PERFORM SPATIAL DOMAIN FILTERING

Abstract:
The purpose of this practical is to study image filtering in spatial domain. It consists of three parts the first one discusses the spatial filtering of an image using a spatial mask 3x3, and then this mask is used in blurring filter and Laplacian. The second part studies the order statistics filters, specially the median filter. The third part discusses the unsharp masking filter as an example we implement the high-boost filter.

Technical Discussion

PART 1:

• Spatial filtering is performed by convolving the image w
ith a mask or a kernel
• Spatial filters include sharpening, smoothing , edge detection, noise removal, etc.
• In general, linear filtering of an image f of size M x N with filter size m x n is given by the expression

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$a = (m-1)/2, \mathrm{b} = (\mathrm{n}-1)/2$$

• Smoothing Spatial Filters

- The output of a smoothing spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.
- These filters are sometimes called averaging filters and also lowpass filters
- Two types of masks of the spatial filter

$$\frac{1}{9} \text{ X } \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{16} \text{ X } \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- **The Laplacian**
  - The Laplacian of an image is define as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) - 2(f(x,y) + f(x-1,y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) - 2(f(x,y) + f(x,y-1)$$

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

Types of masks of the *Laplacian* filter

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

PART 2:

• Order statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in an area covered by the filter
• The best known example in this category is median filter
• median filter

- Median filters replace the value of the pixel by the median of the gray levels in the neighborhood of that pixel

PART 3:

• Unsharp Masking consists of subtracting a blurred version of the image from the image itself

$$f_s(x, y) = f(x, y) - \tilde{f}(x, y)$$

- The best known example in this category is high boost filter

high boost filter

- A slight generalization of unsharp masking is called high boost filters

$$f_{hb}(x, y) = A\ f(x, y) - \tilde{f}(x, y)$$

- Types of masks of the high boost filter

0	−1	0
−1	$A+4$	−1
0	−1	0

−1	−1	−1
−1	$A+8$	−1
−1	−1	−1

Appendix
Program Listings
Part 1(Matlab)

```
function FltrIm=SpatialFilter(Im,Mask)
[M N]=size(Im);
[m n]=size(Mask);
a=(m-1)/2;
b=(n-1)/2;
g=zeros(M,N);
for x=1:M,
   for y=1:N,
      for s=-a:a,
         for t=-b:b
            if((x+s)>0 & (y+t)>0 & (x+s)<=M & (y+t)<=N)
               g(x,y)=g(x,y)+Mask(s+a+1,t+b+1)*Im(x+s,y+t);
            end
         end
      end
   end
end
FltrIm=g;
%%%%%%%%%%%%%main Program
clear all;
clc;
im=readpgm('E:\Classes\MATLAB_PROGRAM\ECE618\pgm_images\bridge.pgm');
L_max=max(max(im))
L_min=min(min(im))
imshow(im,[0 255]);
% m=(1.0/(3*3.0))*ones(3,3);
m=[0 -1 0;-1 4 -1;0 -1 0];
IMG1=SpatialFilter(im,m);
L_mx=max(max(IMG1))
L_mn=min(min(IMG1))
figure,imshow(IMG1,[L_mn L_mx]);
```

Part 2(Matlab)

```
function FltrIm=MedianFilter(Im)
[M N]=size(Im);
a=2;
b=2;

for x=1:M,
    for y=1:N,
        k=1;
        for s=-a:a,
            for t=-b:b
                if((x+s)>0 & (y+t)>0 & (x+s)<=M & (y+t)<=N)
                    g(k)=Im(x+s,y+t);
                else
                    g(k)=0;
                end
                 k=k+1;
            end
        end
        FltrIm(x,y)=median(g);
    end
end
%%%%%%%%%%%%%main Program
clear all;
clc;
im=readpgm('E:\Classes\MATLAB_PROGRAM\ECE618\pgm_images\camera.pgm');
[N M]=size(im);
immax=max(max(im));
L_max=max(max(im))
L_min=min(min(im))
figure(1),imshow(im,[L_min L_max]);
[N M]=size(im);
```

```
L_max=max(max(im))
L_min=min(min(im))
figure(1),imshow(im,[L_min L_max]);
[N M]=size(im);
Vect=reshape(im,N*M,1);
[Pr x]=hist(Vect,[L_min:L_max]);
Pr=Pr/(N*M);
figure(2), plot(x,Pr),grid;
in = immax*imnoise(1.0*im/immax,'salt & pepper',0.02);
L_mx=max(max(in))
L_mn=min(min(in))
figure(3),imshow(in,[L_mn L_mx]);
Vect=reshape(in,N*N,1);
[Pr x]=hist(Vect,[L_mn:L_mx]);
Pr=Pr/(N*N);
figure(4), plot(x,Pr),grid;
IMG1=MedianFilter(in);
L_mx=max(max(IMG1))
L_mn=min(min(IMG1))
figure(5),imshow(IMG1,[L_mn L_mx]);
Vect=reshape(IMG1,N*N,1);
[Pr x]=hist(Vect,[L_mn:L_mx]);
Pr=Pr/(N*N);
figure(6), plot(x,Pr),grid;
```

