**LAPORAN PRAKTIKUM PENGOLAHAN CITRA DIGITAL**

**11. SHARPENING FILTERS IN THE SPATIAL DOMAIN**



**Disusun oleh :**

**Nama : Andhika Rizky Cahya Putra**

**NPM : 2226250071**

**Kelas : IF4A**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER DAN REKAYASA**

**UNIVERSITAS MULTI DATA PALEMBANG**

**2024**

**TUTORIAL : SHARPENING FILTERS IN THE SPATIAL DOMAIN**

**Goal**

The goal of this tutorial is to learn how to implement sharpening filters in the spatial

domain.

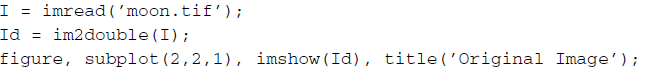
**Objectives**

* Learn how to implement the several variations of the Laplacian mask.
* Explore different implementations of the unsharp masking technique.
* Learn how to apply a high-boost filtering mask.

**Procedure**

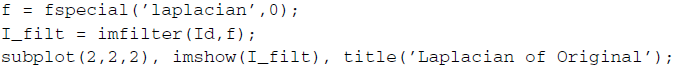
To implement the Laplacian filter, we can either create our own mask or use the fspecial function to generate the mask for us. In the next step, we will use fspecial, but keep in mind that you can just as well create the mask on your own.

1. Load the moon image and prepare a subplot figure.



We are required to convert the image to doubles because a Laplacian filtered image can result in negative values. If we were to keep the image as class uint8, all negative values would be truncated and, therefore, would not accurately reflect the results of having applied a Laplacian mask. By converting the image to doubles, all negative values will remain intact.

1. Create a Laplacian kernel and apply it to the image using the imfilter function.



**Question 1** When specifying the Laplacian filter in the fspecial function, what is the second parameter (in the case above, 0) used for?

|  |
| --- |
|  |

**Question 2** What is the minimum value of the filtered image?

|  |
| --- |
|  |

**Question 3** Verify that a uint8 filtered image would not reflect negative numbers. You can use the image I that was previously loaded.

|  |
| --- |
|  |

You will notice that it is difficult to see details of the Laplacian filtered image. To get a better perspective of the detail the Laplacian mask produced, we can scale the image for display purposes so that its values span the dynamic range of the gray scale.

1. Display a scaled version of the Laplacian image for display purposes.



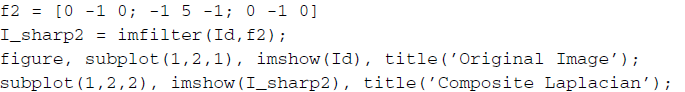
The center coefficient of the Laplacian mask we created is negative. Recall from the chapter that if the mask center is negative, we subtract the filtered image from the original, and if it is positive, we add. In our case, we will subtract them.

1. Subtract the filtered image from the original image to create the sharpened image.



A composite version of the Laplacian mask performs the entire operation all at once. By using this composite mask, we do not need to add or subtract the filtered image—the resulting image is the sharpened image.

1. Use the composite Laplacian mask to perform image sharpening in one step.



**Question 4** You may have noticed that we created the mask without using the fspecial function. Is the fspecial function capable of generating the simplified Laplacian mask?

|  |
| --- |
|  |

**Question 5** Both Laplacian masks used above did not take into account the four corner pixels (their coefficients are 0). Reapply the Laplacian mask, but this time use the version of the mask that accounts for the corner pixels as well. Both the standard and simplified versions of this mask are illustrated in Figure 11.1. How does accounting for corner pixels change the output?

|  |
| --- |
|  |

**Unsharp Masking** Unsharp masking is a simple process of subtracting a blurred image from its original to generate a sharper image. Although the concept is straightforward, there are three ways it can be implemented. Figures 11.2–11.4 illustrate these processes.



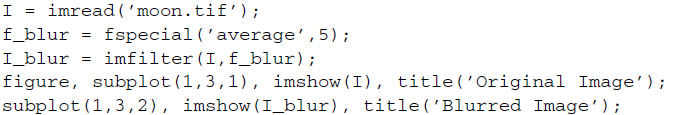
**FIGURE 11.1** Laplacian masks that account for corner pixels (standard and composite).

