Morphological Operations

Project: Marker-based Localization

Niharika Jayanthi, Dheeraj Kamath
Mentor: Sanam Shakya

Goal

The objectives of this documentation are - + To explain what are morphological functions. + Elucidate about erosion, dilation, opening and closing. + Understand through examples and provide practice through exercises.

Theory

Working principle

Introduction

Morphological operations in image processing are a set of operations that are applied on images to process them. These operations are commonly used on binary images (On grayscale images, these operators can reduce noise or brighten the image). Another input that is required to perform these operations is structuring elements. Structuring elements are used to probe a binary image. These probes are themselves binary images shaped in rectangles, ellipses, diamonds, crosses or other (possibly arbitrary) shapes.

Structuring Elements

Structuring elements are small binary images that are represented by a special matrix format. Following are matrix formats of some shapes-

- Rectangle

```
[1 , 1 , 1 , 1 , 1,
 1 , 1 , 1 , 1 , 1,
 1 , 1 , 1 , 1 , 1,
 1 , 1 , 1 , 1 , 1,
 1 , 1 , 1 , 1 , 1]
```

Figure 1: Rectangle

- Diamond
- Ellipse

  [Ellipse](images/Morphological Operations/Images/Ellipse.PNG
- Cross

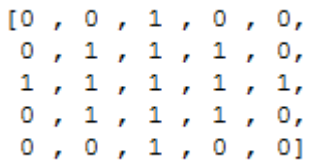


Figure 2: Diamond

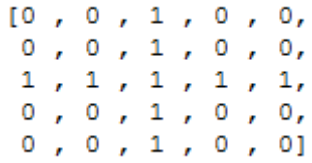


Figure 3: Cross

Morphological operations

As discussed above, morphological operations are a set of techniques used for image processing. Following are two basic morphological operations:

Erosion

Erosion is a basic morphological operation applied on binary (or grayscale) images that erodes away the boundaries of regions of foreground pixels(i.e. white pixels). Thus, any holes within those areas become larger.

Erosion can be mathematically defined as- If X denotes total set of points on the image, S denotes the set of points on the structuring element and Sx denotes translation of T such that its origin is at x, then erosion of X by S is set of all points x such that Sx is a subset of X.

Dilation

Dilation is a basic morphological operation applied on binary (or grayscale) images that increases the size of the foreground pixels(i.e. white pixels). Thus, any holes in the image shrink whereas the boundaries of image expand.

Dilation can be mathematically defined as- If X denoes total set of points in an image, S denotes set of all points on the structuring element and Sx denotes translation of S such that its origin is at x, then dilation of X by S is a set of all point x where the intersection of X and Sx is non-empty.

Opening

Opening operator is derived from erosion and dilation operations. Simply put, opening is erosion followed by dilation using the same structuring element. This operation is useful for removing noise from an image. It is also the dual of closing operation.

Closing

Closing operator is also derived from erosion and dilation operations. It can be defined as dilation of an image followed by erosion. This operator is highly useful for closing small black points on the image. It is the dual of opening operation.

Applications

Morphological operations are used in the following fields: - Robotics : It allows for execution through visual feedback, motion control and recognizing and interpreting objects. - Forensics : It helps reduce noise in fingerprint images and consequently enhance them. - Medicine : It can help with detection of tumors, measurement of size and shapes of organs and much more. - Radar : It can be used in target detection and identification systems.

Example images

Consider the following image:

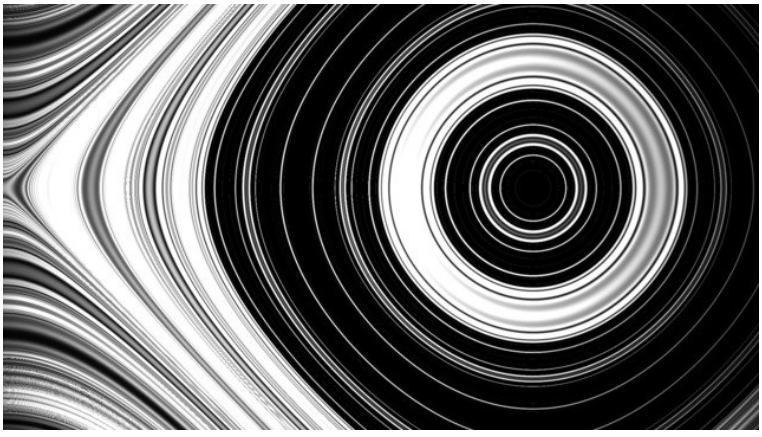
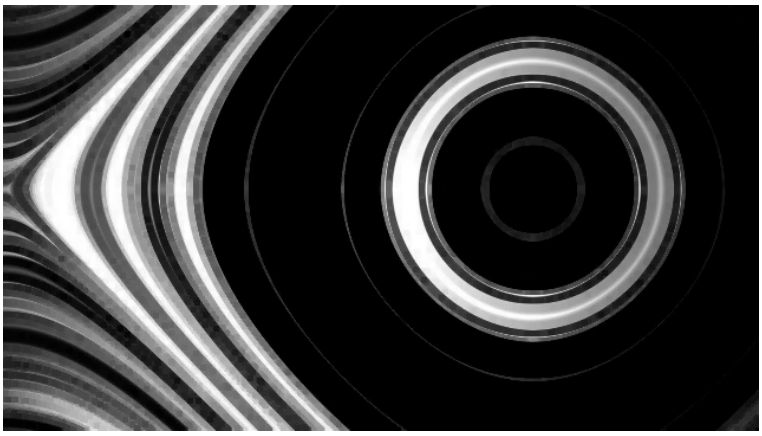