Part 3(Matlab)

```
function FltrIm=SpatialFilter(Im,Mask)
[M N]=size(Im);
[m n]=size(Mask);
a=(m-1)/2;
b=(n-1)/2;
g=zeros(M,N);
for x=1:M,
   for y=1:N,
      for s=-a:a,
         for t=-b:b
            if((x+s)>0 & (y+t)>0 & (x+s)<=M & (y+t)<=N)
               g(x,y)=g(x,y)+Mask(s+a+1,t+b+1)*Im(x+s,y+t);
            end
         end
      end
   end
end
FltrIm=g;
%%%%%%%%%%%%%%main Program
clear all;
clc;
im=readpgm('E:\Classes\MATLAB_PROGRAM\ECE618\pgm_images\horiz.pgm');
```

```
[N M]=size(im);
L_max=max(max(im));
% im=im*1.0/L_max;
L_max=max(max(im))
L_min=min(min(im))
figure,imshow(im,[0 255]);
A=3;
m=[-1 -1 -1;-1 A+8 -1;-1 -1 -1];
IMG1=SpatialFilter(im,m);
L_mx=max(max(IMG1))
L_mn=min(min(IMG1))
figure,imshow(IMG1,[L_mn L_mx]);
dm=im-IMG1;
L_mx=max(max(dm))
L_mn=min(min(dm))
figure,imshow(dm,[L_mn L_mx]);
```

Results

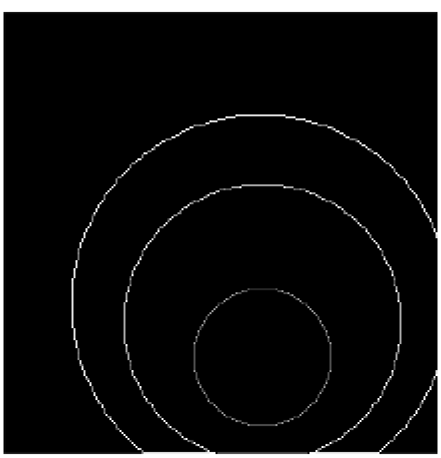
Part 1

A) Spatial filter (blurring & Laplacian filter) (circles image)

- Fig(1-1) shows the original image
- Fig(1-2) shows the blurred image
- Fig(1-3) shows the Laplacian output image

Fig(1-1)

Fig(1-2)



Fig(1-3)

B) Spatial filter (blurring & Laplacian filter)

- Fig(1-4) shows the original image
- Fig(1-5) shows the blurred image
- Fig(1-6) shows the Laplacian output image

Fig(1-4)

Fig(1-5)

Part 2

A) Order statistics filter (Median filter) (lena image)

- Fig(2-1) shows the original image
- Fig(2-2) shows the original image histogram
- Fig(2-3) shows the noisy image ('salt & pepper' noise)
- Fig(2-4) shows the noisy image histogram
- Fig(2-5) shows the output median filter image
- Fig(2-6) shows the output median filter image histogram

Fig.(2-1)

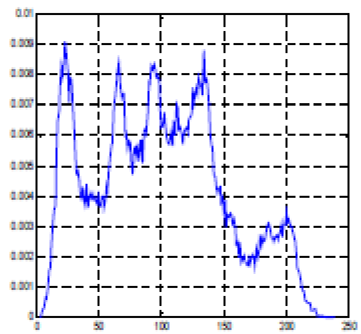


Fig.(2-2)

Fig.(2-3)

