EXPERINMENT 10-11

Aim : To Study Morphological Operations and Image Segmentation.

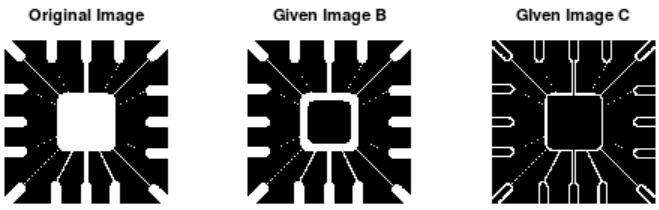
❖ Exercises :

1. **For the given image Fig0905(a)(wirebond-mask).tif image from "chapter 9 images", obtain image B and C using morphological operations.**

Code :

```
pkg load image;
r = imread("/home/nihar/Desktop/SEM 7/LABS/IP/Lab10_11/Fig0905(a)(wirebond-mask).ti
subplot(1,3,1);
imshow(r);
title("Original Image");

struct_ele1 = ones(45);
s1 = imerode(r,struct_ele1);
subplot(1,3,2);
imshow(r-s1);
title("Given Image B");

struct_ele2 = ones(11);
s2 = imerode(r,struct_ele2);
subplot(1,3,3);
imshow(r-s2);
title("GIven Image C");
```

Output :

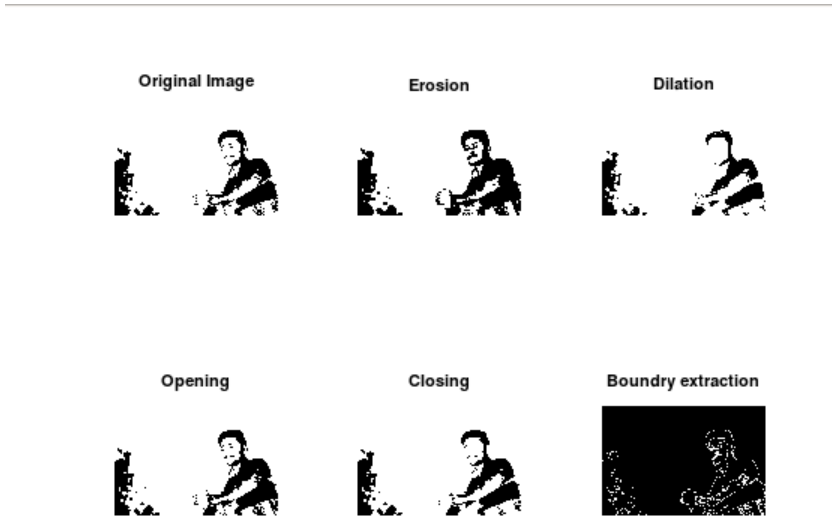


2. Consider your black and white photo, perform erosion, dilation, opening, closing and boundary extraction. Explain the impact of size of the structuring element on these operations.

Code :

```
pkg load image;
r=im2bw(imread("/home/nihar/Desktop/SEM 7/LABS/IP/Lab5/1.jpeg"));
subplot(2,3,1);
imshow(r);
title("Original Image");

#Erosion
struct_ele = ones(7);
s1 = imerode(r,struct_ele);
subplot(2,3,2);
imshow(s1);
title("Erosion");

#Dilation
s2 = imdilate(r,struct_ele);
subplot(2,3,3);
imshow(s2); title("Dilation");

#Opening
s3 = imdilate(imerode(r,struct_ele),struct_ele);
subplot(2,3,4);
imshow(s3); title("Opening");

#Closing
s4 = imerode(imdilate(r,struct_ele),struct_ele);
subplot(2,3,5);
imshow(s4); title("Closing");

#Boundary extraction
s5 = r-s1;
subplot(2,3,6);
imshow(s5); title("Boundry extraction");
```

Output :

