Combining Speech and Sketch to Interpret Unconstrained Descriptions of Mechanical Devices*

David Bischel, Thomas Stahovich, and **Eric Peterson**
Department of Mechanical Engineering
University of California
Riverside, CA 92521 USA
{dbischel, stahov, epeterson}@engr.ucr.edu

Randall Davis and **Aaron Adler**
Computer Science & Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
{davis, cadlerun}@csail.mit.edu

Abstract

Mechanical design tools would be considerably more useful if we could interact with them in the way that human designers communicate design ideas to one another, i.e., using crude sketches and informal speech. Those crude sketches frequently contain pen strokes of two different sorts, one type portraying device structure, the other denoting gestures, such as arrows used to indicate motion. We report here on techniques we developed that use information from both sketch and speech to distinguish gesture strokes from non-gestures — a critical first step in understanding a sketch of a device. We collected and analyzed unconstrained device descriptions, which revealed six common types of gestures. Guided by this knowledge, we developed a classifier that uses both sketch and speech features to distinguish gesture strokes from non-gestures. Experiments with our techniques indicate that the sketch and speech modalities alone produce equivalent classification accuracy, but combining them produces higher accuracy.

1 Introduction

Traditional mechanical design tools hinder creativity in the early phases of design, in part because they require over-specification of details before they are relevant, and because they typically have clumsy interfaces. Such tools would be considerably more useful if we could interact with them in the way that human designers communicate ideas to one another, i.e., using crude sketches and informal speech. Both the sketch and speech are essential to such communication; typically one cannot be understood without the other. Interpreting crude sketches is challenging in part because of the wide variety of information they contain. While many of the pen strokes are used to portray device structure, others denote gestures, such as arrows used to indicate motion, or circles used to single out a component being discussed. Figure 1 shows sketches used in describing a bolt cutter and a portion of an air pump. Consider the challenge such drawings pose for any sketch-understanding software.

*The authors gratefully acknowledge support for this work provided by the National Science Foundation via award No. 0729422.

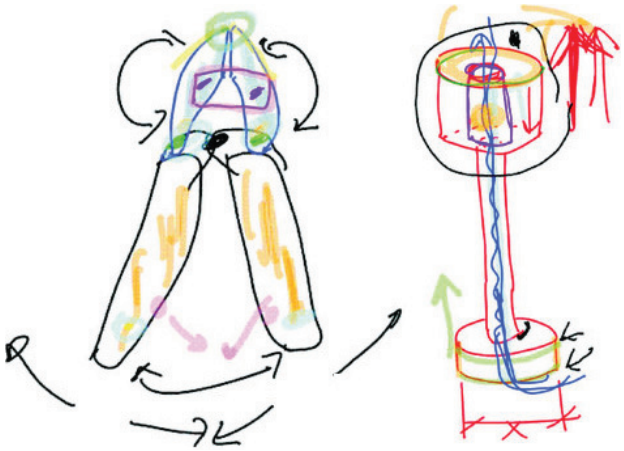

Figure 1: Sketch of bolt cutter (left) and air pump (right).

A critical first step in understanding this sort of sketch is distinguishing gesture strokes from non-gestures. This is difficult for several reasons. First, there is no standard set of gesture shapes, in contrast to domains such as circuit schematics. Second, gesture strokes and geometry strokes often have the same shape. In Figure 1, for example, blobs of ink are used both as pivots (in the bolt cutter) and as selection gestures. Third, the meaning of a stroke can often be determined only from the accompanying speech.

We report here on our efforts to identify and calibrate the discriminatory power available from a variety of sketch and speech features, describing the features we examined and the contribution each made to the gesture/non-gesture classifier we developed. We began by collecting and analyzing a large set of unconstrained device descriptions. Analysis revealed six common types of gestures that are used either to illustrate behavior or to provide spatial context for a part of the description. Guided by this knowledge, we developed a classifier that uses features of both the sketch and speech to distinguish gesture strokes from non-gesture strokes. The sketch features compute geometric properties of the strokes, and the spatial and temporal relationships between them. The speech features compute statistical properties of word groups temporally aligned with the strokes. Experimental evaluation of our techniques indicates that the sketch and speech modalities alone produce roughly equivalent classification accuracy, but combining them produces higher accuracy.

