# Who Dotted That 'i'? : Context Free User Differentiation through Pressure and Tilt Pen Data

Brian David Eoff*
Sketch Recognition Lab
Texas A&M University

Tracy Hammond†
Sketch Recognition Lab
Texas A&M University

**ABSTRACT**

With the proliferation of tablet PCs and multi-touch computers, collaborative input on a single sketched surface is becoming more and more prevalent. The ability to identify which user draws a specific stroke on a shared surface is widely useful in a) security/forensics research, by effectively identifying a forgery, b) sketch recognition, by providing the ability to employ user-dependent recognition algorithms on a multi-user system, and c) multi-user collaborative systems, by effectively discriminating whose stroke is whose in a complicated diagram. To ensure an adaptive user interface, we cannot expect nor require that users will self-identify nor restrict themselves to a single pen. Instead, we prefer a system that can automatically determine a stroke's owner, even when strokes by different users are drawn with the same pen, in close proximity, and near in timing. We present the results of an experiment that shows that the creator of an individual pen strokes can be determined with high accuracy, without supra-stroke context (such as timing, pen-ID, nor location), and based solely on the physical mechanics of how these strokes are drawn (specifically, pen tilt, pressure, and speed). Results from free-form drawing data, including text and doodles, but not signature data, show that our methods differentiate a single stroke (such as that of a dot of an 'i') between two users at an accuracy of 97.5% and between ten users at an accuracy of 83.5%.

**Index Terms:** H.5.2 [User Interfaces]: —

## 1 INTRODUCTION

The rise in the availability of portable Tablet PCs, as well as projector-size SmartBoards and multi-touch displays, has encouraged many people to work collaboratively on hand-drawn documents containing both hand-drawn text and images. Collaboration is commonplace for designers of creative documents, such as mechanical engineering systems, software architecture specications (UML or flowcharts), circuit diagrams and chemical compositions.

We propose that a creator of individual pen strokes – the movement of the pen from placement of the tip of the pen of the tablet surface to when the pen is lifted up – can be determined using features solely describing the physical mannerisms of how those strokes were made. The mannerisms are that of tilts of the pen, the pressure of the pen, and the speed at which the pen moves across the tablet surface. Below we discuss three domains that would be aided by being able to unobtrusively determine the creator of individual pen strokes.

A question that comes up repeatedly to the researchers upon discussion of this work, is "Why not simply assign a pen to each user? Then, user identification becomes resistant to error." While we appreciate this suggestion, it is important to acknowledge that, from a human-computer interaction or intelligent user interaction standpoint, requiring users to keep track of their pens or otherwise faithfully signify to the computer who is drawing if collaborators are sharing pens impedes interaction. We believe computers should not interrupt, constrain, nor damper a human user's creativity. Rather, a computer should instead unobtrusively infer information effective to its purposes. But more importantly, not only is requiring users to keep track of their pen inelegant, it is in some sense actually error-prone. It fails in the following point: users share pens. When users sketch collaboratively, much of what is written, is actually a means of communication. While watching users sketch, even when writing just on pen and paper, users will often place their pen down to emphasize the end of their point in a discussion. And even if the collaborator has their own pen, he or she will often pick up the pen of the last person sketching to emphasize that they are continuing the communication. In a sense in a collaborative sketching context, the pen becomes like the 'conch.' Thus, to require users to draw only with their own pen both infringes on their natural drawing style and is also impractical.

As the strokes on the page in a collaborative setting operate as a method of communication, in a collaborative sketching process, users will often draw intermittently and frequently. Because of this natural flow of ideas, it is impractical, prone-to-errors, and certainly invasive, to require users to self identify before switching users, even more so than requiring each user to keep track of their pen.

Because users are known to 1) pass pens, 2) draw intermittently and frequently, and 3) neglect to self-identify, we wanted instead to create an effective method for identifying the user using only stroke-specific context (i.e, no supra-stroke context such as timing, pen-ID, and other drawing context) involving only the physical mechanics of the pen within a single pen stroke.

Before the rise of tablet-PCs, forensic scientists have looked at user drawing features to try to identify the author of handwritten text. However, their techniques tend to focus on identifying a user across an entire document, taking the document as a whole and using the surrounding context to identify a user. This technique causes some difficulties in a digital collaborative domain where time- and/or location-adjacent strokes may alternate owners.

While we appreciate the possible benefits of using surrounding context to identify a stroke's owner, first we would like to be able to determine as best as possible a stroke's owner without context from the surrounding strokes. As a secondary step, surrounding context can be used to help distinguish ambiguous stroke owners through the use of surrounding context. Before context is relied upon we want the underlying context-free stroke user-identification algorithm to be as accurate as possible. This ensures a more accurate system once context is added.

We have found that using only the physical mechanics of a pen – specifically only the pen tilt, pressure and speed of a stroke – the computer is able to determine the author of a single stroke, as small as the dot of an 'i', with high accuracy.

The rest of this paper is organized as follows: the benefits of unobtrusive user differentiation, the previous work, a description of two experiments performed along with their results, a discussion of those results, a discussion of future work, and a brief conclusion.

