwünscht.

Kontinuierliche Skalenräume

- Gauß- und Laplace-Pyramiden bieten Skalenansatz auf mehreren Gittern (Multigrid).
- Sie sind jedoch durch die feste Abtastung beschränkt.
- Dagegen ist manchmal eine kontinuierliche Darstellung erwünscht.
- **Motivation:** Physikalische Prozesse „glätten“ Kanten.

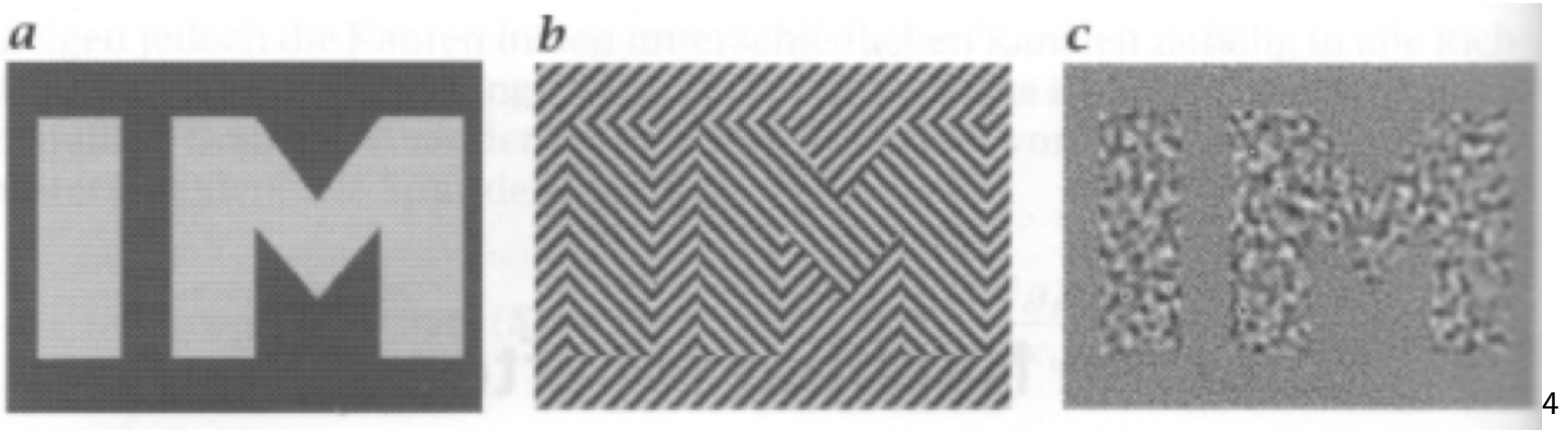
Kontinuierliche Skalenräume

- Gauß- und Laplace-Pyramiden bieten Skalenansatz auf mehreren Gittern (Multigrid).
- Sie sind jedoch durch die feste Abtastung beschränkt.
- Dagegen ist manchmal eine kontinuierliche Darstellung erwünscht.
- **Motivation:** Physikalische Prozesse „glätten“ Kanten.
- Diffusion (durch stochastische Teilchenbewegung) lässt harte Kanten verschwinden.

Kontinuierliche Skalenräume

- Gauß- und Laplace-Pyramiden bieten Skalenansatz auf mehreren Gittern (Multigrid).
- Sie sind jedoch durch die feste Abtastung beschränkt.
- Dagegen ist manchmal eine kontinuierliche Darstellung erwünscht.
- **Motivation:** Physikalische Prozesse „glätten“ Kanten.
- Diffusion (durch stochastische Teilchenbewegung) lässt harte Kanten verschwinden.
  $\rightarrow$ Simulation der Diffusion als Glättungsoperation.

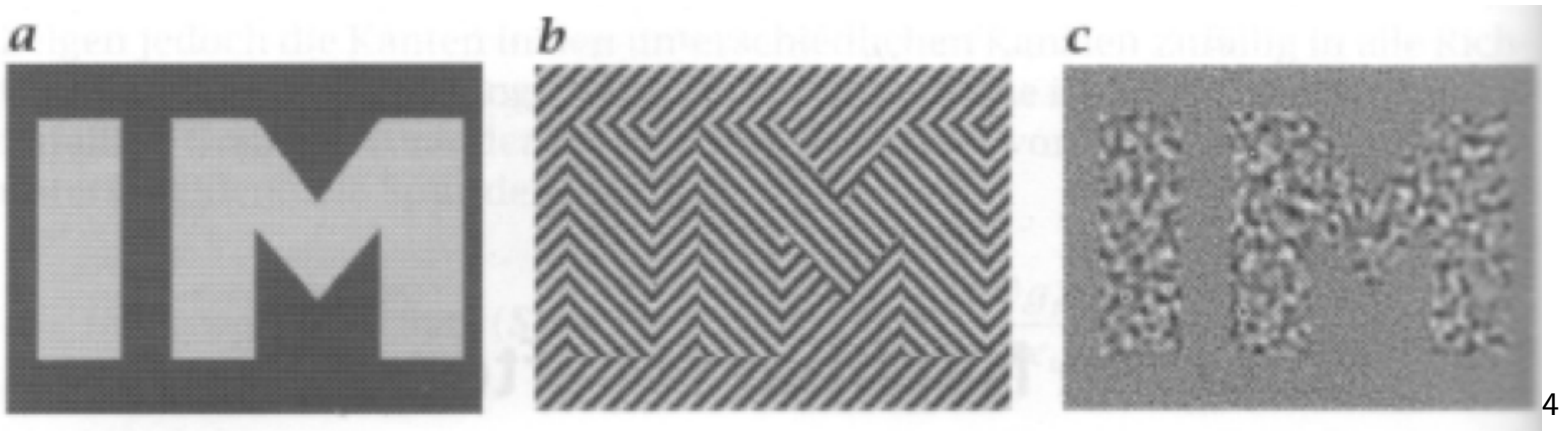
Kantendetektion liefert in einfachen Fällen gute Ergebnisse, oft sind jedoch andere Methoden nötig:



[4]

[4] Bildquelle: Jähne

Kantendetektion liefert in einfachen Fällen gute Ergebnisse, oft sind jedoch andere Methoden nötig:



[4]

Wir suchen ein Maß für die „Gleichförmigkeit“ der Nachbarschaft eines Pixels.

[4] Bildquelle: Jähne

Wir suchen sogenannte **einfache Nachbarschaften**, bei denen sich die Werte nur in eine Richtung ändern.

$$p(x) = \bar{p}(x^T n) \quad x \in U \subset \mathbb{R}^d$$

wobei $\nabla p \| n$ ist.

Wir suchen sogenannte **einfache Nachbarschaften**, bei denen sich die Werte nur in eine Richtung ändern.

$$p(x) = \bar{p}(x^T n) \quad x \in U \subset \mathbb{R}^d$$

wobei $\nabla p \| n$ ist.

- Sucht man nach einem gemittelten Vektor $\bar{n}$ in einer Umgebung um das Pixel (Voxel) x, so würde man keinen Unterschied zwischen einem unkorrelierten Rauschen und einem konstanten Grauwert finden, da der mittlere Vektor $\bar{n} = 0$ wäre.

Wir suchen sogenannte **einfache Nachbarschaften**, bei denen sich die Werte nur in eine Richtung ändern.

$$p(x) = \bar{p}(x^T n) \quad x \in U \subset \mathbb{R}^d$$

wobei $\nabla p \| n$ ist.

- Sucht man nach einem gemittelten Vektor $\bar{n}$ in einer Umgebung um das Pixel (Voxel) x, so würde man keinen Unterschied zwischen einem unkorrelierten Rauschen und einem konstanten Grauwert finden, da der mittlere Vektor $\bar{n} = 0$ wäre.
- Man nimmt daher eine tensorielle Größe zur Beschreibung der Werte um das Pixel x.

Wir betrachten in einer Umgebung um das Pixel x das Quadrat des Skalarproduktes aus Gradient und unbekannter Einheitsnormalen n für eine Fensterfunktion w um 0:

$$\int w(x - x')\langle \nabla g(x'), n\rangle^2 dx' \to \max_n$$

Wir betrachten in einer Umgebung um das Pixel x das Quadrat des Skalarproduktes aus Gradient und unbekannter Einheitsnormalen n für eine Fensterfunktion w um 0:

$$\int w(x-x')\langle \nabla g(x'), n\rangle^2 dx' \to \max_n$$

Dies kann auch als

$$J = \int w(x-x')(\nabla g(x')\nabla g(x')^T)dx'$$

mit

$$n^T Jn \to \max$$

beschrieben werden.

Wir betrachten in einer Umgebung um das Pixel x das Quadrat des Skalarproduktes aus Gradient und unbekannter Einheitsnormalen n für eine Fensterfunktion w um 0:

$$\int w(x-x')\langle \nabla g(x'), n\rangle^2 dx' \to \max_n$$

Dies kann auch als

$$J = \int w(x-x')(\nabla g(x')\nabla g(x')^T)dx'$$

mit

$$n^T Jn \to \max$$

beschrieben werden.
In Komponenten gilt:

$$J_{pq} = \int_{-\infty}^{\infty} w(x-x') \left(\frac{\partial g(x')}{\partial x'_p} \frac{\partial g(x')}{\partial x'_q} \right) dx'$$

J ist ein symmetrischer Tensor, der **Strukturtensor** genannt wird. Man kann J durch Drehung in Diagonalform mit drei zueinander senkrecht stehenden Eigenvektoren n'_1, n'_2 und n'_3 bringen:

$$J = \begin{pmatrix} n'_1 & n'_2 & n'_3 \end{pmatrix} \begin{pmatrix} J'_{11} & & \\ & J'_{22} & \\ & & J'_{33} \end{pmatrix} \begin{pmatrix} n'_1 \\ n'_2 \\ n'_3 \end{pmatrix}$$

J ist ein symmetrischer Tensor, der **Strukturtensor** genannt wird. Man kann J durch Drehung in Diagonalform mit drei zueinander senkrecht stehenden Eigenvektoren n'_1, n'_2 und n'_3 bringen:

$$J = \begin{pmatrix} n'_1 & n'_2 & n'_3 \end{pmatrix} \begin{pmatrix} J'_{11} & & \\ & J'_{22} & \\ & & J'_{33} \end{pmatrix} \begin{pmatrix} n'_1 \\ n'_2 \\ n'_3 \end{pmatrix}$$

Für $J'_{11} > J'_{22} > J'_{33}$ erhält man mit n'_1 die gesuchte Normale als Eigenvektor von J zum größten Eigenwert J'_{11}.

Die Eigenvektoren liefern lokale Strukturinformation (hier für 3D):

Bedingung	**Rang**	**Lokales Verhalten**
$J'_{11} = J'_{22} = J'_{33} = 0$	0	Grauwerte ändern sich nicht, konstante Umgebung
$J'_{11} > 0,\ J'_{22} = J'_{33} = 0$	1	Grauwerte ändern sich nur in eine Richtung $\rightarrow$ Objektgrenze
$J'_{11} > 0,\ J'_{22} > 0,\ J'_{33} = 0$	2	Grauwerte ändern sich in zwei Richtungen, konstant in dritter Richtung. Die dritte Eigenrichtung gibt die Richtung konstanter Grauwerte an.
$J'_{11} > 0,\ J'_{22} > 0,\ J'_{33} > 0$	3	Änderung der Werte in alle drei Richtungen

Als **Kohärenz** der lokalen Struktur definiert man in 2D den Ausdruck

$$c = \frac{J'_{11} - J'_{22}}{J'_{11} + J'_{22}} \in [0, 1]$$

Als **Kohärenz** der lokalen Struktur definiert man in 2D den Ausdruck

$$c = \frac{J'_{11} - J'_{22}}{J'_{11} + J'_{22}} \in [0,1]$$

- $c = 0$: isotrope Verhältnisse
- $c = 1$: einfache Nachbarschaft

Zur Implementierung sind Diskretisierungen von Ableitung und Integral nötig. Man nutzt hier die diskreten Ableitungsoperationen und bestimmt

$$J_{pq} = B(D_p \cdot D_q) \quad p, q \in \{x, y, z\}$$

mit pixelweiser Multiplikation „$\cdot$“.

Zur Implementierung sind Diskretisierungen von Ableitung und Integral nötig. Man nutzt hier die diskreten Ableitungsoperationen und bestimmt

$$J_{pq} = B(D_p \cdot D_q) \quad p, q \in \{x, y, z\}$$

mit pixelweiser Multiplikation „$\cdot$“.
Aufgrund der Symmetrie benötigt man in 2D drei solche Berechnungen, in 3D entsprechend sechs.[a]

[a] http://www.informatik.uni-ulm.de/ni/staff/PBayerl/homepage/misc/strukturtensor.pdf

Fehler

- Die Genauigkeit der Orientierungsbestimmung (Richtung von n) ist das Hauptkriterium für den Fehler.
- Dabei liefert selbst der normale Sobel-Filter ungünstige Resultate.
- Es gibt jedoch in der Literatur weitere Optimierungen, die hier helfen können.

Teil VII

Pixelverarbeitung

Bildverarbeitungsprozess

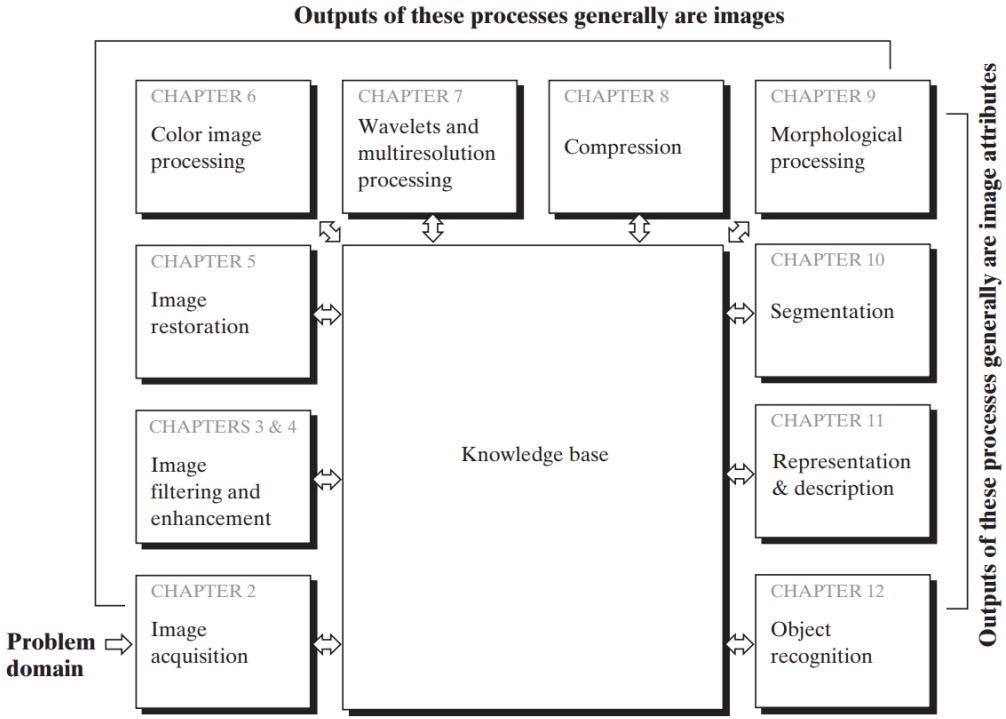


Abbildung: Die Bildverarbeitung umfasst eine reihe von Algorithmen, die entweder wiederum Bilder oder Bildattribute ausgeben. [Digital Image Processing, 3rd Edition, Gonzales and Woods]

Diese Vorlesung: Image Restoration

Pixelverarbeitung: Operationen

Bevor Merkmale extrahiert werden können, müssen Bilder in der Regel noch vorverarbeitet werden, zum Beispiel

- um Kontraste zu verstärken
- um Rauschen zu entfernen
- um Verzerrungen zu korrigieren

Pixelverarbeitung: Operationen

Bevor Merkmale extrahiert werden können, müssen Bilder in der Regel noch vorverarbeitet werden, zum Beispiel

- um Kontraste zu verstärken
- um Rauschen zu entfernen
- um Verzerrungen zu korrigieren

Die hierbei eingesetzten Operationen werden in zwei Klassen eingeteilt:

- **Punktoperationen** modifizieren die Bildpunkte nur in Abhängigkeit des Wertes am jeweiligen Punkt

$$g'_{m,n,o} = P_{m,n,o}(g_{m,n,o})$$

Pixelverarbeitung: Operationen

Bevor Merkmale extrahiert werden können, müssen Bilder in der Regel noch vorverarbeitet werden, zum Beispiel

- um Kontraste zu verstärken
- um Rauschen zu entfernen
- um Verzerrungen zu korrigieren

Die hierbei eingesetzten Operationen werden in zwei Klassen eingeteilt:

- **Punktoperationen** modifizieren die Bildpunkte nur in Abhängigkeit des Wertes am jeweiligen Punkt

$$g'_{m,n,o} = P_{m,n,o}(g_{m,n,o})$$

- **Geometrische Operationen** verändern hingegen nur die Position eines Bildpunktes, nicht aber seinen Wert

$$g'_{m',n',o'} = g'_{G_{m,n,o}(m,n,o)} = g_{m,n,o}$$

Man nennt eine Operation **homogen**, wenn die Operation unabhängig von der Position ist, also gilt

$$g'_{m,n,o} = P(g_{m,n,o})$$

beziehungsweise

$$g'_{m',n',o'} = g'_{G(m,n,o)} = g_{m,n,o}$$

Konstrastverstärkung

Liegen die Intensitäten im Bild in einem kleinen Wertebereich, so lassen sich Unterschiede nur sehr schwer wahrnehmen. Es ist dann oft sinnvoll, eine möglichst homogene Verteilung der Werte im Bild anzustreben. Die einfachste Version hierfür ist die Streckung des verwendeten Wertebereichs auf den zur Verfügung stehenden Wertebereich

$$P(v) = \frac{v - v_{\text{min}}}{v_{\text{max}} - v_{\text{min}}} \cdot (v'_{\text{max}} - v'_{\text{min}}) + v'_{\text{min}}$$

- v_{min} und v_{max}: minimaler und maximaler im Bild gemessener Wert
- v'_{min} und v'_{max}: gewünschte neue Grenzen des Wertebereiches

Beispiel Kontrastverstärkung

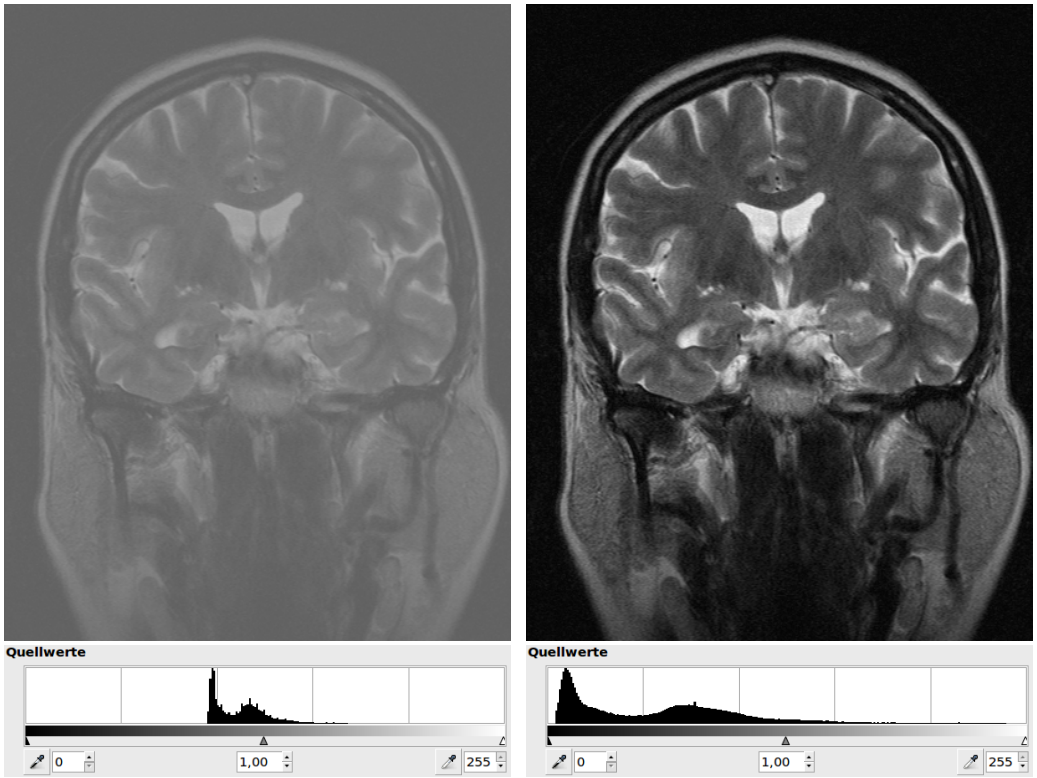
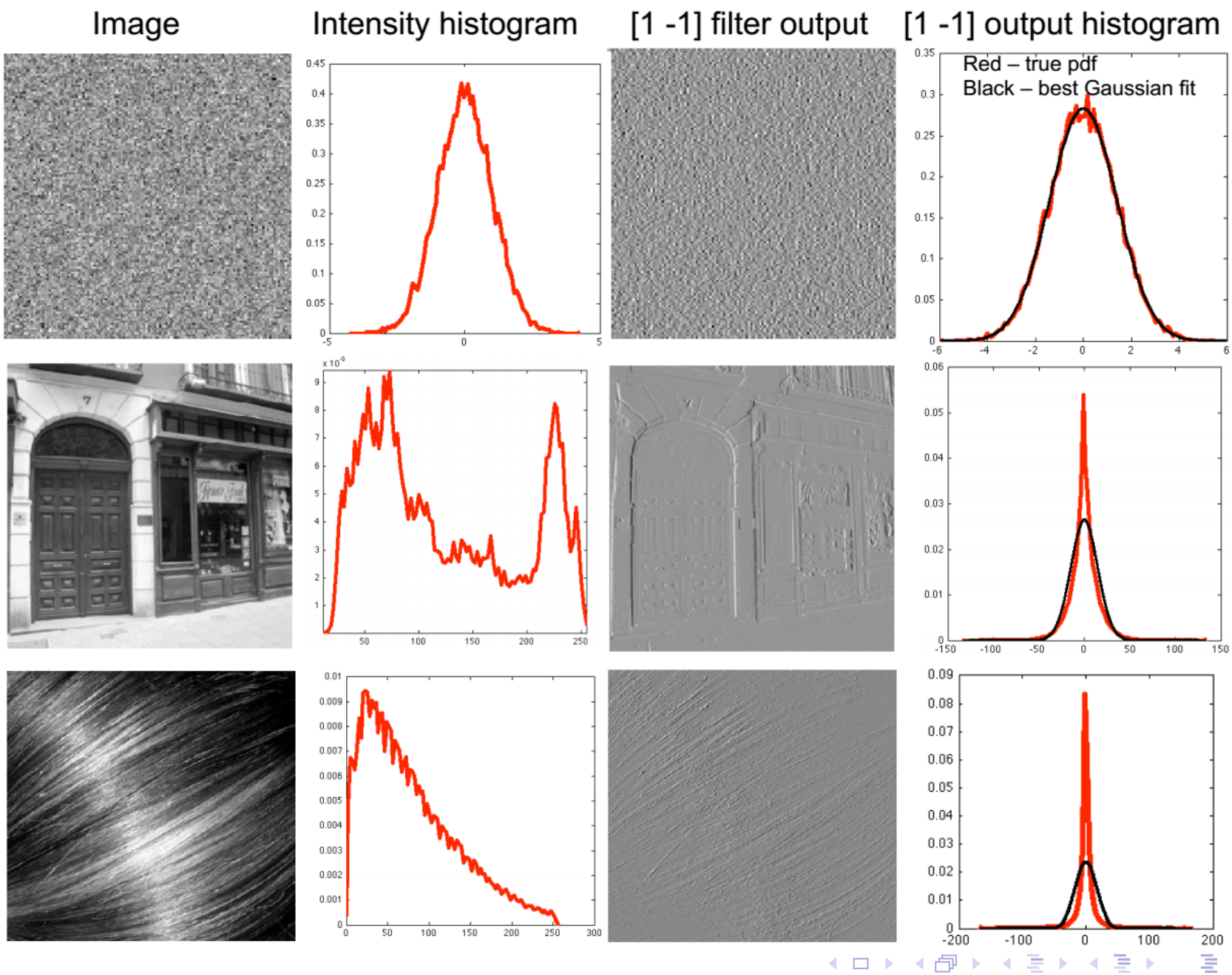


Abbildung: Magnetresonanztomografiebild des menschlichen Kopfes (T1-Bild). Links: geringer Kontrast. Rechts nach der Kontrastverstärkung.

Beispiel Histogramm



Sei G der Wertebereich im Bild. Die Spreizung des Histogramms kann berechnet werden durch: $T_S tretch(g) = (G$
g - $g_{min} g_{max} - g_{min})$

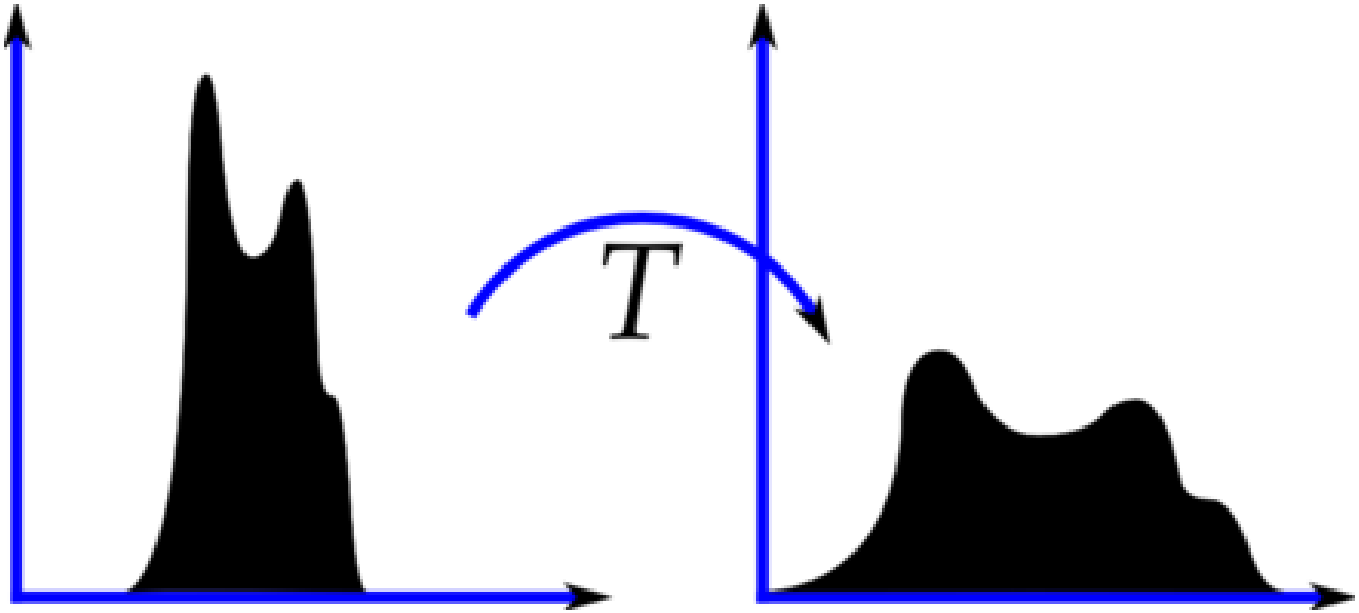


Abbildung: Histogrammspreizung

Reduktion der Farbtiefe

Zu den häufig angewendeten Punktoperationen zählt die Reduktion der Farbtiefe. Wollen wir zum Beispiel ein 14 Bit Bild, wie es vom Sensor einer Digitalkamera kommt, in ein 8 Bit Dateiformat schreiben, so muss die Farbtiefe reduziert werden.

Reduktion der Farbtiefe

Zu den häufig angewendeten Punktoperationen zählt die Reduktion der Farbtiefe. Wollen wir zum Beispiel ein 14 Bit Bild, wie es vom Sensor einer Digitalkamera kommt, in ein 8 Bit Dateiformat schreiben, so muss die Farbtiefe reduziert werden.
Hier wollen wir exemplarisch eine logarithmische Abbildung anstreben. Dazu muss der Wertebereich $[0, \ldots, 2^{14} - 1 = 16383]$ auf den Wertebereich $[0, \ldots, 2^8 - 1 = 255]$ skaliert werden. Dies kann man erreichen, indem man die Werte v nach der Gleichung

$$P(v) = (2^8 - 1)\frac{\log_{10}(v+1)}{\log_{10}(2^{14})}$$

ändert. Die Berechnung von $v+1$ ist hierbei nötig, um für Eingabewerte $v = 0$ sinnvolle Ergebnisse zu erhalten.

Reduktion der Farbtiefe

Da es hier nur 16384 unterschiedliche Eingabewerte gibt und ab Bildgrößen von 128×128 Pixeln mindestens genauso viele Bildpunkte, kann es sinnvoll sein, alle möglichen Ergebnisse vorzuberechnen und in einer Tabelle zu speichern. Dadurch kann vor allem bei großen Bildern eine wiederholte Auswertung der kostenintensiven Logarithmusfunktion vermieden werden. Im konkreten Fall würde man eine Liste mit 16384 8-Bit Werten im Speicher ablegen und auf jeden Wert per Indexoperation direkt zugreifen.

Hervorhebung der Beleuchtungsdifferenzen

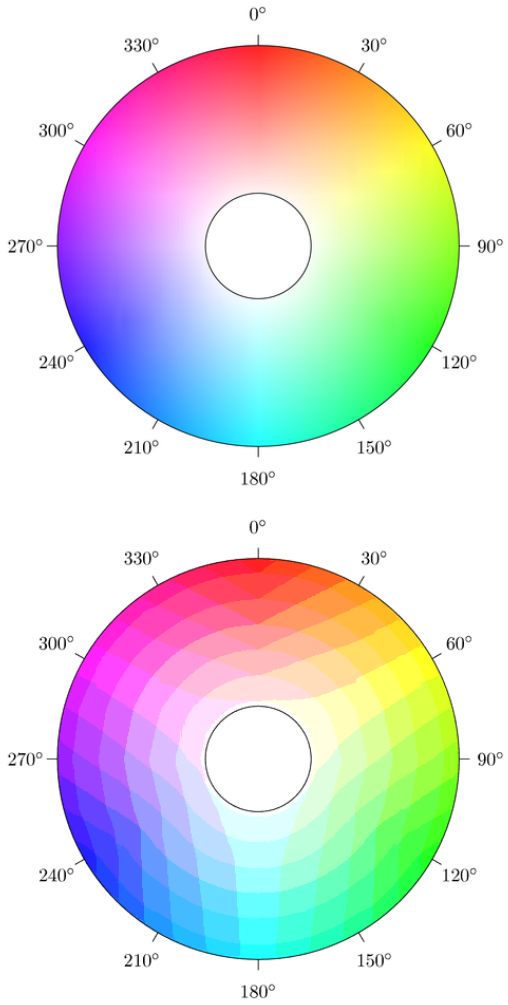
- Ziel: Hervorhebung von Farb-/Beleuchtungsdifferenzen
- Auge ist darauf trainiert, Helligkeitsunterschiede wahrzunehmen, Kanten werden schnell erkannt

Hervorhebung der Beleuchtungsdifferenzen

- Ziel: Hervorhebung von Farb-/Beleuchtungsdifferenzen
- Auge ist darauf trainiert, Helligkeitsunterschiede wahrzunehmen, Kanten werden schnell erkannt
- Sprünge im Farbverlauf heben Beleuchtungsdifferenzen hervor
- Pixelweises „Runden“ auf Zweierpotenzen:

$$P(v) = v \wedge \overline{(2^a - 1)}$$

Hervorhebung der Beleuchtungsdifferenzen



Fensterfunktionen

Im Gegensatz zu den bisherigen Operationen sind Fensterfunktionen **inhomogene** Punktoperationen, bei denen Teile des Bildes ausgeblendet oder verstärkt werden.

Im Gegensatz zu den bisherigen Operationen sind Fensterfunktionen **inhomogene** Punktoperationen, bei denen Teile des Bildes ausgeblendet oder verstärkt werden.
Sie werden angewendet, wenn Bildoperationen lokal ausgeführt werden sollen. Allgemein lassen sie sich als Multiplikation mit einer Fensterfunktion beschreiben

$$P(v_{m,n,o}) = K_{m,n,o} \cdot v_{m,n,o}$$

wobei in der Regel $K \in [0,1]$ gilt.

Bei einer geometrischen Transformation wird nicht der Wert, sondern lediglich die Position des Pixels oder Voxels geändert. Die Transformation der Position lässt sich hierbei auf zwei Arten mathematisch beschreiben:

1. über die Vorwärtstransformation

$$(m', n', o') = G(m, n, o)$$

2. über die Rückwärtstransformation

$$(m, n, o) = G^{-1}(m', n', o')$$

Annahme: die Funktion G ist wenigstens im Kontinuierlichen invertierbar.

Bei einer geometrischen Transformation wird nicht der Wert, sondern lediglich die Position des Pixels oder Voxels geändert. Die Transformation der Position lässt sich hierbei auf zwei Arten mathematisch beschreiben:

1. über die Vorwärtstransformation

$$(m', n', o') = G(m, n, o)$$

2. über die Rückwärtstransformation

$$(m, n, o) = G^{-1}(m', n', o')$$

Annahme: die Funktion G ist wenigstens im Kontinuierlichen invertierbar.
Obwohl dies mathematisch kein Unterschied ist, so sind damit bei diskreten Bildern unterschiedliche Auswertungsmethoden verbunden:

1. Im ersten Fall wird
   - das Originalbild durchlaufen
   - für jeden Punkt der zugehörige Punkt im Ergebnisbild berechnet
2. Im zweiten Fall wird
   - das Ergebnisbild durchlaufen
   - der zugehörige Punkt im Originalbild gesucht

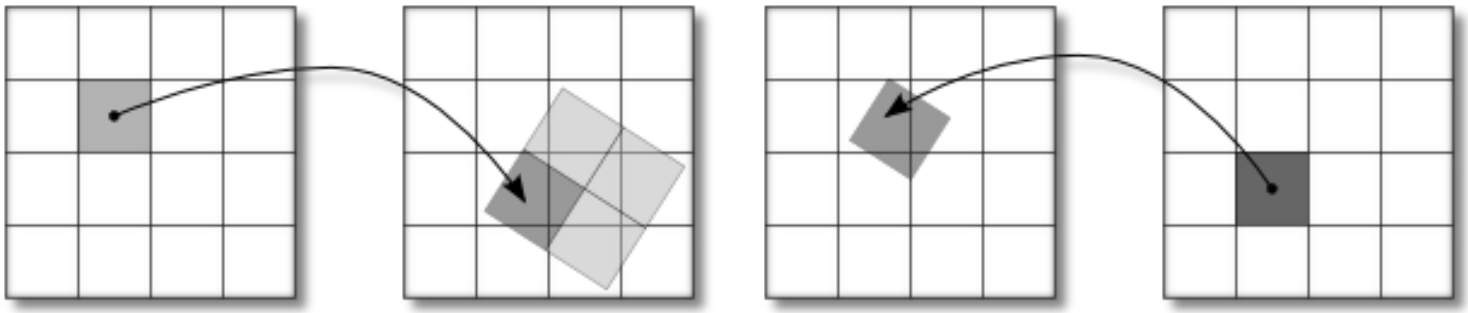


Abbildung: Links: Vorwärtstransformation. Rechts: Rückwärtstransformation.

