ETCHA Sketches: Lessons Learned from Collecting Sketch Data

MIKE OLTMANS and CHRISTINE ALVARADO and RANDALL DAVIS
MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street
Cambridge, MA 02139
{moltmans, calvarad, davis}@csail.mit.edu

Abstract

We present ETCHA Sketches–an Experimental Test Corpus of Hand Annotated Sketches–with the goal of facilitating the development of a standard test corpus for sketch understanding research. To date we have collected sketches from four domains: circuit diagrams, family trees, floor plans and geometric configurations. We have also labeled many of the strokes in these data sets with geometric primitive labels (e.g., line, arc, polyline, polygon, and ellipse). We found accurate labeling of data to be a more complex task than may be anticipated. The complexity arises because labeled data can be used for different purposes with different requirements, and because some strokes are ambiguous and can legitimately be put into multiple categories. We discuss several different labeling methods and some properties of the sketches that became apparent from the process of collecting and labeling the data. The data sets are available online at http://rationale.csail.mit.edu/ETCHASketches.

Introduction

In recent years, improvements have been made in both low-level stroke classification and high level symbol understanding (Davis, Landay, & Stahovich 2002). Although each of these tasks requires the analysis of sketch data, to date there has been no standard test corpus to use in developing or evaluating these systems. Instead, each researcher has collected his or her own data, a process which is both time consuming and makes it difficult to compare sketch recognition technologies. ETCHA Sketches (an Experimental Test Corpus of Hand Annotated Sketches) addresses these problems by providing a corpus of freely-drawn sketches in a number of domains and a medium through which sketch recognition researchers can share their data. Such corpora have proved extremely useful in other research areas, e.g., spoken language systems and computational linguistics (Linguistic Data Consortium 2004).

First, we describe the sketches we collected from a variety of domains. Our goal was to have data from several different domains so that we could explore the differences between them and so that our data set would avoid the particular biases of any one domain.

Second, in order to use the data for evaluation and training tasks it was necessary to label the strokes with the primitive shapes that they depict: line, arc, ellipse, polyline, polygon, or other. The process of assigning labels turned out to be more subtle than it first appeared. We ended up collecting four sets of labels with slightly different semantics and with different intended purposes. For example, when evaluating a low level classifier we may be interested in both how often it outputs the correct answer and how often it outputs one of a set of acceptable answers. We wanted to support both of these types of evaluations and several others that we describe below. We describe how we collected these different labels and how we anticipate them being used.

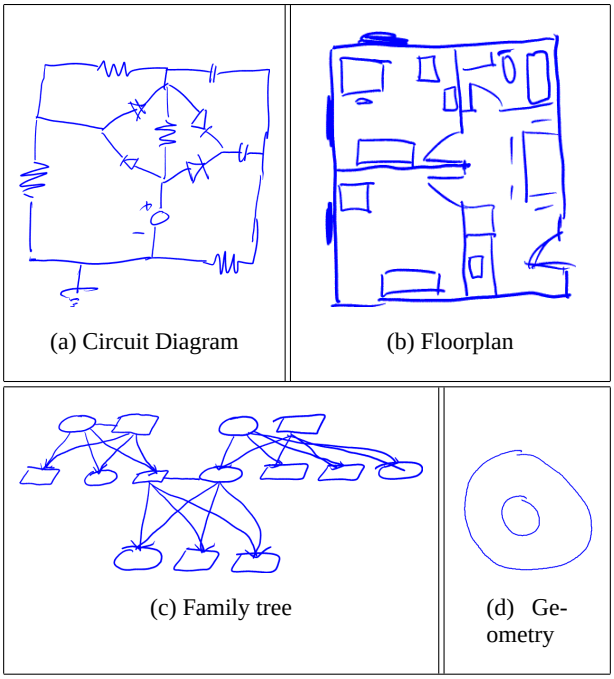


Figure 1: Sample sketches from several domains

Third, we share some observations about the sketches and their labellings that revealed some interesting properties of sketching. For example, we often observed that users did not draw complex shapes with consecutive strokes, but rather drew part of one object, then switched to another before returning to complete the first one.

