Image Processing | Lab 10 - 11
Morphological Operations and Image Segmentation

strel (shape, parameters) :

Create a strel (structuring element) object for morphology operations.

The structuring element can have any type of shape as specified by shape, each one with its parameters.

imdilate(im, SE) | imdilate(im, SE, shape) & imerode(im, SE) | imerode(im, SE, shape):

Perform morphological dilation.

The image im must be a numeric matrix with any number of dimensions. The dilation is performed with the structuring element se which can be a:

- strel object;
- array of strel objects as returned by @strel/getsequence;
- matrix of 0's and 1's.

To perform a non-flat dilation, SE must be a strel object.

imopen (img , SE) :

Perform morphological opening.

The matrix img must be numeric while SE can be a:

- strel object;
- array of strel objects as returned by '@strel/getsequence';
- matrix of 0's and 1's.

The opening corresponds to an erosion followed by a dilation of img, using the same SE, i.e., it is equivalent to:**imdilate (imerode (img, se), se);**

imclose (img, SE) :

Perform morphological closing.

The matrix img must be numeric while SE can be a:

- strel object;
- array of strel objects as returned by '@strel/getsequence';
- matrix of 0's and 1's.

The closing corresponds to a dilation followed by an erosion of img, using the same SE, i.e., it is equivalent to:**imerode (imdilate (img, se), se);**

Function Used in Lab :

my_filter :

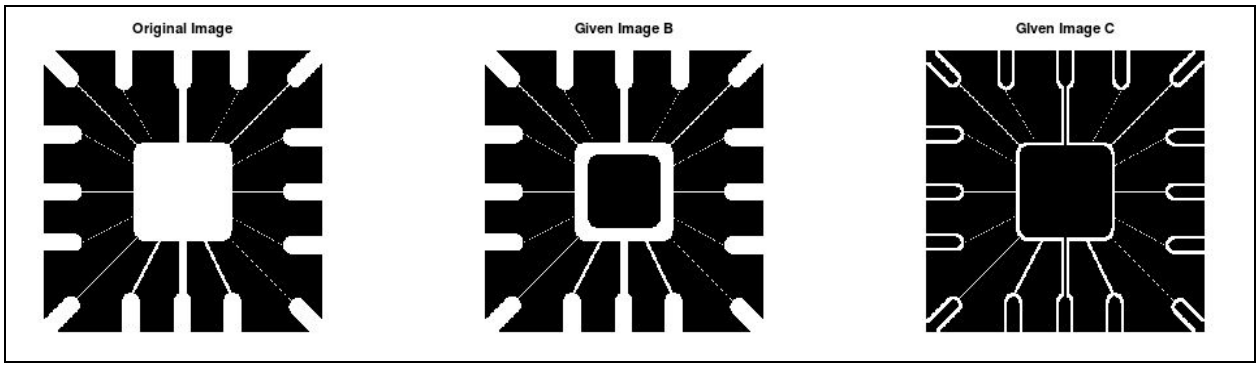
```
function s = my_filter(r,kernel)
  [M,N] = size(r);
  [m,n] = size(kernel);
  a = (m-1)/2;
  b = (n-1)/2;
  new_imsize = zeros(M + 2*a,N + 2*b);
  new_imsize(1+a:M+a,1+b:N+b) = r;
  sub_image = ((1/(m*n)).*ones(m,n));
  s = zeros(size(r));
  for i = 1+a:M+a,
    for j = 1+b:N+b,
      k = new_imsize(i-a:i+a,j-b:j+b);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

Task 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("Ch9-Image/Fig0905(a)(wirebond-mask).tif");
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 :

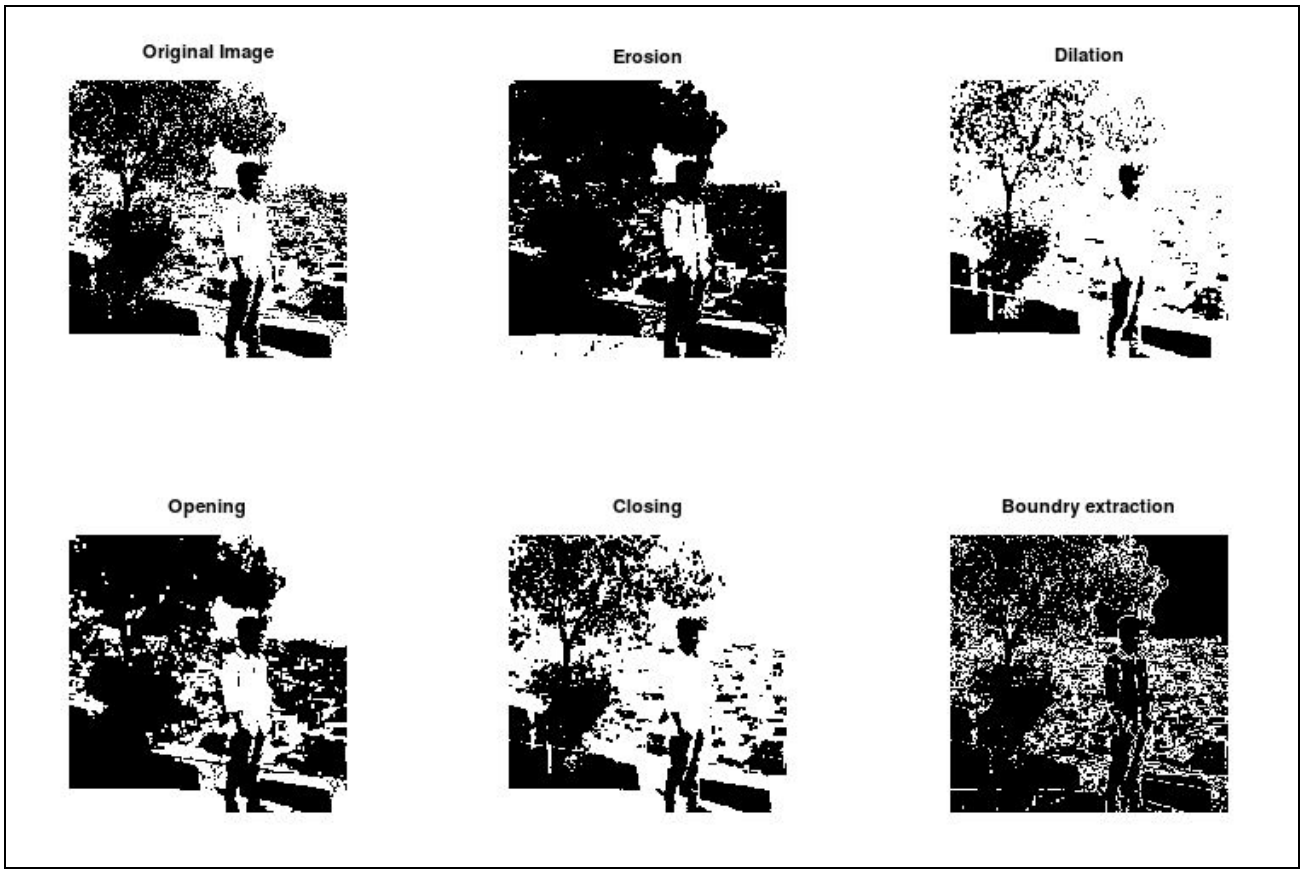


Task 2 : Consider your black and white photo, perform erosion, dilation,opening, closing and boundary extraction. Explain the impact of size of the structuring element on these operations.

Code :

