MATLAB Implementation

Maimoona Khilji

Institute of Management Science

Course Code: Image Processing and Analysis

Muhammad Saad Rashad

17th May, 2022

Task 1

Discuss the following morphological operations using Matlab code, application/usage, and brief working of the operation bwhitmiss bwskel imbothat imtophat bwperim

Morphology is a broad set of image processing operations that process images based on shapes. It basically refers to study of shapes, sizes and structures. Morphological operations apply a structuring element to an input image, creating an output image of the same size.

Morphological Dilation and Erosion

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects, while erosion removes pixels on object boundaries in an image.

The number of pixels added or removed from the objects in an image depends on

- The size
- The shape of the structuring element used to process the image.

Rules for Dilation and Erosion

Dilation:

The value of the output pixel is the *maximum* value of all pixels in the neighborhood. Morphological dilation makes objects more visible and fills in small holes in objects. Lines appear thicker, and filled shapes appear larger.

Erosion:

The value of the output pixel is the *minimum* value of all pixels in the neighborhood. Morphological erosion removes floating pixels and thin lines so that only substantive objects remain. Remaining lines appear thinner and shapes appear smaller.

Operations Based on Dilation and Erosion

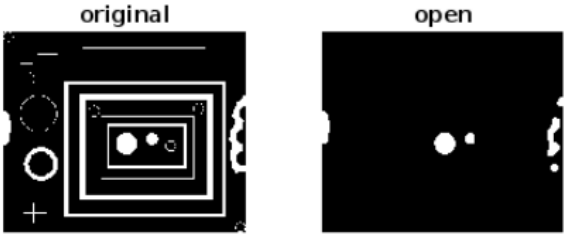
Imopen

Perform morphological opening. The opening operation erodes an image and then dilates the eroded image, using the same structuring element for both operations.

Assignment04.m

```
1   f=imread('blobs.png');
2   subplot(1,2,1);
3   imshow(f);
4   title("original")
5   se= strel('disk',5);
6   g=imopen(f,se);
7   subplot(1,2,2);
8   imshow(g)
9   title("open");
10
```

Figure 1

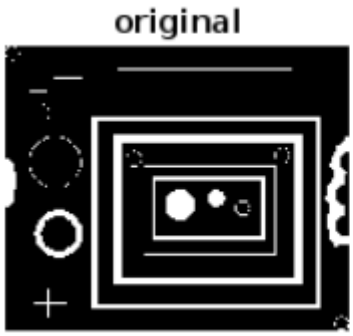


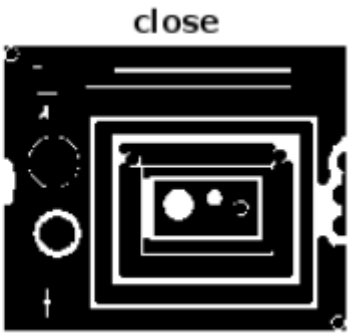
Imclose

Perform morphological closing. The closing operation dilates an image and then erodes the dilated image, using the same structuring element for both operations.

Assignment04.m

```
f=imread('blobs.png');
subplot(1,2,1);
imshow(f);
title("original")
se= strel('disk',5);
g=imclose(f,se);
subplot(1,2,2);
imshow(g)
title("close");
```



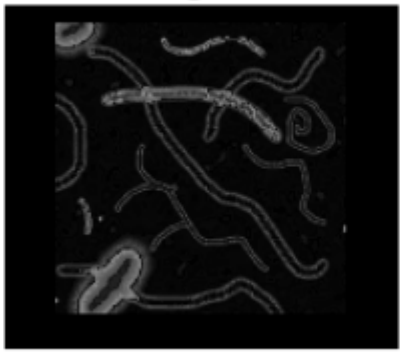


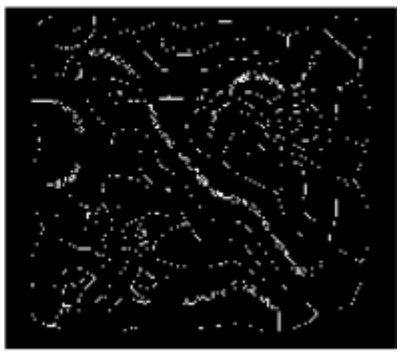
Bwskel

Skeletonize objects in a binary image. The process of skeletonization erodes all objects to centerlines without changing the essential structure of the objects, such as the existence of holes and branches

ent04.m A2.m

```
I=imread('threads.png');

subplot(1,2,1);
imshow(I);
title("original")

Icomplement = im2uint8(imcomplement(I));
BW = imbinarize(Icomplement);

g=im2uint8(bwskel(BW));
subplot(1,2,2);
imshow(labeloverlay(I,g,'Transparency',0))
imshow(g)
title("Bwskel");
```