2 Related Work

Our work is grounded in insights about how people use multimodal explanations to describe devices. Ullman [1990] found that engineers commonly use five different categories of pen strokes in a sketch. His "support" and "draw" strokes are analogous to our categories of gestures and non-gestures. Heiser [2006] concluded that when there are numerous arrow gestures in a sketch, students can more easily understand the functionality of a device, illustrating the importance of gestures in a design sketch.

Gestures are typically understood in the context of accompanying speech. Oviatt [1997] studied humans interacting with dynamic mapping software, quantifying the likelihood that speaking or sketching would occur first, or that they would start simultaneously. This work was extended by Adler [2007] for design descriptions, who found consistent time delay patterns between when a pen stroke was drawn and when the related speech was spoken. Our approach builds on this, and assumes that speaking and sketching about the same ideas exhibit a strong temporal proximity.

Much of the previous work in understanding descriptions of mechanical devices has focused solely on sketching of structure (e.g., [Bloomenthal *et al.*, 1998; Masry *et al.*, 2005]). Several multimedia systems have integrated a limited vocabulary of speech with sketching, e.g., Quickset [Cohen *et al.*, 1997], where spoken words are used as simple menu commands, and MATCH [Johnston *et al.*, 2001], which combines the results at a semantic level, where our system integrates speech and sketch features at a much earlier phase of recognition. GIDeS++ [Silva and Cardoso, 2004] is a multimodal system specifically designed to understand descriptions of mechanical devices, but it uses pen strokes to replace mouse functionality rather than attempting to maintain a natural sketching environment. ASSISTANCE [Oltmans, 2000] incorporates spoken behavioral descriptions to supplement the understanding of mechanical device sketches, but relies on limited vocabularies of speech patterns that must be explicitly identified in advance, where our system can adapt to new patterns via user-provided training data.

Our work is related to the work of Patel [2007] and Bishop [2004] on separating text strokes from non-text strokes. These works differ from ours in considering only features from the sketch, where we examine the accompanying speech. Additionally, in their work text consists of a consistent set of letter and number glyphs, where the gestures in our domain are often unique, and frequently have the same shapes as pen strokes intended as non-gestures.

Our classifier builds on work in shape recognition by using the kinds of features used by feature-based recognizers, as for example in [Rubine, 1991; Patel *et al.*, 2007]. Our system relies on some of the features these systems use, but also extracts new features to address the special nature of identifying free-form gestures.

In examining properties of the accompanying speech, our system does not to try to understand it, but rather simply classify it as that which accompanies either a gesture or a non-gesture. We do this with Bayesian Filters (as in [Graham, 2004]) and Markovian Filters (as in [Yerazunis, 2004]).

3 Experimental Design

We conducted a study to observe how people naturally describe devices to one another, using four devices: a door lock, an air pump for inflating balls, a bolt cutter, and a pair of C-clamp vise grip pliers. These devices were selected because they are planar, include both standard machine components and free-form shapes, and exhibit a variety of behaviors, i.e., both mechanical and fluidic. Participants were given five minutes to describe the lock and bolt cutters, and 10 minutes for the pump and vise grips.

The participants were 16 graduate and senior undergraduate mechanical engineering students at UC Riverside. Each study session involved a pair of participants placed in separate rooms and allowed to communicate using Tablet PCs, microphones, and headphones. The tablets provided a shared drawing environment with a pen, highlighter, and eraser, and the ability to select from several ink colors. The audio and drawing were recorded with timestamps.

One participant was asked to describe a device to his or her partner, who could ask clarifying questions. At the end of the description, both participants were asked survey questions about the structure and behavior of the device. To motivate effective dialog, the participants were informed that their compensation would be based on the accuracy of their answers. (All participants were in fact given the maximum compensation.) The two participants repeated this process three times, switching roles, so that each participant described two devices.