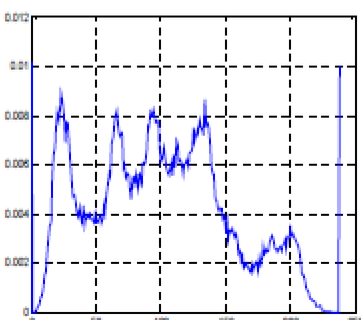


Fig.(2-4)

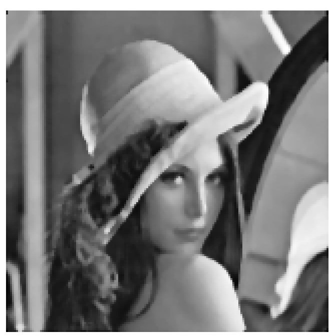


Fig.(2-5)

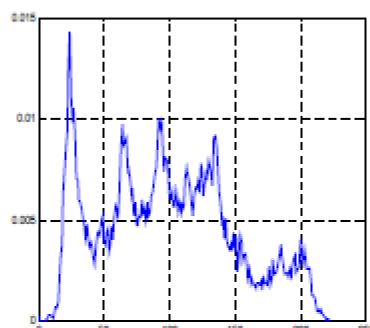


Fig.(2-6)

B) Order statistics filter (Median filter) (camera image)

- Fig(2-7) shows the original image
- Fig(2-8) shows the original image histogram
- Fig(2-9) shows the noisy image ('salt & pepper' noise)
- Fig(2-10) shows the noisy image histogram
- Fig(2-11) shows the output median filter image
- Fig(2-12) shows the output median filter image histogram

Fig.(2-7)

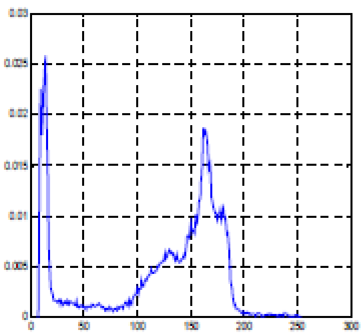


Fig.(2-8)

Fig.(2-9)

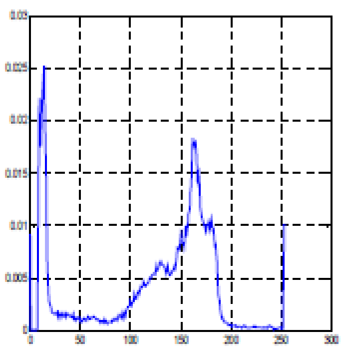


Fig.(2-10)

Fig.(2-11)

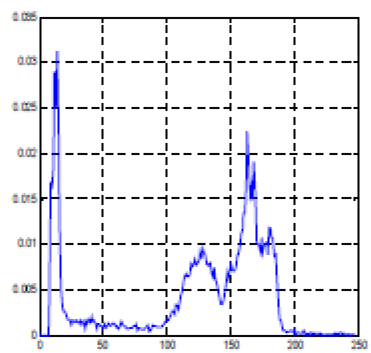


Fig.(2-12)

- Fig(3-5) shows the original image (goldhill image)
- Fig(3-6) shows output High-boost filter image for A=1.4
- Fig(3-7) shows the original image (horiz image)
- Fig(3-8) shows output High-boost filter image for A=1.9

Fig(3-5)

Fig(3-6)

Fig(3-7)

Fig(3-8)

Results Discussion

PART 1:

- As shown from the results the output of a smoothing spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask and the output image become more blurred (e.g. fig.(1.2))

- The output of a Laplacian filter (e.g. Fig.(1.3)) is the edges and there is no blurred parts

PART 2:

- As shown from the results the median filter can remove the salt & paper noise (e.g. Fig.(2-3)) but make the output image (e.g. Fig.(2-5)) is more blurred than original one (e.g. Fig.(2-1)).
- The histogram of the noised image (e.g. Fig.(2-4)) contains two pulses in the sides and the median filter removes them as shown in the histogram of the output image (e.g. Fig.(2-6))

PART 3:

- As shown from the results Unsharp masking filter (High-boost filter) removes the blurred parts and enhances the edges
- We control the ratio of enhancement by the value A if A=0 High-boost filter will become Laplacian filter and if $1<A<3$ the output image (e.g. Fig.(3-2)) will be more clear than original one (e.g. Fig.(3-1)).