

Aufgabenstellungen:	Übungen_03	http://www.uni-forst.gwdg.de/~wkurth/bia17_ub04.pdf
	Übungsblatt_02	http://www.uni-forst.gwdg.de/~wkurth/bia21_ue02.pdf

ÜB2:Aufgabe 3:

Schritt #1: das Thema der Aufgabe bestimmen und begreifen

Geometrische Bildtransformation: ist eine mathematische (im Grunde – eine **formale**) Beschreibung einer **räumlichen** Transformation eines Bildes



Scherung = Rotationen + Skalierungen
Hausaufgabe: warum?



So eine geometrische **T**ransformation des Bildes $B(z, s)$ schaut dann (algebraisch) folgendermaßen aus:

Koordinatenwechsel: entweder bewegt/deformiert sich das Bild oder das ursprüngliche Koordinatensystem (in Gegenrichtung)

$$TB(z, s) = B(z', s')$$

Detailwissen: das sieht nach einer waagerechte Scherungstransformation aus

Wobei:

Normalerweise kommen noch Interpolationstechniken dazu ...

$$\begin{cases} z' = z'(z, s) \\ s' = s'(z, s) \end{cases} \text{ und, bei Umkehrtransformation } T^{-1}: \begin{cases} z = z(z', s') \\ s = s(z', s') \end{cases}$$

Für Mathe Fans☺: es stellt sich die Frage, ob es so eine Inverse überhaupt gibt

~~DISTORTION~~

Bereichseinschränkung: nur **affine** Transformationen, d.h., für fixierte Werte a_i und b_i (wobei $i \in \{t, z, s\}$):

$$\begin{cases} z' = z'(z, s) = a_t + a_z \cdot z + a_s \cdot s \\ s' = s'(z, s) = b_t + b_z \cdot z + b_s \cdot s \end{cases} \text{ oder, kurz in Matrixform: } \begin{bmatrix} z' \\ s' \end{bmatrix} = \begin{bmatrix} a_z & a_s \\ b_z & b_s \end{bmatrix} \cdot \begin{bmatrix} z \\ s \end{bmatrix} + \begin{bmatrix} a_t \\ b_t \end{bmatrix} \text{ oder noch kürzer in homogener Matrixform: } \begin{bmatrix} z' \\ s' \\ 1 \end{bmatrix} = \begin{bmatrix} a_z & a_s & a_t \\ b_z & b_s & b_t \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} z \\ s \\ 1 \end{bmatrix}$$

Weitere **Einschränkung:** die **Invertierbarkeit** der Matrix $\begin{bmatrix} a_z & a_s \\ b_z & b_s \end{bmatrix}$ wird i.d.R. bereits in der Definition von affinen Transformationen vorausgesetzt (**Hausaufgabe:** versucht zu klären, warum). Der **Singularwertzerlegungssatz** besagt dann - so eine affine Transformation besteht aus Kombinationen (d.h. homogenen Matrixmultiplikationen) von:

Verschiebungstransformationen (aka Translationsmatrizen)	(nicht unbedingt gleichmäßigen) Skalierungstransformationen (können verzerrern)
Drehungs- & Spiegelungstransformationen (aka Rotations & Reflexionsmatrizen)	Scherungstransformationen (verursachen Verzerrung aka Dehnung/Stauchung)

Schritt #2: tatsächliche Lösung der Aufgabe (mit einer Prise theoretischer Grundlagen ☺):

Mathematischer **Satz #1:** jede 2D affine Transformation hat eine Inverse (**Hausaufgabe:** checkt ob sie dabei eindeutig ist ☺) und diese Inverse an sich ist auch affin.

Fundamentaler **Satz #2:** für jeden 3 x 3 **bipartiten Graphen** $\vec{p}_1(p_{11}, p_{12}) \rightarrow \vec{p}_1'(p_{11}', p_{12}')$ von **nichtkollinearen** (Pass aka **Referenz**) Punkten auf jeder Seite, es gibt die einzigartige affine Transformation, die die linke Seite auf die rechte Seite abbildet.

(a) + (b)

$$\begin{aligned}
 & \begin{cases} \begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 1 \end{pmatrix} \\ \begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix} \\ \begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix} \end{cases} \Leftrightarrow \begin{cases} \begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 1 \end{pmatrix} \\ \begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix} \\ \begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix} \end{cases} \Leftrightarrow \begin{cases} 2a + 0b + u = 2 \\ 2c + 0d + v = 5 \\ 0a + b + u = 1 \\ 0c + d + v = 3 \\ 0a - b + u = 3 \\ 0c - d + v = 3 \end{cases} + \Leftrightarrow \begin{cases} 2a + 0b + u = 2 \\ 2c + 0d + v = 5 \\ 0a + b + u = 1 \\ 0c + d + v = 3 \\ 2u = 4 \\ 2v = 6 \end{cases} \Leftrightarrow \begin{cases} 2a = 2 - 2 \\ 2c = 5 - 3 \\ b = 1 - 2 \\ d = 3 - 3 \\ u = 2 \\ v = 3 \end{cases} \Leftrightarrow \begin{cases} a = 0 \\ c = 1 \\ b = -1 \\ d = 0 \\ u = 2 \\ v = 3 \end{cases}
 \end{aligned}$$

Also:

$$\begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 2 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \text{Verschiebung}[(2, 3)] \circ \text{Drehung}[90^\circ \circlearrowleft]$$

Korrekte Reihenfolge nicht vergessen

Randbemerkung: der Wolkenkratzer ☺ da oben ist einfach ne Matrizengleichung der Art $[X] \cdot [A] = [B]$ und lässt sich im Allgemeinen (nicht unbedingt zeit)effizienter lösen:

$$\begin{aligned}
 & \begin{cases} \begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 1 \end{pmatrix} \\ \begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix} \\ \begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix} \end{cases} \Leftrightarrow \begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 3 \\ 5 & 3 & 3 \\ 1 & 1 & 1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 3 \\ 5 & 3 & 3 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix}^{-1}
 \end{aligned}$$

Wir haben also das ganze Problem auf die Berechnung von $\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix}^{-1}$ reduziert – und diese Inverse apriorisch existiert, denn die Determinante der Matrix $\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix}$ stellt den vorzeichenbehafteten Flächeninhalt des Dreiecks mit den Spitzen in p_1', p_2', p_3' dar, und diese Fläche ist, unter der Nichtkollinearitätsannahme, ungleich Null.