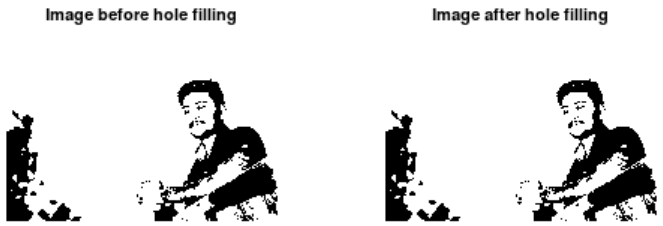


3. Perform hole filling on your black and white photo using morphological operations.

Code :

```
clc
close all
clear all
pkg load image;
r = imread("/home/nihar/Desktop/SEM 7/LABS/IP/Lab5/1.jpeg");
r = im2bw(r);
subplot(1,2,1);
imshow(r);
title("Image before hole filling");

[m,n] = size(r);
rc = not(r);
struct_ele = strel('square',3);
x0 = zeros(m,n);
x0(300,800) = 1;  x0(400,790) = 1;  x0(360,850) = 1;
x0(360,850) = 1;  x0(540,650) = 1;  x0(470,755) = 1;
x0(455,800) = 1;  x0(620,800) = 1;  x0(530,720) = 1;
x0(610,550) = 1;

dilate_image = imdilate(x0,struct_ele);
x1 = dilate_image.*rc;

while(!isequal(x0,x1))
  x0=x1;
  dilate_image = imdilate(x0,struct_ele);
  x1 = dilate_image.*rc;
endwhile

subplot(1,2,2);
s = r+x1;
imshow(s);
title("Image after hole filling");
```

Output :



4. Find a connected component on your black and white photo using morphological operations.

Code :

```
#load image
pkg load image;
r = rgb2gray(imread("/home/nihar/Desktop/SEM 7/LABS/IP/Lab5/1.jpeg"));
imshow(r);
title("Original Image");
r = im2double(r);
figure;

#Horizontal Lines
horizontal_mask = [-1,-1,-1;2,2,2;-1,-1,-1];
horizontal_lines = my_filter(r, horizontal_mask);
subplot(2,2,1);
imshow(abs(horizontal_lines));  title("Horizontal Lines");

#Vertical Lines
vertical_mask = [-1,2,-1;-1,2,-1;-1,2,-1];
vertical_lines = my_filter(r, vertical_mask);
subplot(2,2,2);
imshow(abs(vertical_lines));  title("Vertical Lines");

#Oblique +45 Lines
positive45_mask = [-1,-1,2;-1,2,-1;2,-1,-1];
positive45_lines = my_filter(r,positive45_mask);
subplot(2,2,3);
imshow(abs(positive45_lines));  title("+ 45 Lines");

#Oblique -45 Lines
negative45_mask = [2,-1,-1;-1,2,-1;-1,-1,2];
negative45_lines = my_filter(r,negative45_mask);
subplot(2,2,4);
imshow(abs(negative45_lines));  title("- 45 Lines");
```

Output :

5. **Detect horizontal, vertical, diagonal(+45 degree and -45 degree) lines from your grayscale photo.**

Code :

```
#load image
pkg load image;
r = rgb2gray(imread("/home/nihar/Desktop/SEM 7/LABS/IP/Lab5/1.jpeg"));
imshow(r);
title("Original Image");
r = im2double(r);
figure;

#Horizontal Lines
horizontal_mask = [-1,-1,-1;2,2,2;-1,-1,-1];
horizontal_lines = my_filter(r, horizontal_mask);
subplot(2,2,1);
imshow(abs(horizontal_lines));  title("Horizontal Lines");

#Vertical Lines
vertical_mask = [-1,2,-1;-1,2,-1;-1,2,-1];
vertical_lines = my_filter(r, vertical_mask);
subplot(2,2,2);
imshow(abs(vertical_lines));  title("Vertical Lines");

#Oblique +45 Lines
positive45_mask = [-1,-1,2;-1,2,-1;2,-1,-1];
positive45_lines = my_filter(r,positive45_mask);
subplot(2,2,3);
imshow(abs(positive45_lines));  title("+ 45 Lines");

#Oblique -45 Lines
negative45_mask = [2,-1,-1;-1,2,-1;-1,-1,2];
negative45_lines = my_filter(r,negative45_mask);
subplot(2,2,4);
imshow(abs(negative45_lines));  title("- 45 Lines");
```

Output :