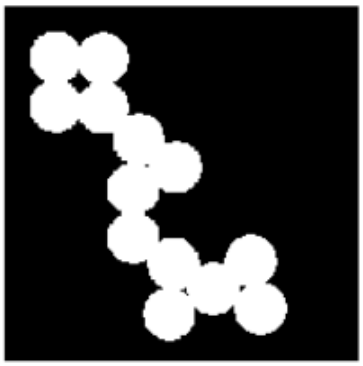


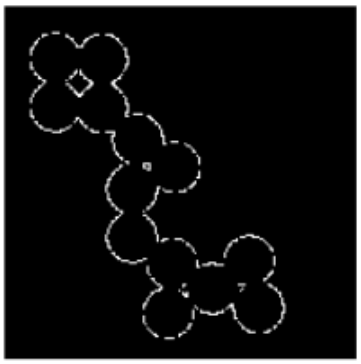
Bwperim

Find perimeter of objects in a binary image. A pixel is part of the perimeter if it is nonzero and it is connected to at least one zero-valued pixel. Therefore, edges of interior holes are considered part of the object perimeter.

Assignment04.m

```
%I=imread('threads.png');
BW = imread('circles.png');
subplot(1,2,1);
imshow(BW);

BW2 = bwperim(BW,8);
subplot(1,2,2);
imshow(BW2);
%imshowpair(BW,BW2,'montage')
```





Bwhitmiss

Perform binary hit-miss transform. The hit-miss transform preserves pixels in a binary image whose neighborhoods match the shape of one structuring element and do not match the shape of a second disjoint structuring element. The hit-miss transforms can be used to detect patterns in an image.

Assignment04.m | A2.m

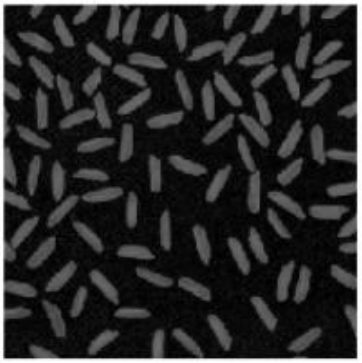
```
bw = [0 0 0 0 0 0
      0 0 1 1 0 0
      0 1 1 1 1 0
      0 1 1 1 1 0
      0 0 1 1 0 0
      0 0 1 0 0 0]
subplot(1,2,1);
imshow(bw);
title("original")

interval = [0 -1 -1
            1  1 -1
            0  1  0];
bw2 = bwhitmiss(bw,interval)


subplot(1,2,2);
imshow(bw2);
title("bwhitmiss")
```

Imtophat

Perform a morphological top-hat transform. The top-hat transform opens an image, then subtracts the opened image from the original image. The top-hat transform can be used to enhance contrast in a grayscale image with non-uniform illumination. The transform can also isolate small bright objects in an image.

Assignment04.m × A2.m ×

```
1
2   original = imread('rice.png');
3   subplot(1,2,1);
4   imshow(original)
5
6   se = strel('disk',12);
7
8   tophatFiltered = imtophat(original,se);
9   subplot(1,2,2);
10  imshow(tophatFiltered)
```



Imbothat

Perform a morphological bottom-hat transform. The bottom-hat transform closes an image, then subtracts the original image from the closed image. The bottom-hat transform isolates pixels that are darker than other pixels in their neighborhood. Therefore, the transform can be used to find intensity troughs in a grayscale image.

Assignment04.m × A2.m ×

```
1   I = imread('pout.tif');
2   subplot(1,2,1);
3   imshow(I)
4   se = strel('disk',5);
5
6   J = imsubtract(imadd(I,imtophat(I,se)),imbothat(I,se));
7   subplot(1,2,2);
8   imshow(J)
```