Multimediale Signalverarbeitung Bildverarbeitung: Filter

Thorsten Thormählen

19. Juni 2020

Teil 6, Kapitel 1

Dies ist die Druck-Ansicht.

Aktiviere Präsentationsansicht

Steuerungstasten

→ nächste Folie (auch Enter oder Spacebar).
← vorherige Folie
d schaltet das Zeichnen auf Folien ein/aus
p wechselt zwischen Druck- und Präsentationsansicht
CTRL + vergrößert die Folien
CTRL - verkleinert die Folien
CTRL 0 setzt die Größenänderung zurück

Typesetting math: 100%

Notation

Typ	Schriftart	Beispiele
Variablen (Skalare)	kursiv	a, b, x, y
Funktionen	aufrecht	$\mathrm{f}, \mathrm{g}(x), \max(x), \mathrm{h}[n]$
Vektoren	fett, Elemente zeilenweise	$\mathbf{a}, \mathbf{b} = \begin{pmatrix} x \\ y \end{pmatrix} = (x, y)^\top$, $\mathbf{B} = (x, y, z)^\top$
Matrizen	Schreibmaschine	$\mathtt{A}, \mathtt{B} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$
Mengen	kalligrafisch	$\mathcal{A}, B = \{a, b\}, b \in \mathcal{B}$
Zahlenbereiche, Koordinatenräume	doppelt gestrichen	$\mathbb{N}, \mathbb{Z}, \mathbb{R}^2, \mathbb{R}^3$

Typesetting math: 100%

Von 1D nach 2D

Bei der bisherigen Betrachtung von zeitdiskreten Signalen wurden schon viele Grundlagen der digitalen Signalverarbeitung kennengelernt: Abtasttheorem, Faltung, Tiefpass, Hochpass, etc.

Im Prinzip handelt es sich bei Bildern auch im diskrete Signale, nur jetzt:

- Ortsdiskret statt zeitdiskret
- Zwei Dimensionen $\mathrm{f}[x, y]$ statt nur einer $\mathrm{f}[n]$

Faltung in 2D

Die zeitdiskrete Faltung zweier Signale $\mathrm{a}[n]$ und $\mathrm{b}[n]$ ist definiert durch:

$$\mathrm{f}[n] = \mathrm{a}[n] * \mathrm{b}[n] = \sum_{k=-\infty}^{\infty} \mathrm{a}[k]\ \mathrm{b}[n-k]$$

Analog kann die Faltung in 2D definiert werden:

$$\mathrm{f}[x,y] = \mathrm{a}[x,y] * \mathrm{b}[x,y] = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \mathrm{a}[j,k]\ \mathrm{b}[x-j, y-k]$$

Thorsten Thormählen 5 / 48

Typesetting math: 100%

Faltung von Bildern

2D Faltung:

$$\mathrm{f}[x,y] = \mathrm{a}[x,y] * \mathrm{b}[x,y] = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \mathrm{a}[j,k]\ \mathrm{b}[x-j, y-k]$$

Als Beispiel soll die Faltung zwischen einem Bild $\mathrm{a}[x,y]$ der Größe 4 x 3 Pixel mit einem so genannten Kernel $\mathrm{b}[x,y]$ der Größe 3 x 3 Pixel betrachtet werden:

$a[x,y]$

$y \backslash x$	0	1	2	3
0	1	2	2	2
1	9	9	8	7
2	1	2	1	6

$b[x,y]$

$y \backslash x$	-1	0	1
-1	1	2	3
0	0	0	0
1	-1	-2	-3

Außerhalb des definierten Bereichs sind alle Werte des Kernels $\mathrm{b}[x,y]$ gleich null, daher

$$\mathrm{f}[x,y] = \mathrm{a}[x,y] * \mathrm{b}[x,y] = \sum_{j=x-1}^{x+1} \sum_{k=y-1}^{y+1} \mathrm{a}[j,k]\ \mathrm{b}[x-j, y-k]$$

Typesetting math: 100%

Faltung von Bildern

2D Faltung für das Beispiel:

$$\mathrm{f}[x,y] = \mathrm{a}[x,y] * \mathrm{b}[x,y] = \sum_{j=x-1}^{x+1} \sum_{k=y-1}^{y+1} \mathrm{a}[j,k]\ \mathrm{b}[x-j, y-k]$$

