Assignment : Histograms and filtering
Signal and Image Processing

February 13, 2023

You can work on this assignment and submit your solution (report and code) as a GROUP. This assignment counts towards your grade and have to be submitted in order to pass the course. You must follow the report guidelines found in `guidelines.pdf`. The page limit for this assignment is **10 pages** including everything, i.e. illustrations and code snippets.

1. Pixel-wise contrast enhancement

1.1. (1 point) Write your own Python function that implements the gamma transform of a gray scale image. Illustrate the function by applying it to an image of your choice. Try different values of gamma and notice how the image details become more or less visible.
Deliverables: Include your function as a code snippet in the report, a figure illustrating your results, and a comment about the effect of changing gamma.

1.2. (1 point) Apply the grayscale gamma-correction function from the previous question to the color image 'autumn.tif' by doing so on each of the RGB component separately.
Deliverables: Include a code snippet in the report showing your solution and plot the original image and the result of this procedure side-by-side.

1.3. (1 point) Apply the grayscale gamma-correction function from the previous question to the color image 'autumn.tif' by first converting to HSV color representation (see `skimage.color.rgb2hsv`) and apply the gamma correction to the v-channel, and finally convert back to RGB representation (see `skimage.color.hsv2rgb`).
Deliverables: Plot the original image and the result of this procedure side-by-side. Compare the result with your solution to Question 1.2, which of the two approaches provide the best result?

2. Image filtering and enhancement

2.1. (1 point) Compare the effect of mean and median filtering on a noisy version of the image *eight.tif* for salt and pepper as well as gaussian noise. You can use the function `skimage.util.random_noise` for doing this.
Deliverables: Include images as illustration. What is the visual effect of increasing the kernel size N ? Store and plot the different computational times obtained for N=1 to N=25 and each time for 100 executions (use Pythons `timeit` package). Comment on your results.

2.2. (1 point) Consider a Gaussian filter of fixed standard deviation $\sigma = 5$ and filter the image with increasing value of kernel size N.
Deliverables: Include images as illustration. What do you observe when N is large enough and how can it be explained?

2.3. (1 point) Experiment with filters of increasing σ's, choosing at each iteration $N = 3\sigma$ for the filter kernel size $N \times N$.
Deliverables: Include images as illustration and comment on the effect with respect to denoising versus sharpness of the image.

3. Complex numbers:

The following questions are intended as a repetition and training of basic algebra of complex numbers.

In the questions below, $\cdot$ denotes multiplication of complex numbers.

3.1. (1 point) Using $i = \sqrt{-1}$, $a_1, a_2, b_1, b_2, d_1, d_2 \in \mathbb{R}$, and $a, b \in \mathbb{C}, a = (a_1 + ia_2), b = (b_1 + ib_2)$, reduce each of the following to the form $d = (d_1 + id_2)$:

i. $d = a + b$
ii. $d = a - b$
iii. $d = a \cdot b$
iv. $d = \frac{a}{b}$

3.2. (1 point) Rewrite $d = \sqrt{-3}$ to the form $d = (d_1 + id_2)$.

3.3. (1 point) Using $i = \sqrt{-1}$, $a_r, a_\theta, b_r, b_\theta, d_r, d_\theta \in \mathbb{R}$, and $a, b \in \mathbb{C}, a = a_r e^{ia_\theta}, b = b_r e^{ib_\theta}$ (polar form), reduce each of the following to the form $d = d_r e^{id_\theta}$:

i. $d = a \cdot b$
ii. $d = \frac{a}{b}$

3.4. (1 point) Write the complex conjugate of $a_r e^{ia_\theta}$ on polar form.

3.5. (1 point) Given a and b as complex numbers on polar form, use Euler's formula $e^{ix} = \cos(x) + i\sin(x)$, simplify the following to the form $d = (d_1 + id_2)$

i. $d = a + b$
ii. $d = a - b$

3.6. (1 point) Given $a = (a_1 + ia_2)$, rewrite it to polar form $d = d_r e^{id_\theta}$.

Deliverables: For each question, please write the essential steps used in order to reach the result.

4. Fourier Transform – Practice:

4.1. (1 point) Use Python to calculate the power spectrum of `trui.png` (we suggest you use `scipy.fft.fft2` for computing the discrete Fourier transform). Apply the function `scipy.fft.fftshift` and interpret the resulting representation of the image.

Deliverables: The answer should include examples of the input and output, a crucial Python code snippet, and a description of you interpretation of the result.