

# Assignment 6

## SPATIAL FILTERING USING MATLAB

Mudit Dholakia

MT-006

Guide:-Prof. Tushar. V. Ratanpara

# AIM 1

- Implement spatial filtering program as follows.
- Load the image data
- Use a spatial filter to get the horizontal edge of the image.
- Use a spatial filter to get the vertical edge of the image (read the MATLAB documentation on fspecial).
- Add the horizontal edge matrix to the vertical edge matrix.

# Code:-

- `clc;`
- `clear all;`
- `close all;`
- `I=imread('lena.jpg');`
- `h=fspecial('sobel');`
- `v=(fspecial('sobel'))';`
- `hor=imfilter(I,h);`
- `ver=imfilter(I,v);`
- `final=hor+ver;`
- `imshow(final);`



Original Image



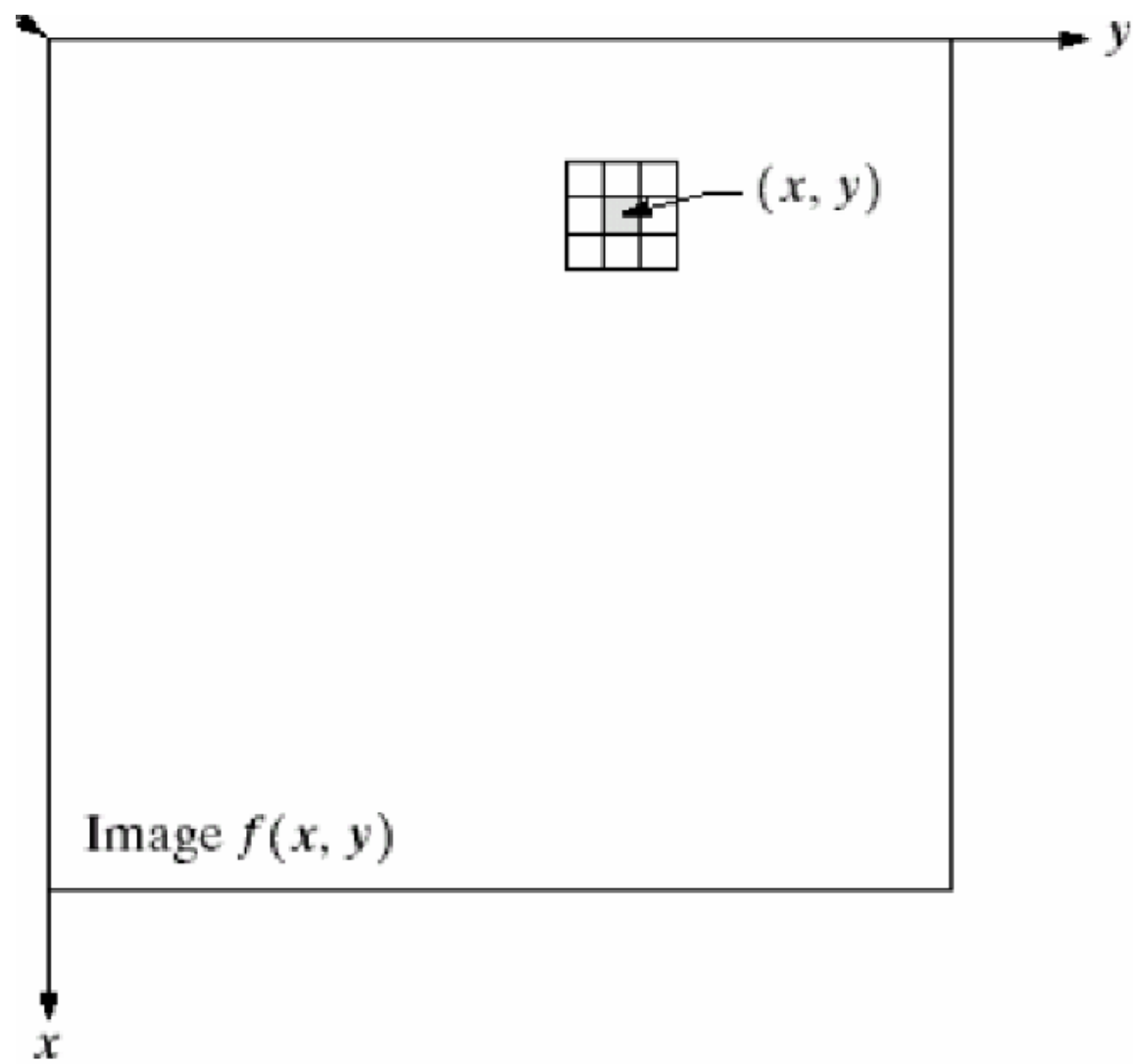
Resultant Image

# AIM 2

- What Is Image Filtering in the Spatial Domain?