*e-mail: bde@cs.tamu.edu
†e-mail:hammond@cs.tamu.edu

## 2 Benefits of Unobtrusive User Differentiation

### 2.1 Security and Verification:

Forensic documentation examination attempts to determine the creator of a hand-written document by examining the document image [8]. Traditionally, global and local stylistic features of the image are gathered from the image as a whole to determine a single owner of the entire document. However, imagine a collaborative document where two people are working on the same document both drawing and editing text and images. We would like to know who drew which stroke on the page, down to even being able to identify that user B added the stroke 'L to user As stroke 'I to change a '1' into a '4' without intruding on the user. Given the new prevalence of digitally signed and edited documents, we can imagine that checks may be signed online in the future. Given current technologies, an intruder can easily change a '1' to a '4' with the addition of a single line, which could have disastrous effects. Security of hand-written documents in our new digital world will become a significant problem, and global techinques that rely on multiple strokes will not be effective. Although this current work only describes how to distinguish between a handful of users, this is a step towards forgery identification. As it is now, it would be useful in preventing forgery in a small team setting where only a handful of users have access to a particular file.

### 2.2 Improving Sketch Recognition through User-Modelled Recognition:

Sketch recognition is the automated understanding of hand-sketched drawings by a computer. Sketch recognition algorithms generally fall under three techniques or a combination thereof: 1) Vision-based algorithms, which recognize shapes by what they look like, purely by examining the pixels on the screen and comparing the bitmap representation to a template, often through the use of neural networks or other template matching technique [11], 2) Geometry-based algorithms, which break down shapes into primitives and recognize shapes by testing perceptually important geometric features to identify the geometric shape of an object; recognition often occurs through the use of Bayesian Networks or other graphical model architectures [1][4]; and 3) Gesture-based algorithms, which recognize how shapes were drawn, purely comparing the path of a stroke against a previously existing template. Gesture-based methods generally recognize single strokes by examining a number of features based on the path of a stroke, and they recognize multi-stroke shapes through HMMs or other order-constraining algorithms [15][9][17][20]. Of the three methods, the first two (vision-based and geometry-based) are user-independent since they recognize shapes independent of user style. Gesture-based recognition, however, is highly user-dependent, as different users may have markedly different methods for drawing something even though the end-product appears the same. While gesture-based recognition has the disadvantage that it has to be trained user-to-user, it does have the advantage that when the techniques are used effectively, recognition can be fast and highly accurate. In a collaborative system when users change rapidly and without notice, a gesture-based system is often unusable unless each user is instructed to draw in a particular manner to abide by the previously trained examples. As strong proponents of natural interaction, we find that constraining users to draw in a predefined way is unacceptable and thus gesture-based recognition methods often perform poorly in a multi-user free-sketch recognition system. However, if a collaborative or multi-user system is able to automatically identify the author of a stroke, then user-dependent sketch recognition methods can be applied to that stroke. Systems can be created that use gesture-based recognition methods, but naturally conform to the user rather than having the user conform to the system. By automatically identifying the user, we can take advantage of the speed and accuracy of a single-user system, without suffering from the constrained input and/or low accuracies traditionally found in a multi-user system.

### 2.3 Collaboration:

Beyond the simple advantages of improved recognition or improved security measures, user-identification in a collaborative system can have many other uses. By determining who produced which part of a diagram, you can give credit for ideas, create a correct drawing history, and more effectively perform studies on collaboration and their effect on creativity, such as studies done by Shah et. al. [19]. Automatic documentation of each stroke's author in a sketch makes the collection of design rationale simpler in that a person looking at a previously drawn sketch can know who contributed what part of the design, and thus ask that person why they chose to draw the design in a particular way; by knowing who is the correct person to ask, the gathering of design rationale documentation becomes a more manageable task. In game playing, who produced what action is incredibly important; sketch-based games could use user-identification to create more personal, interactive and less constrained games. The capabilities of collaborative-based sketching tools are immediately improved by the automatic identification of a stroke's author. In today's world of shared Google Docs and tracked changes capabilities in Word, it would be very useful to be able to automatically label which strokes belong to which author.

## 3 Prior Work

The prior work we reviewed in creating our solution came from three fields: forensic document examination, signature verification and sketch recognition. In this section we give a brief review of the relevant literature from each field.

### 3.1 Signature Verification

Signatures have a legal binding in that they can be used to give consent. In contemporary life our signatures are used so often - credit card receipts, signing for a delivery - that the meaning of the act of scrawling our name can be lost. Automatic signature verification attempts to spot forgeries of a user's signature given a set of samples of said user's signature. This work falls into two categories: off-line, which is only concerned with the static visual record of the user's signature, and on-line, which utilizes knowledge of the dynamics of how the signature was created [13].

Signature verification has an enrollment period in which the user must provide a number of signature samples. At this point, features reflecting their signatures are calculated. These features can be global or local in nature. Global features are those that describe the full signatures (e.g., the length, the total time taken, the number of times the pen is lifted). Local features are concerned with specific portions of the signature (e.g., the total curvature on the first pen stroke). When testing a possible signature, it is verified against the model created of the user's signature. Often, this is a calculation of the features and a comparison to the enrollment samples using a distance calculation [6].

Authors have found success using a variety of techniques — dynamic time warping, hidden markov models, bayesian networks — to accomplish signature verification. Jain et. al. was able to achieve a false reject rate of 2.8% and a false accept rate of 1.6% using spatial and temporal features with a string matching approach. Dolfing et. al. was able to utilize a hidden Markov model approach and achieve error rates between 1% and 1.9% [3]. Also, in their work they used linear discriminative analysis to determine the discriminative value of various features and concluded that the most discriminative features included velocity, pressure, and pen tilt [3]. Similar results were acheived in [14]. Kawamoto et. al. found that pen tilt is able to improve the verification rate drastically [7] to an accuracy rate of 93.3%.