Appeared in *AAAI 2004 Symposium on Making Pen-Based Interaction Intelligent and Natural, 2004.* pp.134-140.

Stroke Collection

Our goal was to gather data from a variety of different domains, to gain an understanding of different types of sketches and to see if they varied in substantive ways. We have restricted ourselves initially to sketches made with very simple interfaces (described below), in order to study sketching styles that are not influenced by specific interface capabilities. Here, we describe the four domains we gathered sketches from, the setup we used to collect them, and the representations we use for the data.

Sketch Collection in Different Domains

By studying a broad range of domains we hope to identify a representative and realistic set of issues facing sketch recognition systems. Representative sketches collected from four different domains can be seen in Figure 1. The collection process for each domain is described below.

Circuit Diagrams To collect the circuit data, we solicited users with circuit design experience and asked them to produce sketches with certain properties. For example, we asked them to sketch a circuit with 1 battery, 3 resistors, 1 capacitor, 1 transistor and ground. The users were members of the MIT community and were all familiar with circuit design and had significant training and experience producing sketches of circuits from coursework and design.

This domain was one of the two based on simple compositions of geometric shapes (e.g., lines for a resistor and an arrow in a circle for a current source). One of the features of the circuit domain was the heavy use of lines in many different shapes (e.g., wires, resisters, ground symbols, etc. . .) and a tendency for users to draw multiple shapes with a single stroke (e.g., two resistors and a wire connecting them were frequently drawn with one stroke).

Family Trees Like circuit diagrams, family tree diagrams were a good source of simple geometric shapes drawn for a particular task. They included a number of shapes that did not appear or were rare in circuit diagrams such as quadrilaterals and ellipses.

Family tree sketches were collected by asking subjects to draw a tree (not necessarily their own) using ellipses for females, quadrilaterals for males, lines for marriages, jagged lines for divorces and arrows for parent to child links. Some subjects used text to fill in the names of people and others did not. We did not use the sketches with text in the stroke labeling process because we see the handling of text and mixed text graphics to be a higher level task than the classification of shapes and wanted to avoid attacking that particular problem at this point. We could have used the sketches with the text removed but this was unnecessary because we had sufficient numbers of sketches without text.

Floor Plans The floor plan domain was an interesting contrast to the other domains because floor plans are not based on compositions of geometric shapes. The important features in floor plans are usually the rooms created by the walls and not the walls themselves. This changes the focus of the drawing task. We wanted to contrast the more geometric nature of the other domains with a more free form domain such as this one, while staying away from completely free form sketches such as maps or artistic drawings.

When collecting floor plan sketches we asked users to draw simple bird's eye views of single story apartments. They were asked first to draw their current apartment (as a warm up task), brainstorm several apartments they would like to live in, choose one of them to make a cleaned up drawing of, and finally, to verbally describe the design to the moderator the while redrawing it one final time. The subjects were not architects and had no explicit sketching experience, but the task was accessible because of people's familiarity with floor plans.

Geometric Configurations We also wanted to include one domain in which the users were not performing any particular task. In this data set, users were simply asked to draw a number of different geometric shapes and configurations. For example: "Draw two concentric circles" or "Draw two lines that meet at an acute angle." As we discuss later, strokes from this domain were more consistently labeled than strokes from the circuit and family tree domains. This implies that the strokes were less ambiguous and it highlights the importance of collecting data in realistic contexts.

