

Bildverarbeitung

Thorsten Hansen

Abteilung Allgemeine Psychologie



Überblick

1. Einleitung
2. Bildformate und Digitalisierung
3. Bildstatistik und Histogramm

→ Auswertung der Pixelwerte, die Anordnung der Pixel im Bild spielt keine Rolle.

4. Filter und Faltungsoperation

→ Analyse der Pixelanordnung, die örtliche Verteilung der Pixel ist jetzt wichtig.

5. Fourier-Transformation (Teil II)

1. Einleitung

1. Bildverarbeitung: Einordnung und verwandte Disziplinen
2. Anwendungen
3. Literatur

Einordnung und verwandte Disziplinen

Bildverarbeitung (im engeren Sinne): Bild-zu-Bild Transformation

Computer Vision: Verarbeitung von Bilddaten zur Extraktion von Merkmalen und Beschreibungen (Bildanalyse)

Computer Graphics: „inverse Computer Vision“: Generierung von Bildern aus abstrakten Beschreibungen (Bildsynthese)

Visual Computing: Kombination von Methoden aus CV und CG

Einsatz und Anwendung

- Merkmalsextraktion zur Objekterkennung (Gesichter, Iris, Fingerabdrücke, Verkehrsschilder, ...)
- Bildverbesserung und Diagnose (Medizinische Bilddaten (Röntgen, PET, CT, MRI), Satellitenbilder (Wetter, Fernerkundung, Astrophysik))
- Entfernungsbestimmung und Vermessung
- Robotik (Wegeplanung, Greifen, ...)
- Modellierung biologischer Verarbeitungsprozesse
- Stimulusgenerierung für die visuelle Psychophysik
- et cetera

2. Bildformate und Digitalisierung

1. Mittelwert und Varianz
2. Histogramme
3. Entropie

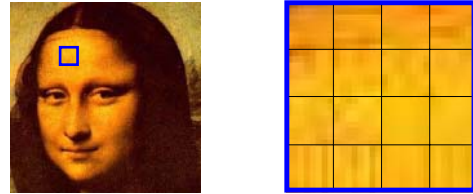
Bildformate

Typische WWW Bildformate sind **ungeeignet!**

- **JPG** gute Bildkompression aber verlustbehaftet, Kachelartefakte

JPG Artefakte

1. Räumlich: Aufteilung des Bildes in Kacheln von 8x8 Pixeln



2. Farbe: Quantisierung, reduzierte Farbpalette

Bildformate

Typische WWW Bildformate sind **ungeeignet!**

- **JPG** gute Bildkompression aber verlustbehaftet, Kachelartefakte
- **GIF** nur 256 Farben

Geeignet ist jedes verlustfreie Bildformat, z.B.

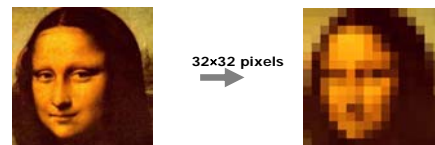
PNG, PBM, TIFF ...

Und was soll ich jetzt nehmen?

Aber Vorsicht: Konvertierung beseitigt Artefakte natürlich nicht (garbage in, garbage out)

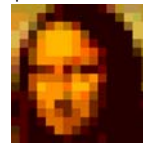
Digitalisierung

1. **Diskretisierung**: quadratische Bildelemente (picture elements, **pixels**)



2. **Quantisierung**: [0,...,255] = 8 Bit pro Farbkanal

hier: 4 bit



Diskretes Grauwertbild

Matrix der Größe $w \times h$ mit Zahlen aus [0,255] (8 Bit)



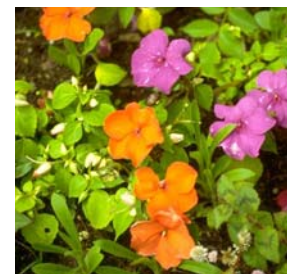
Diskretes Farbbild (true color)

„Dasselbe in grün ... und rot und blau“

Drei Matrizen **R,G,B** (für jeden Farbkanal) der Größe $w \times h$ mit Zahlen aus [0,255] (8 Bit)

Wie viele Farben lassen sich an jedem Pixel darstellen?

$$2^{(8+8+8)} = 16\,777\,216$$



Farbbild (indiziert)

Farbtabelle mit 256 x **R,G,B** Einträgen

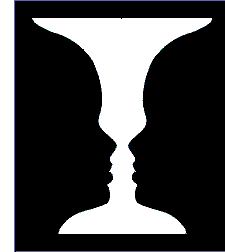
Eine Matrix der Größe $w \times h$ mit Zahlen aus $[0,255]$ (8 Bit) als **Index** in diese Farbtabelle

→ geringerer Speicherbedarf, aber nur 256 darstellbare Farben



Binärbild (schwarz/weiß)

Matrix der Größe $w \times h$ mit Zahlen aus $\{0,255\}$ oder $\{0,1\}$



Datenformate

Eingabebild: Matrix von **ganzen Zahlen** aus $[0,255]$

Interne Berechnung auf **Gleitpunktzahlen**

verlustbehaftete Diskretisierung

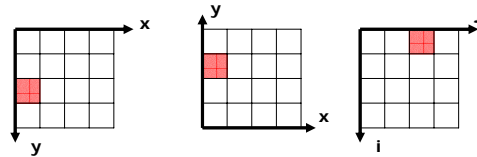
Ausgabebild: Matrix von **ganzen Zahlen** aus $[0,255]$

Koordinatensysteme

Bildverarbeitung

Graph

Matrix



Wo liegt der Pixel (1,3)?