Bildanalyse und Bildverstehen		Georg-August-Universität Göttingen SoSe 21
Fragen/Anregungen zum Stoff? ⇒ 24/7-Support ☺ unter jeos@mail.com emailen		
Lösungen #04	Prof. Winfried Kurth / Alex Tavkheldidze	

ÜB2:Aufgabe 4:

Schritt #1: Einträge der DFT-Bildmatrix berechnen [fürs Ursprungsbild $\mathbf{B} = (b_{jk})$]:

$$DFT(\mathbf{B}) = (g_{mn})$$

$$g_{mn} = \frac{1}{L \cdot R} \cdot \sum_{j=0}^{L-1} \sum_{k=0}^{R-1} b_{jk} \cdot e^{-2\pi i \left(\frac{mj}{L} + \frac{nk}{R} \right)}$$

Schritt #2: Einträge der DFT-Bildmatrix berechnen [fürs modifizierte Bild $\hat{\mathbf{B}} = ((-1)^{j+k} \cdot b_{jk})$]:

$$DFT(\hat{\mathbf{B}}) = (\widehat{g}_{mn})$$

$$\widehat{g}_{mn} = \frac{1}{L \cdot R} \cdot \sum_{j=0}^{L-1} \sum_{k=0}^{R-1} (-1)^{j+k} \cdot b_{jk} \cdot e^{-2\pi i \left(\frac{mj}{L} + \frac{nk}{R} \right)} = \frac{1}{L \cdot R} \cdot \sum_{j=0}^{L-1} \sum_{k=0}^{R-1} b_{jk} \cdot e^{+\pi i(j+k)} \cdot e^{-2\pi i \left(\frac{mj}{L} + \frac{nk}{R} \right)} =$$

$$\frac{1}{L \cdot R} \cdot \sum_{j=0}^{L-1} \sum_{k=0}^{R-1} b_{jk} \cdot e^{-2\pi i \left(\frac{mj}{L} + \frac{nk}{R} + \frac{j+k}{2} \right)} = \frac{1}{L \cdot R} \cdot \sum_{j=0}^{L-1} \sum_{k=0}^{R-1} b_{jk} \cdot e^{-2\pi i \left(\frac{(m \pm \frac{L}{2})j}{L} + \frac{(n \pm \frac{R}{2})k}{R} \right)} = g_{m \pm \frac{L}{2}, n \pm \frac{R}{2}}$$



Aufgabe U6:

Schritt #1: laut **Distributivität** der **Faltung**

$$K * B - K * B_0 = K * (B - B_0) = K * R$$

Dann, laut **Faltungsdefinition**:

$$[K * R](i, j) = \sum_{z=0}^2 \sum_{s=0}^2 \frac{1}{9} \cdot 1 \cdot [R](i - z, j - s) = \frac{1}{9} \cdot \sum_{z=0}^2 \sum_{s=0}^2 [R](i - z, j - s)$$

Schritt #2: wegen **Linearität** des **Erwartungswertoperators**

$$E([K * R](i, j)) = E\left(\frac{1}{9} \cdot \sum_{z=0}^2 \sum_{s=0}^2 [R](i - z, j - s)\right) = \frac{1}{9} \cdot \sum_{z=0}^2 \sum_{s=0}^2 E([R](i - z, j - s)) = \frac{1}{9} \cdot \sum_{z=0}^2 \sum_{s=0}^2 0 = \frac{1}{9} \cdot 0 = 0$$

Schritt #3: laut dem **Verschiebungssatz** für diskrete Zufallsvariablen

$$Var([K * R](i, j)) = E\left(\left([K * R](i, j)\right)^2\right) - \left(E([K * R](i, j))\right)^2 = E\left(\left([K * R](i, j)\right)^2\right) - 0 = E\left(\left([K * R](i, j)\right)^2\right) =$$

$$E\left(\left(\frac{1}{9} \cdot \sum_{z=0}^2 \sum_{s=0}^2 [R](i - z, j - s)\right)^2\right) = \left(\frac{1}{9}\right)^2 \cdot E\left(\left(\sum_{z=0}^2 \sum_{s=0}^2 [R](i - z, j - s)\right)^2\right) =$$

$$\frac{1}{81} \cdot E\left(\sum_{z=0}^2 \sum_{s=0}^2 ([R](i - z, j - s))^2 + \sum_{(i', j') \neq (i'', j'')} [R](i', j') \cdot [R](i'', j'')\right) =$$

$$\frac{1}{81} \cdot \left(E\left(\sum_{z=0}^2 \sum_{s=0}^2 ([R](i - z, j - s))^2\right) + E\left(\sum_{(i', j') \neq (i'', j'')} [R](i', j') \cdot [R](i'', j'')\right) \right) =$$

$$\frac{1}{81} \cdot \left(\sum_{z=0}^2 \sum_{s=0}^2 E\left(\left([R](i - z, j - s)\right)^2\right) + \sum_{(i', j') \neq (i'', j'')} E([R](i', j') \cdot [R](i'', j'')) \right) =$$

$$\frac{1}{81} \cdot \left(\sum_{z=0}^2 \sum_{s=0}^2 E\left(\left([R](i - z, j - s) - E([R](i - z, j - s))\right)^2\right) + \sum_{(i', j') \neq (i'', j'')} E([R](i', j')) \cdot E([R](i'', j'')) \right) = \frac{1}{81} \cdot \left(3 \cdot 3 \cdot Var([R](i - z, j - s)) + \sum_{(i', j') \neq (i'', j'')} 0 \cdot 0 \right) = \frac{1}{81} \cdot (9 \cdot \sigma^2 + 0) = \frac{\sigma^2}{9}$$

Weil es gilt:

$$\left(\sum_{n=1}^N a_n\right)^2 = \sum_{n=1}^N a_n^2 + \sum_{m \neq n} a_n \cdot a_m$$

[lässt sich zB. durch Induktionsbeweis zeigen]

Beispiel für $N = 2$:

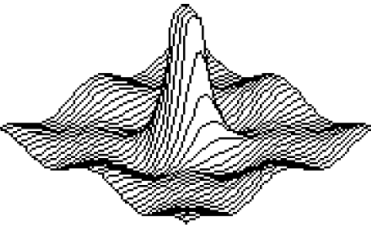
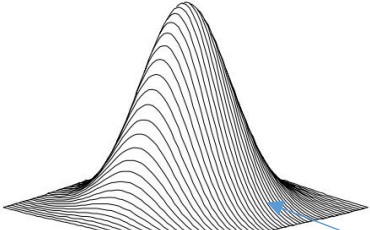
$$(a_1 + a_2)^2 = a_1^2 + a_2^2 + a_1 \cdot a_2 + a_2 \cdot a_1$$

