

Image processing for bioinformatics

Laboratory Morphological Operators

1 Examples, Matlab functions

1.1 Erosion

| Code | Image |
|--|-------|
| <pre>1 %% Erosion 2 I_1 = rgb2gray(imread('box.png')); 3 I_1 = I_1>50; 4 % Structuring element (SE) 5 SE_1 = strel('disk', 6); % Morphological SE 6 SE_2 = ones(11); % Matrix 7 SE_3 = zeros(9);SE_3(:,5)=1; %SE_3(5,:)=1; 8 % Erosion 9 BW1 = imerode(I_1,SE_1); 10 BW2 = imerode(I_1,SE_2); 11 BW3 = imerode(I_1,SE_3);</pre> | |

Table 1: Erosion

1.2 Dilation

| Code | Image |
|--|-------|
| <pre>1 %% Dilation 2 I_1 = imread('box.png'); 3 I_1 = rgb2gray(imread('box.png')); 4 I_1 = I_1>50; 5 % Structuring element 6 SE_1 = strel('disk', 6); 7 SE_2 = ones(11); 8 SE_3 = zeros(9);SE_3(:,5)=1; %SE_3(5,:)=1; 9 % Dilation 10 BW1 = imdilate(I_1,SE_1); 11 BW2 = imdilate(I_1,SE_2); 12 BW3 = imdilate(I_1,SE_3);</pre> | |

Table 2: Dilation

1.3 Opening

| Code | Image |
|--|-------|
| <pre> 1 %% Opening (erosion of A by B followed by a dilation of the result by B) 2 I_1 = rgb2gray(imread('duck.png')); 3 I_1 = I_1>50; 4 5 % Structuring element 6 SE_1 = strel('disk', 5); 7 SE_2 = eye(9); 8 SE_3 = zeros(9);SE_3(:,5)=1; % SE_3(5,:)=1; 9 10 % Opening 11 BW1 = imopen(I_1,SE_1); 12 BW2 = imopen(I_1,SE_2); 13 BW3 = imopen(I_1,SE_3); </pre> | |

Table 3: Opening

1.4 Closing

| Code | Image |
|---|-------|
| <pre> 1 %% Closing (dilation of A by B followed by an erosion of the result by B) 2 I_1 = rgb2gray(imread('duck.png')); 3 I_1 = I_1>50; 4 5 % Structuring element 6 SE_1 = strel('disk', 5); 7 SE_2 = eye(9); 8 SE_3 = zeros(9);SE_3(:,5)=1; %SE_3(5,:)=1; 9 10 % Closing 11 BW1 = imclose(I_1,SE_1); 12 BW2 = imclose(I_1,SE_2); 13 BW3 = imclose(I_1,SE_3); </pre> | |

Table 4: Closing

1.5 Skeleton

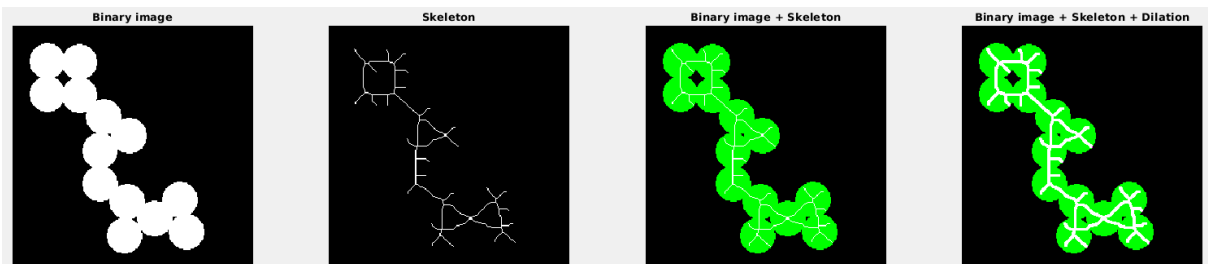
| Code | Image |
|--|---|
| <pre> 1 %% Skeleton 01 2 I_1 = imread('circles.png'); 3 I_1 = I_1(:,:,1); 4 % Skeleton 5 Img_skel = bwmorph(I_1,'skel',Inf); 6 % Structuring element 7 SE_1 = ones(3); 8 % Dilation 9 BW1 = imdilate(Img_skel,SE_1); 10 % Binary image + Skeleton 11 Img_skel_fc = imfuse(I_1, Img_skel); 12 % Binary image + Skeleton + Dilation 13 BW1_fc = imfuse(I_1, BW1); </pre> |  |