A participant's first description often lacked detail necessary to understand how the device operated. The survey questions provided useful feedback on this for both participants. Each participant was given the opportunity to repeat their first description after receiving feedback on the accuracy of their survey answers. As a result, a total of 49 device descriptions were collected.

4 Analysis of Data

We manually transcribed the speech from the study, then used SPHINX [Huang *et al.*, 1993] to align the text with the recorded audio to find timestamps for the words. The words were also labeled with the identity of the speaker. Each pen stroke was manually labeled as either a gesture or non-gesture stroke, where non-gestures included pen strokes representing both device structure and hand-written text. (Our future work will consider text as a separate class.)

We examined the sketches to identify the vocabulary of gestures used for explaining devices. Despite the large variation in drawing styles, participants were quite consistent in the types of gestures they used. We observed two functional categories of gestures: "selection gestures," used to relate a spoken description to a spatial location in the sketch, and "motion gestures," used to give spatial context to how things move or interact. Examples of many of the gesture types are contained in Figure 1.

Selection gestures were the most frequent type. Participants selected objects by circling them, tapping the stylus on them, highlighting them, tracing them, or pointing at them with an arrow. Circling gestures were typically drawn with

a circle or rectangle; participants often selected a new ink color before making the gesture. Tapping gestures consist of repeatedly tapping the stylus at a particular location in the sketch, which produces a tight grouping of short bits of ink. Participants typically did not change ink color before tapping. The highlighting gesture consists of filling in the interior of a part using repeated strokes with either the pen or highlighter. This form of selection gesture was often used in regions with many overlapping parts, where distinguishing one part from another was difficult. When making a tracing gesture, participants typically traced the boundary of a part using a color that differed from that of the part itself. This form of gesture was used both to select entire parts and to emphasize important locations on a part. Two types of arrows were used for selection: single-headed arrows were used to point to objects of interest, while double-headed arrows were often used to indicate that two pieces of the sketch were actually two views of the same object.

Motion gestures always involved some form of arrow. Straight arrows were used to indicate translation, curved arrows were used to indicate rotation. Double headed arrows were used to indicate that parts could move back and forth. Clusters of arrows pointing in the same general direction were often indicative of a fluid flow. Motion arrows were most frequently drawn alongside a part rather than over the top of it, so as to avoid obscuring the part's geometry.

5 Classifier Design

Our classifier is a neural network with two hidden layers, with three nodes in the first hidden layer, two in the second, and one in the output layer. Inputs to the network are the sketch and speech features described below; the output is the classification of the stroke as a gesture or non-gesture. We have found that having additional context for a stroke improves classification accuracy. In particular, we include in the inputs to the network the sketch and speech features of both the previous and subsequent strokes.

We linearly scaled each feature based on its ensemble average, μ, and standard deviation, σ, with a value of $\mu - 3\sigma$ mapped to 0, and a value of $\mu+3\sigma$ mapped to 1. This avoided problems stemming from the vastly different magnitudes of the raw features.

A gesture may actually be comprised of several pen strokes. For example, an arrow may have one stroke for the shaft and another for the head. Our classifier is intended to identify both strokes as gesture strokes. A subsequent process would be required to combine those strokes to form a single object, but that is beyond the scope of this paper.

5.1 Sketch Features

Guided by our analysis of the common gesture types observed in the user study, we developed a set of geometric features to help distinguish gestures from non-gestures. These features consider the properties of individual strokes, and the spatial and temporal relationships between the strokes. The complete set of features is listed in Table 1.

The first six features concern individual strokes. D_{SL} is the length of the pen stroke, and D_{SED} is the distance between its first and last points. Taken together, these features can indicate if a stroke forms a closed shape, potentially representing a circling gesture. D_{AC} is the sum of the absolute value of the curvature (defined as the acute angle between the two line segments formed between a point and the points on either side) along a stroke, and provides a measure of how much the curve "wiggles," or deviates from a straight line. D_{DC} is similar to curvature, but is biased toward diagonal drawing directions. The ink density, D_{ID}, is a measure of the compactness of the stroke and is defined as the ratio of D_{SL}^2 to the area of the stroke's minimum, rotated bounding box. D_{SL} is squared so that the stroke length scales like area. A large value of D_{ID} could indicate a highlighting gesture. The highlighter feature, D_{HL}, has a value of one if the stroke was made with a highlighter, and zero otherwise.