$y \backslash x$	0	1	2	3
0	1	2	2	2
1	9	9	8	7
2	1	2	1	6

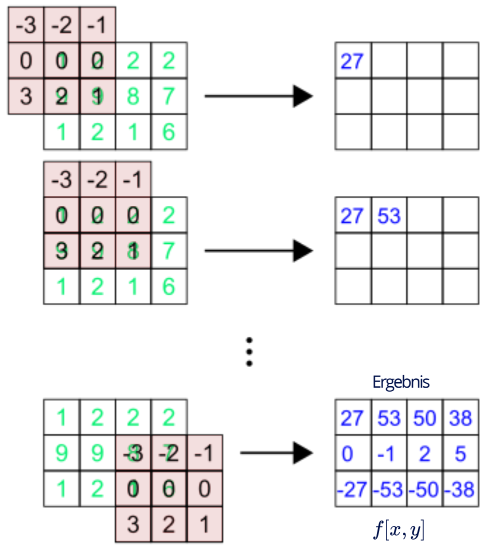
Bild
$a[x,y]$

$y \backslash x$	−1	0	1
−1	1	2	3
0	0	0	0
1	-1	-2	-3

Kern
$b[x,y]$

$y \backslash x$	1	0	−1
1	-3	-2	-1
0	0	0	0
−1	3	2	1

Gespiegelter Kern



Ergebnis

27	53	50	38
0	-1	2	5
-27	-53	-50	-38

$f[x,y]$

Typesetting math: 100%

Filterkerne

Im Folgenden soll zunächst mit verschiedenen Filterkernen $\mathrm{b}[x, y]$ experimentiert werden

Später wird genauer auf die Theorie eingegangen

Hinweis: Oft wird in der Literatur direkt der gespiegelter Kern angegeben. Im Folgenden wird hier immer der mathematisch korrekte Faltungsoperator $b[x, y]$ dargestellt (für symmetrische Kerne ist es natürlich egal)

Filterkerne

Welchen Effekt hat der Filterkern?

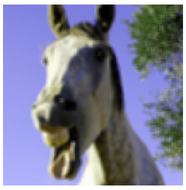
$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Antwort:

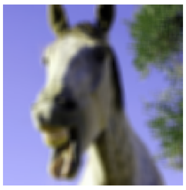
Gleitender Mittelwert (Mittelwertfilter)

Auch eine Art Tiefpass (aber lässt auch hohe Frequenzen durch, da Sinc-Funktion im Frequenzbereich)

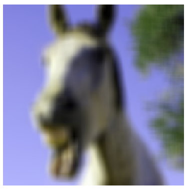
Original



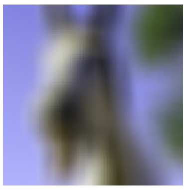
3 x 3



5 x 5



7 x 7



21 x 21

Typesetting math: 100%

Filterkerne

Welchen Effekt hat der Filterkern?

$$\frac{1}{3}\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Antwort:

Mittelwertfilter in y-Richtung

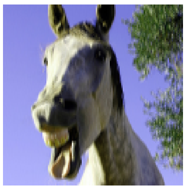
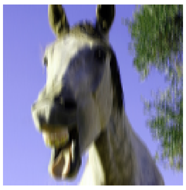
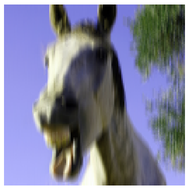
Mittelwertfilter in x-Richtung entsprechend:

$$\frac{1}{3}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

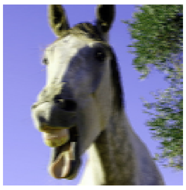
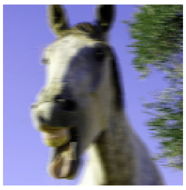
Eigentlich reicht ein Filter mit der Größe 1 x 3 Pixel bzw. 3 x 1 Pixel aus

Filterkerne

Mittelwertfilter in y-Richtung

				
Original	3 x 1	5 x 1	7 x 1	21 x 1

Mittelwertfilter in x-Richtung

				
Original	1 x 3	1 x 5	1 x 7	1 x 21

Typesetting math: 100%

Filterkerne

Welchen Effekt hat der Filterkern?

$$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Antwort:

- Gaußscher Glättungsfilter:
  $$\mathrm{b}[x,y] = \exp\left(-\frac{x^2+y^2}{2\,\sigma^2}\right)$$
  und Normierung der Werte, so dass die Summe gleich 1 ist
- Tiefpass (auch Gauss-Funktion im Frequenzbereich)
- Hier verwendet: Integer-Approximation durch Binomial-Koeffizienten
- Kleine Version: $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

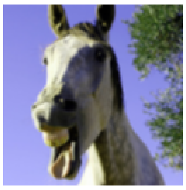
Thorsten Thormählen 12 / 48

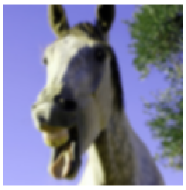
Typesetting math: 100%

Filterkerne

Filterung mit Binomial-Filterkernen verschiedener Größe:


Original


3 x 3


5 x 5


7 x 7


21 x 21

Filterkerne

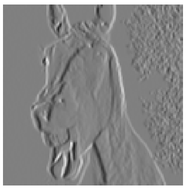
Welchen Effekt hat der Filterkern?

$$\frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Antwort:

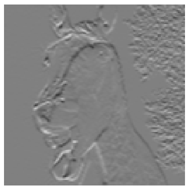
Kantenverstärkung in x-Richtung

Original



x-Richtung

$$\frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$



y-Richtung

$$\frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Typesetting math: 100%

Filterkerne

Welchen Effekt hat der Filterkern?