Die häufig auftretenden affin-linearen Transformationen werden, wie auch in der Computergrafik üblich, über homogene Koordinaten beschrieben

$$\begin{pmatrix} \bar{m} \\ \bar{n} \\ \bar{o} \\ \bar{p} \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} m \\ n \\ o \\ 1 \end{pmatrix}$$

Die häufig auftretenden affin-linearen Transformationen werden, wie auch in der Computergrafik üblich, über homogene Koordinaten beschrieben

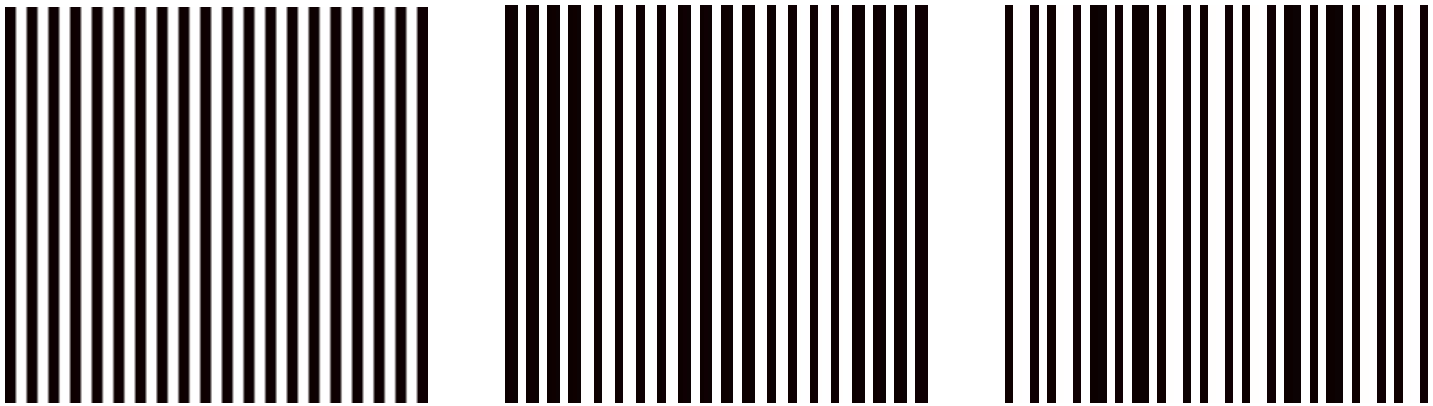
$$\begin{pmatrix} \bar{m} \\ \bar{n} \\ \bar{o} \\ \bar{p} \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} m \\ n \\ o \\ 1 \end{pmatrix}$$

Hierbei enthält der normierte 4D-Vektor

$$\begin{pmatrix} m' \\ n' \\ o' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{\bar{m}}{\bar{p}} \\ \frac{\bar{n}}{\bar{p}} \\ \frac{\bar{o}}{\bar{p}} \\ 1 \end{pmatrix}$$

die neue Position.

Ein Testmuster (links) im Original mit 153 Punkten mit 100 (Mitte) und 50 Punkten (rechts) abgetastet.



- Dirac-Delta (kontinuierlich):

$$\begin{array}{lcll} \delta & : & \mathbb{R} & \rightarrow & \mathbb{R} \cup \{\infty\} \\ \delta & : & x & \mapsto & \begin{cases} 0 & x \neq 0 \\ \infty & x = 0 \end{cases} \end{array}$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

- Dirac-Delta (kontinuierlich):

$$\begin{array}{lcll} \delta & : & \mathbb{R} & \to & \mathbb{R} \cup \{\infty\} \\ \delta & : & x & \mapsto & \begin{cases} 0 & x \neq 0 \\ \infty & x = 0 \end{cases} \end{array}$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

- Physiker:

$$\int_{-\infty}^{\infty} \delta(x) \cdot f(x) dx = f(0)$$

- Dirac-Delta (kontinuierlich):

$$\begin{array}{rcccl} \delta & : & \mathbb{R} & \to & \mathbb{R} \cup \{\infty\} \\ \delta & : & x & \mapsto & \begin{cases} 0 & x \neq 0 \\ \infty & x = 0 \end{cases} \end{array}$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

- Physiker:

$$\int_{-\infty}^{\infty} \delta(x) \cdot f(x) dx = f(0)$$

- Kronecker-Delta (diskret)

$$\delta : x \mapsto \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases}$$

Die Multiplikation im Ortsraum entspricht einer Faltung im Fourierraum.
Die Fouriertransformierte des Dirac-Kamms

$$\Delta_T = \sum_n \delta(x - x_n)$$

mit $x_n = n \cdot T$ ist wiederum ein Dirac-Kamm

$$\Delta_T(t) \circ\!\!-\!\!- \Delta_{\frac{1}{T}}(k)$$

Abtastung

Mit Hilfe der Fouriertransformation ist es möglich, die Abtastung von Signalen genauer zu betrachten.

Mit Hilfe der Fouriertransformation ist es möglich, die Abtastung von Signalen genauer zu betrachten.
Sei $x_n = n \cdot T$ mit der Abtastfrequenz T.

Mit Hilfe der Fouriertransformation ist es möglich, die Abtastung von Signalen genauer zu betrachten.
Sei $x_n = n \cdot T$ mit der Abtastfrequenz T.
Man definiert das Abtasten eines Signals f durch

$$f'(x) = f(x) \cdot \sum_{n=-\infty}^{\infty} \delta(x - x_n)$$

Mit Hilfe der Fouriertransformation ist es möglich, die Abtastung von Signalen genauer zu betrachten.
Sei $x_n = n \cdot T$ mit der Abtastfrequenz T.
Man definiert das Abtasten eines Signals f durch

$$f'(x) = f(x) \cdot \sum_{n=-\infty}^{\infty} \delta(x - x_n)$$

Dann wird das ganze, abgetastete Bild als analoges Bild dargestellt:

$$f'(x_k) = f(x_k) \cdot \sum_{n=-\infty}^{\infty} \delta(x_k - x_n)$$

Sei P ein analoges Bild, das keine Frequenzen größer als $|k| = k_{\text{max}}$ enthält.
Dann lässt sich das Bild eindeutig und exakt rekonstruieren, wenn es mit einer Abtastrate (Abtastfrequenz)

$$k_{\text{abtast}} > 2k_{\text{max}}$$

abgetastet wurde.

Rekonstruktion und Interpolation

Die Interpolation dient zur Konstruktion eines kontinuierlichen Bildes v_r aus den vorhandenen diskreten Daten $v(p)$ an den Positionen r_p.

Rekonstruktion und Interpolation

Die Interpolation dient zur Konstruktion eines kontinuierlichen Bildes v_r aus den vorhandenen diskreten Daten $v(p)$ an den Positionen r_p.
Die einfachsten Interpolationstechniken für reguläre Gitter lassen sich hierbei als lineare, verschiebungsinvariante Operationen mittels Faltung des Bildes mit einer Wichtungsfunktion h schreiben:

$$v_r(x) = \sum_p h(x - r_p) \cdot v(p)$$

Die Interpolation dient zur Konstruktion eines kontinuierlichen Bildes v_r aus den vorhandenen diskreten Daten $v(p)$ an den Positionen r_p.
Die einfachsten Interpolationstechniken für reguläre Gitter lassen sich hierbei als lineare, verschiebungsinvariante Operationen mittels Faltung des Bildes mit einer Wichtungsfunktion h schreiben:

$$v_r(x) = \sum_p h(x - r_p) \cdot v(p)$$

Um den Bezug zur Signalverarbeitung herzustellen, lässt sich das diskrete Bild als gewichteter Dirac-Kamm mit Hilfe der Dirac-δ-Funktion ausdrücken

$$v(p) = \int_{\mathbb{R}} \delta(r_p - x) \cdot v(p) dx$$

Die Interpolation dient zur Konstruktion eines kontinuierlichen Bildes v_r aus den vorhandenen diskreten Daten $v(p)$ an den Positionen r_p.
Die einfachsten Interpolationstechniken für reguläre Gitter lassen sich hierbei als lineare, verschiebungsinvariante Operationen mittels Faltung des Bildes mit einer Wichtungsfunktion h schreiben:

$$v_r(x) = \sum_p h(x - r_p) \cdot v(p)$$

Um den Bezug zur Signalverarbeitung herzustellen, lässt sich das diskrete Bild als gewichteter Dirac-Kamm mit Hilfe der Dirac-δ-Funktion ausdrücken

$$v(p) = \int_{\mathbb{R}} \delta(r_p - x) \cdot v(p) dx$$

Dadurch wird die Faltung als kontinuierliche Operation ersichtlich

$$\begin{aligned} v_r(x) &= \sum_p \int_R h(x - x') \cdot \delta(r_p - x') \cdot v(x') dx' \\ &= \int_R h(x - x') \cdot \sum_p \delta(r_p - x') v(x') dx' \end{aligned}$$

Die Fouriertransformierte ergibt sich über das Faltungstheorem als

$$\hat{v}_r(k) = \hat{h}(k) \sum_u \hat{v}(k - \hat{r}_u)$$

Hierbei stellt $\hat{h}(k)$ einen Dirac-Kamm im Fourierraum mit zu dem Kamm im Bildraum inversen Abständen dar.

Die Fouriertransformierte ergibt sich über das Faltungstheorem als

$$\hat{v}_r(k) = \hat{h}(k) \sum_u \hat{v}(k - \hat{r}_u)$$

Hierbei stellt $\hat{h}(k)$ einen Dirac-Kamm im Fourierraum mit zu dem Kamm im Bildraum inversen Abständen dar.
Hieraus ist ersichtlich, dass ein Signal genau dann korrekt rekonstruiert werden kann, wenn die Rekonstruktion im Bildraum durch die Multiplikation mit einem Rechteckfilter im Frequenzraum (einem idealen Tiefpass) nach der Formel

$$\hat{v}_r(k) = \hat{h}(k) \cdot \hat{v}(k) = \hat{R}\left(\frac{k \cdot \delta x}{2 \cdot \pi}\right) \cdot \hat{v}(k)$$

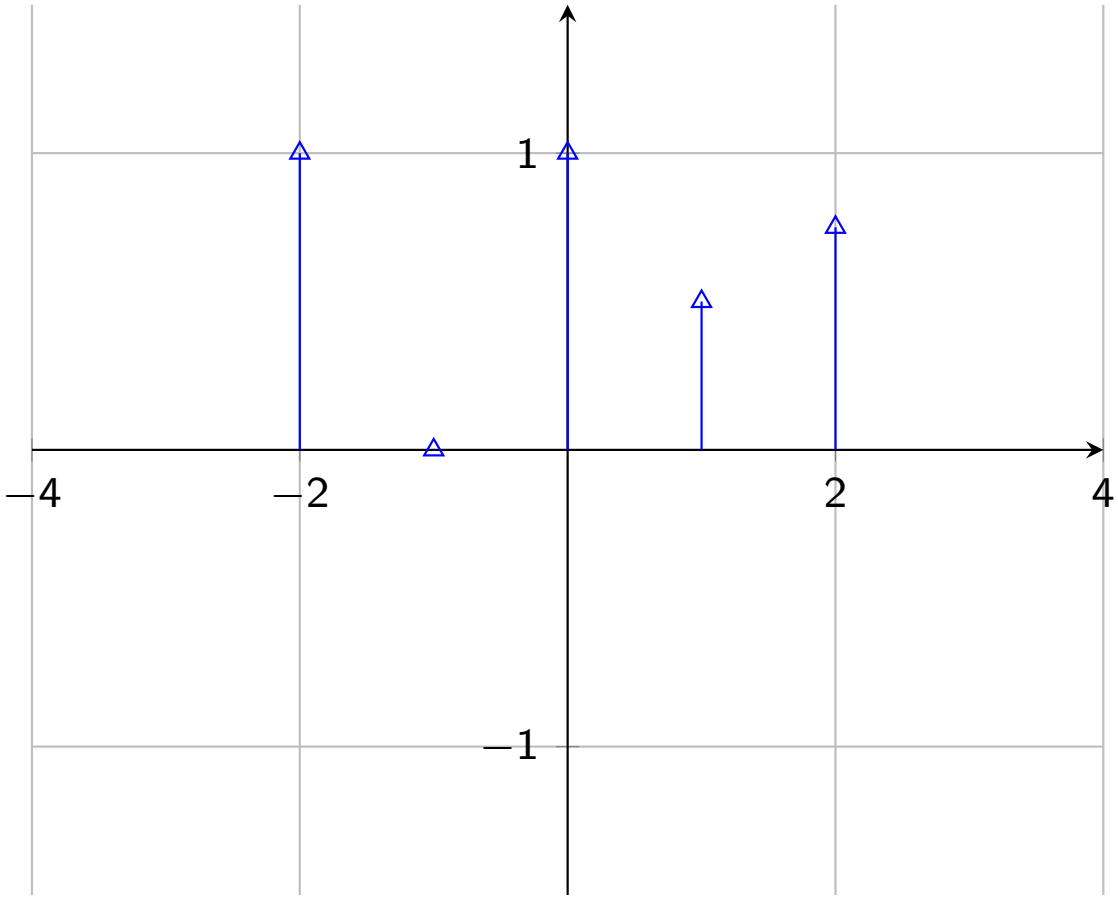
mit der Rechtecksfunktion

$$\hat{R}(j) = \begin{cases} 1 & |k| \leq j \\ 0 & sonst \end{cases}$$

dargestellt werden kann.

Dieser Zusammenhang wird in der Signalverarbeitung **Abtasttheorem** genannt.

Die Rekonstruktion (schwarz) aus mehreren Rekonstruktionskernen (grau und rot) aus einem abgetasteten Signal (blau) kann man sich graphisch folgendermaßen vorstellen.



In der Praxis ist es erstrebenswert Methoden der Interpolation zu wählen, die sowohl im Zeit- als auch im Frequenzbereich hinreichend gute Eigenschaften haben.

In der Praxis ist es erstrebenswert Methoden der Interpolation zu wählen, die sowohl im Zeit- als auch im Frequenzbereich hinreichend gute Eigenschaften haben.
Leider fällt die *sinc*-Funktion nur recht langsam nach außen gegen Null ab, wodurch sie in der Anwendung als Filter oft ungeeignet ist.
Geht man davon aus, dass man recht weit von der minimalen Abtastfrequenz entfernt ist, so lässt sich zum Beispiel der Rechteckfilter im Frequenzbereich durch glättere Filter annähern.

Konstante Interpolation

Sind die Gitterpunkte an den diskreten Stellen n gegeben, so kann man die konstante Interpolation als Filterung mit der Rechtecksmaske

$$h(x) = \begin{cases} 1 & \text{für } |x| < 0,5 \\ 0 & \textit{sonst} \end{cases}$$

verstehen.

Konstante Interpolation

Sind die Gitterpunkte an den diskreten Stellen n gegeben, so kann man die konstante Interpolation als Filterung mit der Rechtecksmaske

$$h(x) = \begin{cases} 1 & \text{für } |x| < 0,5 \\ 0 & \textit{sonst} \end{cases}$$

verstehen.
Die Fouriertransformation des Rechtecks der Breite 1 ergibt dann

$$\hat{h}(k) = \frac{\sin(\pi k)}{\pi k} = \operatorname{sinc}(k)$$

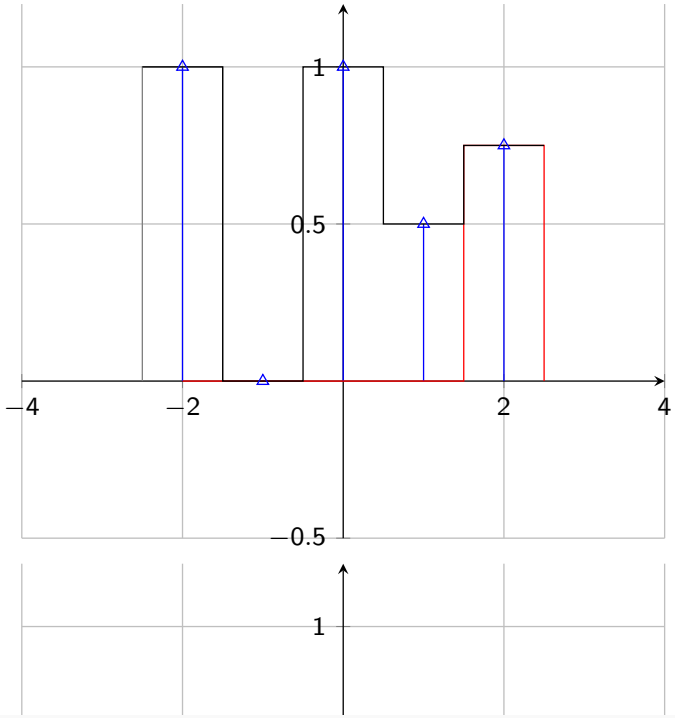
Konstante Interpolation

Sind die Gitterpunkte an den diskreten Stellen n gegeben, so kann man die konstante Interpolation als Filterung mit der Rechtecksmaske

$$h(x) = \begin{cases} 1 & \text{für } |x| < 0,5 \\ 0 & \textit{sonst} \end{cases}$$

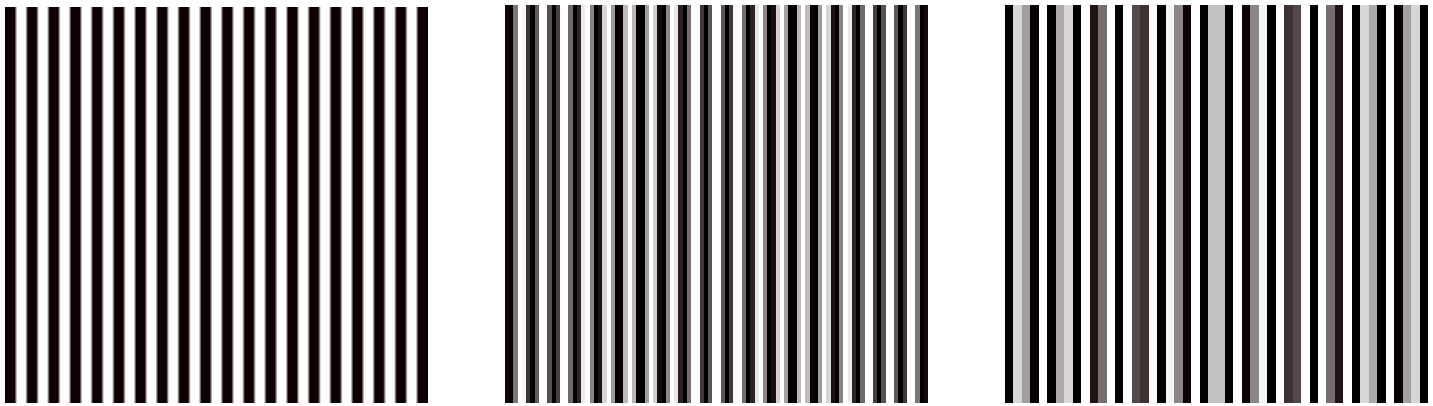
verstehen.
Die Fouriertransformation des Rechtecks der Breite 1 ergibt dann

$$\hat{h}(k) = \frac{\sin(\pi k)}{\pi k} = \operatorname{sinc}(k)$$



Konstante Interpolation

Das Testmuster aus der Einleitung (links) im Original mit 153 Punkten mit 100 (Mitte) und 50 Punkten (rechts) abgetastet; konstante Interpolation (sinc):



Lineare Interpolation

Sind die Gitterpunkte an den diskreten Stellen n gegeben, so kann man die lineare Interpolation als Filterung mit der Dreiecksmaske

$$h(x) = \begin{cases} 1 - |x| & \text{für } |x| < 1 \\ 0 & \textit{sonst} \end{cases}$$

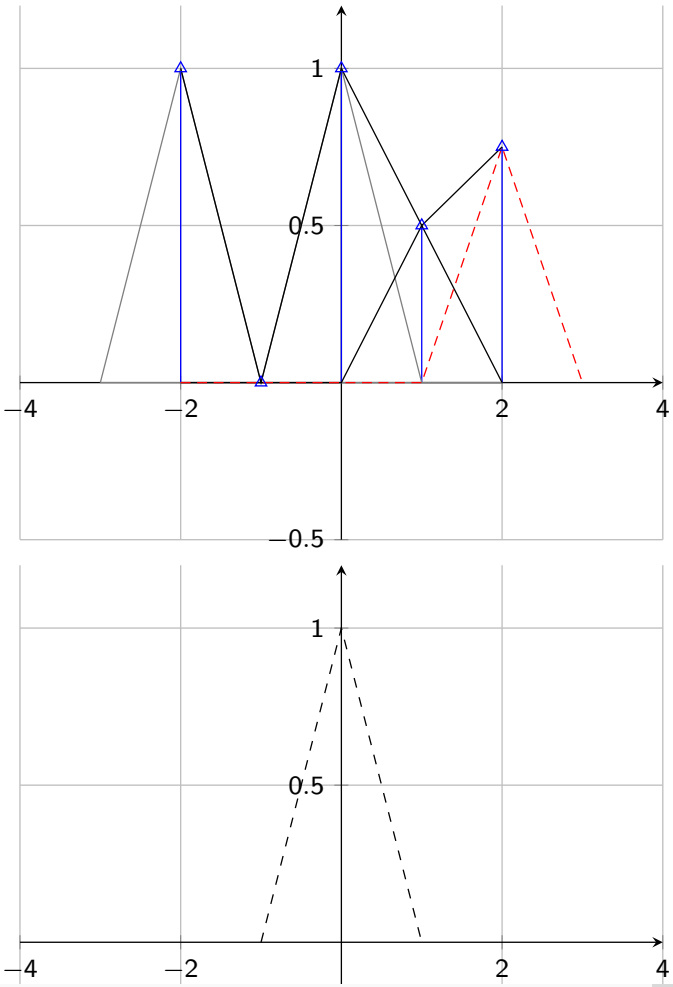
verstehen.

Lineare Interpolation

Sind die Gitterpunkte an den diskreten Stellen n gegeben, so kann man die lineare Interpolation als Filterung mit der Dreiecksmaske

$$h(x) = \begin{cases} 1 - |x| & \text{für } |x| < 1 \\ 0 & \textit{sonst} \end{cases}$$

verstehen.



Dies zeigt zwei Nachteile der linearen Interpolation auf, die jedoch häufig akzeptiert werden:

1. Während kleine Wellenzahlen, insbesondere der Mittelwert der Bilder bei $k = 0$, korrekt interpoliert werden, werden hohe Wellenzahlen in ihrer Amplitude etwas reduziert, was zu einer leichten Glättung der Daten führt. Bei $k = 1$ (Nyquistfrequenz) nimmt die Transferfunktion den Wert

$$\hat{h}(1) = (2/\pi)^2 \approx 0.41$$

an.

Dies zeigt zwei Nachteile der linearen Interpolation auf, die jedoch häufig akzeptiert werden:

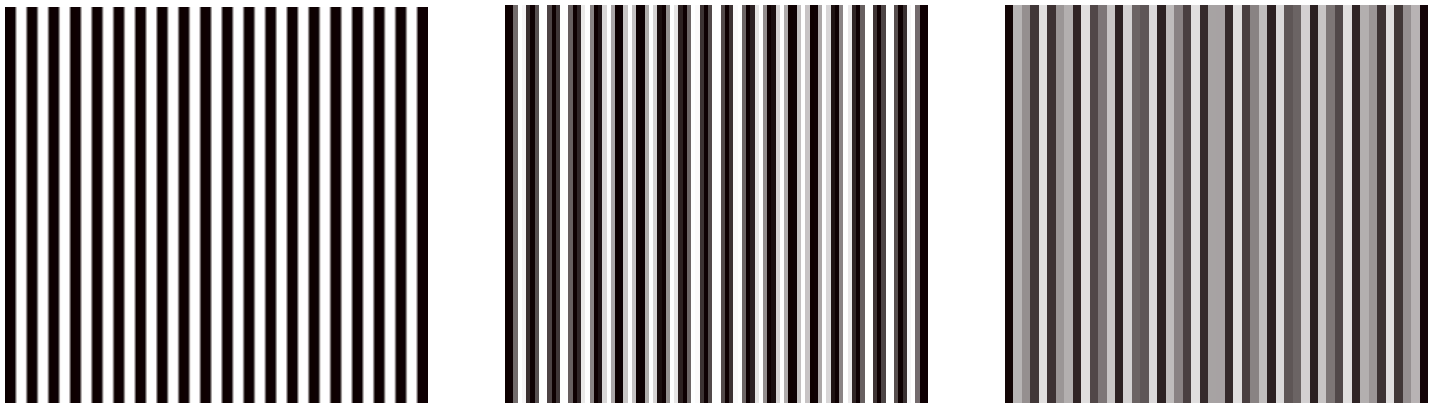
1. Während kleine Wellenzahlen, insbesondere der Mittelwert der Bilder bei $k = 0$, korrekt interpoliert werden, werden hohe Wellenzahlen in ihrer Amplitude etwas reduziert, was zu einer leichten Glättung der Daten führt. Bei $k = 1$ (Nyquistfrequenz) nimmt die Transferfunktion den Wert

$$\hat{h}(1) = (2/\pi)^2 \approx 0.41$$

   an.
2. Da $\hat{h}(k)$ für $k > 1$ nicht gleich 0 ist, werden einige falsche hohe Wellenzahlen erzeugt. Wird das kontinuierlich interpolierte Bild erneut abgetastet, so ergeben sich dadurch **Aliasingeffekte**.

Lineare Interpolation

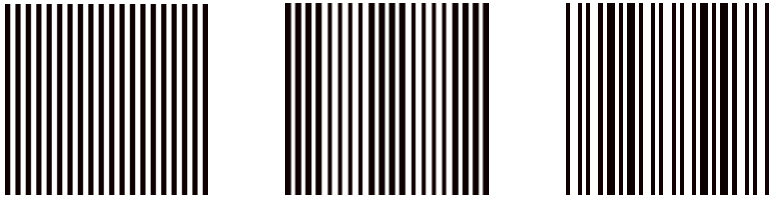
Das Testmuster aus der Einleitung (links) im Original mit 153 Punkten mit 100 (Mitte) und 50 Punkten (rechts) abgetastet; Lineare Interpolation (Dreiecksfunktion):



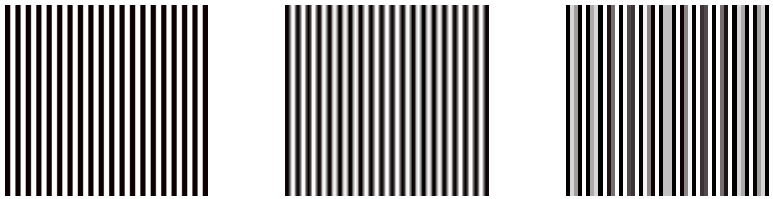
Vergleich der Interpolationsarten

Das Testmuster aus der Einleitung (links) im Original mit 153 Punkten mit 100 (Mitte) und 50 Punkten (rechts) abgetastet.

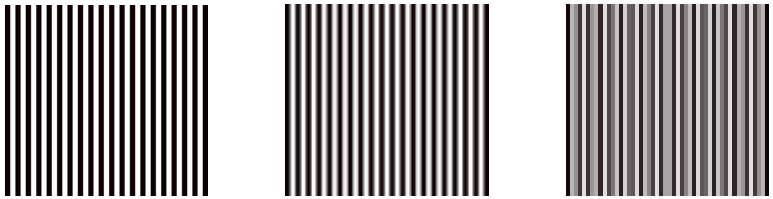
Keine Interpolation



Konstante Interpolation (sinc)



Lineare Interpolation (Dreiecksfunktion)



Teil VIII

Regionendefinition

Bildverarbeitungsprozess

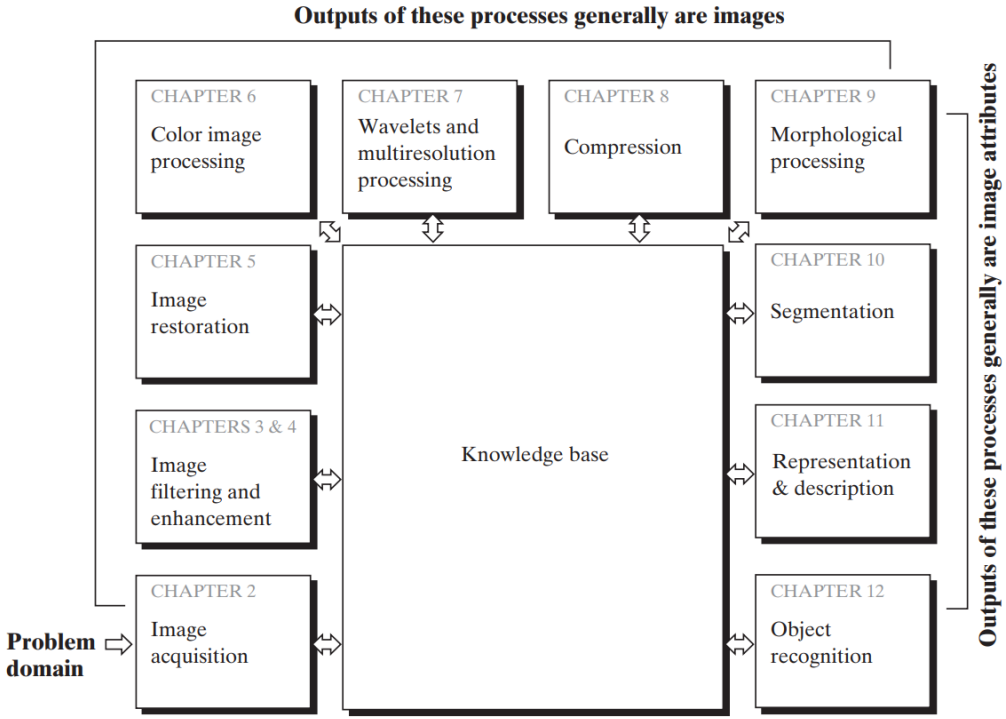


Abbildung: Die Bildverarbeitung umfasst eine reihe von Algorithmen, die entweder wiederum Bilder oder Bildattribute ausgeben. [Digital Image Processing, 3rd Edition, Gonzales and Woods]

Diese Vorlesung: Bildsegmentierung

Bildsegmentierung

Die Bildsegmentierung beschäftigt sich mit der Unterteilung eines Bildes in Regionen.
Diese Ansätze werden dazu verwendet, Objekte zu erkennen.

Bildsegmentierung

Die Bildsegmentierung beschäftigt sich mit der Unterteilung eines Bildes in Regionen.
Diese Ansätze werden dazu verwendet, Objekte zu erkennen.
Die Methoden des vorangegangenen Kapitels hingegen haben sich mit der Identifizierung von Kanten oder Texturgrenzen beschäftigt.

Pixelorientierte Segmentierung

Die einfachste Art der Segmentierung ist die pixelorientierte Segmentierung.

- Für die Segmentierung wird lediglich der Grauwert des Pixels/Voxels berücksichtigt.
- Hierbei spielen Nachbarschaftsinformationen keine Rolle.

Pixelorientierte Segmentierung

Die einfachste Art der Segmentierung ist die pixelorientierte Segmentierung.

- Für die Segmentierung wird lediglich der Grauwert des Pixels/Voxels berücksichtigt.
- Hierbei spielen Nachbarschaftsinformationen keine Rolle.

Für die Bestimmung des Schwellwertes ist es naheliegend, das Histogramm der Grauwerte zu betrachten und daraus Werte abzuleiten für

- den Vordergrund und den Hintergrund
- unterschiedliche Objekte

Die Computertomographie ist ein Verfahren zur Erstellung dreidimensionaler Röntgenbilder. Hier ist es üblich, die gemessenen Werte zu normieren. Hierzu wird der CT-Wert basierend auf des Abschwächungskoeffizienten des betrachteten Gewebes relativ zu Wasser definiert als

$$CT(\mu_{\text{Gewebe}}) := \frac{\mu_{\text{Gewebe}} - \mu_{\text{Wasser}}}{\mu_{\text{Wasser}}} \cdot 1000 \text{ HU}$$

Der Wertebereich ist dabei theoretisch in beide Richtungen offen.

Es ergeben sich folgende Richtwerte verschiedener Stoffe und Gewebe[6]

- **Wasser** hat definitionsgemäß 0 HU.
- **Fettgewebe** absorbieren Röntgenstrahlen minimal weniger als Wasser und sind um -100 HU angesiedelt.
- **Luft** absorbiert Röntgenstrahlen kaum und ihr Wert wird auf -1000 HU definiert. Bei der Kalibrierung wird für Luft $\mu_{\mathrm{Luft}} = 0$ angenommen.

[6] Die Werte wurden von `http://de.wikipedia.org/wiki/Hounsfield-Skala` übernommen

Es ergeben sich folgende Richtwerte verschiedener Stoffe und Gewebe[6]

- **Wasser** hat definitionsgemäß 0 HU.
- **Fettgewebe** absorbieren Röntgenstrahlen minimal weniger als Wasser und sind um -100 HU angesiedelt.
- **Luft** absorbiert Röntgenstrahlen kaum und ihr Wert wird auf -1000 HU definiert. Bei der Kalibrierung wird für Luft $\mu_{\mathrm{Luft}} = 0$ angenommen.
- **Knochen** absorbieren Röntgenstrahlen deutlich und haben je nach Dichte Werte im Bereich von 500-1500 HU.
- **Kontrastmittel** haben, je nach Art und Konzentration Werte im Bereich 100-300 HU.

[6] Die Werte wurden von `http://de.wikipedia.org/wiki/Hounsfield-Skala` übernommen

Tabelle: Beispiel der Klassifikation anhand von Hounsfield-Einheiten.

HU Bereich		Gewebeart	Farbe im Bild
-300	– 0	Fettgewebe	Gelb
1	– 300	Proteinreiches Gewebe (z.B. Muskeln)	Rot
301	– 3000	Knochen	Weiß

Hounsfield-Einheiten zur Klassifizierung Tumorgewebe ist recht nah an Fettgewebe, deshalb schwer erkennbar

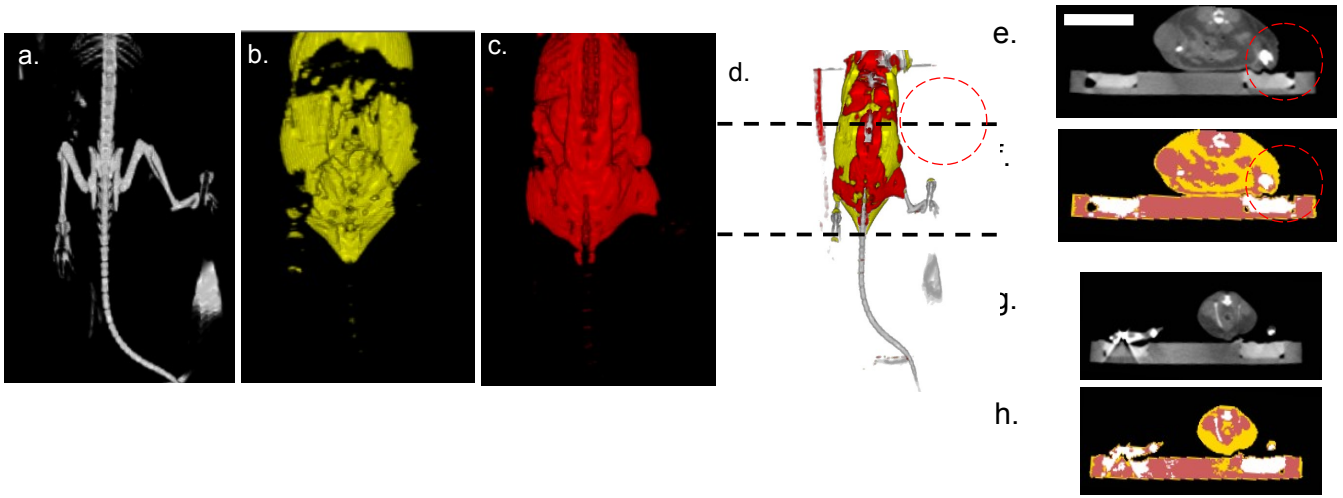


Abbildung: Klassifikation von Gewebe am Beispiel einer Maus. Ein Volumenbild wurde pixelbasiert nach Tabelle **??** segmentiert und Knochen, Fettgewebe und proteinreiches Gewebe in a-c

Schwellwerte

Bei Messdaten müssen vor der Segmentierung Helligkeitsunterschiede ausgeglichen werden, welche eine Verwendung eines einheitlichen Schwellwerts ausschließen.