Bildverarbeitung = Grauwertbildverarbeitung?

Die meisten Operationen der Bildverarbeitung arbeiten auf **Grauwertbildern**:

- **Binärbilder** enthalten zuwenig Information und werden meist nicht effizienter gespeichert und verarbeitet als Grauwertbilder
- **Grauwertbilder**: wesentliche Bildstrukturen sind durch Helligkeit definiert
- **Farbbilder**: hohe Kosten der Farbverarbeitung (3x soviel Daten und Verarbeitungsoperationen)

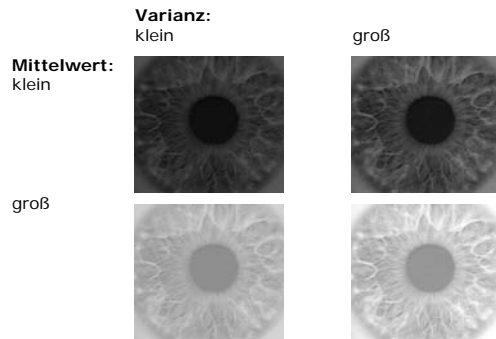
"The use of color in Computer Vision is surprisingly primitive. One difficulty is some legitimate uncertainty about what it is good for."
Forsyth/Ponce (2003)

3. Bildstatistik und Histogramme

1. Mittelwert und Varianz
2. Histogramme
3. Entropie

Auswertung der **Pixelwerte**, die Anordnung der Pixel im Bild spielt keine Rolle.

Mittelwert und Varianz in Bildern



Mittelwert und Varianz

Betrachten des Bildes als **ungeordnete** Menge von Werten, d.h. die Position im Bild spielt keine Rolle:

Mittelwert: mittlere **Helligkeit** im Bild

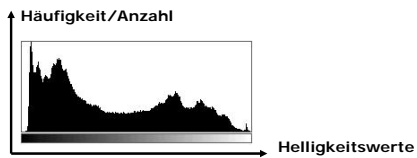
$$\bar{I} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y)$$

Varianz: mittlerer **Kontrast** im Bild

$$\sigma_I^2 = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (I(x, y) - \bar{I})^2$$

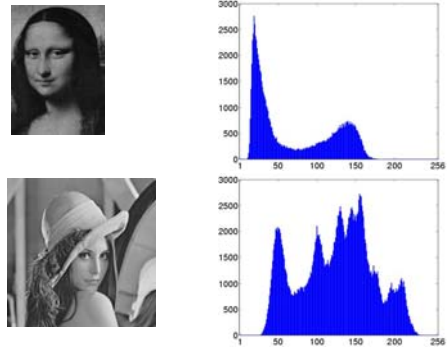
Histogramm

Darstellung der Häufigkeit aller Werte im Bild geordnet nach Helligkeit; wieder spielt die Position keine Rolle

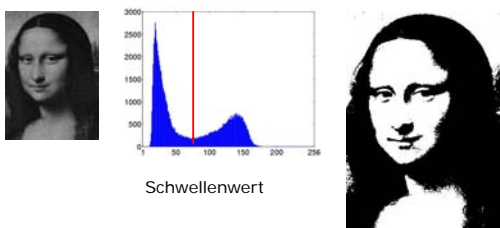


- oft **relative** Häufigkeit: Teilen durch die Pixelanzahl (Normierung)
- bei **nicht-diskreten Bilddaten**: Aufteilen des Wertebereichs in (aquidistante) Bereiche (*bins*)

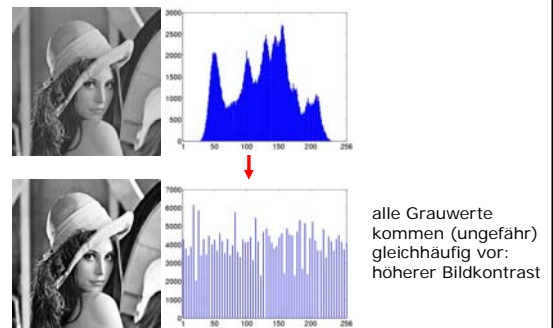
Histogramme: Beispiel



Histogramme zur Schwellenwertbestimmung



Histogrammeinebnung (*histogram equalization*)

