Assignment 2

Dana Kianfar - Jose Gallego Posada

11391014 - 11390689

{dana.kianfar, jose.gallegoposada}@student.uva.nl

1 Gaussian versus Box

In Figure 1, we display our box-filtered images at varying filter sizes along with the original picture. In a similar manner, we display our results using a median filter in Figure 2. We find that the box filter achieves better results. When comparing the filtered image obtained using $K = 3$ for both filters, we observe that the box filter is better at removing the noise.

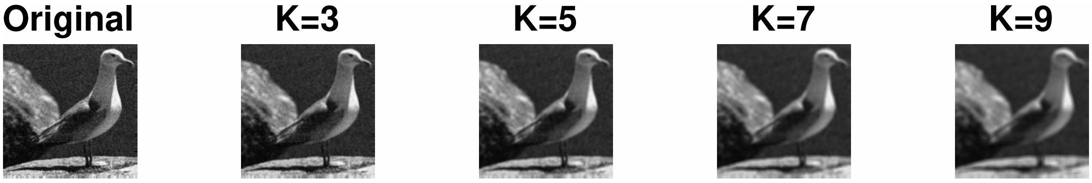

Figure 1: Box filter with different filter size K.

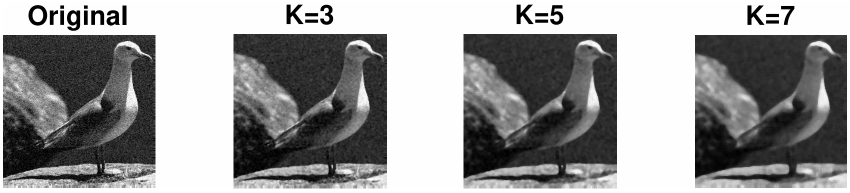

Figure 2: Median filter with different filter size K.

In this case, box performs better due to the properties of the noise present in the image. It is likely that the noise has been by Gaussian process, which agrees with the assumptions for applying a box kernel: pixels in the image should be spatially correlated and the noise should be stationary, so adding the surrounding pixels will have 0 mean. On the other hand, median filtering is more suited for handling noise in the form of outlier values, like *salt and pepper* noise.

2 Histogram Matching

As can be seen in Figure 3, the transformed image is significantly brighter than the input, which is reasonable given the characteristics of the reference image.

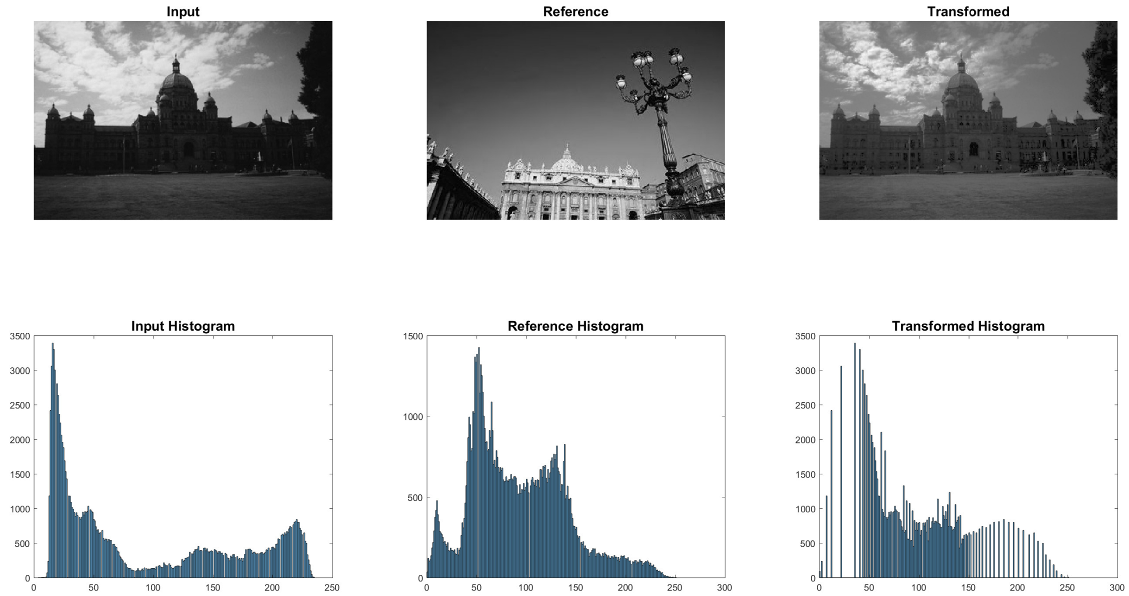

Figure 3: Histogram Matching

3 Gradient Magnitude and Direction

We present our Sobel kernel gradient results in Figure 4. As expected, the gradients in the x and y directions are able to detect vertical and horizontal edges respectively. Note that in our implementation positive gradients in the y direction are pointing downwards.

