

# Intelligente Sehsysteme – Übungsblatt 4

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

E-Mail: [{wolfgang.koch, felix.govaers, henry.hoelzemann, florian.osswald}@fkie.fraunhofer.de](mailto:{wolfgang.koch, felix.govaers, henry.hoelzemann, florian.osswald}@fkie.fraunhofer.de)

**Abgabe bis Sonntag, 16.11.2025, 12:00 Uhr in Gruppen von 3 Personen.**

## 1 Medianfilter (1P)

Implementieren Sie das Medianfilter in Python. Wenden Sie das Medianfilter auf die Bilder `Testbild_Lena_512x512.png` und `Testbild_Rauschen_640x480.png` an, die Sie von eCampus herunterladen können. Fügen Sie die Ergebnisbilder Ihrer Lösung als Anlage bei.

## 2 Diffusionsfilter (5P)

Implementieren Sie ein isotropes inhomogenes Diffusionsfilter in Python, wobei Sie von einem Grauwertbild als Eingabe ausgehen können. Der benötigte Parameter  $\epsilon_0$  sowie die Anzahl der Iterationen sollen vom Benutzer wählbar sein. Gehen Sie dabei in jeder Iteration wie folgt vor:

- A. Schätzen Sie zunächst die Gradienten  $\frac{\partial I(x,y)}{\partial x}$  sowie  $\frac{\partial I(x,y)}{\partial y}$  durch Differenzen in x- bzw. y-Richtung wie auf Folie 24 von Vorlesung 4 vorgeschlagen.
- B. Erstellen Sie daraus den Diffusionstensor laut Folie 29 in Vorlesung 4 und berechnen für jeden Pixel den Fluss  $\vec{j}(x,y)$  laut Folie 20.
- C. Schätzen Sie nun die Gradienten  $\frac{\partial j_x(x,y)}{\partial x}$  und  $\frac{\partial j_y(x,y)}{\partial y}$ , indem Sie die Differenzen des Flusses  $\vec{j}(x,y)$  in x- bzw. y-Richtung betrachten – wie auf Folie 24 von Vorlesung 4 vorgeschlagen.
- D. Die Divergenz an den Koordinaten  $(x,y)$  ergibt sich nun als

$$\operatorname{div} \vec{j}(x,y) = \frac{\partial j_x(x,y)}{\partial x} + \frac{\partial j_y(x,y)}{\partial y} \quad (1)$$

- E. Die Ergebnisintensität des Pixels  $(x,y)$  in Iteration  $i$  ergibt sich durch

$$I(x,y,t_{i-1}) - \operatorname{div} \vec{j}(x,y,t_i) \quad (2)$$

Testen Sie Ihr Filter auf `Testbild_Rauschen_640x480.png`, das Sie von eCampus herunterladen können mit  $\epsilon_0 = 1$  und 500 Iterationen. Fügen Sie das Ergebnisbild Ihrer Lösung als Anlage bei.

### 3 Vertiefende Überlegungen (2P)

Wendet man das Medianfilter aus Aufgabe 1 sowie ein isotropes inhomogenes Diffusionsfilter auf das Bild `Testbild_Rauschen_640x480.png` an, so erhält man als Ergebnisse die beiden Bilder `Testbild_Rauschen_640x480_Median.png` bzw.

`Testbild_Rauschen_640x480_Diffusion.png`, die Sie ebenfalls auf eCampus finden.

- A. Welche visuellen Unterschiede bemerken Sie?
- B. Schreiben Sie ein Filter in Python, das die pixelweise Differenz zwischen zwei Eingabebildern berechnet und anschließend eine lineare Streckung auf den gesamten Intensitätsbereich durchführt. Entspricht das Ergebnis Ihren Erwartungen? Wie können Sie die Unterschiede zwischen den Bildern erklären?

### 4 Tensorberechnung für anisotropes inhomogenes Diffusionsfilter (2P)

Gegeben sei der folgende  $5 \times 5$ -Grauwertbilddausschnitt:

10	10	10	10	20
10	10	10	20	20
10	10	20	20	20
10	20	20	20	20
20	20	20	20	20

Der Diffusionstensor wird durch Eigenwertzerlegung ermittelt (Folien 32 ff.)

- A. Berechnen Sie mit expliziter Herleitung die beiden Eigenvektoren und Eigenwerte für die Anwendung des Diffusionstensors im zentralen Pixel des obigen Ausschnitts. Die Intensitätsgradienten in x- und y-Richtung sind dabei durch Differenzen der Intensitäten zu approximieren:  $I(x+1, y) - I(x-1, y)$  bzw.  $I(x, y+1) - I(x, y-1)$ . (1,0 P)
- B. Berechnen Sie mit expliziter Herleitung den resultierenden Diffusionstensor für die Anwendung im zentralen Pixel des obigen Ausschnitts. Setzen Sie dabei der Einfachheit halber  $\epsilon_0 = 1$  und  $\lambda = 1$ . (0,5 P)
- C. Begründen Sie, warum der resultierende Diffusionstensor positiv definit ist. (0,5 P)