Let us first implement the process described in Figure 11.2



**FIGURE 11.2** Unsharp masking process including histogram adjustment.

1. Close all open figures and clear all workspace variables.
2. Load the moon image and generate the blurred image.



**Question 6** What does the second parameter of the fspecial function call mean?

|  |
| --- |
|  |

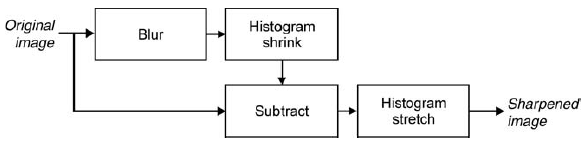
We must now shrink the histogram of the blurred image. The amount by which we shrink the histogram will ultimately determine the level of enhancement in the final result. In our case, we will scale the histogram to range between 0.0 and 0.4, where the full dynamic grayscale range is [0.0 1.0].

1. Shrink the histogram of the blurred image.

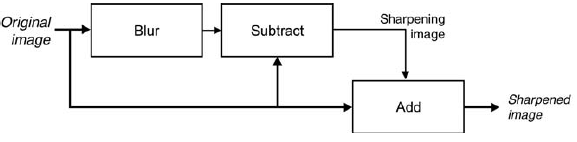


1. Now subtract the blurred image from the original image.





**FIGURE 11.3** Unsharp masking process including histogram adjustment.



**FIGURE 11.4** Unsharp masking process with sharpening image.

We must now perform a histogram stretch on the new image in order to account for previously shrinking the blurred image.

1. Stretch the sharpened image histogram to the full dynamic grayscale range and display the final result.



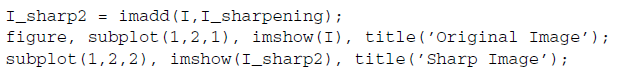
**Question 7** We learned that by shrinking the blurred image’s histogram, we can control the amount of sharpening in the final image by specifying the maximum range value. What other factor can alter the amount of sharpening?

|  |
| --- |
|  |

1. Subtract the blurred image from the original image to generate a sharpening image.



1. Add sharpening image to original image to produce the final result.

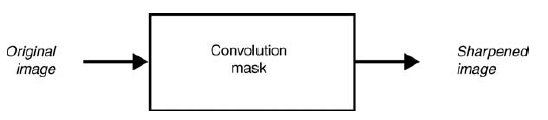


1. Generate unsharp masking kernel using the fspecial function.

**Question 8** How can we adjust the amount of sharpening when using this implementation?

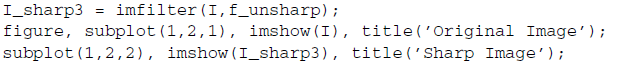
|  |
| --- |
|  |

The third implementation uses a convolution mask, which can be generated using the fspecial function. This implementation is illustrated in Figure 11.5.



**FIGURE 11.5** Unsharp masking process using convolution mask.

1. Apply the mask to the original image to create a sharper image.

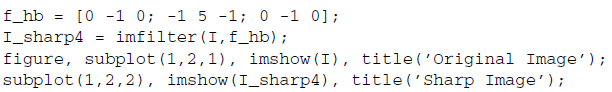


**Question 9** How do we control the level of sharpening with this implementation?

|  |
| --- |
|  |

**High-Boost Filtering** High-boost filtering is a sharpening technique that involves creating a sharpening image and adding it to the original image. The mask used to create the sharpening image is illustrated in Figure 11.6. Note that there are two versions of the mask: one that does not include the corner pixels and another that does.

1. Close any open figures.
2. Create a high-boost mask (where A = 1) and apply it to the moon image.



**Question 10** What happens to the output image when A is less than 1? What about when A is greater than 1?

|  |
| --- |
|  |

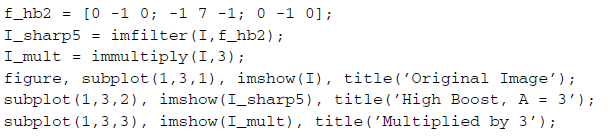
You may have noticed that when A = 1, the high-boost filter generalizes to the

composite Laplacian mask discussed in step 5. As the value of A increases, the output image starts to resemble an image multiplied by a constant.



**FIGURE 11.6** High-boost masks with and without regard to corner pixels.

1. Show that a high-boost mask when A = 3 looks similar to the image simply multiplied by 3.



**Question 11** At what value of A does this filter stop being effective (resemble the image multiplied by a constant)?

|  |
| --- |
|  |