The remaining 10 features describe the temporal and spatial relationships between strokes. D_{DPS} and D_{DNS} are the distance to the previous and next strokes, respectively. D_{TPS} and D_{TNS} are the time between the stroke and the previous and next strokes. D_{TCS} is the time between the stroke and the closest previously drawn stroke. These five features give a measure of the spatial and temporal isolation of a stroke. D_{ET} is the total elapsed time. Strokes drawn later in the sketch may be more likely to be gestures. The underlying color similarity, D_{UCS}, measures the extent to which earlier nearby strokes have the same color as the stroke. This feature is computed by first constructing a minimum, axis-aligned bounding box. This box is then expanded by a threshold, and all earlier strokes are clipped, so that only the earlier ink inside the bounding box is retained. D_{UCS} is defined as the fraction of that ink that has the same color as the stroke. If a stroke's color is different from that of the surrounding strokes, it may be a gesture stroke. Underlying ink density, D_{UID}, is the density of the ink in the bounding box used to compute D_{UCS}. If a stroke is drawn over a dense region, that stroke may be a gesture.

The two Hausdorff features [Kara and Stahovich, 2005] measure the extent to which a stroke traces underlying strokes. For each point on the stroke, we determine the closest distance to a point on another stroke. D_{MHD} is the maximum of these minimum distances. D_{AHD} is computed similarly, except that we compute the average minimum distance. As in [Kara and Stahovich, 2005], we use a distance map to enable efficient computation of the Hausdorff features, and use Manhattan rather than Euclidean distances.

5.2 Speech Features

To compute the speech features, it is first necessary to determine which words are associated with each pen stroke. Both [Adler and Davis, 2007] and [Oviatt *et al.*, 1997] suggest there is a strong temporal correlation between speaking and drawing. Based on that insight, we define a temporal window that extends a time ΔT before and after the stroke, and assume that any words that fall at least partially within that window are associated with the stroke. It is possible that a word may be associated with more than one stroke, or that a stroke may have no words associated with it. We chose a value of 3 sec for ΔT (a value consistent with the results in [Oviatt *et al.*, 1997]), although as described in Section 7, future work will be aimed at creating improved methods of

Name	Description
D_{SL}	Stroke length
D_{SED}	Start to end distance
D_{AC}	Total absolute curvature
D_{DC}	Diagonally-biased curvature
D_{ID}	Ink density
D_{HL}	Highlighter
D_{DPS}	Distance to the previous stroke
D_{DNS}	Distance to the next stroke
D_{TPS}	Time to the previous stroke
D_{TNS}	The time to the next stroke
D_{TCS}	Time to the closest prior stroke
D_{ET}	Total elapsed time
D_{UCS}	Underlying color similarity
D_{UID}	Underlying ink density
D_{MHD}	Max. Hausdorff distance to underlying ink
D_{AHD}	Ave. Hausdorff distance to underlying ink
W_{TPS}	Time to previous speaker
W_{WC}	# of word in temporal window
W_{BF}	Bayesian filter
W_{TBF}	Thesaurus Bayesian filter
W_{MF}	Markovian filter

Table 1: Features: D_x = sketch (drawing) feature; W_x = speech (word) feature.

defining the temporal window.

The first speech feature is the time to the previous speaker, W_{TPS}. A change in speaker may indicate that a question is being asked or answered. W_{TPS} is capped at 30 sec as the ability to make inferences decreases with longer intervals. W_{WC} is the number of words in the temporal window. The greater the number of words, the more reliable the information extracted from the speech. The other speech features concern the words themselves. Understanding grammatically correct speech is difficult enough; our speech is ungrammatical, filled with pauses, repetitions, and disfluencies like "um" and "ah." Trying to perform semantic analysis on such ungrammatical text is intractable at present. Instead, we create features using statistical models that attempt to predict whether a set of words corresponds to a gesture or non-gesture.