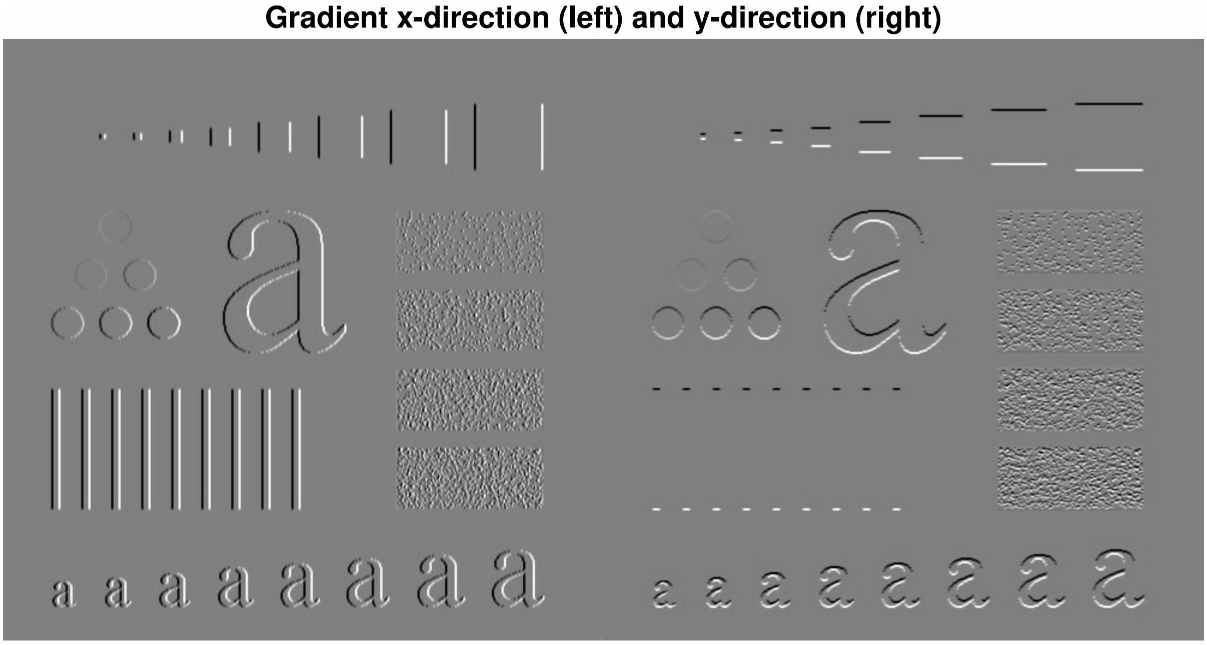

Figure 4: Gradients in the x and y direction of the Sobel operator

We display the magnitude and direction of the gradient at each pixel on Figure 5. The gradient magnitude, shown on the left, clearly allows to visualize all the edges present in the image. Large magnitudes are shown in white, and correspond to edges with a sharp change in pixel intensity. The gradient direction, shown on the right, displays the angle of the gradient in degrees. A gradient with a 180° angle is shown in white, and conversely a gradient with an angle of 0° degrees in shown in black. Gray pixels denote regions with no gradients (magnitude of zero). Noise and compression artifacts in the original image contribute to the flaky, salt and pepper patterns in our gradients.