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Antwort:

Laplace-Filter (Ableitung 2. Ordnung)

Kantenverstärkung in x-Richtung und y-Richtung

Separierbare Filter

Separierbare Filter

Bei einem separierbare Filterkern kann die Faltung in mehrere Faltungen aufgeteilt werden, d.h.

$$\mathrm{f}[x,y] = \mathrm{a}[x,y] * \mathrm{b}[x,y] = (\mathrm{a}[x,y] * \mathrm{b}_1[x,y]) * \mathrm{b}_2[x,y]$$

Dabei sind die separierten Filterkerne $\mathrm{b}_1[x,y]$ und $\mathrm{b}_2[x,y]$ kleiner und daher schneller ausführbar

Die meisten der bisher betrachteten Filterkerne sind x/y-separierbar, beispielsweise

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

Für einen $K \times K$ großen Filterkern reduziert sich der Rechnenaufwand durch die Separation von K^2 auf $2\,K$

Typesetting math: 100%

Separierbare Filter

Im Allgemeinen ergibt sich der kombinierte Filterkern der Größe $K \times K$ aus den x/y-separierten Kernen der Größe K durch

$$\frac{1}{w_y}\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{bmatrix} * \frac{1}{w_x}\begin{bmatrix} x_1 & x_2 & \dots & x_K \end{bmatrix} = \frac{1}{w_y w_x}\begin{bmatrix} y_1 x_1 & y_1 x_2 & \dots & y_1 x_K \\ y_2 x_1 & y_2 x_2 & \dots & y_2 x_K \\ \vdots & \vdots & \ddots & \vdots \\ y_K x_1 & y_K x_2 & \dots & y_K x_K \end{bmatrix}$$

Wie kann dieser Gauss-Filter separiert werden?

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Antwort:

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Typesetting math: 100%

Separierbare Filter

Wie kann dieser Filter, der auf Kanten in x-Richtung anspricht, separiert werden?

$$\frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Antwort:

$$\frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \frac{1}{2}\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

Interessant. Es handelt sich offenbar um eine Kombination aus einem Gauss-Filter in y-Richtung und einem Kantenfilter in x-Richtung

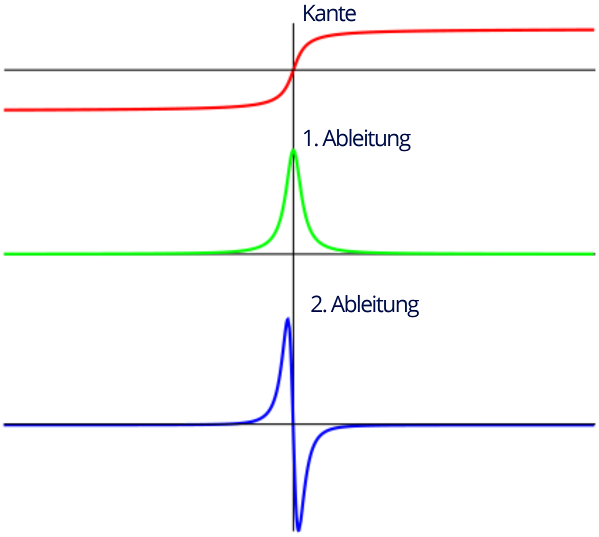
Typesetting math: 100%

Kantenfilter

Kantenfilter

Zur Kantendetektion gibt es zwei Möglichkeiten:

1. Ableitung berechnen: Je höher die Absolutwerte, desto steiler die Kante
2. Ableitung berechnen: Nulldurchgang bei Kante



Typesetting math: 100%

1. Ableitung berechnen

Die partiellen Ableitung $\partial \mathrm{f}/\partial x$ und $\partial \mathrm{f}/\partial y$ können bei Bildern im Allgemeinen nicht über den Differentialquotient bestimmt werden, da die Funktion f nicht analytisch vorliegt:

$$\frac{\partial \mathrm{f}}{\partial x} = \lim_{x_1 \to x_0} \frac{\partial \mathrm{f}(x_1) - \mathrm{f}(x_0)}{x_1 - x_0} \quad \text{und} \quad \frac{\partial \mathrm{f}}{\partial y} = \lim_{y_1 \to y_0} \frac{\partial \mathrm{f}(y_1) - \mathrm{f}(y_0)}{y_1 - y_0}$$