Equipment and Software

We collected all of the sketches on Tablet-PCs using pen sized styli. We chose Tablet-PCs over technologies that record physical ink (Mimio, CrossPad, HP's Digital Pen) because our goal is to build interactive recognition systems that work with digital ink. In all cases users were presented with a simple interface that contained a description of what they should draw and a large area in which to draw the figure. There were no editing capabilities (e.g. copy, paste) in the interface. However, the eraser end of the stylus could be used to delete strokes. The use of the eraser had one significant impact because it was configured to delete full strokes rather than pixels. Users informed us that they quickly learned to compensate for this by drawing more and smaller strokes to avoid deleting more than they intended. In the future we plan to support the more natural pixel-based deletion mechanism.

No recognition feedback was given to the users because it can significantly modify their drawing behavior. The modification occurs because users adapt their drawing style so that their figures are recognized more reliably. Furthermore, as described in (Hong *et al.* 2002) users do not necessarily want recognition feedback. As a result we avoided giving users any feedback.

The sketching interface was implemented in C# using Microsoft's Ink API. This allowed us to display pressure sensitive ink that was visually realistic.

Data Storage

In collecting our data we have found it useful to have three different storage media. First, the sketches are collected using Microsoft's instrumented GIF images, which contain both an image and the stroke data. Second, we extract just

	Best	CanBeA	Context	IsA	Total(*)
# of labelers	19	50	19	44	105
# of strokes labeled	466	175	543	186	814
# of strokes in corpus	*387*	*154*	*467*	*162*	*750*

(*) Some labelers and strokes appeared in multiple conditions so the total is not equal to the sum of the columns

Table 1: Sizes of the different label sets.

the information that concerns the strokes, including: x position, y position, and time at each data point. This format is much simpler than Microsoft's, and is easily accessible without depending upon the Microsoft Ink API. In the future we plan to include pressure information, but have not yet done so because we have yet to use that property.

Third, we have found it useful to organize the data in an SQL database. Postgres has built-in support for geometric data types and was a natural choice. The database has been extremely useful in organizing the large amounts of data we have collected that spans many dimensions, such as different domains, tasks within domains, authors, sketches, strokes, and labeling information. With the database we can retrieve, for example, all of the strokes over 20 pixels long that were labeled as lines from any sketch in any domain. Without the database such a query requires extensive indexing (essentially a custom built database) or huge numbers of file accesses to load all of the strokes we have collected. The database can answer such queries efficiently, is easy to update, and provides a convenient central storage for our data.

The data is published on our website at http://rationale.csail.mit.edu/ETCHASketches/ a downloadable archive of text based files. We are currently considering API's or web based UI's to allow researchers elsewhere to directly browse and contribute to the database. Suggestions from the community as to how this may be useful would be appreciated. A summary of the quantity of data acquired from each domain is listed in Table 2.

Stroke Labeling

If the data is to be useful for evaluation and training, we need to know what the right answer(s) are. As an initial step we have focused on the labeling of individual strokes as one of the following primitive shapes: line, arc, polyline, polygon, ellipse, and other). We anticipated that the assignment of strokes into classes of primitive shapes would be straightforward, but the process was more complex than we anticipated. It taught us some lessons that we feel are widely relevant.

The primary issue we encountered was that labels can be used for different purposes. Such as evaluating two classifiers, one that returns a single class and one that returns ranked outputs. They will need different types of data in order to perform both evaluations. As a result, we have collected four different label sets in an attempt to cover a wide range of possible uses of the labeled strokes. Each of these label sets was collected with slightly different processes, described below.

The Four Label Sets

Best in Isolation The first label set indicates the most likely interpretation of the stroke when it is evaluated by itself. These labels are best suited for training and evaluating low level classifiers by asking the question: How many times does the classifier return the correct label for a stroke? In this case we define the correct label to be the one that the stroke most resembles when it is evaluated in isolation. We chose this definition because it matches the common purpose of low level classifiers, to classify single strokes. This definition is also the most appropriate for training a statistical classifier that operates on individual strokes. It would be misleading to have strokes in the training set labeled in context, because this information is not available to a single stroke classifier.

Context The second label set contains labels that take into account the context of the entire sketch. These labels are more suited for comparison with a higher level recognition system that analyzes more than a single stroke at a time. This label set also allows us to study the role of context in classifying strokes, both for the people doing the labeling and for an automated system.

