Homework4 , CS 682 Spring20
S M Hasan Mansur

- I have posted the results and scripts to http://mason.gmu.edu/~smansur4/cs682hw4/
- Login credentials:
  Username: testuser
  Password: testpass

Problem 1

In this problem I computed boundaries/contours, polygonal approximation of computed boundaries, convex hull, deficits of convexity for the shapes. I also computed area, perimeter, all first and second order image moments, number of convexity deficits and their total area. For implementing these, I used various OpenCV methods like findContours, arcLength, approxPolyDP, convexHull, convexityDefects, moments, contourArea etc. For drawing corresponding polygons I used drawContours method.

Problem2

For all 126 images, I have computed the values of area, perimeter, first & second order moments, number of convexity deficits and their total area. For computing the number of convexity deficits and their total area, I have used the start points, end points & farthest points provided in deficits data.

As per suggested by Professor, though I computed values for all 126 images, in the webpage I presented the values for some of them (image: 00000051.png to 00000094.png).

Problem3

In this section, I implemented the method from the textbook (Concise Computer Vision, page 106-109). to compute curvature along the boundary. I experimented with different window sizes (k = 1, k = 2, k = 3) to determine a better estimation for curvature. To display the curvature estimation, I used the color scheme such that the higher curvature values are represented by ‘hotter’ colors. I have used Matplotlib for plotting the curvature estimation.

Problem4

For computing distance transform, I implemented the Euclidean distance transform algorithm given in the textbook (Concise Computer Vision, page 112).

Later I also computed the same thing using the OpenCV implementation (cv.distanceTransform) of this algorithm. Both provide the same result.

Problem5

According to the provided slides in Blackboard, I implemented chamfer matching and used it to match all pairs of gait images in the provided sequence. I saved the match score result in a 126*126 matrix. Later I plotted this result using Matplotlib & color scheme.

Problem6

(6a)

We observe periodicity in both the results from part 2 & part 5, which means some frames are similar to others. In part 5, if we look into the chamfer match scores result image, we observe the periodicity along the diagonals. We also observe periodicity along the x axis & y axis. In part 2, for example, if we plot the area values (y axis) of all the images (x axis), it will show the periodicity. (plot is provided in the webpage)

(6b)

Two most distinct phases of gait correspond to the widest and the narrowest profiles. We can detect this from the values of the area. If we look into the plot 'images vs area' in 6a (plot is provided in the webpage), we notice that area values change between high & low values in a periodic manner. This clearly depicts the correspondence to the widest and the narrowest profiles.

(6c)

We can use curvature to detect joints and segment body parts. If we look into the image of curvature estimation using k=3 in Problem3, we notice higher curvature values along shoulder, neck, head, chin etc. We can use this information to segment the whole body into two parts, where the upper part holds higher curvature values.

References:

1. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html
2. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contours_begin/py_contours_begin.html
3. https://matplotlib.org/tutorials/colors/colormaps.html