Low, average and high pass filters/transformation using MATLAB. (Sep 2020)

G10_LT_A1

Arzoo Khan(MIT2020028), Ashutosh Gupta(MIT2020029),Yashwant Kumar Chandra(MIT2020030)
Indian Institute of Information Technology, Allahabad

***Abstract*— This paper presents fundamental introduction to low, average and high pass finite impulse filters and their implementation using MATLAB. Low pass filter is implemented as Image smoothening filter and high pass as Image sharpening filter. We have implemented the algorithm, where the input image is converted into a matrix and then the filter matrix is applied to generate the output matrix. The filtered image is the convolution of original image. In smoothening filter the basic idea is that each pixel is replaced by the average of the pixels in a square window surrounding the pixel. Application of linear transformation in Filter designing is well explained through example. MATLAB Image processing toolbox is used for performing experiments.**

***Keywords*—Low pass filter, High pass filter, Impulse response, Image Sharpening, Image smoothing, Filter coffiecient**

I. Introduction

In signal processing, a filter is a device that is used to remove unwanted components from a signal. Filters are broadly classified into three categories. The first one is a low pass filter that passes all frequencies of a digital signal which are below a certain predefined cutoff frequency. It attenuates the frequencies which are above the cutoff frequency. Low-pass filters are basically used to remove noise from signals. Its cutoff frequency is the frequency at which output is equal to 70.7% of input. The above cutoff frequency output is lower than 70.7% of input.[2]The second category of filters is high-pass filters. They pass all frequencies of a digital signal which are higher than a certain predefined cutoff frequency. It attenuates the frequencies which are below the cutoff frequency. In case of high pass filter, above the cutoff frequency, output is higher than 70.7% of input. The third category is the average filter also known as Finite Impulse Response(FIR).It is a special implementation of low pass filter. They are mainly used in the mixed domain like television signals. Inorder to design a filter, two major steps are followed. In first step, we deal with problems like defining filter type, design method and filter order. The second step consists of defining specifications like frequency unit , sampling frequency, amplitude entry format and many more.[1]

The user can choose either Infinite Impulse Response filter(IIR) or a Finite Impulse Response filter(FIR).FIR uses past and present input samples to calculate output, whereas IIR output is calculated , based on previous output as well as past and present input samples.[1]

II. Background

A. Description

In, this project we designed the low, avg, high pass filers using MATLAB. Filtering is a fundamental signal processing operation, and often a pre-processing operation before further processing.

• Applications:

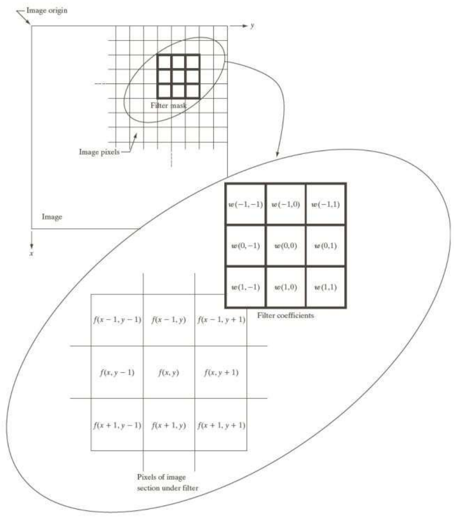
I. Image noise destruction.

II. Image enhancement (i.e., “make the image more vivid”).

III. Edge detection.

B. Basic Idea

When you give a image as a input to our code it convert the whole image in very large matrix (size of matrix depends on the resolution of a input) then we convert that matrix/image in grayscale then we generated the low pass filter matrix and high pass filter matrix using a technique which we will be discussing later in this document. I have attached the image to describe “How the filter matrix is applied to image matrix and a new image matrix is generated” the given image not describes the filtering technique its just an example which represents the filter and imge matrix. In, the image below the filter is taken is of size 3x3 and the filter coefficients are taken randomly , and also it contain the image of size 8x8.