IsA and CanBeA The third and fourth label sets assign a set of labels to each stroke. For example, a stroke that looks like both a line and an arc (because it is only slightly curved) would get two labels. The distinction between the *IsA* and *CanBeA* sets is in the semantics of class membership. For *IsA* the semantics are that the stroke *is* an instance of each assigned type. For *CanBeA* the semantics are that the stroke *could possibly be* an instance of each assigned type. The two sets have the same general purpose but the second allows for a larger range of possible interpretations. The most important question these label sets can answer is: How well matched to the list of possible labels from the label set is the output of a classifier that ranks multiple possible interpretations. Good performance on this task is extremely important when using the output of a stroke classifier as the input to a higher level recognition process, because the higher level process must have a large enough set of options to perform recognition accurately but too many possibilities lead to prohibitively inefficient recognition.

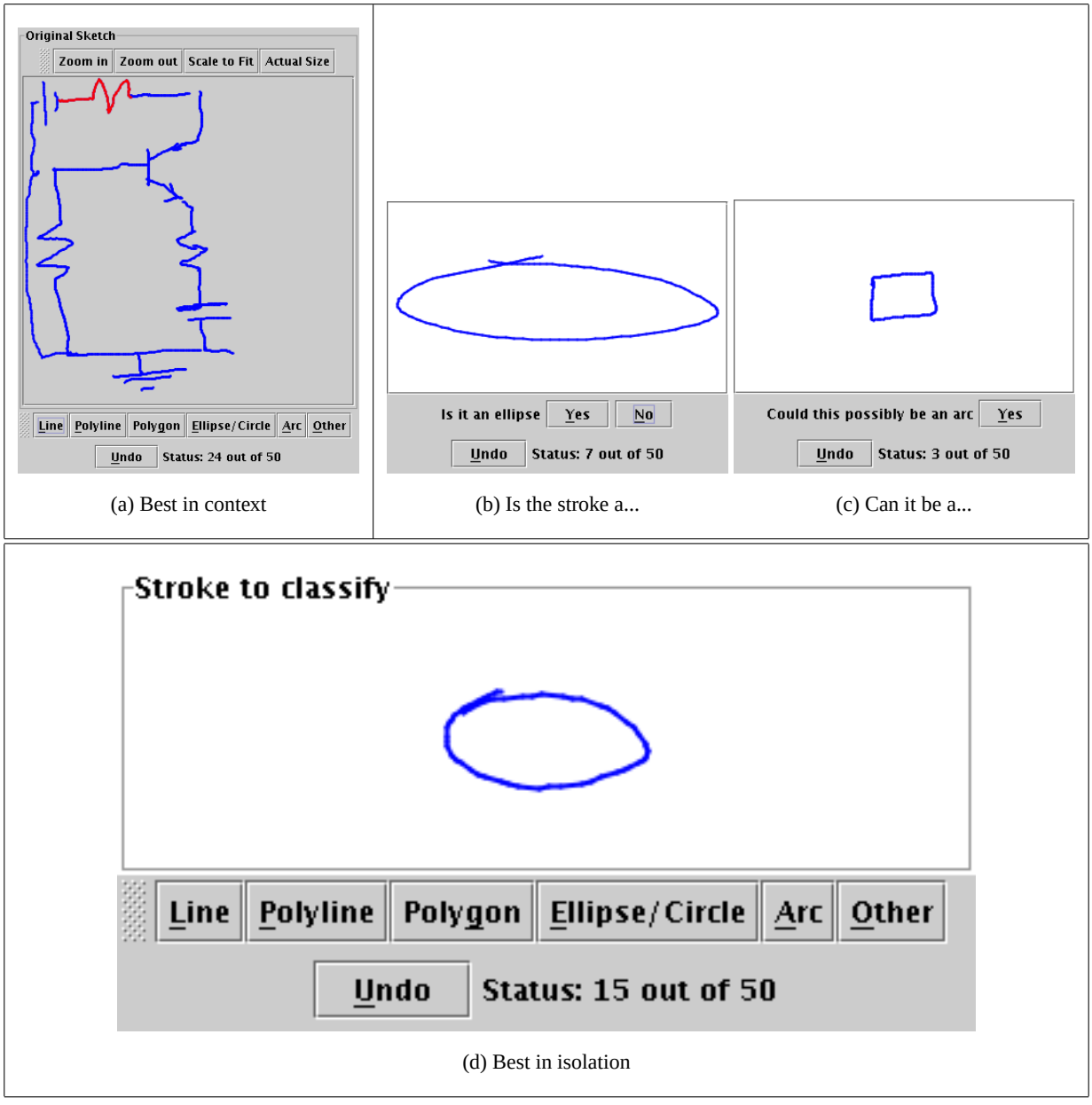
The Labeling Processes

After breaking down the types of labels we wanted to gather, we constructed four different graphical interfaces to implement the labeling schemes described above (pictured in Figure 2).

In order to collect a large number of labels quickly we

	Floor plans	Circuits	Family trees	Geometry	Total
Number of Sketches	127	360	70	3055	3612
Number of Strokes	11261	4349	1990	5797	23397
Number of users	23	10	10	27	70

Table 2: The composition of the full (unlabeled) dataset.



(a) Best in context

(b) Is the stroke a...

(c) Can it be a...

(d) Best in isolation

Figure 2: The four labeling interfaces

implemented the interfaces as web-based Java applets and invited users to spend a few minutes labeling strokes in exchange for being entered into a drawing each time they completed a session. The participants were largely from the MIT community, although it was open to anyone with a Java-compatible web browser. Participants generally did not have any experience with sketch recognition research. We avoided labeling the data ourselves because we didn't want to bias the labeling with our own definitions of the classes, which have evolved in parallel with our recognizers.