Stattdessen wird der Differenzenquotient zu Approximation der Ableitung verwendet:

$$\frac{\mathrm{f}}{\partial x} \approx \frac{\mathrm{f}(x_1) - \mathrm{f}(x_0)}{x_1 - x_0} \quad \text{und} \quad \frac{\partial \mathrm{f}}{\partial y} \approx \frac{\partial \mathrm{f}(y_1) - \mathrm{f}(y_0)}{y_1 - y_0}$$

Bespielsweise

$$\frac{\partial \mathrm{f}}{\partial x} \approx \mathrm{f}[x+1, y] - \mathrm{f}[x, y] \quad \text{und} \quad \frac{\partial \mathrm{f}}{\partial y} \approx \mathrm{f}[x, y+1] - \mathrm{f}[x, y]$$

oder auch beliebt, weil ohne Phasenverschiebung

$$\frac{\partial \mathrm{f}}{\partial x} \approx \frac{1}{2}\left(\mathrm{f}[x+1, y] - \mathrm{f}[x-1, y]\right) \quad \text{und} \quad \frac{\partial \mathrm{f}}{\partial y} \approx \frac{1}{2}\left(\mathrm{f}[x, y+1] - \mathrm{f}[x, y-1]\right)$$

Thorsten Thormählen 22 / 48

Typesetting math: 100%

1. Ableitung berechnen

Diese Berechnungsvorschriften

$\frac{\partial \mathrm{f}}{\partial x} \approx \frac{1}{2}\left(\mathrm{f}[x+1, y]-\mathrm{f}[x-1, y]\right)$ und $\frac{\partial \mathrm{f}}{\partial y} \approx \frac{1}{2}\left(\mathrm{f}[x, y+1]-\mathrm{f}[x, y-1]\right)$

können durch Faltungen mit den entsprechenden Filterkernen realisiert werden:

$$\frac{\partial \mathrm{f}}{\partial x} \approx \mathrm{f}[x, y] * \frac{1}{2}\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

und

$$\frac{\partial \mathrm{f}}{\partial y} \approx \mathrm{f}[x, y] * \frac{1}{2}\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

Dies erklärt warum der Filterkern bei dem gezeigten Beispiel für Kantenfilterung so gewählt wurde

Typesetting math: 100%

Unterdrücken von Rauschen

Das Bilden der Ableitung verstärkt hohe Frequenzen

Bildrauschen ist typischerweise hoch-frequent und wird daher durch die Kantenfilterung besonders verstärkt

Daher kann die Kantenfilterung mit einer rauschunterdrückenden Tiefpassfilterung (Gauss-Filter) in orthogonaler Richtung kombiniert werden:

$$\frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \frac{1}{2}\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

Thorsten Thormählen 24 / 48

Typesetting math: 100%

Sobel-Filter

Diese Kombination eines Tiefpass und der Ableitung in x- und y-Richtung wurden 1968 von Irwin Sobel vorgeschlagen und wird daher Sobel-Filter genannt:

$$\frac{\partial \mathrm{f}}{\partial x} \approx g_x = \mathrm{f}[x,y] * \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \text{und}$$

$$\frac{\partial \mathrm{f}}{\partial y} \approx g_y = \mathrm{f}[x,y] * \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Aus den Ableitung in x- und y-Richtung kann die Stärke der Kante ermittelt werden:

$$g = \sqrt{g_x^2 + g_y^2}$$

Die Richtung der Kante ergibt sich durch:

$$\theta = \mathrm{atan2}(g_x, g_y)$$

2. Ableitung berechnen

Die zweite partiellen Ableitung kann durch Anwendung des Differenzenquotient auf zwei benachbarte Pixel (1. Ableitung) und anschließendes Bilden des Differenzenquotient auf dem Ergebnis (2. Ableitung) erzeugt werden

Beispielsweise für die x-Richtung:

$$\frac{\partial^2 \mathrm{f}}{\partial x^2} \approx (\mathrm{f}[x+1,y] - \mathrm{f}[x,y]) - (\mathrm{f}[x,y] - \mathrm{f}[x-1,y])$$

$$= \mathrm{f}[x+1,y] - 2\mathrm{f}[x,y] + \mathrm{f}[x-1,y]$$

Ausgedrückt als Faltungskern:

$$\frac{\partial^2 \mathrm{f}}{\partial x^2} \approx \mathrm{f}[x,y] * \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

Entsprechend in y-Richtung:

$$\frac{\partial^2 \mathrm{f}}{\partial y^2} \approx \mathrm{f}[x,y] * \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

Laplace-Operator

Sei $\nabla \mathrm{f}$ ein 2-Vektor der die 1. Ableitungen als Elemente enthält:

$$\nabla \mathrm{f} = \begin{pmatrix} \frac{\partial \mathrm{f}}{\partial x} \\ \frac{\partial \mathrm{f}}{\partial y} \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix}}_{\nabla} \mathrm{f}$$