C. *Mathematical Understanding(convolution)*

The filtered image is the convolution of the original image with the filter impulse response (or "mask"). So, if i(x,y) denotes the original image and r(x,y) denotes the filter impulse response, then their convolution is

$$i'(x,y) = (r * i)(x,y)$$

$$= \sum_{s=-a}^{a} \sum_{t=-b}^{b} r(s,t)\, i(x-s, y-t)$$

Note that, with respect to the indices (s, t), this means that either the impulse response or the original image (original image in the equation) is mirrored vertically and horizontally before the pixel-wise product and sum.

D. *Mathematical Understanding(correlation)*

The concept of correlation, which is somewhat similar to convolution. Mathematically is,

$$(r * f)(x,y) = \sum_{s=-a}^{a} \sum_{s=-b}^{b} r(s,t)\, i(x+s, y+t)$$

#No Mirroring involved

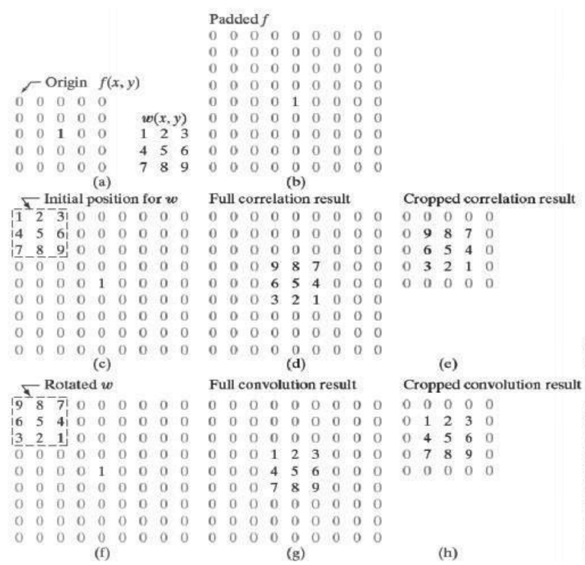
• Differences to statistics:

- This is called cross-correlation in statistics.
- Here, the normalizing factor can be neglected.

This concept is important for applications such as matched filtering. [5]

E. *Convolution versus Correlation*

Filter designing in convolution technique uses the matrix which is just the mirrored image of the correlation technique matrix.



III. APPLICATION OF LINEAR TRANSFORMATION IN FILTER DESIGNING

Separable Filter act in place of subspace of all possible linear filters. An image filter is separable if it is often expressed as the outer product of two vectors. For example,

$$h = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 2 \end{bmatrix} = \mathbf{uv}^T$$

• Thus, separable filters are often applied in two steps:
1. Filter along rows,
2. Filter along columns
(or vice-versa).
• The advantage is reduced computation.
For an I × J image and an L × M mask,

I. Direct approach is O(IJLM)
II. Separable approach is O(IJ(L + M)).

IV. Image Smoothing filters (Averaging filter or low pass filter)

Basic idea: replace each pixel by the average of the pixels in a square window surrounding the pixel.
• Example: for a 3 × 3 averaging filter,

$$i'(x,y) = \frac{1}{9} * \sum_{s=-1}^{1} \sum_{t=-1}^{1} i\,(x-s, y-t)$$

• Extends the thought of "moving average" for images.
• this suggests that the mask is constant,
with all values adequate to 1/9 during this case.

V. Image Smoothing filters (Averaging filter or low pass filter without constant multiplication)

General case: For an n × n averaging filter,

$$r\,(x,y) = \frac{1}{n^2} \underbrace{\left.\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}\right\}}_{n \text{ columns}} n \text{ rows.}$$

Where, typically, n is an odd number.
There is fact that, in all the filters if we add all elements of mask then the sum always will be 1. Let's see why ?