Kantenbasierte Segmentierung

Wenn Helligkeitsverläufe eine Rolle spielen, ist eine grauwertbasierte Segmentierung oft nicht ausreichend, da sich durch überlappende Spektren nur selten Schwellwerte definieren lassen.

Kantenbasierte Segmentierung

Wenn Helligkeitsverläufe eine Rolle spielen, ist eine grauwertbasierte Segmentierung oft nicht ausreichend, da sich durch überlappende Spektren nur selten Schwellwerte definieren lassen.
Diese Probleme der unterschiedlichen Grauwerte für Objekte sowie des Einflusses unterschiedlicher Lichtverhältnisse bei der Aufnahme lassen sich durch kantenbasierte Verfahren umgehen.

Kantenbasierte Segmentierung

- Aus dem Kapitel zur Kantenextraktion wissen wir, dass Extrema der ersten Ableitung sowie Nulldurchgänge (Vorzeichenwechsel) der zweiten Ableitung auf Kanten hinweisen.

Kantenbasierte Segmentierung

- Aus dem Kapitel zur Kantenextraktion wissen wir, dass Extrema der ersten Ableitung sowie Nulldurchgänge (Vorzeichenwechsel) der zweiten Ableitung auf Kanten hinweisen.
- Nimmt man den Ansatz über die zweiten Ableitungen, so ist dieser unabhängig von monotonen Helligkeitsverläufen im Bild.

Kantenbasierte Segmentierung

- Aus dem Kapitel zur Kantenextraktion wissen wir, dass Extrema der ersten Ableitung sowie Nulldurchgänge (Vorzeichenwechsel) der zweiten Ableitung auf Kanten hinweisen.
- Nimmt man den Ansatz über die zweiten Ableitungen, so ist dieser unabhängig von monotonen Helligkeitsverläufen im Bild.
- Die kantenbasierte Segmentierung ist nun ein sequentieller Prozess:
  1. Zuerst wird das Bild mit einem Kantenfilter durchlaufen wird.
  2. Danach „sucht“ man im Bild nach Pixeln, die auf einer Kante liegen.
  3. Hat man eine Kante gefunden, wird diese so lange verfolgt, bis man wieder am Anfangspunkt angekommen ist.

Kantenbasierte Segmentierung

- Aus dem Kapitel zur Kantenextraktion wissen wir, dass Extrema der ersten Ableitung sowie Nulldurchgänge (Vorzeichenwechsel) der zweiten Ableitung auf Kanten hinweisen.
- Nimmt man den Ansatz über die zweiten Ableitungen, so ist dieser unabhängig von monotonen Helligkeitsverläufen im Bild.
- Die kantenbasierte Segmentierung ist nun ein sequentieller Prozess:
  1. Zuerst wird das Bild mit einem Kantenfilter durchlaufen wird.
  2. Danach „sucht“ man im Bild nach Pixeln, die auf einer Kante liegen.
  3. Hat man eine Kante gefunden, wird diese so lange verfolgt, bis man wieder am Anfangspunkt angekommen ist.
- Im Idealfall erhält man so alle geschlossenen Ränder im Bild.

Regionenwachstum (Region Growing)

- Bei diesem Verfahren zielt man ausgehend von Saatvoxeln auf eine sukzessive Ausdehnung von Regionen.
- Dabei werden zu jeder Region die Randvoxel bestimmt und dann deren Nachbarn getestet.
- Wenn ein Nachbar noch zu keiner Region gehört und sein Wert vom Mittelwert der Region wenig abweicht so wird das Voxel in die Region aufgenommen.
- „wenig“ ist hierbei oft als Benutzerparameter vorgegeben.

Regionentrennung (Region Splitting)

1. Bei diesem Verfahren wird zunächst das gesamte Bild als eine Region definiert.
2. Diese wird dann in Oktanten zerlegt, sofern die Abweichung vom Mittelwert eines Voxels zu groß ist.
3. Dies wird fortgesetzt, bis die Abweichung klein genug ist – notfalls bis zum isolierten Voxel.

Regionentrennung (Region Splitting)

1. Bei diesem Verfahren wird zunächst das gesamte Bild als eine Region definiert.
2. Diese wird dann in Oktanten zerlegt, sofern die Abweichung vom Mittelwert eines Voxels zu groß ist.
3. Dies wird fortgesetzt, bis die Abweichung klein genug ist – notfalls bis zum isolierten Voxel.

- Wenn die Dicke einer Region in eine Achsenrichtung 1 erreicht, unterteilt man nur noch in Quadranten.
- Wenn in zwei Richtungen die Dicke 1 erreicht ist, wird nur noch halbiert.
- Der Vorteil dieses Verfahrens ist, dass die Regionen durch einen Octree beschrieben werden können.

Regionenverschmelzung (Region Merging)

- Hier ist zunächst jeder Voxel eine eigene Region.
- Die Regionen werden verschmolzen, wenn die Abweichung vom gemeinsamen Mittelwert unter der vom Anwender angegebenen Grenze bleibt.
- Dabei wird in der Regel in der Reihenfolge Schicht-Zeile-Spalte vorgegangen, also zunächst in positiver z-Richtung nach geeigneten Nachbarregionen für den Fusionsprozess gesucht.

Region Split-Merge

- Nach dem Trennen gibt es in der Regel ein „übersegmentiertes“ Bild, in dem man durch Verschmelzen ausreichend wenig abweichender Regionen eine bessere Segmentierung erreichen kann.
- Wenn mehrere Regionen zur Vereinigung in Frage kommen, wird diejenige mit der geringeren Abweichung ausgewählt.
- Der Vereinigungsschritt ist allerdings sehr aufwändig.

Pyramid Linking

- Die bisherigen Verfahren funktionieren nur gut, wenn Objekt und Hintergrund schön getrennte uniforme Werte aufweisen und keine fließenden Übergänge vorhanden sind.
- Wenn dies nicht der Fall ist, erzeugt man zuerst ein Merkmalsbild und segmentiert dieses.

Pyramid Linking

- Die bisherigen Verfahren funktionieren nur gut, wenn Objekt und Hintergrund schön getrennte uniforme Werte aufweisen und keine fließenden Übergänge vorhanden sind.
- Wenn dies nicht der Fall ist, erzeugt man zuerst ein Merkmalsbild und segmentiert dieses.
- Einem Objekt sollen nur die Teile zugeordnet werden, welche zweifelsfrei zu ihm gehören.
- In der Übergangszone zum nächsten Objekt oder zum Hintergrund soll möglichst keine Glättung in die Merkmalsbestimung miteinbezogen werden.
- Da das Objekt zuerst gefunden werden muss, bevor man an seinen Grenzen die Masken zur Merkmalsbestimmung verkleinern kann, ist hier ein mehrstufiger Prozess nötig.

Algorithmus von Burt [a]

[a] [Burt, The Pyramid as a structure for efficient computation. In Rosenfeld (Hrsg.), Multiresolution Image Processing and Analysis, John Wiley & Sons, 2001, S. 269-307.]

- Durch einen Rechteckfilter der Länge vier wird eine Gausspyramide erzeugt, wobei in jedem Schritt die Auflösung halbiert wird.
  - Jeder Bildpunkt einer Ebene der Pyramide gehört nun zu zwei Punkten der nächsten Ebene geringerer Auflösung.

Algorithmus von Burt [a]

[a] [Burt, The Pyramid as a structure for efficient computation. In Rosenfeld (Hrsg.), Multiresolution Image Processing and Analysis, John Wiley & Sons, 2001, S. 269-307.]

- Durch einen Rechteckfilter der Länge vier wird eine Gausspyramide erzeugt, wobei in jedem Schritt die Auflösung halbiert wird.
  - Jeder Bildpunkt einer Ebene der Pyramide gehört nun zu zwei Punkten der nächsten Ebene geringerer Auflösung.
- Jetzt wird jeder Punkt dem Punkt der nächsten Ebene zugeordnet, dessen Grauwert näher an dem Wert des Punktes liegt.
  - Dadurch entsteht eine Baumstruktur.

Algorithmus von Burt [a]

[a] [Burt, The Pyramid as a structure for efficient computation. In Rosenfeld (Hrsg.), Multiresolution Image Processing and Analysis, John Wiley & Sons, 2001, S. 269-307.]

- Durch einen Rechteckfilter der Länge vier wird eine Gausspyramide erzeugt, wobei in jedem Schritt die Auflösung halbiert wird.
  - Jeder Bildpunkt einer Ebene der Pyramide gehört nun zu zwei Punkten der nächsten Ebene geringerer Auflösung.
- Jetzt wird jeder Punkt dem Punkt der nächsten Ebene zugeordnet, dessen Grauwert näher an dem Wert des Punktes liegt.
  - Dadurch entsteht eine Baumstruktur.
- Die Werte der Vaterknoten werden jeweils auf die Mittelwerte all ihrer Kinderknoten gesetzt.
  - Dabei durchläuft man den Baum von den Blättern bis zur Wurzel (Tiefensuche, depth-first).

Algorithmus von Burt [a]

[a] [Burt, The Pyramid as a structure for efficient computation. In Rosenfeld (Hrsg.), Multiresolution Image Processing and Analysis, John Wiley & Sons, 2001, S. 269-307.]

- Durch einen Rechteckfilter der Länge vier wird eine Gausspyramide erzeugt, wobei in jedem Schritt die Auflösung halbiert wird.
  - Jeder Bildpunkt einer Ebene der Pyramide gehört nun zu zwei Punkten der nächsten Ebene geringerer Auflösung.
- Jetzt wird jeder Punkt dem Punkt der nächsten Ebene zugeordnet, dessen Grauwert näher an dem Wert des Punktes liegt.
  - Dadurch entsteht eine Baumstruktur.
- Die Werte der Vaterknoten werden jeweils auf die Mittelwerte all ihrer Kinderknoten gesetzt.
  - Dabei durchläuft man den Baum von den Blättern bis zur Wurzel (Tiefensuche, depth-first).
- Die letzten beiden Schritte werden so lange wiederholt, bis sich eine stabile Lösung eingestellt hat.

Algorithmus von Burt [a]

[a] [Burt, The Pyramid as a structure for efficient computation. In Rosenfeld (Hrsg.), Multiresolution Image Processing and Analysis, John Wiley & Sons, 2001, S. 269-307.]

Die Segmentierung ergibt sich nun aus den Trennlinien des Graphen mit den größten Grauwertunterschieden. Dies ist insbesondere dort der Fall, wo sich sich Knoten einer Ebene deutlich stärker unterscheiden als die darunter liegenden Ebenen.

Pyramid Linking: Initiales Averaging

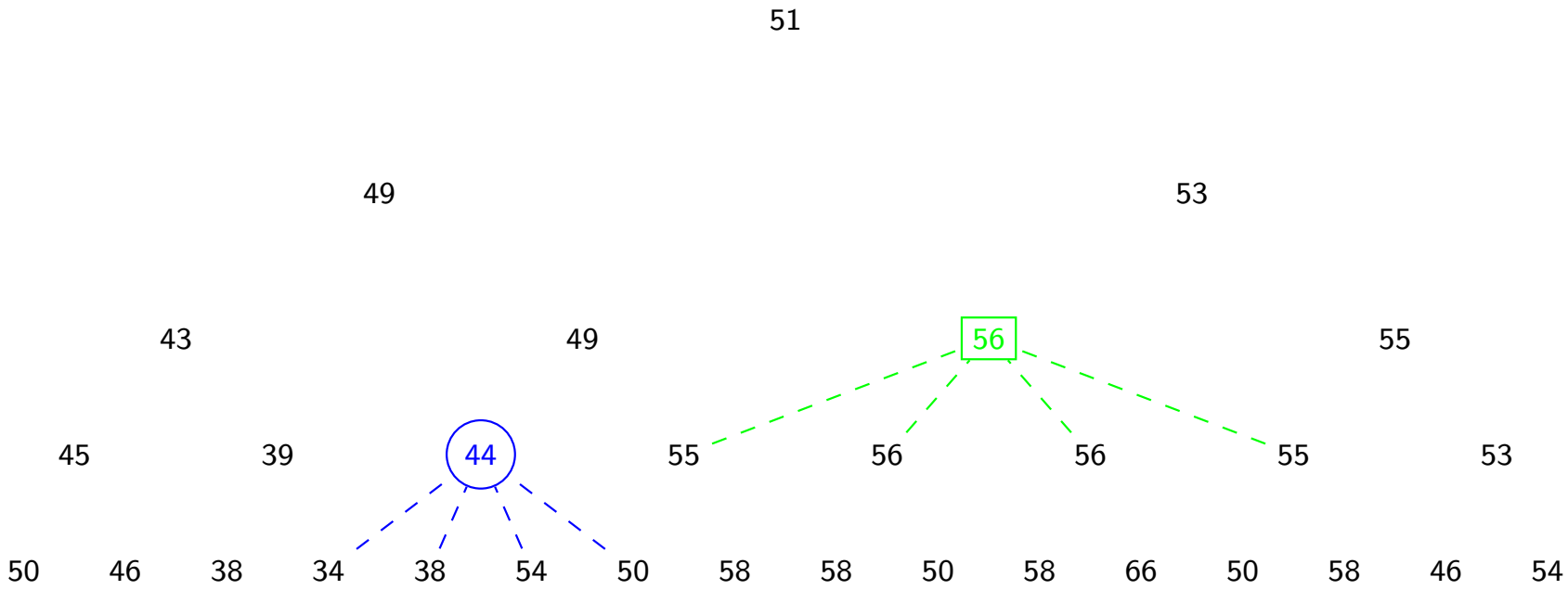


Abbildung: Initialisierung der Baumstruktur: Mittelung von 4 Nachbarn

Pyramid Linking: Linking

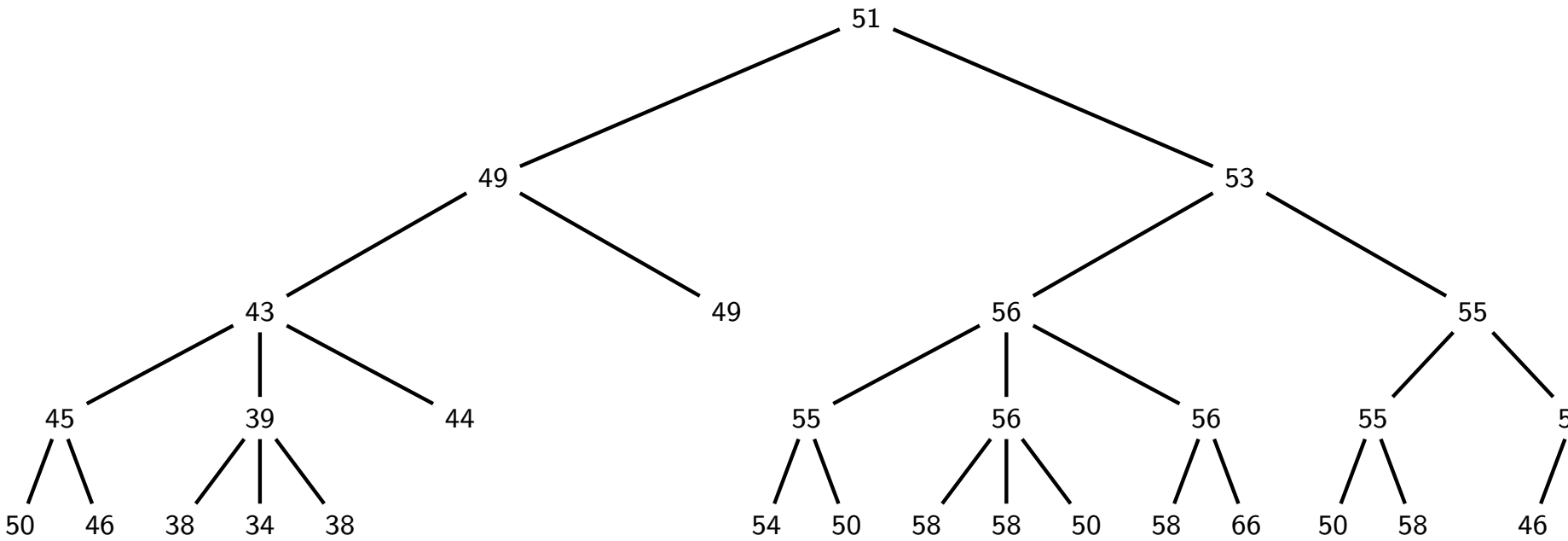


Abbildung: Linking anhand der nächsten Werte

Pyramid Linking: Averaging

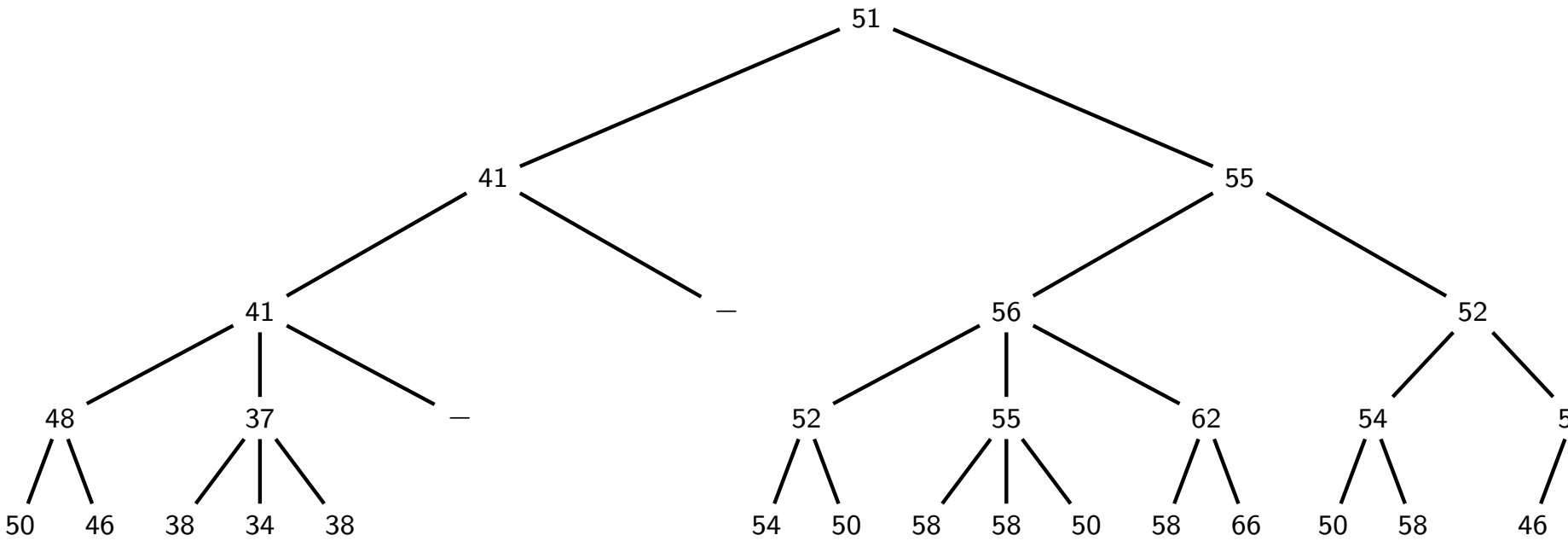


Abbildung: Mittelung in der Baumstruktur

Pyramid Linking: Re-Linking

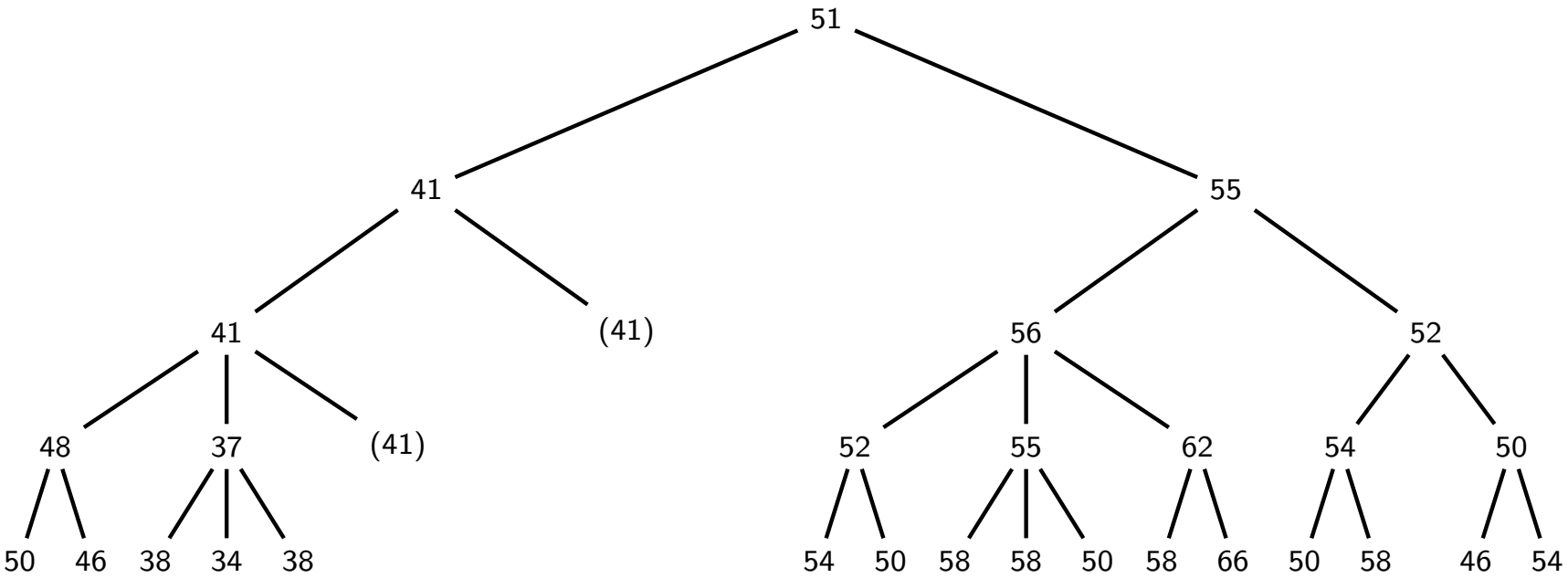


Abbildung: Werte von oben propagieren und Relinking, zweite Phase (keine Änderung)

Pyramid Linking: Zuweisung der Pixelwerte

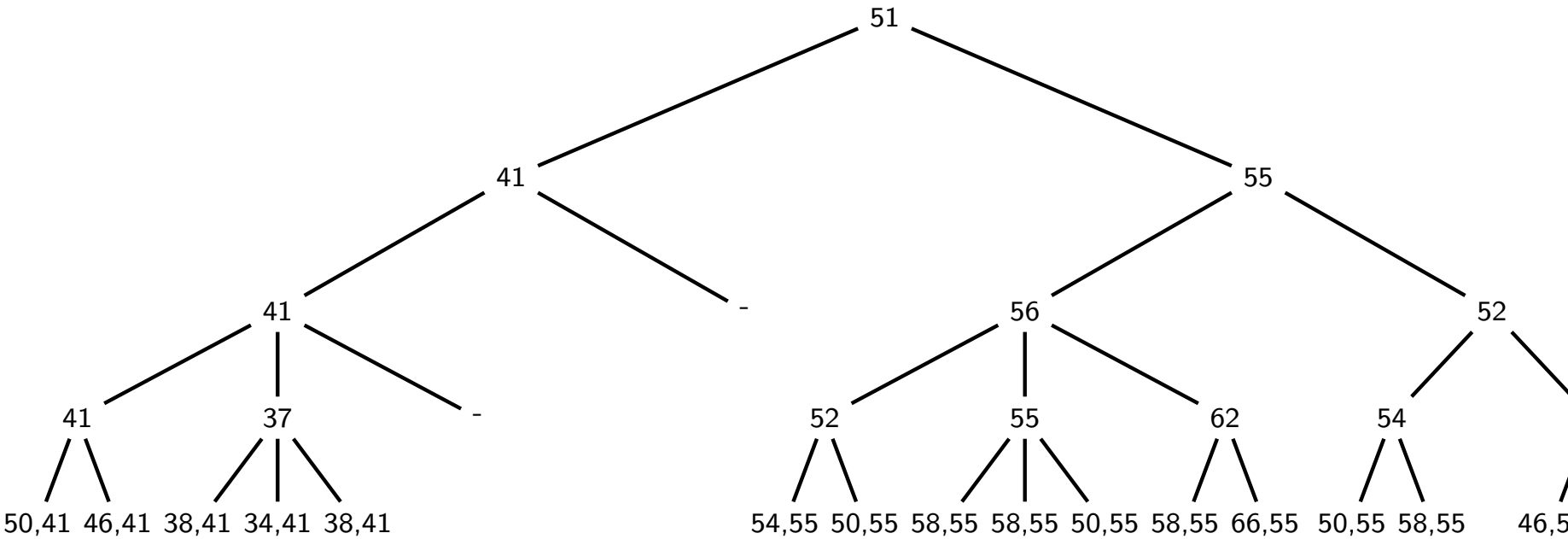


Abbildung: Ende des Algorithmus: Zuweisung der Pixelwerte

Modellbasierte Segmentierung

- Die vorangegangenen Algorithmen machen wenige Annahmen über das Aussehen eines Bildes.
- Hat man jedoch Zusatzinformationen über die vorhandenen Objekte im Bild, so ergeben sich wesentlich bessere Segmentierungen.
- Das menschliche Auge leistet solche Objekterkennungen mit oft verblüffenden Ergebnissen.

Modellbasierte Segmentierung

- Die vorangegangenen Algorithmen machen wenige Annahmen über das Aussehen eines Bildes.
- Hat man jedoch Zusatzinformationen über die vorhandenen Objekte im Bild, so ergeben sich wesentlich bessere Segmentierungen.
- Das menschliche Auge leistet solche Objekterkennungen mit oft verblüffenden Ergebnissen.

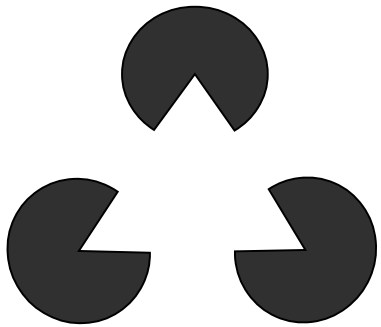


Abbildung: Der Mensch ist in der Lage, Vorwissen schnell mit Bildfragmenten zu kombinieren und so Muster zu erkennen oder Objekte wahrzunehmen die im

(gesprochen [hΛf])
Als Beispiel der modellbasierten Extraktion von Objekten soll hier die Suche nach Geraden in zweidimensionalen Bildern dienen.
Die Idee dahinter ist die Darstellung jeder Geraden über die Geradengleichung

$$y = a_0 + a_1 x \tag{1}$$

mit

- x und y: die Position des Punktes auf der Geraden
- a_0: y-Verschiebung der Geraden
- a_1: Steigung der Geraden

Für einen gegebenen Punkt im Bild kann man nun alle Geraden durch diesen Punkt angeben und erhält durch Umstellung der Gleichung 1

$$a_1 = \frac{y_0}{x_0} - \frac{1}{x_0} a_0 \tag{2}$$

was einer neuen Geradengleichung entspricht, mit

- x_0 und y_0: konkreter Punkt
- $\frac{y_0}{x_0}$: y-Verschiebung
- $-\frac{1}{x_0}$: Steigung der Geraden

Wir bezeichnen den Raum der Punkte (a_0, a_1) als **Merkmalsraum** und können jetzt für jeden Punkt (x_i, y_i), der auf einer Kante im Bild liegt, eine Gerade mit den Lösungen der Gleichung 2 in den Merkmalsraum einzeichnen.
Addiert man hierbei die Intensitäten der Pixel auf, so haben die Pixel eine höhere Intensität, die eine Gerade darstellen, durch die viele Kantenpunkte durchquert.

Wir bezeichnen den Raum der Punkte (a_0, a_1) als **Merkmalsraum** und können jetzt für jeden Punkt (x_i, y_i), der auf einer Kante im Bild liegt, eine Gerade mit den Lösungen der Gleichung 2 in den Merkmalsraum einzeichnen.
Addiert man hierbei die Intensitäten der Pixel auf, so haben die Pixel eine höhere Intensität, die eine Gerade darstellen, durch die viele Kantenpunkte durchquert.
Durch Definition eines Schwellwertes im Merkmalsraum lassen sich nun die dominanten Geraden im Bild bestimmen.
Das Verfahren lässt sich durch Schnitte der Geraden sowie durch Analyse der Punkte entlang einer Geraden im Bildraum auf komplexere geometrische Objekte (Strahlen, Linien, Vielecke, ...) ausdehnen.

Merkmalsraum: Alternative Definition

Manchmal wird der Merkmalsraum auch über die Geradengleichung

$$\mathbf{nx} = d$$

oder

$$x\cos\theta + y\sin\theta = d$$

definiert.

Houghtransformation

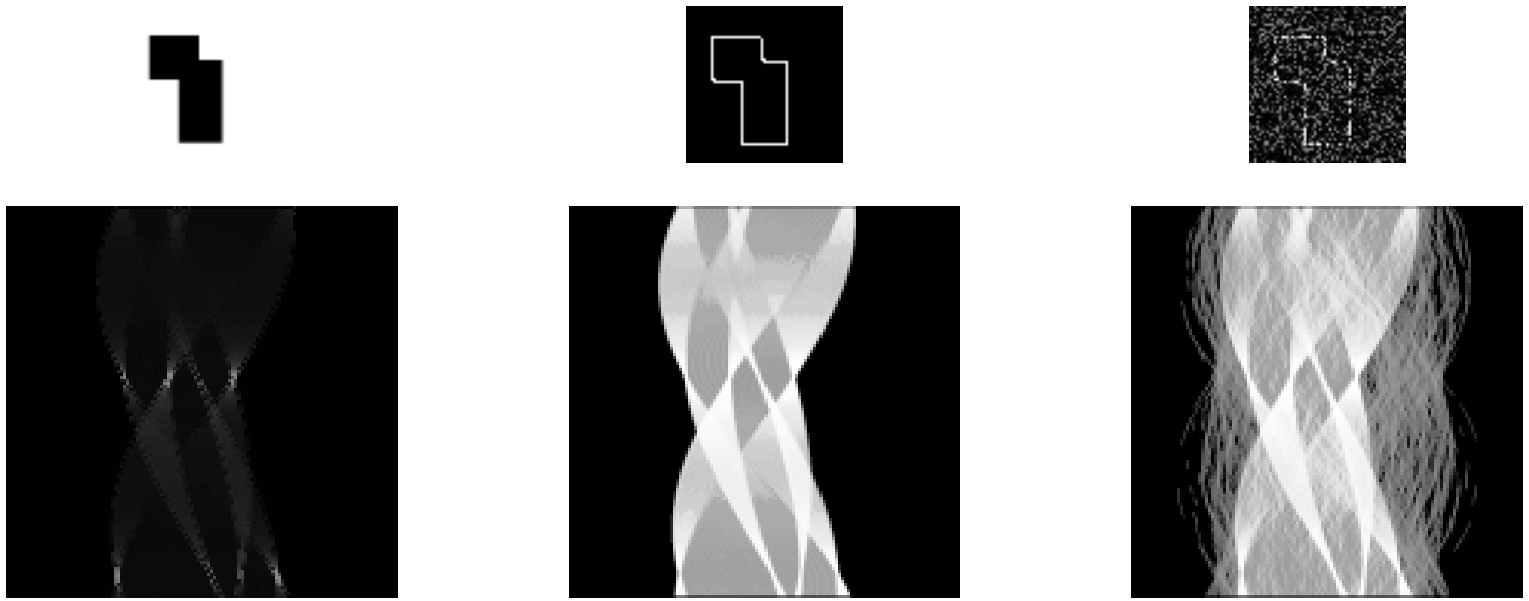


Abbildung: Beispiele; oben: Bild, unten: Houghtranformation. [8]

[8] http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm

Im allgemeinen ist die Houghtransformation rechenintensiv, lässt sich jedoch recht gut parallelisieren und bis zu einem bestimmten Grad optimieren.
Verwendet man Wissen des Strukturtensors, so lässt sich durch das lokale Wissen des Kantenverlaufs das Verfahren beschleunigen.

Globale Optimierungsansätze

Die modellbasierten Ansätze haben den Nachteil, dass das zu segmentierende Objekt bis auf wenige Parameter bekannt sein muss, was oft zu unflexibel ist.
Bei globalen Ansätzen hingegen geht man von grundlegenden Annahmen über Objekte im Bild aus und versucht, ein globales Optimierungsproblem zu lösen.

Vorgeschmack: Bisher wurde jedoch nur darauf geachtet, dass sich Merkmalsbild und Originalbild ähneln und das Merkmalsbild selbst glatt ist. Wichtig für die Extraktion von Objekten ist jedoch, dass Diskontinuitäten, die im Originalbild die Ränder von Objekten beschreiben, erhalten bleiben und sich der Glattheitsterm hauptsächlich auf das innere eines Objektes erstreckt. Hierzu wollen wir, dass f an den Objektgrenzen deutliche Diskontinuitäten aufweist. Ideal wäre hierbei eine Beschränkung der Integration von L auf die Objekte selbst. Dazu würde man aber eine vorherige Segmentierung der Objekte benötigen. Hingegen versucht man den Glattheitsterm so zu modifizieren, dass dieser an vermeintlichen Objekträndern, d.h. an Pixeln/Voxeln, an denen eine Kante gefunden wurde, abgeschwächt wird. Da f Kantenverläufe vorschlägt, kann dies iterativ geschehen.

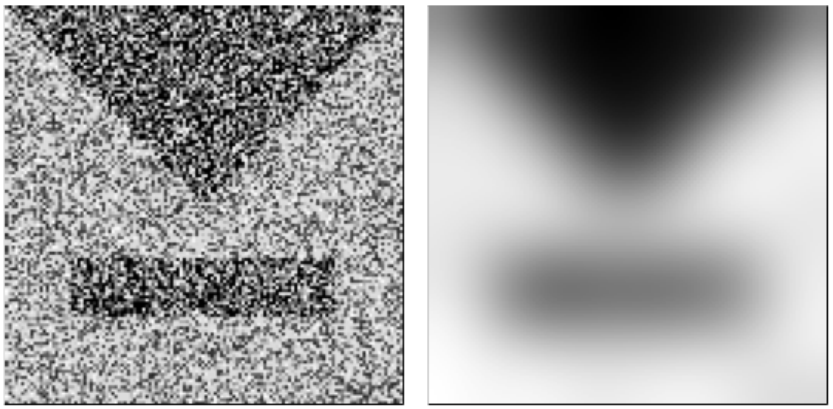


Abbildung: Glättung eines Testbildes mit 128×128 Pixeln. Links Testbild, rechts lineare Diffusion: $t = 80$. [10]

[9] http://www.mia.uni-saarland.de/weickert/Papers/book.pdf Fig. 5.2, S. 117

[10] http://www.mia.uni-saarland.de/weickert/Papers/book.pdf Fig. 5.2, S. 117

Idee ist es nun, Eigenschaften von Objekten zu verlangen, welche die Klasse der zu erkennenden Objekte möglichst nicht einschränkt. Solche Eigenschaften sind in der Regel:

- Innerhalb eines Objektes ändern sich die Werte nur langsam oder gar nicht (Werte sind hier die Grauwerte im Bild oder Eigenschaften des Strukturtensors bei Mustern).
- Die Ränder eines Objektes bestehen vor allem aus glatten Kurven bzw. Kanten und sind geschlossen.
- Die Objektgrenzen liegen nahe an den ermittelten Kanten.