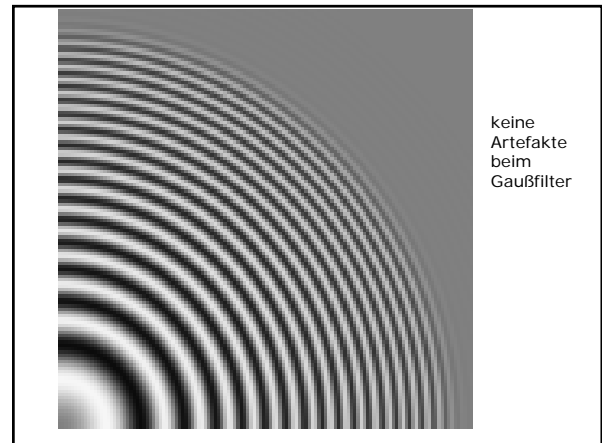
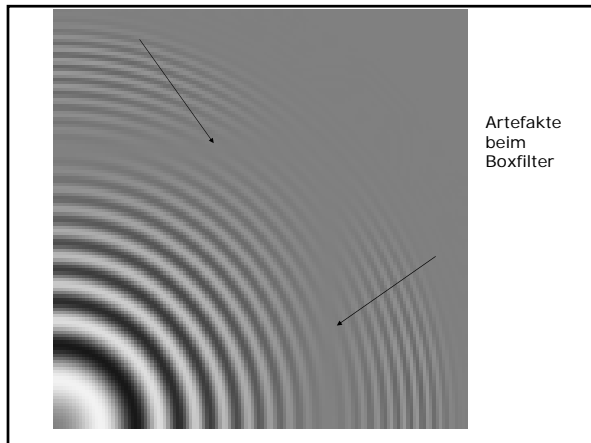
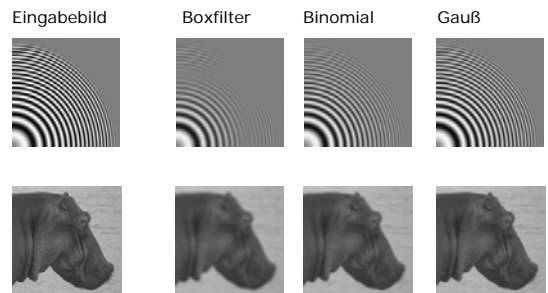


4. Filter und Faltungsoperation

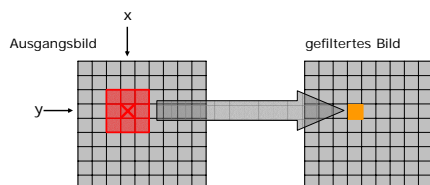
1. Filteroperation
2. Glättungsfilter
3. Kontrastdetektion
4. Kantendetektion

Analyse der Pixel**anordnung**, die örtliche Verteilung der Pixel ist jetzt wichtig.

Glättungsfilter: verschiedene 5x5 Filter

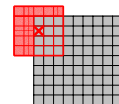


Filteroperation anschaulich



Für jeden Pixel (x,y) : Bestimmung des neuen Wertes (■) als Summe der Produkte von Maske und Bild

Randbehandlung



Verschiedene Randbehandlungen sind möglich:

- Ignorieren: das Bild wird dann mit jeder Faltung kleiner
- Auffüllen mit Nullen: Starke Randeffekte (Dirichletsche Randbedingung)
- Auffüllen mit den Randwerten (Neumannsche Randbedingung, 1. Ableitung = 0)
- Zirkuläre Fortsetzung (Bild als periodische Struktur; motiviert durch Fourier-Transformation)
- et cetera

Filteroperation

Filtern ist eine gewichtete lokale Mittelung: der neue Wert an der Position (x,y) ist die gewichteten Summe der Bildwerte in einer lokalen Nachbarschaft:

$$I_{\text{new}}(x, y) = \sum_i \sum_j M(i, j) I(x+i, y+j)$$

Die Gewichtsmatrix $M(i,j)$ heißt Filter, Maske oder *kernel*.

Die Filteroperation wird für jeden Bildpunkt durchgeführt.

Für ein Bild der Größe 512x512 und eine Maske der Größe 15x15 sind das ca. 60 Millionen Multiplikationen.

Korrelation und Konvolution

Bisher: **Korrelation** des Bildes mit der Filtermaske:

$$I \oplus M = \sum_i \sum_j M(i, j) I(x+i, y+j)$$

Verbreitet: **Konvolution (Faltung)** von Bildes und Maske:

$$I * M = \sum_i \sum_j M(i, j) I(x-i, y-j)$$

Die Konvolution entspricht der Korrelation mit einer punktgespiegelten Filtermaske: für punktsymmetrische Masken sind daher beide Operation identisch.