Researchers have debated on the use of pen pressure, tilt, and altitude features as features in successful signature verification. An approach that made use of only pen position was able to win the SVC2004 contest. This approach outperformed many that were utilizing pen pressure and pen tilt. Muramatsu and Matsumoto argue that by including pressure and tilt they were able to lower error rates from 5.79% to 3.61%[12]. While this algorithm uses similar features as those described in our work to recognize a particular signature, it requires all of the strokes of a signature to be used in concert to identify the signer's identity. It requires the user to write the same text - their signature - each time. Our work builds on ideas from this work, attempting to use similar features to recognize a single stroke out of context no matter what the user draws.

### 3.2 Forensic Document Examination

Forensics expand on the signature verification in that the field of forensic document examination deals with the process of determining the authenticity of an document, not just a signature. This could be a typed document, a printed document, a handwritten document or a mixture (an example would be forged checks). Much of the work is done by making observations between the style in which characters are created from a verified authors work and how those same marks appear in the questioned document.

In his book *Forensic Handwriting Identification*, author Ron Morris states the first principle in handwriting identification, "No two people write exactly alike [10]." In a similar vein, the second principle states "No person writes exactly the same way twice." Koppenhaver declares Albert Osborn to be the originator of forensic document examination. Osborn was famously involved in the Rice Will case, the Lindbergh kidnappings, and the examination of a forged Howard Hughes will. Forensic document examiners have been used in legal proceedings for over a century [8].

The field of forensics differs from our work in that it requires the document as a whole to be looked at to determine authenticity, not just a single stroke. Sometimes, forensic specialists even look at the paper itself to determine age of when it was written, etc. Currently, very little if any forensic work happens online, and most work is completed by a person.

### 3.3 Sketch Recognition

Currently, no research has been done in the field of sketch recognition to automatically identify a stroke's author. However, user dependent features, such as the speed of the pen have been used to recognize drawings. Christopher Herot [5] is the first person to have used pen speed to aid in the automated recognition of hand drawings. He used pen speed to help find corners in automated stroke segmentation to convert hand-drawn strokes to cleaned up polylines. The intuition was that the user would often slow down at a corner, and this information could help distinguish a polyline from an arc. This intuition is repeated in the work by Sezgin and Stahovich [18].

Rubine [15] and Long [9] also used user-specific features to recognize shapes. Although only Rubine used speed, both used other user-specific features, such as the total curvature or jitteriness of a stroke as an identifying feature of a drawn shape. Choi [2] uses user specific features to recognize shapes in a manifold learning recognition algorithm.

Additionally, user specific features have been used to recognize multi-stroke objects. For instance, Sezgin and Davis [16] found user-specific drawing orders common in multi-stroke objects and used these in HMM-based recognition.

While these recognition algorithms apply user-specific features to recognize shapes, no systems yet utilizes pen tilt nor pressure to recognize drawings. Additionally, no work has yet been done in the field of sketch recognition to automatically recognize the user from these features. But, we repeat that by automatically identifying the user, these user-specific recognition algorithms suddenly become much more user friendly.

## 4 Implementation

For the purposes of our experiments a drawing application was created in Cocoa for Mac OS X. Data was collected on a Cintiq, which provided the values for the pen tilt and pressure. This application allowed us to record the following information about the users strokes; the X and Y position of the pen, at what time the pen was in contact with the surface, the pressure of the pen on the drawing surface, and the tilt of the pen in the X and Y direction. This data was output to an XML file that allowed us to study how the user drew. Using this file we could recreate the order and the physical manner of how the user created their sample.

When the pen is held perpendicular to the tablet the tilt of the pen in the X and Y directions is 0. As the pen is tilted to the left the tilt in the X direction approaches $-90°$; as it is tilted to the right direction it approaches $90°$. As the pen is tilted to the top of the tablet the tilt in the Y direction becomes more and more negative until it is parallel to the tablet. When the pen is tilted to the bottom the values become more and more positive.

## 5 Experiment and Results

### 5.1 Experiment One

The purpose of our first experiment was to determine whether a user was consistent in his or her drawing mechanics during the course of a drawing sample, and whether those metrics were consistent over a few days. The experiment also hoped to determine if users drew differently from one another in terms of these physical drawing mechanisms to see whether they might be usable in determining the creator of an individual stroke.

Experiment One had six participants. All participants were familiar with using a tablet drawing surface. Over the course of three days, each participant was asked to use our drawing panel and to provide us a sample. The users were suggested to write a simple list of what they wished to accomplish that day. They were unconstrained as to how they should write, or how much they needed to write. Each sample contained between 100 to 200 strokes. The average pressure, as well as tilt in the X direction and tilt in the Y direction, were recorded for each stroke of the pen. The averages of these values and the standard deviation were calculated for each day. The purpose of this test was exploratory to see 1) if certain features could be used to disambiguate users, 2) see which features were more useful for disambiguating users, and 3) give insight as to how and if we could best disambiguate users using tilt and pressure.

### 5.2 Results

Figure 1 shows the data from Experiment One, from this data we can perceive two concepts: 1) Users are fairly consistent with the physical manner in which they go about sketching. During our study, we noted that some participants were more consistent than others. For instance, on a day-to-day examination, User 1 and User 6 have a fairly consistent daily average X-tilt, 2) Users' sketching mannerisms are distinct from user to user. To give us some insight about the power of disambiguation for each feature we calculated a number of t-ests shown in Figure 2 on the data from Figure 1. These t-tests compute the statistical significance (p-value) in using a single feature to compare a user with either one other user or across the entire set. Note that these t-tests are not to be used to compute actual classification or to prove that we can in fact disambiguate users; rather we present them here to provide insight as to whether or not users can be disambiguated any of the features, as well as insight as to which features might be most useful in disambiguation. For instance, the top left number in Figure 2 shows that using average pressure alone, User 1 and User 2 can be distinguished on a stroke by stroke basis with a p-value of .000382433. We note that the table also shows that the differences in average pressure between User 2 and User 6 are not statistically significant. The features that were determined to be statistically significant methods for distinguishing a set of two or six users are listed in bold. This table, in its attempt to provide us with insight for solving our problem, suggests to us that 1) there is no single 'magic' feature that can be used to disambiguate users, 2) some features seem to be better at disambiguating different users than others (for instance, the average tilt of Y would have greater success in differentiating users than the standard deviation of the tilt in Y), and 3) even poor features still provide some value (e.g., the standard deviation of the tilt of X seems to be less helpful in differentiating users than does the average tilt of Y, the data suggests it might be helpful in differentiating User 3 and User 5).