Dann ist der Laplace-Operator definiert als das Skalar-Produkt von ∇ mit sich selbst:

$$\Delta = \nabla^2 = \nabla^\top \nabla = \begin{pmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{pmatrix}^\top \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Das heisst $\Delta \mathrm{f}$ ist die Summe der zweiten Ableitungen von f:

$$\Delta \mathrm{f} = \frac{\partial^2 \mathrm{f}}{\partial x^2} + \frac{\partial^2 \mathrm{f}}{\partial y^2}$$

Thorsten Thormählen 27 / 48

Typesetting math: 100%

Laplace-Filter

Entsprechend dem Laplace-Operator ist der Laplace-Filter definiert als die Summe der 2. Ableitungen in x- und y-Richtung

Mit dem vorherigen Ergebnis für den Filterkern der 2. Ableitung, ergibt sich für den Laplace-Filter:

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

Laplacian of Gaussian (LoG)

Der Laplace-Filter verstärkt aufgrund der 2. Ableitung Rauschen besonders stark

Daher wird er in der Regel immer mit einem Gauss-Filter kombiniert

Da die Reihenfolge der Faltung egal ist, kann das Eingabebild auch direkt mit der Kombination aus Gauss-Filter und Laplace-Filter gefaltet werden

Bzw. es kann auch direkt die analytische 2. Ableitung einer 2D Gauss-Funktion verwendet werden:

$$\mathrm{b}[x,y] = \exp\left(-\frac{x^2+y^2}{2\,\sigma^2}\right)$$

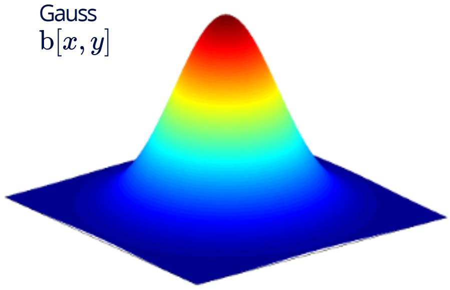
$$\frac{\partial \mathrm{b}}{\partial x} = -\frac{x}{\sigma^2}\exp\left(-\frac{x^2+y^2}{2\,\sigma^2}\right)$$

$$\frac{\partial^2 \mathrm{b}}{\partial x^2} = \frac{x^2}{\sigma^4}\exp\left(-\frac{x^2+y^2}{2\,\sigma^2}\right) - \frac{1}{\sigma^2}\exp\left(-\frac{x^2+y^2}{2\,\sigma^2}\right) = \frac{x^2-\sigma^2}{\sigma^4}\exp\left(-\frac{x^2+y^2}{2\,\sigma^2}\right)$$

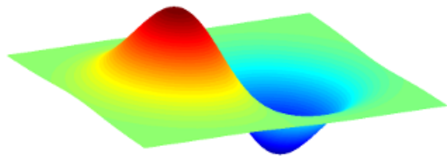
$$\Delta \mathrm{b} = \frac{\partial^2 \mathrm{b}}{\partial x^2} + \frac{\partial^2 \mathrm{b}}{\partial y^2} = \frac{x^2+y^2-2\,\sigma^2}{\sigma^4}\exp\left(-\frac{x^2+y^2}{2\,\sigma^2}\right)$$

Laplacian of Gaussian (LoG)

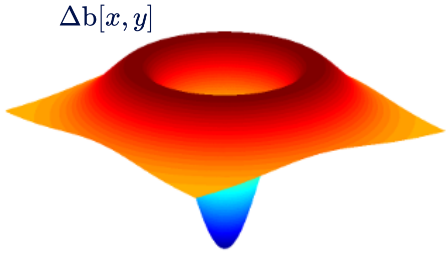
Gauss
$\mathrm{b}[x,y]$



1. Ableitung
$\frac{\partial \mathrm{b}[x,y]}{\partial x}$



Laplacian of Gaussian
$\Delta \mathrm{b}[x,y]$



Typesetting math: 100%

Difference of Gaussian

Der Difference-of-Gaussian-Filter entsteht durch Subtraktion der Ergebnisse von Faltungen mit unterschiedlich starken Gauß-Filtern

$$\mathrm{f}[x,y] = \mathrm{a}[x,y] * \frac{1}{\sigma_1^2} \exp\left(-\frac{x^2+y^2}{2\,\sigma_1^2}\right) - \mathrm{a}[x,y] * \frac{1}{\sigma_2^2} \exp\left(-\frac{x^2+y^2}{2\,\sigma_2^2}\right)$$

