

Aufgabenblatt 2

Übung zu Einführung in die Bildverarbeitung

Laszlo Korte, Thang Le, Juri Schalück, Maris Hillemann, Tim Rolff, Christian Wilms
SoSe 2023

Ausgabe: 28. April 2023 - **Abgabe bis: 8. Mai 2023, 10:00**

Abgabe per Moodle

Juri Schalück (Gruppe 2)	0schalue@informatik.uni-hamburg.de
Thang Le (Gruppe 3)	phuoc.thang.le@uni-hamburg.de
Laszlo Korte (Gruppe 4)	9korte@informatik.uni-hamburg.de
Maris Hillemann (Gruppe 5, 6)	maris.nathanael.hillemann@studium.uni-hamburg.de
Tim Rolff (Gruppe 1)	tim.rolff@uni-hamburg.de
Christian Wilms (Gruppe 1)	christian.wilms@uni-hamburg.de

Aufgabe 1 — Basisoperationen mit Bildern - $5+5+10+5+15=40$ Punkte - Programmieraufgabe
Erlaubte (Sub-)Pakete: `numpy`, `matplotlib`, `skimage.io`

Diese Aufgabe dient dem ersten Kontakt mit Bildern in Python. Dazu sollt ihr ein gegebenes Bild manipulieren und alle Schritte in einem Python-Skript festhalten.

1. Zunächst soll ein Bild geladen, angezeigt und ein Ausschnitt gespeichert werden.
 - a) Speichert euch das Bild `astronaut.png` aus dem Moodle ab und ladet es in Python mithilfe der Funktion `skimage.io.imread`.
 - b) Zeigt das Bild mithilfe von Matplotlib an.
 - c) Extrahiert den Bildausschnitt des Abzeichens als neues Array, das, wie in Abbildung 1a visualisiert, etwa das gesamte Abzeichen der Astronautin beinhaltet.
 - d) Speichert den Bildausschnitt mit der Funktion `skimage.io.imsave` ab.
2. Nun soll das Bild manipuliert werden.
 - a) Erstellt eine Kopie des Bildes und setzt ein einziges Pixel (Arrayelement bzw. Bildelement) auf schwarz. Lasst euch das manipulierte Bild anzeigen. Erkennt man den Unterschied? (kein Antworttext nötig)
 - b) Erstellt eine weitere Kopie des Bildes und setzt den gesamten Bereich des Helms auf schwarz. Lasst euch das manipulierte Bild erneut anzeigen. Erkennt man nun den Unterschied? (kein Antworttext nötig)
3. Erstellt ein neues Bild basierend auf drei Variationen des Bildes `astronaut.png` sowie dem Originalbild.
 - a) Spiegelt das Originalbild (`astronaut.png`) an der vertikalen Achse (linke und rechte Seite tauschen) und speichert das Ergebnis in eine neue Variable als erste Variation.
 - b) Spiegelt das Originalbild nun an der horizontalen Achse und speichert das Ergebnis erneut in einer neuen Variable. Dies ist die zweite Variation.
 - c) Führt jetzt beide Spiegelungen nacheinander auf dem Originalbild aus speichert das Ergebnis wiederum in einem neuen Bild als dritte Variation.
 - d) Erzeugt nun ein neues Bild, das doppelt so hoch und breit ist wie das Bild `astronaut.png`. Setzt in jeden Quadranten des neuen Bildes eine der vier Versionen des Bildes ein. Unten links soll das Originalbild zu sehen sein, rechts daneben das an der vertikalen Achse gespiegelte Bild, links oben das an der horizontalen Achse gespiegelte Bild und rechts oben das an beiden Achsen gespiegelte Bild. Das Ergebnis sollte etwa wie in Abbildung 1b aussehen.
Tipp: Zum Erzeugen von Arrays bzw. Bildern gibt es in NumPy mehrere Funktionen neben `np.array`.