```
pkg load image;

r=im2bw(imread("my_img.jpg"));
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 :



Impact of size of structuring element :

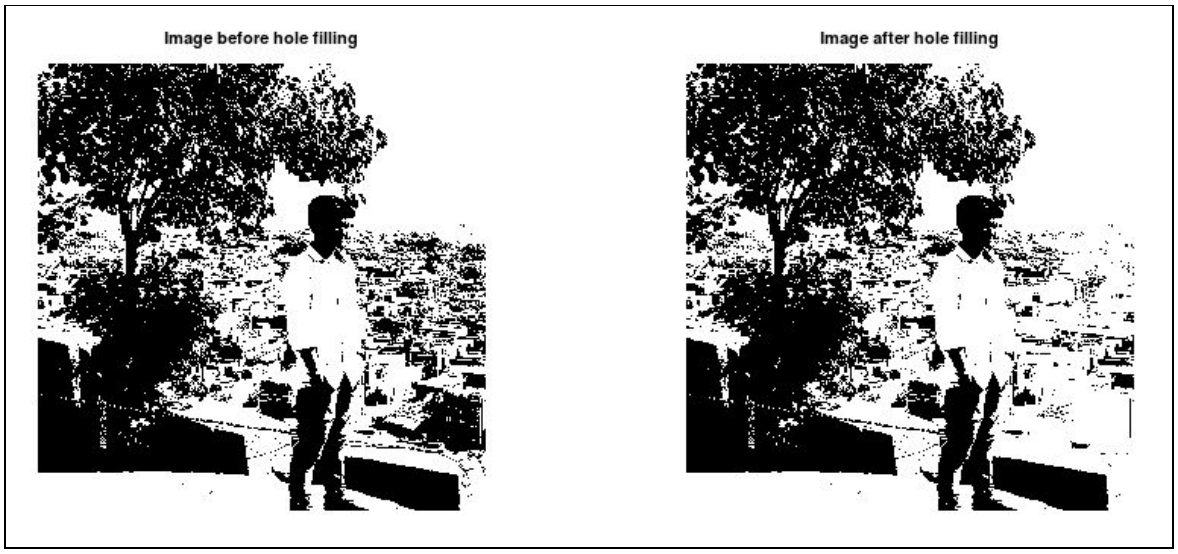
- ★ Increasing size of SE on binary image , more erosion, dilation, opening, closing operation effects on image.