Der allgemeine Variationsansatz hat folgende Gestalt:

- Sei $V \subseteq \mathbb{R}^3$ der Bildraum.
- Sei $g : V \to \mathbb{R}$ ein Merkmalsbild.
- Wir suchen nun ein Modell $f : V \to R$, das folgendes Funktional minimiert

$$\int_V L(f, f_x, x)dx \to \min$$

- Hierbei beschreibt f_x die Ableitungen der Funktion f.

Typische Forderungen lassen sich nun in Gleichungen ausdrücken:

- Ähnlichkeitsbedingung

$$S(f,x) = \|f(x) - g(x)\|_n$$

Typische Forderungen lassen sich nun in Gleichungen ausdrücken:

- Ähnlichkeitsbedingung

$$S(f,x) = \|f(x) - g(x)\|_n$$

- Glattheitsbedingung des Bildes (Regularisierung)

$$R(f_x) = \alpha^2 \cdot |\nabla f|^2$$

Typische Forderungen lassen sich nun in Gleichungen ausdrücken:

- Ähnlichkeitsbedingung

$$S(f,x) = \|f(x) - g(x)\|_n$$

- Glattheitsbedingung des Bildes (Regularisierung)

$$R(f_x) = \alpha^2 \cdot |\nabla f|^2$$

Verwendet man ausschließlich diese beiden Terme, so ergibt sich das Optimierungsproblem

$$\int_V (S(f,x) + R(f_x))\, dx \to \min$$

Berechnung
Die Variationsrechnung ist eine Teildisziplin der Mathematik, zu deren wichtigen Aussagen gehört, dass die Gleichung über die notwendige Bedingung der Euler-Lagrange-Gleichung

$$L_f - \sum_{p=1}^{3} \frac{\partial}{\partial x_p} L_{f_{x_p}} = 0$$

gelöst werden kann.

Berechnung

Die Variationsrechnung ist eine Teildisziplin der Mathematik, zu deren wichtigen Aussagen gehört, dass die Gleichung über die notwendige Bedingung der Euler-Lagrange-Gleichung

$$L_f - \sum_{p=1}^{3} \frac{\partial}{\partial x_p} L_{f_{x_p}} = 0$$

gelöst werden kann.
Dies ist eine Differentialgleichung in f, die mit Hilfe der Numerik in der Regel approximativ gelöst wird.
In der Bildverarbeitung ist die Diskretisierung bereits über das Eingabebild gegeben und die entsprechenden Operatoren finden hier Anwendung.

Diffusionsmodelle

Man kann das Erzeugen eines glatten Modells als Diffusionsprozess verstehen.
Hierbei beschreibt

- der Glattheitsterm die Diffusion
- der Ähnlichkeitsterm die Funktion zusätzlicher Quellen im Bild

Dies führt zu einer aus der Chemie bekannten Diffusions-Reaktions-Gleichung.

Man kann das Erzeugen eines glatten Modells als Diffusionsprozess verstehen.
Hierbei beschreibt

- der Glattheitsterm die Diffusion
- der Ähnlichkeitsterm die Funktion zusätzlicher Quellen im Bild

Dies führt zu einer aus der Chemie bekannten Diffusions-Reaktions-Gleichung.
Es wird das Diffusions-Reaktions-Systems

$$\frac{df}{dt} = \sum_{p=1}^{3} \frac{\partial}{\partial x_p} L_{f_{x_p}} - L_f$$

betrachtet.
Für dieses betrachtet man die Euler-Lagrange-Gleichung

$$\sum_{p=1}^{3} \frac{\partial}{\partial x_p} L_{f_{x_p}} - L_f = 0$$

Letztere kann als Gleichgewichtslösung des Diffusions-Reaktions-Systems betrachtet werden und wird iterativ gelöst.

Um den Glattheitsterm abhängig von Kanten lokal zu modifizieren, kann man die Diffusionskonstante D im Gleichungssystem

$$j = -D\nabla f$$

mit

$$\frac{df}{dt} + \nabla j = 0$$

von $|\nabla f|^2$ abhängig machen.

[11] [Perona, Malik, Scale Space and Edge Detection Using Anisotropic Diffusion, Proceedings of IEEE Computer Society Workshop on Computer Vision, IEEE Computer Society, Washington, USA, 1987, S. 16-20.]

Um den Glattheitsterm abhängig von Kanten lokal zu modifizieren, kann man die Diffusionskonstante D im Gleichungssystem

$$j = -D\nabla f$$

mit

$$\frac{df}{dt} + \nabla j = 0$$

von $|\nabla f|^2$ abhängig machen.
Perona und Malik[11] schlagen hierfür

$$D = D_0 \frac{\lambda^2}{|\nabla f|^2 + \lambda^2}$$

vor, wobei D_0 der Wert außerhalb der Kanten und λ ein freier Parameter ist.

- Ist $|\nabla f|^2$ deutlich kleiner als λ so erhält man $D \to D_0$.
- Dominiert hingegen der Gradient, so erhält man $D \to 0$.

[11] [Perona, Malik, Scale Space and Edge Detection Using Anisotropic Diffusion, Proceedings of IEEE Computer Society Workshop on Computer Vision, IEEE Computer Society, Washington, USA, 1987, S. 16-20.]

Um eine beständige Verstärkung von $|\nabla f|^2$ im Prozess zu umgehen, schlägt Weickert[12]

$$D = 1 - \exp \frac{c_m}{\left(\frac{|\nabla (B^r \star f)(x)|}{\lambda} \right)^m}$$

vor, wobei er unter anderem

- $m = 4$
- $c_m = 3.31488$

setzt.

[12] [Weickert, Anisotropic Diffusion in Image Processing, Dissertation, Universität Kaiserslautern, 1996]
`http://www.mia.uni-saarland.de/weickert/book.html`

Um eine beständige Verstärkung von $|\nabla f|^2$ im Prozess zu umgehen, schlägt Weickert[12]

$$D = 1 - \exp \frac{c_m}{\left(\frac{|\nabla (B^r \star f)(x)|}{\lambda}\right)^m}$$

vor, wobei er unter anderem

- $m = 4$
- $c_m = 3.31488$

setzt.
Dies ergibt

- $D = 1$ für $|\nabla f| = \lambda$
- $D = 0.15$ für $|\nabla f| = 2\lambda$

[12] [Weickert, Anisotropic Diffusion in Image Processing, Dissertation, Universität Kaiserslautern, 1996]
`http://www.mia.uni-saarland.de/weickert/book.html`

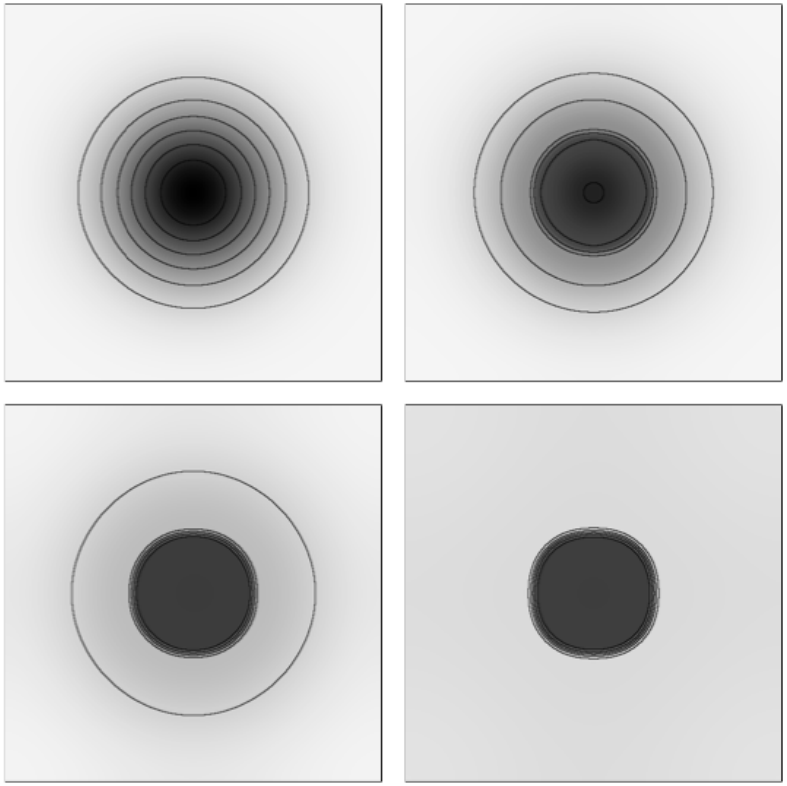


Abbildung: Anisotrope Diffusion einer Gaußartigen Funktion von 256×256 Pixeln. Grauwertdarstellung und Isokonturen. $\lambda = 3,6,\ \sigma = 2,\ t = 0, 125, 625, 3125$.

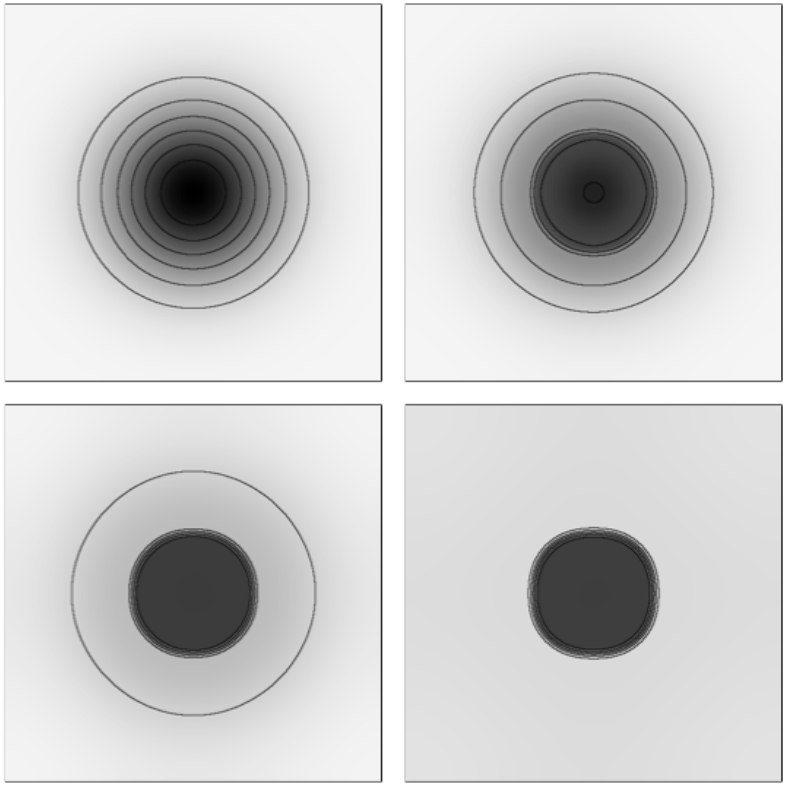
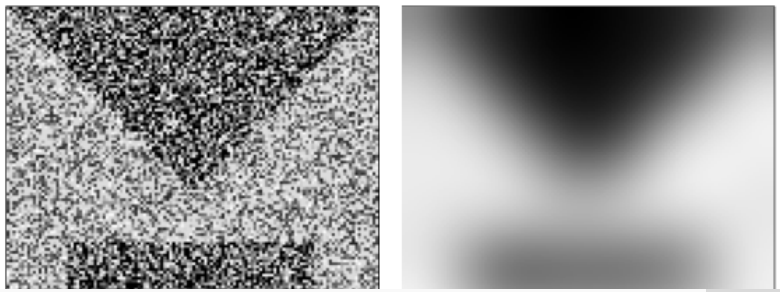


Abbildung: Anisotrope Diffusion einer Gaußartigen Funktion von 256×256 Pixeln. Grauwertdarstellung und Isokonturen. $\lambda = 3,6,\ \sigma = 2,\ t = 0, 125, 625, 3125.$



Anisotrope Diffusion

- Die **inhomogene Diffusion**
  - verhindert das Glätten über die Kanten
  - hinterlässt aber verrauschte Segmente an den Kanten
- Man wünscht sich deshalb eine Glättung entlang der Kanten.
- Dies führt zur Idee der **anisotropen Diffusion**.

Anisotrope Diffusion

Man ersetzt den Diffusionsterm

$$j = -D\nabla f$$

mit dem Skalar D durch den Tensor $\mathbf{D}$ und erhält

$$j = -\mathbf{D}\nabla f = -\begin{pmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{pmatrix} \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}$$

Anisotrope Diffusion

Man ersetzt den Diffusionsterm

$$j = -D\nabla f$$

mit dem Skalar D durch den Tensor $\mathbf{D}$ und erhält

$$j = -\mathbf{D}\nabla f = -\begin{pmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{pmatrix} \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}$$

Das Diffusions-Reaktionssystem ändert sich zu

$$\frac{df}{dt} = \nabla(\mathbf{D}(\nabla f \nabla f^T)\nabla f) - L_f$$

Man ersetzt den Diffusionsterm

$$j = -D\nabla f$$

mit dem Skalar D durch den Tensor $\mathbf{D}$ und erhält

$$j = -\mathbf{D}\nabla f = -\begin{pmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{pmatrix} \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}$$

Das Diffusions-Reaktionssystem ändert sich zu

$$\frac{df}{dt} = \nabla(\mathbf{D}(\nabla f \nabla f^T)\nabla f) - L_f$$

Zum besseren Verständnis diagonalisiert man den Tensor $\mathbf{D}$

$$\begin{aligned} j' &= -\begin{pmatrix} D'_1 & & \\ & D'_2 & \\ & & D'_3 \end{pmatrix} \begin{pmatrix} \frac{\partial f'}{\partial x} \\ \frac{\partial f'}{\partial y} \\ \frac{\partial f'}{\partial z} \end{pmatrix} \\ &= \begin{pmatrix} D'_1 \cdot \frac{\partial f'}{\partial x} \\ D'_2 \cdot \frac{\partial f'}{\partial y} \\ D'_3 \cdot \frac{\partial f'}{\partial z} \end{pmatrix} \end{aligned}$$

Ausgeschrieben ergibt sich nun

$$f(x,t) = \left[\frac{1}{2 \cdot \pi \cdot \sigma_1(t) \cdot \sigma_2(t) \cdot \sigma_3(t)} e^{\left(\frac{-x_1'}{2 \cdot \sigma_1(t)} \right)} e^{\left(\frac{-x_2'}{2 \cdot \sigma_2(t)} \right)} e^{\left(\frac{-x_3'}{2 \cdot \sigma_3(t)} \right)} \right] * f(x,0)$$

also eine dreidimensionale Faltung mit Gaußfiltern unterschiedlicher Standardabweichung σ_i in den Orientierungen x_i.

Nun setzt man

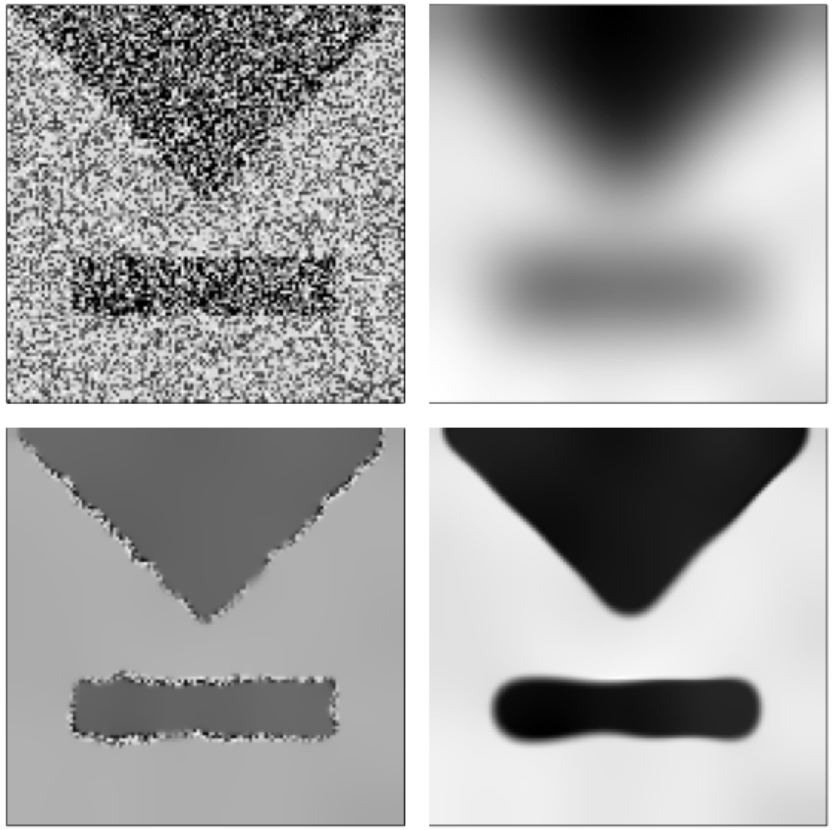
$$D_1' = 1 - \exp\left(\frac{c_m}{\left(\frac{|\nabla (B^r * f)(x)|}{\lambda} \right)^m} \right)$$

um eine Glättung senkrecht zum Rand zu verhindern und

$$\begin{aligned} D_2' &= 1 \\ D_3' &= 1 \end{aligned}$$

um sie entlang des Randes zu erlauben.

Anisotrope Diffusion



Teil IX

Morphologische Operationen

Bildverarbeitungsprozess

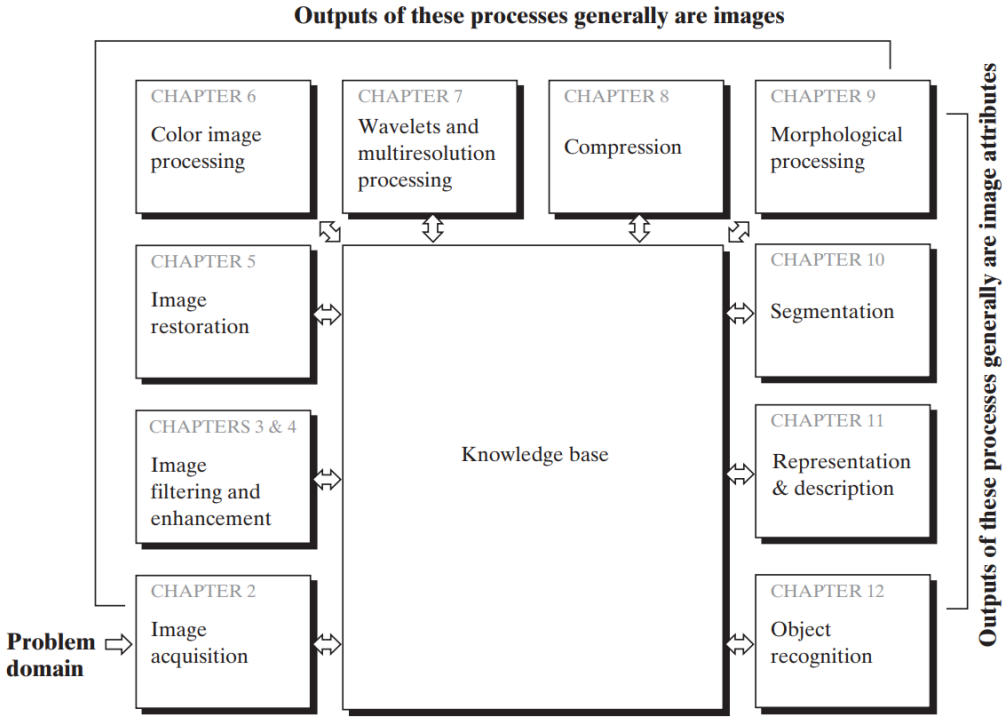


Abbildung: Die Bildverarbeitung umfasst eine reihe von Algorithmen, die entweder wiederum Bilder oder Bildattribute ausgeben. [Digital Image Processing, 3rd Edition, Gonzales and Woods]

Diese Vorlesung: Morphologische Operationen

Morphologische Operationen

Die Morphologie (die Lehre von der Form von Objekten) verwendet innerhalb der Bildverarbeitung binäre Nachbarschaftsoperatoren auf segmentierten Bildern, um die Gestalt von Objekten zu analysieren.

Sei $g_{m,n}$ das binäre, segmentierte Bild. Sei $M_{m,n}$ die Operatormaske.
Dilatation

- Wikipedia: (von lat. dilatare) verlängern, ausdehnen, vergrößern
- Duden: (von spätlateinisch dilatatio) Erweiterung
- dehnt Objekte aus
- füllt Löcher

$$g_{m',n'} = \vee_{m',n' \neq 0} M_{m',n'} \wedge g_{m+m',n+n'}$$

Sei $g_{m,n}$ das binäre, segmentierte Bild. Sei $M_{m,n}$ die Operatormaske.

Dilatation

- Wikipedia: (von lat. dilatare) verlängern, ausdehnen, vergrößern
- Duden: (von spätlateinisch dilatatio) Erweiterung
- dehnt Objekte aus
- füllt Löcher

$$g_{m',n'} = \vee_{m',n' \neq 0} M_{m',n'} \wedge g_{m+m',n+n'}$$

Erosion

- Wikipedia: (von lat. erodere) abtragen
- Duden: (lat. erosio) das Zerfressenwerden
- verkleinert Objekte
- entfernt kleine Objekte

$$g_{m',n'} = \wedge_{m',n' \neq 0} M_{m',n'} \wedge g_{m+m',n+n'}$$

Beispiel

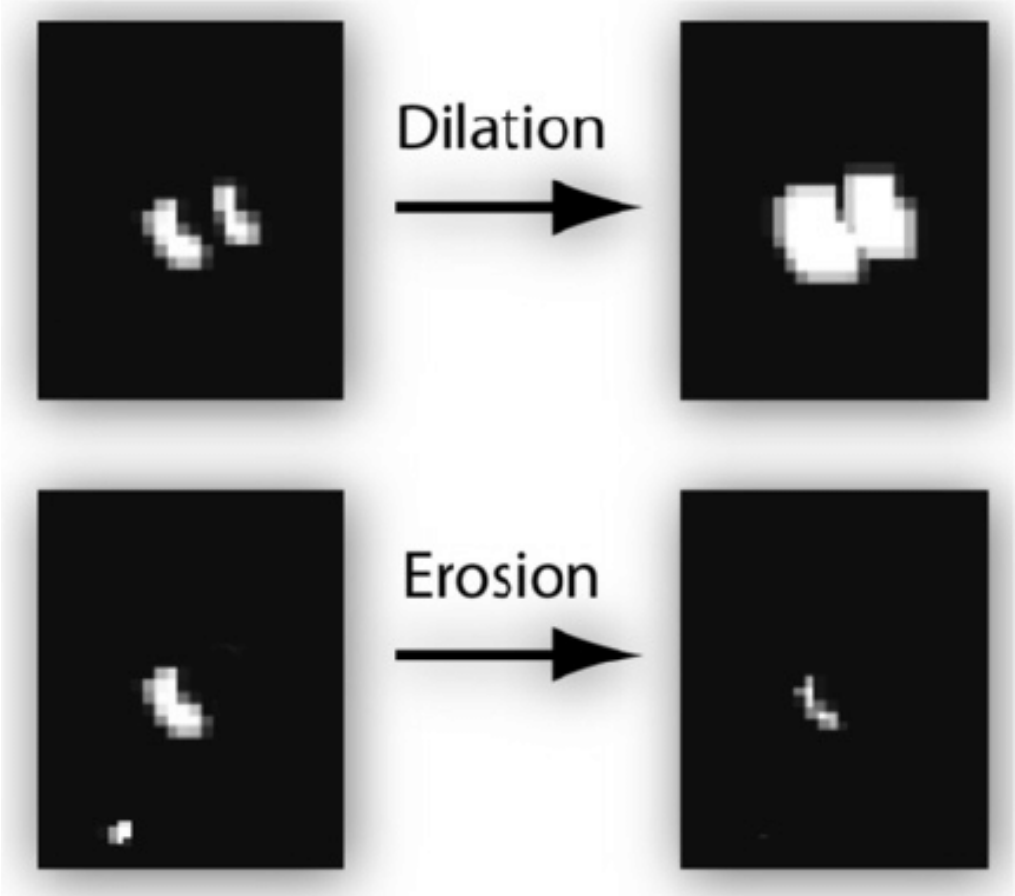


Abbildung: Dilatation und Erosion

Beispiel

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

Abbildung: Dilatation

Beispiel

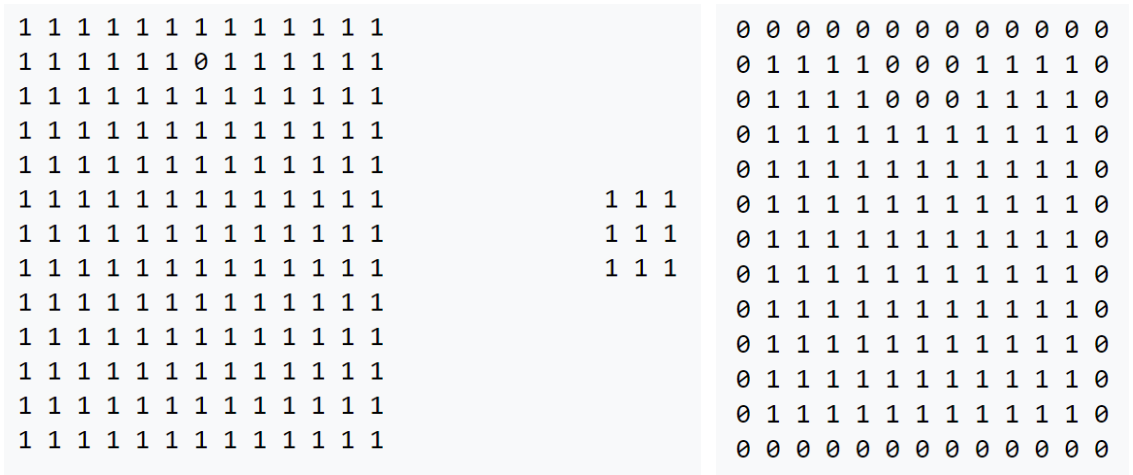


Abbildung: Erosion

Man kann die Operationen auch mengentheoretisch auffassen.

- Sei G die Menge der Pixel im Bild ungleich 0
- Sei M die Menge der Pixel in der Maske ungleich 0
- Sei M_p die auf Pixel p verschobene Maske

Man kann die Operationen auch mengentheoretisch auffassen.

- Sei G die Menge der Pixel im Bild ungleich 0
- Sei M die Menge der Pixel in der Maske ungleich 0
- Sei M_p die auf Pixel p verschobene Maske

Dann gilt

- Dilatation

$$G \oplus M := \{p : M_p \cap G \neq 0\}$$

- Erosion:

$$G \ominus M := \{p : M_p \subseteq G\}$$

Nun lassen sich die Eigenschaften der Operationen studieren.

Verschiebungsinvarianz

- Dilatation und Erosion sind über binäre Faltungsmasken definiert.
- Somit sind beide Operatoren verschiebungsinvariant.
- Mit dem Verschiebungsoperator S_{mn} gilt deshalb

$$M(S_{mn}G) = S_{mn}(MG)$$

Es gilt bezüglich der **Kommutativität**

$$M_1 \oplus M_2 = M_2 \oplus M_1$$

aber

$$M_1 \ominus M_2 \neq M_2 \ominus M_1$$

da für $M_2 \subset M_1$

- die Erosion $M_1 \ominus M_2$ immer zur leeren Menge führt
- die Erosion $M_2 \ominus M_1$ nicht zwangsläufig zur leeren Menge führt

Es gilt bezüglich der **Kommutativität**

$$M_1 \oplus M_2 = M_2 \oplus M_1$$

aber

$$M_1 \ominus M_2 \neq M_2 \ominus M_1$$

da für $M_2 \subset M_1$

- die Erosion $M_1 \ominus M_2$ immer zur leeren Menge führt
- die Erosion $M_2 \ominus M_1$ nicht zwangsläufig zur leeren Menge führt

Es gilt jedoch in beiden Fällen, dass die Reihenfolge der Anwendung der Masken vertauscht werden kann:

$$\begin{aligned}
(G \oplus M_1) \oplus M_2 &= G \oplus (M_1 \oplus M_2) \\
&= (G \oplus M_2) \oplus M_1 \\
(G \ominus M_1) \ominus M_2 &= G \ominus (M_1 \oplus M_2) \\
&= (G \ominus M_2) \ominus M_1
\end{aligned}$$

Im Sinne von $G_1 \subseteq G_2$ gilt

$$G_1 \subseteq G_2 \quad \rightarrow \quad G_1 \oplus M \subseteq G_2 \oplus M$$
$$G_1 \subseteq G_2 \quad \rightarrow \quad G_1 \ominus M \subseteq G_2 \ominus M$$

Die Teilmengenrelationen sind also invariant bezüglich der Dilatation und der Erosion.

Im Sinne von $G_1 \subseteq G_2$ gilt

$$\begin{aligned} G_1 \subseteq G_2 \quad &\rightarrow \quad G_1 \oplus M \subseteq G_2 \oplus M \\ G_1 \subseteq G_2 \quad &\rightarrow \quad G_1 \ominus M \subseteq G_2 \ominus M \end{aligned}$$

Die Teilmengenrelationen sind also invariant bezüglich der Dilatation und der Erosion.
Bezüglich der Mengenoperatoren $\cap, \cup$ verhalten sich Dilatation und Erosion nur eingeschränkt distributiv. Es gilt

$$\begin{aligned} (G_1 \cap G_2) \oplus M \quad &\subseteq \quad (G_1 \oplus M) \cap (G_2 \oplus M) \\ (G_1 \cap G_2) \ominus M \quad &\subseteq \quad (G_1 \ominus M) \cap (G_2 \ominus M) \end{aligned}$$

sowie

$$\begin{aligned} (G_1 \cup G_2) \oplus M \quad &= \quad (G_1 \oplus M) \cup (G_2 \oplus M) \\ (G_1 \cup G_2) \ominus M \quad &\supseteq \quad (G_1 \ominus M) \cup (G_2 \ominus M) \end{aligned}$$

Dilatation und Erosion sind im Sinne von

$$\begin{aligned} \overline{G} \ominus M &= \overline{G \oplus M} \\ \overline{G} \oplus M &= \overline{G \ominus M} \end{aligned}$$

dual zueinander.

Dilatation und Erosion sind im Sinne von

$$\begin{aligned} \overline{G} \ominus M &= \overline{G \oplus M} \\ \overline{G} \oplus M &= \overline{G \ominus M} \end{aligned}$$

dual zueinander.
Das heißt z.B., dass eine Erosion auf dem Hintergrund über eine Dilatation des Vordergrundes berechnet werden kann.

Opening und Closing

- Bei der Erosion werden Objekte, die kleiner als die Maske sind, entfernt, aber alle anderen Objekte auch verkleinert.
- Durch anschließende Dilatation lässt sich dies korrigieren.
- Die Kombination von Erosion mit anschließender Dilatation wird auch Öffnen (**opening**) genannt.

$$G \circ M = (G \ominus M) \oplus M$$

Opening und Closing

- Bei der Erosion werden Objekte, die kleiner als die Maske sind, entfernt, aber alle anderen Objekte auch verkleinert.
- Durch anschließende Dilatation lässt sich dies korrigieren.
- Die Kombination von Erosion mit anschließender Dilatation wird auch Öffnen (**opening**) genannt.

$$G \circ M = (G \ominus M) \oplus M$$

- Im Gegensatz dazu schließt Dilatation kleine Löcher und Risse, wobei zugleich alle Objekte größer werden.
- Dies lässt sich durch eine nachfolgende Erosion korrigieren.
- Diese Kombination führt auf das Schließen (**closing**) von Objekten.

$$G \cdot M = (G \oplus M) \ominus M$$

Opening und Closing

Sowohl Opening als auch Closing sind **idempotent**, also gilt

$$\begin{aligned} (G \circ M) \circ M &= G \circ M \\ (G \cdot M) \cdot M &= G \cdot M \end{aligned}$$

Das heißt, dass eine mehrfache Anwendung des gleichen Operators das Bild nur so verändert, wie es die einfache Anwendung getan hätte.

Beispiel

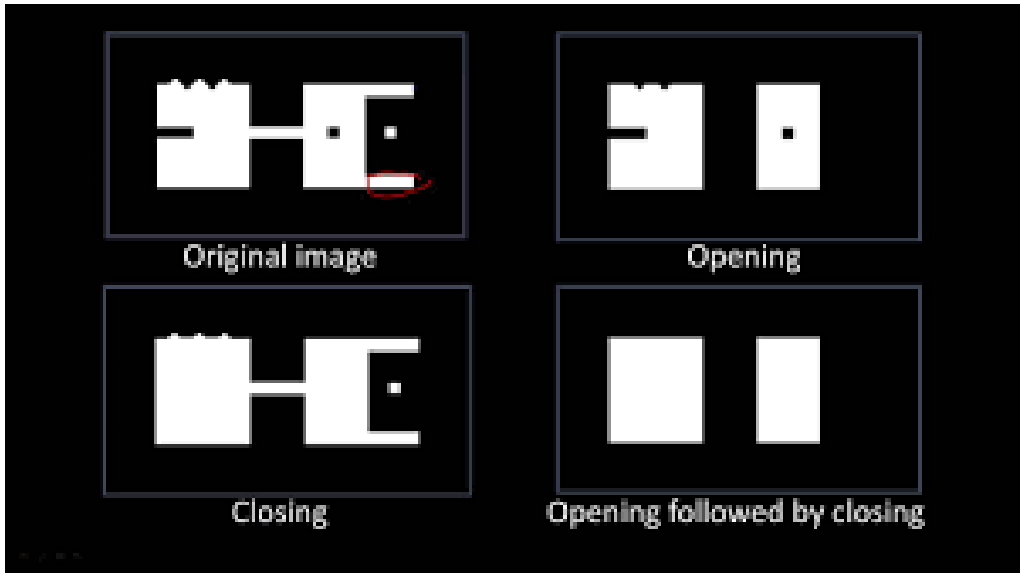


Abbildung: Opening und Closing

Hit-Miss-Operatoren

- Morphologische Operationen lassen sich noch vielseitiger verwenden um Objekte erkennen zu können.
- Dazu definiert man den **Hit-Miss-Operator**.

Hit-Miss-Operatoren

- Morphologische Operationen lassen sich noch vielseitiger verwenden um Objekte erkennen zu können.
- Dazu definiert man den **Hit-Miss-Operator**.

1. Zuerst nutzt man die Erosion mit der 1×3 Maske

$$M_1 = [1\ 1\ 1]$$

Diese entfernt alle kleineren Objekte.

Hit-Miss-Operatoren

- Morphologische Operationen lassen sich noch vielseitiger verwenden um Objekte erkennen zu können.
- Dazu definiert man den **Hit-Miss-Operator**.

1. Zuerst nutzt man die Erosion mit der 1×3 Maske

$$M_1 = [1\ 1\ 1]$$

Diese entfernt alle kleineren Objekte.

2. Um größere Objekte zu entfernen wählt man die inverse Maske

$$M_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & & & & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

und erodiert auf dem inversen Ausgangsbild.

Der Schnitt beider Bilder markiert die Zentren der gesuchten Objekte.