While designing the interfaces we wanted to avoid order effects. For example, when being shown strokes in isolation, after seeing several carefully drawn lines, a slightly curvy line may be more likely to be classified as an arc due to the contrast between the previous strokes. To avoid the problem the interface randomly selects strokes from a pool of all strokes that are selected across different sketches, authors, and data sets.

The four desired label sets fell into two categories: single label and multiple label.

Collecting the Single Label Sets: *Best* and *Context* These two interfaces both asked the user to choose the best interpretation for a stroke. The interface for the *Best* label set presented the stroke in isolation. The *Context* interface presented the stroke in the context of the original sketch by coloring it in red while the remainder of the sketch was rendered in blue. Unlike the other conditions that randomized the stroke ordering, the *Context* interface presented strokes in the order they were drawn to to avoid having to switch back and forth between sketches. These two interfaces also allowed the user to assign the label *other* if the stroke did not naturally fit into any category. We decided to include this category so that users would not randomly assign labels to strokes that did not fit into any of the categories.

Collecting the Multiple Label Sets: *IsA* and *CanBeA* To collect these two data sets we viewed the classification as a binary decision for each label. To capture this we showed labelers one stroke in isolation and asked them if it belonged in a particular category (line, arc, etc.). To avoid ordering effects the labeling process was organized by the label categories instead of by strokes. This meant that the labelers were asked to classify strokes against a specific category and were then presented with a sequence of five strokes. The category was then changed and they were presented another set of strokes. For example, they were asked if five strokes were lines and then asked if the next five strokes were arcs. This process repeated until all of the strokes were evaluated for each category. These two interfaces varied only in the instructions presented to the user; "is it a…" versus "can this possibly be a…."

Generating Label Sets From the Raw Label Data

After collecting the labels for the strokes we compiled the results to produce the four label sets described above. For the sets with single interpretations, *Best* and *Context*, this was a straightforward task, we simply included the strokes that both labelers agreed on.

