

## **Matsakis Paper**

Professor had wanted us to read the beginning for preprocessing steps.

Canonical ordering is very useful.

A question was asked basically wondering about an ordering standard. It seems like it doesn't matter as long as you are consistent.

The paper uses a number of points per canonical order. The stroke amount per gesture/symbol can influence the points per stroke. The amount of points to have per gesture/symbol, however, will mostly be dependant upon the algorithm that you have chosen.

Stroke reversal is important and should be understood. This allows you to use information from the stroke to determine if it is horizontal, vertical, closed, or diagonal.

Directional data can also help detect closed figures. For more accurate detection when hitting figures of ambiguity use some more context from near by, or make an easy to use correction interface.

## **Guerfali Paper**

Designed for cursive hand writing (curves and a slant). Finds the base line and orientation.

Professor wanted to make sure everyone knew that just because we remove information to help with one step, doesn't make the information useless, and it may even help solve a problem in another step later.

## **Sezgin Paper**

Useful for cusp detection for assignment 2, however, the professor thinks the methods used in the paper are a little overkill. The speed detection method should work well enough for cusp detection.

## **Assignment 2**

Strokes collected event will give you stylus points that can be changed before being drawn.

Just need to do 5 shapes and operation on the shapes.

Use threshold value on end points for gap elimination.

Stay in ink editing mode (cannot use the fancy tools from the last assignment).

Always have gestures being detected.

Beautify the shape for the user.

Intersection testing – easiest way is  $O(n^2)$ . Check every line segment of the stroke against the other line segments of the stroke for intersection.

Leave the unrecognized ink on the canvas.

Transform point to origin? You need to decide if this will help you. It may hurt you it may not.

Cusp detection – lots of ways to do this. Very difficult. Keep the least squares cheat sheet for help.

Approach 1 – small segments (5 points or so)

1. Get direction information.
  - a. Calculate the change in x and y. and take their arctan.
  - b. Find Y or orthogonal best fit line.
2. Get curvature data.
  - a. Partial derivative with respect to arc length of tangent vector.

Approach 2 – recommend using a Gaussian filter.

1. People slow down at corners.
  - a. Keep speed data to help determine cusp.
  - b. Minimums of speed data are probable corners.

Remember: calculating a derivative of noisy data makes it even noisier. (Integrals smooth the data down.)