wobei gilt $\sigma_1 > \sigma_2$

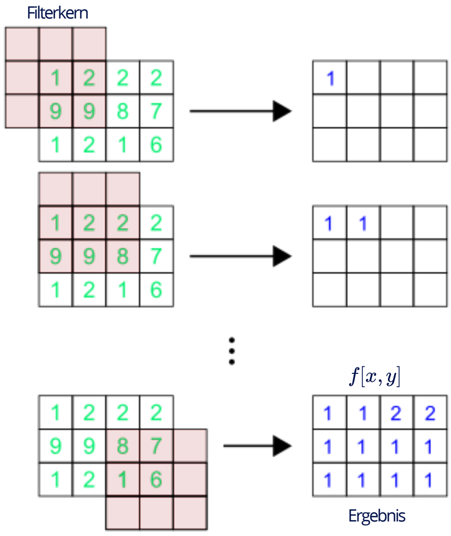
Dies liefert sehr ähnliche Ergebnisse wie der Laplacian of Gaussian. Warum?

Morphologische Operationen

Erode

- Erode ist ein Beispiel für ein Filter, das nicht mit einer Faltungsoperation implementiert werden kann
- Trotzdem wird mit einem Filterkern (Kernel) einer bestimmten Größe über das Bild gelaufen, z.B. 3 x 3, 5 x 5 usw.
- Der Wert im gefilterten Bild $\mathrm{f}[x,y]$ entspricht dem Minimum aller Pixel innerhalb des Filterkerns $\mathcal{K}$ im Eingangsbild $\mathrm{a}[x,y]$:

$$\mathrm{f}[x,y] = \min_{i,k \,\in\, \mathcal{K}} (\mathrm{a}[j,k])$$



Thorsten Thormählen 33 / 48

Erode

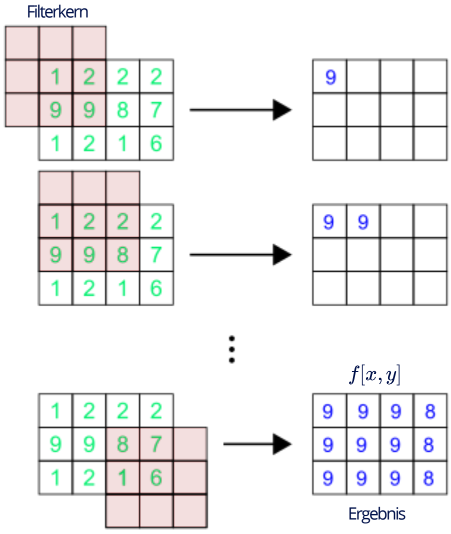
Typesetting math: 100%

Dilate

Beim Dilate-Filter wird statt des Minimums das Maximum verwendet

Der Wert im gefilterten Bild $\mathrm{f}[x, y]$ entspricht dem Minimum aller Pixel innerhalb des Filterkerns $\mathcal{K}$ im Eingangsbild $\mathrm{a}[x, y]$:

$$\mathrm{f}[x, y] = \max_{i,k \in \mathcal{K}} (\mathrm{a}[j, k])$$



Typesetting math: 100%

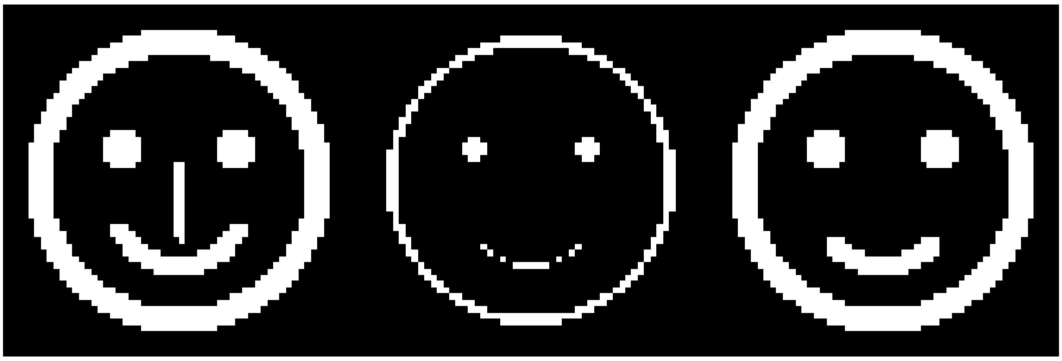
Dilate

Typesetting math: 100%

Open

Open wird zum Öffnen von Strukturen verwendet

Erst Erode anwenden, dann Dilate



Typesetting math: 100%

Close

Close wird zum Verbinden von Strukturen verwendet

Erst Dilate anwenden, dann Erode

Typesetting math: 100%

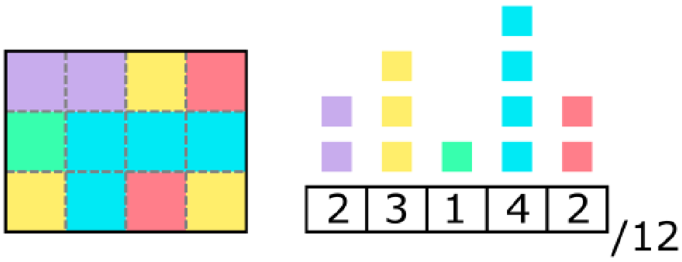
Histogramme

Histogramme

Ein Histogramm enthält die Anzahl der Pixel für jeden Intensitätswert geteilt durch die Gesamtzahl der Pixel