Der Schnitt beider Bilder markiert die Zentren der gesuchten Objekte.
Man definiert also den **Hit-Miss-Operator** für $M_1 \cap M_2 = \emptyset$. als

$$\begin{aligned} G \otimes (M_1, M_2) &= (G \ominus M_1) \cap (\overline{G} \ominus M_2) \\ &= (G \ominus M_1) \cap (\overline{G \oplus M_2}) \end{aligned}$$

Der Schnitt beider Bilder markiert die Zentren der gesuchten Objekte.
Man definiert also den **Hit-Miss-Operator** für $M_1 \cap M_2 = \emptyset$. als

$$\begin{aligned} G \otimes (M_1, M_2) &= (G \ominus M_1) \cap (\overline{G} \ominus M_2) \\ &= (G \ominus M_1) \cap (\overline{G \oplus M_2}) \end{aligned}$$

Falls $M_1 \cap M_2 \neq \emptyset$, so folgt

$$G \otimes (M_1, M_2) = \emptyset$$

Flexible Operatoren

Wählt man M_2 nicht genau als Negativmaske zu M_1, so ergeben sich flexible Operatoren.

Flexible Operatoren

Wählt man M_2 nicht genau als Negativmaske zu M_1, so ergeben sich flexible Operatoren.

$$M_1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & & & & & & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

finden zum Beispiel alle Objekte der Breite 3 bis 5.

Wählt man M_2 nicht genau als Negativmaske zu M_1, so ergeben sich flexible Operatoren.

$$M_1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & & & & & & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

finden zum Beispiel alle Objekte der Breite 3 bis 5.
Man vereinfacht die Schreibweise zu

$$M_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x & 1 & 1 & 1 & x & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In dieser Schreibweise können isolierte Punkte über

$$M_I = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

gefunden werden.

In dieser Schreibweise können isolierte Punkte über

$$M_I = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

gefunden werden.
Untere rechte Ecken:

$$M_{\mathrm{RU}} = \begin{bmatrix} x & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Ausdünnen

- Häufig möchte man Objekte so ausdünnen, dass nur noch deren Topologie übrig bleibt.
- Topologie beschreibt hierbei den Zusammenhang der Objektteile, nicht aber deren Ausdehnung.
- Die Erosion „zerfrisst“ Objekte langsam, garantiert aber keinen Erhalt der Topologie (sie entfernt zum Beispiel kleine Objekte).

Ausdünnen

- Häufig möchte man Objekte so ausdünnen, dass nur noch deren Topologie übrig bleibt.
- Topologie beschreibt hierbei den Zusammenhang der Objektteile, nicht aber deren Ausdehnung.
- Die Erosion „zerfrisst“ Objekte langsam, garantiert aber keinen Erhalt der Topologie (sie entfernt zum Beispiel kleine Objekte).

Topologische Randbedingungen:

- Ein Objekt darf nicht geteilt werden. (Zudem dürfen Verbindungen nicht aufgebrochen werden)
- Ein Objekt darf nicht verschwinden.
- Ein Endpunkt darf nicht entfernt werden. (Zum Erhalt der Lage von „Armen“)

Ein Satz verwendeter Masken für die 4er-Nachbarschaft ist:

$$\begin{bmatrix} 0 & x & 1 \\ 0 & 1 & 1 \\ 0 & x & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & x \\ 0 & 1 & 1 \\ x & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ x & 1 & x \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & x \end{bmatrix}$$

$$\begin{bmatrix} 1 & x & 0 \\ 1 & 1 & 0 \\ 1 & x & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & x \\ 1 & 1 & 0 \\ x & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ x & 1 & x \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & x \end{bmatrix}$$

[13] http://www.imagemagick.org/Usage/morphology/

Ein Satz verwendeter Masken für die 4er-Nachbarschaft ist:

$$\begin{bmatrix} 0 & x & 1 \\ 0 & 1 & 1 \\ 0 & x & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & x \\ 0 & 1 & 1 \\ x & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ x & 1 & x \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & x \end{bmatrix}$$

$$\begin{bmatrix} 1 & x & 0 \\ 1 & 1 & 0 \\ 1 & x & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & x \\ 1 & 1 & 0 \\ x & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ x & 1 & x \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & x \end{bmatrix}$$

Algorithmus

- Die Masken werden nacheinander auf das ganze Bild angewandt und das Ergebnis für den nächsten Schritt gespeichert.
- Der Durchlauf wird wiederholt, bis sich das Bild nicht mehr ändert.[13]

[13] `http://www.imagemagick.org/Usage/morphology/`

Ränder von Objekten kann man entsprechend der Nachbarschaften mit den Masken

$$M_{b4} = \begin{bmatrix} & 1 & \\ 1 & 1 & 1 \\ & 1 & \end{bmatrix} \quad M_{b8} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

finden.
Jedem Randpixel fehlt ein Nachbarpixel, somit wird das entsprechende Pixel durch die Masken gelöscht.
Das invertierte Bild ergibt den Rand des Objektes.

Ränder von Objekten kann man entsprechend der Nachbarschaften mit den Masken

$$M_{b4} = \begin{bmatrix} & 1 & \\ 1 & 1 & 1 \\ & 1 & \end{bmatrix} \quad M_{b8} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

finden.
Jedem Randpixel fehlt ein Nachbarpixel, somit wird das entsprechende Pixel durch die Masken gelöscht.
Das invertierte Bild ergibt den Rand des Objektes.
Die Mengendifferenz

$$\begin{aligned} \partial G &= G - (G \ominus M_b) \\ &= G \cup (\overline{G \ominus M_b}) \\ &= G \cap (\overline{G} \oplus M_b) \end{aligned}$$

liefert also die Randpixel.

Ränder von Objekten kann man entsprechend der Nachbarschaften mit den Masken

$$M_{b4} = \begin{bmatrix} & 1 & \\ 1 & 1 & 1 \\ & 1 & \end{bmatrix} \quad M_{b8} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

finden.
Jedem Randpixel fehlt ein Nachbarpixel, somit wird das entsprechende Pixel durch die Masken gelöscht.
Das invertierte Bild ergibt den Rand des Objektes.
Die Mengendifferenz

$$\begin{aligned} \partial G &= G - (G \ominus M_b) \\ &= G \cup (\overline{G \ominus M_b}) \\ &= G \cap (\overline{G} \oplus M_b) \end{aligned}$$

liefert also die Randpixel.
Alternativ erhält man den Rand des Hintergrunds durch Dilatation des Objektes und anschließendes Entfernen des Objektes

$$\partial G_B = (G \oplus M_b) - G$$

Die Distanztransformation weist jedem Pixel seinen Abstand zum Rand zu.
Dies kann durch (n-1)-fache Erosion erfolgen: [14]

$$D = (\cup_{n=1}^{\infty}(G \ominus M_b^{n-1})/(G \ominus M_b^n)) \cdot n$$

[14] [Jähne, Haußecker (Hrsg.) Computer Vision and Applications. A Guide for Students and Practioners, Academic Press, San Diego, 2000, Kap. 1]
[P. Soille, Morphologische Bildverarbeitung, Grundlagen, Methoden, Anwendungen, Springer, Berlin, 1998]
[Abmayr, Einführung in die digitale Bildverarbeitung, Teubner, Stuttgart, 1994., Kap. 4]

Die Distanztransformation weist jedem Pixel seinen Abstand zum Rand zu.
Dies kann durch (n-1)-fache Erosion erfolgen: [14]

$$D = (\cup_{n=1}^{\infty}(G \ominus M_b^{n-1})/(G \ominus M_b^n)) \cdot n$$

Das Ergebnisbild gibt an jeder Stelle an, wie groß die größte Kugel ist, die man an diese Stelle in das Bild legen kann, sodass sie den Rand nicht schneidet.
Hierbei ist die Definition von „Kugel“ im Bezug zum Abstandsmaß der Nachbarschaft zu sehen ist.

[14] [Jähne, Haußecker (Hrsg.) Computer Vision and Applications. A Guide for Students and Practioners, Academic Press, San Diego, 2000, Kap. 1]
[P. Soille, Morphologische Bildverarbeitung, Grundlagen, Methoden, Anwendungen, Springer, Berlin, 1998]
[Abmayr, Einführung in die digitale Bildverarbeitung, Teubner, Stuttgart, 1994., Kap. 4]

Teil X

Formrepräsentation

Bildverarbeitungsprozess

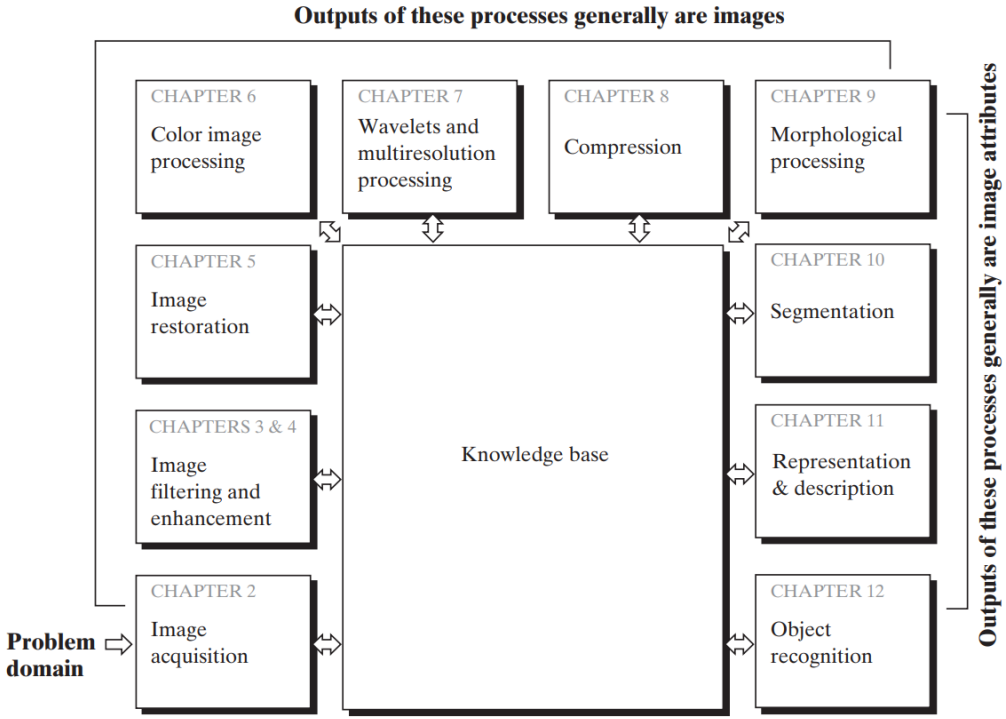


Abbildung: Die Bildverarbeitung umfasst eine reihe von Algorithmen, die entweder wiederum Bilder oder Bildattribute ausgeben. [Digital Image Processing, 3rd Edition, Gonzales and Woods]

Diese Vorlesung: Formrepräsentation

Um Objekte mit vorgegebenen Modellen sowie mit anderen Objekten vergleichen zu können, müssen geeignete Beschreibungen von Objekten erzeugt werden.
Die **Form** des Randes eines Objektes ist hierfür eine wichtige Möglichkeit, die bei binären Bildern zur vollständigen Beschreibung auch ausreicht.
In der Regel sucht man für die Beschreibung nach möglichst invarianten Beschreibungen bezüglich der Basisoperationen wie Translation, Rotation, Skalierung oder gegebenenfalls auch affinen oder projektiven Abbildungen.

Um Objekte mit vorgegebenen Modellen sowie mit anderen Objekten vergleichen zu können, müssen geeignete Beschreibungen von Objekten erzeugt werden.
Die **Form** des Randes eines Objektes ist hierfür eine wichtige Möglichkeit, die bei binären Bildern zur vollständigen Beschreibung auch ausreicht.
In der Regel sucht man für die Beschreibung nach möglichst invarianten Beschreibungen bezüglich der Basisoperationen wie Translation, Rotation, Skalierung oder gegebenenfalls auch affinen oder projektiven Abbildungen.
Statistische Maße wurden bereits zur Erkennung von Texturen verwendet. Integriert man über das ganze Objekt, so ergeben sich Formparameter der Form

$$\mu_{p_0,p_1} = \int (x_0 - \bar{x}_0)^{p_0} (x_1 - \bar{x}_1)^{p_1} g(x) dx$$

mit den Koordinaten des **Schwerpunkts**

$$\bar{x} = \frac{\int x \cdot g(x) dx}{\int g(x) dx}$$

- Die Werte μ_{p_0,p_1} heißen **zentrale Momente**

Momente

- Die Werte μ_{p_0,p_1} heißen **zentrale Momente**
- $\mu_{0,0} = \int g(x)dx$ ist die **Fläche** oder **Masse** des Objekts

- Die Werte μ_{p_0,p_1} heißen **zentrale Momente**
- $\mu_{0,0} = \int g(x)dx$ ist die **Fläche** oder **Masse** des Objekts
- Da $\bar{x}$ genau den Schwerpunkt des Objekts beschreibt, gilt immer

$$\mu_{1,0} = \mu_{0,1} = 0$$

- Die Werte μ_{p_0,p_1} heißen **zentrale Momente**
- $\mu_{0,0} = \int g(x)dx$ ist die **Fläche** oder **Masse** des Objekts
- Da $\bar{x}$ genau den Schwerpunkt des Objekts beschreibt, gilt immer

$$\mu_{1,0} = \mu_{0,1} = 0$$

- μ_{p_0,p_1} ist nicht skaleninvariant. Für eine skalierte Version des Bildes

$$g'(x) = g(x/\alpha)$$

gilt

$$\mu'_{p_0,p_1} = \alpha^{p_0+p_1+2}\mu_{p_0,p_1}$$

Daher definiert man die **skaleninvarianten Momente**

$$\bar{\mu}_{p_0,p_1} = \frac{\mu_{p_0,p_1}}{\left((\mu_{0,0})^{\frac{p_0+p_1+2}{2}}\right)}$$

Für die normierten Werte gilt nun

$$\bar{\mu}_{0,0} = 1$$

$$\bar{\mu}_{0,1} = \bar{\mu}_{1,0} = 0$$

Die ersten Formparameter sind die Werte $\bar{\mu}_{2,0}, \bar{\mu}_{1,1}, \bar{\mu}_{0,2}$, die in der Physik als Trägheitstensor

$$J = \begin{pmatrix} \bar{\mu}_{2,0} & -\bar{\mu}_{1,1} \\ -\bar{\mu}_{1,1} & \bar{\mu}_{0,2} \end{pmatrix}$$

bekannt sind.

Die ersten Formparameter sind die Werte $\bar{\mu}_{2,0}, \bar{\mu}_{1,1}, \bar{\mu}_{0,2}$, die in der Physik als Trägheitstensor

$$J = \begin{pmatrix} \bar{\mu}_{2,0} & -\bar{\mu}_{1,1} \\ -\bar{\mu}_{1,1} & \bar{\mu}_{0,2} \end{pmatrix}$$

bekannt sind.
Dieser Tensor ist symmetrisch. Aus ihm können weitere Formparameter abgeleitet werden, wie zum Beispiel der Winkel der Achse der größten Ausdehnung zur x-Achse des Koordinatensystems

$$\varphi = \frac{1}{2} \arctan \frac{2\bar{\mu}_{1,1}}{\bar{\mu}_{2,0} - \bar{\mu}_{0,2}}$$

Formparameter: elliptische Exzentrizität

Ein weiterer Formparameter ist die **Exzentrität der Ellipse**, die das Objekt über die Parameter von J annähert

$$\epsilon = \frac{(\mu_{2,0} - \mu_{0,2})^2 - 4\mu_{1,1}}{(\mu_{2,0} - \mu_{0,2})^2}$$

mit $\epsilon \in [0, 1]$.
Allgemein gilt: Je mehr höhere Momente in die Beschreibung mit eingehen, desto vollständiger wird die Formbeschreibung des Objekts.

Richtungsketten

Der Rand eines zweidimensionalen Bildes lässt sich effektiv über **Richtungsketten** speichern:

- Man beginnt mit einem möglichst eindeutig beschreibbaren Punkt. Z.B., der am weitesten links liegende Punkt der obersten Zeile des Bildes, in dem das Objekt sichtbar ist.

Richtungsketten

Der Rand eines zweidimensionalen Bildes lässt sich effektiv über **Richtungsketten** speichern:

- Man beginnt mit einem möglichst eindeutig beschreibbaren Punkt. Z.B., der am weitesten links liegende Punkt der obersten Zeile des Bildes, in dem das Objekt sichtbar ist.
- Anschließend wird ausschließlich der relative Verlauf des Randes über die Nachbarschaft gespeichert. Hierzu benötigt man zwei Bit für die Vierer- und drei Bit für die Achter-Nachbarschaft.

Richtungsketten

Der Rand eines zweidimensionalen Bildes lässt sich effektiv über **Richtungsketten** speichern:

- Man beginnt mit einem möglichst eindeutig beschreibbaren Punkt. Z.B., der am weitesten links liegende Punkt der obersten Zeile des Bildes, in dem das Objekt sichtbar ist.
- Anschließend wird ausschließlich der relative Verlauf des Randes über die Nachbarschaft gespeichert. Hierzu benötigt man zwei Bit für die Vierer- und drei Bit für die Achter-Nachbarschaft.

Eigenschaften

- Richtungsketten sind translationsinvariant.

Richtungsketten

Der Rand eines zweidimensionalen Bildes lässt sich effektiv über **Richtungsketten** speichern:

- Man beginnt mit einem möglichst eindeutig beschreibbaren Punkt. Z.B., der am weitesten links liegende Punkt der obersten Zeile des Bildes, in dem das Objekt sichtbar ist.
- Anschließend wird ausschließlich der relative Verlauf des Randes über die Nachbarschaft gespeichert. Hierzu benötigt man zwei Bit für die Vierer- und drei Bit für die Achter-Nachbarschaft.

Eigenschaften

- Richtungsketten sind translationsinvariant.
- Richtungsketten ändern sich bei
  - Rotation
  - Skalierung

Richtungsketten

Der Rand eines zweidimensionalen Bildes lässt sich effektiv über **Richtungsketten** speichern:

- Man beginnt mit einem möglichst eindeutig beschreibbaren Punkt. Z.B., der am weitesten links liegende Punkt der obersten Zeile des Bildes, in dem das Objekt sichtbar ist.
- Anschließend wird ausschließlich der relative Verlauf des Randes über die Nachbarschaft gespeichert. Hierzu benötigt man zwei Bit für die Vierer- und drei Bit für die Achter-Nachbarschaft.

Eigenschaften

- Richtungsketten sind translationsinvariant.
- Richtungsketten ändern sich bei
  - Rotation
  - Skalierung

Bemerkungen

- Ist bekannt, dass Objekte Löcher haben können, so muss zusätzlicher Aufwand betrieben werden, um diese zu speichern.

Richtungsketten

Der Rand eines zweidimensionalen Bildes lässt sich effektiv über **Richtungsketten** speichern:

- Man beginnt mit einem möglichst eindeutig beschreibbaren Punkt. Z.B., der am weitesten links liegende Punkt der obersten Zeile des Bildes, in dem das Objekt sichtbar ist.
- Anschließend wird ausschließlich der relative Verlauf des Randes über die Nachbarschaft gespeichert. Hierzu benötigt man zwei Bit für die Vierer- und drei Bit für die Achter-Nachbarschaft.

Eigenschaften

- Richtungsketten sind translationsinvariant.
- Richtungsketten ändern sich bei
  - Rotation
  - Skalierung

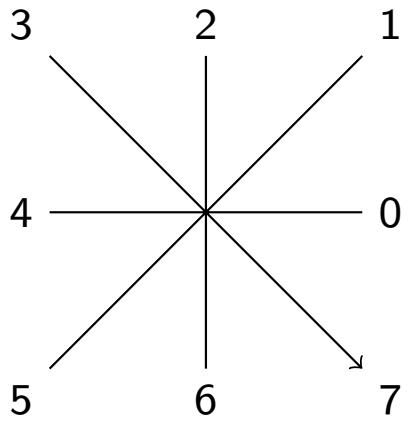
Bemerkungen

- Ist bekannt, dass Objekte Löcher haben können, so muss zusätzlicher Aufwand betrieben werden, um diese zu speichern.
- Für dreidimensionale Objekte bietet sich eine scheibenweise Abspeicherung des Randes an.

Richtungsketten: Beispiele

Ein Pfad kann dann über eine Folge codiert werden.
8er-Nachbarschaft:

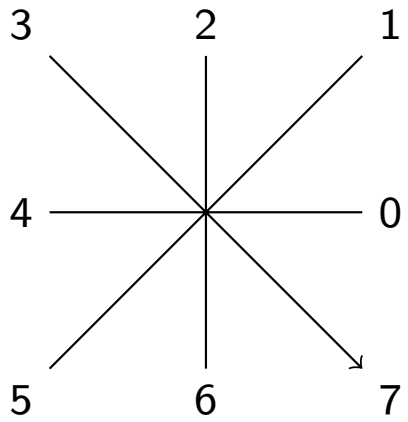
$(1,1)0007654566422222$



Richtungsketten: Beispiele

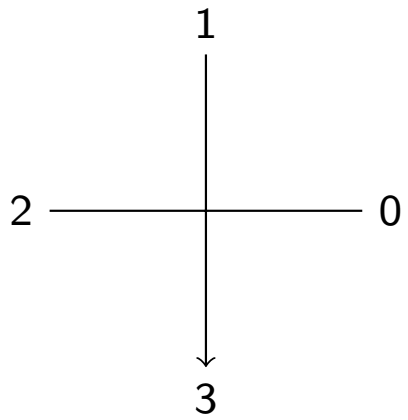
Ein Pfad kann dann über eine Folge codiert werden.
8er-Nachbarschaft:

$$(1,1)0007654566422222$$



4er-Nachbarschaft:

$$(1,1)00030323223332111 11$$



Allgemeine Formparameter

Weitere wichtige Formparameter, die teilweise in den bisher vorgestellten schon enthalten sind, sind gegeben durch

- die **Fläche** A eines Objektes,

Allgemeine Formparameter

Weitere wichtige Formparameter, die teilweise in den bisher vorgestellten schon enthalten sind, sind gegeben durch

- die **Fläche** A eines Objektes,
- den **Umfang** P der Form, die sich z.B. über Richtungsketten recht gut bestimmen lässt und

Allgemeine Formparameter

Weitere wichtige Formparameter, die teilweise in den bisher vorgestellten schon enthalten sind, sind gegeben durch

- die **Fläche** A eines Objektes,
- den **Umfang** P der Form, die sich z.B. über Richtungsketten recht gut bestimmen lässt und
- der **Rundheit** $c := \frac{P^2}{A}$. Dabei ist der Kreis mit $c = 4\pi$ optimal rund und die Werte steigen drastisch an, wenn das Objekt länglich wird.

Allgemeine Formparameter

Weitere wichtige Formparameter, die teilweise in den bisher vorgestellten schon enthalten sind, sind gegeben durch

- die **Fläche** A eines Objektes,
- den **Umfang** P der Form, die sich z.B. über Richtungsketten recht gut bestimmen lässt und
- der **Rundheit** $c := \frac{P^2}{A}$. Dabei ist der Kreis mit $c = 4\pi$ optimal rund und die Werte steigen drastisch an, wenn das Objekt länglich wird.
- Einfacher zu berechnende Annäherungen wie die Fläche des **kleinsten umgebenden Rechtecks**

Klassifikation

Klassifikation darauf ab, Objekte in (bekannte oder vorher unbekannte) Klassen aufzuteilen

1. Beschreiben Objekte durch die bekannten Methoden
2. Vergleichen der Objekte miteinander
3. Gruppieren die Objekte

Klassifikation darauf ab, Objekte in (bekannte oder vorher unbekannte) Klassen aufzuteilen

1. Beschreiben Objekte durch die bekannten Methoden
2. Vergleichen der Objekte miteinander
3. Gruppieren die Objekte

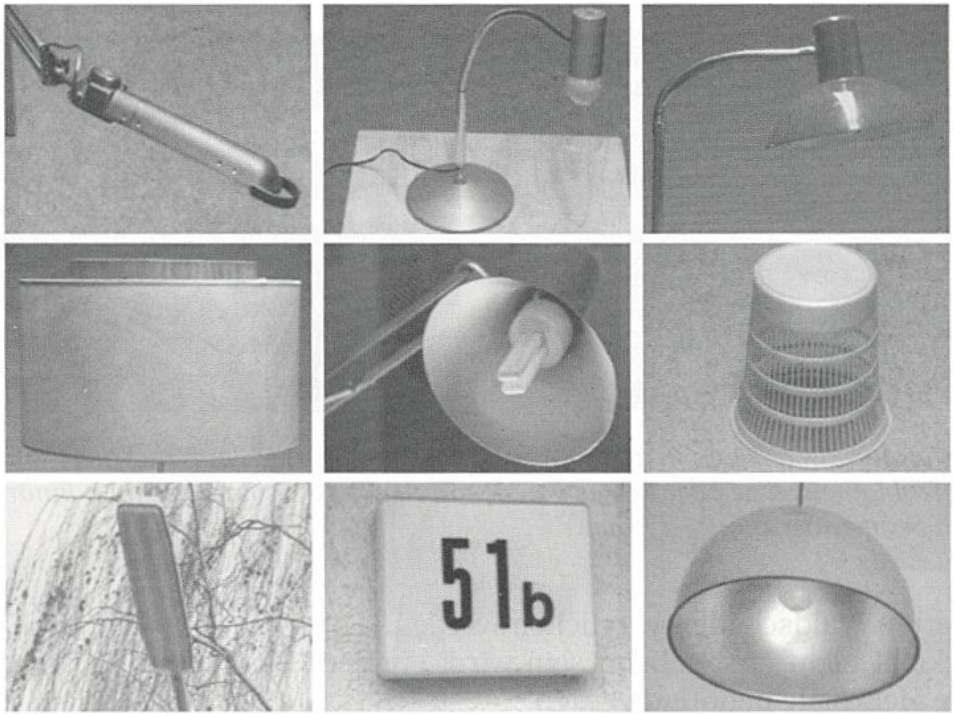


Abbildung 20.3: *Wie können wir erkennen, dass alle Objekte bis auf eines Lampen sind?*

Klassifikation

- Um die Verfahren so einfach wie möglich halten zu können, ist eine möglichst präzise Erfassung der Beziehung von Bildeigenschaften zu Objektklassen erforderlich.
- Anschließend muss ein optimaler Satz an Bildeigenschaften bestimmt werden, der eine Klassifikation mit möglichst wenig Fehlern zulässt.

Klassifikation

- Um die Verfahren so einfach wie möglich halten zu können, ist eine möglichst präzise Erfassung der Beziehung von Bildeigenschaften zu Objektklassen erforderlich.
- Anschließend muss ein optimaler Satz an Bildeigenschaften bestimmt werden, der eine Klassifikation mit möglichst wenig Fehlern zulässt.

Unterscheidung

- Bei der **pixelbasierten** Klassifikation wird versucht, jedem Pixel einen Objekttyp zuzuweisen.
- Bei der **objektbasierten** Klassifikation erfolgt die Merkmalsbestimmung auf dem ganzen Objekt.

Klassifikation

- Wenn Objekte durch P Merkmale beschrieben werden, wird jedes Objekt auf einen P-dimensionalen Vektor reduziert.
- Wenn die Merkmale die Objektklassen gut unterscheiden, liegen Objekte der gleichen Klasse nahe beieinander und bilden **Cluster**.
- Andere Klassen bilden davon getrennte Cluster.

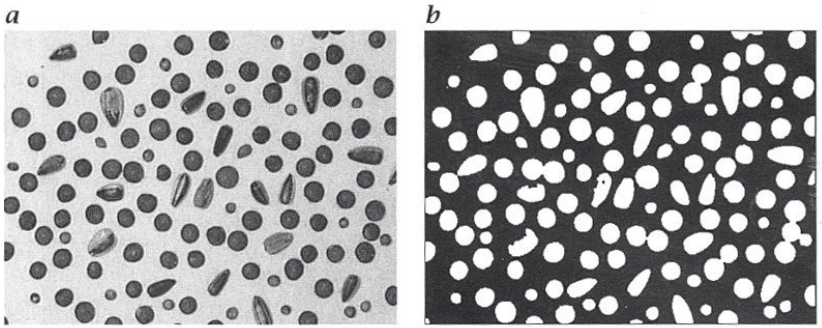


Abbildung 20.2: *Klassifikation: Welche der Samen sind Pfefferkörner, Linsen, Sonnenblumenkerne oder keines von den dreien?* **a** *Originalbild und* **b** *Binärbild nach Segmentierung.*

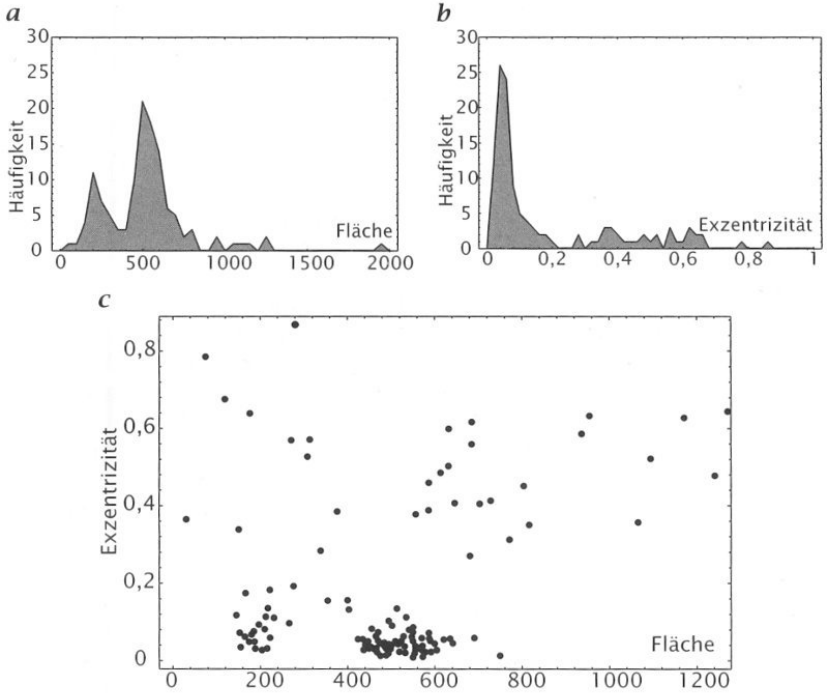


Abbildung 20.4: *Merkmale zur Klassifizierung verschiedener Samenkörner aus Abb. 20.2 in die Klassen Pfefferkörner, Linsen und Sonnenblumenkerne: Histogramm der Merkmale* **a** *Fläche und* **b** *Exzentrizität;* **c** *zweidimensionaler Merkmalsraum mit beiden Eigenschaften.*

Klassifikation

- Gute Klassifikation mittels der Merkmale Fläche und Exzentrizität
  - Linsen
  - Pfefferkörner
- Schlechte Klassifikation mittels der Merkmale Fläche und Exzentrizität
  - Sonnenblumenkerne
  - Nur halb im Bild befindliche Samen
  - Übereinander liegende Samen

Die Auswahl der Merkmale ist also sehr wichtig.

Man könnte erwarten, dass mehr Merkmale zu einer besseren Klassifikation führen.

Man könnte erwarten, dass mehr Merkmale zu einer besseren Klassifikation führen.
Wenn die Merkmale jedoch korreliert sind, gilt dies nicht.

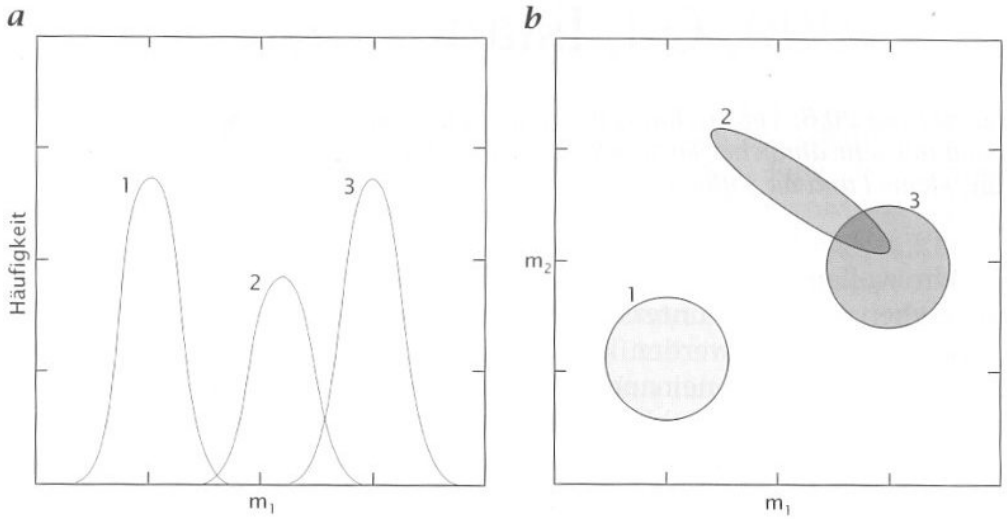


Abbildung 20.5: ***a*** *Eindimensionaler Merkmalsraum mit drei Objektklassen.* ***b*** *Erweiterung des Merkmalsraums mit einem zweiten Merkmal. Die grau eingefärbten Flächen geben die Bereiche an, in denen die Häufigkeit der entsprechenden Klasse ungleich null ist. In beiden Fällen sind dieselben Objektklassen gezeigt.*

Ziel: Erzeuge unkorrelierte Merkmale aus korrelierten Merkmalen.

- Berechne die (Kreuz-)korrelationen C_{pq} zweier Merkmale m_p, m_q

$$C_{pq} = \overline{(m_p - \overline{m_p})(m_q - \overline{m_q})}$$

- Erzeuge die symmetrische Kovarianzmatrix

$$C = \begin{pmatrix} C_{11} & \dots & C_{1P} \\ \vdots & & \vdots \\ C_{1P} & \dots & C_{PP} \end{pmatrix}$$

Ziel: Erzeuge unkorrelierte Merkmale aus korrelierten Merkmalen.

- Berechne die (Kreuz-)korrelationen C_{pq} zweier Merkmale m_p, m_q

$$C_{pq} = \overline{(m_p - \overline{m_p})(m_q - \overline{m_q})}$$

- Erzeuge die symmetrische Kovarianzmatrix

$$C = \begin{pmatrix} C_{11} & \dots & C_{1P} \\ \vdots & & \vdots \\ C_{1P} & \dots & C_{PP} \end{pmatrix}$$

Da die Matrix symmetrisch ist, gibt es eine orthonormale Basis aus Eigenvektoren e_i mit abfallenden Eigenwerten $\lambda'_1 > \lambda'_2, \dots$.
Eine Klassifikation entlang der Eigenvektoren großer Eigenwerte erlaubt in der Regel die bessere Klassifikation, da man weniger Merkmale braucht.

Hauptachsentransformation

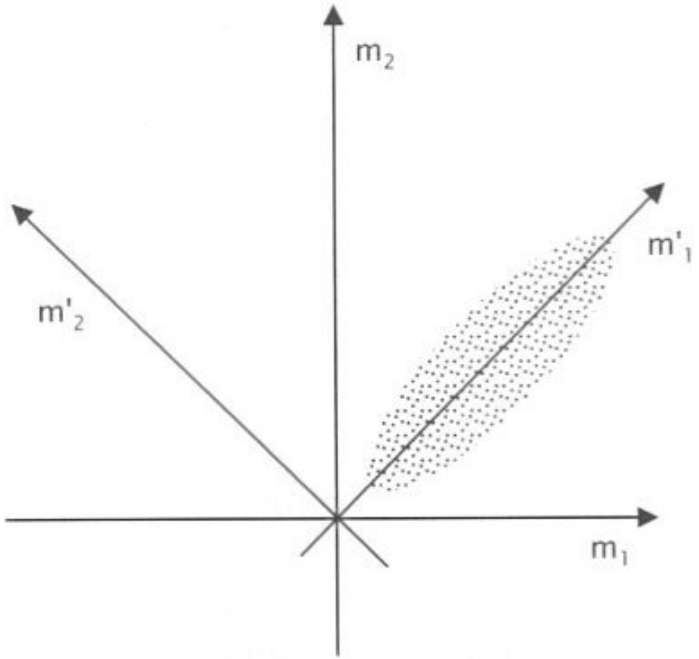


Abbildung 20.7: *Veranschaulichung korrelierter Eigenschaften und der Hauptachsentransformation.*

Klassifikation: Probleme

Nicht immer lassen sich Objekte anhand der gegebenen Merkmale in Klassen unterteilen.