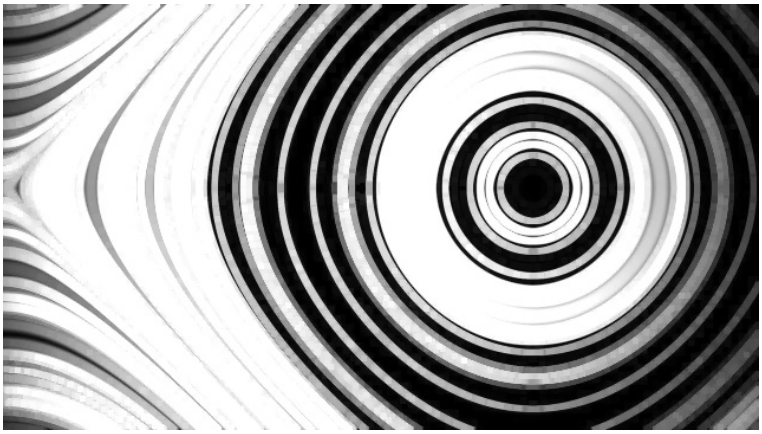
Figure 4: Music

On performing erosion, it looks like this:



Notice how the concentric white boundaries of the circe almost disappear. The black regions (holes) get bigger.

On performing dilation, the same image would look like-



Here, the opposite happens. The white pixels increase whereas the black pixels (holes) decrease.

Now, consider the next sample image:

Figure 5: Sunflower

There are small white dots in the center of the flower whereas black dots are littered around the petals.

When we apply opening operation on the above image, we get the following image:

The small white dots (noise) are removed from the image.

Now let us see the result of performing closing operation on the same:

The noise in the center of the flower is not removed but the small black dots on the foreground(white part) are removed.

Code

Let us now have a look at the available functions for the above morphological operations in python.

```
cv2.erode(src, kernel, iterations)

This function is used for erosion where-
      src :        Image to be eroded
      kernel :     Structuring element used
      iterations : Number of iterations to be applied

cv2.dilate(src, kernel, iterations)

This function is used for dilation where-
      src :        Image to be dilated
      kernel :     Structuring element used
      iterations : Number of iterations to be applied
```

```
cv2.morphologyEx(src, op, kernel)

This function is used for opening and closing where-
      src :    Image to be opened
      op:      Morphological operation to be performed. It's values can be one of the follow
               cv2.MORPH_OPEN-  For opening
               cv2.MORPH_CLOSE- For closing
      kernel : Structuring element to be used
```

These functions are available in cv2 package. So cv2 must be imported in the beginning of the code.

Now, let us see how to write the code.

- All codes involving morphological operations must first import cv2 package to use the above functions and numpy package to create the structural element.

```
import cv2
import numpy
```

- Next, the image has to be read. This is done by cv2.imread(filename, flag) function. Flag=0 returns a grayscale image.

```
img = cv2.imread("example.jpg", 0)
```

- The structuring element for the morphological operation is created using the following code-

```
kernel = np.ones((5,5), np.uint8)
```

- Now we will perform morphological operations on the grayscale image-

Erosion

We use cv2.erode() function for this.

```
erosion = cv2.erode(img, kernel, iterations = 1)
```

Dilation

We use cv2.dilate() function for this.

```
dilation = cv2.dilate(img, kernel, iterations = 1)
```

Opening

We use cv2.morphologyEx() fuction.

```
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
```

Closing

We use cv2.morphologyEx() fuction.

```
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
```

- We can display the modified image to see and compare the changes. We use the cv2.imshow(windowName, img) function for the same. For example, if you want to see the eroded image, this is the code-

```
cv2.imshow("Eroded image", erosion)
```

- Alternatively, we can also save the modified image with cv2.imwrite(name, img) function.

```
cv2.imwrite("Eroded image.jpg", erosion)
```

- Whenever we use cv2.imshow(), we should also include the following code-

```
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Waitkey displays the image for specified milliseconds. 0 means the window will be open infinitely until a keypress occurs.

Resources

- http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html#morphological-ops
- http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm
- http://www.academia.edu/3482702/Morphological_Image_Processing
- 'EGGN 510 - Lecture 10-1 Morphological Processing'https://www.youtube.com/watch?v=gmi4ah7YAi0