MEHMET TUGRUL SAVRAN

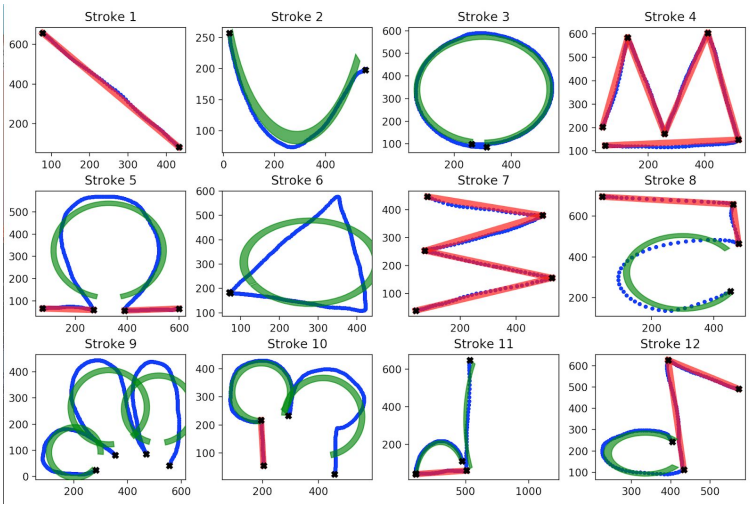
MIT 6.835 INTELLIGENT MULTIMODAL INTERFACES

SPRING 2018

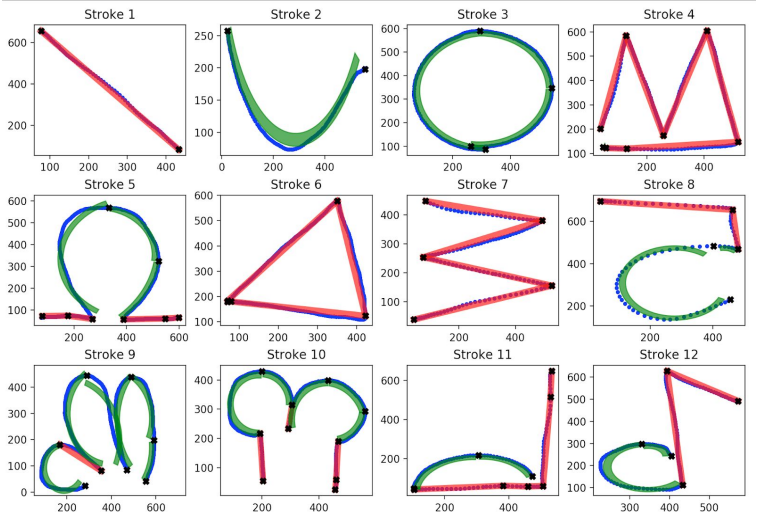
MINI-PROJECT 1
STROKE SEGMENTATION

Some Brief Discussion Before Mandatory Questions

I observed two different "performance versions" of my code. In the first one, where I didn't use the given "detect_peak" method, everything looks very good in shape except the triangle, please see below:



With the detect_peak method, things look quite alright, but a bit "overfit." One could argue that the extra corners identified in Stroke 9 (the cauliflower shaped stroke) actually make the system fit better circles, but hey, they are not corners anyways. See the picture below, and contrast the two versions -especially Stroke 9 and 10.



(a) Answer to the question in step 4:

Purely using Python atan's function returns results in range [-pi,pi]. However, this results in the angle being "wrapped." The slope changes direction at vertical tangents, which causes discontinuity.