- Dies kann an den Merkmalen liegen.
- Es kann aber auch an den Objekten liegen, etwa bei der Zeichenerkennung. Hier helfen Zusatzinformationen zum Kontext:
  - Großbuchstaben nur am Wortanfang oder wenn alle Buchstaben groß sind
  - Ziffern treten häufig gemeinsam auf, usw.

Klassifikation: Probleme

Nicht immer lassen sich Objekte anhand der gegebenen Merkmale in Klassen unterteilen.

- Dies kann an den Merkmalen liegen.
- Es kann aber auch an den Objekten liegen, etwa bei der Zeichenerkennung. Hier helfen Zusatzinformationen zum Kontext:
  - Großbuchstaben nur am Wortanfang oder wenn alle Buchstaben groß sind
  - Ziffern treten häufig gemeinsam auf, usw.

Beispiel: Schrifterkennung als Spezialgebiet der Formerkennung

1990, Oil, Island, L7R 4A6

- Ähnliche Zeichen wie „0“ und „O“ bereiten oft Probleme
- Hier ist Zusatzwissen über den Aufbau der Sprache oder der zu erkennenden Beschreibung hilfreich

Klassifikation: Probleme

Manchmal sind die Klassen nicht klar getrennt sind.
Beispiel: deformierte Blutzellen, die einen graduellen Übergang von “gesund” nach “pathologisch” zeigen.

Überwachte und unüberwachte Klassifikation

Die Klassifikation analysiert den Merkmalsraum, um Cluster zu finden.

- Dabei kann sie **überwacht** vorgehen: die Cluster werden mittels vorklassifizierten Objekten definiert.
- Bei der **unüberwachten** Klassifikation werden zunächst die Merkmale der Objekte ausgerechnet. Danach wird nach Häufungen gesucht. Hier können Lernverfahren zum Einsatz kommen
  - Neuronale Netze
  - Selbstorganisierende Karten
  - Support vector machines
  - ...

Klassifikation durch Nachsehen

- Wenn der Merkmalsraum nicht groß ist, kann man ihn in regelmäßige Zellen zerlegen und jeder Zelle eine Klasse (oder die „Nichtklasse“ für nicht klassifizierbare Objekte) zuordnen.
- Dann erhält jedes Objekt die Klasse der Zelle, in der sein Merkmalsvektor liegt.

- Dieser Ansatz approximiert die Cluster der Klassen mit einem Quader.
- Liegt der Merkmalsvektor in einem Quader, gehört das Objekt zur Klasse, sonst nicht.

Tabelle: Parameterverteilung und Klassifizierung am Beispiel der Linsen, Pfeffer- und Samenkörner.

	Fläche	Exzentrizität	Anzahl
Pfefferkörner	100–300	0,0–0,22	21
Linsen	320–770	0,0–0,18	67
Gesamtzahl			122
Sonnenblumenkerne	530–850	0,25–0,65	15
Nicht Klassifiziert			19

Methode des geringsten Abstands

- Diese Methode beschreibt jedes Cluster durch den Schwerpunkt.
- Der Merkmalsraum wird dann in Bereiche zerlegt, die jeweils einem Schwerpunkt am nächsten gelegen sind.

Methode des geringsten Abstands

- Diese Methode beschreibt jedes Cluster durch den Schwerpunkt.
- Der Merkmalsraum wird dann in Bereiche zerlegt, die jeweils einem Schwerpunkt am nächsten gelegen sind.
- Man kann auch noch Größenskalierungen einführen, so dass ein Objekt näher an kleinen Clustern liegen muss.
- Außerdem kann man einen maximalen Abstand einführen, so dass Objekte außerhalb dieses Radius keinem Cluster zugeordnet werden.

Methode des geringsten Abstands

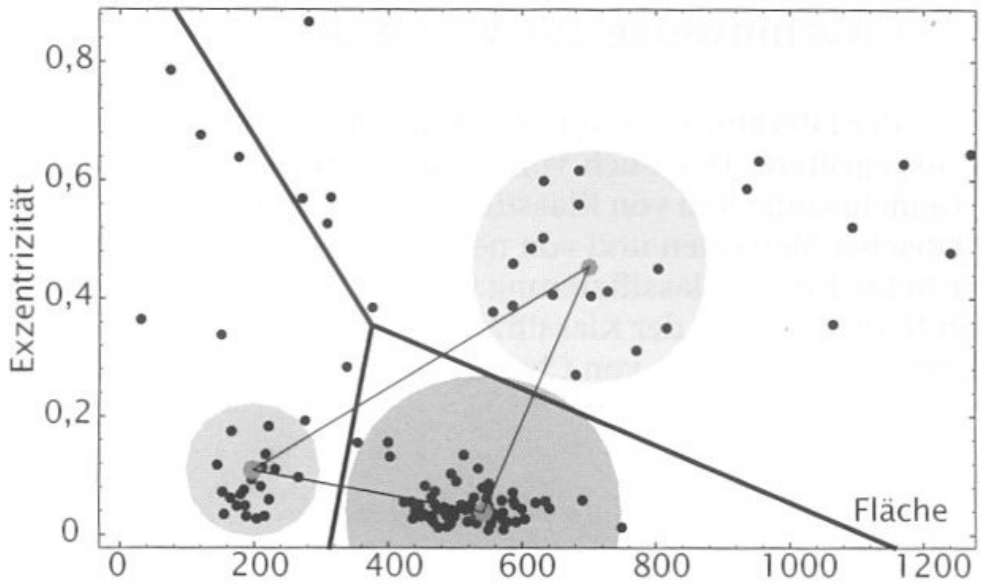


Abbildung 20.10: *Veranschaulichung der Klassifizierung verschiedener Samenkörner aus Abb. 20.2 mit der Methode des geringsten Abstandes in die Klassen Pfefferkörner, Linsen und Sonnenblumenkerne mit Hilfe der Merkmale Fläche und Exzentrizität. Ein Merkmalsvektor gehört zu dem Cluster, zu dessen Zentrum er den geringsten Abstand hat.*

Methode des geringsten Abstands: Körner klassifiziert

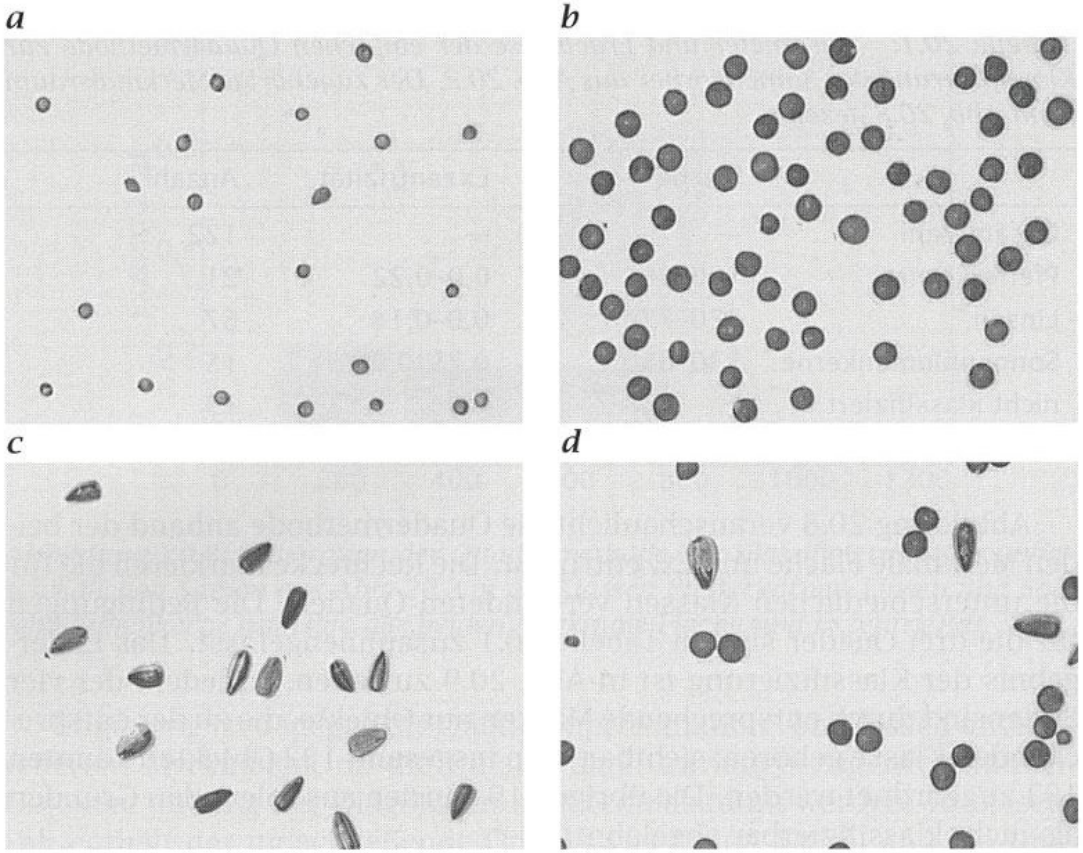


Abbildung 20.9: *Klassifizierte Objekte aus Abb. 20.2, maskiert nach den drei Klassen* ***a*** *Pfefferkörner,* ***b*** *Linsen,* ***c*** *Sonnenblumenkerne sowie* ***d*** *nicht klassifizierte Objekte.*

Ausblick

Weitere Möglichkeiten: Verfahren aus den Bereichen

- Mustererkennung
- Künstliche Intelligenz
- Data Mining

bieten viele weitere Möglichkeiten.

Teil XI

Maschinelles Lernen

Bildverarbeitungsprozess

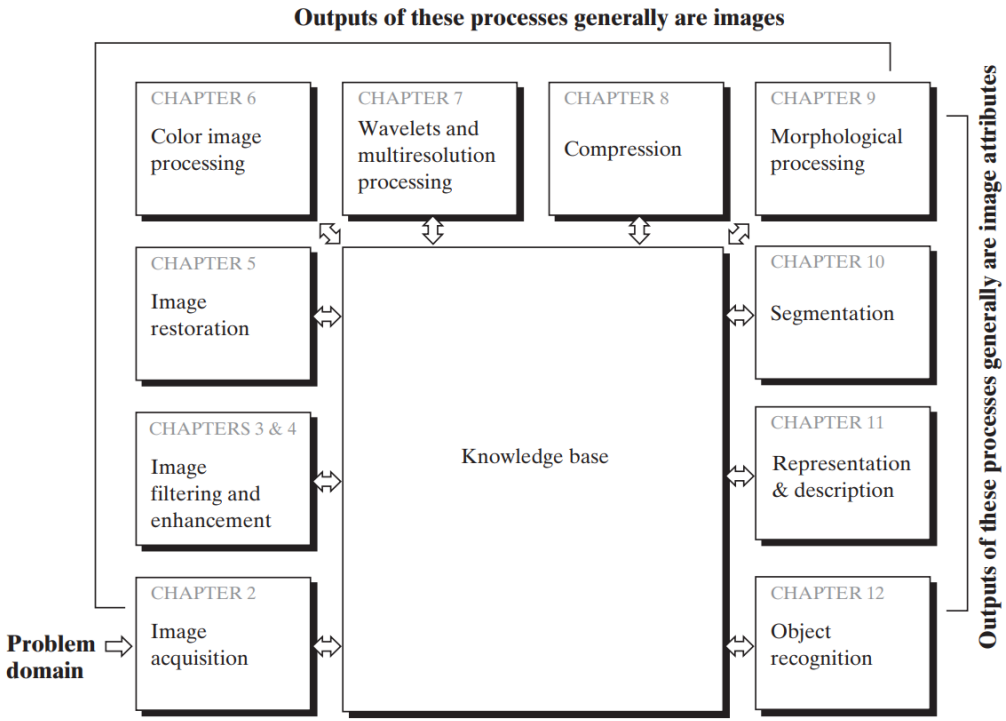


Abbildung: Die Bildverarbeitung umfasst eine reihe von Algorithmen, die entweder wiederum Bilder oder Bildattribute ausgeben. [Digital Image Processing, 3rd Edition, Gonzales and Woods]

Diese Vorlesung: Segmentierung / Objektbeschreibung

Einsatzgebiete

- Sprachgenerierung und Spracherkennung: Telefonbanking, Fremdsprachenübersetzung, automatische Ansagetexte
- Zukunftsprognosen und Trendbestimmungen: Wetter, Aktienkurse
- Selbstlernende Regel- und Steuersysteme: Robotik, autonome Fahrzeuge oder Produktionsautomaten
- Muster- und Zeichenerkennung: Bildinhalte, Schrift, Gesichter
- Spielentwicklung, Unterhaltung und Kultur: Schach, Skat, Musikkomposition, Bilderzeugung wie die App Prisma
- Optimierungsaufgaben: Optimierung von Reiserouten unter Einbeziehung varianter Faktoren wie Staumeldungen, Wetterdaten, Verkehrsdaten etc. oder im Bereich der Energieversorgung.
- Simulation: Training von Roboterarme in der virtuellen Realität zum anschließenden Einsatz in der realen Welt.

Maschinelles Lernen

Maschinelles Lernen ist ein Oberbegriff für die „künstliche“ Generierung von Wissen aus Erfahrung: Ein künstliches System lernt aus Beispielen und kann diese nach Beendigung der Lernphase verallgemeinern. Dazu bauen Algorithmen beim maschinellen Lernen ein statistisches Modell auf, das auf Trainingsdaten beruht.

Der Algorithmus erzeugt für eine gegebene Menge von Eingaben ein statistisches Modell, das die Eingaben beschreibt und erkannte Kategorien und Zusammenhänge enthält und somit Vorhersagen ermöglicht. Dabei gibt es Clustering-Verfahren, die die Daten in mehrere Kategorien einteilen, die sich durch charakteristische Muster voneinander unterscheiden. Das Netz erstellt somit selbständig Klassifikatoren, nach denen es die Eingabemuster einteilt. (siehe Clustering aus der letzten Vorlesung)

Arten des Machinellen Lernens - Unüberwachtes Lernen

Der Algorithmus lernt eine Funktion aus gegebenen Paaren von Ein- und Ausgaben. Dabei stellt während des Lernens ein „Lehrer“ den korrekten Funktionswert zu einer Eingabe bereit. Ziel beim überwachten Lernen ist, dass dem Netz nach mehreren Rechengängen mit unterschiedlichen Ein- und Ausgaben die Fähigkeit antrainiert wird, Assoziationen herzustellen. Ein Teilgebiet des überwachten Lernens ist die automatische Klassifizierung. Ein Anwendungsbeispiel wäre die Handschrifterkennung.

- Teilüberwachtes Lernen (englisch semi-supervised learning) Nur für einen Teil der Eingaben sind die dazugehörigen Ausgaben bekannt.
- Bestärkendes Lernen (englisch reinforcement learning) Der Algorithmus lernt durch Belohnung und Bestrafung eine Taktik, wie in potenziell auftretenden Situationen zu handeln ist, um den Nutzen des Agenten (d. h. des Systems, zu dem die Lernkomponente gehört) zu maximieren. Dies ist die häufigste Lernform eines Menschen.
- Aktives Lernen (englisch active learning) Der Algorithmus hat die Möglichkeit für einen Teil der Eingaben die korrekten Ausgaben zu erfragen. Dabei muss der Algorithmus die Fragen bestimmen, welche einen hohen Informationsgewinn versprechen, um die Anzahl der Fragen möglichst klein zu halten.
- Selbständiges Lernen (englisch self-training) Dieser Algorithmus kann in zwei wesentliche Komponenten eingeteilt werden. Die erste Algorithmuskomponente (Lehrer) leitet aus einem bestehenden gelabelten Datensatz weitere Datensätze mit Pseudolabeln her. Die zweite Algorithmuskomponente lernt nun aus dem erweiterten gelabelten Datensatz und wendet gefundene Muster für ihr eigenes Modell an.

Trainigs- und Testdatensätze

Ein Datensatz zum überwachten Lernen benötigt eine Groundtruth (Goldstandard).
Hier ist der gewünschte Output für einen bestimmten Inout bekannt
Man unterteilt den Datensatz aus dem man lernen möchte in zwei Gruppen:

- Trainingsdatensatz: ein Teil des Datensatzes wird genutzt um das Netzwerk zu trainieren
- Testdatensatz: ein Teil des Datensatzes wird genutzt um das trainierte Netzwerk zu testen

Diese Unterteilung ist nicht trivial!

Künstliche Neuronale Netze (KNN) sind inspiriert durch das menschliche Gehirn und lassen sich für maschinelles Lernen und die Künstliche Intelligenz einsetzen. Es lassen sich mit diesen Netzen verschiedene Problemstellungen computerbasiert lösen.

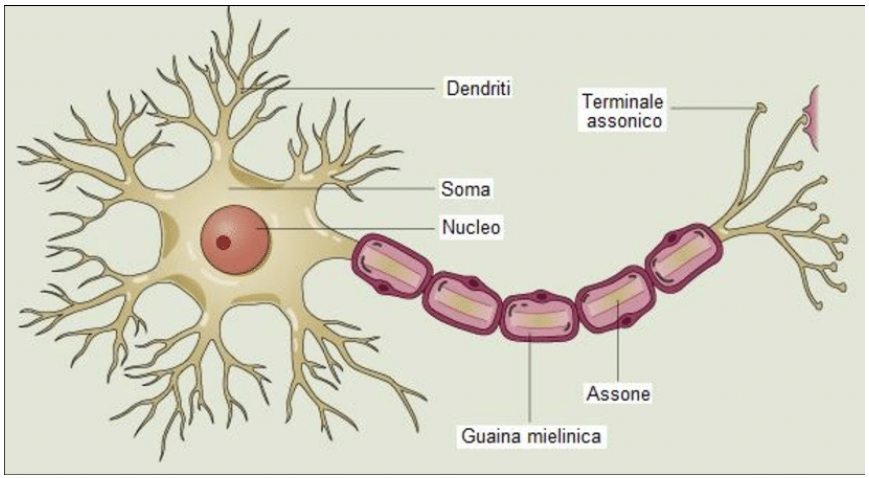


Abbildung: Neuron im Gehirn

Ein künstliches Neuron

Eine grundlegende Art ein Neuron zu modellieren ist das künstliche Neuron (node, unit). Es kann als Input von mehreren Quellen gespeißt werden und generiert mit Hilfe einer Aktivierungsfunktion einen Output.

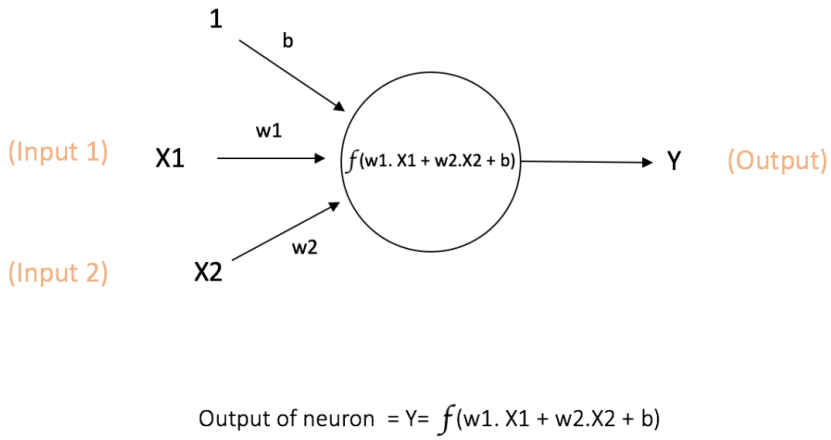
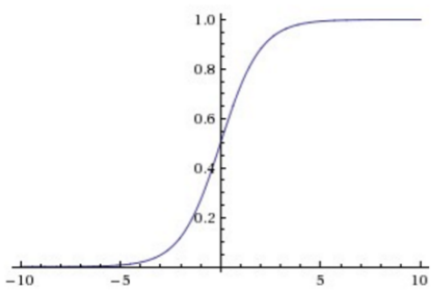


Abbildung: Neuron im Gehirn

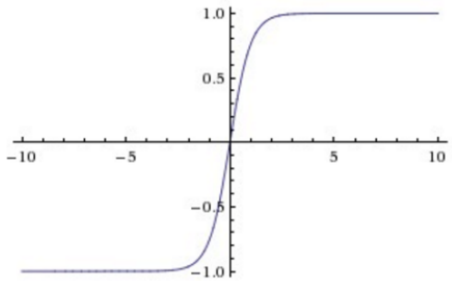
Das künstliche Neuron enthält folgende Kompenenten

- $X_1 ... X_N$ Inputs
- $w_1 ... w_N$ Gewichte
- f Aktivierungsfunktion
- b Bias
- $Y = f(w_1 * X1 + ... + w_N * X_N + b)$

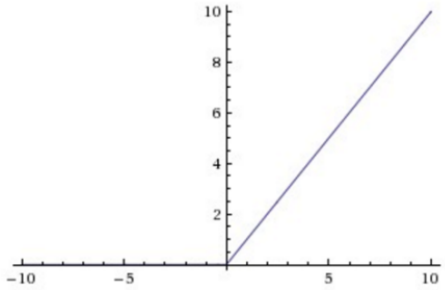
Aktivierungsfunktionen



Sigmoid



tanh



ReLU

Abbildung: Aktivierungsfunktionen

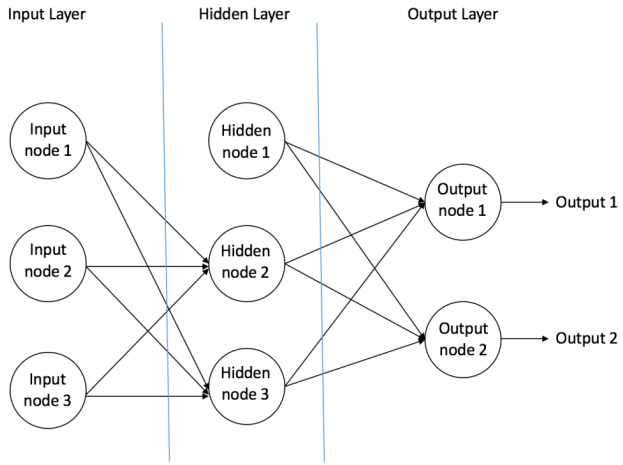


Abbildung: Zusammenschaltung von mehreren Neuronen

- Input Layer: Eingabe der Information von außen
- Hidden Layer(s): Keine direkte Verbindung nach außen. Hier findet die Berechnung statt
- Putput Layer: Kodieren das Ergebnis

Gegeben sei eine Menge von **features** $X = (x_1, x_2, \dots)$ und ein **target** y, dann kann ein MLP die Zusammenhänge zwischen X und Y erlernen.

Trainieren des Netzwerks

Das Netzwerk wird mithilfe eines Algorithmus namens **Backpropagation** oder auch **BackProp** trainiert. Es handelt sich hier um überwachtes Lernen. Das bedeutet, dass es Trainingsdatensätze gibt, bei denen der Output bekannt ist. Das Netzwerk soll von diesen bekannten Outputs lernen und diese Erkenntnisse verallgemeinern. Ziel des Algorithmus ist es die Parameter des Netzwerkes (Gewichte, Bias) so zu adjustieren, dass Input und Output möglichst gut im Netzwerk codiert werden.

Backpropagation Algorithmus

Der Backpropagation funktioniert wie folgt:

- Zufällige Bestimmung der Gewichte
- Forward Propagation
- Back Propagation und Anpassung der Gewichte
- Softmax Operation in der Output Schicht

Forward Propagation - Anhand eines Beispiels

Hours Studied	Mid Term Marks	Final Term Result
35	67	1 (Pass)
12	75	0 (Fail)
16	89	1 (Pass)
45	56	1 (Pass)
10	90	0 (Fail)

Abbildung: Bekannte Daten

Hours Studied	Mid Term Marks	Final Term Result
25	70	?

Abbildung: Daten die prädiziert werden sollen

Forward Propagation - Anhand eines Beispiels

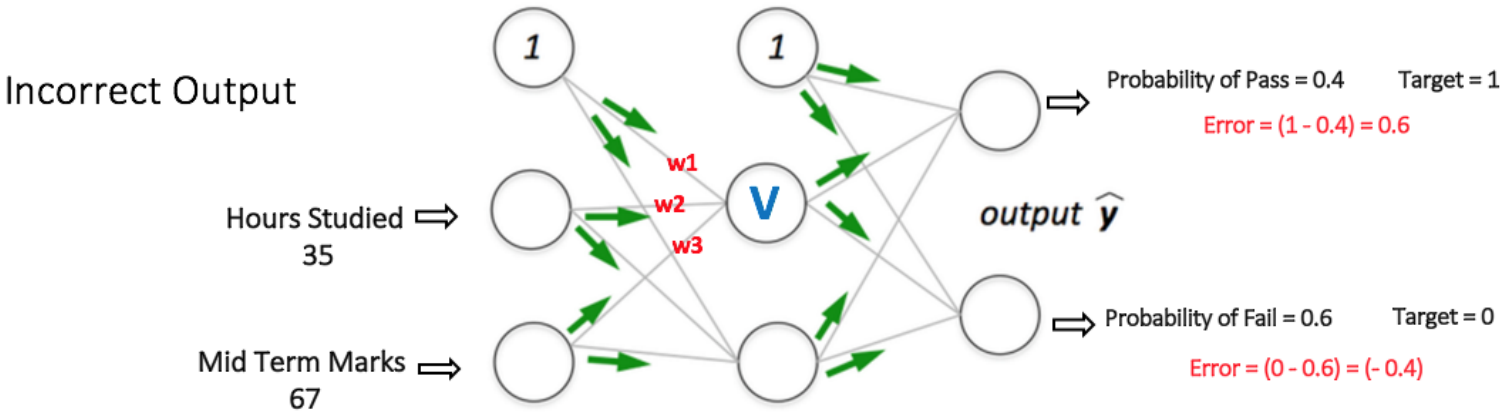


Abbildung: Forward Propagation zur Ermittlung des Fehlers

Backpropagation - Anhand eines Beispiels

Backpropagation
+
Weights Adjusted

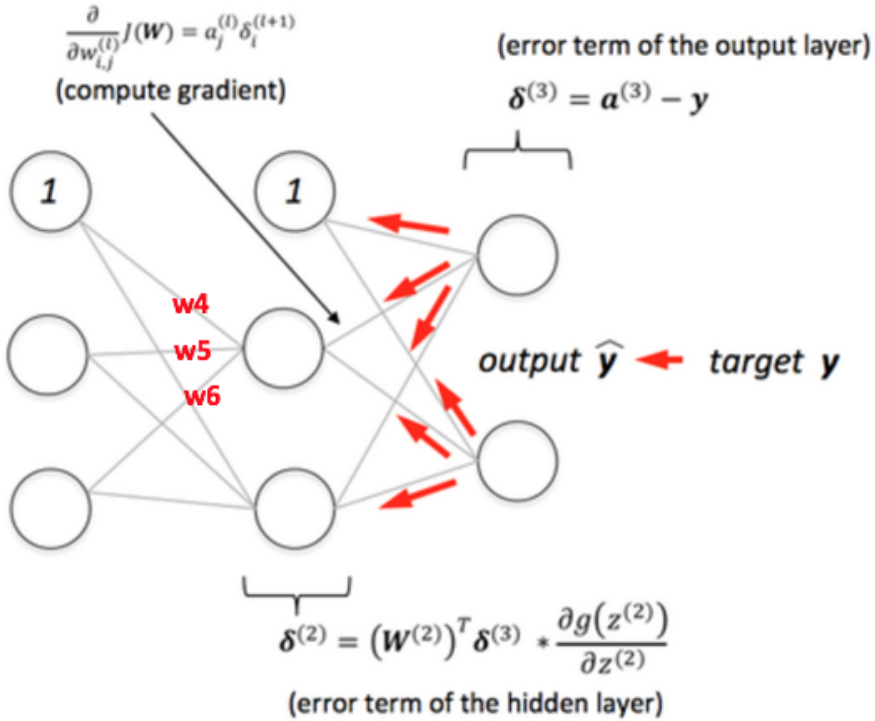


Abbildung: Back Propagation

Backpropagation - Anhand eines Beispiels

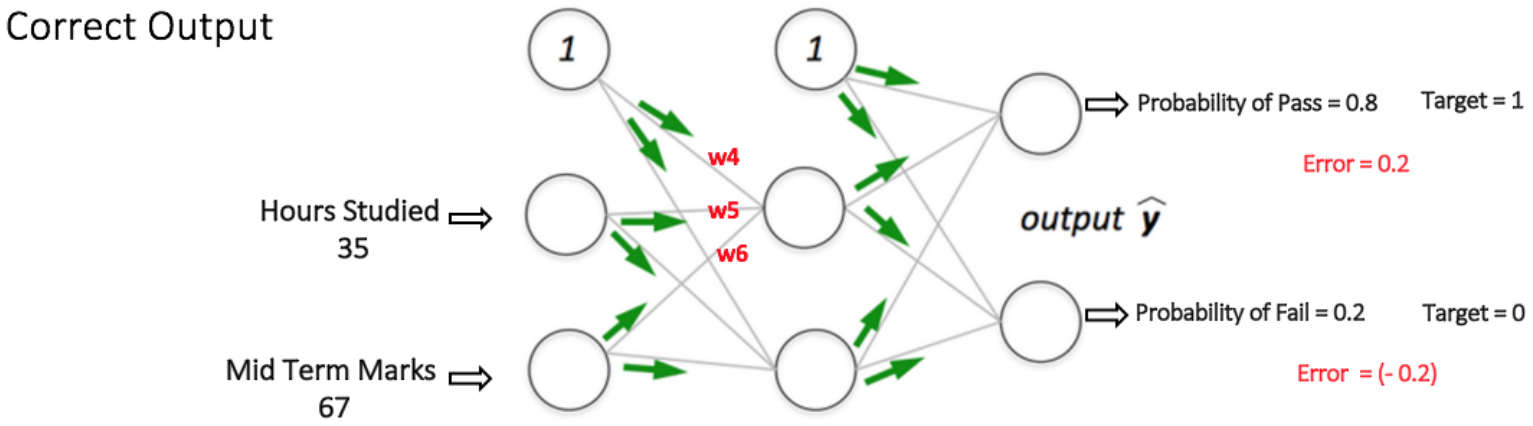


Abbildung: Resultierendes Netzwerk

Nachteile MLP

Das MLP hat folgende Nachteile:

- Der Input muss ein Vektor sein
- Die Anzahl der Schichten und Neuronen pro Schichten muss ërtestet werden"
- Der Output muss ein Vektor sein

Convolutional Neural Networks

Ein Convolutional Neural Network (CNN oder ConvNet), zu Deutsch etwa „faltendes neuronales Netzwerk“, ist ein künstliches neuronales Netz. Es handelt sich um ein von biologischen Prozessen inspiriertes Konzept im Bereich des maschinellen Lernens.
Sie sind besonders gut geeignet um Bilddaten zu analysieren.

Grundsätzlich besteht die Struktur eines klassischen Convolutional Neural Networks aus einem oder mehreren

- Convolutional Layer, gefolgt von einem
- Pooling Layer.

Diese Einheit kann sich prinzipiell beliebig oft wiederholen, bei ausreichend Wiederholungen spricht man dann von Deep Convolutional Neural Networks, die in den Bereich Deep Learning fallen. Architektonisch können im Vergleich zum mehrlagigen Perzeptron drei wesentliche Unterschiede festgehalten werden

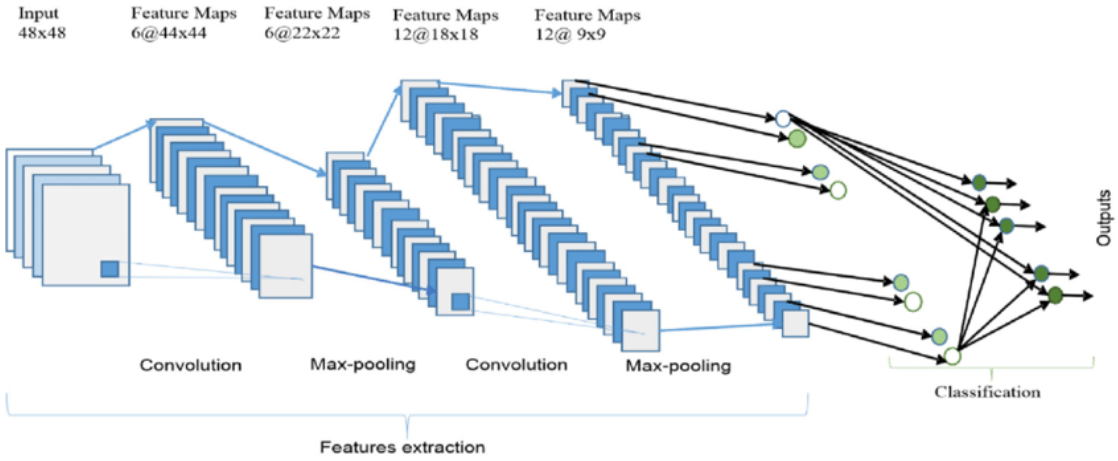


Abbildung: Pooling Layer

Convolutional Layer

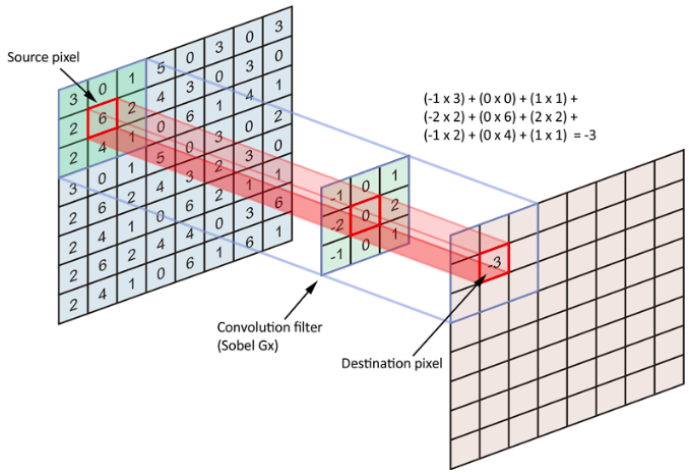


Abbildung: Convolutional Layer

Pooling Layer

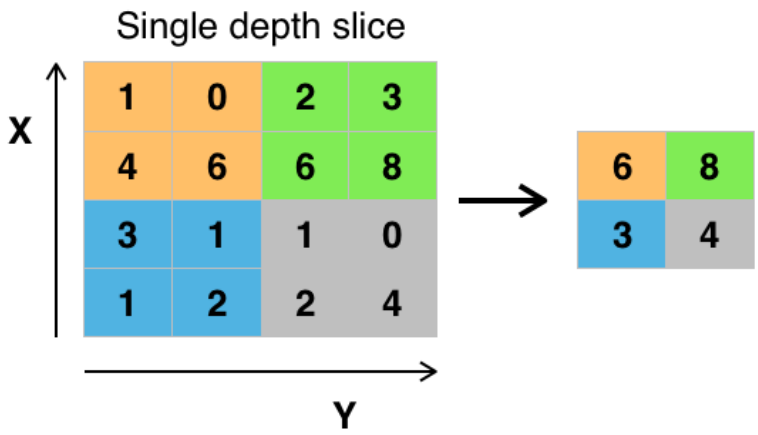


Abbildung: Pooling Layer

Training von CNN

Das Trainieren von CNNs funktioniert im wesentlichen wie auch das Trainieren eines MLP, jedoch werden zusätzlich die Kernel der Faltung optimiert. Hier ist das Ziel Kernel zu erhalten, die besonders gut wichtige Features im Bild erkennen und somit eine möglichst genaue Klassifizierung zu erhalten.

