Computer vision course

Proff. Stefano Ghidoni | Matteo Terreran

Lab 2 - Filters and histogram equalization

Task 1
Write a program that loads the image provided (Garden) and shows it. It then converts the image to grayscale (cv::cvtColor() function) and saves it as Garden_grayscale.

Task 2
Expand Task 1 **implementing** a max filter and a min filter in two dedicated functions (implemented by you in separate header/source files named ImageFilters.h and ImageFilters.cpp) that manipulate the pixels directly. Such functions consider square kernels whose size is provided as an argument. The size should be an odd number—if it is even, the function does not process the image and prints an error message.
Apply the filters using several kernel sizes and find the option that:

- best removes the noise from the Lena_corrupted image (compare with the original version);
- best removes the noise from the Astronaut_corrupted image (compare with the original version);
- best removes the electric cables in the background from the Garden_grayscale without corrupting the image too much.

Task 3
Expand Task 2 implementing the median filter in a dedicated function in the same header/source files used for task 2. Test the filter on the same images.

Task 4
Expand Task 3 using the gaussian smoothing filter provided by OpenCV (you should find the right function in the documentation). Show the results on the same images.

Task 5
Expand Task 1 plotting the histogram of Garden_grayscale using 256 bins and range [0, 255]. Try also different number of bins. You are free to evaluate the histogram using your code or the functions provided by OpenCV.

Task 6
Expand Task 5 including histogram equalization using the dedicated OpenCV function and visualize the result.

Image sources
https://vincmazet.github.io/bip/restoration/denoising.html
https://what-when-how.com/embedded-image-processing-on-the-tms320c6000-dsp/non-linear-filtering-of-images-image-processing-part-1/