For the *IsA* and *CanBeA* label sets, the situation was complicated by the presence of multiple labels per stroke. When labeling the data, each stroke were presented and judged independently. Accordingly we chose to include stroke labels in the final dataset on a per label basis because it maintained the same semantics of the labels and was analogous to the criterion for the single label cases. For example, if one labeler labeled a stroke as a line and an arc but the other labeler labeled it as a line, we included the stroke in the label set but only with the line label.

For the multiple label data sets we considered, and rejected, the criterion of including all of the labels that were assigned by any labeler. Although this criterion captures the idea of including a wider range of possible interpretations, it does not account for errors made by the labelers and therefore fails to ensure the accuracy of the labels.

One drawback to the agreement based inclusion of strokes is that it prefers strokes that are less ambiguous. It is possible that these strokes will be easier for recognizers to classify and will not be representative of the range of strokes present in sketches. To address this problem, we are considering other criterion for including labels in the final label sets. One option, that could be applied to all of the label sets, is to have more than two labelers for each stroke and include labels which were agreed upon at least some percentage of the time. Alternatively, the number of votes for each label could be used to weight or rank the labels assigned to a stroke. Our label collection is ongoing and when we obtain sufficient data to experiment with these different criteria we will be able to more thoroughly evaluate the suitability of each of these options.

The number of strokes, to date, included from each condition are listed in Table 1.

Discussion

Here we present lessons we learned about the nature of the sketches from different domains, the differences between the label sets, and ways to improve the final labeled corpus.

Observations from the Data and Label Sets

The labeling data revealed an important lesson about the different classes of sketches that we collected. The strokes from the geometry data set were generally easier to label. There was a higher level of agreement between labelers on strokes from the geometry data than from the circuit and family tree domains (shown in Table 3). This indicates that they were less ambiguous. The Kappa value is a measure of agreement that takes into account how likely two labelers are to agree by chance (Carletta 1996). We hypothesize that this difference between conditions is attributable to the fact that the strokes were drawn outside of any realistic context and therefore the person drawing was focused on producing careful clear figures. The fact that people perceived these strokes differently serves as a warning that training and test data should be selected from data that is collected from similar contexts to the one that will be used in practice.

While we provided some guidance above about how to choose a label set for different purposes, we did not discuss

	%Agreement	Kappa
Geometry	90.4	0.86
Family Trees	80.1	0.73
Circuits	82.0	0.71

Table 3: Agreement between labelers using the labels from the *Best* labeling condition and on a per study basis.

	Labeled Strokes		Final Set of Labeled Strokes	
	Total # strokes	**# with multiple labels**	**Total # strokes**	**# with multiple labels**
IsA	186	45 (24.2%)	162	6 (3.7%)
CanBeA	175	82 (46.9%)	154	19 (12.3%)

Table 4: The number of strokes with multiple labels as compared with the total number of strokes in each category

the differences between the *CanBeA* and *IsA* label sets. After analyzing the current state of the two data sets we have determined that the *CanBeA* set should be used instead of the *IsA* data set. After compiling the final label sets with our current, agreement based, criteria for which labels to include, we found that in the *IsA* label set only 3.7% of strokes meeting the agreement criteria had more than one label. This was somewhat surprising because before applying the criterion 24.2% of all the strokes labeled in the *IsA* condition had received more than one label by at least one of the two labelers. This means that although labelers generally agreed on at least one label they rarely agreed on a second label. The results from the *CanBeA* case are higher with 12.3% of the labels meeting the inclusion criterion having multiple labels and 46.9% of the total number of labeled strokes. We believe that these percentages (summarized in Table 4), especially for the *IsA* case, are artificially low as a result of requiring unanimous agreement for a label to be included. We plan to reevaluate this label set when more labelers have evaluated each stroke and we have experimented with other label inclusion criteria. However, visual inspection of the stroke labellings included in the current *CanBeA* label set suggests that they capture a reasonable amount of variation without being overly liberal with label assignments, and can therefore be used for cases needing multiple stroke labels.

