BİL 467/561 – Image Processing

Exercise #4

1. Implement the following adaptive mean filter:

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_{S_{xy}}^2}\left[g(x,y) - \hat{z}_{S_{xy}}\right]$$

where:

- S_{xy} is the filter kernel support (5x5).
- $g(x,y)$ is the noisy input image.
- σ_η^2 is the variance of the noise (Assume $\sigma_\eta^2 = 0.001225$).
- $\hat{z}_{S_{xy}}$ is the local average intensity of the pixels in S_{xy}.
- $\sigma_{S_{xy}}^2$ is the local variance of the intensities of pixels in S_{xy}.

After loading your image as 8-bit per pixel, normalize the intensity values to the [0 – 1] range as floats (numpy.single) and work on this normalized image. After the filtering operations are done, do not forget to convert your pixel values back to 0-255 (8-bit) range and cast into unsigned char. Once your implementation is ready, apply it to the provided noisy test image (GaussianNoise.jpg). Call your output image output_1_1.

Then apply the following denoising filters:

1. OpenCV's 5x5 box filter. Call this filter's output output_1_2.
2. OpenCV's 5x5 Gaussian filter (with auto var ➔ σ = 0). Call this filter's output output_1_3.

Using OpenCV's PSNR calculation function and the original clean image (cleanImage.jpg), compute and compare the PSNR values for output_1_1, output_1_2 and output_1_3.

[Observation: Change the value of σ_η^2, for example try 0.009, what's the result?]

2. Implement the following adaptive median filter:

Level A	If $z_{min} < z_{med} < z_{max}$, go to Level B
	Else increase the size of S_{xy}
	If $S_{xy} < S_{max}$, repeat Level A
	Else, output z_{med}
Level B	If $z_{min} < z_{xy} < z_{max}$, output z_{xy}
	Else output z_{med}

where:

- S_{xy} is the filter kernel support (3x3, 5x5, 7x7 from minimum to maximum).
- z_{min} is the minimum intensity value in S_{xy}.
- z_{max} is the maximum intensity value in S_{xy}.
- z_{med} is median of intensity values in S_{xy}.
- z_{xy} is the intensity at coordinates (x,y).
- S_{max} (7x7) is the maximum allowed size of S_{xy}.

Once your implementation is ready, apply it to the provided noisy test image (SaltPepperNoise_1.jpg). Call your output image output_2_1.

Then apply the following denoising filters:

1. OpenCV's 3x3 median filter. Call this filter's output output_2_2.
2. OpenCV's 5x5 median filter. Call this filter's output output_2_3.
3. OpenCV's 7x7 median filter. Call this filter's output output_2_4.
4. Your 3x3 center weighted (center weight 3) median filter. Call this filter's output output_2_5.
5. Your 5x5 center weighted (center weight 5) median filter. Call this filter's output output_2_6.
6. Your 7x7 center weighted (center weight 7) median filter. Call this filter's output output_2_7.

Using OpenCV's PSNR calculation function and the original clean image (cleanImage.jpg), compute and compare the PSNR values for output_2_1, output_2_2, output_2_3, output_2_4, output_2_5, output_2_6 and output_2_7.

[Observation: Change the noisy input image to SaltPepperNoise_2.jpg. What's the result?]