4. Schneidet erneut das Abzeichen wie in Abbildung 1a dargestellt als neues Bild aus. Ändert nun in dem neu erzeugten Ausschnitt ein Pixel und findet heraus, ob diese Änderung auch im Originalbild durchgeführt wird.
5. Erstellt nun eine Maske für den Bereich des Raumschiffs auf dem Abzeichen.
 - a) Erzeugt dazu zunächst ein neues Bild, das nur aus Nullen besteht, und die gleiche Größe wie das Originalbild hat. Dieses Bild wird eure Maske.
 - b) Setzt nun in der Maske alle Pixel, die das Raumschiff beinhaltet, auf 1.
 - c) Verrechnet anschließend Maske und Originalbild, sodass der Bereich außerhalb des Raumschiffs schwarz ist und nur im Bereich des Raumschiffs das eigentliche Bild zu sehen ist. Benutzt für diese Operation keine Schleife! Das Ergebnis sollte etwa wie in Abbildung 1c aussehen.

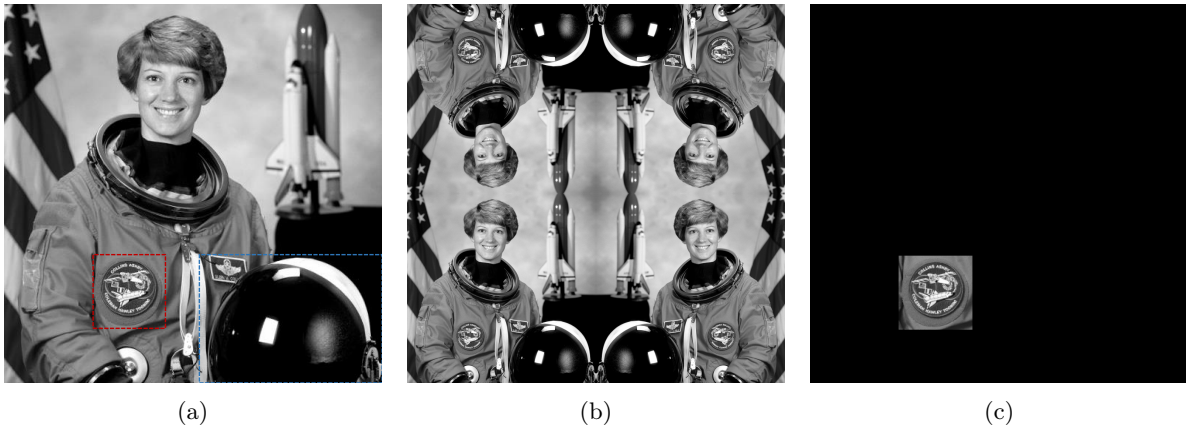


Abbildung 1: **(a)**: Bildausschnitt des Abzeichens aus dem Originalbild (links) als neues Bild. **(b)**: Komposition eines neuen Bildes aus vier, teilweise gespiegelten Originalbildern. **(c)**: Maskiertes Originalbild in dem nur das Abzeichen übrig ist.

Aufgabe 2 — Abtasttheorem - 10+5+10+10=35 Punkte - Theorieaufgabe

Ihr sitzt an Bord eines Propellerflugzeugs am Fenster mit bester Sicht auf einen der Propeller und könnt dessen Rotation klar und korrekt wahrnehmen. Nun wollt ihr die Rotation des Propellers mit eurer Kamera in einem Video festhalten. Ist dies aus Sicht der Signalverarbeitung eine sinnvolle Idee? Beantwortet dazu folgende Fragen schriftlich mit kurzer Begründung. Nehmt dazu an, dass sich der Propeller pro Minute 1600 mal um die eigene Achse dreht und eure Kamera eine Bildrate von $\frac{1}{30}$ s hat, d.h., alle $\frac{1}{30}$ s ein vollständige Bild aufnimmt. Nehmt weiter an, dass eure Belichtungszeit infinitesimal klein ist.

1. Ist das Abtasttheorem im Sinne der zeitlichen Abtastung der Propellerdrehung durch die Kamera erfüllt? Rechnet es nach!
2. Was wird schätzungsweise auf dem Video erkennbar sein und was nicht?
3. Skizziert nun grob die Propellerstellung des zweiten, dritten und vierten Einzelbildes des Videos. Geht dabei davon aus, dass der Propeller zum Zeitpunkt des ersten Einzelbildes so positioniert ist, wie in Abb. 2a dargestellt und genau einen roten Propellerblatt hat. Geht zudem davon aus, dass der Propeller sich im Uhrzeigersinn dreht. Achtet dabei vor allem auf die Positionierung des rot gefärbten Propellerblatts. In welche Richtung scheint der Propeller basierend auf diesen Einzelbildern zu rotieren?
4. Was wäre die maximale Aufnahmerate der Kamera (vollständige Bilder pro Sekunde), die das Abtasttheorem gegeben die aktuelle Rotation des Propeller **nicht** erfüllt?