One of our goals in collecting more label data is to increase the number of strokes that are labeled in all four conditions. In the current corpus there are a relatively small numbers of such strokes because we selected strokes randomly from a pool to avoid order effects in the labeling process. However, our initial pool of strokes was too large and we did not get a dense labeling of the space and therefore there are less strokes that appear in all the label sets than we had hopped. Having a more dense labeling of strokes would allow more complete comparisons between the different label sets.

Qualitative Observations

In addition to providing a test corpus, the stroke and label data tells us how people make free sketches. Based on our current data sets and our observations of people using our systems over the past couple of years, we have observed several phenomena that contradict assumptions made by designers of some interactive sketching systems:

- *Observation #1: Users do not always draw each object with a sequence of consecutive strokes.* We observed numerous sketches in which the user drew part of an object, left it unfinished, drew a second object, and only then returned to finish the first.
- *Observation #2: Users drew more than one object using a single stroke.* In circuit diagrams, for example, users often drew several resistors, wires, and even voltage sources (circles) all with a single pen stroke.
- *Observation #3: Erasing whole strokes instead of individual pixels affects drawing style.* Some users were surprised that the eraser removed an entire stroke instead of just part of it. They mentioned that they learned to compensate for this limitation by drawing shorter strokes.
- *Observation #4: Users draw differently when using an interactive system than when freely sketching.* We informally observed that many users drew more precisely when using an interactive system that displayed recognition feedback, than when using a system that performed no recognition.

Related Work

We are unaware of any publicly available corpus of categorized and labeled sketch data. Other work relating to the analysis of sketches has been done in analyzing the relationships between sketching and a number of cognitive processes. Tversky studies the importance of sketching in the early stages of design (Tversky 2002). Kavakli et al. investigate style differences between novice and expert architects (Kavakli & Gero 2002) and between designers performing different tasks, including tasks at different stages of the design process (Kavakli, Scrivener, & Ball 1998). These studies have provided us insight into the types of variations we might expect and have suggested key variables to record and analyze. It would be worthwhile to reproduce these types of studies to see how the results vary with the use of digital ink and to see if the differences between different tasks and users can be more quantitatively characterized.

Conclusion

We have created the ETCHA Sketches database that, to date, contains sketches from four domains and four different sets of labels which can be used for different evaluation and training tasks. We have presented some of our insights into both the process of collecting and labeling sketches and some properties of free sketches.

The ETCHA Sketches database is publicly available to the community through our group's web page: http://rationale.csail.mit.edu/ETCHASketches. In the future we intend to provide a more powerful web based interface to facilitate the addition of new sketches and the selective retrieval and browsing of the data sets. As a work in progress, the datasets will continue to grow and cover more domains. Researchers at other institutions with similar classes of datasets or even just raw stroke data are encouraged to contact the authors to incorporate their data into the database.

References

Carletta, J. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics* 22(2):249–254.

Davis, R.; Landay, J.; and Stahovich, T., eds. 2002. *Sketch Understanding*. AAAI Press.

Hong, J.; Landay, J.; Long, A. C.; and Mankoff, J. 2002. Sketch recognizers from the end-user's, the designer's, and the programmer's perspective. *Sketch Understanding, Papers from the 2002 AAAI Spring Symposium* 73–77.

Kavakli, M., and Gero, J. S. 2002. The structure of concurrent cognitive actions: A case study on novice and expert designers. *Design Studies* 23(1):25–40.

Kavakli, M.; Scrivener, S. A. R.; and Ball, L. J. 1998. Structure in idea sketching behaviour. *Design Studies* 19(4):485–517.

Linguistic Data Consortium. 2004. LDC–linguistic data consortium. http://www.ldc.upenn.edu.

Tversky, B. 2002. What do sketches say about thinking? *Sketch Understanding, Papers from the 2002 AAAI Spring Symposium* 148–151.