Erwartungswert des Produkts von **stochastisch unabhängigen** Zufallsvariablen

Aufgabe U7:

Schritt #1: besserer Glättungsfilter als Mittelwertfilter – intuitiver Ansatz

Fun Fact: $\frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{1+1+1+1} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \frac{1}{1+1+1+1} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

Transferfunktion aka Übertragungsfunktion von Tiefpassfiltern			
3x3-er Mittelwertfilter		3x3-er Binomialfilter aka ganzzahlige & schnelle Gaußfilter-Approximation	
Sinc-Funktion aka Kardinalsinus	Dunkle Materie ☹	Gauß-Glocke $F(u, v) = (\cos(\pi u/Z))^2 \cdot (\cos(\pi v/S))^2$ ← Hausaufgabe: checkt's	
	$\frac{1}{1+1+1} \cdot \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} * \frac{1}{1+1+1} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ $\frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$		$\frac{1}{1+2+1} \cdot \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \frac{1}{1+2+1} \cdot \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$ $\frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

bei ausreichend großer Anzahl von Frequenzen im Frequenzraum

Beobachtung: schöne Isotropie (d.h. Richtungsunabhängigkeit)

Mögliches **Erfindungsschema** ☺ von 1D Binomialfiltern – wie kommt man auf Binomialfilter?

Bausteine	Verknüpfung	normalisierte aka normierte 1D Binomialfilter	coole ☺ Eigenschaften
[1 1]	[1 1] * [1 1]	$[b_1] = \left(\frac{1}{1+1}\right)^1 \cdot [1 \ 1]^{*1} = \frac{1}{1+1} \cdot [1 \ 1]$ $[b_2] = \left(\frac{1}{1+1}\right)^2 \cdot [1 \ 1]^{*2} = \frac{1}{1+1} \cdot [1 \ 1] * \frac{1}{1+1} \cdot [1 \ 1] = \frac{1}{1+2+1} \cdot [1 \ 2 \ 1]$ $[b_3] = \left(\frac{1}{1+1}\right)^3 \cdot [1 \ 1]^{*3} = \frac{1}{1+3+3+1} \cdot [1 \ 3 \ 3 \ 1]$ <p>.....</p>	$\sum_{i=0}^k [b_k](i) = 2^k$ $b_n * b_m = b_{n+m}$ <hr/> $b_k \rightarrow G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
triviale 1D Glättungsfilter	Faltung	Verkettungen	Zentraler Grenzwertsatz beweist diese zu Hause ☺

Hausaufgabe: unter welchen Voraussetzungen an μ und σ erreicht man die Annäherung $G(i) \cong 2^{-k} \cdot \binom{k}{i}$?
 [Hinweis: was wäre bei $\mu = \frac{k}{2}$ und $\sigma^2 = \frac{k}{4}$? – guckt euch die [Binomialverteilung](#) und das [Zufallsbrett](#) an]

Pascalshes Dreieck: Veranschaulichung von Koeffizienten
 eines Binoms $(x + y)^k = \sum_{i=0}^k \binom{k}{i} \cdot x^i \cdot y^{k-i}$

Schritt #2: Zuwachs an Dimension (1D -> 2D)

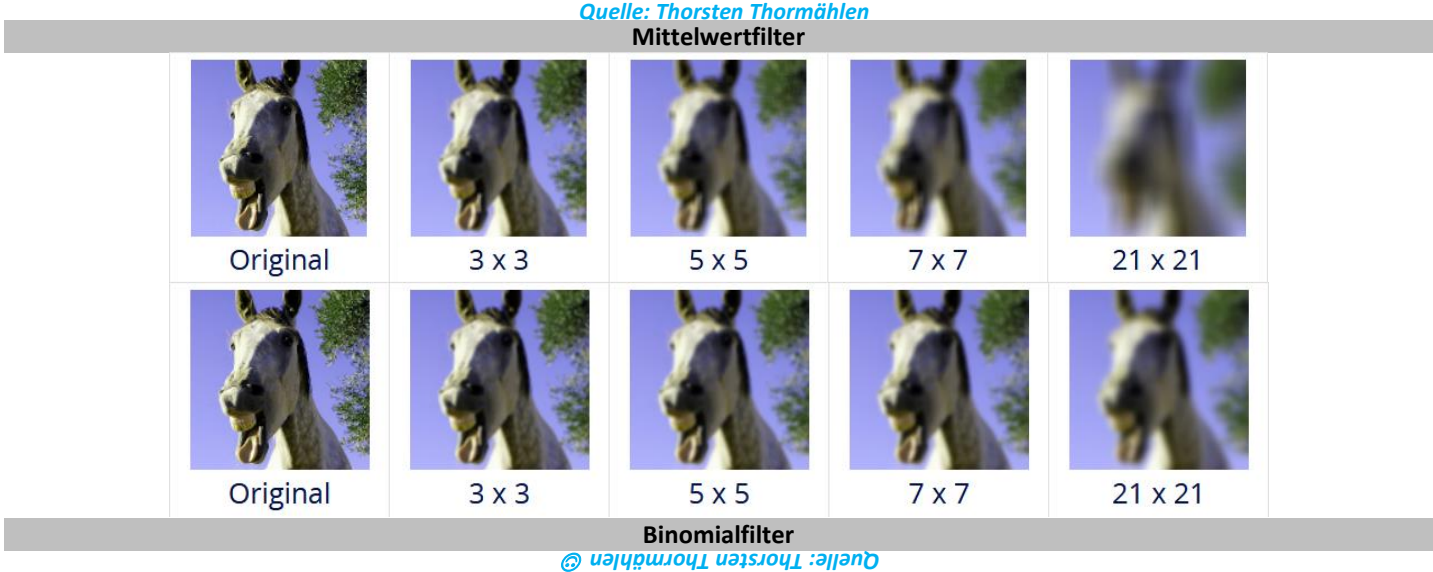
2D Binomialfilter = 1D Binomialfilter & x/y *-Separierbarkeit

$$[{}^2b_{m \times n}] = [b_m]^T * [b_n]$$

denn die 2D-Gaußfunktion $G(x, y) = \text{Amplitude} \cdot e^{-\left(\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right)}$ ist **x/y-separierbar** (wegen $e^{x^2+y^2} = e^{x^2} \cdot e^{y^2}$)

Aufgabe U7:

