Technische Universität Hamburg
Mathematical Image Processing (Winter Term 2022/2023)
Prof. Dr. Marko Lindner, Kristof Albrecht

Exercise Sheet 3

Deadline: 07.11.22, 12:00pm

Exercise 1 (Contrast Enhancement)**.**
In the lecture, we learned that the histogram of an image can be used to expand the color range of the image in order to enhance the constrast.

a) Write a function

```
[H] = calcHistogram(A)
```

that calculates the histogram H of an 8-bit grayscale image A. The function should return the histogram as a vector of size 256×1.

b) Use the function from part a) or the built-in histogram function *imhist* from Matlab / Octave to write a function

```
[B] = enhContrast(A, method)
```

that enhances the contrast of an 8-bit grayscale image A by a given *method* ('stretching' or 'equalization'). Here, the methods 'stretching' and 'equalization' stand for *contrast stretching* and *histogram equalization* known from the lecture.

Warning: Be careful when calculating with variables of type 'uint8'. You might want to convert A to type 'double' before applying the formulas. At the end, make sure that you convert back to 'uint8'.

Exercise 2 (Otsu's method)**.**
In this exercise, we want to discuss *Otsu' method* for B/W-conversion more precisely. Let

$$u : \{1, ..., M\} \times \{1, ..., N\} \to \{0, ..., 255\}$$

be a grayscale image and $H_u : \{0, ..., 255\} \to \mathbb{N}_0$ denote its histogram. For a given threshold $s \in [0, 255]$, we divide $\Omega = \{1, ..., M\} \times \{1, ..., N\}$ into the two classes

$$C_1(s) = \{(i,j) \in \Omega \mid u(i,j) \in [0,s]\} \quad \text{and} \quad C_2(s) = \{(i,j) \in \Omega \mid u(i,j) \in (s,255]\}.$$

Note that $C_1(s)$ contains all pixels that will be colored black if we perform a B/W-conversion with the threshold s, whereas C_2 contains all white pixels.

a) Let $P_i(s)$ denote the probability that a randomly chosen pixel (uniform distribution) is in $C_i(s)$ for $i \in \{1, 2\}$. Show that these probabilites can be written as

$$P_1(s) = \frac{1}{|\Omega|} \sum_{i=0}^{\lfloor s \rfloor} H_u(i) \quad \text{and} \quad P_2(s) = \frac{1}{|\Omega|} \sum_{i=\lfloor s \rfloor + 1}^{255} H_u(i),$$

where $\lfloor \cdot \rfloor : \mathbb{R} \to \mathbb{Z}$ denotes the floor function.

b) The mean color values on the subsets $C_1(s)$ and $C_2(s)$ are then given via

$$\mu_1(s) = \begin{cases} \frac{1}{|C_1(s)|} \sum\limits_{i=0}^{\lfloor s \rfloor} i \cdot H_u(i), & \text{if } C_1(s) \neq \emptyset \\ 0, & \text{if } C_1(s) = \emptyset \end{cases} \quad \text{and} \quad \mu_2(s) = \begin{cases} \frac{1}{|C_2(s)|} \sum\limits_{\lfloor s \rfloor + 1}^{255} i \cdot H_u(i), & \text{if } C_2(s) \neq \emptyset \\ 255, & \text{if } C_2(s) = \emptyset. \end{cases}$$

Prove the following statement: For any given threshold $s \in [0, 255]$, we have

$$P_1(s) \cdot \mu_1(s) + P_2(s) \cdot \mu_2(s) = \mu,$$

where μ is the overall mean value

$$\mu = \frac{1}{|\Omega|} \sum_{i=0}^{255} i \cdot H_u(i).$$

c) Let $\tilde{H}_u : \{0, ..., 255\} \to [0, 1]$ be the normalized histogram of u, i.e.

$$\tilde{H}_u(i) := \frac{1}{|\Omega|} H_u(i) \qquad \text{for all } i \in \{0, ..., 255\}.$$

Otsu's method then aims to minimize the variance

$$\sigma^2(s) = \sum_{i=0}^{\lfloor s \rfloor} (i - \mu_1(s))^2 \cdot \tilde{H}_u(i) + \sum_{i=\lfloor s \rfloor + 1}^{255} (i - \mu_2(s))^2 \cdot \tilde{H}_u(i).$$

Notice that this is the sum of variances on $C_1(s)$ and $C_2(s)$. Show that the equation

$$\sigma^2(s) = \sum_{i=0}^{255} i^2 \cdot \tilde{H}_u(i) - \left(P_1(s) \cdot \mu_1(s)^2 + P_2(s) \cdot \mu_2(s)^2\right)$$

holds for any $s \in [0, 255]$. Conclude that minimizing σ^2 is equivalent to maximizing

$$G(s) := P_1(s) \cdot \mu_1(s)^2 + P_2(s) \cdot \mu_2(s)^2 \qquad \text{for all } i \in \{0, ..., 255\}.$$

Remark: In practical cases, we will only consider thresholds $s \in \{0, ..., 255\}$. This means that we can simply evaluate $(G(s))_{0 \leq s \leq 255}$ and determine the maximum of this array.

Exercise 3 (B/W Conversion)**.**
Write a function

```
[B, t] = bwConversion(A, method)
```

that calculates a threshold t for the B/W-conversion of an 8-bit grayscale image A by a given *method* ('median' or 'otsu') and additionally returns the B/W-converted image B.

If *method* is 'median', t should be chosen as the median of all color values. You can use the built-in function *median* from Octave / Matlab, but make sure that you convert to 'double' before applying the function, as it contains some bugs when dealing with type 'uint8'.

For the implementation of *Otsu's method*, see Exercise 2. Use the function from exercise 1a) or the built-in function *imhist* from Matlab / Octave to calculate the histogram of A.

($\star$) **Exercise 4** (Isodata method)**.**
In this exercise, we discuss the well-posedness of the *Isodata method*. Therefore, we consider the maps μ_1 and μ_2 from Exercise 2b). Recall that the *Isodata method* chooses the treshold $s^* \in [0, 255]$ as a fixed point of the function

$$f(s) := \frac{1}{2}(\mu_1(s) + \mu_2(s)) \qquad \text{for all } i \in \{0, ..., 255\}. \tag{1}$$

We will show that at least one fixed point of f exists and how to calculate a fixed point with a standard fix point iteration.

a) Show that μ_1, μ_2 are non-decreasing functions, such that f is also non-decreasing.

b) Prove the following theorem:

Theorem. *Let $g : [a, b] \to [a, b]$ be a non-decreasing function. Then there exists $s^* \in [a, b]$ with $s^* = g(s^*)$, i.e. g has at least one fixed point in $[a, b]$.*

c) Let $g : [a, b] \to [a, b]$ be a non-decreasing function and consider the fixed point iteration

$$s_{n+1} = g(s_n) \qquad \text{for all } n \in \mathbb{N}_0$$

with starting point $s_0 \in [a, b]$. Show that the sequence $(s_n)_{n \in \mathbb{N}_0}$ converges. Give an example of a non-decreasing function g and a starting point s_0 such that the limit

$$\hat{s} = \lim_{n \to \infty} s_n$$

is <u>not</u> a fixed point of g. This means that this fixed point iteration does not converge in general.

d) Given an arbitrary starting point $s_0 \in [0, 255]$, show that the fixed point iteration from part c) with respect to the function f from (1) converges to a fixed point of f after a finite number of steps.

Hint: The function f is piecewise constant.