**LAPORAN PRAKTIKUM PENGOLAHAN CITRA DIGITAL**

**15. NOISE REDUCTION USING SPATIAL-DOMAIN**

**TECHNIQUES**



**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 : HIGH-PASS FILTERS IN THE FREQUENCY**

**DOMAIN**

**Goal**

The goal of this tutorial is howto implement high-pass filters in the frequency domain.

**Objectives**

* Learn how to implement the arithmetic mean filter, as well as some of its variations, such as the contraharmonic mean, the harmonic mean, and the geometric mean filters.
* Learn how to perform order statistic filtering, including median, min, max, midpoint, and alpha-trimmed mean filters.

**What You Will Need**

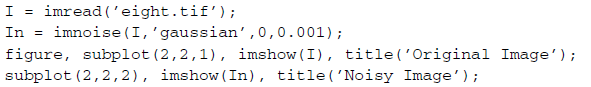
* atmean.m
* geometric.m
* harmonic.m
* c\_harmonic.m

**Procedure**

**Arithmetic Mean Filter**

The arithmetic mean filter, also known as an averaging or low-pass (from its frequency-domain equivalent) filter, is a simple process of replacing each pixel value with the average of an N × N window surrounding the pixel. The averaging filter can be implemented as a convolution mask. As in previous tutorials, we will use the function fspecial to generate the averaging convolution mask.

1. Load the eight image, add (Gaussian) noise to it, and display the image before and after adding noise.



1. Apply an averaging filter to the image using the default kernel size (3 × 3).

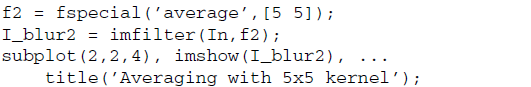




**Question 1** What is the general effect of the arithmetic mean filter?

|  |
| --- |
|  |

1. Implement an averaging kernel with a 5 × 5 mask.



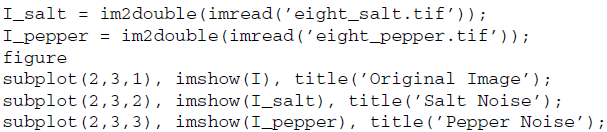
**Question 2** How does the size of the kernel affect the resulting image?

|  |
| --- |
|  |

**Contraharmonic Mean Filter**

The contraharmonic mean filter is used for filtering an image with either salt or pepper noise (but not both). When choosing a value for r, it is important to remember that negative values are used for salt noise and positive values are used for pepper noise. As we will see, using the wrong sign will give undesired results. This filter does not have a convolution mask equivalent, so we must implement it as a sliding neighborhood operation using the nlfilter function. This function allows us to define how we want to operate on the window, which can be specified within a function of our own.

1. Close any open figures.
2. Load two noisy versions of the eight image: one with salt noise and the other with pepper. Also, display the original image along with the two affected images.



The contraharmonic function requires that images be of class double. This is

why we convert the image when loading it.

1. Filter the salt noise affected image using −1 for the value of r.



Our function c\_harmonic takes two parameters: the current window matrix and a value for r. The window matrix, which gets stored into variable x, is passed implicitly by the nlfilter function. Note how we specified the c\_harmonic function as the third parameter of the nlfilter function call. When using the nlfilter function, if you want to pass any additional parameters to your function, you can specify those parameters after the function handle (labeled with a ’@’ in front of it). Notice above how we specified the value of r directly after the function handle.

1. Filter the pepper noise affected image using 1 for the value of r.



As mentioned previously, using the wrong sign for the value of r can lead to

unwanted results.

1. Filter the pepper noise image using the wrong sign for r.



**Question 3** What is the effect of using the wrong sign when filtering with the

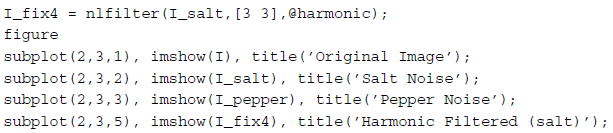
contraharmonic mean filter?

|  |
| --- |
|  |

**Harmonic Mean Filter**

The harmonic mean filter is another variation of the mean filter and is good for salt and Gaussian noise. It fails, however, when used on pepper noise.

1. Close any open figures.
2. Filter the salt noise affected image with the harmonic filter.



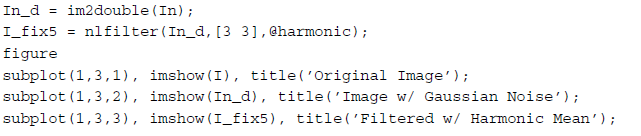
1. Filter the pepper noise image and display the result.



**Question 4** Why does the harmonic mean filter fail for images with pepper noise?

|  |
| --- |
|  |

1. Try to filter the In image (I with additive Gaussian noise) with the harmonic mean filter. The image must be converted to double first.



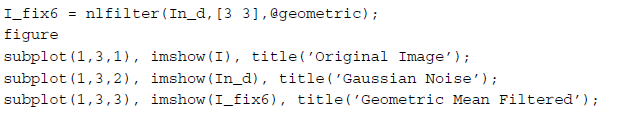
**Question 5** How does the size of the window affect the output image?

|  |
| --- |
|  |

**Geometric Mean Filter**

The last variation of the mean filters we will look at is the geometric mean filter. This filter is known to preserve image detail better than the arithmetic mean filter and works best on Gaussian noise.