# Spatial Filtering

- Define a center point  $(x,y)$
- Perform an operation that involves only the pixels in a predefined neighbourhood
- Result of the operation response of the process at that point
- Repeat the process for every pixel in the image



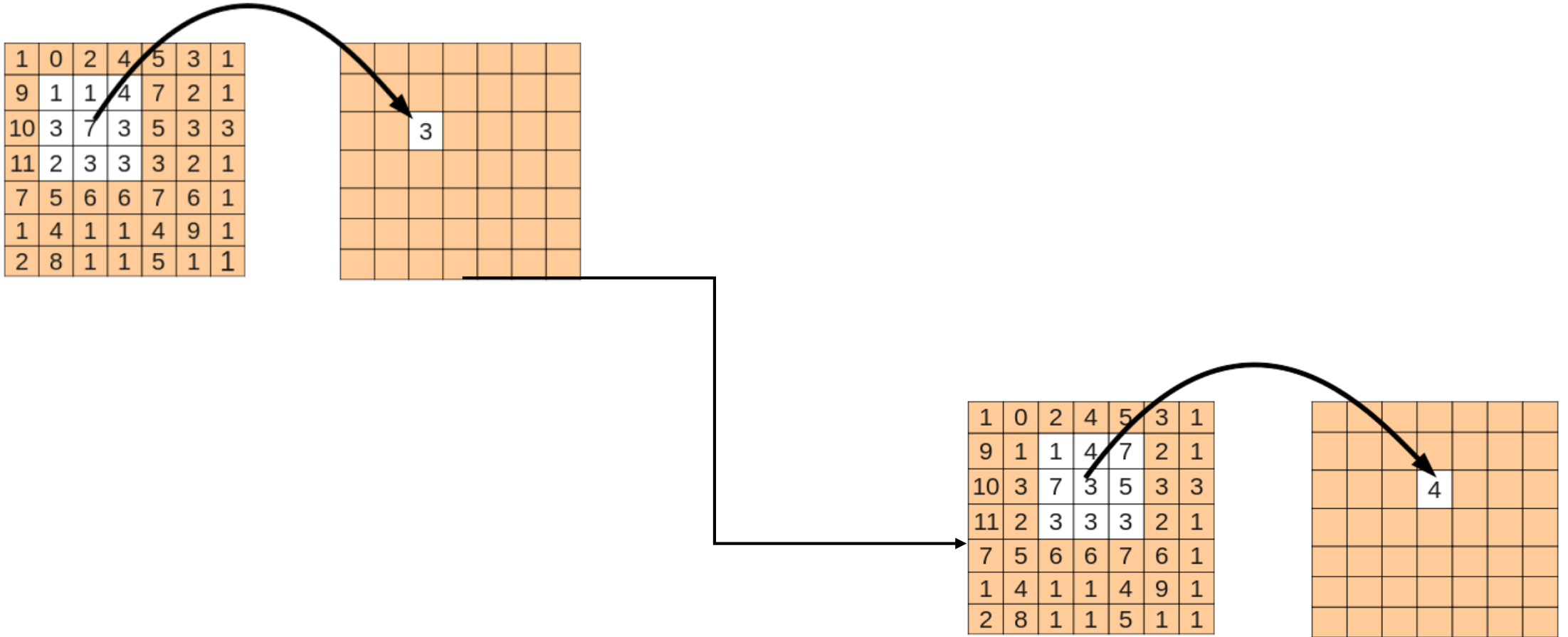
# Spatial Filtering

- Output is a function of a pixel value and its neighbours
- Possible operations are: sum, weighted sum, average, weighted average, min, max, median etc.
- Example: 3\*3 neighbourhood

$$\text{OPERATION} \left( \begin{bmatrix} f(x-1, y-1) & f(x-1, y) & f(x-1, y+1) \\ f(x, y-1) & f(x, y) & f(x, y+1) \\ f(x+1, y-1) & f(x+1, y) & f(x+1, y+1) \end{bmatrix} \right)$$