Schritt #3: Anwendungsbeispiele



Schritt #4: Lösung der Aufgabe

$$B_2^T \cdot B_3 = \frac{1}{2^2} \cdot \left[\begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right]^T \cdot \frac{1}{2^3} \cdot \left[\begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 3 \end{pmatrix} \right] = \frac{1}{4} \cdot [1, 2, 1]^T \cdot \frac{1}{8} \cdot [1, 3, 3, 1] = \frac{1}{4} \cdot \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot \frac{1}{8} \cdot [1, 3, 3, 1] = \frac{1}{4} \cdot \frac{1}{8} \cdot \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot [1, 3, 3, 1] = \frac{1}{32} \begin{bmatrix} 1 & 3 & 3 & 1 \\ 2 & 6 & 6 & 2 \\ 1 & 3 & 3 & 1 \end{bmatrix}$$

Interessante Nebenbeobachtung für Mathe Fans ☺:

Achtung: **Faltung** ist im Allgemeinen **nicht** das Gleiche wie **Matrixmultiplikation**

$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * [1, 3, 3, 1] = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 2 & 6 & 6 & 2 \\ 1 & 3 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot [1, 3, 3, 1]$

Beobachtung: **Faltung** zweier Vektoren (also, **Spalten-** und **Zeilenvektoren**) liefert **das gleiche Ergebnis** wie deren **Matrixmultiplikation** (also **dyadisches** aka **tensorielles** aka **äußeres Produkt**)

➔ **Hausaufgabe:** warum und wann gilt sowas im Allgemeinen?

Hinweis: konstruiert ein Verfahren bei dem man **jede Faltung** [einfachheitshalber, wählt den Faltungsmodus „valid“] **zweier Matrizen** durch die **Matrixmultiplikation** deren **Umformungen darstellt** – nimmt eine kleine Bildmatrix, einen noch kleineren ☺ Kern und versucht so ein Verfahren selber zu basteln.

dünnbesetzte doppelblock-zyklische Matrizen [aka **Toeplitz-Blockmatrizen**] könnten euch auf dem Lösungsweg recht behilflich sein ☺

Aufgabe U7:

Schritt #5: Eignung als ein besserer Tiefpassfilter [am Beispiel zweier **periodischen Störungen**]

$$\frac{1}{32} \begin{bmatrix} 1 & 3 & 3 & 1 \\ 2 & 6 & 6 & 2 \\ 1 & 3 & 3 & 1 \end{bmatrix} * \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \vdots & -2 & 1 & 1 & \vdots \\ \vdots & -2 & 1 & 1 & \vdots \\ \vdots & -2 & 1 & 1 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \vdots & -\frac{1}{8} & \frac{1}{4} & -\frac{1}{8} & \vdots \\ \vdots & -\frac{1}{8} & \frac{1}{4} & -\frac{1}{8} & \vdots \\ \vdots & -\frac{1}{8} & \frac{1}{4} & -\frac{1}{8} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad \Bigg| \quad \frac{1}{32} \begin{bmatrix} 1 & 3 & 3 & 1 \\ 2 & 6 & 6 & 2 \\ 1 & 3 & 3 & 1 \end{bmatrix} * \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \vdots & -1 & 1 & -1 & \vdots \\ \vdots & -1 & 1 & -1 & \vdots \\ \vdots & -1 & 1 & -1 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \vdots & 0 & 0 & 0 & \vdots \\ \vdots & 0 & 0 & 0 & \vdots \\ \vdots & 0 & 0 & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

2D Binomialfilter zeigen hiermit ein **besseres Tiefpassverhalten** als Mittelwertfilter

Aufgabe U8:

Kurze Anmerkung: im Grunde genommen, das gleiche „Ent-„/„De-“ Thema (also **Inversen** aka **Umkehrtransformationen**)

(a)

Schritt #1: mathematische Reifungsphase der Aufgabe (paraphrasiert, ein „**Warum** diese Aufgabe“-Kontext)

im mathefreien ☺ Anwendungsbereich wird das wie „**will mein pures Bild zurück**“ klingen ☺

Da die Faltung assoziativ ist (**Hausaufgabe:** übrigens beweist es oder schaut euch einen von gültigen Beweisen an – hunderte Quellen allein im WWW-Raum vorhanden ☺):

$$K^{*-1} * K * B = K^{*-1} * (K * B) = (K^{*-1} * K) * B = I_* * B = B$$

1D-Faltungsmasken sind einfacher zu behandeln und repräsentieren zudem (in Bezug auf **Schachtelung/Komposition**) Bausteine einer wesentlich schnelleren Umsetzung entsprechendes 2D-Faltungsvorgangs:

Beispiele	
Glättungsfilter [am Beispiel vom 3x3-Mittelwertfilter]	Kantenfilter [am Beispiel vom waagerechten Sobelfilter]
$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * \frac{1}{3} [1 \ 1 \ 1] \Leftrightarrow \frac{1}{3} [1 \ 1 \ 1] * \frac{1}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * [1 \ 0 \ -1] \Leftrightarrow [1 \ 0 \ -1] * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$

Hausaufgabe: checkt's

Zusatzbeobachtung: aus vorgegebenen **Ausgangs-9**, **Filter-3** und **Bildvektorlänge 7** ergibt sich der **Faltungsmodus „full“**, denn $9 = 3 + 7 - 1$

Schritt #2: Lösung der Aufgabe

Laut **1D-Faltungsformel**, für $n = 0..8$:

$$[F * G](n) \stackrel{\text{def}}{=} \sum_{i=0}^2 F(i) \cdot G(n-i) = \frac{1}{4} \cdot G(n) + \frac{1}{2} \cdot G(n-1) + \frac{1}{4} \cdot G(n-2) = \frac{1}{4} \cdot (1 \cdot G(n) + 2 \cdot G(n-1) + 1 \cdot G(n-2)) = I(n)$$

In Matrixschreibweise:

$$\frac{1}{4} \cdot M_{9 \times 7} \cdot G_{7 \times 1} = I_{9 \times 1} \Leftrightarrow \frac{1}{4} \cdot \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 2 & 1 & 0 & \dots & 0 \\ 1 & 2 & 1 & \dots & 0 \\ 0 & 1 & 2 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 2 \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} G(0) \\ G(1) \\ G(2) \\ G(3) \\ G(4) \\ G(5) \\ G(6) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Flacherdler-Witz☺: das Zeichen \dots steht für „**turtles all the way down**“ ☺

Laut **Lösbarkeitssatz linearer Gleichungssysteme (LS-LGS)**: dieses da oben ist lösbar genau dann, wenn $\text{Rang}(M_{9 \times 6}) = \text{Rang}(M_{9 \times 6} | I_{9 \times 1})$: falls = 7, dann **ein**deutig
falls < 7, dann **un**-deutig

Bildanalyse und Bildverstehen		Georg-August-Universität Göttingen SoSe 21
Fragen/Anregungen zum Stoff? ⇒ 24/7-Support ☺ unter jeos@mail.com emailen		
Lösungen #04	Prof. Winfried Kurth / Alex Tavkheldidze	

Hausaufgabe: überprüft (beispielsweise durch die Anwendung [gaußsches Eliminationsverfahrens](#)), dass sich dieses LGS **nicht lösen** lässt, und zwar für beliebige Bildvektorenlänge.