The exploratory data suggests that at least one feature can be used to help differentiate a user from 1-5 users. Looking at the data from Figure 2, we surmise that a successful algorithm would be to automatically generate a decision tree that removed possible users based on the value with the smallest p-value, and then progressively moved up the chain. (i.e., since the smallest p-value is found comparing the average pressure between User 1 and User 6, the first rule in the decision tree could be: if average pressure is above some threshold value v, then the possibles users are 1,2, 3,4,5 else 2,3,4,5,6, then progressively remove possible users based on increasing p-values using these values.) Automatically generating a decision tree for each collection of users may be a valid solution (which we did implement), we chose instead to differentiate users using a K-Nearest Neighbor classifier because in resulted in higher accuracy.

| | |
|---|---|
| 1 | Standard Deviation of Tilt in Y Direction |
| 2 | Average Tilt in Y Direction |
| 3 | Standard Deviation of Tilt in X Direction |
| 4 | Average Tilt in X Direction |
| 5 | Standard Deviation of Pressure |
| 6 | Average Pressure |
| 7 | Maximum Tilt in X Direction |
| 8 | Minimum Tilt in X Direction |
| 9 | Maximum Tilt in Y Direction |
| 10 | Minimum Tilt in Y Direction |
| 11 | Minimum Pressure |
| 12 | Maximum Pressure |
| 13 | Average Tilt in Y Direction for First Third of Stroke |
| 14 | Average Tilt in Y Direction for Second Third of Stroke |
| 15 | Average Tilt in Y Direction for Third Third of Stroke |
| 16 | Average Tilt in X Direction for First Third of Stroke |
| 17 | Average Tilt in X Direction for Second Third of Stroke |
| 18 | Average Tilt in X Direction for Third Third of Stroke |
| 19 | Average Pressure Direction for First Third of Stroke |
| 20 | Average Pressure Direction for Second Third of Stroke |
| 21 | Average Pressure Direction for Third Third of Stroke |
| 22 | Average Speed |
| 23 | Minimum Speed |
| 24 | Maximum Speed |

Figure 3: The features used by our learner to classify stroke creator.

### 5.3 Experiment Two

The purpose of Experiment Two was to determine if we could create a classifier that can accurately determine the creator of an individual stroke from the set of possible creators. Ten users participated in this study. Each user contributed three samples on three consecutive days. The data was collected using our draw panel and a Wacom Cintiq. The Cintiq was laid flat to ensure that tilt of the pen would be consistent.

Figure 3 shows the twenty-four features that were used by the classifier. We used 14 features based on tilt of the pen, seven features based on the pressure of the pen, and three based on the speed of the pen. A variety of learners were used (Linear Classifier, Quadratic Classifier, Naive Bayes Classifier, Decision Tree, Neural Network); we found our best results using K nearest neighbor. We empirically determined that a K = 10 gave the best results. Weighted distance was used to determine which of the K possible creators was the best. The stroke data collected from the participants was reduced to a tuple consisting of the 24 features and the identity of the creator. These tuples were then shuffled, and a 10-fold cross validation was performed using the K-Nearest Neighbor classifier and the complete data set. Using 10-fold cross validation, the first fold (or the first 1/10th of the data set) is used as the testing set, and the remainder of the set is used as training. On the second fold, the next 1/10th is used for testing and the remainder is used for training. This approach ensures that all strokes are used once as a testing sample.

We also experimented to determine how the size of the possible creator set influences the accuracy of our ability to determine who created the stroke. We assumed that with more possible creators the accuracy would decrease. A power set — the set of all possible sets — of the creators was created. The empty set and the sets of size one were ignored. The approach to testing these subsets was identical to testing the full set. The strokes of the creators from the power sets were reduced to tuples of the features describing the stroke. This set of tuples was shuffled, and 10-fold cross validation using K-Nearest Neighbor was performed on the set. We recorded the average accuracy for each of the set sizes (size of two to the full data set of ten creators).

### 5.4 Results

Figure 4 shows the accuracy of the classifier when attempting to differentiate two through ten different users. Our average identification rate for two collaborating users was 97.5%. Even in the unlikely case of ten users sketching simultaneously, we were still able to accurately identify the creator of an individual sketch stroke with an accuracy of 83.5%. We define accuracy as the number of strokes correctly classified divided by the total number of strokes tested. The stroke length did not drastically influence if the stroke was accurately classified. The average length of a correctly classified stroke was 57.76 px (with a standard deviation of 184.42). The average length of an incorrect stroke was longer at 76.776 px (with a standard deviation of 493.35). Using the full data set, but only testing on strokes that were less than 10 pixels in length we were able to achieve an accuracy of 72.7%. When only testing on strokes of less that 5 pixels in length we achieved an accuracy of 70.5%. Unexpectedly, when testing only on strokes of less than 2 pixels in length our accuracy was 74.8%.