# Spatial Filtering



# Spatial Filtering

- Linear Filters:
  - Smoothing Filters
    - Mean Filter
    - Gaussian Filter
  - Edge enhancing Filters
    - Sobel Filter
    - Prewitt Filter
    - Laplace Filter
- Non-linear Filters:
  - Min, Max
  - Median, Percentile Filters

# AIM 3

- Implement matlab programs of smoothing linear filters.(without using fspecial and imfilter function)
  - Average filter
  - Weighted average filter

# Average Filter

# Code:-

- `clc;`
- `clear all;`
- `close all;`
- `a=imread('b.tif');`
- `figure, imshow(a)`
- `an=imnoise(a,'gaussian');`
- `[m,n]=size(an);`
- `figure, imshow(uint8(an))`
- `b=input('Enter Averaging Mask size: ');`
- `z=ones(b);`
- `[p,q]=size(z);`
- `w=1:p;`
- `x=round(median(w));`
- `anz=zeros(m+2*(x-1),n+2*(x-1));`

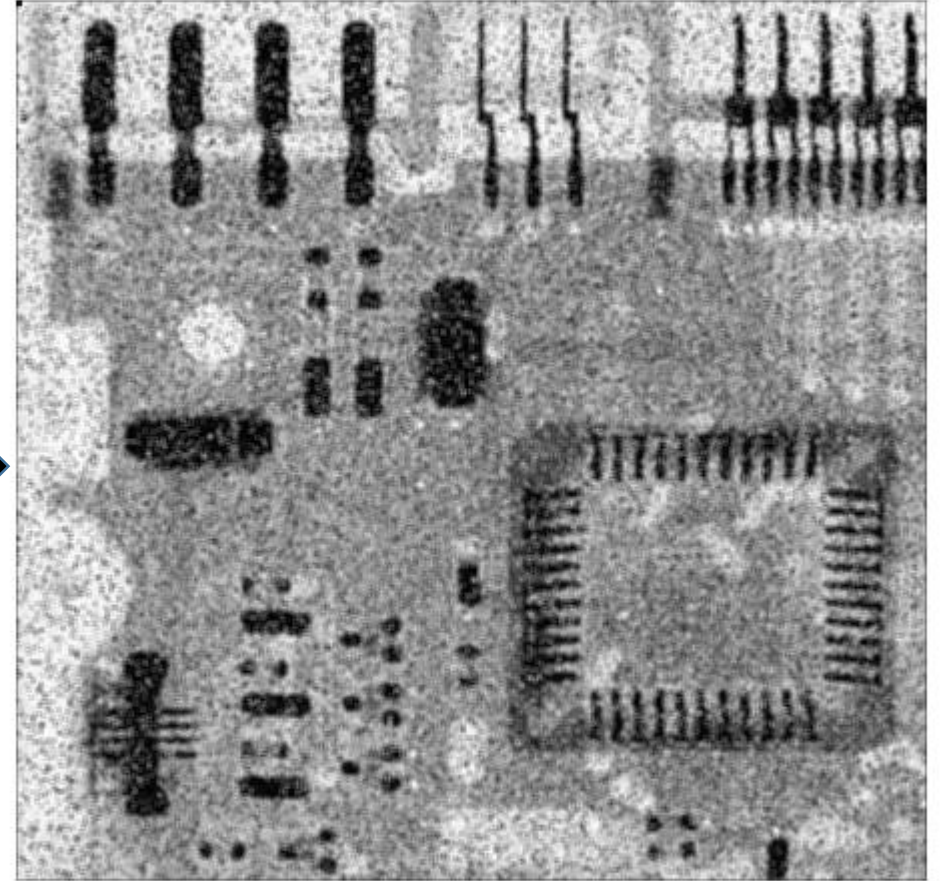
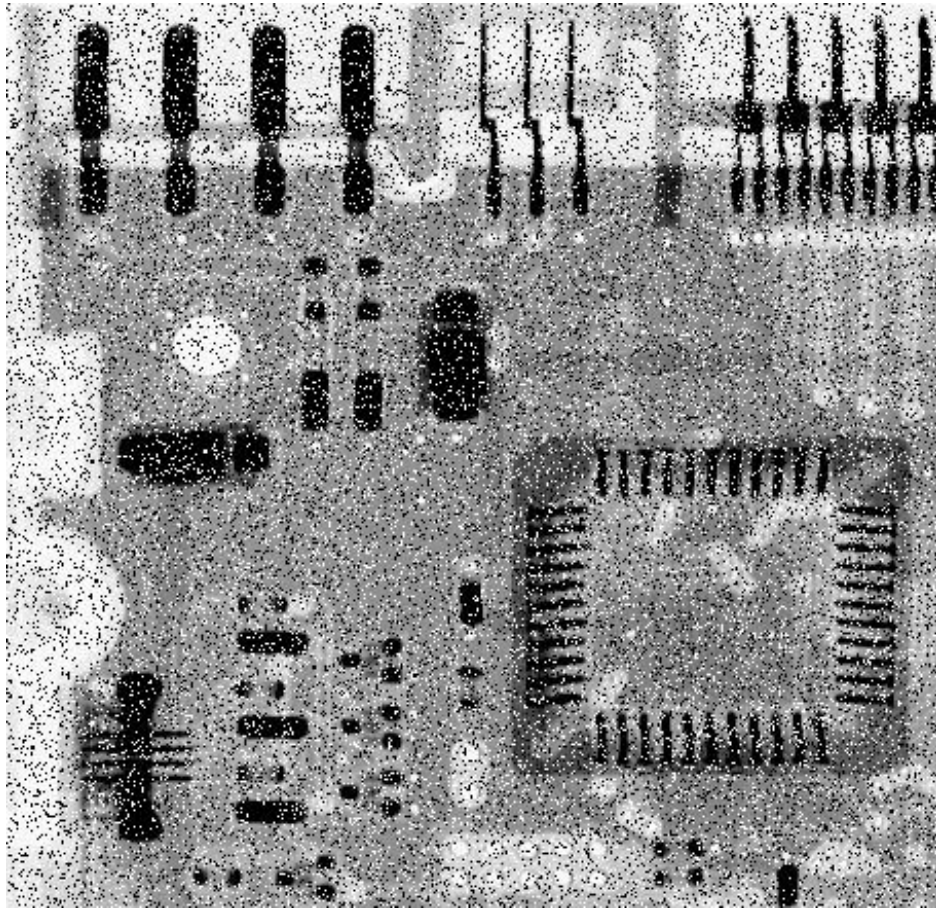
## Code:-(Cont'd)