The first statistical speech feature, W_{BF}, is based on a Bayesian filter, a form of naïve Bayesian classifier that has had some success in spam recognition [Graham, 2004], even though spammers actively attempt to defeat the technology. In our domain, the designer's speech and sketch are intended to compliment one another, with no intentional misdirection.

To construct the Bayesian filter, it is necessary to learn the conditional probability that a stroke is a gesture, given a specific word, w_i. $p_i = Pr\left(Gesture \mid w_i\right)$ can be computed using Bayes' Theorem:

$$p_i = \frac{Pr\left(w_i \mid Gesture\right) \cdot Pr\left(Gesture\right)}{Pr\left(w_i\right)} \tag{1}$$

where $Pr\left(w_i \mid Gesture\right)$ is the conditional probability that word w_i will be observed, given that a gesture stroke is observed; $Pr\left(Gesture\right)$ is the prior probability of observing a gesture; and $Pr\left(w_i\right)$ is the prior probability of observing word w_i. To estimate the values of these three quantities from the training corpus, we first define g_i and n_i as the number of times w_i appeared in the training corpus with a gesture or non-gesture, respectively. Likewise, g_{Tot} and n_{Tot} are defined as the total number of words associated with gestures and non-gestures, respectively. The conditional probability of observing a gesture given w_i can now be computed as:

$$p_i = \frac{\left(\frac{g_i}{g_{Tot}}\right) \cdot \left(\frac{g_{Tot}}{n_{Tot}+g_{Tot}}\right)}{\left(\frac{n_i+g_i}{n_{Tot}+g_{Tot}}\right)} = \frac{g_i}{\left(n_i+g_i\right)} \tag{2}$$

Strokes typically have multiple words in their temporal windows. To classify a stroke, it is necessary to combine the probabilities associated with each of those words. Using Bayes' Theorem and assuming the words are independent events, we obtain:

$$W_{BF} = \frac{p_1 p_2 \ldots p_c}{p_1 p_2 \ldots p_c + \left(1-p_1\right)\left(1-p_2\right)\ldots\left(1-p_c\right)} \tag{3}$$

where c is the number of words in the temporal window.

A condition probability, p_i, of 50% provides no information about the classification of a stroke. Accordingly, we define the relevance of a word as the deviation of its conditional probability from 50%:

$$Relevance\left(p_i\right) = \left|0.5 - p_i\right| \tag{4}$$

When computing W_{BF} with Equation 3, we first rank all of the words in the stroke's temporal window from high to low relevance, and then use only the top ten words for calculating the combined probability. Additionally, we bound the values of p_i between 1% and 99% to prevent individual words from dominating the combined probability in Equation 3. Finally, we exclude from consideration any words that appear fewer than five times in the training corpus, as their small sample sizes result in imprecise probability estimates.

In our study, we observed that a varied vocabulary was used to describe the same objects and gestures. If the Bayesian filter encounters a word that was not in the training corpus, it will be unable to produce a probability. However, synonyms of such words may be in the training corpus, and these could be used to estimate probabilities. This is the insight behind our Thesaurus Bayesian filter feature, W_{TBF}. This feature is similar to the Bayesian filter feature, except that as a preprocessing step before learning probabilities, the training corpus is expanded by inserting the synonyms for each original word. The value of W_{TBF} is computed just as in Equation 3. However, when retrieving the value of p_i for a particular word w_i, we first create a list containing that word and all of its synonyms. We then retrieve the conditional probability, p_j, for each word in the list. p_i is assigned the most relevant value of p_j, where relevance is defined by Equation 4.