Figure 6 shows a confusion matrix of the classifications of the full data set. The horizontal axis is the creator of the stroke, and the vertical axis specifies to whom the stroke was classified as being created by. The gradient represents the percentage of a user's strokes classified as belonging to some other user. A black square at the intersection of A and B implies that none of User A's strokes were classified as belonging to User B. A white square at the intersection of A and B implies that all of User A's strokes were classified as belonging to User B. A gray square at the intersection of A and B implies that some of User A's and strokes were classified as belonging to User B. Note that there was some confusion in Figure 6 between User 1 and User 7. A solid white diagonal would indicate perfect classification. User 6's drawing mannerisms were different from all the other participants, this resulted in their strokes being able to be classified with a much higher accuracy. The classifier has the most trouble differentiating User 1 from User 7. User

| User Consistency | | | | | | | |
|---|---|---|---|---|---|---|---|
| User | Study | Avg. Pressure | Std. Dev. Pressure | Avg. Tilt X | Std. Dev. Tilt X | Avg. Tilt Y | Std. Dev. Tilt Y |
| One | 1 | 0.744 | 0.237 | 0.372 | 0.036 | -0.017 | 0.045 |
| | 2 | 0.763 | 0.213 | 0.381 | 0.031 | -0.043 | 0.063 |
| | 3 | 0.785 | 0.209 | 0.382 | 0.041 | -0.056 | 0.050 |
| Two | 1 | 0.527 | 0.167 | 0.365 | 0.068 | -0.344 | 0.051 |
| | 2 | 0.484 | 0.163 | 0.390 | 0.061 | -0.376 | 0.049 |
| | 3 | 0.537 | 0.163 | 0.287 | 0.069 | -0.414 | 0.050 |
| Three | 1 | 0.543 | 0.166 | 0.469 | 0.036 | 0.001 | 0.051 |
| | 2 | 0.434 | 0.149 | 0.508 | 0.040 | -0.010 | 0.076 |
| | 3 | 0.455 | 0.148 | 0.450 | 0.036 | -0.032 | 0.048 |
| Four | 1 | 0.477 | 0.106 | 0.119 | 0.051 | -0.468 | 0.034 |
| | 2 | 0.611 | 0.179 | 0.099 | 0.088 | -0.440 | 0.094 |
| | 3 | 0.686 | 0.220 | 0.116 | 0.084 | -0.434 | 0.135 |
| Five | 1 | 0.577 | 0.179 | 0.371 | 0.031 | 0.003 | 0.053 |
| | 2 | 0.616 | 0.197 | 0.351 | 0.036 | -0.009 | 0.053 |
| | 3 | 0.582 | 0.169 | 0.354 | 0.032 | 0.007 | 0.051 |
| Six | 1 | 0.478 | 0.152 | 0.283 | 0.063 | -0.222 | 0.062 |
| | 2 | 0.491 | 0.162 | 0.279 | 0.054 | -0.185 | 0.010 |
| | 3 | 0.525 | 0.166 | 0.287 | 0.072 | -0.159 | 0.072 |

Figure 1: Results of the first experiment.

| TTest | | | | | | |
|---|---|---|---|---|---|---|
| Comparison | Avg. Pressure | Std. Dev. Pressure | Avg. Tilt X | Std. Dev. Tilt X | Avg. Tilt Y | Std. Dev. Tilt Y |
| One to Two | **0.000382433** | **0.01725204** | 0.435536174 | **0.00276649** | **0.000732615** | 0.652904734 |
| One to Three | **0.00748901** | **0.00185164** | **0.028762167** | 0.749070108 | **0.033168325** | 0.372081673 |
| One to Four | 0.101364168 | 0.344055988 | **0.000906354** | **0.087523905** | **0.002912816** | 0.33586354 |
| One to Five | **0.000530288** | **0.07274996** | 0.161615813 | 0.582815271 | 0.096144707 | 0.913494539 |
| One to Six | **0.000163231** | **0.043227175** | **0.00145259** | **0.006308128** | **0.037682283** | 0.053307125 |
| Two to Three | 0.37589914 | 0.167211712 | **0.018968748** | **0.016980818** | **0.000882276** | 0.478481896 |
| Two to Four | 0.342442855 | 0.913713798 | **0.02107247** | 0.580901878 | 0.147977218 | 0.327094283 |
| Two to Five | **0.022283365** | 0.158530774 | 0.742406549 | **0.011588526** | **0.003392672** | 0.131036555 |
| Two to Six | 0.452809327 | 0.519252708 | 0.194409873 | 0.484846961 | **0.038759737** | 0.144589726 |
| Three to Four | 0.33816071 | 0.754101059 | **0.003749464** | 0.075444953 | **0.001966234** | 0.434057263 |
| Three to Five | 0.061784583 | 0.10676146 | **0.027639824** | **0.005292259** | 0.380159676 | 0.571000982 |
| Three to Six | 0.676380942 | 0.637375503 | **0.009905556** | 0.060155884 | **0.023954531** | **0.047064071** |
| Four to Five | 0.99799043 | 0.76227634 | **0.000306463** | 0.060765947 | **0.000711033** | 0.355160416 |
| Four to Six | 0.264438949 | 0.798735826 | **0.000663408** | 0.473408359 | **0.00110406** | 0.741992126 |
| Five to Six | **0.007862188** | 0.143231149 | **0.006767433** | **0.041506763** | **0.009170439** | 0.15008821 |
| One to All | **1.30434E-07** | **0.005130715** | **0.072660735** | **0.006326478** | **0.004743238** | 0.18982183 |
| Two to All | 0.060926554 | 0.176868129 | 0.584098468 | **0.006754251** | **0.000186996** | **0.034780588** |
| Three to All | **0.042463757** | **0.036015121** | **7.65219E-05** | **0.005095152** | **0.001097271** | 0.61409626 |
| Four to All | 0.766129811 | 0.846232807 | **3.03587E-10** | 0.129921463 | **6.3525E-07** | 0.417968842 |
| Five to All | 0.508711568 | 0.522884142 | 0.25374001 | **0.000400761** | **0.000464594** | 0.081104651 |
| Six to All | **0.013786719** | 0.08553831 | 0.139329507 | 0.108151897 | 0.813892884 | 0.262198294 |

Figure 2: Results of multiple T-Test comparing users. Notice that each user's feature value is significantly different from other users using at least one feature, even with only three data points per user, and even when comparing against either one other user or all of the other users. Bolded items are statistically significant (p-value ¡ .05).

