Exercise 1 for MA-INF 2201 Computer Vision WS15/16
26.10.2015
Submission on 01.11.2015

1. **Rectangles and Integral Images**
   Read the image `bonn.png` and convert it into a gray image.

   (a) Generate 100 random rectangles within the image and display the image with the rectangles.

   For each rectangle, compute the average intensity by

   (b) summing up each pixel value inside of the rectangle, i.e., $\frac{1}{|R|}\sum_{p\in R} I(p)$,

   (c) computing an integral image using the function `integral`,

   (d) computing an integral image without using the function `integral`

   and compare the runtime for the three versions in seconds using `getTickCount` and repeating the computation of the sum for 10 times. Use the command line (`std::cout`) to print the results.
   (2 Points)

2. **Histogram Equalization**
   Read the image `bonn.png`, convert it into a gray image, and display it.

   (a) Perform histogram equalization using `equalizeHist`.

   (b) Perform histogram equalization without using `equalizeHist`.

   (c) Compute the absolute pixelwise difference between the results using `absdiff`. Print the maximum pixel error using `minMaxLoc`.

   Display the result for (a) and (b).
   (2 Points)

3. **Convolution Theorem** (theoretical)
   Proof that convolutions are in the continuous case associative.
   (2 Points)

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

   (a) using `GaussianBlur`

   (b) using `filter2D` without using `getGaussianKernel`

   (c) using `sepFilter2D` without using `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. **More on Convolution** (theoretical)
Proof that convolution two times with a Gaussian kernel with standard deviation σ is the same as convolution once with a Gaussian kernel with standard deviation $\sqrt{2}\sigma$.
(2 Points)

6. **Multiple Gaussian Filters**
Read the image `bonn.png`, convert it into a gray 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.
(2 Points)

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

   (a) a Gaussian kernel

   (b) Median filter `medianBlur`

   (c) Bilateral filter `bilateralFilter`

   and display the three results. Select the filter parameters that give the best visual results.
(2 Points)

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

   (a) Filter the images using `filter2D` and the two 2D-filter kernels given below and display the result.

   (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 using `sepFilter2D` and display the results.

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

   (6 Points)

0.0113	0.0838	0.0113
0.0838	0.6193	0.0838
0.0113	0.0838	0.0113

kernel 1

-0.8984	0.1472	1.1410
-1.9075	0.1566	2.1359
-0.8659	0.0573	1.0337

kernel 2