Our Markovian filter feature, M_{MF}, considers word sequences. The string of words in the temporal window is first expanded into all possible substrings of length five or less. Then each of these substrings is further expanded into all possible strings in which one or more of the interior words

Feature	Weight
This	1
This is	4
This $\langle skip \rangle$ a	4
This is a	16
This $\langle skip \rangle$ a piston	16
This is $\langle skip \rangle$ piston	16
This $\langle skip \rangle$ $\langle skip \rangle$ piston	4
This is a piston	64

Table 2: Word sequences and weights

is replaced by a wildcard. For example, Table 2 shows the possible substrings of length four of "This is a piston".

$Pr\,(Gesture \mid s_i)$, the conditional probability that a stroke is a gesture given word sequence s_i, is computed using a modified version of Bayes' Theorem. The derivation begins with an equation analogous to Equation 2, then after algebraic manipulation and the introduction of a weighting factor, produces:

$$P_i = 0.5 + \frac{(G_i - N_i) \times W_i}{2 \times (G_i + N_i) \times W_{Max}} \quad (5)$$

where G_i and N_i are the number of times s_i appeared in the training corpus with a gesture or non-gesture, respectively. W_i is a weight that depends on the length of the sequence. The value is computed as $W_i = 2^{2(h_i-1)}$, where h_i is the number of words in s_i excluding wildcards. W_{Max} is the maximum weight across all sequences. Weighting the probability in this fashion biases the result so that shorter sequences of words are scaled to neutral probabilities (i.e., 50%), while probabilities for longer sequences are left unmodified.

6 Results

We used a form of holdout-validation to evaluate the accuracy of our classifier.[1] Each of the four holdout sets was comprised of 39 randomly selected sketches for training, and 10 for testing. The four holdout sets thus generated are named sets "A" through "D". For each of these sets, we used a beam search process, with a beam width of 10, to determine which sets of features are the most effective at classification. To begin, all possible single-feature classifiers were trained and then evaluated on the test data. The 10 most accurate classifiers were then expanded to produce a set of two-feature classifiers. The 10 best of these were then expanded to produce three-feature classifiers, and so on. To provide additional insights about which features are the most important, this process was performed three times for each holdout set: once considering only sketch features, once considering only speech features, and once considering both. The results are shown in Figure 2 and Table 3.

The best sketch-only classifiers used between 6 and 9 features and achieved classification accuracy ranging from 75.1% to 78.0%. The most important features varied from one holdout set to another. For example, the best single-feature classifiers for the four holdout sets variously used

[1]In the future, we will use leave-one-out cross-validation, but that computation was too time-consuming for our current experiment.

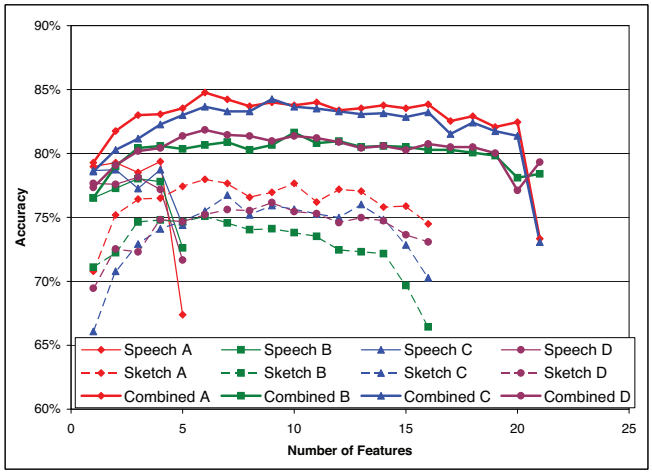

Figure 2: Accuracy vs. # of features for sketch-only, speech-only, and all-feature classifiers.

D_{DNS}, D_{DPS}, D_{TCS}, and D_{TPS}. Generally, the best classifiers with few features employed features that give spatial and temporal proximity, such as distances and times to the next, previous, or closest stroke. The best classifiers with three or less features selected from among only D_{DNS}, D_{DPS}, D_{TNS}, D_{TPS}, D_{TCS}, D_{UID}, D_{MHD}, D_{AC}, and D_{ET}.