Nachteile bei CNNs

CNNs haben folgende Nachteile:

- Wie bei MLP ist die Anzahl der Schichten frei zu wählen und muss durch testen gefunden werden
- CNN können Klassifizieren, jedoch nicht lokalisieren
- Der Trainingsprozess kann sehr lange dauern

Ausblick: U-Net

U-Nets sind eine spezielle Art von CNNs, die in der Lage sind nicht nur zu klassifizieren, sondern auch zu lokalisieren.
Beispiel: Erkennung einer Lesion im Gehirn

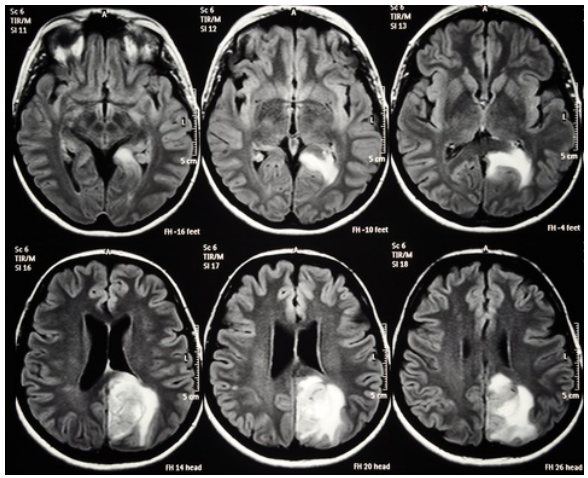


Abbildung: Lesionen im Gehirn

Ausblick: U-Net

Ein U-Net hat zum normalemn Aufbau von Pooling und Convolutional Layers zusätzlich noch eine Deconvolutional Layer, die die Positionen der identifizierten Pixel wieder rekonstruiert

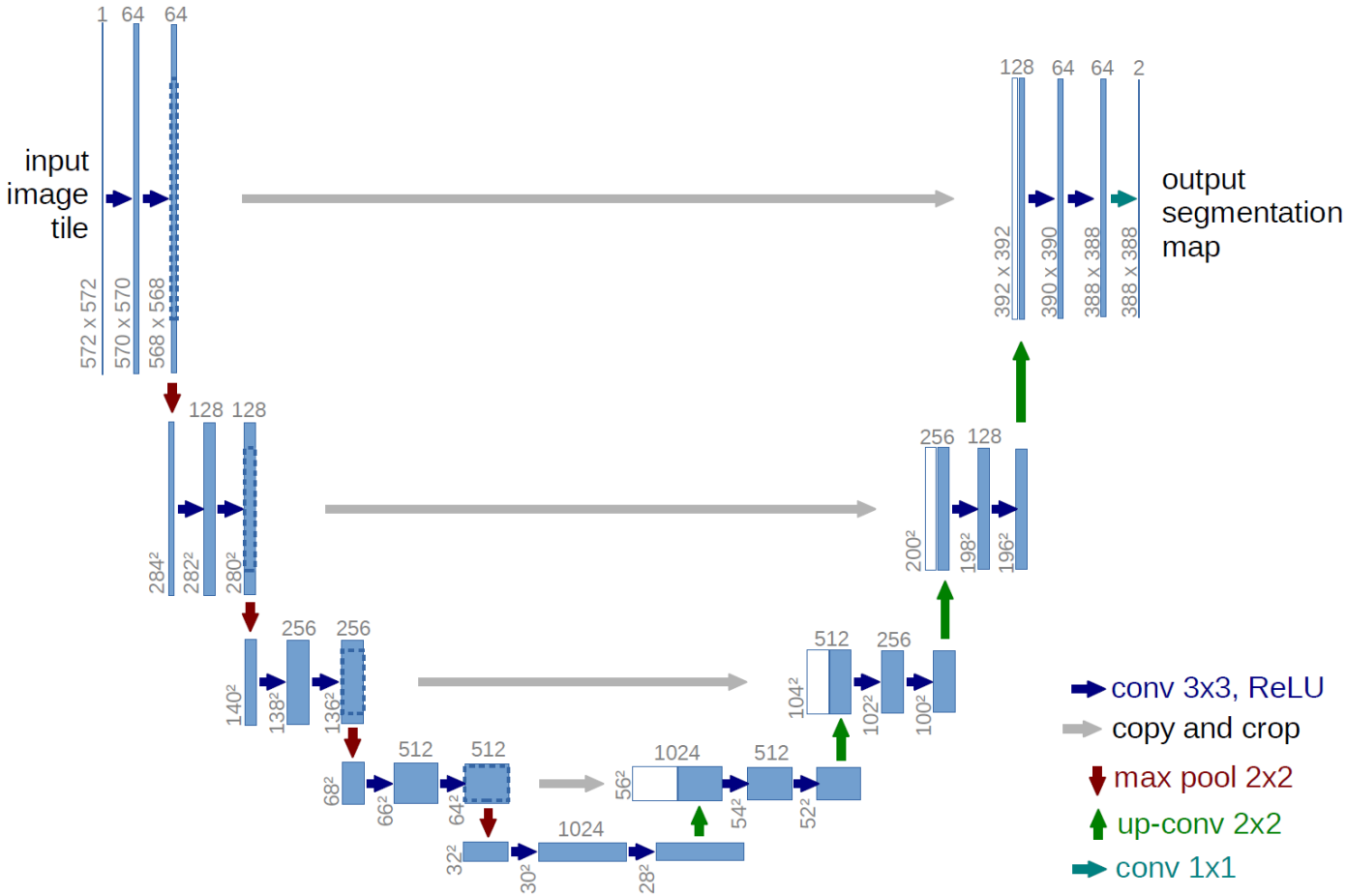


Abbildung: U-Net Architektur

Generelle Probleme beim überwachten lernen

Überwachtes Lernen hat folgende Probleme:

- Es werden Trainingsdaten benötigt
- Der Aufbau des Netzes muss getestet werden
- Es ist nicht klar, wie das Netzwerk seine Entscheidung trifft
- Es gibt noch viele weitere Neurale Netze, welches ist das beste?

Teil XII

Visualisierung von medizinischen Daten

Bildverarbeitungsprozess

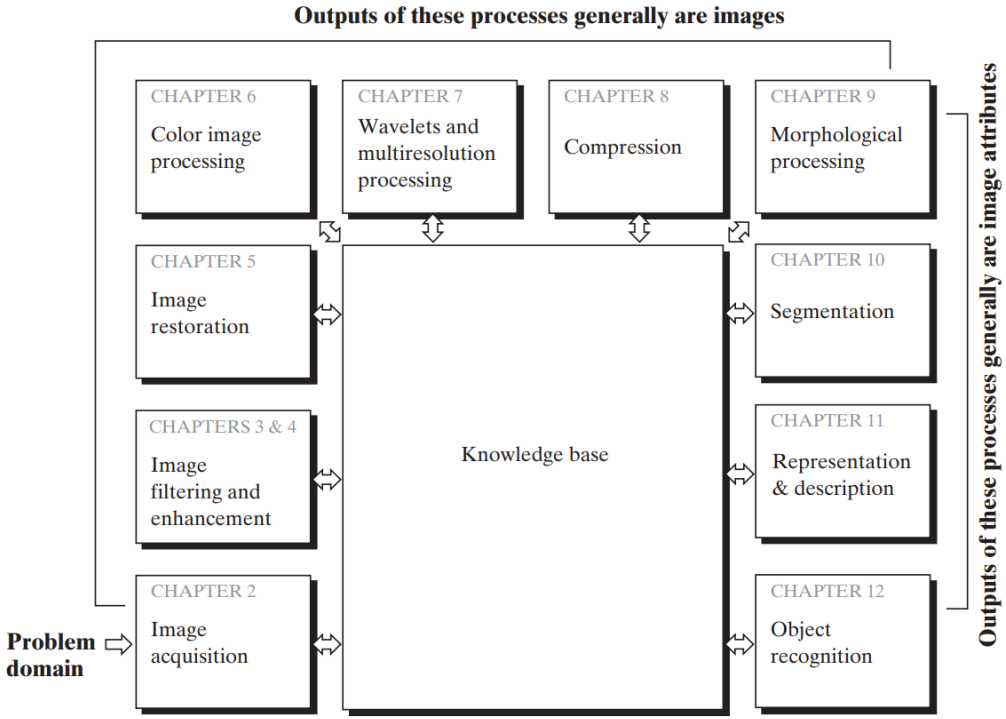


Abbildung: Visualisierung von medizinischen Bilddaten [Digital Image Processing, 3rd Edition, Gonzales and Woods]

Diese Vorlesung: Wichtig in allen Themenbereichen

Visualisierung- Motivation

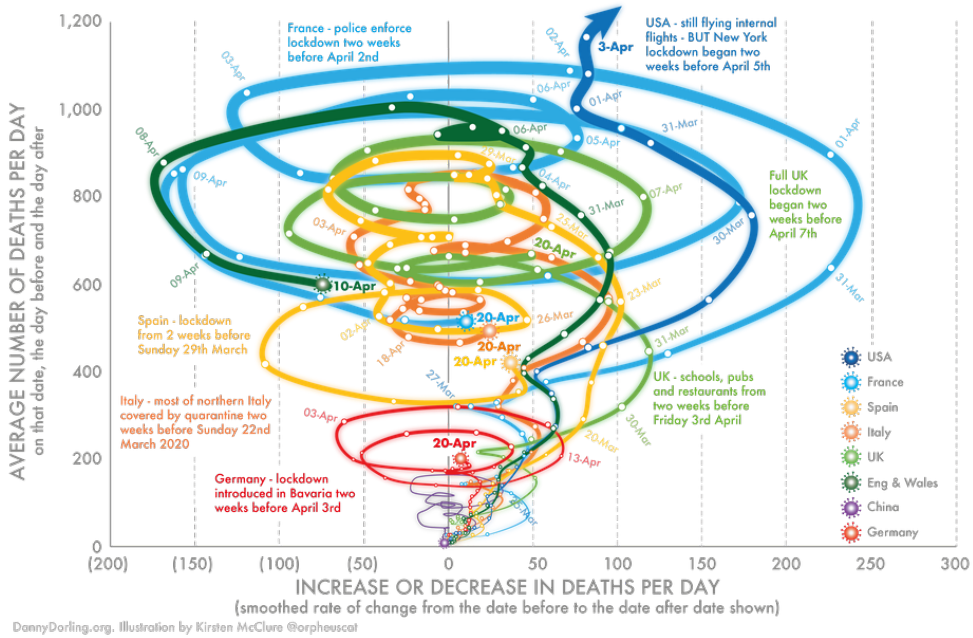


Abbildung: Beispiel einer schlechten Visualisierung

Visualisierung

Visualization or visualisation (see spelling differences) beschreibt Techniken zum Erstellen von Diagrammen, Bildern oder Animationen um einen oder mehrere Sachverhalte zu kommunizieren. Visualisierung durch Bilder hat sich als ein effektives Mittel zur Kommunikation von Konkreten Ideen oder abstrakten Konkepten herausgestellt.

Teilgebiete

- Scientific Visualization
- Educational Visualization
- Information Visualization
- Knowledge Visualization
- Product Visualization
- Visual Communication
- Visual Analytics

Anwendungen

- Flow Visualization
- Astrophysics Visualization
- Biochemical Visualization
- Digital Humanities Visualization
- Computer Aided Desgin
- **Medical Visualization**
- ...

Was macht eine gute Visualisierung aus?

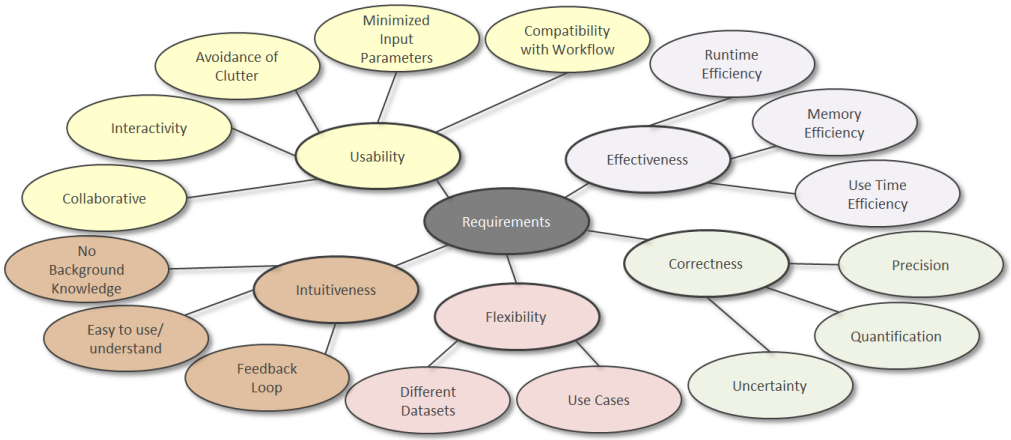


Abbildung: Kriterien für gute Visualisierungen

Medizinische Visualisierung

Medizinische Visualisierung beschreibt den Prozess der Erstellung von visuellen Representationen des inneren eines Körpers. Dies soll bei Diagnosen oder Eingriffen hilfreich sein. Das Ziel ist wichtige Strukturen und Anomalien im menschlichen Körper sichtbar zu machen. Dies geschieht normalerweise auf Grundlage von medizinischen Bildern.

Herausforderungen beim Visualisieren von medizinischen Daten

- 3D Daten (was soll angezeigt werden?)
- Wie soll es angezeigt werden?
- Wie kann man eine Interaktion ermöglichen?
- Sicherheit

Wichtige Visualisierungstechniken in der Medizin

- Slice-by-slice Visualisierung
- Direktes Volume Rendering
- Indirect Volume Rendering
- Flattening-basierte Techniken
- Sketch-basierte Techniken
- Multi-modale Visualisierung

Slice-by-Slice Visualisierung

Diese Technik ist bei Medizinern das Mittel der Wahl!

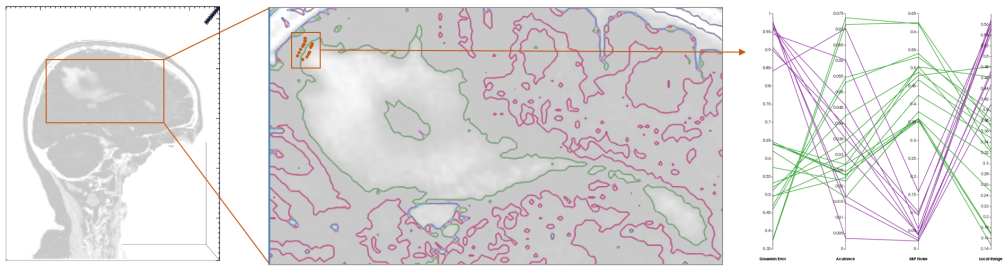


Abbildung: Slice-by-Slice Visualisierung

Pro: Unmanipulierte Daten
Con: Der Datensatz kann nicht zu komplett zu einem Zeitpunkt visualisiert werden

Bekannt aus der Computergrafik Vorlesung

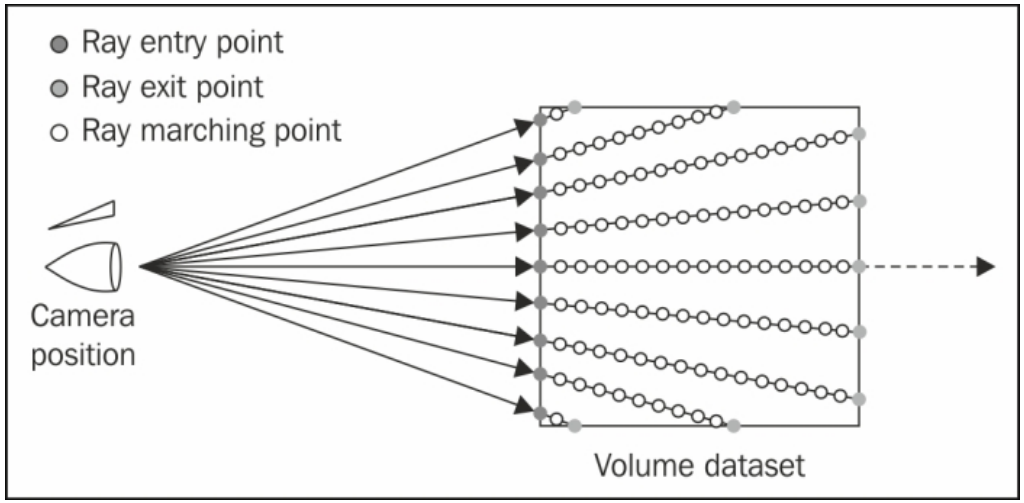


Abbildung: Funktionsweise des direkten Volume Renderings

Für Volumendaten wird eine sinnvolle Tranferfunktion benötigt, die entscheidet, was sichtbar gemacht wird

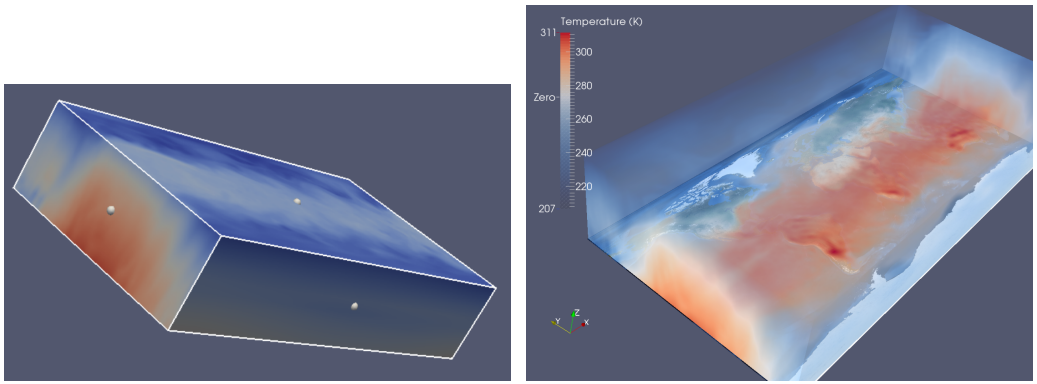
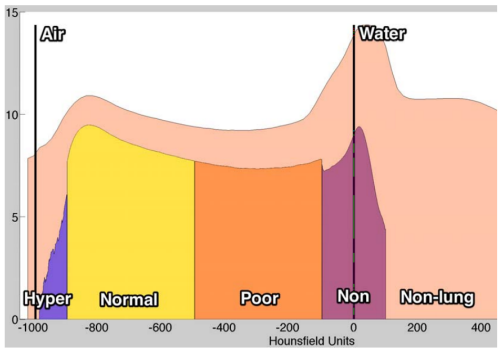


Abbildung: Notwendigkeit einer Transferfunktion

Das finden einer passenden Transferfunktion ist nicht trivial!



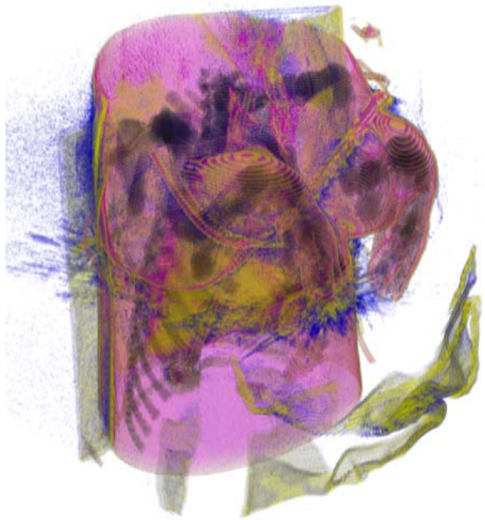


Abbildung: 1D Transferfunktionen reichen in der Medizin nicht aus

Mehrdimensional Transferfunktionen möglich

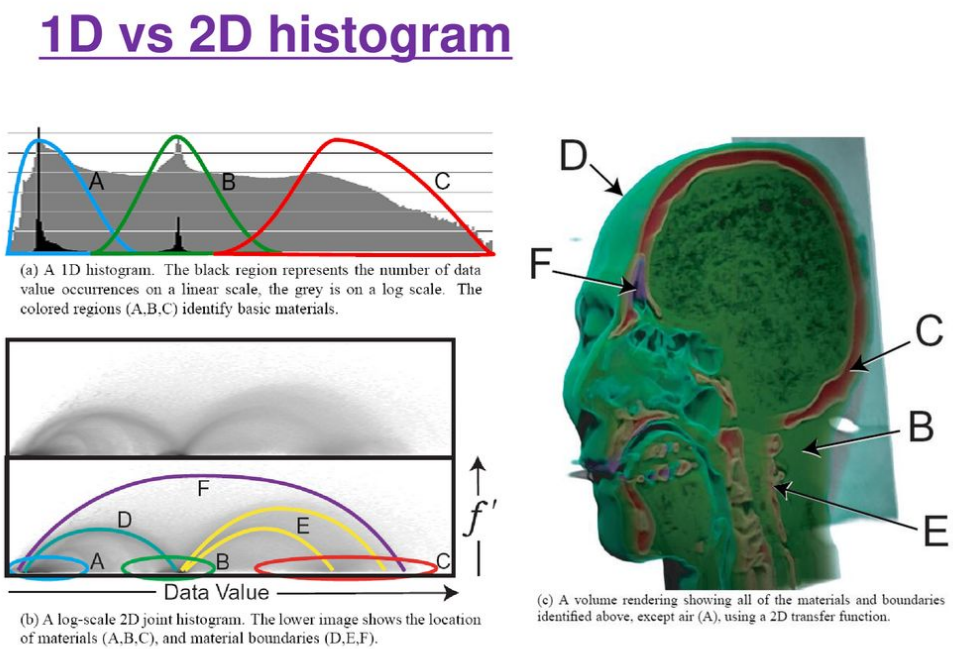


Abbildung: 2D Transferfunktionen erzeugen deutlich bessere Ergebnisse

Pro: besseres Ergebnis
Con: Das finden einer solchen Funktion ist deutlich komplexer!

Zur Auswahl der richtigen Bereiche können vor dem Volume Rendering auch Segmentierungen angewandt werden

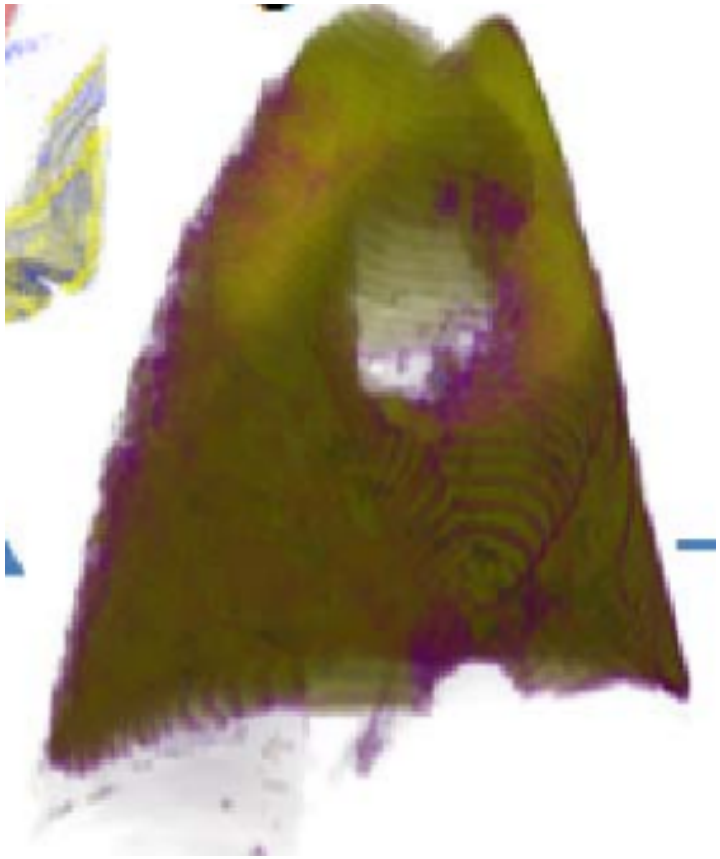


Abbildung: Einschränkung des zu visualisierunden Bereichs durch eine Segmentierung.

Pro: besseres Ergebnis
Con: Das finden einer solchen Funktion ist deutlich komplexer!

Hierbei wird die zu visualisierende Struktur durch eine Geometry angenähert

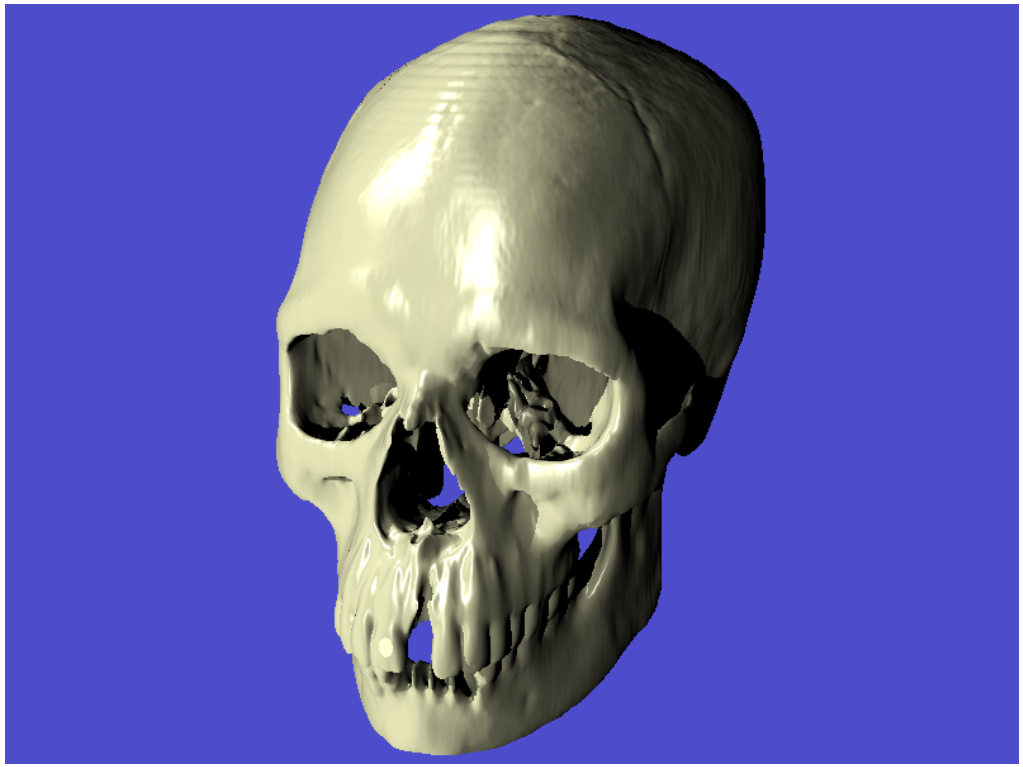


Abbildung: Indirektes Volume Rendering eines Schädels

Hierbei wird die zu visualisierende Struktur durch eine Geometry angenähert

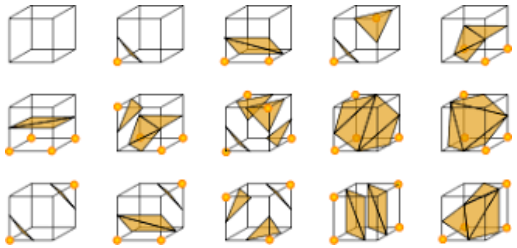


Abbildung: Bekannter Algorithmus: Marching Cubes

Pro: keine Transferfunktion notwenig
Con: Auch hier benötigt man oft eine Segmentierung um gute Ergebnisse zu erziehlen

Diese kommen oft bei vaskulären Strukturen zum Einsatz

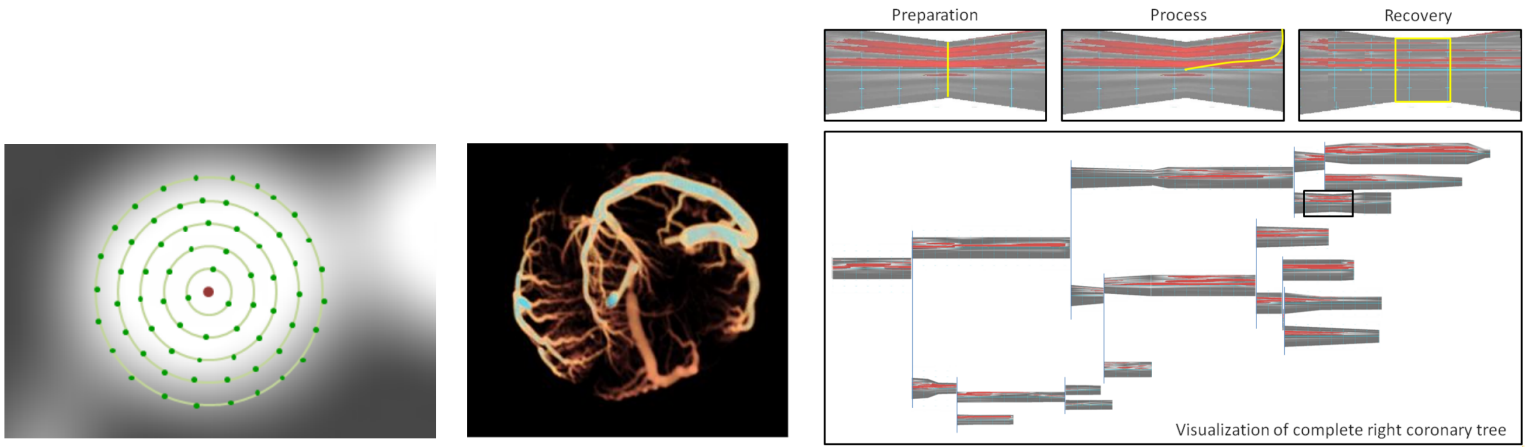


Abbildung: Die vaskulären Strukturen werden auf ein 2D Bild gefaltet.

Pro: 2D eliminiert Verdeckung
Con: Informationen gehen verlohren

Diese Techniken werden oft zum Unterrichten benutzt

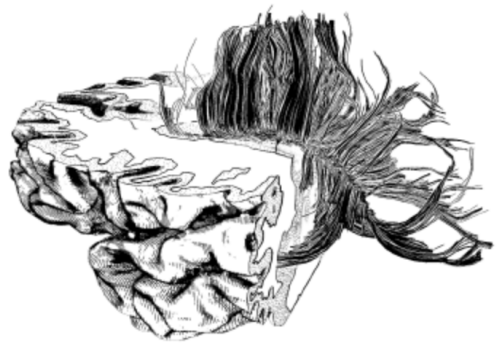


Abbildung: Sketch-basierte Visualisierung eines Gehirns

Pro: Vereinfachung auf wichtige anatomische Strukturen
Con: Kein Bezug zu realen klinischen Verhältnissen

Oft werden in der Medizin mehrere Bildgebende Verfahren verwendet um eine ganzheitliche Diagnose zu erziehlen.

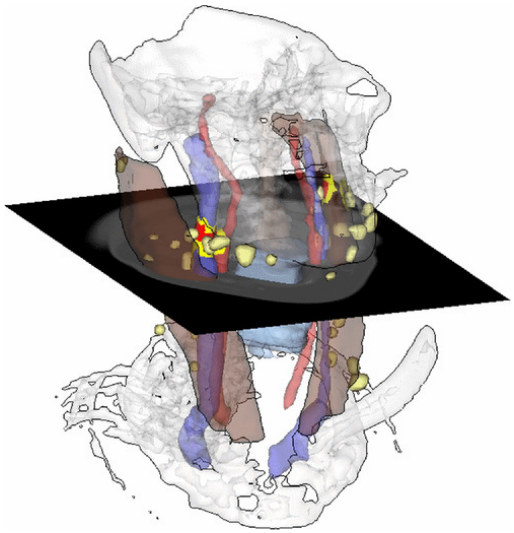


Abbildung: Multi-modale Visualisierung

Pro: Alle Bildergebungen können gleichzeitig gesichtet werden
Con: Informationen müssen selektiert werden
Con: Bilder müssen registriert werden

Beispiel - Verwednung mehrerer Techniken

Meistens ergibt sich aus einer Kombination mehrerer Techniken das beste Ergebnis.

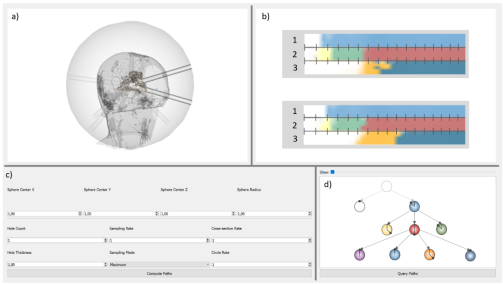


Abbildung: Kombination verschiedener Visualisierungsarten

Achtung: Die Wahl der richtigen Visualisierungstechniken ist nicht trivial!