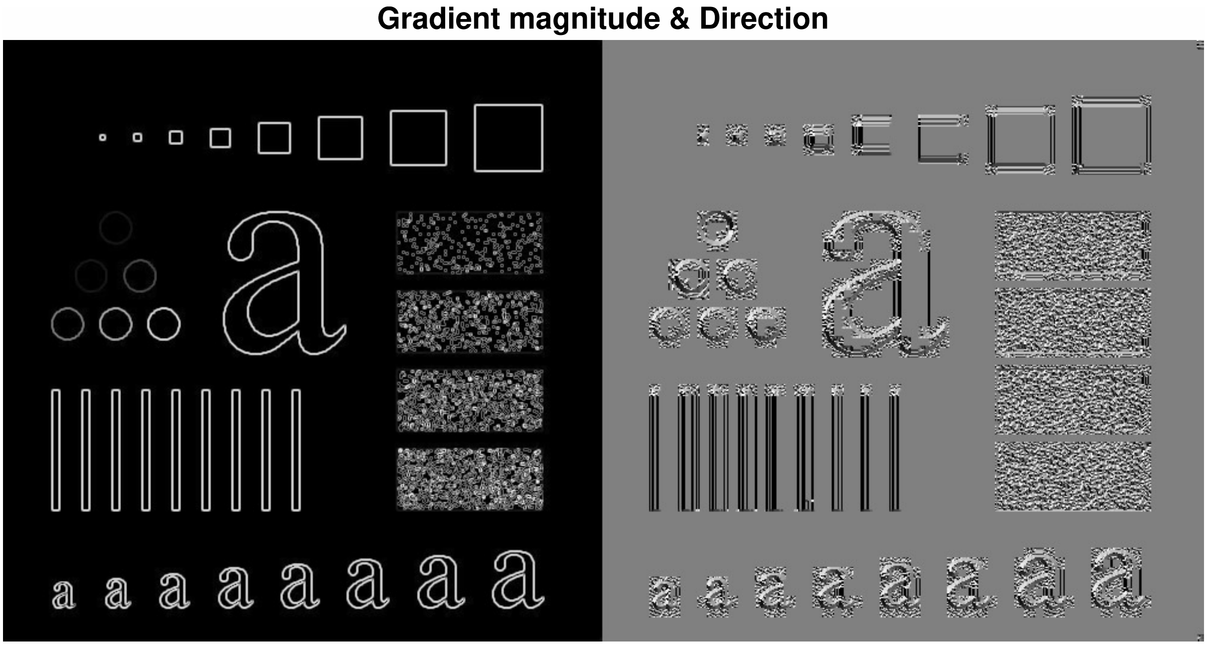

Figure 5: Gradient magnitude (left) and direction (right) at each pixel of the Sobel operator

4 Unsharp Mask

The unsharp mask first performs smoothing using a Gaussian kernel to obtain a low-passed version of the image, and subsequently subtracts it from the original image to get the high-passed version, which can be viewed as the fine details. It then strengthens the low-pass version by a constant scaling co-efficient k and adds it back to the original image. We visualize our results by displaying an image representing each stage of this process.

We experimented with various kernel sizes $n \in \{3, 11, 21, 121\}$, sigma values $\sigma \in \{1, 2, 5, 10, 20\}$ and $k \in \{1, 2, 5, 10\}$. We only present a representative subset of these experiments below, due to size considerations.

We observed that the kernel size n affects the sensitivity of the unsharp mask to sharp edges. Larger kernels smooth a larger selection of pixels and produce a blurred image with the the most common pixel intensities of the original image. Subtracting this from the original image, we obtain a high-pass version with very high sensitivity to changes in intensity. We consider very high values of n to be oversensitive to noise, and the resulting image is over-sharpened. An application of this method with $n = 121, \sigma \in \{2, 20\}$ is shown in Figures 6 and 7.