The best speech-only classifiers used 2 to 4 features, and achieved accuracy between 78.0% and 79.4%. However, nearly the same results could be obtained using only a single speech feature, the Bayesian filter for some sets and the Thesaurus Bayesian filter for others. These single-feature classifiers achieved between 76.5% and 79.0% accuracy.

With all features considered, the best classifiers used between 6 and 10 features, and achieved classification accuracy ranging from 81.6% to 84.8%. The best single-feature classifier used one of the Bayesian filters. Classifiers with a few more features typically employed a Bayesian filter feature along with the sorts of features used in the best low-feature-count, speech-only classifiers.

	Type	Best Feature Set	Acc.
Set A	Sketch	D_{ID} D_{DPS} D_{TPS} D_{TNS} D_{ET} D_{UCS}	78%
	Speech	W_{WC} W_{BF} W_{TBF} W_{MF}	79%
	All	D_{ID} D_{DPS} D_{DNS} D_{TPS} D_{ET} W_{BF}	84%
Set B	Sketch	D_{SL} D_{HL} D_{DNS} D_{TPS} D_{TNS} D_{TCS}	75%
	Speech	W_{TPS} W_{BF} W_{TBF}	78%
	All	D_{SL} D_{HL} D_{DPS} D_{DNS} D_{TPS} D_{UCS} D_{UID} W_{BF} W_{TPS} W_{MF}	82%
Set C	Sketch	D_{SED} D_{ID} D_{DPS} D_{TCS} D_{ET} D_{UID} D_{AHD}	77%
	Speech	W_{WC} W_{TBF}	79%
	All	D_{DPS} D_{TPS} D_{TNS} D_{TCS} D_{UID} D_{MHD} D_{AHD} W_{WC} W_{TBF}	84%
Set D	Sketch	D_{SL} D_{AC} D_{DC} D_{HL} D_{DPS} D_{ET} D_{UID} D_{MHD} D_{AHD}	76%
	Speech	W_{WC} W_{BF} W_{TBF}	78%
	All	D_{HL} D_{TCS} D_{ET} D_{MHD} W_{TPS} W_{BF}	82%

Table 3: Best performing feature sets.

7 Conclusion

Our set of 49 sketches included 3533 strokes intended as device structure, 658 as text, and 2404 as gestures. Overall 36.5% of the strokes were gestures, and 63.5% non-gestures. Thus, a naïve classifier would achieve 63.5% accuracy. Using both sketch and speech features, our classifier achieved between 81.6% and 84.8%, depending on the holdout set. While this is significantly better than the naïve approach, the real value of this work is that it provides a measure of the information content of the sketch and speech. The accuracy using only sketch features was similar to that achieved using only speech features. Thus, these two modalities have roughly equivalent levels of content. However, that content is clearly different as is evident from the fact that combining the two modalities results in higher accuracy than either alone.

While our results are extremely encouraging, there are several areas of possible improvement. For example, we formulated our problem as a two-way classification task, but many strokes did not fit well within that structure. For instance, text comprised 10% of the pen strokes, but did not fit into either the gesture or non-gesture categories. We may be able to achieve more accurate results by considering a greater number of classes. Similarly, we use a simple approach based on a fixed time threshold to define the temporal window of speech associated with each pen stroke. Our window did not always align well with the beginning of a sentence, making it harder to detect relevant patterns in the grammatical structure. It may be possible to select the window based on pauses in drawing or speaking, repetitions, etc., by building on the work of Adler [2007] and Oviatt [1997]. There is also a need to improve our representation of the visual context of a stroke. We currently use a variety of contextual features, such at the distance to the nearest previously drawn stroke, and the color similarity of the underlying ink. Similarly, the classifier considers the features of a stroke as well as those of the subsequent and previous strokes. For objects such as dotted lines, however, this may provide inadequate visual context. Bishop's [2004] work in distinguishing text from non-text in a single-modal environment may provide a route to a solution. Finally, our analysis to determine the most important features was based on four subsets of the data. To reduce the chances of over-fitting, it would be beneficial to perform this analysis using leave-one-out validation to find the most effective features for the entire data set.