(b)

Schritt #1: unverbindliche ☺ **Annahme**

Zwecks Einfachheit von Berechnungen & Senkung der Anzahl von Veränderlichen, sei F^+ ein symmetrischer Vektor (aka **Palindrom**): $F^+ = [d \ c \ b \ a \ b \ c \ d]$

Schritt #2: dann wird das [obige LGS](#) zu

$$\frac{1}{4} \cdot \begin{bmatrix} 1 & 0 & 0 & \ddots & 0 \\ 2 & 1 & 0 & \ddots & 0 \\ 1 & 2 & 1 & \ddots & 0 \\ 0 & 1 & 2 & \ddots & 0 \\ 0 & 0 & 1 & \ddots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \ddots & 1 \\ 0 & 0 & 0 & \ddots & 2 \\ 0 & 0 & 0 & \ddots & 1 \end{bmatrix} \cdot \begin{bmatrix} F^+(0) \\ F^+(1) \\ F^+(2) \\ F^+(3) \\ F^+(4) \\ F^+(5) \\ F^+(6) \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow \frac{1}{4} \cdot \begin{bmatrix} 1 & 0 & 0 & \ddots & 0 \\ 2 & 1 & 0 & \ddots & 0 \\ 1 & 2 & 1 & \ddots & 0 \\ 0 & 1 & 2 & \ddots & 0 \\ 0 & 0 & 1 & \ddots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \ddots & 1 \\ 0 & 0 & 0 & \ddots & 2 \\ 0 & 0 & 0 & \ddots & 1 \end{bmatrix} \cdot \begin{bmatrix} d \\ c \\ b \\ a \\ b \\ c \\ d \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow \frac{1}{4} \cdot \begin{bmatrix} d \\ 2d+c \\ d+2c+b \\ c+2b+a \\ 2b+2a \\ a+2b+c \\ b+2c+d \\ c+2d \\ d \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} d \\ 2d+c \\ d+2c+b \\ c+2b+a \\ 2b+2a \\ a+2b+c \\ b+2c+d \\ c+2d \\ d \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 4 \\ 0 \\ 0 \end{bmatrix}$$

Schritt #2: lass uns 4-stellige **legendresche Verlustfunktion** (aka **Summe der Abweichungsquadrate**) und dadurch unsere **Optimierungsaufgabe** (aka **MKQ**) bilden:

$$S(a, b, c, d) = \sum_{n=0}^8 ([M \cdot F^+](n) - 4 \cdot I(n))^2 \rightarrow \text{Min}$$

In entwickelter quadratischer Form:

$$S(a, b, c, d) = 16 - 16a - 16b + 6a^2 + 16ab + 4ac + 14b^2 + 16bc + 4bd + 12c^2 + 16cd + 12d^2 \rightarrow \text{Min}$$

Schritt #3: laut [Algorithmus zum Auffinden von mehrdimensionalen Extremstellen](#)

- Wir berechnen den **Gradienten** unserer Funktion $S(a, b, c, d)$:

$$\nabla S(a, b, c, d) \stackrel{\text{def}}{=} \left[\frac{\partial S}{\partial a}, \frac{\partial S}{\partial b}, \frac{\partial S}{\partial c}, \frac{\partial S}{\partial d} \right]^T = \begin{bmatrix} 12a + 16b + 4c - 16 \\ 16a + 28b + 16c + 4d - 16 \\ 4a + 16b + 24c + 16d \\ 4b + 16c + 24d \end{bmatrix}$$

- Wir lösen das LGS $\nabla S(a, b, c, d) = \mathbf{0}$:

$$\begin{bmatrix} 12a + 16b + 4c - 16 \\ 16a + 28b + 16c + 4d - 16 \\ 4a + 16b + 24c + 16d \\ 4b + 16c + 24d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 4 & 1 & 0 \\ 4 & 7 & 4 & 1 \\ 1 & 4 & 6 & 4 \\ 0 & 1 & 4 & 6 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 0 \\ 0 \end{bmatrix}$$

Hausaufgabe: löst dieses LGS (beispielsweise) durch die Anwendung gaußsches Eliminationsverfahrens

Falls ich keine Rechenfehler gemacht hab, solltet ihr die folgende Lösung bekommen:

$$(a, b, c, d) = \left(\frac{40}{9}, -\frac{8}{3}, \frac{4}{3}, -\frac{4}{9}\right) = \frac{1}{9} \cdot (40, -24, 12, -4)$$

- Wir berechnen die übrigs symmetrische **Hesse-Matrix** von $S(a, b, c, d)$:

$$H_S(a, b, c, d) = \begin{bmatrix} \frac{\partial^2 S}{\partial a \partial a} & \frac{\partial^2 S}{\partial a \partial b} & \frac{\partial^2 S}{\partial a \partial c} & \frac{\partial^2 S}{\partial a \partial d} \\ \frac{\partial^2 S}{\partial b \partial a} & \frac{\partial^2 S}{\partial b \partial b} & \frac{\partial^2 S}{\partial b \partial c} & \frac{\partial^2 S}{\partial b \partial d} \\ \frac{\partial^2 S}{\partial c \partial a} & \frac{\partial^2 S}{\partial c \partial b} & \frac{\partial^2 S}{\partial c \partial c} & \frac{\partial^2 S}{\partial c \partial d} \\ \frac{\partial^2 S}{\partial d \partial a} & \frac{\partial^2 S}{\partial d \partial b} & \frac{\partial^2 S}{\partial d \partial c} & \frac{\partial^2 S}{\partial d \partial d} \end{bmatrix} = \begin{bmatrix} 12 & 16 & 4 & 0 \\ 16 & 28 & 16 & 4 \\ 4 & 16 & 24 & 16 \\ 0 & 4 & 16 & 24 \end{bmatrix}$$

Hesse-Matrix einer quadratischen Form ist immer konstant – hier: unabhängig von (a, b, c, d) Werten