Mit anderen Worten, ein Histogramm beschreibt die relative Häufigkeit von Intensitätswerten eines Bildes

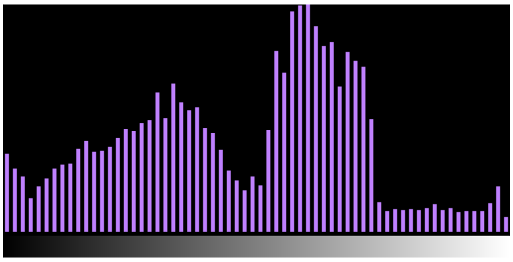
Für ein Bild mit 4 x 3 Pixeln und 5 verschiedenen Intensitätswerten ergibt sich beispielsweise:

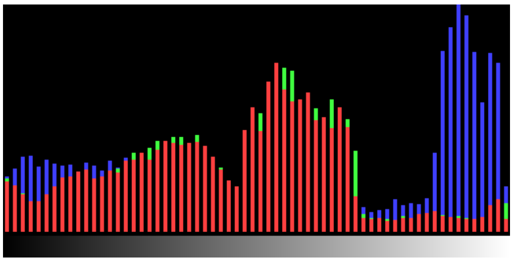


Typesetting math: 100%

Histogramme

Das Histogramm kann auf den Grau- oder den Farbwerten berechnet werden:





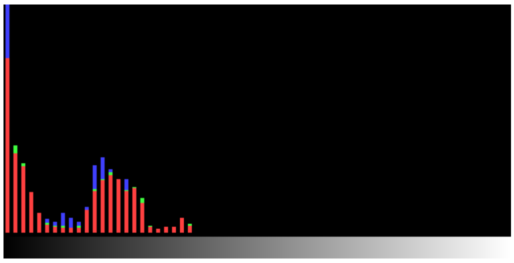
Typesetting math: 100%

Histogramme

Ein Histogramm kann z.B. bei der Aufnahme eines Bildes hilfreich sein, um zu kontrollieren, ob der zur Verfügung stehende Wertebereich gut abgedeckt ist.

Überbelichtet:

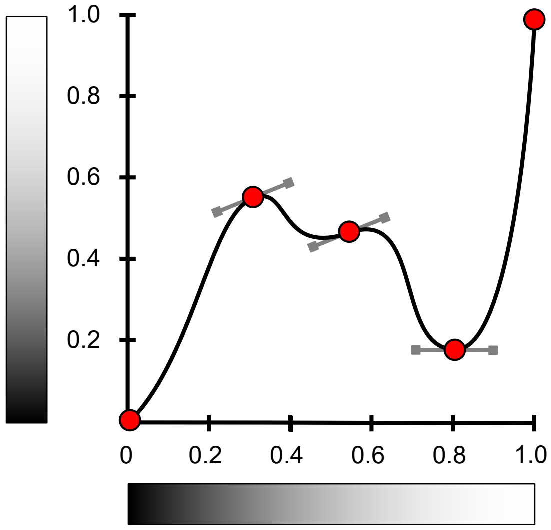
Unterbelichtet:



Typesetting math: 100%

Manipulation von Farbwerten

Viele Bildverarbeitungsprogramme bieten die Möglichkeit, eine beliebige Kurve zu definieren, die von Eingangs- auf Ausgangsintensitäten abbildet



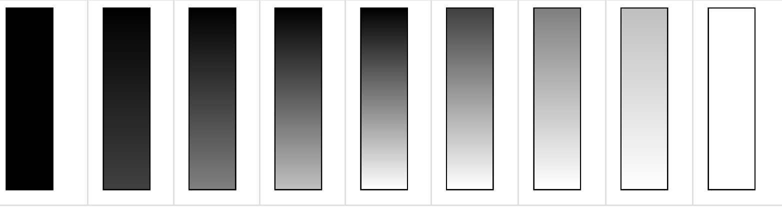
Typesetting math: 100%

Helligkeit / Kontrast

Helligkeit / Kontrast

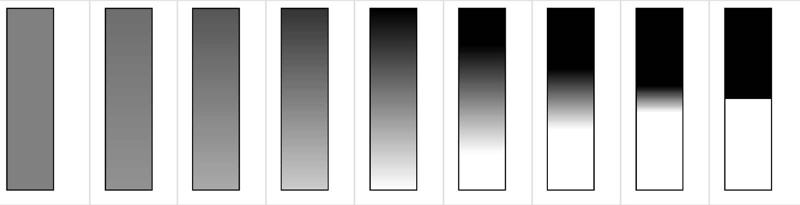
Helligkeitsänderung:	-100	-75	-50	-25	0	25	50	75	100

