**Image Processing EEE412**

**Lab 4：Morphological Operations**

XUN XU 1926930

2019/12/11

# 1.Task1

## 1.(1)

There are several methods to extract the image boundary. For example, the edge function can be used to extract the boundary if the appropriate operator is selected. Here, the morphological method is used. The image is first eroded and then subtracted from the original image to obtain the boundary.

Outcome：

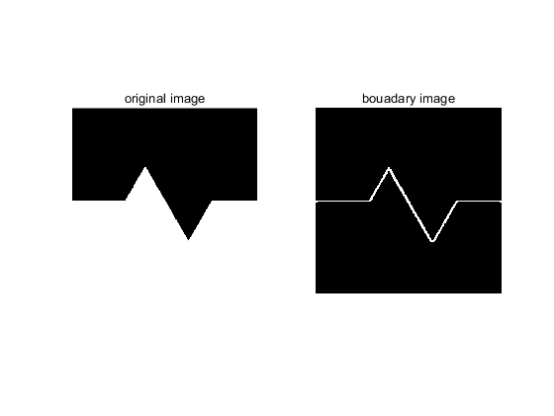


Figure 1：The original image and the extracted boundary image

Code:

f=imread('sawtooth.bmp');  
  
figure;  
subplot(1,2,1);  
imshow(f);  
title('original image');  
  
f=im2bw(f); %Image binarization  
  
se=strel('square',10);%Select 3 by 3 square structure elements  
Ie=imerode(f,se);%Erode the original image  
boundary=f-Ie;%Original image minus corrosion results  
subplot(1,2,2);  
imshow(boundary);  
title('bouadary image');

## 1.(2)

Do the operations of erosion, dilation, opening, and closing. Please use the function of strel to create the structuring element with the shape of disk (You can set your preferred radius). Show the results after each operations and calculate the number of foreground pixels. Write your comments on comparing the results of erosion and opening.

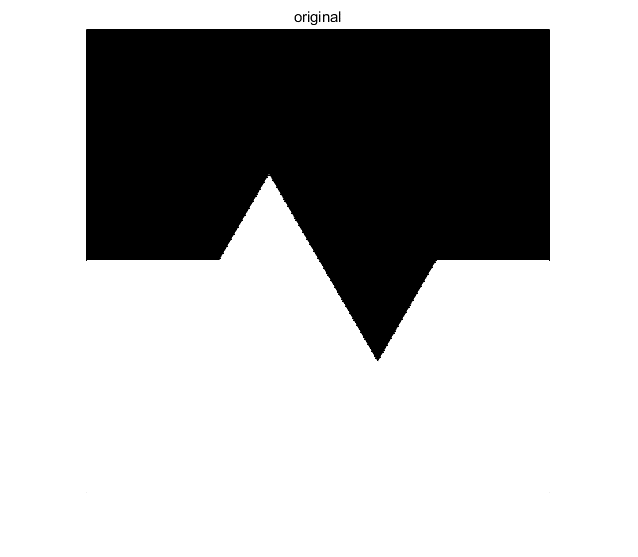


Figure 2：The original image

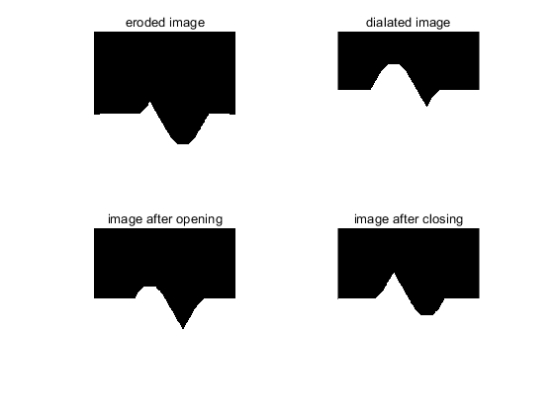


Figure 3： image after four different operations

Result:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Image | Erode | Dilate | Open | Close |
| Number of foreground pixels | 79918 | 132725 | 104822 | 107931 |

Table 1: Number of foreground pixels

Results analysis:

In theory, erosion is the process of eliminating boundary points and causing them to shrink inward, which can be used to eliminate small and meaningless objects. The process of erosion followed by dilation is called open operation, which can be used to eliminate small objects, separate objects at fine points, smooth the boundaries of larger objects without significantly changing the foreground area. Comparing the results of erosion and opening, after open operation, the foreground area of the image is larger and more information of the original image is retained.