1. Close any open figures.
2. Perform a geometric mean filter on the eight image with Gaussian noise (currently loaded in the variable In\_d).



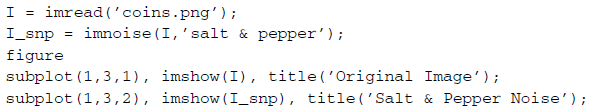
**Question 6** Filter the salt and pepper noise images with the geometric mean filter. How does the filter perform?

|  |
| --- |
|  |

**Order Statistic Filters**

The median filter is the most popular example of an order statistic filter. This filter simply sorts all values within a window, finds the median value, and replaces the original pixel value with the median value. It is commonly used for salt and pepper noise. Because of its popularity, the median filter has its own function (medfilt2) provided by the IPT.

1. Close any open figures and clear all workspace variables.
2. Load the coins image and apply salt and pepper noise.



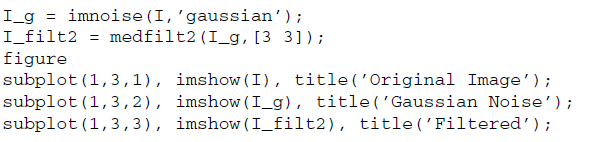
1. Filter the image using the medfilt2 function.



**Question 7** How does the size of the window affect the output image?

|  |
| --- |
|  |

1. Apply the filter to an image with Gaussian noise.

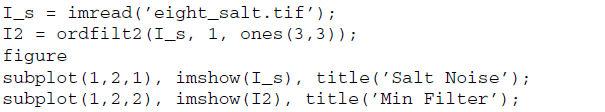


**Question 8** Why do you think the median filter works on salt and pepper noise but not Gaussian noise?

|  |
| --- |
|  |

A quick way to get rid of salt noise in an image is to use the min filter, which simply takes the minimum value of a window when the values are ordered. Recall that we previously used the imfilter function when dealing with convolution masks and nlfilter for sliding neighborhood operations that could not be implemented as a convolution mask. Similarly, the ordfilt2 function is used for order statistic operations.

1. Close any open figures and clear all workspace variables.
2. Use the ordfilt2 function to implement a min filter on an image with salt noise.



**Question 9** Why would this filter not work on pepper noise?

|  |
| --- |
|  |

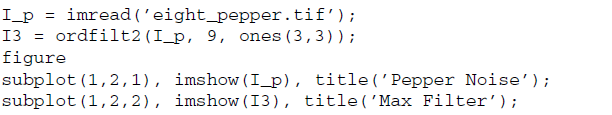
The first parameter specified in the ordfilt2 function is the image we wish to filter. The second parameter specifies the index of the value to be useed after all values in the window have been ordered. Here, we specified this parameter as 1, which means we want the first value after reordering, that is, the minimum value. The last parameter defines the size of the window as well as which values in that window will be used in the ordering. A 3 × 3 matrix of 1’s would indicate a 3 × 3 window and to use all values when ordering. If we instead specified a 3 × 3 matrix where only the first row was 1’s and the last two rows were zeros, then the sliding window would consist of a 3 × 3 matrix, but only the top three values would be considered when ordering. In addition, keep in mind that even though we used a special function to implement the median filter, it is still an order statistic filter, which means we could have implemented it using the ordfilt2 function.

**Question 10** Implement the median filter using the ordfilt2 function.

|  |
| --- |
|  |

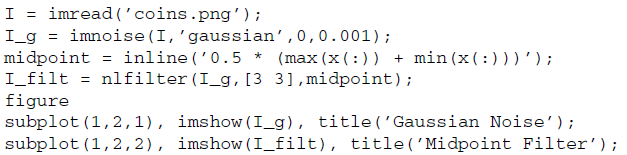
The max filter is used for filtering pepper noise, similar to the technique of the min filter.

1. Filter a pepper noise affected image with the max filter.



Although the midpoint filter is considered an order statistic filter, it cannot be directly implemented using the ordfilt2 function because we are not selecting a particular element from the window, but instead performing a calculation on its values—namely, the minimum and maximum values. Rather, we will implement it using the familiar nlfilter function. This noise removal technique is best used on Gaussian or uniform noise.

1. Filter an image contaminated with Gaussian noise using the midpoint filter.

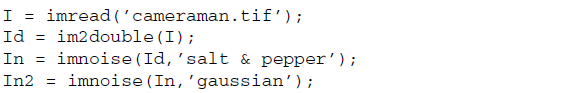


You may have noticed that we have used an inline function instead of creating a separate function, as we did in previous steps. Inline functions are good for quick tests, but—as you may have realized—they are much slower than regular functions.

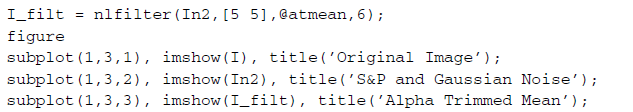
**Alpha-Trimmed Mean Filters**

The alpha-trimmed mean filter is basically an averaging filter whose outlying values are removed before averaging. To do this, we sort the values in the window, discard elements on both ends, and then take the average of the remaining values. This has been defined in the function atmean.

1. Close any open figures and clear all workspace variables.
2. Generate a noisy image with Gaussian noise and salt and pepper noise.



1. Filter the image using the alpha-trimmed mean filter.



**Question 11** When filtering an image with both types of noise, how does the alpha-trimmed mean filter compare to the arithmetic mean filter?

|  |
| --- |
|  |