Assignment 3 – Exploring $-Family Recognizers
CAP6105

Due: 10/21/16 11:59pm

This purpose of this assignment is twofold. First, it is designed to give you experience with simple recognizers. Second, it is to give you experience in implementing a state of the art algorithms from research papers and to extend one of them to improve upon it.

Requirements

There are two main requirements for this assignment. First, you will implement 3 $-family recognizers and compare them. The first is the $N recognizer discussed here:

Anthony, L. and Wobbrock, J.O. (2010). A lightweight multistroke recognizer for user interface prototypes. *Proceedings of Graphics Interface (GI '10).* Ottawa, Ontario (May 31-June 2, 2010). Toronto, Ontario: Canadian Information Processing Society, pp. 245-252.

The second recognizer is known as Protractor and details can be found here:

Li, Yang (2010). Protractor: a fast and accurate gesture recognizer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10).* ACM, New York, NY, USA, 2169-2172, April 2010.

The third recognizer is known as Penny Pincher and the details can be found here:

Taranta, E. and LaViola, J. Penny Pincher: A Blazing Fast, Highly Accurate $-Family Recognizer, *Proceedings of Graphics Interface 2015*, 195-202, June 2015.

You should test these algorithms with the following symbols:

0,1,2,3,4,5,6,7,8,9,+,-,*,t,a,n,s,c,i, and the square root symbol.

You should use your scribble erase from the last assignment to erase symbols.

Train each recognizer with 1, 3, and 5 samples per symbol. Test each recognizer by writing each symbol 5 times, which should give you a good accuracy number. You should run separate experiments using the data you collected using the stylus and the touch screen with a finger. Please put the results of your experiment in the README file.

Second, you will improve upon one of these algorithms to try to improve recognition accuracy. Your approach can be based on any strategy you want including machine learning methods, heuristic methods, etc…

Strategy

To implement your symbol recognizers, there are some things you need to consider.

1. You need to find a way to invoke the recognizer. You can have it run in real time or in batch mode (for ex. lassoing the symbol or symbols and taping to invoke the recognizer).

2. Regardless of the invocation method, you will need some form of ink segmentation since you must be able to detect when a symbol has 2 or more strokes. Simple line segment intersection should suffice here since it is relatively easy to determine if you have a multi-stroke symbol in our alphabet.

3. You will need to show recognition results to the user. A simple text box is fine but if you want to be more elaborate feel free to do so.

Deliverables

You place your source on the ISUE Lab drive and include a README file describing what works and what does not, the results of your experiments, any known bugs, and any problems you encountered.

Grading

Grading will be loosely based on the following:

50% correct implementation of the $N recognizer
30% extending a recognizer’s accuracy
20% documentation