Konvolution und lineare Systeme

System S: Antwort auf eine Eingabe



Lineares System (*linear shift-invariant system, LSI system*):

- Superposition und Skalierung: $S(ax+by) = aS(x) + bS(y)$
- Verschiebungsinvarianz: $S(\text{shift}(x)) = \text{shift}(S(x))$

Konvolution und lineare Systeme

Ein lineares System läßt sich durch die Antwort auf ein bestimmtes Signal, den Einheitsimpuls oder δ -Impuls, vollständig beschreiben.

Die Systemantwort eines Systems S auf einen δ -Impuls sei die **Impulsantwort M** (*point spread function*):

$$S(\delta) = M$$

Dann ergibt sich die Systemantwort auf eine beliebige Eingabe x durch Faltung mit der Impulsantwort M:

$$S(x) = M * x$$

Filtern ist dann die Antwort eines linearen Systems auf ein Eingabebild.

Faltungsalgebra

- Assoziativität $f*(g*h) = (f*g)*h$
- Distributivität $f*(g+h) = f*g + f*h$
- Kommutativität $f*g = g*f$

Wann soll man ein Bild glätten: vor oder nach der Kantendetektion?

Das ist egal (Assoziativität): Erst Bild I glätten mit Filter G, dann Kantendetektion mit Kantensfilter K: $K*(G*I)$

Umgekehrt: $G*(K*I)$

Filtertypen

- **linear**
Glättungsfilter (Tiefpass, „blurring“, „smoothing“,)
Kontrastdetektion (Hochpass)
Kantendetektion (de facto Standard: Sobel-Operator)
- **nichtlinear**
Median
Kantendetektion (Canny)

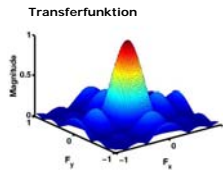
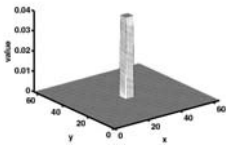
Boxfilter: lokaler Mittelwert

Matrix der Größe N×N mit Werten 1/N

N=5:

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

1/25



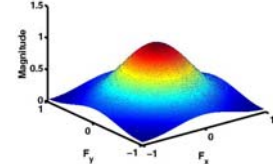
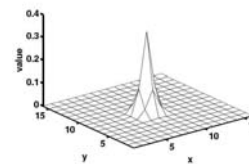
Gaußfilter: gaußgewichtete Mittelung

Filterwerte ergeben sich aus Gaußfunktion (=Normalverteilung)
2D Gaußfunktion: bivariate unkorrelierte NV

N=5 (reelle Filterwerte): $g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$

$G(x, y) = g(x)g(y)$

separierbar in x,y



Medianfilter: nichtlinear !

Einfache Möglichkeit zur Beseitigung lokaler Störungen:
Ersetzen eines Bildpunktes durch den Median der Bildwerte
in einem N×N Fenster

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	150
111	116	110	120	130

- Sortierte Liste der Bildwerte in N×N Fenster (hier: N=3)
115, 119, 120, 123, 124
125, 126, 127, 150
- Ersetzen des zentralen Pixels durch den Wert in der Mitte der Liste (=Median)

124

4.3. Kontrast- und Kantendetektion

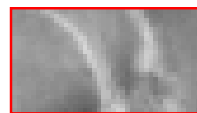
- Finite Differenzen
- Sobel-Operator
- DoG und LoG
- Canny
- Gabor-Filter

Kantendetektion (edge detection)

... ist nur für uns einfach!

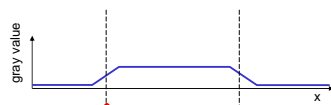


Lokale Helligkeitsänderung und Objektkanten



Kanten und 1./2. Ableitung

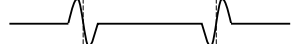
1D Schnitt durch ein Bild



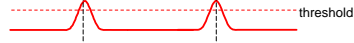
1. Ableitung $f'(x)$



2. Ableitung $f''(x)$



Betrag $|f'(x)|$



Kanten:
 $|f'(x)| > \text{threshold}$
 oder $f''(x) = 0$



Finite Differenzen

Approximation der Ableitung durch finite Differenzen

$$\frac{\partial}{\partial x} u \approx \frac{u(x) - u(x-h)}{h}$$

Diskrete Maske $\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$

Problem: Sehr anfällig für lokale Störungen

Verbesserte Detektion durch Glätten

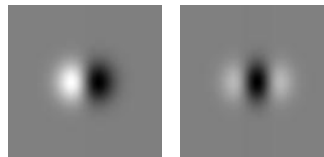
Sobel-Operator

Finite Differenz mit Glättung senkrecht zur Ableitungsrichtung



$$|G| = \sqrt{G_x^2 + G_y^2}$$

Gaußableitungen



1. Ableitung

2. Ableitung

LoG and DoG

Laplace-Operator
 Laplace der Gaußfunktion: LoG

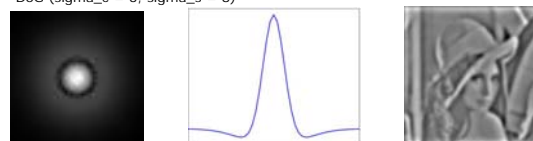
Approximation durch Differenz von Gaußfunktion mit unterschiedlicher Varianz: DoG