- for i=x:(m+(x-1))
- for j=x:(n+(x-1))
- anz(i,j)=an(i-(x-1),j-(x-1));
- end
- end
- figure, imshow(uint8(anz))
- 
- sum=0;
- x=0;
- y=0;

# Code:-(Cont'd)

- for i=1:m
- for j=1:n
- for k=1:p
- for l=1:q
- sum= sum+anz(i+x,j+y)\*z(k,l);
- y=y+1;
- end
- y=0;
- x=x+1;
- end
- x=0;
- ans(i,j)=(1/(p\*q))\*(sum);
- sum=0;
- end
- end
- figure, imshow(uint8(ans))

# Average Filter





# Weighted Average Filter

# Code:-

- `clc`
- `clear all`
- `close all`
- `a=imread('cameraman.tif');`
- `figure, imshow(a)`
- `an=imnoise(a,'gaussian');`
- `[m,n]=size(an);`
- `figure, imshow(uint8(an))`
- `b=input('Enter Averaging Mask size: ');`
- `%z=ones(b);`

# Code:-(Cont'd)

- for i=1:b
- for j=1:b
- z(i,j)=input('Mask Values:-')
- end
- end
- smsk=0;
- for i=1:b
- for j=1:b
- smsk=smsk+sum(z(i,j));
- end
- end
- [p,q]=size(z);

