

## 1. Übung

### Farbbilder und Spiegelung

In dieser Übung sollen Sie sich mit den Python Bibliotheken scikit-image und Numpy vertraut machen. Inhalt der Übung ist neben der Einarbeitung in die IDE Ihrer Wahl (PyCharme, VSCode, Jupyter-Notebook, Spyder), das Einlesen eines Bildes mithilfe von scikit-image und die Implementierung einer ersten eigenen Funktion zur Spiegelung eines Bildes.

---

1. Starten Sie die IDE und legen Sie ein neues Projekt auf dem Z-Laufwerk an.
2. Lesen Sie ein Bild mit Hilfe von scikit-image ein und lassen Sie sich das Bild anzeigen.
3. Untersuchen Sie den Datentyp des eingelesenen Bildes. Wie ist ein Numpy Array aufgebaut?
4. Lassen Sie sich die drei Farbkanäle eines Bildes getrennt anzeigen.
5. Implementieren Sie eine Funktion, die ein eingelesenes Bild wahlweise horizontal oder vertikal spiegelt. (Verwenden Sie hierfür keine Numpy Funktionen.)

### Histogramme, Binning und Lookup-Tabellen

In dieser Übung werden Sie das Histogramm eines Bildes berechnen. Sie werden außerdem lineares Binning zur Histogramm Erstellung implementieren. Des Weiteren werden Sie mithilfe einer Lookup-Tabelle Ihre erste Punktoperation auf einem Bild anwenden.

---

1. Laden Sie das Buch "Digitale Bildverarbeitung" aus Moodle herunter. Lesen Sie die Kapitel 4 - 5 (Histogramme und Punktoperationen).
2. Lesen Sie die Bilder mit Hilfe von scikit-image ein.
3. Schreiben Sie eine Funktion, die das Histogramm eines 8-Bit-Graustufenbildes (256 Intensitätsstufen) berechnet.
  - Zunächst muss eine Funktion geschrieben werden, die ein RGB-Bild in ein Graustufenbild (mit nur einem Kanal) umwandelt.
  - **Prototyp:** `histo = computeHisto(image) # histo as numpy array of size 256`
  - Das Histogramm kann mithilfe der Bibliothek *matplotlib* angezeigt werden. Das Anzeigen eines Histogramms (in Form eines 1-dimensionalen *Numpy Arrays*) ist in *template.py* bereits implementiert.

4. Berechnen Sie das Histogramm von den Bildern 01 bis 05. Sie werden zum einen Aufnahmefehler (Belichtungsfehler) in deren Histogrammen erkennen können. Zum anderen werden Sie die Bearbeitungsschritte in zwei bearbeiteten Bild anhand des Histogramms erkennen können.
  - a) Welche Aufnahmefehler sind in 01 und 03 zu erkennen? Woran ist dies im Histogramm erkennbar?
  - b) Bild01 ist das aufgenommene Bild. Bild02 wurde nachbearbeitet. Die Helligkeit wurde erhöht. Woran ist dies im Histogramm erkennbar? Welche Daten gehen dabei verloren?
  - c) Bild04 ist das aufgenommene Bild. Bild05 wurde einem Bearbeitungsschritt unterzogen. Was wurde in Bild05 verändert? Woran kann man dies in seinem Histogramm erkennen?
5. Implementieren Sie eine Funktion, die eine Punktoperation mithilfe einer Lookup-Tabelle zum Aufhellen eines Bildes durchführt.
  - a) In Aufgabe (4b) gingen Daten beim Aufhellen eines Bildes verloren. Wie könnte dies vermieden werden?
  - b) Damit beim Aufhellen von Bild01 keine Daten verloren gehen, soll eine Lookup-Tabelle verwendet werden. Versuchen Sie mit der Lookup-Tabelle die dunklen Bildbereiche des Bildes aufzuhellen ohne die hellen Bereiche zu stark zu verändern.

### **Abgabe**

Die Aufgaben werden per Git-Tag (<https://git.ios.htwg-konstanz.de> ) bis jeweils zur kommenden Übungsstunde abgegeben. Zudem müssen die Lösungen in der nächsten Übungsstunde mündlich präsentiert werden. Es ist nicht nötig einen eigenen Branch pro Aufgabe zu erstellen.

### **Bild Referenzen**

<http://weburbanist.com/2011/09/21/incredible-rgb-art-shifts-as-lighting-colors-change/>

<http://www.manuela-kerkhoff.de/hidden-in-the-woods-rgb-technique/>

<http://100-beste-plakate.de/plakate/ikea-rgb-billboard/>