Code:

image=imread('sawtooth.bmp');  
  
figure;  
imshow(image);  
title('original');  
  
image=im2bw(image);%Image binarization  
  
se=strel('disk',40);%use the disk strel  
  
erode\_im=imerode(image,se);  
dilate\_im=imdilate(image,se);  
open\_im=imopen(image,se);  
close\_im=imclose(image,se);  
  
figure;  
subplot(2,2,1);  
imshow(erode\_im);  
title('eroded image');  
subplot(2,2,2);  
imshow(dilate\_im);  
title('dialated image');  
subplot(2,2,3);  
imshow(open\_im);  
title('image after opening');  
subplot(2,2,4);  
imshow(close\_im);  
title('image after closing');  
%calculate the pixel numbers  
num\_erode=length(find(erode\_im==1));  
disp(num\_erode);  
num\_dilate=length(find(dilate\_im==1));  
disp(num\_dilate);  
num\_open=length(find(open\_im==1));  
disp(num\_open);  
num\_close=length(find(close\_im==1));  
disp(num\_close);

## 1.(3)

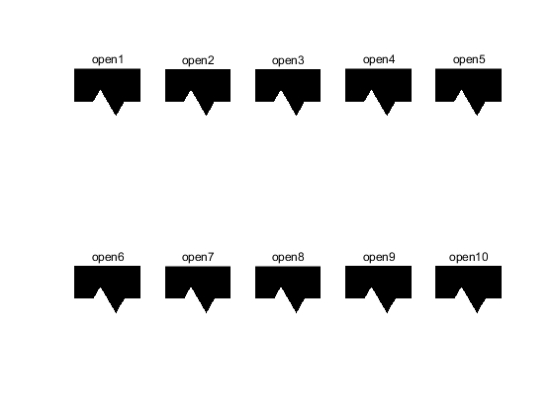


Figure 4: results after Continuing to perform operations on the previous results

Results analysis:

According to the figure 4, the results of multiple opening operations are consistent with the results of one opening operation. Because the same structuring element is used, from the second opening operation, substantially altering the image does not alter the image. The idempotent property shows that the results of successive opening or closing operations are the same as result of a single open or closed operation.

Code:

%Task1\_3  
image=imread('sawtooth.bmp');  
  
figure;  
imshow(image);  
title('original');  
  
image=im2bw(image);%Image binarization  
  
se=strel('disk',5);  
figure;  
in=image;  
%Continue to perform operations on the previous results  
for i=1:1:10  
out=imopen(in,se);  
subplot(2,5,i);  
imshow(out);  
title(['open' num2str(i)]);  
in=out;  
end