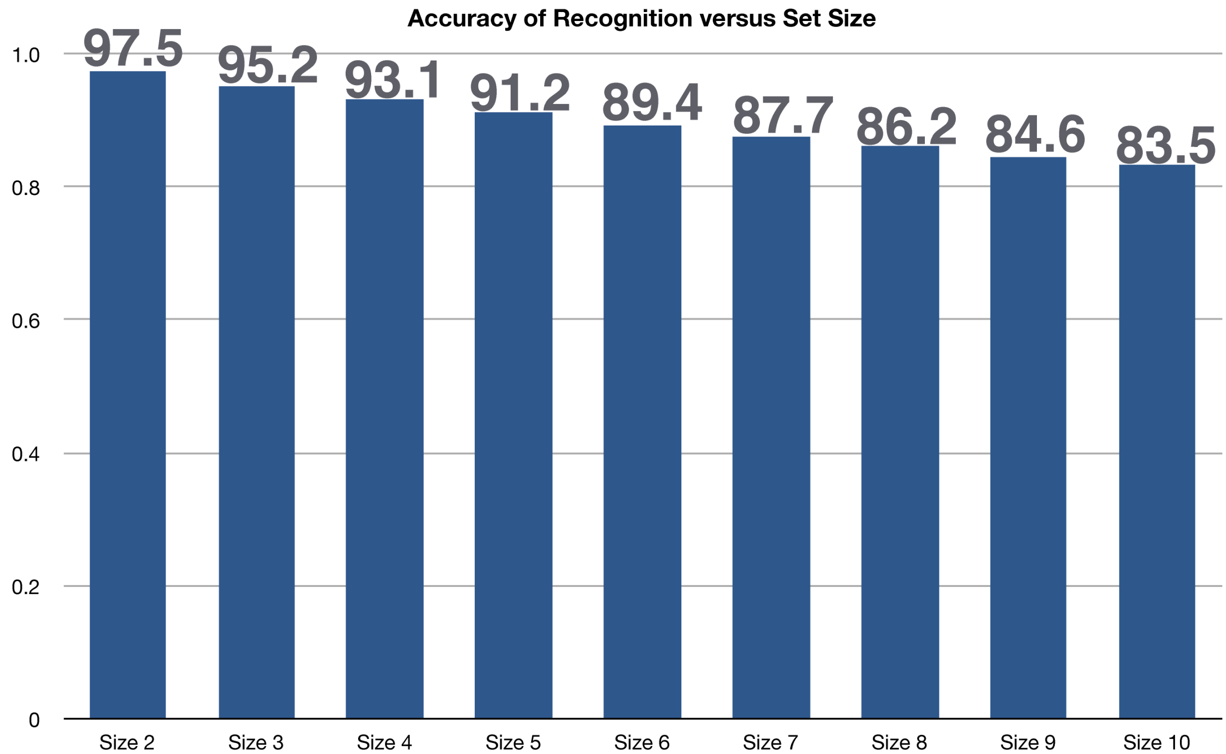


Figure 4: The accuracy of our learner on the power set. The X axis represents the set size, the Y axis is the accuracy.

7's drawing style was a midpoint between three other users, which resulted in User 7's strokes being occasionally misclassified.

The Figure 5 shows a random sample of the data collected from the users and how the strokes were classified given a grouping of two possible users, five possible users and nine possible users.

If there are only two possible users to choose from the classifier accurately identifies the creator of the stroke correctly with an accuracy of 97.5%. Figure 4 shows how accuracy decreases as the size of the set of the possible creators increases. Size and accuracy are inversely related, yet the drop in accuracy of the K-NN approach is much smaller that the drop in accuracy of a majority classifier. Also, the decrease in accuracy with the increase of possible creator set size slows.

## 6 Discussion

The results of the first experiment gave us confidence in using features based on how a user physically drew to determine the creator of individual strokes. A variety of classifiers were tested, and a K-Nearest Neighbor classifier provided the best results. With a set of ten possible creators, a classifier utilizing no information beyond how the stroke was physically drawn has an accuracy of 83.5%. At this time, we do not forsee more than ten users working on a collaborative drawing surface, and using our technique we should be able to determine the creator of individual strokes with a relatively high accuracy. This will allow the labeling of users contributions to a sketch without constraining the means in which the user draws on the surface.

One of the positive aspects of this approach is scalability. While it is true that with the addition of more possible sketchers, the accuracy does decline; however, the drop in accuracy slows. Adding four additional users, to go from a user set size of two to six, results in a 8.1% drop in accuracy. Contrast that with adding an additional four users, to go from a set size of six to ten and the drop in accuracy is 5.9%.

Certain participants (six in particular) had very distinct drawing mannerisms, which resulted in their strokes being more accurately classified. Some participants were similar to each other, but those similarities were only shared with at most three other participants. If a stroke was misclassifed, it was often attributed to a small subset of the original set — as was the case with participant seven whose strokes when misclassified were attributed to participant one, three or ten, but never to any of the six remaining participants.