Averaging with a 3 × 3 averaging filter:

Original image

100	100	100	100	100
100	200	205	203	100
100	195	200	200	100
100	200	205	195	100
100	100	100	100	100

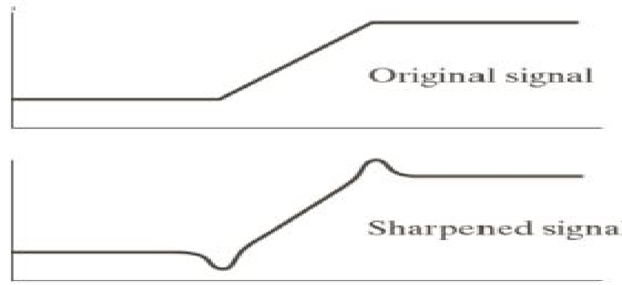
$\longrightarrow$

Filtered image

56	89	101	90	56
88	144	167	145	89
99	167	200	168	100
88	144	166	144	88
56	89	100	89	55

VI. Image sharpening filter (High pass filter)

The main idea of sharpening is that the enhance line Structures of other details in a picture .
• Thus, the improved image contains the first image with the road structures and edges within the image emphasized.



• Line structures and edges are often obtained by applying a difference operator (an high-pass filter) on the image.
• Resulting operation may be a weighted averaging operation, during which some weights are going to be negative(why?).
Criteria for designing an high-pass filter:

$$\sum_{s=-s0}^{s1} \sum_{t=-t0}^{t1} h\,(s,t) = 0$$

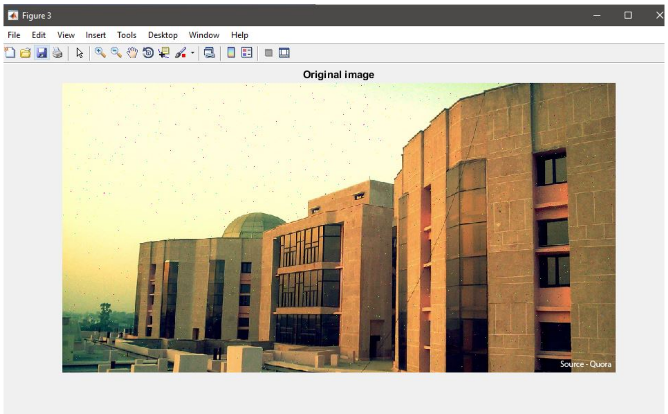
• A filter is isotropic if it's rotation invariant.
• Samples of isotropic high-pass filters:

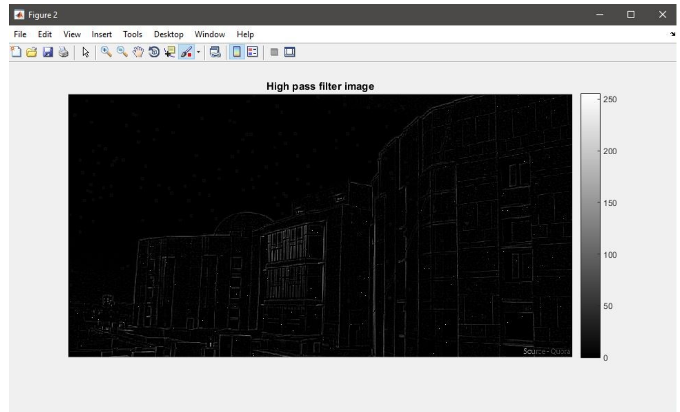
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

VII. Experiment and Analysis using MATLAB

For, experiments we extensively used the MATLAB's Image Processing Toolbox™ for performing the analysis and experiments in our designed filters.
Original Image,

Salt pepper noise added with intensity 0.0002,





Generated low pass filter matrix with coefficients,

	1	2	3	4	5
1	0.0400	0.0400	0.0400	0.0400	0.0400
2	0.0400	0.0400	0.0400	0.0400	0.0400
3	0.0400	0.0400	0.0400	0.0400	0.0400
4	0.0400	0.0400	0.0400	0.0400	0.0400
5	0.0400	0.0400	0.0400	0.0400	0.0400