We will also be working to integrate this classifier into the broader sketch recognition process. We are interested in measuring how effective our current classifier can be when used as a pre-processing step. That is, given our current level of performance at distinguishing gestures from non-gestures, how much guidance does that provide in tackling recognition tasks as formidable at those in Figure 1?

Our system takes an early fusion approach to integrating speech and sketch information. Given the coarse nature of design sketches, effectively including speech directly in sketch recognition is critical for understanding user intent. In fact, the most surprising observation of our work is that the speech features are slightly more effective than sketch features in identifying gestures, which at first glance would appear to be a shape recognition problem.

References

[Adler and Davis, 2007] A. Adler and R. Davis. Speech and sketching: An empirical study of multimodal interaction. In *Proc. SBIM '07*, pages 83–90, 2007.

[Bishop *et al.*, 2004] C.M. Bishop, M. Svensen, and G.E. Hinton. Distinguishing text from graphics in on-line handwritten ink. In *Proc. of the Int. Workshop on Frontiers in Handwriting Recognition*, pages 142–147, 2004.

[Bloomenthal *et al.*, 1998] M. Bloomenthal, R. Zeleznik, and M. Cutts. Sketch-N-Make: Automated machining of CAD sketches. In *DETC98/CIE-5708*, pages 1–11, 1998.

[Cohen *et al.*, 1997] P.R. Cohen, M. Johnston, and D. McGee. Quickset: multimodal interaction for distributed apps. In *Proc. MULTIMEDIA*, pages 31–40, 1997.

[Graham, 2004] P. Graham. *Hackers and Painters, Big Ideas from the Computer Age*. O'Reilly, 2004.

[Heiser and Tversky, 2006] J. Heiser and B. Tversky. Arrows in comprehending and producing mechanical diagrams. *Cognitive Science*, 30(3):581–592, 2006.

[Huang *et al.*, 1993] X. Huang, F. Alleva, and H. Hon. The SPHINX-II speech recognition system: An overview. *Computer, Speech and Language*, 7:137–148, 1993.

[Johnston *et al.*, 2001] M. Johnston, S. Bangalore, and G. Vasireddy. Match: an architecture for multimodal dialogue systems. In *Proc. of ACL*, pages 376–383, 2001.

[Kara and Stahovich, 2005] L.B. Kara and T.F. Stahovich. An image-based, trainable symbol recognizer for hand-drawn sketches. *C&G*, 29(4):501–517, 2005.

[Masry *et al.*, 2005] M. Masry, D.J. Kang, and H. Lipson. A freehand sketching interface for progressive construction of 3D objects. *C&G*, 29(4):563–575, 2005.

[Oltmans, 2000] M. Oltmans. Understanding naturally conveyed explanations of device behavior. Master's thesis, MIT, 2000.

[Oviatt *et al.*, 1997] S. Oviatt, A. Deangeli, and K. Kuhn. Integration and synchronization of input modes during multimodal human-computer interaction. In *SIGCHI conf. on Human factors in comp. systems*, pages 415–422, 1997.

[Patel *et al.*, 2007] R. Patel, B. Plimmer, J. Grundy, and R. Ihaka. Ink features for diagram recognition. In *Proc. of SBIM '07*, pages 131–138, 2007.

[Rubine, 1991] D. Rubine. Specifying gestures by example. *Computer Graphics*, 25:329–337, 1991.

[Silva and Cardoso, 2004] N. Silva and T. Cardoso. GIDeS++ – using constraints to model scenes. Technical report, Information Society Technologies, 2004.

[Ullman *et al.*, 1990] D. G. Ullman, S. Wood, and D. Craig. The importance of drawing in the mechanical design process. *Computers & Graphics*, 14(2):263–274, 1990.

[Yerazunis, 2004] W.S. Yerazunis. The spam-filtering accuracy plateau at 99.9 percent accuracy and how to get past it. In *Proc. MIT Spam Conference 2004*, 2004.