Computer Vision Module – Session 8

Computer Vision

Dr. Sunita Dhavale, DIAT

Online Training & Certification Course on Artificial Intelligence & Machine Learning

Defence Institute of Advanced Technology (DU), Pune.

Computer Vision

Dr Sunita Dhavale

Boundary tracking procedures, active contours

Online Training & Certification Course on AI & ML
Defence Institute of Advanced Technology (DU), Pune.

Outline of Presentation

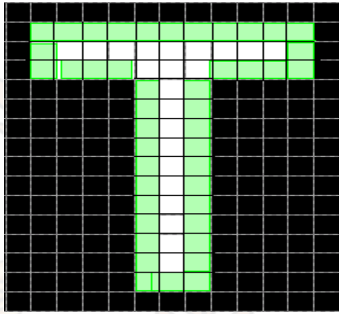
- **Region and Boundaries**
- **Boundary tracking procedures**
- **active contours**

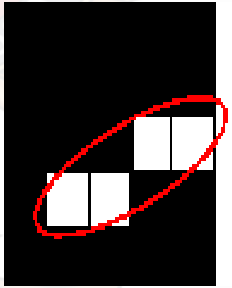
Region

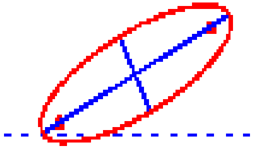
- A subset R of pixels in an image is called a Region of the image if R is a connected set.
- The boundary of the region R is the set of pixels in the region that have one or more neighbors that are not in R.

Region Properties

- Area
- Perimeter
- Ellipse-MajorAxisLength, MinorAxisLength
- Compactness
- Elongation
- Eccentricity
- Circularity
- Convexity
- Aspect ratio
- Curl
- Convex hull
- Solidity
- Shape variances
- Rectangularity
- Bounding box
- direction
- Orientation







Region Properties

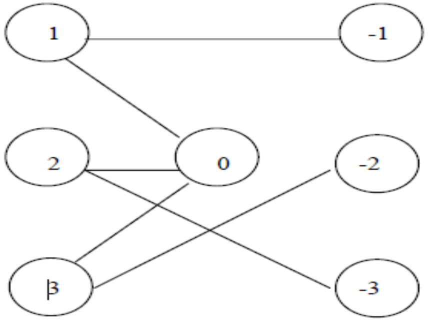
- ConvexHull-Smallest convex polygon that can contain the region, returned as a *p*-by-2 matrix. Each row of the matrix contains the *x*- and *y*-coordinates of one vertex of the polygon.
- ConvexImage-Image that specifies the convex hull, with all pixels within the hull filled in (set to on), returned as a binary image (logical). The image is the size of the bounding box of the region.
- ConvexArea-Number of pixels in ConvexImage.

Region Properties

- Some more measures are: circularity, mean radial distance, standard deviation of radial distance.
- Bounding box-rectangle that encloses shape and touches extremal points can be used.
- Region adjacency graph- each node represents a region of image and edge connects 2 nodes if 2 regions are adjacent.

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	2	2	0
0	1	-1	-1	-1	1	0	2	2	0
0	1	1	1	1	1	0	2	2	0
0	0	0	0	0	0	0	2	2	0
0	3	3	3	0	2	2	2	2	0
0	3	-2	3	0	2	-3	-3	2	0
0	3	-2	3	0	2	-3	-3	2	0
0	3	3	3	0	2	2	2	2	0
0	0	0	0	0	0	0	0	0	0

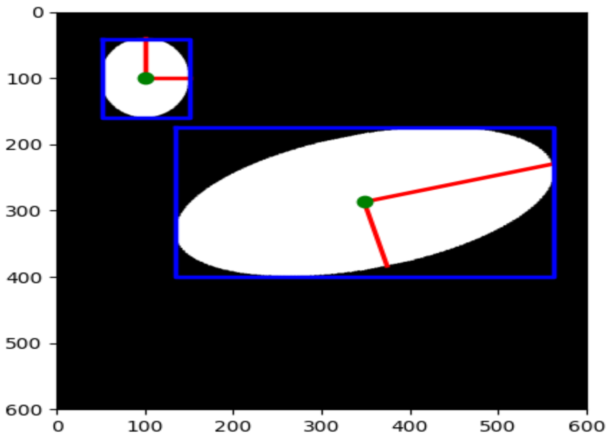
a) Labeled image of foreground and background regions



b) Region adjacency graph

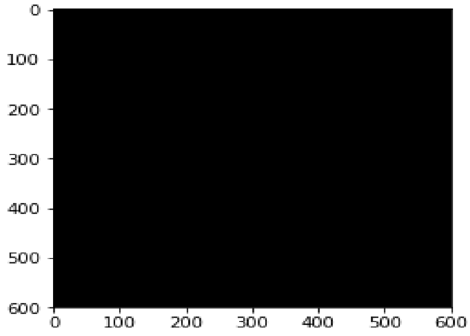
Case study

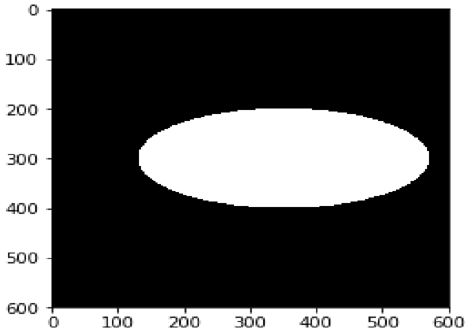
- how to measure properties of labeled image regions, for image with two ellipses.