Table 5: Skeleton

1.6 Boundary extraction

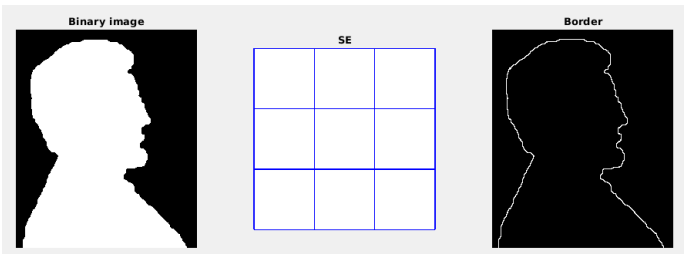
| Code | Image |
|--|--|
| <pre> 1 %% Boundary extraction 2 I_1 = rgb2gray(imread('perfil2.png')); 3 I_1 = I_1>150; 4 5 % Structuring element 6 SE_1 = ones(3); 7 8 % Dilation 9 BW1 = imdilate(I_1,SE_1); 10 11 boundary_01 = I_1 == BW1; </pre> |  |

Table 6: Boundary extraction

2 Assignment

Implement the operations of: Opening and Closing

1. Opening (erosion of A by B followed by a dilation of the result by B)

$$A \circ B = (A \ominus B) \oplus B$$

2. Closing (dilation of A by B followed by an erosion of the result by B)

$$A \bullet B = (A \oplus B) \ominus B$$

- Erosion of A by B:

$$C = A \ominus B = \{z | (B)_z \subseteq A\} = \{z | (B)_z \cap A^c = \emptyset\} = \{z | z + b \in A \text{ for every } b \in B\};$$

all the points z in A such that B is included in A when the origin of B coincides with z , i.e. all the points in A at which B fits A .

$C(u, v) = 1$ if the origin of B is located in (u, v) and all the elements in B with value 1 are 1 in the same region in A .

- Dilation of A by B

$$C = A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} = \{z | [(\hat{B})_z \cap A] \subseteq A\} = \{z | z = a + b, a \in A \text{ and } b \in B\};$$

all the points z in A such that B flipped and translated by z has non-empty intersection with A , i.e. all the points in A at which B hits A .

$C(u, v) = 1$ if the origin of B is located in (u, v) and at least one of the elements in B with value 1 is 1 in the same region in A .

Compare your results with those of the Matlab functions, but don't take into account the pixels near to the margin since the treatment of these may be different.

3 Solutions

Implement the operations of: Opening and Closing

Function myOpening

```
1 function BW = myOpening(Img,SE)
2   BW = myErosion(Img,SE);
3   BW = myDilation(BW,SE);
4 end
```

Function myClosing

```
1 function BW = myClosing(Img,SE)
2   BW = myDilation(Img,SE);
3   BW = myErosion(BW,SE);
4 end
```

Function myErosion

```

1 function BW = myErosion(Img,SE)
2 [m,n] = size(Img);
3 [a,b] = size(SE);
4 sumSE = sum(SE(:));
5 BW = zeros(m,n);
6 ma = fix((a-1)/2);
7 nb = fix((b-1)/2);
8 maCenter = ceil(ma+1);
9 nbCenter = ceil(nb+1);
10 % Cicle for Img
11 for u = 1 + ma : m - ma
12     for v = 1 + nb : n - nb
13         tmp = 0;
14         % Cicle for SE
15         for i = -ma:ma
16             for j = -nb:nb
17                 if Img(u+i,v+j) == 1 && SE(maCenter + i,nbCenter + j) == 1
18                     tmp = tmp + 1;
19             end
20         end
21     end
22     if tmp == sumSE % all 1's in the same position
23         BW(u,v) = 1;
24     end
25 end
26 end
27 end

```

Function myDilation

```

1 function BW = myDilation(Img,SE)
2 [m,n] = size(Img);
3 [a,b] = size(SE);
4 BW = zeros(m,n);
5 ma = fix((a-1)/2);
6 nb = fix((b-1)/2);
7 maCenter = ceil(ma+1);
8 nbCenter = ceil(nb+1);
9 % Cicle for Img
10 for u = 1 + ma : m - ma
11     for v = 1 + nb : n - nb
12         tmp = 0;
13         % Cicle for SE
14         for i = -ma:ma
15             for j = -nb:nb
16                 if Img(u+i,v+j) == 1 && SE(maCenter + i,nbCenter + j) == 1
17                     tmp = tmp + 1;
18             end
19         end
20     end
21     if tmp > 0 % at least one 1 in the same position
22         BW(u,v) = 1;
23     end
24 end
25 end
26 end

```