# 2.Task2:

## 2.a

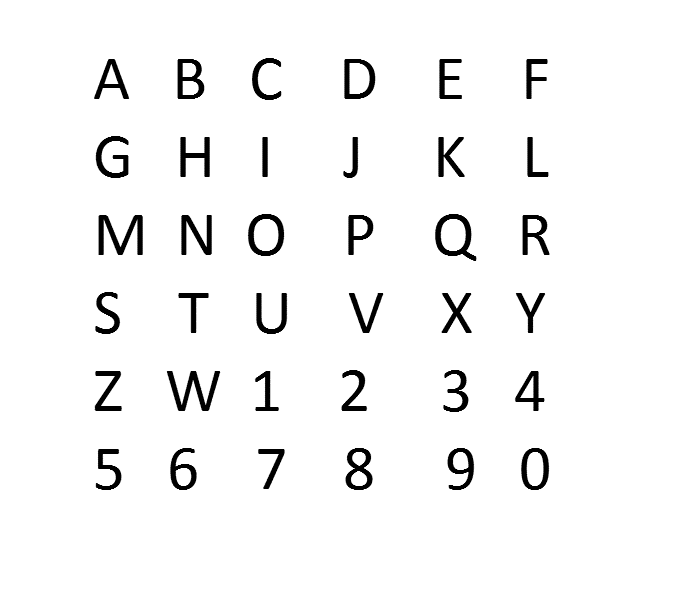


Figure 5: gray image of alphanumeric templates

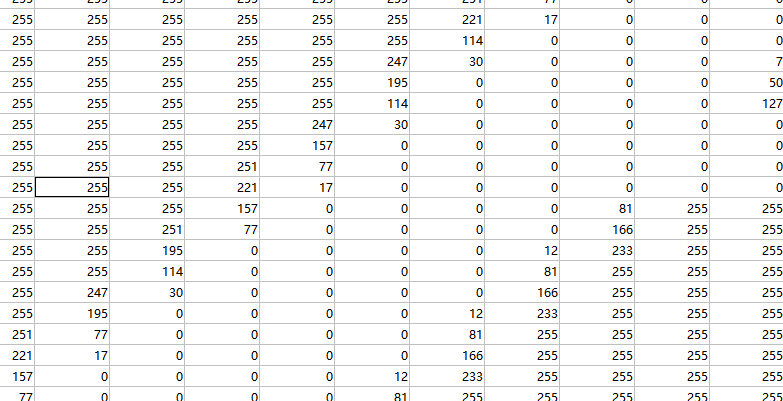


Figure 6: values of grayscale image of alphanumeric templates(zoom in)

For alphanumeric templates, after converting the RGB image to grayscale, we can see its grayscale value. We need to separate the white letters from the black background, which have values of 1 and 0, so the threshold can be set to any number in (0,1) during binarization.



Figure 7: gray image of carplate

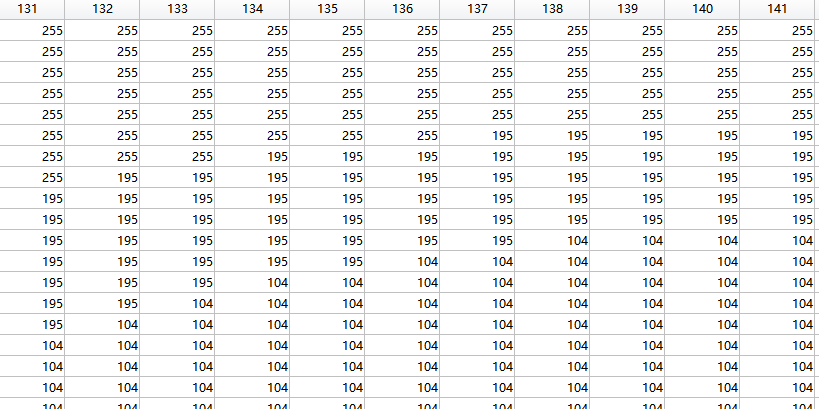


Figure 8: values of grayscale image of carplate(zoom in)

For carplate, after converting the RGB image to grayscale, we can see its grayscale value. We need to separate the white letters from the black background, which have values of 1 and 0, so the threshold can be set to any number in (0,1) during binarization.

## 2.b

Result:

figure4: Quantized image

Results analysis:

As can be seen from the results, the larger the QP value is, the larger the PSNR value is. As the value of QP increases, the value of S keeps decreasing. In the program, the smaller S is, the larger the result is, and the more weight obtained after round is retained. If S is large, the result after round is close to 0. So the result is easy to be ignored. The more weight is retained, the closer the image is to the original

# 3.Task3