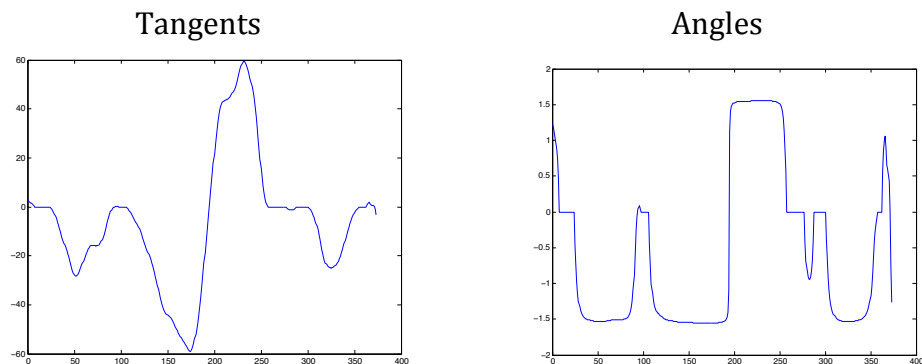
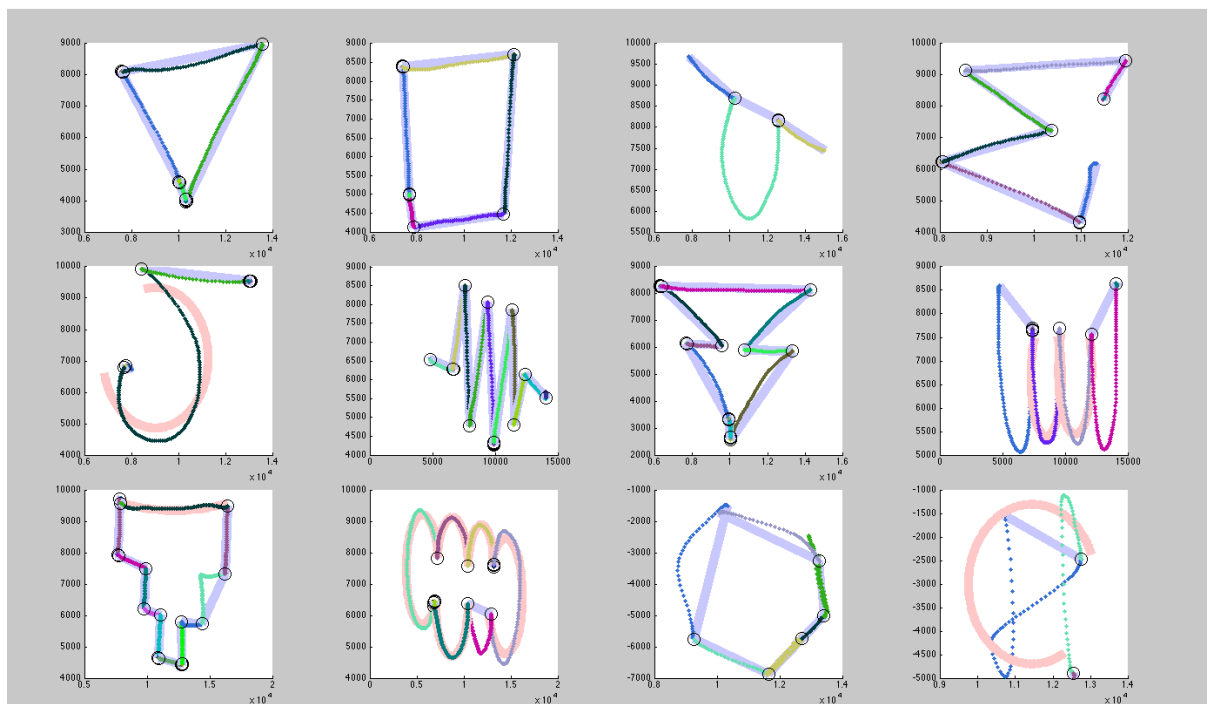