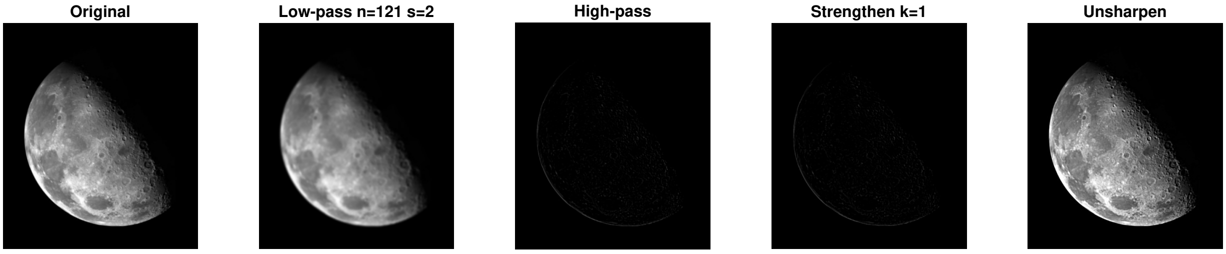

Figure 6: Unsharp mask results with a large kernel

The sigma σ dictates how the strongly the Gaussian kernel smooths the image. When a highly-smoothed version of the image is subtracted from the original image, only the fine details remain. Therefore, higher values of σ also affects the sensitivity of the unsharp mask to small changes in intensity. It should be noted that the magnitude of σ should correspond to the size of the kernel. A very large n with a small σ will not be very different from a smaller kernel with the same σ, as outside of three standard deviations the coefficient of the kernel will be negligible, and no smoothing will occur. Visualizations of two small and large value for sigma $\sigma \in \{2, 20\}$ are respectively shown in Figures 6 and 7. Note that as other parameters are kept constant, a large sigma results in over-sharpening of the image.

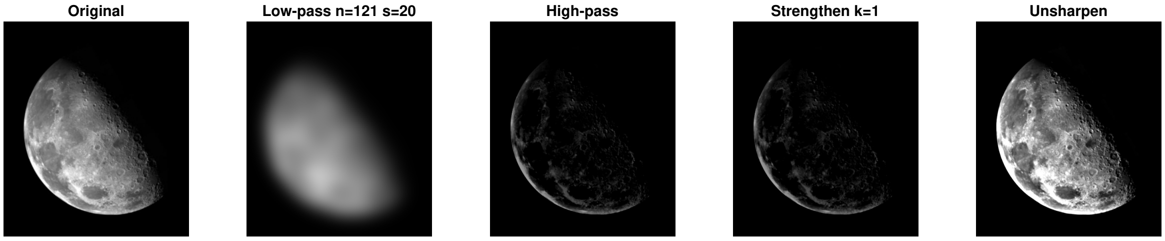

Figure 7: Unsharp mask results with a large sigma

The coefficient k affects how strongly the high-passed version is added to the original image. At $k = 1$, the high-passed version is added to the image without any strengthening, as shown in Figure 8.

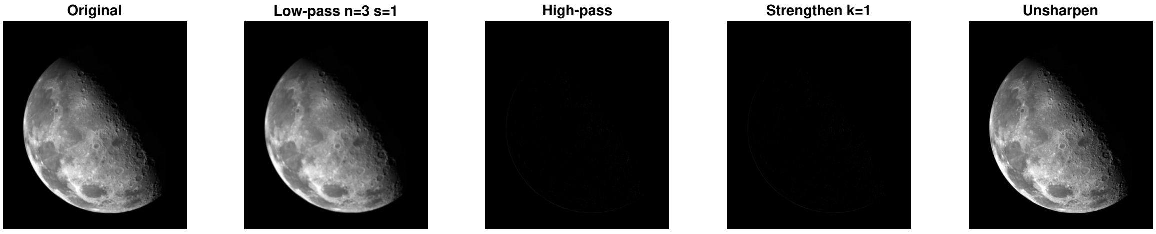

Figure 8: Unsharpening without a strengthening factor

With a larger k, the fine details of the image become more apparent resulting in a crisper image, as shown in 9

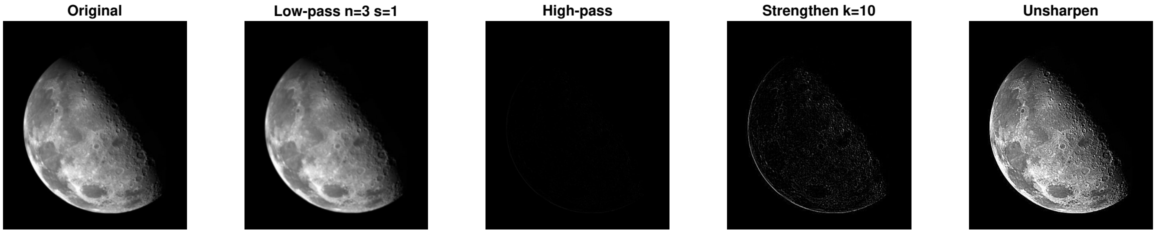

Figure 9: Unsharp mask results with a high strengthening factor

Unsharpening masks are often used in image processing software to sharpen images. Similarly, cetain hardware such as imaging microscopes or scanners often use unsharpening masks to reduce blurring.

