# State of the Research – Recognition

Miranda Parker

Fall 2011

#### Note:

If you want a full summary of the currently recognition process, please read “State of the Research-Recognition” for Summer 2011. This paper only goes into further results on testing the Adaptive Image Recognizer.

## **Goals**

My primary goal for this semester was to further understand the recognition process. We received test results at the end of the summer, but they were vague and not true to the user experience. In order to properly access how recognition was working, I wanted to further test all aspects of the recognition.

I also wanted to differentiate between the different ways a recognizer can be trained, and determine which method is better for the user. On one hand, we can train the recognizer on perfect data, which is how our system tells them to draw. On the other hand, we can train the recognizer on semi-perfect data taken from a user, so that it gives the system experience with user’s drawing styles.

## **State of the Code and Activities**

### **Shape Recognition Training**

Over the summer, we trained our recognizers on a set of “perfect” data: data that was hand-drawn exactly how we wanted the user to draw. We used a template to accomplish this, so that all templates were the same except for human inaccuracies in the drawing. However, the perfect data removes a lot of the human drawing aspect of the recognition process. Most users will not have perfect drawing styles, no matter how much we want them to. Thus, I experimented on the recognizers by training them on different data sets to see what provided the most accurate recognitions.

### **Adaptive Image Recognizer Phase out**

Our image recognizer is adaptive due to its ability to learn from a template. Usually, the recognizer will learn from a template when a user corrects a misrecognition error. The recognizer learns in part because each template keeps track of what shape type it is and what “score” it has. When we learn from a given template, we check to see if the template that was matched to the shape was of the wrong shape type (i.e. if an OR template was used for an AND gate recognition), and if so we deduct a given amount from their score. When we remove a template during the learning process, we find the template with the lowest score in the list of templates and remove it. After we remove the template with the lowest score, we add the shape (the one that the learning method was called on, such as a re-labeled shape) to our list of templates. It should be noted that we only remove a template if adding the template would cause us to go over our limit of templates.

I worked specifically on proving that our recognizer is adaptive and can obtain near-perfect recognition rates after a few sketches. To do this, I altered a testing procedure that we created over the summer, so that the tests would better model real-life scenarios. To test the Adaptive Image Recognizer, I recognize all the shapes in the testing set, and split them into “right” and “wrong” groups based on their recognition. I then create “test” and “train” groups, where all the right templates automatically go to the test set, and the wrong ones are split up so that half are trained on and half are tested on. A template is randomly selected from the training set, learned from, removed from the list of templates, and then the recognition accuracy is noted. Then the whole process repeats itself, with the “right” and “wrong” templates being split up based on how the recognizer analyzed the data after the addition of the previous template. The process stops if accuracy reaches a hundred percent, or if there are not enough “wrong” templates left to properly split them between the testing and training sets.

## **Results**

### **Shape Recognition**

In order to ascertain the appropriate training set for the recognizers, I trained the Image and Adaptive Image recognizers on differing data sets and tested them all on the same set to obtain initial recognition rates. Although all recognition rates were above seventy percent, there was an obvious difference between the data sets.

To obtain these results, I trained the Image and Adaptive Image recognizer on our “perfect” data (which can be found in Data\ Adaptive Image Testing Data), as well as the final circuits from three users from our Drawing Style Study (Users 6, 7, and 9, all can be found in Data\DrawingStyleStudyData). The users were chosen semi-arbitrarily, based on a previous test that looked at their initial recognition rates. User 7 had extremely low initial recognition rates, User 6 was average, and User 9 was above average. Every time that the recognizers were trained, they were then tested on the complete set of Drawing Style Study data. The test was done with the Symbol Stage in Test Rig, with the Adaptive Image recognizer as the default recognizer. This means that it was not adapting to the sketches, which is what was desired since only the initial recognition rates were necessary.

The graph below displays the results from these tests. The “perfect” data set consistently had the lowest accuracy, and thus is not ideal for our recognizer to be trained on. However, the recognizer trained on User 9 had the highest initial accuracy a majority of the time. Please note that the results from testing on Users 6, 7, and 9 are omitted from the graph since those are what the recognizer was trained on in some cases.

The Image and Adaptive Image recognizers that I commit at the end of the semester will be trained on User 9.

This is the pure Adaptive Image Recognizer with learning turned off. It is trained on 3 different data sets, mentioned above and in the legend. It is tested on data from the Drawing Style Study data.

### **Adaptive Image Recognizer**

In order to prove how adaptive our image recognizer could be, I compared the Adaptive Image Recognizer data to the Image recognizer data. The revised testing procedure for the adaptive image recognizer was previously discussed in the “Adaptive Image Recognizer Phase Out” section. For these tests, I ran three trials of the phase out test. This is because part of the stage is randomly picking a gate to learn on. Depending on how much the user changed their drawing style, one gate can make a huge difference. Thus, I felt it was fair and necessary to run three trials of the phase out test and then average the final recognition rates together.

For this test, I used the Adaptive and Image Recognizers trained on User 9 from the drawing style study data. I tested them both on all twenty-four users from the Gate Study data. However, I only used the data from the study where the user’s used a tablet pc, and not where they used a Wacom tablet with paper overlay. I felt that this was necessary to properly model our system, since the drawing interfaces between the two systems are wildly different.

It should be noted that although the final recognition accuracy for the Adaptive Image Recognizer is always greater than the accuracy for the Image Recognizer, there is a large amount of time and work put into the Adaptive Image Recognizer to get to that point. Thus I have also included a graph to illustrate, on average, how many templates it took for the Adaptive Image Recognizer to add at reach its final accuracy.

This is the pure Adaptive Image Recognizer against the pure Image Recognizer. They are trained on User 9 from the Drawing Style Study data. It is tested on data from the Gate Study data.

This is the number of templates that the AIR adds by the time it reaches its final accuracy. The AIR was trained on User 9 from the Drawing Style Study data. It was tested on the tablet data from all users in the Gate Study Data.

I ran another round of tests considering the recognizer in the entire pipeline, and thus not pure. This is only the Image Recognizer in the pipeline, but it still provides a sense of how the shape recognition is doing after classification and grouping has occurred. The graph displays the recognition quality compared to something called “real recognition quality,” which is explained in the caption beneath the graph.

This is the Image Recognizer in the pipeline. The recognition quality is the number of correctly recognized shapes out of the number of correctly grouped strokes. The real recognition quality is the number of correctly recognized shapes out of the total number of groups. The recognizer is trained on User 9 from the Drawing Style Study data. It is tested on data from the Drawing Style Study data.

## **Future Work**

* Create a testing set for the Adaptive Image Recognizer Phase out stage that is perpetuated form one wrong template in the data set. This would end any worries about not having enough gates to test and train on.
* Applying LogiSketch to other domains, such as electrical circuits. Some of the code is there, but there is no training in the recognizers.
* The grouper needs to be improved. Currently, the grouper is what is causing overall recognition rates to decrease so much in pipeline testing and real life scenarios. The first thing that can be inspected is if it is trained on proper or enough data.