HW3

Houlin Li

June 28, 2023

Exercise 1

The gray-level co-occurrence matrix are as follow:

d=(0,2)	0	1	2
0	9	0	8
1	8	12	0
2	0	8	3

(a) d=(0,2)

d=(0,2)	0	1	2
0	5	8	4
1	9	5	7
2	4	6	0

(b) d=(2,0)

d=(0,2)	0	1	2
0	6	4	3
1	7	6	3
2	1	4	3

(c) d=(2,2)

Exercise 2

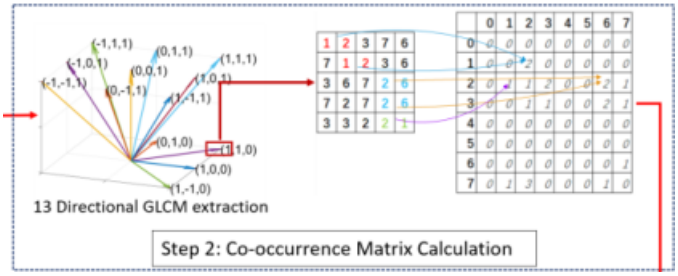
- **Gray Level Image Conversion**

  First, perform gray level scaling on the original CT image pixel values (or Hounsfield units) to an appropriate value range.

  In this study, they adapted histogram based adaptive scaling method to scale the CT images. Using the histogram from all the polyp's regions of inter-est (ROIs) in the CT image database, we generate the gray level bins such that each bin contains roughly the same numberof voxels.

- **3D-GLCM Generation**

  For the 3D volumetric image data, the direction is deter-mined by the three variables of the basis vectors in the 3D space. Following the philosophy of the 2D GLCM used in the classical Haralick model [38], we only sampled along the direction linking the center voxel with its nearest 26 neighbors considering up to second nearest neighbors. As shown in the figure below:



Then they :

1. Resize polyps using an interpolation method
2. Get the 13 × 32 × 32 GLCMs. There is 13 sampling direction. On each direction the size of the matrix is 32 × 32.

herefore, we can get the 3D Gray-Level Co-Occurrence Matrix.

Exercise 3

We know that the opening operation can be used for noise removal. However, it is worth noting that these outgoing noises must be smaller than the kernel functions used for denoising. In the figure of this question, the kernel's size is 13*13, It is smaller than the largest 15 * 15 pixels. So 15 * 15 square will recover after erosion and dilation

Exercise 4

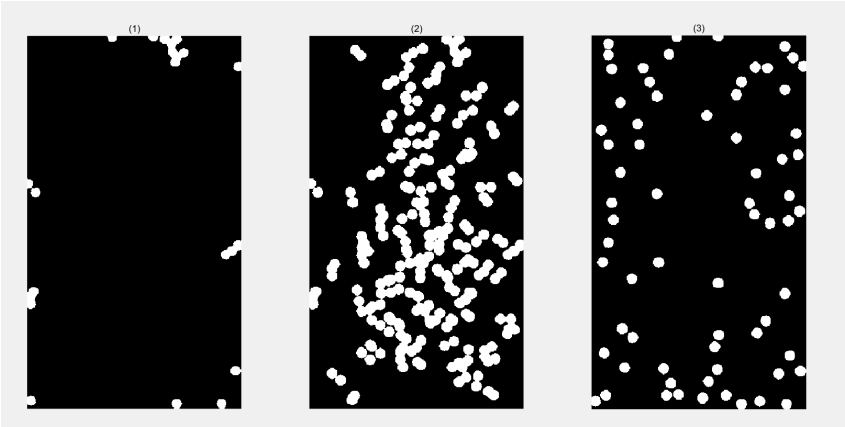
The code is as follows:

```
origin = imread('./HW3.png');
I0 =rgb2gray(origin);
I0 = imbinarize(I0);
%imshow(I0);
%hold on;
I1 = I0;
%让边界上联通域最大的同时，缩小不在边界上的连通域
se = strel('disk', 1);
Erode = imerode(I1, se);
I11 = Erode;
%让所有边界成为一整个连通域
I11(:,1) =1;
I11(:,357) =1;
I11(1,: ) =1;
I11(622,:) =1;
%imshow(f);
L = bwlabel(I11);   % 对连通区域进行标记
% 统计上一步标记图像中的连通域的面积分布
stats = regionprops(L);
Ar = cat(1, stats.Area);
ind = find(Ar ==max(Ar));%找到最大连通区域的标号
I11(find(L~=ind))=0;%将其区域置为0
%填充回原来图像的数值
I11(:,1) =I1(:,1);
I11(:,357) =I1(:,357);
I11(1,: ) =I1(1,:);
I11(622,:) =I1(622,:);
Output1 = I11;
```

```
%imshow(I11);
I2=I0 ;
Output2=bwareaopen( I2 ,300) ;
% 从原图中去掉（去掉单独的点）的 图 像
Output3=I0 - Output2 ;
subplot (1 ,3 ,1) ,imshow(Output1) , title("(1)") ;
subplot (1 ,3 ,2) ,imshow(Output2) , title("(2)");
subplot (1 ,3 ,3) ,imshow(Output3) , title("(3)");
```

Figure 1: Code

the results are below:



- The first figure corresponds to question 1. My idea is that after all the points on the boundary are connected, a maximum connected domain is formed, so that the connected domain can be found and retained
- According to the MATLAB programming, the connected domain composed of multiple points occupies more than 300 pixels
- The connected component of the original graph minus the connected component of multiple points in question 2 is the connected component of a single point