Aufgabe 3 — Salt-and-pepper Noise - 25 Punkte - Theorieaufgabe

Nehmt an ihr habt drei Varianten eines Bildes bei dem ursprünglich jedes Pixel den Wert 36 hatte. Die drei Varianten sind jeweils mit einem individuellen salt-and-pepper noise (Wahrscheinlichkeit 2%) verrauscht. D.h. für jedes Pixel in jeder der Varianten gilt unabhängig voneinander, dass es mit einer Wahrscheinlichkeit von 2% entweder schwarz (0) oder weiß (255) verrauscht ist. Nun vergleicht ihr die Pixel der drei Varianten an der jeweils selben Position (bspw. oben links) und bildet den Median dieser drei Varianten. Wie groß ist nun die Wahrscheinlichkeit, dass dieser Median dem eigentlichen, unver-

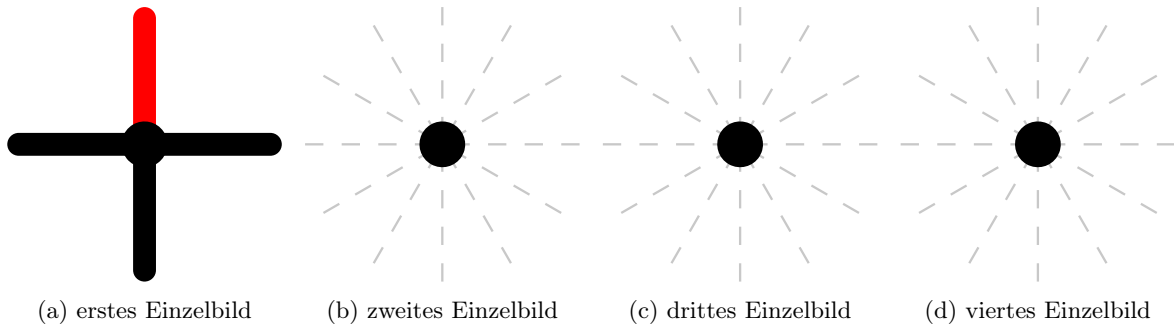


Abbildung 2: Ausgangsposition des Propellers in den ersten vier Einzelbildern des Videos.

rauschten Wert (36) entspricht? Nehmt dazu an, dass, sofern eine Verrauschung eines Pixels vorliegt, die Wahrscheinlichkeiten für schwarz und weiß identisch sind.

Zusatzaufgabe 4 — Rauschen - 30 Punkte - Programmieraufgabe

Erlaubte (Sub-)Pakete: `numpy`, `skimage.io`, `matplotlib`

Erstellt ein Python-Skript mit zwei Funktionen, die jeweils das Gaußsche Rauschen bzw. das salt-and-pepper noise auf ein Bild applizieren. Bei der Funktion zum Gaußschen Rauschen soll die Standardabweichung übergeben werden können und bei der Funktion zum salt-and-pepper noise die Wahrscheinlichkeit für eine Veränderung (salt und pepper sind dann gleich wahrscheinlich). Testet die beiden Funktionen am Bild `astronaut.png` und betrachtet die Ergebnisse. Achtet darauf, dass das Ergebnis wieder im Wertebereich $\{0, 1, 2, \dots, 255\}$ sein muss. Beschreibt kurz, welche Auswirkungen die Veränderung der jeweiligen Parameter der beiden Funktionen (Standardabweichung bzw. Wahrscheinlichkeit für eine Veränderung) auf das Ergebnis hat.

Hinweis 1: Nutzt das Paket `numpy.random`, um zufällige Werte zu ziehen.

Hinweis 2: Mit der Funktion `numpy.clip` lässt sich ein bestimmter Wertebereich für ein Array erzwingen.