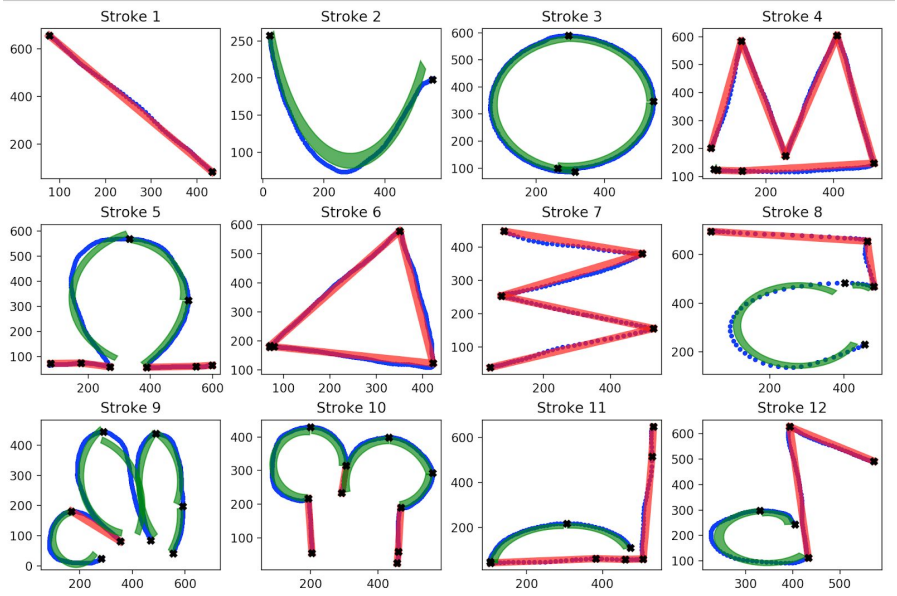
Hence, the remedy for that was to transform the domain from [-pi,pi] to [0,2*pi].

(b) My definition of nearly coincident was based on *index proximity*. If candidate corner points were close listed within 12 indices in stroke data, then they were considered *nearly coincident* and only the first one was taken as the representative. I chose 12 based on empirical running of the code.

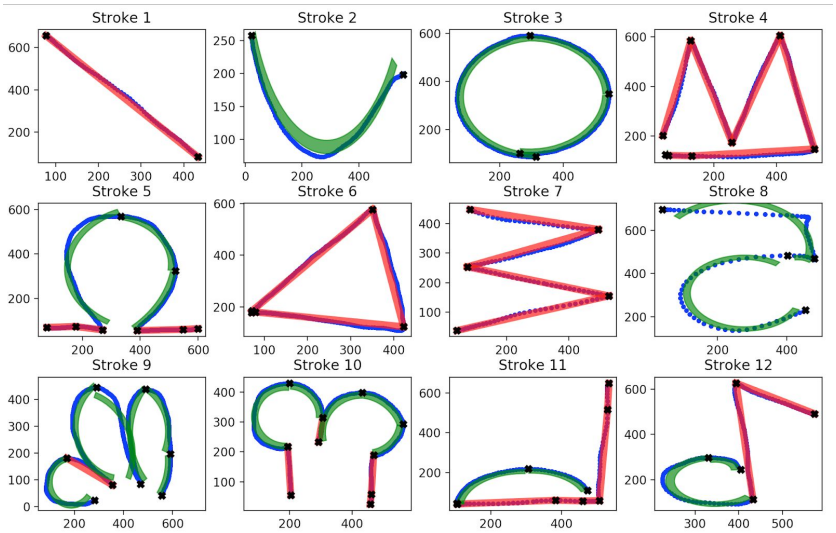
(c) With observation and experimentation, I "hand-tuned" 2 parameters which I believe improved my performance. *My code worked better with the "tangent window" parameter 7* rather than 11 used in the paper. Values are reproduced below on the table, and the contrast of my results is shown on the figures below.

Parameter	Value Used In Paper	Value I Used
Size of window for smoothing Pen speed	5 total (2 points on each side)	5 total (2 points on each side)
Size of window for computing tangent	11	**7**
Speed Threshold 1	0.25 of average	0.25 of average
Curvature Threshold	0.75 Degree/Pixel	0.75 Degree/Pixel
Speed Threshold 2	0.80 of average	0.80 of average
Minimum distance allowed between 2 corners	unspecified	**12 indices**
Minimum Arc Angle	36 degrees	36 degrees

SEGMENTATION WITH MY PARAMETERS



SEGMENTATION WITH DEFAULT PARAMETERS



(d) Reading the **Future Work** section of the paper, I saw that most of the discussion boils down to one specific aspect: tuning of parameters. Indeed, one suggestion they made is really apt: to have the system adapt to the user. I believe this could be a very nice Machine Learning task to help system identify the user and tune the parameters. For example, whenever the system is *just right in between choosing a line versus circle,* a sleek pop-up window could ask the user whether he/she intended to draw a circle or a line. I believe my system, which is a re-implementation of this paper, would also benefit from this approach.