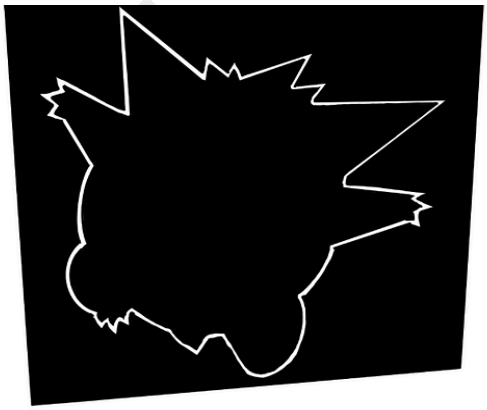
Image Processing for Computer Vision
Session 13

Morphological Transformations



collected

Topics

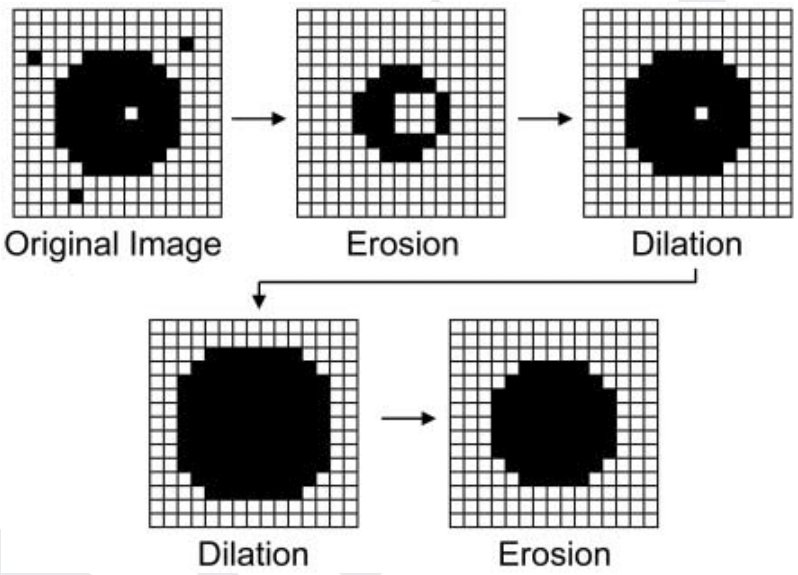
- Morphological Transformation
- Erosion
- Dilation
- Opening
- Closing
- Contour Detection

Morphological Transformations

- a set of image processing operations that process images based on their shapes or structures.
- particularly useful in binary images
- can also be applied to grayscale images

Morphological transformations are some simple operations based on the image shape. It is normally performed on binary images. It needs two inputs, one is our original image, second one is called **structuring element** or **kernel** which decides the nature of operation.

1. Erosion



collected

The basic idea of erosion is just like soil erosion, it erodes away the boundaries of foreground object (Always try to keep foreground in white). So what it does? The kernel slides through the image (as in 2D convolution). A pixel in the original image (either 1 or 0) will be considered 1 only if all the pixels under the kernel are 1, otherwise it is eroded (made to zero).

So what happens?
All the pixels near boundary will be discarded depending upon the size of kernel. So the thickness or size of the foreground object decreases or simply white region decreases in the image. It is useful for removing small white noises (as we have seen in colorspace chapter), detach two connected objects etc.

cv.erode(img, kernel, iterations = 1)

2. Dilation

It is just opposite of erosion. Here, a pixel element is '1' if at least one pixel under the kernel is '1'. So it increases the white region in the image or size of foreground object increases. Normally, in cases like noise removal, erosion is followed by dilation. Because, erosion removes white noises, but it also shrinks our object. So we dilate it. Since noise is gone, they won't come back, but our object area increases. It is also useful in joining broken parts of an object.

cv.dilate(img, kernel, iterations = 1)

3. Opening

Opening is just another name of **erosion followed by dilation**. It is useful in removing noise.

cv.morphologyEx(img, cv.MORPH_OPEN, kernel)