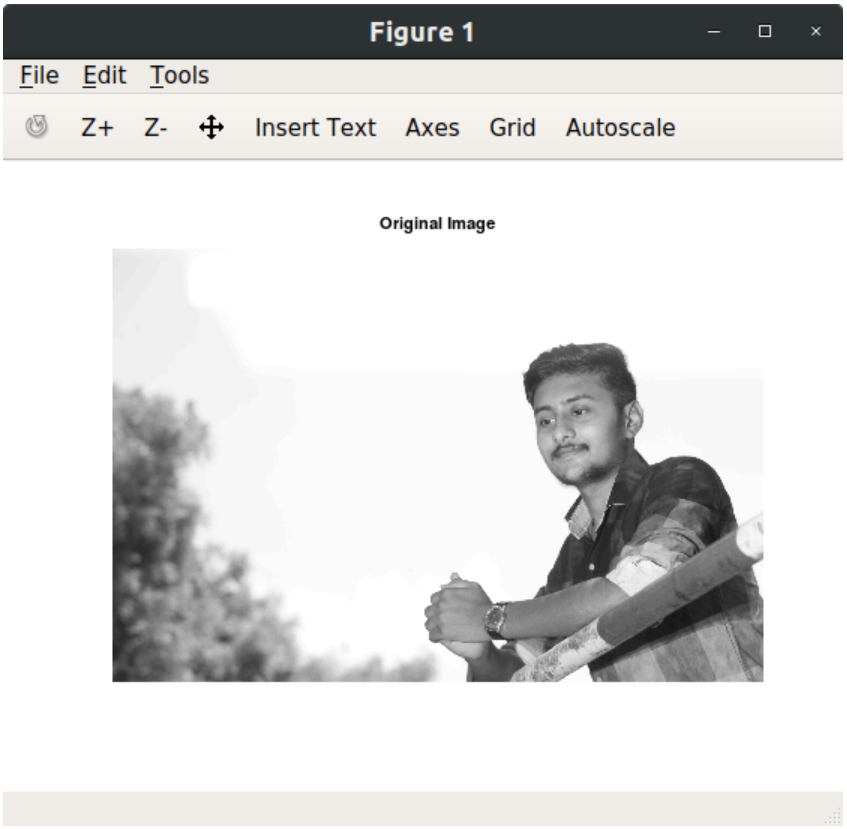


Figure 1
File Edit Tools
Z+ Z- Insert Text Axes Grid Autoscale
Original Image

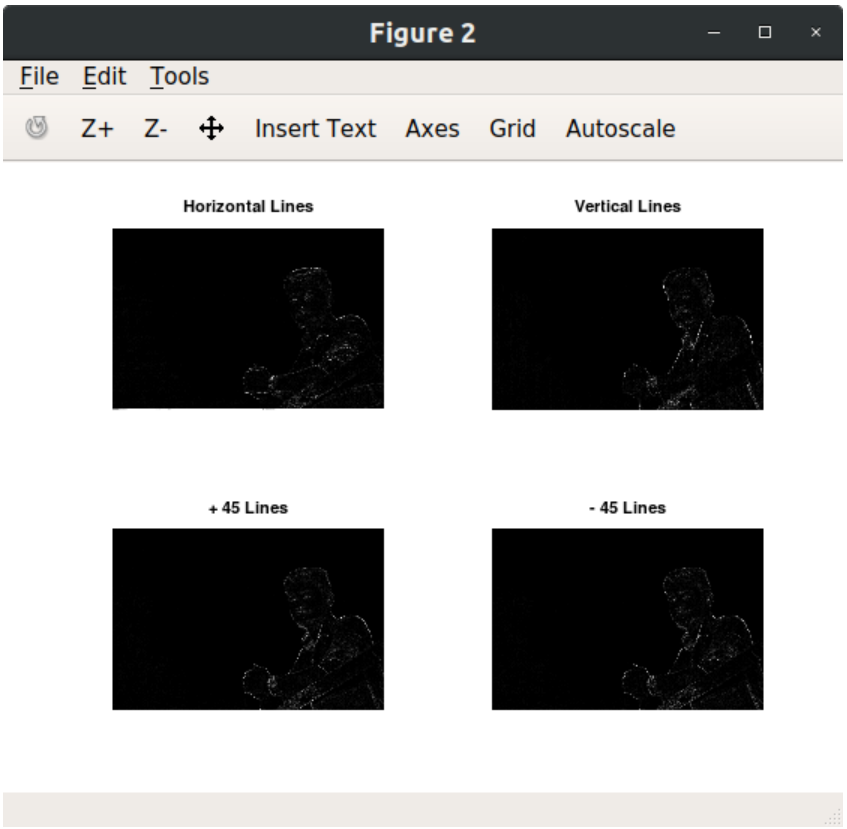


Figure 2
File Edit Tools
Z+ Z- Insert Text Axes Grid Autoscale
Horizontal Lines
Vertical Lines
+ 45 Lines
- 45 Lines