- Wir bestimmen die Definitheit unserer Hesse-Matrix an gefundenen Nullstellen (hier: aka **kritischen Stellen**) des LGS $\nabla S(a, b, c, d) = \mathbf{0}$

Eigenwerte der in unserem Fall konstanten Hesse-Matrix sind:

$$\lambda_1 \approx 0.23922, \lambda_2 \approx 6.49262, \lambda_3 \approx 27.68160, \lambda_4 \approx 53.58653$$

Also, **alle E-Werte sind positiv**, d.h. unsere Verlustfunktion $S(a, b, c, d)$ erreicht ihr einziges lokales **Minimum** an der Stelle $(a, b, c, d) = \left(\frac{40}{9}, -\frac{8}{3}, \frac{4}{3}, -\frac{4}{9}\right)$

Also, der beste Kandidat für die Rolle einer Quasi-Faltungsinverse:

$$F^+ = \frac{1}{9} \cdot [-4 \quad 12 \quad -24 \quad 40 \quad -24 \quad 12 \quad -4]$$

Näherungsfehler durch den direkten elementweisen **Vergleich** mit dem **Idealfall** (d.h. bei „echter“ Faltungsinverse):

grausame ☹ Realität	$F * F^+ = \frac{1}{9} \cdot [-1 \quad 1 \quad -1 \quad 1 \quad 8 \quad 1 \quad -1 \quad 1 \quad -1] \approx [-0.11 \quad 0.11 \quad -0.11 \quad 0.11 \quad 0.89 \quad 0.11 \quad -0.11 \quad 0.11 \quad -0.11]$
Idealfall	$F * F^+ = \frac{1}{9} \cdot [-0 \quad 0 \quad -0 \quad 0 \quad 9 \quad 0 \quad -0 \quad 0 \quad -0] = [-0.00 \quad 0.00 \quad -0.00 \quad 0.00 \quad 1.00 \quad 0.00 \quad -0.00 \quad 0.00 \quad -0.00]$

Randbemerkung: die Annäherung kann auch bei der Einschränkung von symmetrischen Faltungsmasken akzeptabel sein.

Aufgabe U9:

Schritt #1: lineare Filter einer anderen Art – **Differenzfilter** aka **Kantenoperatoren** aka **Ableitungsfiler** aka **Gradientenfilter**

- **Koeffizienten** haben **unterschiedliche Vorzeichen** – dadurch sind **Farbintensitätsänderungen** (d.h. u.a. ☺ **Kanten**) ausgeprägter [**Hochpassfilter**]
- **Koeffizienten** **summieren** sich **zu Null** – je **homogener** ein Bildbereich, umso näher sind i.d.R. dessen gefaltete Ausgangswerte zu Null (**zum Schwarz**)
- Der Randbehandlungsmodus „**Zero Padding**“ wird gegenüber „**Last Value**“ (d.h. **Außenbereich**(x, y) = **Bild**(**nächster Randpunkt**)) und „**zyklische Wiederholung**“ (d.h. **Ortsraum**($z \pm [m] \cdot Z, s \pm [n] \cdot S$) = **Bild**(z, s)) Modi **bevorzugt** – sonst können unerwünschte **Artefakte** entstehen (Quelle für **falsche Interpretierung/Erkennung** & **Pseudokanten**)
- sind **extrem rauschempfindlich** – um diese Empfindlichkeit **zu verringern**, man **faltet Differenzfilter mit Glättungsfiltern** [es wird i.d.R. **orthogonal** zu **Ableitungsrichtung** **geglättet**]
- basieren auf **diskrete** (aka **endliche** aka **finite**) **Differenzen**:

Bildquelle: Wiki

diskrete Approximation von $\frac{\partial f}{\partial x}(x, y)$	Faltungsmaske	nachvollziehbare Begründung – kontinuierliches Schema
vorwärts $\Delta_1^1 f_x \stackrel{\text{def}}{=} f(x+1, y) - f(x, y)$	$[0 \quad -1 \quad 1]$	
rückwärts $\nabla_1^1 f_x \stackrel{\text{def}}{=} f(x, y) - f(x-1, y)$	$[-1 \quad 1 \quad 0]$	
symmetrisch ($\frac{\text{vorwärts} + \text{rückwärts}}{2}$) $\delta_1^1 f_x \stackrel{\text{def}}{=} \frac{1}{2} \cdot (f(x+1, y) - f(x, y) + f(x, y) - f(x-1, y))$	$\frac{1}{2} \cdot [-1 \quad 0 \quad 1]$	

Schritt #2: Herleitung & Genauigkeitsbereich von diskreten Approximationen

Für **n-mal differenzierbare** Funktionen $f(X)$ [intuitiverweise erweiterbar auf mehrere Variablen], laut **Taylorformel**:

$$f(X, y) = \sum_{k=0}^n \frac{\partial^k f(x, y)}{(\partial x)^k} \cdot (X - x)^k + \text{Restglied}_n f(X, x),$$

wobei $\lim_{X \rightarrow x} \left| \frac{\text{Restglied}_n f(X, x)}{(X-x)^n} \right| = 0$ – also, in einfachen Worten, das **Restglied** **wächst langsamer** als $|X - x|^n$ (d.h. als **Schrittweite in n-ter Potenz**)

Aufgabe U9:

In unseren Formaten:

$X = x + 1$	$X - x = 1$
$X = x - 1$	$X - x = -1$

Schritt #3: Taylorformel in diesen Formaten

n = 1 ⇒ Genauigkeit
1. Grades (d.h. **niedriger**)

$$f(x + 1, y) \approx f(x, y) + \frac{\partial f(x, y)}{\partial x} \cdot (1)^1 = f(x, y) + \frac{\partial f(x, y)}{\partial x} \Rightarrow \frac{\partial f(x, y)}{\partial x} \approx f(x + 1, y) - f(x, y) \quad \text{vorwärts}$$

$$f(x - 1, y) \approx f(x, y) + \frac{\partial f(x, y)}{\partial x} \cdot (-1)^1 = f(x, y) - \frac{\partial f(x, y)}{\partial x} \Rightarrow \frac{\partial f(x, y)}{\partial x} \approx f(x, y) - f(x - 1, y) \quad \text{rückwärts}$$

n = 2 ⇒ Genauigkeit
2. Grades (d.h. **höher**)

$$f(x + 1, y) \approx f(x, y) + \frac{\partial f(x, y)}{\partial x} \cdot (1)^1 + \frac{1}{2} \cdot \frac{\partial^2 f(x, y)}{(\partial x)^2} \cdot (1)^2 = f(x, y) + \frac{\partial f(x, y)}{\partial x} + \frac{1}{2} \cdot \frac{\partial^2 f(x, y)}{(\partial x)^2}$$

