Intelligente Sehsysteme – Übungsblatt 3

PD. Dr. W. Koch, Dr. F. Govaers, H. Hoelzemann

*E-Mail: {wolfgang.koch, felix.govaers, **henry.hoelzemann**}@fkie.fraunhofer.de*

Abgabe bis Sonntag, 09.11.2025, 12:00 Uhr in Gruppen von 3 Personen.

1. Scharr-Operator (2P)

Es gibt alternative Operatoren zum Sobel-Operator. So zeigt der Scharr-Operator eine etwas bessere Rotationssymmetrie. Der 3×3-Scharr-Operator zeigt die beiden folgenden Filter für die horizontale und vertikale Gradientenapproximation:

$G_h :=$

47	0	-47
162	0	-162
47	0	-47

$G_v :=$

47	162	47
0	0	0
-47	-162	-47

Völlig analog zur Anwendung des Sobel-Operators wenden Sie diesen Scharr-Operator auf die **zentralen neun Pixel** (kursiv gekennzeichnet) im folgenden 5×5-Grauwertbild:

$I :=$

0	0	0	100	100
0	*100*	*100*	*100*	100
0	*100*	*100*	*100*	0
100	*100*	*100*	*100*	0
100	100	0	0	0

A. Berechnen Sie die horizontalen Gradienten durch Korrelation. (0,5P)

B. Berechnen Sie die vertikalen Gradienten durch Korrelation. (0,5P)

C. Berechnen Sie die Gradientenbeträge. (0,5P)

D. Berechnen Sie die Gradientenorientierungen. (0,5P)

Aufgabe 1:

A:

$$S(1,3) = \sum_{i=-1}^{2} \sum_{j=-1}^{2} g_L(i,j) \cdot I(1+i, 3+j) = -20900$$

$$\begin{array}{ccc} -20900 & -4700 & 0 \\ -20900 & 0 & 20900 \\ 0 & 4700 & 20900 \end{array}$$

$S(2,3) = -4700$

$S(3,3) = 0$

$S(1,2) = -20900$

$S(2,2) = 0$

$S(3,2) = 20900$

$S(1,1) = 0$

$S(2,1) = 4700$

$S(3,1) = 20900$

B

$$S(1,3) = \sum_{i=-1}^{2} \sum_{j=-1}^{2} g_L(i,j) \cdot I(1+i, 3+j) = -20900$$

$S(2,3) = -20900$

$S(3,3) = 0$

$S(1,2) = -4700$

$S(2,2) = 0$

$S(3,2) = 4700$

$S(1,1) = 0$

$S(2,1) = 20900$

$S(3,1) = 20900$

C

$S(1,3) = \|S_x(1,3) + S_y(1,3)\|_2 = 29557$

$S(2,3) = \|S_x(2,3) + S_y(2,3)\|_2 = 21422$

$S(3,3) =$ " — " $= 0$

$S(1,2) =$ " — " $= 21422$

$S(2,2) =$ " — " $= 0$

$S(3,2) =$ " — " $= 21422$

$S(1,1) =$ " — " $= 0$

$S(2,1) =$ " — " $= 21422$

$S(3,1) =$ " — " $= 29557$

D) $\arctan(1) = 45$

$\arctan(4700/22489) = 12,6738$

$\arctan(0/0) =$ undefined

$\arctan(-20900/-4700) = 77,326$

$\arctan(0/0) =$ undefined

$\arctan(20900/4700) = 77,326$

$\arctan(0/0) =$ undefined

$\arctan(4700/22488) = 12,6738$

$\arctan(20900/20900) = 45$

2. Sobel-Operator in Python (5P)

Implementieren Sie den Sobel-Operator für ein Graustufenbild. Visualisieren Sie den Gradientenbetrag, die Gradientenorientierung (mit 4-Bin-Binning; siehe Folie 17 der 3. Vorlesung) und beides kombiniert über eine kleine GUI. Die Aufgabe ist in folgende Teilschritte unterteilt:

