
Intelligente Sehsysteme – Übungsblatt 1

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

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

Abgabe bis Sonntag, 26.10.2025, 12:00 Uhr in Gruppen von 3 Personen.

1 OpenCV: Installation

Installieren Sie `Python 3.11` von der offiziellen Python-Website¹ und richten Sie Ihre Entwicklungsumgebung ein. Die Programmieraufgaben im Modul „Intelligente Sehsysteme“ bauen auf der Bibliothek `OpenCV`² auf, welche wiederum von `NumPy`³ Gebrauch macht. Zur Visualisierung von Diagrammen kann `Matplotlib`⁴ verwendet werden. Informationen zur Installation und Verwendung der benötigten Bibliotheken `OpenCV`, `NumPy` und `Matplotlib` finden Sie auf deren jeweiligen offiziellen Seiten.

Es wird empfohlen, die Installation der Bibliotheken mit einem Paketmanager wie `pip` oder `conda` durchzuführen. Achten Sie darauf, dass alle Bibliotheken erfolgreich installiert sind, bevor Sie mit den Programmieraufgaben beginnen. Führen Sie das Skript `opencv_test.py` aus, um die Installation der Bibliotheken zu überprüfen. Es sollte ein schwarzes Bild mit dem Schriftzug „ISS“ in Blau zu sehen sein.

Verschaffen Sie sich einen Einblick in die grundlegende Funktionsweise von `OpenCV`, insbesondere in die Erstellung und Anwendung von Filtern⁵.

Hinweis: Bei Abgaben von Programmierübungen müssen alle beteiligten Autoren explizit genannt werden. Außerdem müssen der Namen der abzugebenden Dateien so gewählt sein, dass sie Ihre Initialen enthalten und eindeutig angeben, zu welcher Aufgabe sie gehören. Zum Beispiel sollte die Code-Datei für dieses Übungsblatt von „John Doe“ und „Mary Jane“ folgendermaßen benannt werden: *sheet01_JD_MJ.py*. Bitte behalten Sie diese Konvention auch bei allen folgenden Übungsblättern bei.

2 Selektive Farbverbesserung (1P)

Viele Bildbearbeitungsprogramme bieten die Möglichkeit, ausgewählte Farben oder Farbbereiche im Vergleich zu anderen Farben zu verstärken. Oft wird dabei die Sättigung des Farbbereichs erhöht. Hier wollen wir uns aber auf einen der drei RGB-Kanäle und deren Intensität beschränken.

¹<https://www.python.org/downloads/>

²<https://opencv.org/get-started/>

³<https://numpy.org/install/>

⁴<https://matplotlib.org/stable/install/index.html>

⁵https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html



(a) Lena_512x512.png



(b) Maske_Lena_512x512.png

Betrachten Sie als einführendes Beispiel das Skript `checkerboard_example.py`. Nutzen Sie dieses als Grundlage, um auf einzelne Pixel und dessen RGB-Kanäle in OpenCV und NumPy zuzugreifen.

Implementieren Sie anschließend ein Filter, das zu einem RGB-Farbbild $I_{\text{RGB}} = [I(x, y, k)]$ den Kanal $k \in \{R, G, B\}$ wie folgt verstärkt:

$$I'(x, y, k) = a \cdot I(x, y, k)$$

für alle Pixelkoordinaten (x, y) und mit $a \in [0, 10]$, $k \in \{R, G, B\}$. Der Parameter a und der Kanal k sollen als Parameter an die Funktion übergeben werden können. Ein Anfang dieser und folgender Implementierungen ist in der Datei `sheet01.py` gegeben.

Wenden Sie Ihr Filter auf das Testbild `Lena_512x512.png` an. Machen Sie dabei **keinen** Gebrauch von bestehenden OpenCV-Funktionen (mit Ausnahme von Funktionen zum Einlesen, Anzeigen und Speichern von Bildern).

Bonus (0.5P): Der Ansatz im Beispiel-Skript `checkerboard_example.py` verdeutlicht den Zugriff auf einzelne Pixel, ist jedoch mit zwei verschachtelten `for loops` in Python nicht performant. Finden Sie eine Lösung für die selektive Farbverbesserung, welche ohne Schleife auskommt.

3 Maskierung (1P)

Die Maskierung von Signalen ist ein wichtiges Werkzeug in der Bildverarbeitung. Hierbei wird ein Eingabebild mit einem anderen Bild „maskiert“, um spezifische Regionen im Bild auszublenden und den restlichen Teil durchzulassen. Die Maskierung eines RGB-Bildes $I_{\text{RGB}} \in \{0, \dots, I_{\text{max}}\}^{W \times H \times 3}$ mit einer binären Maske $M \in \{0, 1\}^{W \times H}$ kann wie folgt aussehen:

$$I'_{\text{RGB}}(x, y, k) = I_{\text{RGB}}(x, y, k) \cdot M(x, y)$$

Implementieren Sie ein Filter, das mithilfe von zwei Eingabebildern eine Maskierung vornimmt und das maskierte Bild zurückgibt.