$$f(x - 1, y) \approx f(x, y) + \frac{\partial f(x, y)}{\partial x} \cdot (-1)^1 + \frac{1}{2} \cdot \frac{\partial^2 f(x, y)}{(\partial x)^2} \cdot (-1)^2 = f(x, y) - \frac{\partial f(x, y)}{\partial x} + \frac{1}{2} \cdot \frac{\partial^2 f(x, y)}{(\partial x)^2}$$



$$\frac{1}{2} \cdot (f(x + 1, y) - f(x - 1, y)) \approx \frac{1}{2} \cdot \left(\frac{\partial f(x, y)}{\partial x} - \left(-\frac{\partial f(x, y)}{\partial x} \right) \right) = \frac{1}{2} \cdot 2 \cdot \frac{\partial f(x, y)}{\partial x} = \frac{\partial f(x, y)}{\partial x} \Rightarrow \frac{\partial f(x, y)}{\partial x} \approx \frac{1}{2} \cdot (f(x + 1, y) - f(x - 1, y)) \quad \text{symmetrisch}$$

$$f(x + 1, y) + f(x - 1, y) \approx 2 \cdot f(x, y) + \frac{\partial^2 f(x, y)}{(\partial x)^2} \Rightarrow \frac{\partial^2 f(x, y)}{(\partial x)^2} \approx f(x - 1, y) - 2 \cdot f(x, y) + f(x + 1, y) \quad \text{entspricht der Maske } \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

Zentrale ☺ **Randbemerkung #1:** **y**-Richtungsableitungen 1. & 2. Ordnungen (d.h. **partielle Ableitungen** nach der **zweiten Variable**) werden auf die gleiche Weise behandelt

Zentrale ☺ **Randbemerkung #2:** da es für jede Richtung \vec{r} gilt \vec{r} -Richtungsableitung = $\langle \text{Gradient}, \vec{r} \rangle$, zeigt die **Richtung des Gradientenfeldes** (aka **Gradientenvektors**) zum **steilsten Anstieg** der zugehörigen Funktion (deren Gradient also berechnet wird – in unserem Fall, **der Bildfunktion**), der **Gradientenbetrag** ist dann die **Stärke dieser Steigung**

Hausaufgabe: beweist die Gleichung \vec{r} -Richtungsableitung = $\langle \text{Gradient}, \vec{r} \rangle$



Aufgabe U9:

Schritt #4: einige Beispiele von Differenzenfiltern

beachtet: $\frac{1}{2} \cdot [1 \ 0 \ -1] + \frac{1}{4} \cdot [1 \ 2 \ 1]^T = -\frac{1}{8} \cdot \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \frac{1}{8} \cdot \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$

Nebenbemerkung: $[-1 \ 0 \ 1] = [-1 \ 1] * [1 \ 1]$
[also, bisschen **Glättung** schon mal drin ☺]

Richtungsabhängige (aka anisotrope) Kantenoperatoren		
Glättungsfilter	Differenzenfilter	D * G
Mittelwert- aka Boxfilter	$\frac{1}{3} \cdot [1 \ 1 \ 1]^T$	$\frac{1}{3} \cdot \begin{bmatrix} -1 & 1 & 0 \\ -1 & 1 & 0 \\ -1 & 1 & 0 \end{bmatrix}$
Mittelwert- aka Boxfilter	$\frac{1}{3} \cdot [1 \ 1 \ 1]$	$\frac{1}{3} \cdot \begin{bmatrix} -1 & -1 & -1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$
Binomialfilter	$\frac{1}{4} \cdot [1 \ 2 \ 1]^T$	$\frac{1}{8} \cdot \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \stackrel{\text{def}}{=} \text{Sobel}_x$
Binomialfilter	$\frac{1}{4} \cdot [1 \ 2 \ 1]$	$\frac{1}{8} \cdot \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \stackrel{\text{def}}{=} \text{Sobel}_y$
einigermaßen richtungsunabhängige (aka quasi-isotrope) Kantenoperatoren		
Differenzenfilter _x 2. Ordnung	Differenzenfilter _y 2. Ordnung	Differenzenfilter _x + Differenzenfilter _y
$[1 \ -2 \ 1]$	$[1 \ -2 \ 1]^T$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \stackrel{\text{def}}{=} \text{Laplace}$

Nebenbemerkung: $[1 \ -2 \ 1] = [-1 \ 1] * [-1 \ 1]$

Anmerkungen zu Laplace-Filter:

- Im kontinuierlichen Fall weist der Laplace-Operator die **Rotationsinvarianz** auf – im diskreten Fall wird’s versucht solche **Drehungsinvarianz zu approximieren**
- Laplace** zieht nur 4-Nachbarschaft in Betracht – zwecks Berücksichtigung aller 8-Nachbarn: $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
- Laplace-Filter besitzt **hohe Rausanfälligkeit** – diesbezüglich gibt es einige Modifikationen mit Glättung (zB. **LoG-Filter** – engl. „*Laplacian of Gaussian*“)
- Im Gegensatz zu anderen Kantenoperatoren (von **1. Ordnung**), man kriegt einen **Hinweis auf eine Kante** bei einem **Nulldurchgang** (**Hausaufgabe: warum?**)

Anmerkungen zu Sobelfiltern:





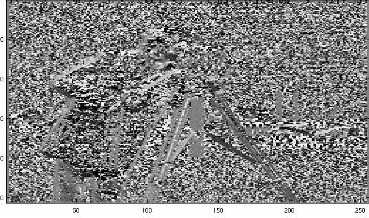
- das **Subskript x** bei Sobel_x bezeichnet die **Ableitungsrichtung**, nicht die Richtung von (am besten) erkennbaren Kanten – **Kantenrichtung ist immer orthogonal zu Ableitungsrichtung** (also, bei Sobel_x sind es **senkrechte Kanten**)
- man kann (nicht unbedingt ☺ **Sobel**)filter auch für die **anderen Ableitungsrichtungen** haben:

Lifhack: Seitenverhältnis (engl. “aspect ratio”) 4Z: 5S entspricht (ungefähr) der Winkelgröße 38°: $\arctan2(4/5) \approx 38^\circ$

$\text{Sobel}_{45^\circ \cup} = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} = \text{Sobel}_{135^\circ \cup}$	$\text{Sobel}_{135^\circ \cup} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} = \text{Sobel}_{-45^\circ \cup}$	$\text{Sobeloid}^{\ominus 120^\circ \cup} = \begin{bmatrix} -1 & -1 & -1 & -1 & \pm 0 \\ -1 & -1 & \pm 0 & \pm 0 & \pm 0 \\ \pm 0 & \pm 0 & \pm 0 & +1 & +1 \\ \pm 0 & +1 & +1 & +1 & +1 \end{bmatrix} = \text{Sobeloid}^{\ominus 30^\circ \cup}$
---	--	---

