EECS 332 Introduction to Computer Vision

Machine Problem 2: Morphological Operators

Name: Runlong Lin ID:3075255

1. Introduction

Boundary detection is an important issue in computer vision's study. However, it's not easy to detect the boundaries if the image has noise, including pepper noise, gaps and breaks. Therefore we use tools to modify the images and get rid of noise. The most common binary image operations are called morphological operators, since they change the shape of the binary objects. Morphological operator often takes the original image and structuring element as input and combine them using different operators. In Machine Problem 2, we build up 2 basic operators using Matlab: dilation and erosion. Based on the combination of dilation and erosion, we can implement more operators like opening and closing. After morphological operations, we can eliminate the noise if image and obtain clear boundary.

2. Algorithms description

We implement 5 morphological operators:

```
function[Dilation_A]=dilation(A,SE)
function[Erosion_A]=erosion(A,SE)
function [opening_A]=opening(A,SE)
function[closing_A]=closing(A,SE)
function[boundary_A]=boundary(A,SE)
```

where A is the input image and SE is structing element.

Among these functions, dilation and erosion are the most basic the critical ones. Another three functions are combinations of dilation and erosion.

● dilation

The purpose of dilation is to grow or expand the image region. In this way, the gaps and breaks in the original images will be covered by the expanded pixels. The main steps of dilation algorithms are listed below:

Step 1:

(Supposing structuring element is 3x3 size.)

We scan each pixel from left to right, from top to down.

Step 2:

We check the pixels of 3x3 region which take the current scanned pixel as center point. If some of the pixels in this region are equal to one, we set the current pixel as one. (as the code shown)

```
for i =bias:(length_A(1)-length_SE(1))
      for j =bias:(length_A(2)-length_SE(2))
          for u=1:length_SE(1)
              for v=1:length_SE(2)
                  if A(i-bias+u,j-bias+v)==SE(u,v)
                      new(i,j)=1;
                  end
              end
          end
```

```
        end
    end
```

- **erosion**

We use erosion to erode the boundary of the foreground of the image. In this way, we can shrink the image's size. The core steps of erosion are shown below:

Step 1:

(Supposing structuring element is 3x3 size.)

We scan each pixel from left to right, from top to down.

Step 2:

We check the pixels of 3x3 region which take the current scanned pixel as center point. If some of the pixels in this region are not equal to one, we set the current pixel as zero. (as the code shown)

```
for i =bias:(length_A(1)-length_SE(1))
    for j =bias:(length_A(2)-length_SE(2))
        flag=1;
        for u=1:length_SE(1)
            for v=1:length_SE(2)
                if A(i-bias+u,j-bias+v)~=SE(u,v)
                    flag=0;
                end
            end
        end
        new(i,j)=flag;
    end
end
```

- **opening**

Opening can be regarded as a slightly erosion. It tends to remove some of the edge pixels of the foreground. Both opening and closing function can be derived from dilation and erosion operators. The core code of opening is shown below:

```
function [opening_A]=opening(A,SE)
    c=erosion(A,SE);
    d=dilation(c,SE);
    opening_A=d;
end
```

- **closing**

Closing is similar to dilations in some ways as it slightly enlarges the boundary of the foreground. The code of closing is shown below:

```
function[closing_A]=closing(A,SE)
    c=dilation(A,SE);
    d=erosion(c,SE);
    closing_A=d;
end
```

- **boundary**

After morphological operations, we can get rid of most the noise. Therefore, we can obtain the clear boundary. To extract the boundary, we can remove the erosion part from the original

image.

```
function[boundary_A]=boundary(A,SE)
    c=erosion(A,SE);
    boundary_A=A-c;
end
```

3. Result Analysis

We use 3x3 size structuring element for gun.bmg and palm.bmg. We also use 5x5 and 7x7 size SE to explore the effect of different size SE. Here is some of the results:

- **palm.bmg using 3x3 SE**

Fig.1.3x3 dilation

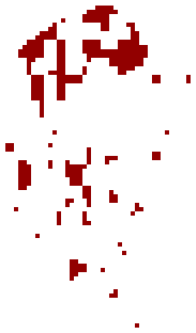

Fig.2. 3x3 erosion

Fig.3. 3x3 opening

Fig.4. 3x3 closing

Fig.5. boundary of dilation img

Fig.6. boundary of closing img

- **gun.bmg using 3x3 SE and size_filter**

Fig.7. 3x3dilation

Fig.8.erosion

Fig.9 3x3 opening

Fig.10 3x3 closing

Fig.11boundary of dilation img

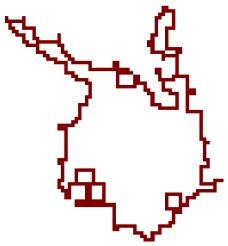

Fig.12boundary of closing img

- **palm.bmg using 5x5 SE**

Fig.13.5x5dilation

Fig.14.5x5erosion

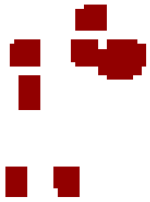

Fig.15. 5x5 opening

Fig.16. 5x5 closing

Fig.17. boundary of dilation img

Fig.18. boundary of closing img

- **palm.bmg using 7x7 SE**

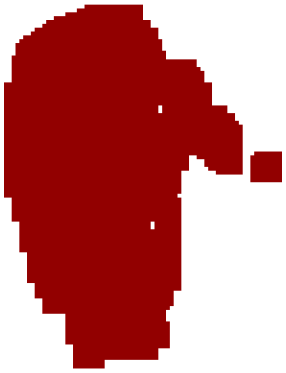

Fig.19. 7x7 dilation

Fig.20. 7x7 erosion

Fig.21. opening

Fig.22. closing

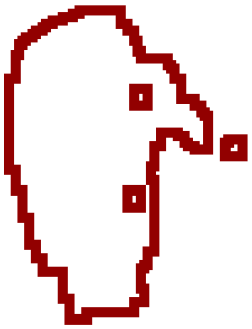

Fig.23. boundary of dilation img

Fig.24. boundary of closing img

Analysis:

For the gun.bmg, we use size_filter function built in MP1 to filter the noise.

For the palm.bmg, we use 3x3,5x5 and 7x7 size structuring element to implement the morphological operations. As the output pictures show, when we use large size SE, we can eliminate more noise inside the object and get a clearer boundary. But at the same time, some details in the object will be ignored if we use large size structuring element.

If we want to obtain clear boundary, we should dilate the original image to get rid of the noise.