Wenden Sie Ihr Filter auf das Testbild `Lena_512x512.png` mit dem beiliegenden Bild `Maske_Lena_512x512.png` an. Machen Sie dabei kein Gebrauch von bestehenden OpenCV-Funktionen (mit Ausnahme von Funktionen zum Einlesen, Anzeigen und Speichern von Bildern).

4 Interpretation von Histogrammen (2P)

Zur einfachen Rechnung seien die beiden folgenden 4×4 -Grauwertbilder $I_1 = [I_1(x, y)]$ und $I_2 = [I_2(x, y)]$ mit Intensitätsspektrum $\{0, 1, 2, \dots, 7\}$ gegeben:

$$I_1 = \begin{bmatrix} 4 & 2 & 5 & 1 \\ 5 & 3 & 2 & 3 \\ 4 & 2 & 6 & 2 \\ 4 & 1 & 2 & 1 \end{bmatrix} \quad I_2 = \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 7 & 6 & 1 \\ 0 & 6 & 6 & 2 \\ 1 & 1 & 2 & 1 \end{bmatrix}$$

- Berechnen Sie die unnormalisierten Intensitätshistogramme $h(I_1)$ sowie $h(I_2)$ und stellen Sie diese tabellarisch dar.
- Berechnen Sie die normalisierten Intensitätshistogramme $p(I_1)$ sowie $p(I_2)$ und stellen Sie diese tabellarisch dar.
- Berechnen Sie die Mittelwerte m_{I_1} und m_{I_2} sowie die mittleren quadratischen Abweichungen q_{I_1} und q_{I_2} .
- Welche vergleichenden Aussagen sind über die Bilder I_1 und I_2 anhand ihrer Mittelwerte und mittleren quadratischen Abweichungen ableitbar? Was ist bzgl. beider Werte für I_2 kritisch zu bedenken?

5 Lineare Histogrammspreizung (2P)

Zur einfachen Rechnung sei das folgende 4×2 -Grauwertbild $I = [I(x, y)]$ mit Intensitätsspektrum $\{0, 1, 2, \dots, 7\}$ gegeben:

$$I = \begin{bmatrix} 2 & 3 & 3 & 5 \\ 2 & 4 & 4 & 5 \end{bmatrix}$$

- Berechnen Sie das unnormalisierte Intensitätshistogramm $h(I)$ und stellen Sie dies tabellarisch dar.
- Berechnen Sie das normalisierte Intensitätshistogramm $p(I)$ und stellen Sie dies tabellarisch dar.
- Berechnen Sie mit Herleitung die beiden Konstanten c_1 und c_2 für die lineare Histogrammspreizung $T(I)$.
- Wenden Sie die so ermittelte lineare Histogrammspreizung T auf das obige 4×2 -Grauwertbild $I = [I(x, y)]$ an und geben Sie das so gespreizte neue Grauwertbild $I' = T(I)$ wieder.

6 Gamma-Korrektur (1P)

Zur einfachen Rechnung sei das folgende 4×2 -Grauwertbild $I = [I(x, y)]$ mit Intensitätsspektrum $\{0, 1, 2, \dots, 7\}$ gegeben:

$$I = \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 4 \\ \hline 0 & 2 & 2 & 4 \\ \hline \end{array}$$

- a) Berechnen Sie die Gamma-Korrektur für $I = [I(x, y)]$ mit $\gamma = 0.5$ und geben Sie das so korrigierte neue Grauwertbild $I' = T_{\gamma=0.5}(I)$ wieder.
- b) Berechnen Sie die Gamma-Korrektur für $I = [I(x, y)]$ mit $\gamma = 2.0$ und geben Sie das so korrigierte neue Grauwertbild $I' = T_{\gamma=2.0}(I)$ wieder.
- c) Welche der beiden Gamma-Korrekturen ($\gamma = 0.5$, $\gamma = 2.0$) ist angemessen? Begründen Sie Ihre Antwort einerseits mit der Qualität des Eingabebildes und andererseits mit der Eigenschaft der jeweiligen Gamma-Korrektur.

7 Histogrammlinearisierung bzw. Maximierung der Entropie (1P)

Zur einfachen Rechnung sei das folgende 4×2 -Grauwertbild $I = [I(x, y)]$ mit Intensitätsspektrum $\{0, 1, 2, \dots, 7\}$ gegeben:

$$I = \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 7 \\ \hline 2 & 6 & 6 & 7 \\ \hline \end{array}$$

- a) Berechnen Sie das unnormalisierte Intensitätshistogramm $h(I)$ und stellen Sie dies tabellarisch dar.
- b) Berechnen Sie das normalisierte Intensitätshistogramm $p(I)$ und stellen Sie dies tabellarisch dar.
- c) Berechnen Sie das kumulative Intensitätshistogramm $s(I)$ und stellen Sie dies tabellarisch dar.
- d) Wenden Sie nun die Histogrammlinearisierung T_H auf das obige 4×2 -Grauwertbild $I = [I(x, y)]$ an und geben Sie das so korrigierte neue Grauwertbild $I' = T_H(I)$ wieder.