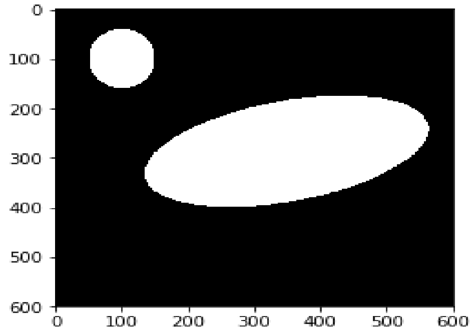


Try:

```
import math
import numpy as np
import pandas as pd
from skimage.draw import ellipse
from skimage.measure  import label, regionprops, regionprops_table
from skimage.transform import rotate
image = np.zeros((600, 600))
plt.imshow(image,cmap="gray")
rr, cc = ellipse(300, 350, 100, 220)
image[rr, cc] = 1
plt.imshow(image,cmap="gray")
image = rotate(image, angle=15, order=0)
rr, cc = ellipse(100, 100, 60, 50)
image[rr, cc] = 1
plt.imshow(image,cmap="gray")
```

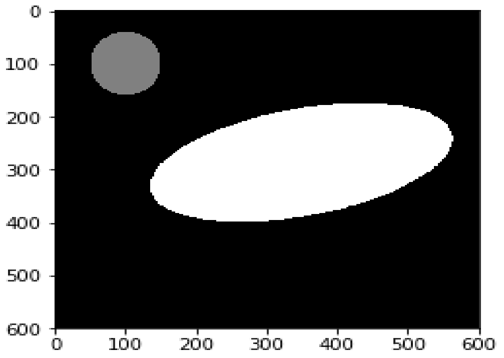


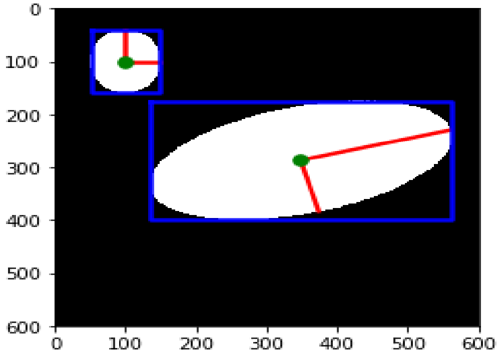




Try:

- label_img = label(image)
- regions = regionprops(label_img)
- fig, ax = plt.subplots()
- ax.imshow(image, cmap=plt.cm.gray)
- for props in regions:
- y0, x0 = props.centroid
- orientation = props.orientation
- x1 = x0 + math.cos(orientation) * 0.5 * props.minor_axis_length
- y1 = y0 - math.sin(orientation) * 0.5 * props.minor_axis_length
- x2 = x0 - math.sin(orientation) * 0.5 * props.major_axis_length
- y2 = y0 - math.cos(orientation) * 0.5 * props.major_axis_length
- ax.plot((x0, x1), (y0, y1), '-r', linewidth=2.5)
- ax.plot((x0, x2), (y0, y2), '-r', linewidth=2.5)
- ax.plot(x0, y0, '.g', markersize=15)
- minr, minc, maxr, maxc = props.bbox
- bx = (minc, maxc, maxc, minc, minc)
- by = (minr, minr, maxr, maxr, minr)
- ax.plot(bx, by, '-b', linewidth=2.5)
- ax.axis((0, 600, 600, 0))
- plt.show()





- #https://scikit-image.org/docs/dev/auto_examples/segmentation/plot_regionprops.html
- props = regionprops_table(label_img, properties=('centroid', 'orientation', 'major_axis_length', 'minor_axis_length'))
- pd.DataFrame(props)

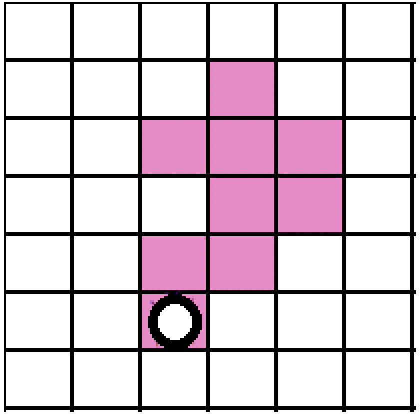
	centroid-0	centroid-1	orientation	major_axis_length	minor_axis_length
0	100.000000	100.000000	0.000000	119.807049	99.823995
1	286.914167	348.412995	-1.308966	440.015503	199.918850