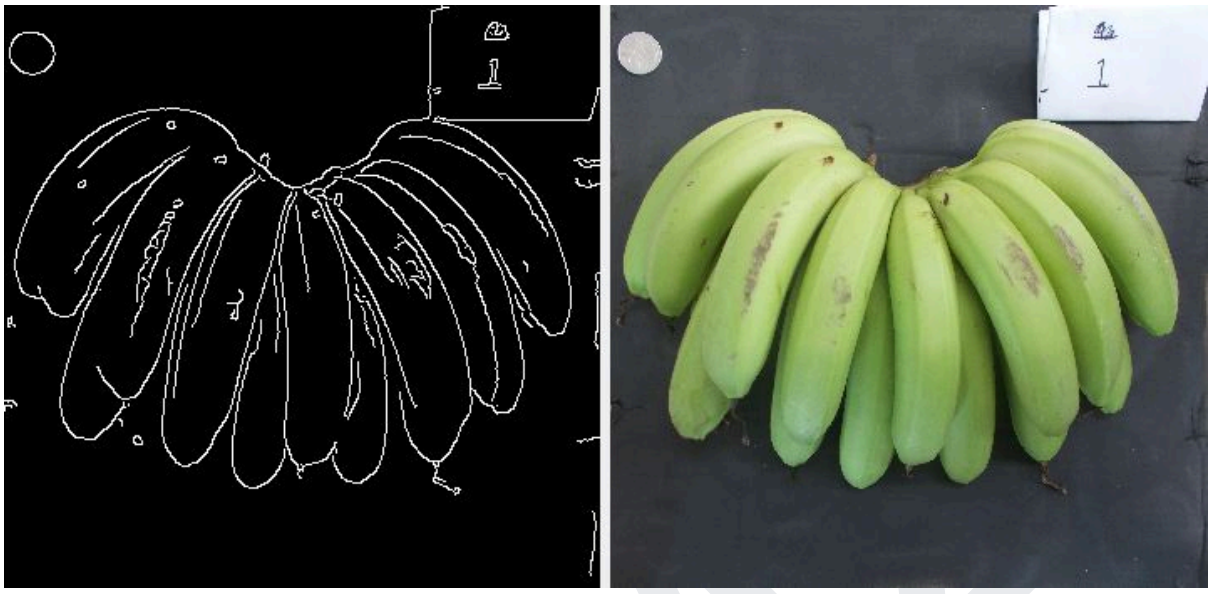
4. Closing

Closing is reverse of Opening, **Dilation followed by Erosion**. It is useful in closing small holes inside the foreground objects, or small black points on the object.

cv.morphologyEx(img, cv.MORPH_CLOSE, kernel)

Learn more: Morphological Gradient, Top Hat, Black Hat …

Contour



Stack overflow

What are contours?

- a continuous curve that outlines the shape of an object in an image.
- often derived from the edges detected in the image.

Applications:

- shape analysis
- object detection and recognition.

Steps:

1. Convert to grayscale and apply other necessary preprocessing
2. Apply thresholding method or edge detection method
3. Find contours
4. Draw contours for visualization (optional)

syntax:
contours, hierarchy = cv2.findContours(image, mode, method, offset=None)
cv2.drawContours(target_img, contours, contourIdx, color, thickness)

offset: *Optional offset by which every contour point is shifted*
contourIdx: *index of the desired contours, if -1 then it will use all contour points*

Mode: Contour retrieval modes
cv.RETR_EXTERNAL
cv.RETR_LIST

cv.RETR_TREE

Learn more about the modes here.

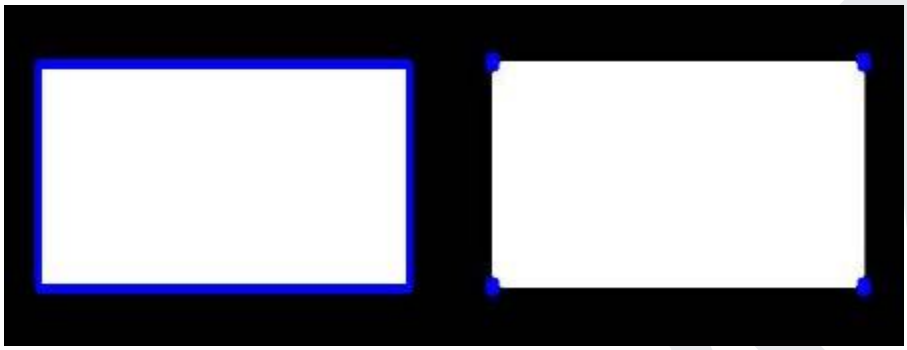
Method: Contour approximation method

cv.CHAIN_APPROX_NONE

cv.CHAIN_APPROX_SIMPLE

Lear more about the approximation methods here.

Contour Approximation Method



OpenCV

cv.CHAIN_APPROX_NONE vs cv.CHAIN_APPROX_SIMPLE

Learn about different Contour Features: Contour Area, Contour Approximation, convex hull, bounding rectangle, line fitting …