Hausaufgaben: überprüft ob sich Laplace-Filter separieren lässt (d.h. ob $\text{Laplace} = 1\text{DFilter}_1 * 1\text{DFilter}_2$) & berechnet die **Übertragungsfunktion des Laplace-Filters**

Schritt #5: Anwendungsbeispiele (Kameramann voll im Trend ☺)

$S_x * B$ vertical edge image 	$S_y * B$ horizontal edge image 	Laplace-Filter 
		
$\sqrt{([S_x * B](i,j))^2 + ([S_y * B](i,j))^2}$		$\arctan2\left(\frac{[S_y * B](i,j)}{[S_x * B](i,j)}\right)$

prinzipieller Vergleich von Laplace und Sobel: Nullstellen von Ableitungen 2. Ordnung sind Extremstellen von Ableitungen 1. Ordnung (d.h. Maxima des Gradientenbetrages)

„alternativ“ einfach $|[S_x * B](i,j)| + |[S_y * B](i,j)|$, also die Matrix der komponentenweisen Summe von Beträgen aus $S_x * B$ und $S_y * B$ (Grund: rechnerisch effizienter)

Gradientenrichtung für jedes Pixel (i, j)

lokale Sobel-Maxima = Laplace-Nulldurchgänge

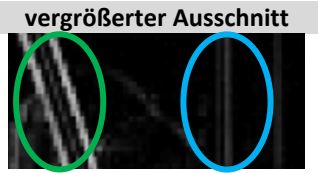
Hausaufgabe: checkt ob die obigen Ergebnisbilder eher von Prewitt-Operator (d.h. $\frac{1}{3} \cdot [1 \ 1 \ 1]^T * \frac{1}{2} \cdot [-1 \ 0 \ 1]$ und $\frac{1}{3} \cdot [1 \ 1 \ 1] * [-1 \ 0 \ 1]^T$) und nicht von Sobel sind [Matlab / OpenCV-Python / ImageJ & Kameramänner ☺ stehen zu eurer Verfügung]

Kritische Anmerkung: beim $S_x * B$ (Bild oben links), warum sind einige schräge Kanten heller als streng vertikale Kanten?



beim Ursprungsbild B liefert 2 Gründe dazu:

- ✓ schräge Kanten: Helligkeitsänderungen beim Übergang in x -Richtung sind stärker
- ✓ schräge Kanten sind zwangsmäßig (wegen pixeliger Natur) vom Treppeneffekt betroffen – sind also stückweise vertikal (der Intensitätsverlauf jedes solchen Stückes hängt vom Glättungsgrad des Bildes ab)



Schritt #5: Lösung der Aufgabe

innere Bildpunkte

Nebenbeobachtung: Minuszeichen führt lediglich zur Inversion von Farbtintensitäten - strukturelles Verhalten wird nicht gestört [Minus ist also von kosmetischen Wirkungsstoffen ☺]

(a)

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix} = \begin{bmatrix} -1 & -4 & -8 & -12 & -16 & -14 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & +4 & +8 & +12 & +16 & +14 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix} = \begin{bmatrix} 3 & 6 & 6 & 6 & 6 & -12 \\ 4 & 8 & 8 & 8 & 8 & -16 \\ 4 & 8 & 8 & 8 & 8 & -16 \\ 4 & 8 & 8 & 8 & 8 & -16 \\ 4 & 8 & 8 & 8 & 8 & -16 \\ 3 & 6 & 6 & 6 & 6 & -12 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -2 & -3 & -4 & -11 \\ 1 & 0 & 0 & 0 & 0 & -6 \\ 1 & 0 & 0 & 0 & 0 & -6 \\ 1 & 0 & 0 & 0 & 0 & -6 \\ 1 & 0 & 0 & 0 & 0 & -6 \\ 1 & -1 & -2 & -3 & -4 & -11 \end{bmatrix}$$

Beispiel von Randartefakten

Nebenbeobachtung: Ausgangswerte (auch wenn negativ) werden verhältnismäßig umskaliert und in $[0; Max]$ eingeschlossen (entsprechend der Farbtiefe des Bildformats)

(c)

Ausführliche Details zu Richtung: Werte -1 entsprechen der Differenzmaske $\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$ (zunächst war es $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$, die Faltung $S_x * B$ musste aber die Maske definitionsmäßig umdrehen) – deshalb heißt es „rechte Pixelwerte von linken abziehen“

Gradientenbetrag: $\frac{1}{8} \cdot \begin{bmatrix} \sqrt{0^2 + (-8)^2} & \dots & \dots & \sqrt{0^2 + (-8)^2} \\ \sqrt{0^2 + (-8)^2} & \dots & \dots & \sqrt{0^2 + (-8)^2} \\ \sqrt{0^2 + (-8)^2} & \dots & \dots & \sqrt{0^2 + (-8)^2} \\ \sqrt{0^2 + (-8)^2} & \dots & \dots & \sqrt{0^2 + (-8)^2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

Erläuterung: Einsen-Matrix bedeutet dass Intensitätsänderungen überall auf dem Bild konstant sind und die Änderungsgröße eine Intensitätsstufe beträgt

Gradientenrichtung: $\begin{bmatrix} atan2(0/-1) & \dots & \dots & atan2(0/-1) \\ atan2(0/-1) & \dots & \dots & atan2(0/-1) \\ atan2(0/-1) & \dots & \dots & atan2(0/-1) \\ atan2(0/-1) & \dots & \dots & atan2(0/-1) \end{bmatrix} = \begin{bmatrix} \pi & \pi & \pi & \pi \\ \pi & \pi & \pi & \pi \\ \pi & \pi & \pi & \pi \\ \pi & \pi & \pi & \pi \end{bmatrix}$

Erläuterung: π -Matrix deutet die waagerechte Richtung der größten Intensitätsänderungen überall auf dem Bild an

Bildanalyse und Bildverstehen		Georg-August-Universität Göttingen SoSe 21
Fragen/Anregungen zum Stoff? ⇒ 24/7-Support ☺ unter jeos@mail.com emailen		
Lösungen #04	Prof. Winfried Kurth / Alex Tavkheldze	

Aufgabe U10:

Schritt #1: Lösung der Aufgabe

siehe Seiten 12 & 13