LoG and DoG: Beispiele

LoG (sigma = 8)



DoG (sigma_c = 3, sigma_s = 8)



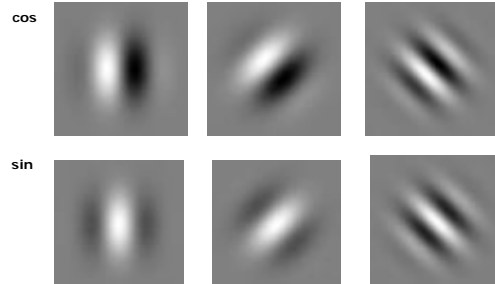
Filter und visuelle Verarbeitung

DoG: Einfaches lineares Model retinaler Ganglienzellen

Enroth-Cugell & Robson (1966), The contrast sensitivity of retinal ganglion cells of the cat. *J. Physiol.*, 187(6):517–522.

Gaborfilter

cos und sin Funktionen gewichtet mit Gaußfenster



Gaborfilter und lokale Energie

sin und cos Versionen des Gaborfilters haben einen Phasenunterschied von $\pi/2$ und bilden ein sogenanntes **Quadraturfilterpaar**.

Addierte man die quadrierten Filterantworten eines Quadraturfilterpaares, ergibt sich eine Funktion, die **phasenunabhängig und proportional zur Energie** ist.

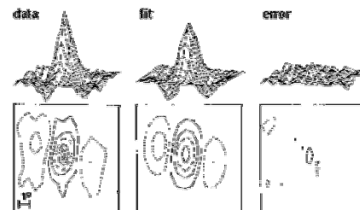
Die lokale Energie für eine bestimmte Frequenz ergibt sich damit als Summe der quadrierten Filterantworten gerader (cos) und ungerader (sin) Gaborfilter:

$$E = \sqrt{(G_{\sin} * I)^2 + (G_{\cos} * I)^2}$$

Morrone & Burr (1988). Feature detection in human vision: A phase-dependent energy model. *Proc. Roy. Soc. Lond. (B)* 235:221–245.

Gaborfilter und visuelle Verarbeitung

Gaborfilter als einfaches lineares Model kortikaler simple cells



Jones & Palmer (1987), *J. Neurophysiol.* 58(6):1233–1256.

Filter—Bemerkungen

- **Filter als Template matching**

Die größte Filterantwort ist dort, wo die Bildstruktur dem Filter am ähnlichsten ist (Filtern als inneres Produkt von Bild und Maske)

- **Filter und lineare Systeme**

Ein Filter M ist die Ausgabe eines linearen Systems auf eine lokalen Impuls als Eingabe; die Ausgabe auf eine beliebige Eingabe I ergibt sich durch Faltung mit der Maske M

- **Filter als rezeptive Felder (RFs)**

RFs als Punktantwort eines (angenähert) linearen Systems. RFs beschreiben den Stimulus, durch den ein Neuron am effektivsten erregt wird.

Effiziente Filterung

- Separierbarkeit

- Symmetrie

- Faltungstheorem

Faltung in Ortsbereich entspricht

Multiplikation im Frequenzbereich:

Anstatt ein Bild mit einer Maske zu falten, kann man die Fourier-Transformation von Bild und Maske multiplizieren und invers fouriertransformieren:

$$I * M = F^{-1}(F(I)F(M))$$

Bei großen Masken und einer schnellen Implementation der Fourier-Transformation und ihrer Inversen ist dies schneller als die Faltung.

Literatur

Einstieg:

- Mallot: Sehen und die Verarbeitung visueller Information Vieweg, Braunschweig, 1998.

Detailliert:

- Jähne: Bildverarbeitung. Springer, Berlin, 1997
- Gonzales, Woods: Digital Image Processing. Addison-Wesley, Reading, MA, 1986.
- Forsyth/Ponce: Computer Vision-A Modern Approach Prentice-Hall, Upper Saddle River, NJ, 2003.

Klassiker:

- Marr: Vision. Freeman, San Francisco, CA, 1982.

Bildverarbeitung und Fourier-Transformation

Thorsten Hansen

Fouriertransformation



Französischer Physiker
und Mathematiker

Erfinder der
Fouriertransformation.

Jean Baptiste Joseph Fourier
1768–1830

Fouriertransformation

Transformation

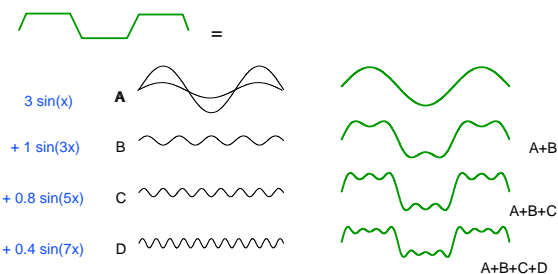
Darstellung einer Größe in einem anderen
Bezugssystem, z.B. Koordinatentransformation oder
Farbtransformation RGB ↔ HSV.