Our approach uses no context information, which allowed us to do testing using the cross validation approach. Each stroke was studied in a vacuum, with no regard as to which strokes proceeded. Based on the physical metrics describing an individual stroke, our approach can accurately classify the creator of that stroke. We believe this is a baseline of the possible accuracies. Using additional context would provide higher accuracy, but this can only be verified using a new data set (in which multiple users were working on the same surface at multiple times).

## 7 Future Work

We intend to collect a new data set that would allow us to test the use of additional context information in accurately identifying the creator of a stroke. We are also experimenting using EM clustering to determine how many sketchers were involved in the creation of a sketch, and identifying their contributions. Our current approach is dependent on having training samples for all possible sketchers, and thus is not capable of determining the number of contributors without knowing the set of possible contributors. We will also experiment with different models (possibly SVM). In the future, we

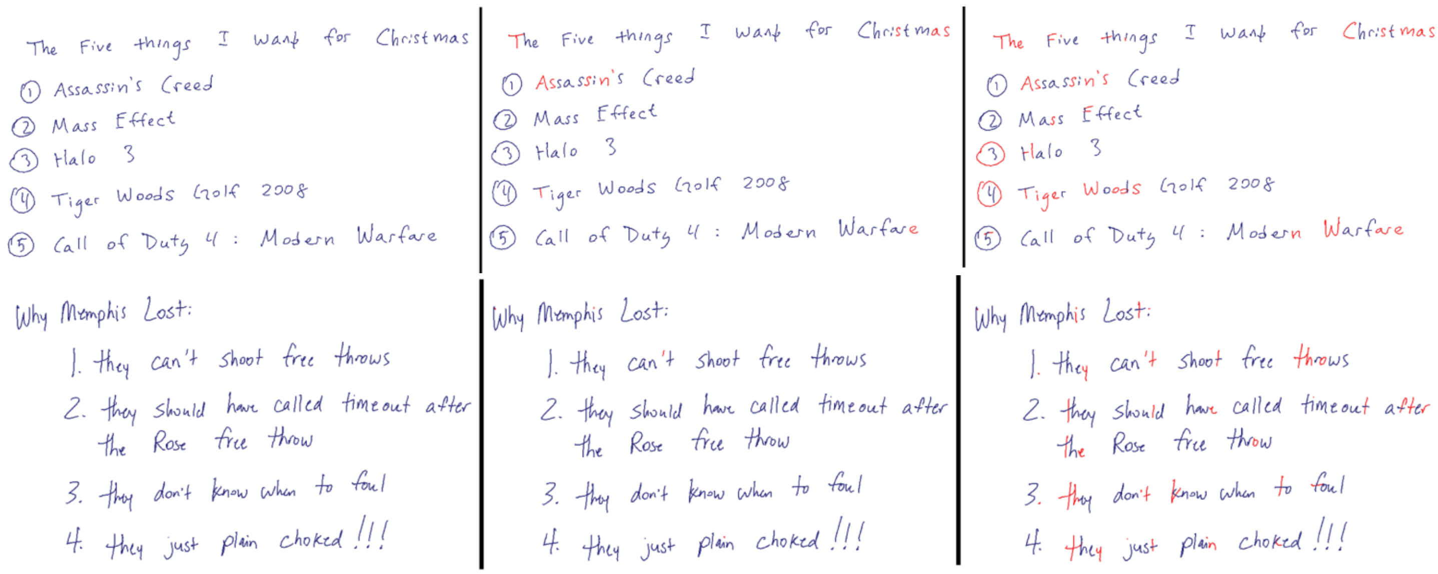


Figure 5: In these samples the strokes in blue were correctly classified, the strokes in red were incorrect. The first sample is from a set of two possible users, the middle sample from five, and the final sample from a set of nine possible users.

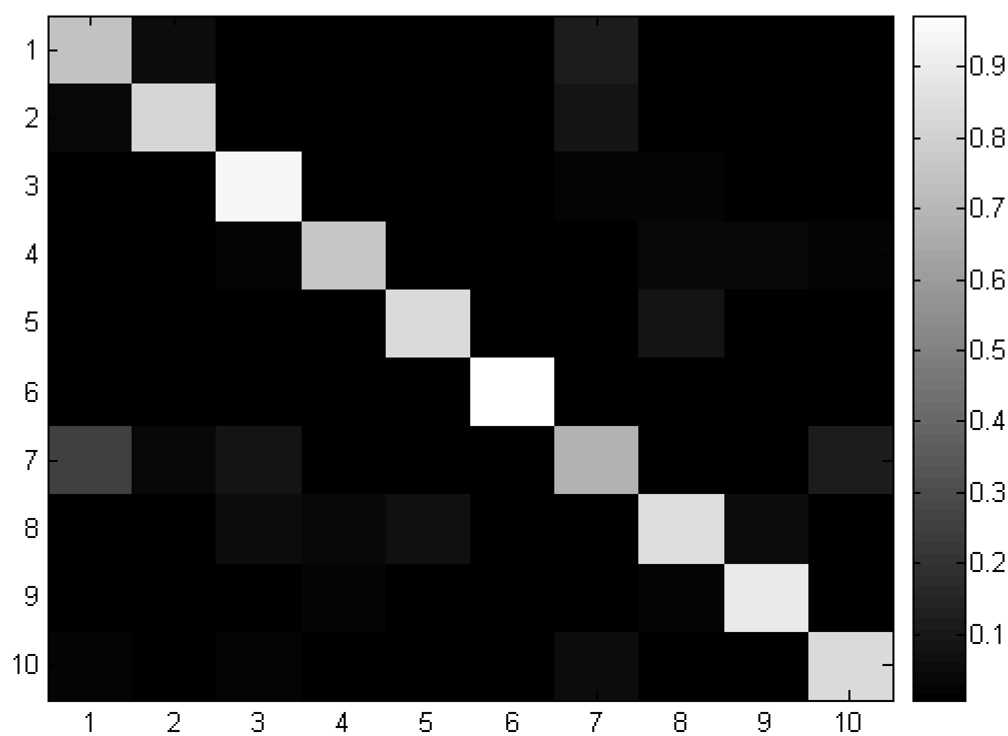


Figure 6: Confusion matrix on the K-NN learner on the full ten participant data set.

will also utilize a variety of feature sub-set selection techniques so that we can determine which features are relevant to this problem.