## Code:-(Cont'd)

- `w=1:p;`
- `x=round(median(w));`
- `anz=zeros(m+2*(x-1),n+2*(x-1));`
- `for i=x:(m+(x-1))`
- `for j=x:(n+(x-1))`
- `anz(i,j)=an(i-(x-1),j-(x-1));`
- `end`
- `end`
- `figure, imshow(uint8(anz))`

# Code:-(Cont'd)

- sum=0;
- x=0;
- y=0;
- for i=1:m
- for j=1:n
- for k=1:p
- for l=1:q
- sum= sum+anz(i+x,j+y)\*z(k,l);
- y=y+1;
- end
- y=0;
- x=x+1;
- end

- x=0;
- ans(i,j)=(1/(p\*q))\*(sum);
- sum=0;
- end
- end
- for i=1:b
- for j=1:b
- ans(i,j)=ans(i,j)/smsk;
- end
- end
- figure, imshow(uint8(ans))

# Weighted Average Filter



# AIM 4

- Implement matlab programs of order-statistic (Nonlinear) filters.  
(without using fspecial and imfilter)
- Median filter
- Max filter
- Min filter

# Mean Filter



# Code:-

- clear all
- close all
- clc
- I = imread('cameraman.tif');
- I = imnoise(I, 'salt & pepper', 0.01);
- [r c] = size(I);
- Rep = zeros(r + 2, c + 2);
- for x = 2 : r + 1
- for y = 2 : c + 1
- Rep(x,y) = I(x - 1, y - 1);
- end
- end

# Code:-(Cont'd)

- `B = zeros(r, c);`
- `for x = 1 : r`
- `for y = 1 : c`
- `for i = 1 : 3`
- `for j = 1 : 3`
- `q = x - 1;   w = y - 1;`
- `array((i - 1) * 3 + j) = Rep(i + q, j + w);`
- `end`
- `end`
- `B(x, y) = median(array(:));`
- `end`
- `end`
- `figure, imshow(I);`
- `figure, imshow(uint8(B));`

# Mean Filter



# Max Filter

# Code:-

- `clc`
- `clear all`
- `close all`
- `a=imread('cameraman.tif');`
- `figure, imshow(a)`
- `an=imnoise(a,'gaussian');`
- `[m,n]=size(an);`
- `figure, imshow(uint8(an))`
- `b=input('Enter Averaging Mask size: ');`
- `z=ones(b);`
- `[p,q]=size(z);`

## Code:-(Cont'd)

- 
- `w=1:p;`
- `x=round(median(w));`
- `anz=zeros(m+2*(x-1),n+2*(x-1));`
- 
- `for i=x:(m+(x-1))`
- `for j=x:(n+(x-1))`
- `anz(i,j)=an(i-(x-1),j-(x-1));`
- `end`
- `end`
- `figure, imshow(uint8(anz))`

# Code:-(Cont'd)

- sum=0;
- x=0;
- y=0;
- minn=0;
- for i=1:m
- for j=1:n
- for k=1:p
- for l=1:q
- sum=anz(i+x,j+y)\*z(k,l);
- if sum >= minn
- minn=sum;
- end
- y=y+1;
- end

- y=0;
- x=x+1;
- end
- x=0;
- ans(i,j)=minn;
- sum=0;
- minn=0;
- end
- end
- figure, imshow(uint8(ans))

# Max Filter





Min Filter

# Code:-

- `clc`
- `clear all`
- `close all`
- 
- `a=imread('cameraman.tif');`
- `figure, imshow(a)`
- `an=imnoise(a,'gaussian');`
- `[m,n]=size(an);`
- `figure, imshow(uint8(an))`
- `b=input('Enter Averaging Mask size: ');`
- `z=ones(b);`
- `[p,q]=size(z);`

## Code:-(Cont'd)

- `w=1:p;`
- `x=round(median(w));`
- `anz=ones(m+2*(x-1),n+2*(x-1));`
- 
- `for i=x:(m+(x-1))`
- `for j=x:(n+(x-1))`
- `anz(i,j)=an(i-(x-1),j-(x-1));`
- `end`
- `end`
- `figure, imshow(uint8(anz))`

# Code:-(Cont'd)

- sum=0;
- x=0;
- y=0;
- minn=0;
- for i=1:m
- for j=1:n
- for k=1:p
- for l=1:q
- sum=anz(i+x,j+y)\*z(k,l);
- if sum < minn
- minn=sum;
- end
- y=y+1;
- end