Boundary

- The boundary (also called border or contour) of a region R is the set of pixels in the region that have one or more neighbors that are not in R.
- If R happens to be an entire image, then its boundary is defined as the set of pixels in the first and last rows and columns in the image
- Boundary/contour tracing algorithms
  - Square Tracing algorithm and Moore-Neighbor Tracing
  - Active countour

Contour tracing-Square Tracing algorithm

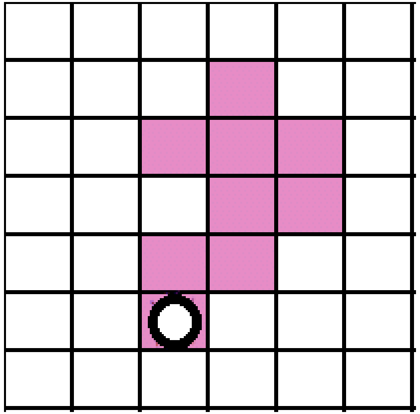
- This was one of the first approaches to extract contours and is quite simple.
- Suppose background is black (1’s) and object is white (0’s).
- start at the bottom left corner of the grid, scan each column of pixels from the bottom going upwards -starting from the leftmost column and proceeding to the right- until we encounter a black pixel. We'll declare that pixel as our "**start**" pixel.
- every time you find yourself standing on a black pixel, turn left and mark the previous pixel as boundary pixel
- every time you find yourself standing on a white pixel, turn right
- repeat, until you encounter the **start** pixel again.
- The black pixels you walked over will be the contour of the pattern.
- This works best with 4-connectivity as it only checks left and right and misses diagonal directions.



http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/square.html

Contour tracing- Moore Boundary Tracing algorithm

- Moore neighborhood of a pixel, **P**, is the set of 8 pixels
- start at the bottom left corner of the grid, scan each column of pixels from the bottom going upwards -starting from the leftmost column and proceeding to the right- until we encounter a black pixel.
- We'll declare that pixel as our "**start**" pixel.
- every time you hit a black pixel, **P**, backtrack i.e. go back to the white pixel you were previously standing on, then, go **around** pixel **P** in a clockwise direction, visiting each pixel in its Moore neighborhood, until you hit a black pixel.
- The algorithm terminates when the start pixel is visited for a second time.
  The black pixels you walked over will be the contour of the pattern.



Snake Contour

- def image_show(image, nrows=1, ncols=1, cmap='gray'):
- fig, ax = plt.subplots(nrows=nrows, ncols=ncols, figsize=(14, 14))
- ax.imshow(image, cmap='gray')
- ax.axis('off')
- return fig, ax
- def circle_points(resolution, center, radius):
- #Generate points which define a circle on an image
- # Centre refers to the centre of the circle

  radians = np.linspace(0, 2*np.pi, resolution)
- c = center[1] + radius*np.cos(radians)
- #polar co-ordinates
- r = center[0] + radius*np.sin(radians)
- return np.array([c, r]).T

```
# Exclude last point because a closed path should not have duplicate points
points = circle_points(200, [80, 250], 80)[:-1]
fig, ax = image_show(img1)
ax.plot(points[:, 0], points[:, 1], '--r', lw=3)
import skimage.segmentation as seg
snake = seg.active_contour(img1, points)
fig, ax = image_show(img1)
ax.plot(points[:, 0], points[:, 1], '--r', lw=3)
ax.plot(snake[:, 0], snake[:, 1], '-b', lw=3);
```

Reference Material

- 1. E. R. Davies, “Computer & Machine Vision”, Fourth Edition, Academic Press, 2012.
- 2. R. Szeliski, “Computer Vision: Algorithms and Applications”, Springer 2011.
- 3. Simon J. D. Prince, “Computer Vision: Models, Learning, and Inference”, Cambridge University Press, 2012.
- 4. Mark Nixon and Alberto S. Aquado, “Feature Extraction & Image Processing for Computer Vision”, Third Edition, Academic Press, 2012.
- 5. Sunita Dhavale, “Advanced Image-Based Spam Detection and Filtering Techniques”, Book Published by CyberTech: An Imprint of MKP Technologies, Hershey, PA, USA IGI Global, March 2017, ISBN13: 9781683180135|ISBN10: 1683180135|EISBN13: 9781683180142|DOI: 10.4018/978-1-68318-013-5.

<<Epilogue>>

- We will meet in next scheduled lecture.
- Implement and try code in python.
- Feel free to ask your questions.
- Email: sunitadhavale@diat.ac.in

Thank You!