Generated high pass filter matrix with coefficients,

	1	2	3	4	5
1	-0.0400	-0.0400	-0.0400	-0.0400	-0.0400
2	-0.0400	-0.0400	-0.0400	-0.0400	-0.0400
3	-0.0400	-0.0400	0.9600	-0.0400	-0.0400
4	-0.0400	-0.0400	-0.0400	-0.0400	-0.0400
5	-0.0400	-0.0400	-0.0400	-0.0400	-0.0400

Image generated using low pass filter,

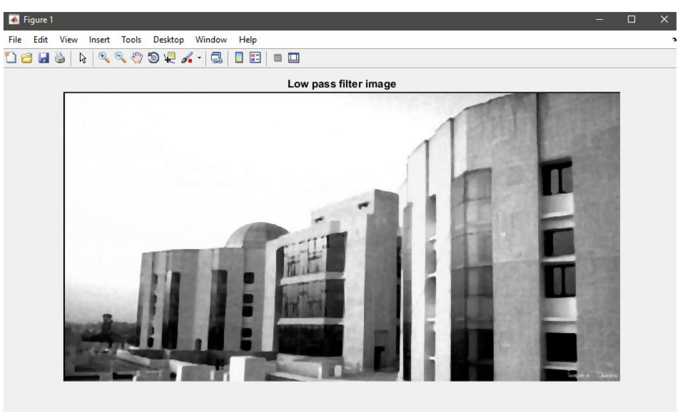


Image generated using high pass filter,

VIII. CONCLUSION

This report draws attention towards mathematical implementation of different categories of filters and their application in the field of Image processing. Apart from that, they also contribute extensively in the field of Finite Impulse Response audio system, speech signal processing and image processing. Not only the design of filter in the specified field is essential but also the concept itself can be generalized to solve many future object rendering problems.

IX. APPENDIX

MATLAB running code and the image link attached after that,

```
cls

clear all
close all

a=imread('D:\tuts\d5.jpg');

b=size(a);

if size(b,2)==3
    a = rgb2gray(a);
End

a = imnoise(a,'salt & pepper');

n=input("Enter the size of mask");

n1=ceil(n/2);

a=double(a);

lpf=(1/n^2)*ones(n);
```

```
hpf=-lpf;

hpf(n1,n1)=(n^2-1)/n^2;
c=0;
h=0;
for i=n1:b(1)-n1
   for j=n1:b(2)-n1
      p=1;
      for k=1:n
         for l=1:n
         c(p)=a(i-n1+k,j-n1+l);
         p=p+1;
         end
      End

      d(i,j)=median(c);
      c=0;
   end
end
e=uint8(d);
figure;imshow(e);title('low pass image');

for i=n1:b(1)-n1
   for j=n1:b(2)-n1
      for k=1:n
         for l=1:n
         h=h+a(i-n1+k,j-n1+l)*hpf(k,l);
         end
      end

      g(i,j)=h;
      h=0;
   end
End

f=uint8(g);
figure;imshow(f);title('high pass image');
figure;imshow(uint8(a));title('Original
image');
```

Image Link,

https://drive.google.com/file/d/1QhoOBA7sA4yFgMgDTCd8EDtdRJKBvCjS/view?usp=sharing

X. Acknowledgment

We would like to express our special thanks of gratitude to our teacher **Dr. Mohammed Javed** , who gave us the assignment which helped me to enhance our understanding in the field of digital image processing and application of linear transformation. We also want to thank **Mr. Bulla Rajesh**, who mentored us to complete this assignment, Successfully.

References

[1] http://literature.cdn.keysight.com/litweb/pdf/ads2008/dfilter/ads2008/Designing_a_Digital_Filter.html
[2] https://en.m.wikipedia.org/wiki/Low-pass_filter
[3] https://web.stanford.edu/class/ee368/
[4] http://www-math.mit.edu/~gs/
[5] https://sisu.ut.ee/sites/default/files/imageprocessing/files/image_filtering_in_spatial_domain.pdf