6. **Detect edges in your grayscale photo. Find gradient magnitude and angle using sobel and prewitt masks. Demonstrate the impact of smoothing on it.**

Function my_filter :

```
function s = my_filter(r,kernel)
  [M,N] = size(r);
  [m,n] = size(kernel);
  a = (m-1)/2;
  b = (n-1)/2;
  new_imsize = zeros(M + 2*a,N + 2*b);
  new_imsize(1+a:M+a,1+b:N+b) = r;
  sub_image = ((1/(m*n)).*ones(m,n));
  s = zeros(size(r));
  for i = 1+a:M+a,
    for j = 1+b:N+b,
      k = new_imsize(i-a:i+a,j-b:j+b);
      s(i-a,j-b) = sum(sum(k.*kernel));
    endfor
  endfor
endfunction
```

Code :

```
pkg load image;
r = rgb2gray(imread("/home/nihar/Desktop/SEM 7/LABS/IP/Lab5/1.jpeg"));
r = imresize(r,[1024,1024]);
r = im2double(r);

figure;

#Prewitt Filter
x = [-1,-1,-1;0,0,0;1,1,1];
y = [-1,0,1;-1,0,1;-1,0,1];

#without Smoothing
bx = my_filter(r,x);
by = my_filter(r,y);

m = abs(bx) + abs(by);

subplot(2,2,1);
imshow(m);
title("Gradient Magnitude without Smoothing(Prewitt)");

alpha = atan(by./bx);

subplot(2,2,2);
imshow(alpha);
title("Angle without Smoothing(Prewitt)");

#with Smoothing
r1 = my_filter(r,ones(5)*(1/25));
bx = my_filter(r1,x);
by = my_filter(r1,y);

```

```
m = abs(bx) + abs(by);

subplot(2,2,3);
imshow(m);
title("Gradient Magnitude with Smoothing(Prewitt)");

alpha = atan(by./bx);
subplot(2,2,4);
imshow(alpha);
title("Angle with Smoothing(Prewitt)");

figure;

#Sobel Filter
x = [-1,-2,-1;0,0,0;1,2,1];
y = [-1,0,1;-2,0,2;-1,0,1];

#without Smoothing
bx = my_filter(r,x);
by = my_filter(r,y);

m = abs(bx) + abs(by);

subplot(2,2,1);
imshow(m);
title("Gradient Magnitude without Smoothing(Sobel)");

alpha = atan(by./bx);

subplot(2,2,2);
imshow(alpha);
title("Angle without Smoothing(Sobel)");

#with Smoothing
r2 = my_filter(r,ones(5)*(1/25));
bx = my_filter(r2,x);
by = my_filter(r2,y);

m = abs(bx) + abs(by);

subplot(2,2,3);
imshow(m);
title("Gradient Magnitude with Smoothing(Sobel)");

alpha = atan(by./bx);

subplot(2,2,4);
imshow(alpha);
title("Angle with Smoothing(Sobel)");
```

- Applying smoothing before edge detection,final result being dominated mostly by the principal edges.

Output :