5 Laplacian of Gaussian

We present our Laplacian of Gaussian (LoG) results in Figure 10.

All of the methods are able to detect edges in the picture successfully. However, we consider that the results of method 3 are more robust than the previous methods in the sense that the detected edges are more clearly recognizable in this image, as can be seen in the region around the shoulder and the right side of Lena's hat. Besides, there is less noise in the regions in which no edges are present compared to the results of the other methods.

Given that this operator is approximating a second derivative measurement on the image, it is important to apply a noise removal preprocessing (such as Gaussian smoothing) before applying the Laplacian filter, due to its high sensitivity to noise [1].

According to [2], "the optimal choice on engineering grounds for $\frac{\sigma_i}{\sigma_e}$ is about 1.6", where σ_i and σ_e are associated to the inhibitory to excitatory Gaussian distributions, respectively. Note that the values chosen for σ_1 and σ_2 for method 3 in Figure 10 satisfy this ratio.

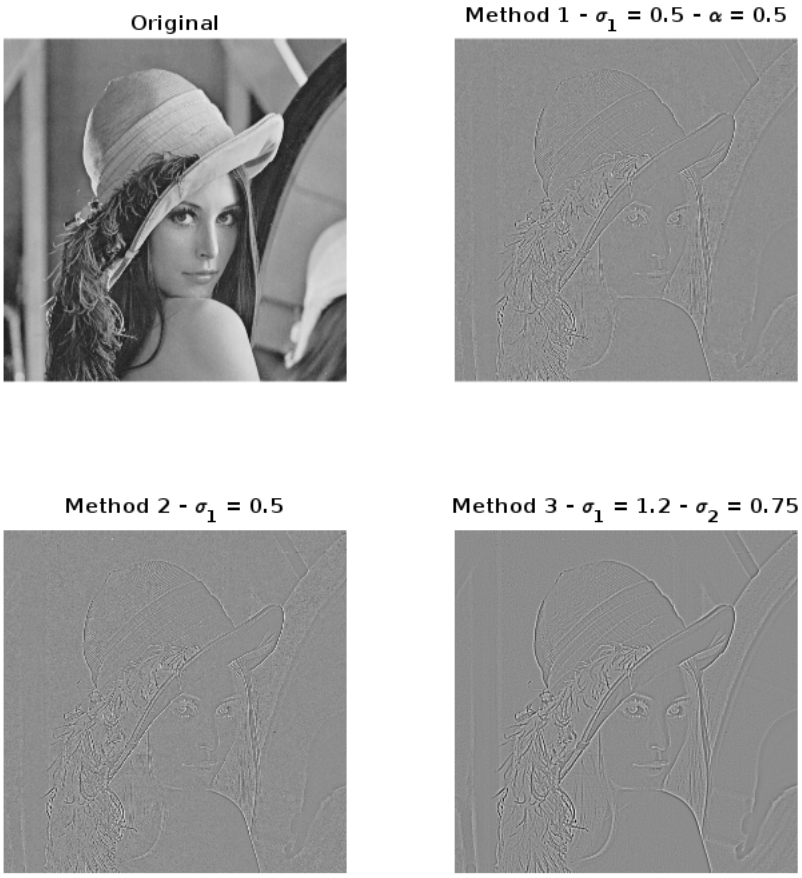
Original
Method 1 - σ_1 = 0.5 - α = 0.5
Method 2 - σ_1 = 0.5
Method 3 - σ_1 = 1.2 - σ_2 = 0.75

Figure 10: Results of LoG operator

The LoG operator calculates the second spatial derivative of an image, which finds regions of rapid intensity change. Thus, it is usually used for edge and corner detection, and as an enhancement technique in remote sensing applications [1].

References

[1] R. Fisher, S. Perkins, A. Walker, and E. Wolfart, "Spatial Filters - Laplacian/Laplacian of Gaussian." http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm, 2003.

[2] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.