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

Code :

```
pkg load image;
r = imread("my_gray_img.jpg");
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(750,830) = 1;
x0(430,880) = 1;
x0(360,785) = 1;
x0(360,785) = 1;
x0(540,885) = 1;
x0(470,755) = 1;
x0(455,855) = 1;
x0(620,800) = 1;
x0(530,720) = 1;
x0(610,430) = 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 :



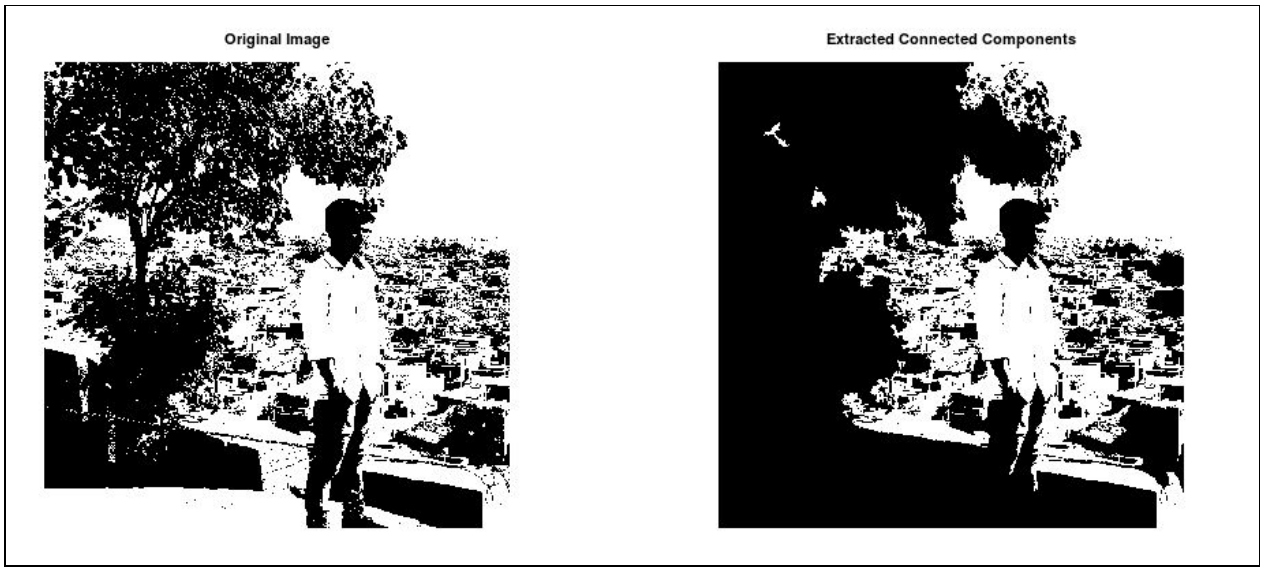
Task 4 : Find a connected component on your black and white photo using morphological operations.

Code :

```
pkg load image;
r = imread("my_gray_img.jpg");
r = im2bw(r);
subplot(1,2,1);
imshow(r);
title("Original Image");

[m,n] = size(r);
struct_ele = strel('diamond',1);
x0 = zeros(m,n);
x0(460,585) = 1;
x0(265,190) = 1;
x0(125,110) = 1;
dilate_image = imdilate(x0,struct_ele);
x1 = dilate_image.*r;