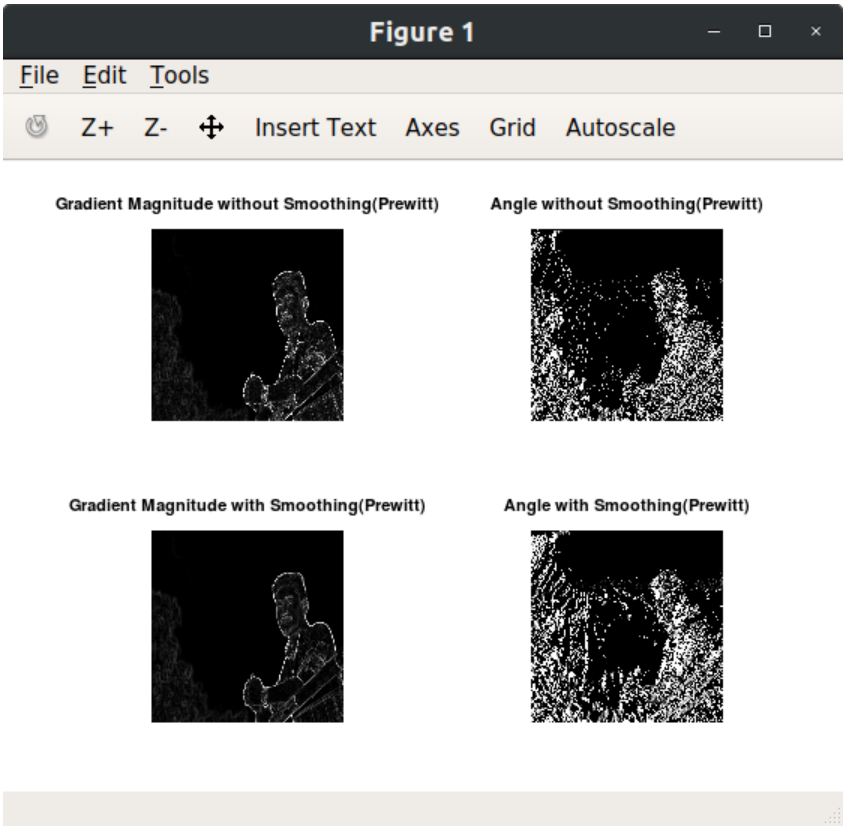


Figure 1
File Edit Tools
Z+ Z- Insert Text Axes Grid Autoscale
Gradient Magnitude without Smoothing(Prewitt)
Angle without Smoothing(Prewitt)
Gradient Magnitude with Smoothing(Prewitt)
Angle with Smoothing(Prewitt)

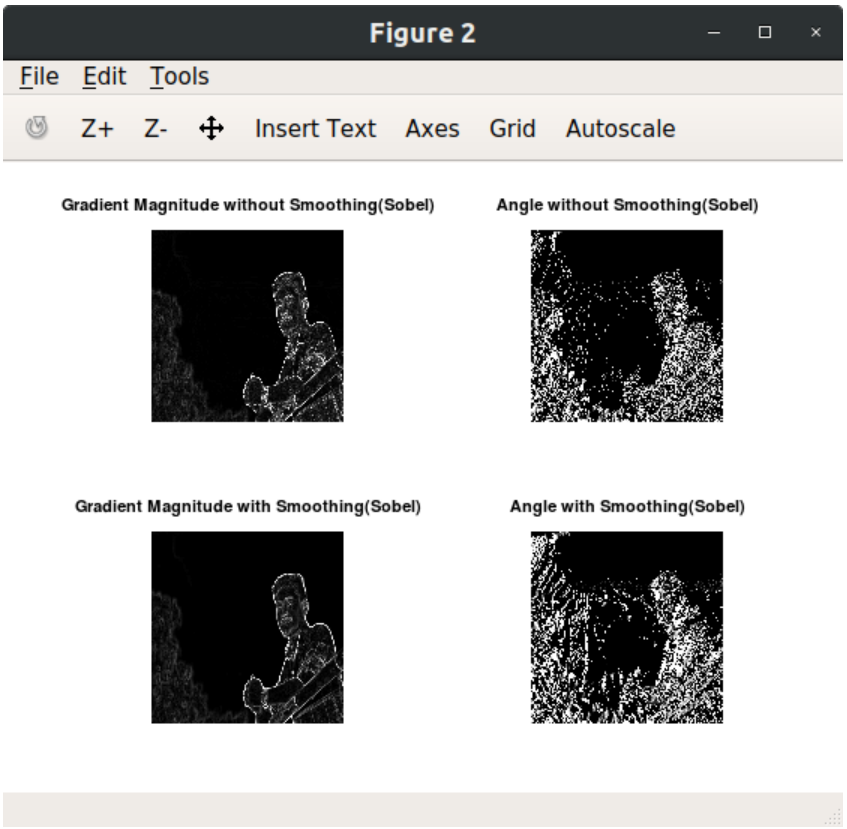


Figure 2
File Edit Tools
Z+ Z- Insert Text Axes Grid Autoscale
Gradient Magnitude without Smoothing(Sobel)
Angle without Smoothing(Sobel)
Gradient Magnitude with Smoothing(Sobel)
Angle with Smoothing(Sobel)