Bei einer Transformation gewinnt man keine neuen
Daten, sondern stellt die vorhandenen Daten anders
dar, so dass bestimmte Operationen einfacher werden.
Beispiel: RGB: Monitordarstellung; HSV: Farbauswahl

Transformation lassen sich wieder rückgängig machen.

Fourier-Analyse

Darstellung einer periodischen Funktion
durch **cos** und **sin** Funktionen unterschiedlicher Frequenzen



Fourier's Theorem

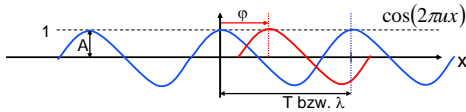
Jede beliebige periodische Funktion
läßt sich darstellen
als **Summe von cos** und **sin** Funktionen
unterschiedlicher Frequenzen.

1. Ist die Funktion nicht periodisch und auf einen bestimmten Definitionsbereich beschränkt, kann man diesen Bereich einfach kopieren (periodisch fortsetzen) und hat damit wieder eine periodische Funktion.
2. Ein Bild kann man als Zeilen und Spalten von nicht periodischen Funktionen auffassen. Man kann also auch ein Bild fouriertransformieren.

Periodische Funktionen

Parameter

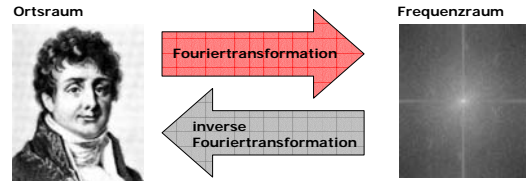
- A Amplitude: Intensität des Signals
 ϕ Phase: Verschiebung gegenüber dem Ursprung
 verschiedene Größen zur Beschreibung der "Frequenz" [Einheit]
- | | |
|--|--------------------------------------|
| zeitlich $f(t)$ | räumlich $f(x)$ |
| T Periodendauer [s] | λ Wellenlänge [m] |
| f Frequenz $f = 1/T$ [1/s=Hz] | f Raumfrequenz $f = 1/\lambda$ [1/m] |
| ω Kreisfrequenz $\omega = 2\pi f$ | k Wellenzahl $k = 2\pi/\lambda$ |



Fouriertransformation

Bisher: Darstellung des Bildes im **Ortraum** durch den Grauwert an einem bestimmten Ort
 Jetzt: Darstellung im **Frequenzraum** durch die Ähnlichkeit mit cos und sin Funktionen verschiedener Frequenzen

Ein Bild kann nun eindeutig und vollständig in beiden Räumen dargestellt werden. Die Transformation zwischen beiden Räumen hat *Jean Baptiste Joseph Fourier* zuerst beschrieben.



Anwendungen der FT

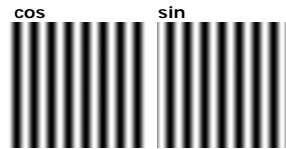
- Analyse von Filtermasken
- effiziente Faltung
- Analyse und Manipulation der Bildfrequenzen
- Datenkompression (z.B. JPEG, MPEG, MP3)

Fouriertransformation anschaulich

Welche periodischen Strukturen gibt es im Bild?



Beispiel:
 vertikale Strukturen die **8x** im Bild vorkommen



Nach dem Fouriertheorem benötigen wir **sin** und **cos** Funktionen zur vollständigen Analyse.

Fouriertransformation anschaulich

cos:

$$\sum_{(x,y)} \left(\text{Image} \cdot \cos \right) = 0.24$$

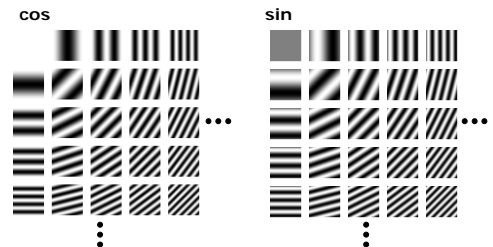
z.B. (reelle Zahl)

... und dasselbe für sin

Bei der Fouriertransformation wird das Bild auf das **cos** und **sin** Muster projiziert:
 Jeder Bild und Muster werden punktwise multipliziert (\cdot), und die Produkte werden summiert (\sum):
 man erhält zwei reelle Zahlen welche die Ähnlichkeit des Bildes mit dem Musterpaar beschreibt.

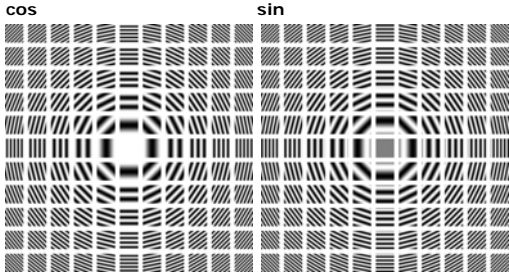
Fouriertransformation anschaulich

Bei der Fouriertransformation wird die Projektion für alle möglichen Frequenzen und Orientierungen berechnet...