Ergebnis:



Kontraständerung:	-100	-75	-50	-25	0	25	50	75	100

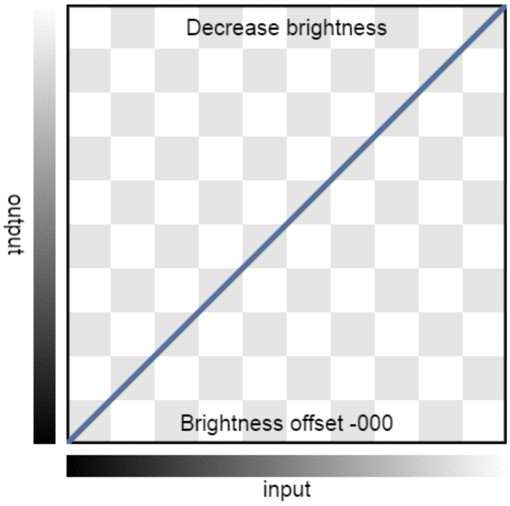
Ergebnis:

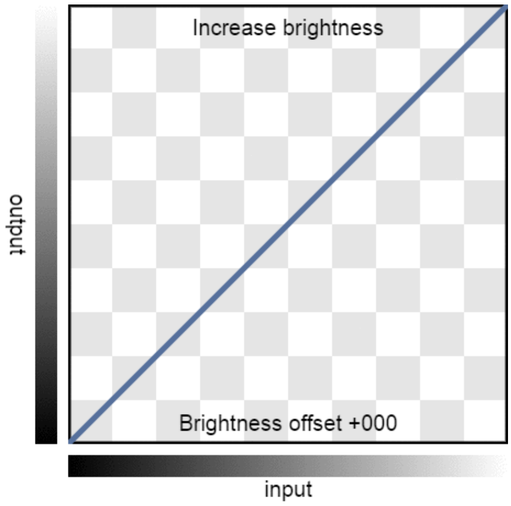


Thorsten Thormählen 45 / 48

Typesetting math: 100%

Helligkeit

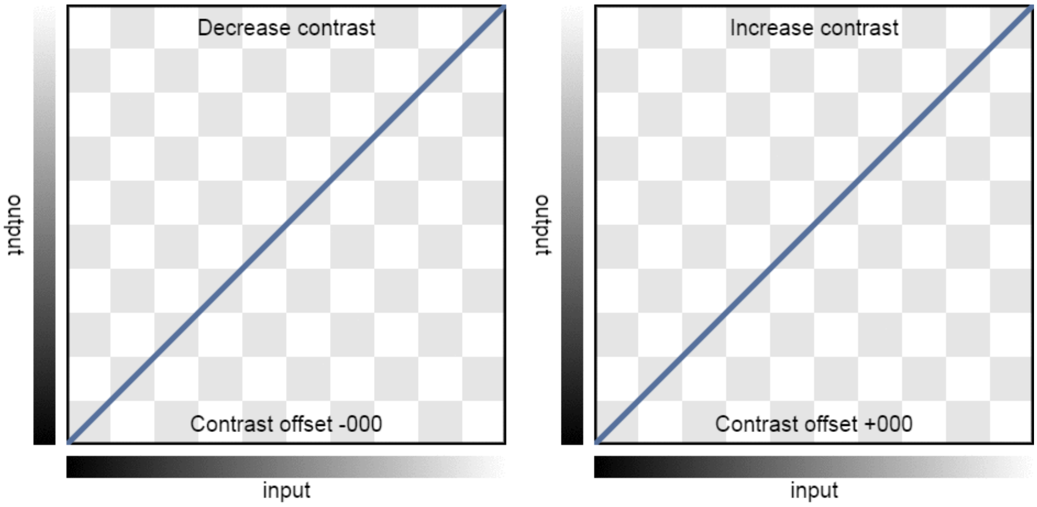




Eine negative Helligkeitsänderung konzentriert alle Farben in Richtung Schwarz

Eine positive Helligkeitsänderung konzentiert alle Farben in Richtung Weiß

Kontrast



Eine negative Kontraständerung verändert alle Farben hin zu Mittelgrau

Eine positive Kontraständerung verändert alle Farben weg von Mittelgrau

Typesetting math: 100%

Gibt es Fragen?

Anregungen oder Verbesserungsvorschläge können auch gerne per E-mail an mich gesendet werden: Kontakt

Weitere Vorlesungsfolien

Typesetting math: 100%