A. Implementieren Sie eine Funktion, die aus einem Graustufenbild die Gradienten G_x, G_y (3×3 Sobel-Kernel), den Gradientenbetrag

$$M = \sqrt{G_x^2 + G_y^2}$$

und die Orientierung

$$\theta = \operatorname{atan2}(G_y, G_x)$$

berechnet. Strecken Sie den Betrag linear auf den vollen 8-Bit-Bereich $[0, 255]$ (Min–Max-Stretching), und testen Sie Ihre Implementierung mit:

- `Lena 512×512.png` (zuvor in Graustufen konvertieren)
- `Werkzeuge 768×576.png`

Fügen Sie die erzeugten Ergebnisbilder Ihrer Abgabe bei. (2P)

B. Erweitern Sie Ihre Implementierung so, dass wahlweise nur die Beträge *oder* nur die Orientierungen angezeigt werden. Verwenden Sie für die Orientierungen ein 4-Bin-Binning mit den Mittelpunkten 0°, 45°, 90°, 135° (jeweils ±22.5°[1]). Achten Sie auf die korrekte Behandlung negativer Winkel von `atan2` (z. B. $-45° \equiv 315°$) und die Reduktion auf den Bereich $[0°, 180°)$. Fügen Sie ein Beispielbild der Gradientenorientierungen Ihrer Abgabe bei. (2P)

C. Wenn Betrag *und* Orientierung implementiert sind, gestalten Sie eine kleine GUI (z. B. mit OpenCV HighGUI) mit drei Modi:

(a) Nur Gradientenbeträge

(b) Nur Gradientenorientierungen (nur dort, wo Betrag $\geq$ wählbarer Schwellwert)

(c) Kombination: Farbe = Orientierung, Helligkeit proportional zum Betrag

Stellen Sie sicher, dass der Schwellwert der Gradientenorientierungen während der Laufzeit verändert werden kann. (1P)

[1] Siehe Vorlesung 3, Folie 17

3. Laplace-Operator (1P)

Für eine zweimal stetig partiell differenzierbare Funktion f gilt:

$$\nabla^2 f = \sum_{k=1}^{n} \frac{\partial^2}{\partial x_k^2}$$

Berechnen Sie den Laplace-Ausdruck für folgende Funktionen:

A. $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$

B. $f(x_1, x_2, x_3) = x_1^2 \cdot x_2^2 \cdot x_3$

Aufg 3:

A: $\nabla^2 f := \sum_{k=1}^{n} \frac{\partial^2}{\partial x_k^2} = 2+2+2$

B: $\nabla^2 f = \sum_{k=1}^{n} \frac{\partial^2}{\partial x_k^2} = 2x_2^2 x_3 + 2x_1^2 x_3 + 0$

$$\left(\begin{array}{l} f'(x) = \underbrace{2x_1 x^2 x_3}_{x_1'} + \underbrace{2x_2 x_1^2 x_3}_{x_2'} + \underbrace{x_1^2 x_2^2}_{x_3'} \\ f''(x) = 2x_2^2 x_3 + 2x_1^2 x_3 + 0 \end{array} \right)$$

4. Laplace-Filter in Python (2P)

Implementieren Sie ein Laplace-Filter in Python. Gehen Sie auch hier wieder von einem Graustufenbild als Eingabe aus. Leiten Sie die Positionen der Kanten ab, wie auf Folie 36 von Vorlesung 3 beschrieben. Liegt ein Nulldurchgang vor, schreiben Sie das Ergebnis der Anwendung der Filtermaske in das Ergebnisbild, sonst setzen Sie den Wert auf 0. Führen Sie anschließend eine lineare Streckung des Ergebnisbildes über den gesamten Intensitätsbereich durch. Durch Verwendung der Filter-Eigenschaften sollen sowohl die L_4- als auch die L_8-Variante per GUI wählbar sein. Wenden Sie Ihr Laplace-Filter auf folgende Bilder an:

- `Lena 512×512.png` (zuvor in Graustufen konvertieren)
- `Werkzeuge 768×576.png`

Fügen Sie die Ergebnisbilder Ihrer Lösungsabgabe bei.