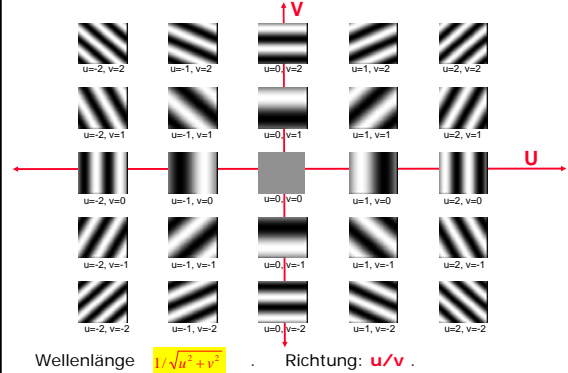
Fouriertransformation anschaulich

... bei einem Bild der Größe NxN sind das 2xNxN Basisbilder, auf die das Ausgangsbild projiziert wird.



2D Basisfunktionen

$$e^{2\pi i(ux+vy)}$$



Satz von Euler

Die zwei Komponenten **cos** und **sin** lassen sich durch eine **komplexe Zahl z** darstellen:

$$z = \exp(i\varphi) = \cos(\varphi) + i\sin(\varphi), \quad \text{mit } i = \sqrt{-1}$$

Das ist zwar etwas abstrakter, erleichtert aber das Rechnen und spart Schreibarbeit.

Diskrete Fouriertransformation

Bild der Größe MxN

Diskrete Fouriertransformation

$$G(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g(x, y) \exp(-i2\pi (\frac{ux}{M} + \frac{vy}{N}))$$

mit $u = 0, 1, \dots, M-1$ und $v = 0, 1, \dots, N-1$

Rücktransformation

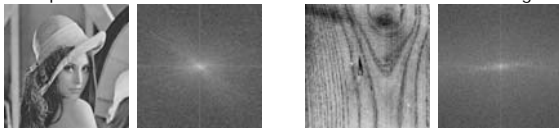
$$g(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} G(u, v) \exp(i2\pi (\frac{ux}{M} + \frac{vy}{N}))$$

Die Rücktransformation unterscheidet sich nur durch den Normierungsfaktor und das Vorzeichen.

FT: Beispielspektren

Kompression des Amplitudenspektrum zur Darstellung $\log(|G(u, v)| + 1)$

Die Spektren natürlicher Bilder unterscheiden sich visuell wenig:

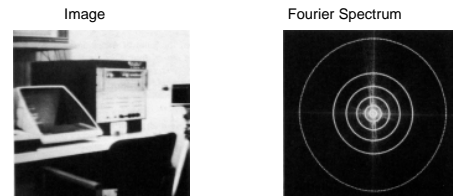


Spektren synthetischer Bilder sind auch visuell unterschiedlich:



Gonzales&Woods (1993)

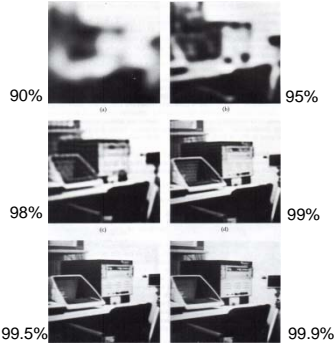
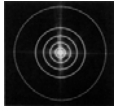
Frequenzbänder



Percentage of image power enclosed in circles (small to large) :

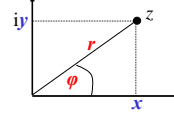
90%, 95%, 98%, 99%, 99.5%, 99.9%

Tiefpassfilterung



Komplexe Zahlen

Zwei Darstellungsweisen einer komplexen Zahl z :



kartesisch: $z = (x+iy)$

polar: $z = r \cos(\varphi) + r i \sin(\varphi) = r \exp(i\varphi)$

Amplitude $r = \sqrt{x^2 + y^2}$

Phase $\varphi = \text{atan2}(y, x)$

Phasen- und Amplitudenspektrum

Darstellung der komplexwertigen Fouriertransformation durch **Amplituden-** und **Phasenspektrum**:

Amplitudenspektrum

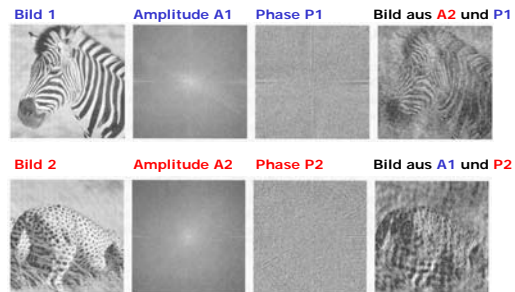
$$|G(u,v)| = \sqrt{G(u,v)G(u,v)^*}, \text{ mit } z = (x+iy)^* = (x-iy)$$

