Image Processing – HW3 (05/31/2021)

Instructions – Follow these carefully:

1. Please upload your work as a zip file attachment to Moodle. In the zip file, it must have the source code and a PDF report where you explain and display the outputs for each problem.
2. You can use either Python or Matlab to do the work.
3. Please feel free to read related materials available in the official Matlab/Python documentation.
4. The due date is 6/14 before 11:55pm. No late submission is allowed.

Assignment:

1. Remove salt-and-pepper noise
   1. (20%) Write a routine that performs two-dimensional 5x5 median filtering to try to clean up the noise of ‘bab\_noise.bmp’ and ‘peppers\_noise.’ You need to exclude the noise pixels before applying median filtering and report PSNR before and after denoising as the table below shows.

|  |  |  |
| --- | --- | --- |
| PSNR | Before denoising | After denoising |
| Baboon |  |  |
| Peppers |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.0232 | 0.0338 | 0.0383 | 0.0338 | 0.0232 |
| 0.0338 | 0.0492 | 0.0558 | 0.0492 | 0.0338 |
| 0.0383 | 0.0558 | 0.0632 | 0.0558 | 0.0383 |
| 0.0338 | 0.0492 | 0.0558 | 0.0492 | 0.0338 |
| 0.0232 | 0.0338 | 0.0383 | 0.0338 | 0.0232 |

* 1. (20%) Following the previous question, use two-dimensional 5x5 Gaussian filtering with its kernel as below and report the PSNR results:

* 1. (10%) Write a rotine that performs two-dimensional median filtering with an adaptive kernel size on bab\_noise\_90.bmp. For each noise pixel “p”, the adaptive kernel size needs to be the same as the smallest size of the sliding window centered at “p” with at least one non-noise pixel. Please also report the PSNR results in the table format like Q1.a.

1. Edge Detection
   1. (25%) Please implement Sobel filtering to find the edge map for ‘pepper.bmp’, whose result should look like <https://www.mathworks.com/discovery/edge-detection.html> (please implement the Sobel filter by yourself)
   2. (25%) Following the previous question, please use 5x5 Marr-Hildreth operator (shown below) to find the edge map.