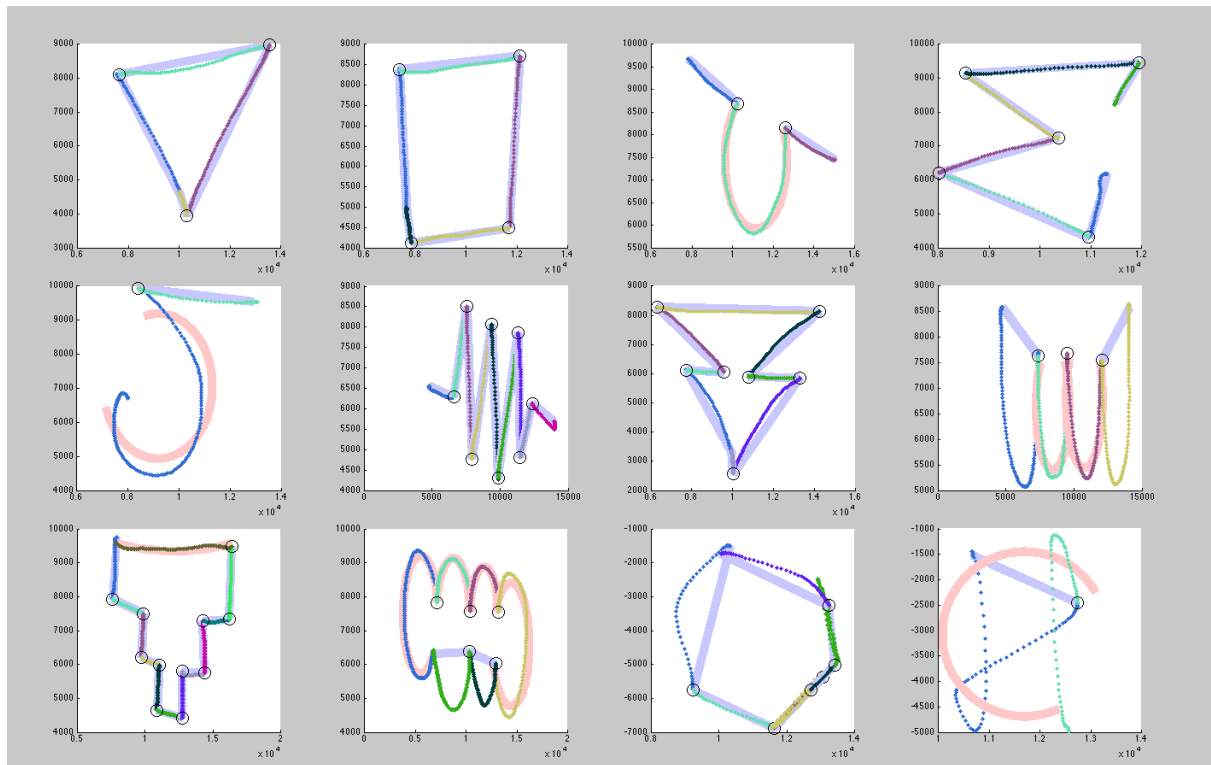
1. All files in Stellar submission under [andrewmo@mit.edu](mailto:andrewmo@mit.edu)
2. When plotting the angles after using the arctangent function, we see that there is a slight problem. For example, we notice this with the  $\Omega$  symbol in strokes(3). Since the matlab atan function returns a radian between  $\pi/2$  and  $-\pi/2$ , a sudden transition can occur when drawing a full circle. In particular, a small change of  $\Delta$  can switch suddenly from being relative to  $\pi/2$  or  $-\pi/2$ . Since curvature is the derivative of angle, a delta spike will be observed in the non-continuous function.



3. “Nearly Coincident” is the minimum allowed distance between two corners. After much testing, I chose a nearly coincident value of 240px. Taking into consideration the pen speed when drawing corners, too low added more corners than necessary and too high failed to recognize obvious corners. This value chosen had the smallest number of deviations in most of the cases.
4. evalAll(strokes) – Default Params



## 5. evalAll(strokes) – Updated Params



- I kept most of the parameters the same except for a few. Testing allowed me to see slightly better depictions of strokes when I slightly increased or decreased the windows for calculation. As described in part 3 above, I had to determine the minimum allowed distance between two corners (“nearly coincident”) through trial and error. The main addition was the use of a distance threshold between a corner compared to the end/start of a stroke. This helped drop unnecessary corners when the user was changing speed in there stroke (slowing down, changing direction, etc).

Parameter	Value I used
Size of window for smoothing pen speed	5
Size of window for computing tangent	10
Speed threshold 1	.25
Curvature threshold	.75
Speed threshold 2	.80
Minimum allowed distance between two corners	240
Minimum arc angle	36
Minimum start distance	450

- As described in the paper, there are multiple ways to improve the system. A better implementation of curve type could help refine the granularity of the curve strokes. Also, users have different behaviors. Wit the help from machine learning, we could refine the parameters as a specified user uses the system and improve the results. Though my system s incomplete with still some flaws, there are many more ways to improve existing factors or add additional ones.