Teil XIII

Unsicherheiten

Bildverarbeitungsprozess

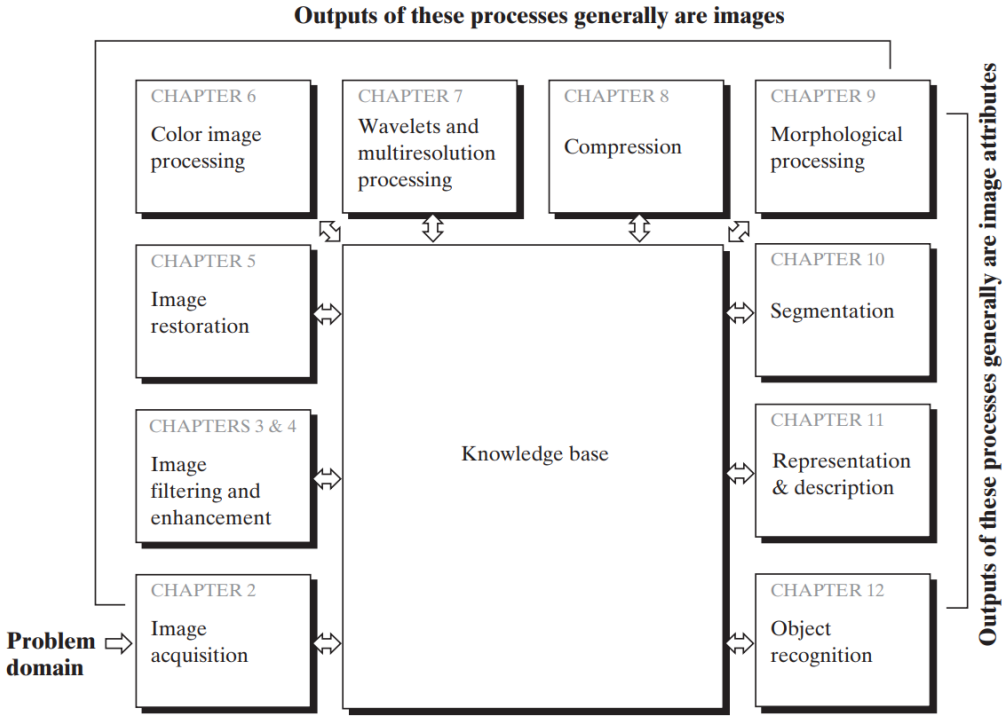


Abbildung: Unsicherheiten in medizinischen Bilddaten betreffen jeden Schritt

Diese Vorlesung: Unsicherheiten

Unsicherheiten- Motivation

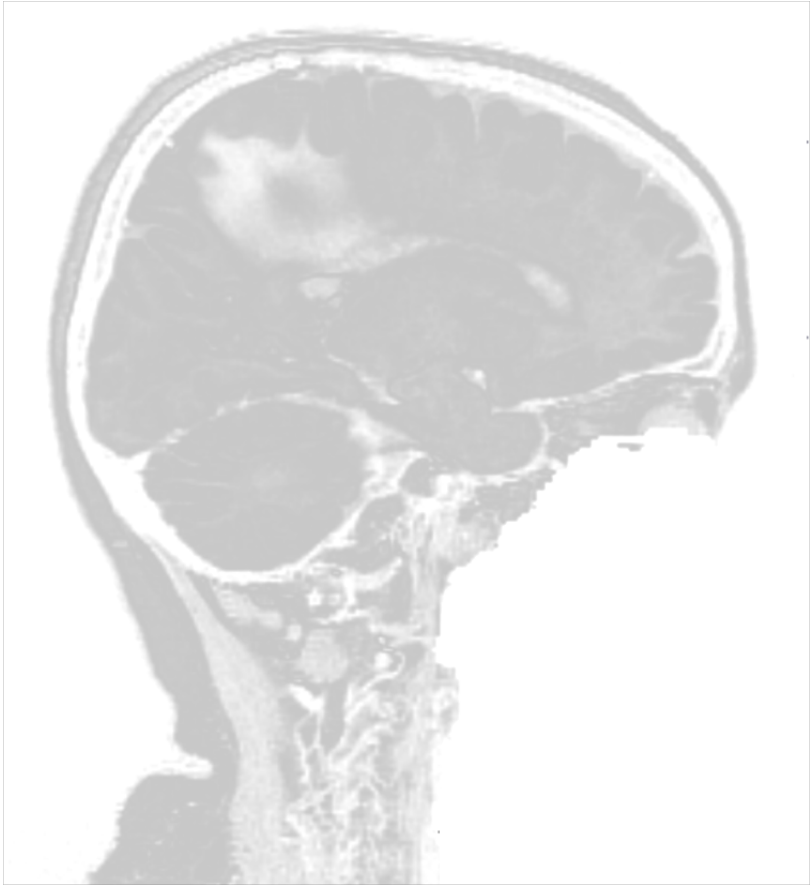


Abbildung: Tumor in Magnet Resonanz Tomographie

- Wie groß ist der Tumor?
- Welche Behandlung kommt in Frage?
- Wie kann man mit solchen Werten rechnen?

Unsicherheiten- Motivation

Der berichtete sensitive R-Wert kann durch Verwendung eines gleitenden 4-Tage-Mittels der durch das Nowcasting geschätzten Anzahl von Neuerkrankungen geschätzt werden. Dieser 4-Tage-Wert bildet das Infektionsgeschehen vor etwa einer bis zwei Wochen ab. Dieser Wert reagiert auf kurzfristige Änderungen der Fallzahlen empfindlich, wie sie etwa durch einzelne Ausbruchsgeschehen verursacht werden können. Dies kann insbesondere bei einer insgesamt kleinen Anzahl von Neuerkrankungen zu verhältnismäßig großen Schwankungen führen. Mit Datenstand 24.06.2020, 0:00 Uhr wird der 4-Tage-R-Wert auf 0,72 (95%-Prädiktionsintervall: 0,56 – 0,91) geschätzt.

Analog dazu wird das 7-Tage-R durch Verwendung eines gleitenden 7-Tage-Mittels der Nowcasting-Kurve geschätzt. Schwankungen werden dadurch stärker ausgeglichen, da dieser Wert das Infektionsgeschehen vor etwa einer bis etwas mehr als zwei Wochen abbildet. Mit Datenstand 24.06.2020, 0:00 Uhr wird der 7-Tage R-Wert auf 1,17 (95%- Prädiktionsintervall: 1,08 – 1,25) geschätzt.

Abbildung: Reproduktionszahl bei Corona

- Wie viele Menschen stecken sich mit dem Corona-Virus an?
- Welche Maßnahmen sollen ergriffen werden?

Unsicherheiten - Definition

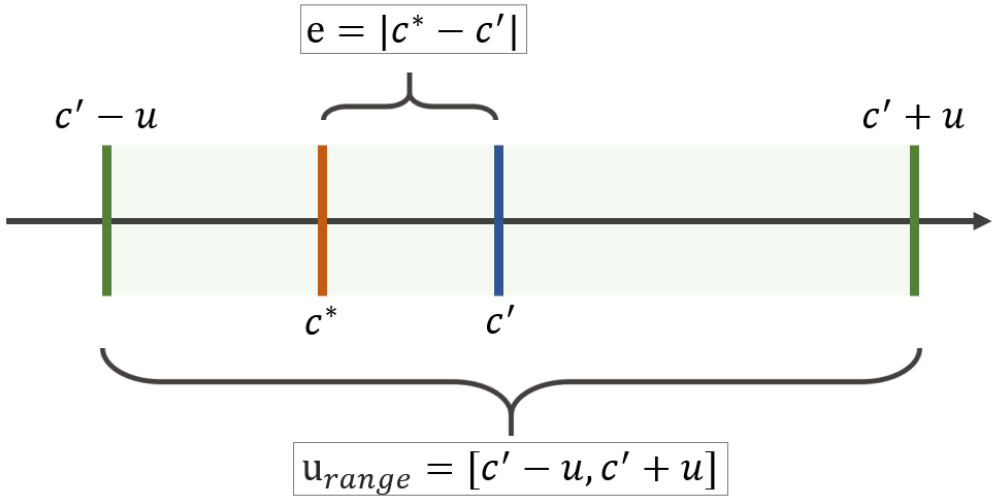


Abbildung: Definition von Unsicherheiten

- Fehler $\neq$ Unsicherheit!
- Keine einheitliche Definition von Unsicherheit
- Unsicherheit kann nur geschätzt werden

Quellen von Unsicherheiten beim Aufnehmen von Bildern

- unvollständige Definition des Messwertes
- nicht perfekte implementierung eines Messwertes
- nicht representatives Sampling
- unvollständiges Wissen über die phzsikalischen Begebenheiten
- persönlicher Bias beim betrachten von Bildern
- endliche Auflösung des Sensors
- nicht perfekte Referenzwerte
- nicht perfekte Parameter
- Approximationen und Annahmen im Model
- Abweichung bei mehrmaligem Messen

Taxonomie von Unsicherheiten

Unsicherheiten lassen sich grob in zwei verschiedene Arten unterteilen

- aleatorisch, vom Zufall abhängig
- epistemisch, das Wissen betreffend

Epistemische Unsichheit ist in der Regel die interessantere Unsicherheit.

Beschreibung von Unischerheiten

Unsicherheiten kann man mathematisch ausdrücken durch:

- Schranke $(p \mp x)$
- Wahrscheinlichkeitsverteilung (p, σ)

Im Prinzip ist jedoch jede Art von mathematischer Beschreibung zulässig

Wie wirken sich Unsicherheiten in Bildern aus?

- Positionelle Unsicherheit
- Unsicherheit der Pixel(Voxel)- werte

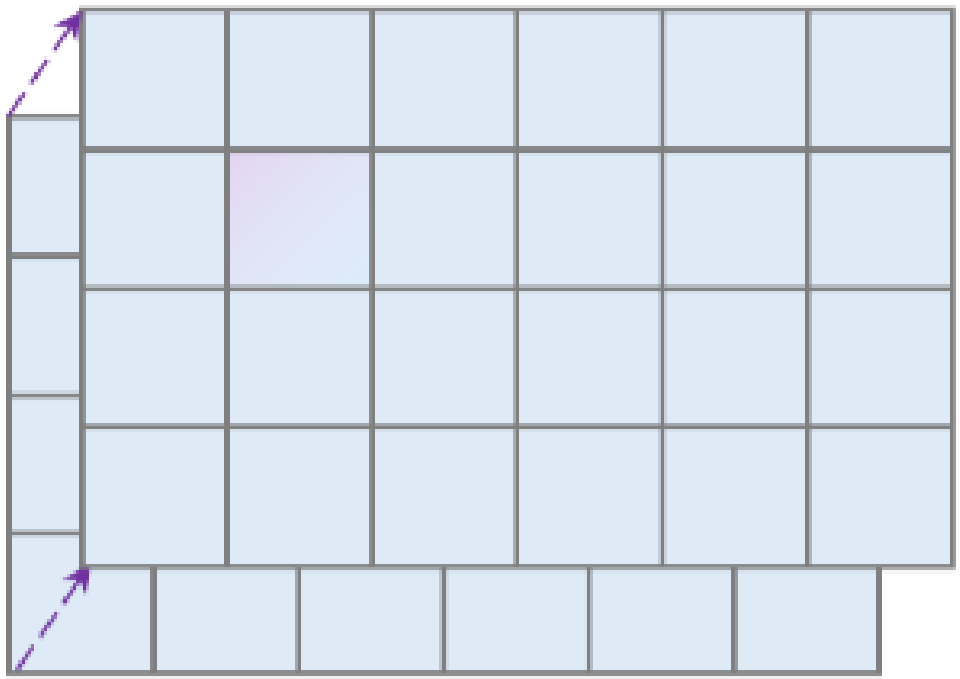


Abbildung: Unsicherheiten in Bilddaten

Wie kann man Unsicherheiten Quantifizieren

Die Quantifizierung von Unsicherheiten ist nicht eindeutig! Sie hängt von folgenden Faktoren ab:

- Anwendung
- Wissen über die Daten

Mögliche Methoden zur Quantifizierung

- Schätzen von Rauschen
- Schätzen der Entropie
- Schwellwertunsicherheiten

Annahme: Wenn ich die Nachbarschaft eines Pixels $I(v)$ betrachte ($I(n(I(v))$ und der aktuelle Pixel stark vom Mittel dieser Nachbarschaft abweicht, dann handelt es sich wahrscheinlich um einen falschen Wert.

$$u_A(I(v)) = \frac{|\Sigma(I(n(I(v), k))) - I(v)|}{I_r} \tag{3}$$

I_r beschreibt die Größe des Wertebereichs
$I(n(I(v)$ beschreibt das Mittel der Nachbarpixel jedoch ohne den Pixel selbst (Boxfilter ohne zentralen Pixel)
Vorteil: Bilder mit Rauschen können sehr gut quantifiziert werden.
Nachteil: Bei alternierenden Mustern funktioniert diese Quantifizierung nicht gut

Annahme: Je häufiger ein Pixelwert in einem Histogramm vertreten ist, desto höher ist, desto mehr Information wird von ihm abgedeckt.

$$u_E(I(v)) = -\Sigma_1^n h(v) * log(h(v)) \tag{4}$$

Vorteil: Unabhängig von Nachbarschaft eines Pixels
Nachteil: Bei Kontrastarmen Bildern kann es zu falschen Quantifizierungen kommen

Annahme: Je näher ein Pixelwert an dem gewählten Schwellenwert (d) liegt, desto unsicherer ist er.

$$u_T(I(v)) = -\frac{1}{\sqrt{2\pi\sigma}} * e^{\frac{(I(v)-d)^2}{2\sigma^2}} \tag{5}$$

σ kann frei gewählt werden, jedoch bieten sich Werte wie 1 oder 2 an
Vorteil: Schwellenwert ist kein fester Wert mehr
Nachteil: Schwellenwert muss gewählt werden

Quantifizieren von Unsicherheiten

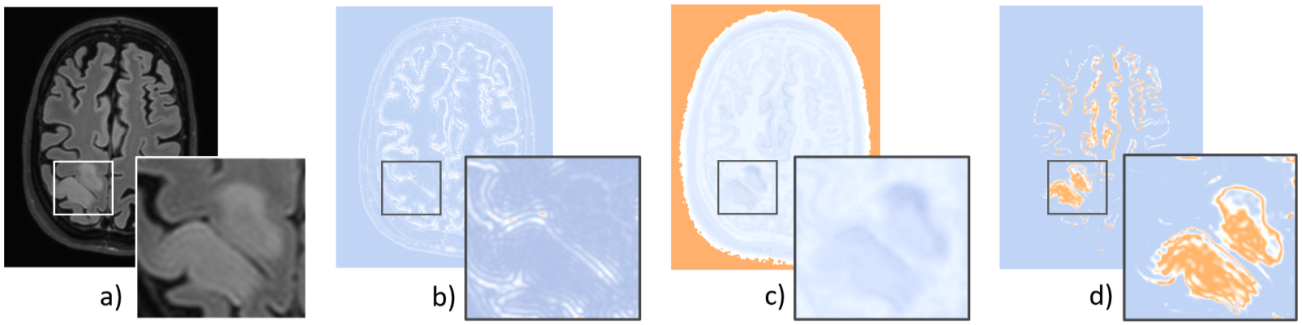


Abbildung: a) Original Bild. b) Schätzen von Rauschen. c) Schätzen von Entropie d) Schätzen von Schwellwertunsicherheiten

Was bringt uns diese Quantifizierung nun?

Quantifizierung von Unsicherheiten bringt folgende Vorteile:

- Wir können sehen welche Datenpunkte sicher sind
- Wir können beschreiben wie Sicher eine Schlussfolgerung ist
- Wir können verschiedene Szenarien betrachten

Wie wir in der Vorlesung schon gesehen haben werden aufgrund von Bilddaten Ergebnisse berechnet (Kantendetektion, Segmentierung ...).
Wie wirken sich diese Unsicherheiten nun auf solche Ergebnisse aus?

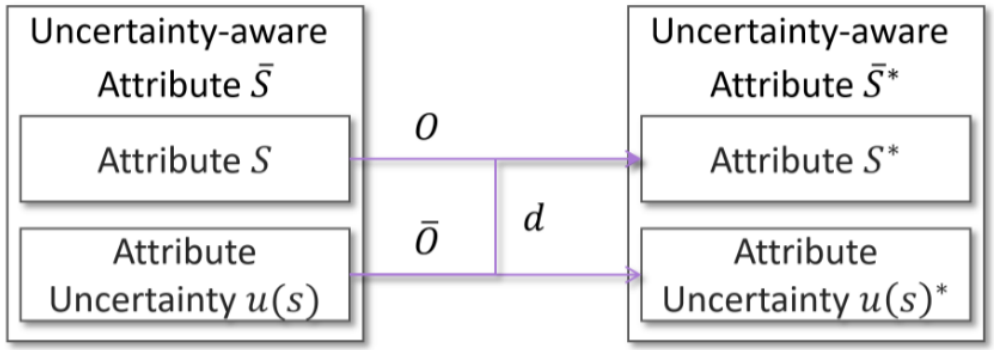


Abbildung: Propagationsregeln bei Unsicherheiten

Sie besteht nun aus drei Rechnenwegen:

- Der bereits bekannte Rechenweg O
- Die Propagation der Unsicherheiten $\bar{O}$
- Die Unterdrückung von Unsicheren Werten d

Wie wir in der Vorlesung schon gesehen haben werden aufgrund von Bilddaten Ergebnisse berechnet (Kantendetektion, Segmentierung ...).
Wie wirken sich diese Unsicherheiten nun auf solche Ergebnisse aus?
Lösung: Propagation von Unsicherheiten

Equation	**Uncertainty Propagation**
$y = c \cdot x$	$\Delta y = c \cdot \Delta x$
$z = x_1 \pm \ldots \pm x_n$	$z = \sqrt{(\Delta x_1)^2 + \ldots + (\Delta x_n)^2}$
$z = x \cdot y$ or $z = x/y$	$\Delta z = \sqrt{\frac{(\Delta x)^2}{x} + \frac{(\Delta y)^2}{y}}$
$y = x^n$	$\Delta y = n\frac{\Delta x}{x}$
$y = f(x)$	$\Delta y = \frac{\delta f}{\delta x}\Delta x$
$y = f(x_1, x_2, \ldots x_n)$	$\Delta y = \frac{\delta f}{\delta x_1}\Delta x_1^2 + \frac{\delta f}{\delta x_2}\Delta x_2^2 + \ldots + \frac{\delta f}{\delta x_n}\Delta x_n^2$

Abbildung: Propagationsregeln bei Unsicherheiten

Unterdrücken von Unsicheren Werten

Wir möchten, dass Unischere Werte bei einer Berechnung weniger stark berücksichtigt werden als sichere Werte.
Wir benötigen eine mathematische Methode, die in der Lage ist Werte zu unterdrücken (englisch "damping")
Dies kann durch folgende Funktion erreicht werden:

$$d(I)_V = \frac{1}{u(I)_V}$$

Abbildung: Damping bei unsicheren Werten

wenn $u(v) = 0$, dann gilt $d(v) = 1$

Gaußfilter:

$$B^2 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Anwendung auf Bildausschnitt:

$$X = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix}$$

$$u(X) = \begin{bmatrix} u(x_1) & u(x_2) & u(x_3) \\ u(x_4) & u(x_5) & u(x_6) \\ u(x_7) & u(x_8) & u(x_9) \end{bmatrix}$$

Ergibt:

$$X' = \frac{1}{16}(x_1 + 2x_2 + x_3 + 2x_4 + 4x_5 + 2x_6 + x_7 + 2x_8 + x_9)$$

Beispiel einer Propagation - Gaußfilter

Wie berechnet man $u(X')$?

$$u(X') = u(\frac{1}{16}(x_1 + 2x_2 + x_3 + 2x_4 + 4x_5 + 2x_6 + x_7 + 2x_8 + x_9))$$

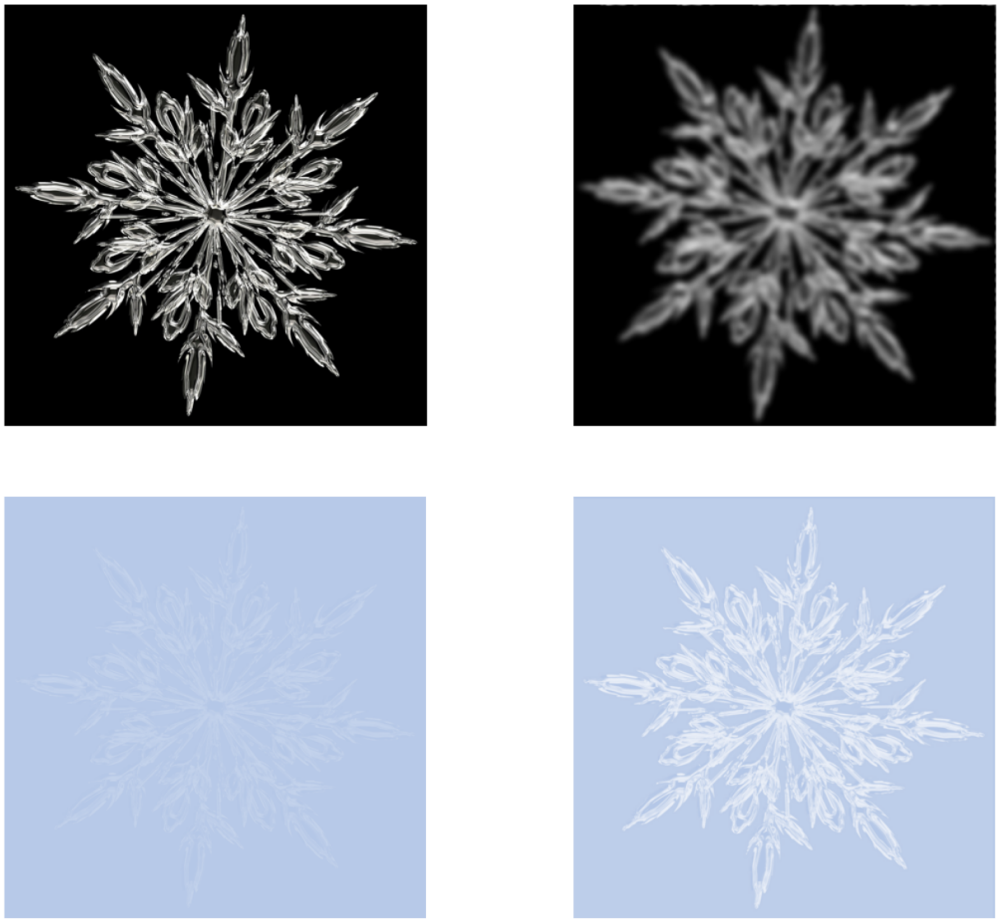
$$= \sqrt{(\tfrac{1}{16}u(x_1)^2 + 2u(x_2)^2 + u(x_3)^2 + 2u(x_4)^2 + 4u(x_5)^2 + 2u(x_6)^2 + u(x_7)^2 + 2u(x_8)^2 + u(x_9)^2}$$

Dies erhalten wir durch die Anwendung von Regel 1 und 2

$$\bar{X} = \frac{1}{16}(x_1 \cdot \frac{1}{u(x_1)} + 2x_2 \cdot \frac{1}{u(x_2)} + x_3 \cdot \frac{1}{u(x_3)} + 2x_4 \cdot \frac{1}{u(x_4)} + 4x_5 \cdot \frac{1}{u(x_5)} +$$

$$2x_6 \cdot \frac{1}{u(x_6)} + x_7 \cdot \frac{1}{u(x_7)} + 2x_8 \cdot \frac{1}{u(x_8)} + x_9 \cdot \frac{1}{u(x_9)})$$

Ergebnis - Gaußfilter



a) b)

Abbildung: Gaußfilter mit Unsicherheiten

Unsicherheiten in der Segmentierung

Bis jetzt: jeder Pixel wurde exakt einer Klasse zugeordnet.
Problem:

1	1	1	1.5	2	2	2
1	1	1	1.5	2	2	2
1	1	1	1.5	2	2	2
1	1	1	1.5	2	2	2

Abbildung: Problem bei bekannten Segmentierungsverfahren

Lösung Fuzzy Segmentation

Unsicherheiten in der Segmentierung

Jeder Pixel enthält eine Wahrscheinlich für jede Klasse in dieser enthalten zu sein.

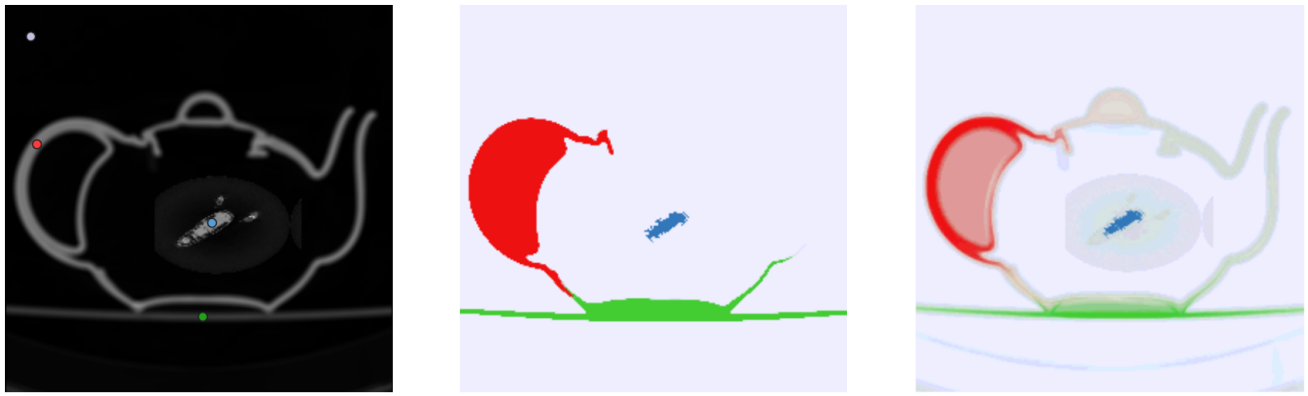


Abbildung: Fuzzy Segmentierung

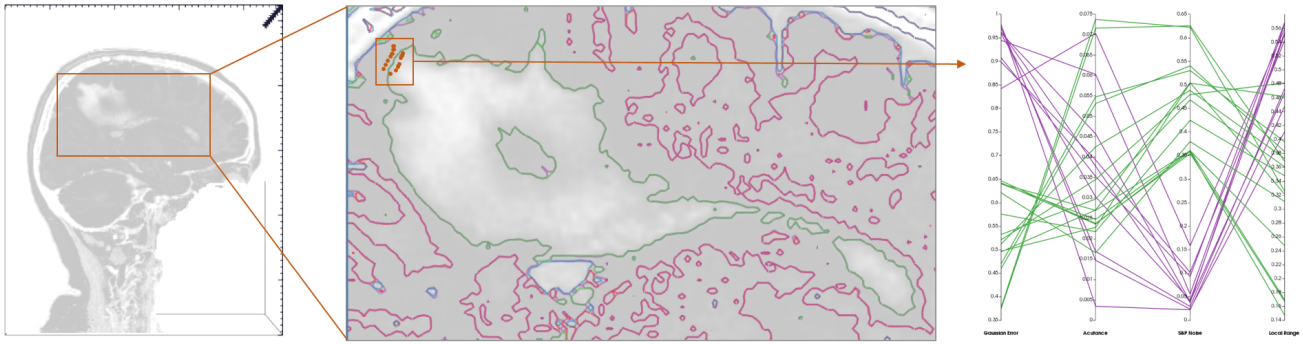


Abbildung: Codierung von Unsicherheiten in der Slice-by-Slice Technik

Unsicherheiten vergrößern sich je mehr Algorithmen man anwendet!

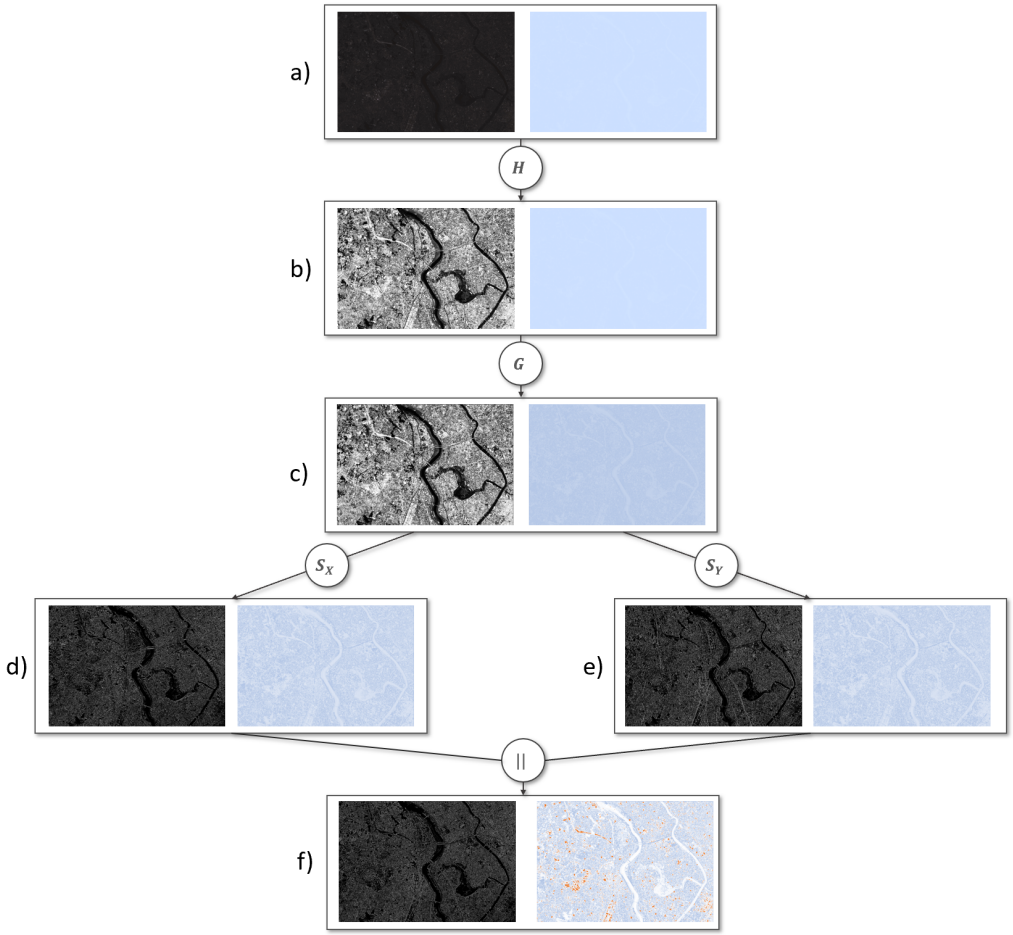


Abbildung: Aufbau einer Pipeline mit verschiedenen Bildverarbeitungsfunktionen

Visuelle kodieren von Abweichungen in der Geometrie

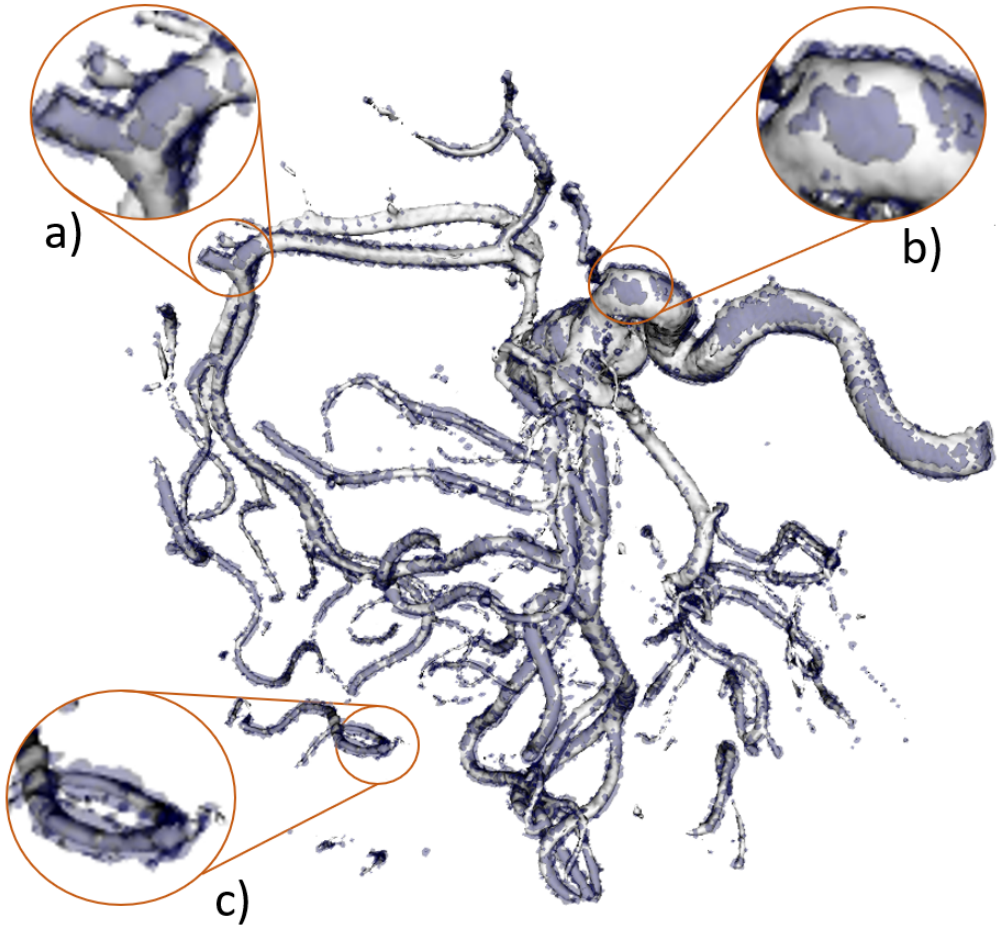


Abbildung: Aneurisma mit Visualisierung von Unsicherheiten

Fazit - Analyse von Unsicherheiten

Vorteile

- Analyse von Unsicherheiten bietet erweitertet Wissen
- Risiken können besser eingeschätzt werden
- Qualität der Daten kann besser eingeschätzt werden

Vorteile

- Mehraufwand in der Berechnung
- Mehraufwand bei der Darstellung
- Umgang muss erlernt werden

Teil XIV

Prüfung und Wiederholung

Bildverarbeitungsprozess

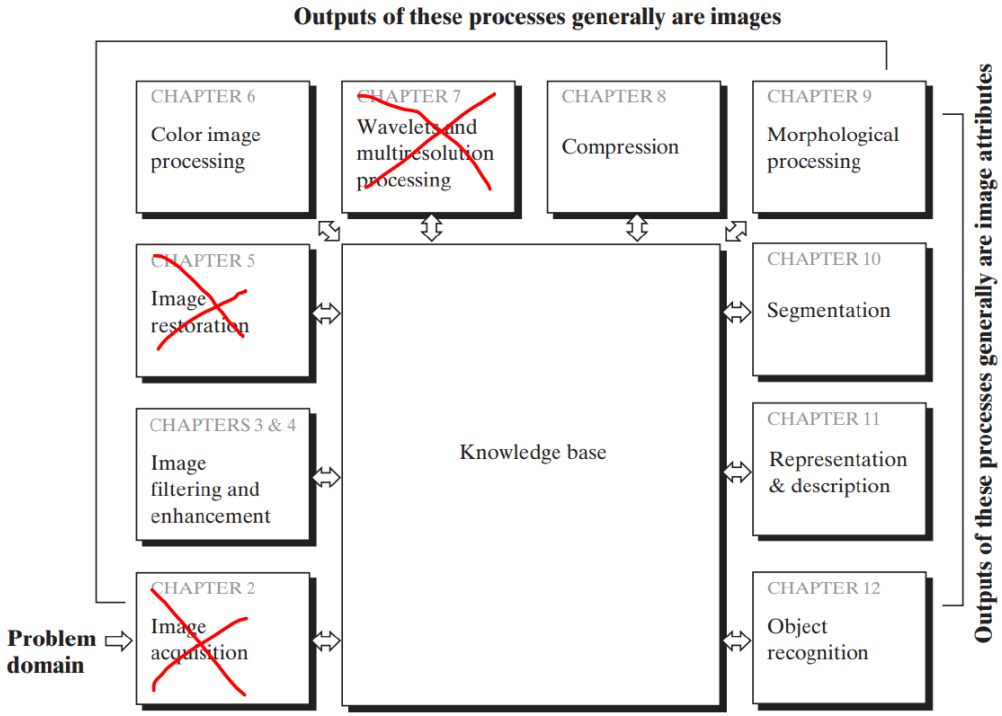


Abbildung: Themen der Vorlesung

Zu jedem Thema sollten sie wissen:

- Wie definiert sich das Thema?
- Was ist das Ziel?
- Welche Algorithmen sind aus der Vorlesung bekannt?
- Wie funktionieren diese Algorithmen?
- Was sind Vor- und Nachteile der Algorithmen?

Sie wählen eins der Themen für die Prüfung

- Dieses Thema teilen sie mir in der Prüfung mit
- Auch hier gelten die Fragen der allgemeinen Kapitel
- Aber: in diesem Gebiet werde ich etwas genauer nachfragen (mit Transferfrage)

Ich werde sie keinen Code schreiben lassen, aber:

- Sie sollten in der Lage sein mir zu erklären wie sie einen Algorithmus umsetzen würden
- Sie sollten wissen was Python in der Bildverarbeitung kann/ nicht kann
- Die Aufgaben die sie bearbeitet haben sind vom reinen Inhalt (Beispiel zusätzliche Filter, Algorithmen ...) Prüfungsrelevant

Wiederholung am 15.07.2020

Bis jetzt habe ich auf der Liste

- Kantenverfolgung
- Momente
- Hit-Miss-Operatoren

Bitte gehen sie nochmal das Skript durch und überlegen welche Dinge noch unklar sind. Am besten überlegen sie sich auch schon ihr Spezialthema und prüfen auch dort ob alles klar ist. Ich werde in der nächsten Übung abfragen ob es noch weitere Themen für die Fragestunde gibt.