- y=0;
- x=x+1;
- end
- x=0;
- ans(i,j)=minn;
- sum=0;
- minn=0;
- end
- end
- figure, imshow(uint8(ans))

# Min Filter



# AIM 5

- Implement matlab programs of Sharpening filters using following operators.
  - Laplacian operator
  - Sobel operator
  - Prewitt operator

# Code:-

- `clc;`
- `clear all;`
- `close all;`
- `I=imread('lena.jpg');`
- `s=fspecial('sobel');`
- `l=fspecial('laplacian');`
- `p=fspecial('prewitt');`
- `sb=imfilter(I,s);`
- `lp=imfilter(I,l);`
- `pw=imfilter(I,p);`

## Code:-(Cont'd)

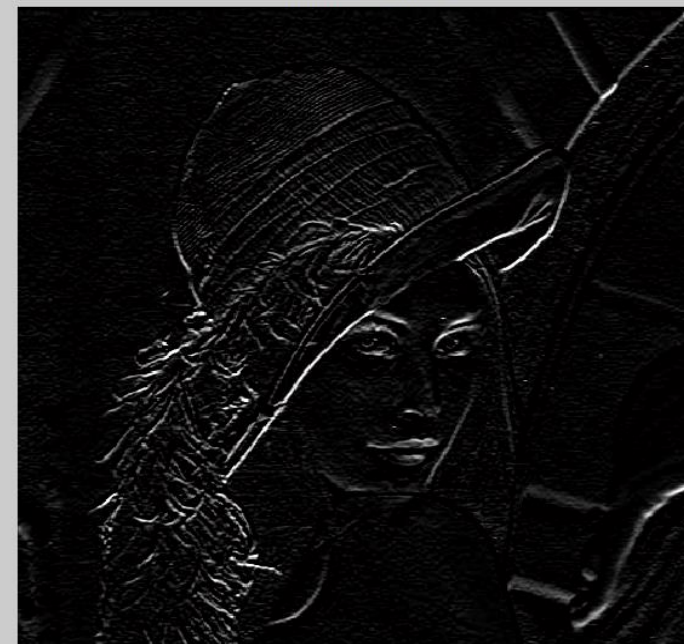
- `figure,imshow(sb);title('Sobel operator in fspecial()')`
- `figure,imshow(lp);title('Laplacian operator in fspecial()')`
- `figure,imshow(pw);title('Prewitt operator in fspecial()')`



Sobel operator in fspecial()

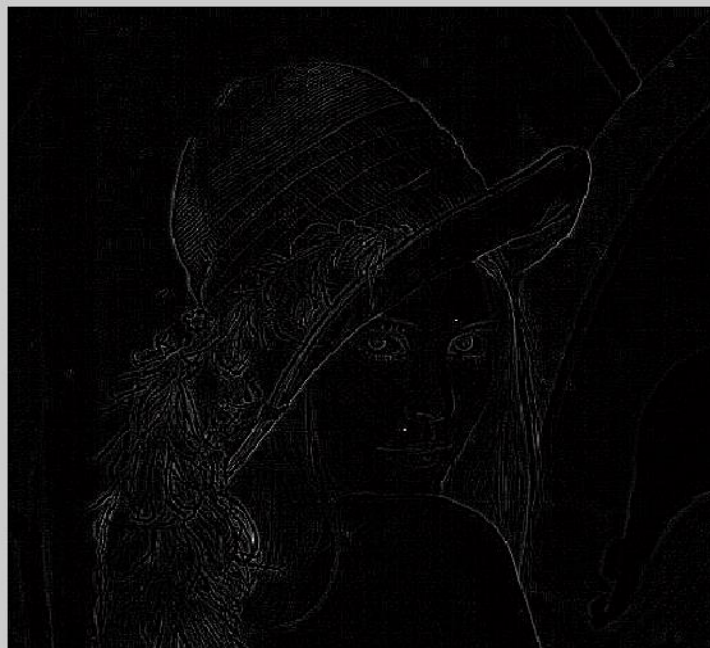


Prewitt operator in fspecial()



# Laplacian

Laplacian operator in fspecial()



# Sobel

# Prewitt

THANK YOU