

# Activities and Findings - Summer 2009

Denis Aleshin

August 11, 2009

## 1 Summary of Activities

We started the summer by getting acquainted with the code, and doing some major restructuring to some of the files. Most of my work was concerned with CircuitSimulatorUI. The tools were made more reliable, functionality was added to file management and tools were created for testing. Furthermore, the code was refactored to abstract away the single stroke recognition, grouping and shape recognition modules, so that these could be switched out and tested more easily. Finally, the visual feedback was expanded to demonstrate all of the different parts of the recognition process.

One major objective of the summer was to investigate the use of context in free drawn sketch recognition. We considered several ways of doing this—by creating a universal heuristic to guide a search process, by applying a local heuristic in the hope of improving the overall quality of recognition, and utilizing connectedness as a feature in the recognition process.

In the process of looking at these options, I created a refiner tool that would take a partially recognized sketch, and attempt to improve the recognition by applying local changes in the form of shuffling strokes between adjacent shapes.

We then considered using connectedness as a feature, for which I developed a tool for recognizing connections. This tool looked for connections between endpoints of wires and other shapes. With this tool we could now augment sketches that have been recognized with the single stroke classifier and grouped with connection information.

After Josh left, I helped Eric with his ImageAligner recognizer, that was designed to recognize parts of shapes, and identify when extra strokes were present. I incorporated the recognizer into CircuitSimulatorUI and tested it, as well as trained a user-independent version, as Eric's recognizer was user-specific.

Finally, I looked into using alternate grouping strategies. I implemented and tested 3 groupers, one that worked purely off of how the strokes were connected, one that used the sequence in which the strokes were drawn and looked for loops to identify gates, and one that attempted to recognize the strokes as primitive shapes and then combine the primitive shapes together using geometric restrictions.

## 2 Findings

While considering a probabilistic approach to the overall heuristic for the sketch, we found that it eventually reduced down to a CRF, which has been investigated before, so we decided not to pursue this particular strategy.

The refiner also did not work as expected, since it uses the recognition result to guide the refinement process. The recognizers that we were using did not give useful results for partial shapes or shapes with extra strokes. The result of this was unpredictable behaviour, where the refiner would lump together large chunks of strokes, or not do anything at all. Furthermore, the search space is extremely wide, and full of plateaus, since there are many options for regrouping, reclassifying and relabeling strokes.

The recognizers we worked with during the summer- the Combination Recognizer, the Naive Bayes Recognizer and the Partial Shape recognizer- all gave decent and comparable results when applied to perfect groups, however all of them suffered drawbacks. All but the Naive Bayes Recognizer could not distinguish between wires, labels, and gates and relied purely on the result of the stroke classification to distinguish between them. When applied to single stroke classifier and grouper results, all recognizers performed poorly. <sup>1</sup>

These errors were due to the mistakes made in the single stroke recognition and grouping process. <sup>2</sup> With this in mind, I developed alternate grouping strategies that instead utilized time information, and primitive shape recognition.

By using just fits to arc segments and lines, we can distinguish parts of gates with roughly 90 percent accuracy. <sup>3</sup>

The grouping strategies did not work completely- as many of the shapes were still not complete. Still, over 90 percent of the pairings that the groupers made were correct. Combining the groupers, while doing poorly on separating labelboxes from labels, did very well on grouping AND and OR gates, getting 242 out of 281 AND gates correctly, and 179 out of 202 OR gates correctly. <sup>4</sup>

## 3 Future Research

Some of the assumptions made about the connection information could be removed. Currently, we assume that every wire is connected to something on either side.

A major obstacle is still grouping- more reliable ways to create complete and correct groupings needs to be developed. For the existing methods, combining

---

<sup>1</sup>Results for recognizer testing can be found on the wiki: <https://www.cs.hmc.edu/twiki/bin/view/Sketchers/RecognizerTesting>

<sup>2</sup>Results for the spacial and neural groupers can be seen here: <https://www.cs.hmc.edu/twiki/bin/view/Sketchers/GroupTesting>

<sup>3</sup>Results for primitive recognition can be seen here: <https://www.cs.hmc.edu/twiki/bin/view/Sketchers/PrimitiveRecognizer>

<sup>4</sup> Results for the time based and primitive based grouper, and their combination can be found here: <https://www.cs.hmc.edu/twiki/bin/view/Sketchers/GrouperTestResults>

the time, primitive and neural groupers could yield interesting results. Perhaps the simplest and most effective improvement would be to implement a parabola primitive to improve the recognition of gates.

Another idea is to use primitive recognition information to beautify the sketch- to group strokes that are parts of similar primitive fits.

Once a solid grouping is achieved, the refiner may be used in combination with the ImageAligner recognizer to improve the grouping and finish the recognition.