Phasenspektrum

$$\text{ang}(G(u,v)) = \text{atan2}(\text{Imag}(G(u,v)), \text{Real}(G(u,v)))$$

Phasen- und Amplitudenspektrum

Für die Wahrnehmung ist das **Phasenspektrum** wichtiger:



Forsyth/Ponce (2003), p. 149

Fast Fourier Transform (FFT)

Effiziente Implementierung der diskreten FT durch divide and conquer Algorithmus

Idee

1. Rekursive Aufteilung der Summe in gerade und ungerade Komponenten
2. ... bis nur noch eine Zahl zu transformieren ist; dies ist einfach die Zahl selbst
3. rekursive Kombination der Terme

Erster FFT Algorithmus von Cooley&Tuckey 1965.
Reduktion der Zeitkomplexität von N^2 auf $N \log N$.

Fensterfunktionen

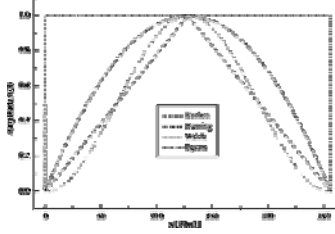
Um Leckeffekte (leakage effects) zu vermeiden, muss das Bild mit einer Fensterfunktion multipliziert werden:



Wir erinnern uns: Ein Bild ist keine periodische Funktion, Sondern muss zur FT periodisch fortgesetzt werden.

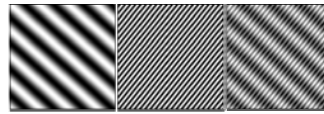
Der Leckeffekt ist eine Auswirkung der Multiplikation mit einem Rechteckfenster zur Darstellung des Bildes aus einer (gedachten) periodischen Struktur.

Verschiedene Fensterfunktionen

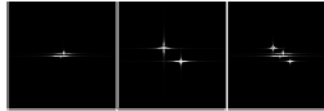


Hanning Fenster: $A(f) = 2 \left[\frac{1 - \cos\left(\frac{2\pi f}{N}\right)}{2} \right]$ [3.18]
 Hamming Fenster: $A(f) = \frac{1}{2} \left[1 + \cos\left(\frac{2\pi f}{N}\right) \right]$ [3.20]
 Blackman Fenster: $A(f) = 1 - \frac{1}{2} \left[\frac{1 - \cos\left(\frac{2\pi f}{N}\right)}{2} \right]^2$ [3.21]

Fensterfunktion: Beispiel



Eingebild

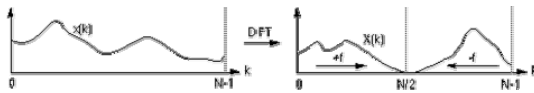


ohne Fensterfunktion



mit Fensterfunktion

Nyquistfrequenz



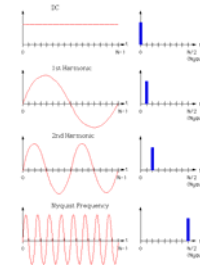
Die Nyquistfrequenz ist die höchste Frequenz des kontinuierlichen Signals, die im diskreten Signal dargestellt werden kann.

Wenn das Signal keine höheren Frequenzen enthält, kann das Ursprungssignal **exakt** aus den abgetasteten Signalen rekonstruiert werden.

Aber: Enthält das kontinuierliche Signale höhere Frequenzen, führt dies zu Fehlern (alias Effekte).

Nyquistfrequenz

Bei N Bildpunkten (Abtastpunkten) ist die Nyquistfrequenz $N/2$.



Kontinuierliche Fouriertransformation

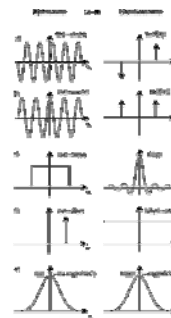
Kontinuierliche Fouriertransformation

$$S(\omega) = \int_{-\infty}^{\infty} s(t) \exp(-i\omega t) dt \quad \text{mit } \omega = 2\pi f$$

Rücktransformation

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) \exp(i\omega t) d\omega$$

Transformationspaare



Wellenfunktion	Transformierte
$f(t) = \delta(t)$	$F(\omega) = 1$
$f(t) = \begin{cases} 1 & t \leq T \\ 0 & t > T \end{cases}$	$F(\omega) = 2T \text{sinc}\left(\frac{\omega T}{2}\right)$
$f(t) = \exp(-a t)$	$F(\omega) = \frac{2a}{\omega^2 + a^2}$
$f(t) = \exp(-a t) \cos(\omega_0 t)$	$F(\omega) = \frac{a}{\omega^2 - \omega_0^2 + a^2} + \frac{a}{\omega^2 + \omega_0^2 + a^2}$
$f(t) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{t^2}{2\sigma^2})$	$F(\omega) = \exp(-\frac{\omega^2 \sigma^2}{2})$