Exercise 1 for MA-INF 2201 Computer Vision WS22/23
Submission on 20.10.2022

October 14, 2022

1. **Rectangles and Integral Images**
   Read the image *bonn.png* and convert it to a grey image.
   (a) Compute and display the integral image without using the function *integral*.
   (b) Compute the mean grey value of the image by:
      i. summing up each pixel value in the image, i.e., $\frac{1}{R}\sum_{p \in R} I(p)$,
      ii. computing an integral image using the function *integral*,
      iii. computing an integral image with your implementation.
   (c) Select 10 random squares of size 100x100 within the image and compute the mean grey value using the three versions. Output the runtime of this task for the three versions in seconds using *time*.

   (3 Points)

2. **Histogram Equalization**
   Read the image *bonn.png* and convert it to a grey image and perform histogram equalization:
   (a) using *equalizeHist*
   (b) using your implementation of the function *equalizeHist*

   Display both results. Compute the absolute pixelwise difference between the results and print the maximum pixel error. ***(2 Points)***

3. **Convolution Theorem**
   Prove that convolutions are associative in the continuous domain. ***(2 Points)***

4. **2D Filtering**
   Read the image *bonn.png* and convert it to a grey image, and display it. Filter the image with a Gaussian kernel with $\sigma = 2\sqrt{2}$

(a) using the function *GausianBlur*

(b) using the function *filter2D* without using the function *getGaussianKernel*

(c) using the function *sepFilter2D* without using the function *getGaussianKernel*

and display the three results. Compute the absolute pixel-wise difference between all pairs (there are three pairs) and print the maximum pixel error for each pair. ***(2 Points)***

5. **Multiple Gaussian Filters**
Read the image bonn.png, convert it into a grey image, and display it. Filter the image

   (a) twice with a Gaussian kernel with $\sigma = 2$

   (b) once with a Gaussian kernel with $\sigma = 2\sqrt{2}$

   and display both results, compute the absolute pixel-wise difference between the results, and print the maximum pixel error. ***(1 Point)***

6. **More on Convolution**
Prove that convolution two times with a Gaussian kernel with the standard deviation σ is the same as convolution once with a Gaussian kernel with the standard deviation $\sigma\sqrt{2}$. ***(2 Points)***

7. **Denoising**
Read the image *bonn.png*, convert it into a grey image, add 30% salt and pepper noise (the chance that a pixel is converted to a black or white pixel is 30%), and display it. Filter the image by

   (a) a Gaussian kernel

   (b) a Median filter using the function *medianBlur*

   (c) a Bilateral filter using the function *bilateralFilter*

   and display the three results. Select the filter size from the range [1, 3, 5, 7, 9] that minimizes the mean distance between the filtered image and the original grey image. ***(3 Points)***

8. **Separability of Filters**
Read the image *bonn.png* and convert it into a grey image.

   (a) Filter the image using the two 2D kernels given below.

   (b) Use the class SVD of OpenCV to separate each kernel. If a kernel is not separable, use an approximation by taking only the highest singular value. Filter the images with the obtained 1D kernels and display the results.

(c) Compute the absolute pixel-wise difference between the results of (a) and (b), and print the maximum pixel error.

$$K_1 = \begin{bmatrix} 0.0113 & 0.0838 & 0.0113 \\ 0.0838 & 0.6193 & 0.0838 \\ 0.0113 & 0.0838 & 0.0113 \end{bmatrix}, K_2 = \begin{bmatrix} -0.8984 & 0.1472 & 1.1410 \\ -1.9075 & 0.1566 & 2.1359 \\ -0.8659 & 0.0573 & 1.0337 \end{bmatrix}$$

(5 Points)

You are only permitted to use the libraries imported in the template. Please write the names of your group members in the README.