while(!isequal(x0,x1))
  x0=x1;
  dilate_image = imdilate(x0,struct_ele);
```

```
  x1 = dilate_image.*r;
endwhile

subplot(1,2,2);
imshow(x1);
title("Extracted Connected Components");
```

Output :

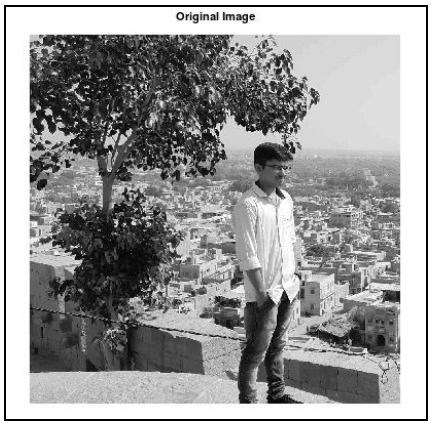


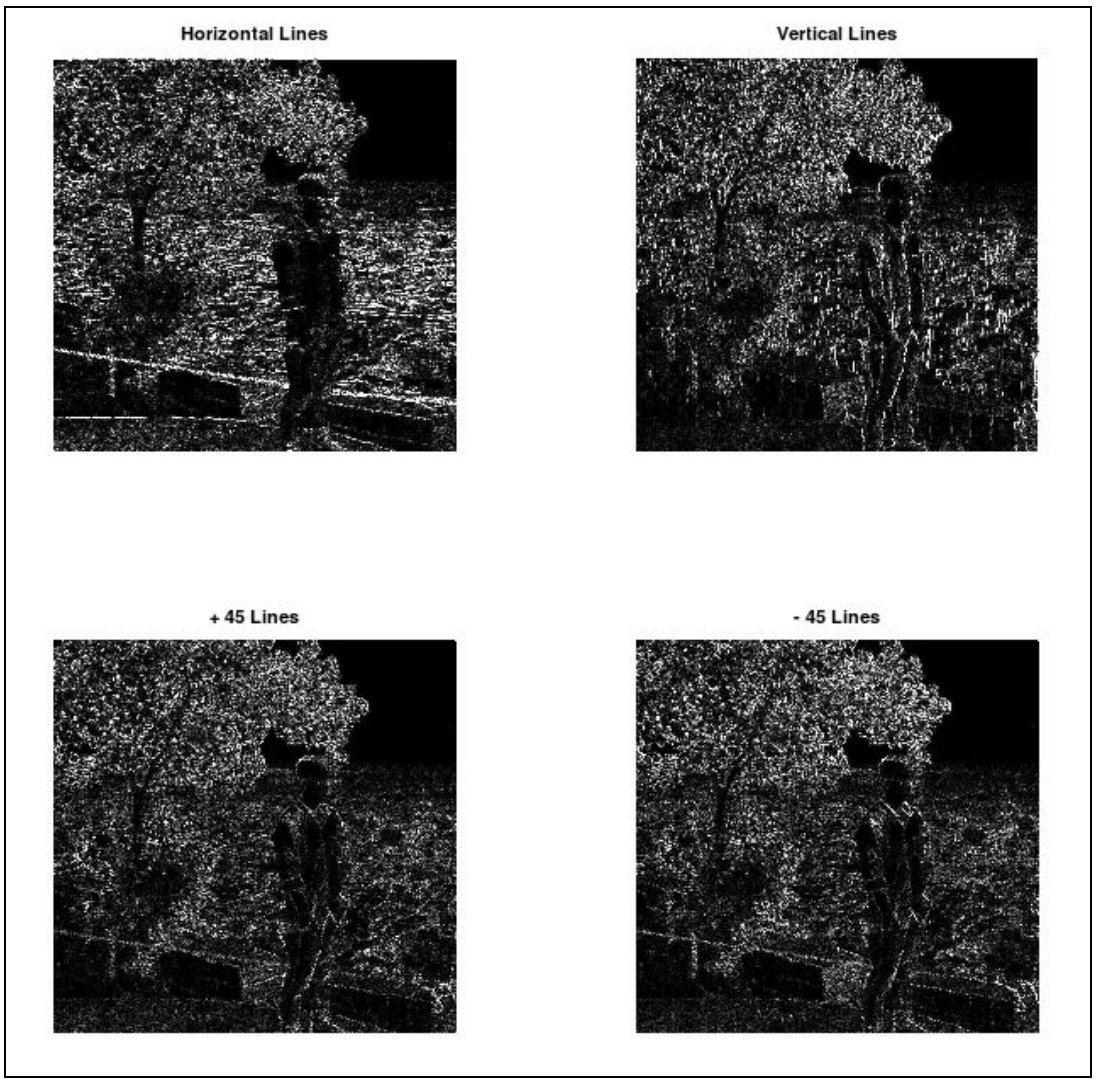
Task 5 : Detect horizontal, vertical, diagonal(+45 degree and -45 degree) lines from your grayscale photo.

Code :

```
#load image
pkg load image;
r = imread("my_gray_img.jpg");
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 :


Original Image


Horizontal Lines
Vertical Lines
+ 45 Lines
- 45 Lines

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

Code :

```
pkg load image;
r = imread("my_gray_scale.jpg");
r = imresize(r,[1024,1024]);
imshow(r);
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)");
```

Output :

For Prewitt mask :

Gradient Magnitude without Smoothing(Prewitt)

Angle without Smoothing(Prewitt)

Gradient Magnitude with Smoothing(Prewitt)

Angle with Smoothing(Prewitt)

For Sobel mask:

Gradient Magnitude without Smoothing(Sobel)

Angle without Smoothing(Sobel)

Gradient Magnitude with Smoothing(Sobel)

Angle with Smoothing(Sobel)

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