In the future we will also do users studies in more specific domains. For the purposes of this study we suggested the user make a to-do list, but they were free to write whatever they wished. In the future we will experiment to see if user differentiation works in domain specific areas such as UML diagrams, military course of action (COA) diagrams and Asian text. It is our hypothesis that the domain will not affect our ability to differentiate user strokes, but at this time we have not formally experimented on the matter.

While for this paper we have foregone the use of any information beyond the physical characteristics of the stroke to differentiate creators we are aware that it would be beneficial to use additional context. In future experiments we will record the pen ID (synchronized with timing information to still allow pens to be exchanged) and stroke history to determine how this knowledge aids user differentiation. The work presented here is a baseline of what can be accomplished; we believe utilizing more context information will increase the accuracy and allow us to expand the pool of possible creators.

## 8 Conclusion

Using only features describing the tilt, pressure and speed of a user's pen, a learner is able to classify the creator of a stroke from a set of ten possible creators with an accuracy of 83.5%. When classifying the creator of a stroke from a set of two possible creators, the accuracy is 97.5%. Being able to identify the creator of a stroke without interfering with their drawing habits has many uses. In a collaborative sketch environment, we are able to identify the creator of individual strokes without forcing the user to use a specific pen, or to select a specific mode to indicate who is drawing. We expect this research find to enable broader impacts in the field of forensics, signature verification, sketch recognition, and collaborative interfaces. Users can switch back and forth without explicitly informing the system and still get the benefits of personalization. The results we have accomplished are the baseline of what identification can be accomplished. By using additional context the accuracy can only improve.

## Acknowledgements

This work is sponsored in part by the following NSF grants:

NSF IIS Creative IT Grant #0757557 Pilot: Let Your Notes Come Alive: The SkRUI Classroom Sketchbook

NSF IIS HCC Grant #0744150 Developing Perception-based Geometric Primitive-shape and Constraint Recognizers to Empower Instructors to Build Sketch Systems in the Classroom

## References

[1] C. Alvarado and R. Davis. Sketchread: a multi-domain sketch recognition engine. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 23–32. ACM, 2004.

[2] H. Choi, B. Paulson, and T. Hammond. Gesture recognition based on manifold learning. In *Proceedings of the 12th International Workshop on Structural and Syntactic Pattern Recongition*, December 2008.

[3] E. v. O. J. Dolfing, J.G.A.; Aarts. On-line signature verification with hidden markov models. *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, 2:1309–1312 vol.2, 16-20 Aug 1998.

[4] T. Hammond and R. Davis. Ladder, a sketching language for user interface developers. *Computers & Graphics*, 29(4):518–532, 2005.

[5] C. Herot. Graphical input through machine recognition of sketches. In *Proceedings of the 3rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 97 – 102, 1976.

[6] A. K. Jain, F. D. Griess, S. D. Connell, E. Lansing, and M. J. On-line signature verification. *Pattern Recognition*, 35:2963–2972, 2002.

[7] M. Kawamoto, T. Hamamoto, and S. Hangai. Improvement of on-line signature verification system robust to intersession variability. pages 168–175. 2002.

[8] K. Koppenhaver. *Forensic Document Examination*. Humana Press, 2006.

[9] A. C. Long, J. A. Landay, and L. A. Rowe. "those look similar!" issues in automating gesture design advice. In *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–5, New York, NY, USA, 2001. ACM Press.

[10] R. N. Morris. *Forensic Handwritting Identification*. Academic Press, 2000.

[11] K. Murakami and H. Taguchi. Gesture recognition using recurrent neural networks. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 237–242, New York, NY, USA, 1991. ACM.

[12] D. Muramatsu and T. Matsumoto. Effectiveness of pen pressure, azimuth, and altitude features for online signature verification. *Advances in Biometrics*, pages 503–512, 2007.

[13] V. Nalwa. Automatic on-line signature verification. *Proceedings of the IEEE*, 85(2):215–239, Feb 1997.

[14] A. Richiardi, J.; Hamed Ketabdar; Drygajlo. Local and global feature selection for on-line signature verification. *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 625–629 Vol. 2, 29 Aug.-1 Sept. 2005.

[15] D. Rubine. Specifying gestures by example. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 329–337, New York, NY, USA, 1991. ACM Press.

[16] T. M. Sezgin and R. Davis. Hmm-based efficient sketch recognition. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 281–283, New York, NY, USA, 2005. ACM.

[17] T. M. Sezgin, T. Stahovich, and R. Davis. Sketch based interfaces: early processing for sketch understanding. In *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–8, New York, NY, USA, 2001. ACM Press.

[18] T. M. Sezgin, T. Stahovich, and R. Davis. Sketch based interfaces: Early processing for sketch understanding. In *The Proceedings of 2001 Perceptive User Interfaces Workshop (PUI'01)*, Orlando, FL, November 2001.

[19] J. J. Shah, N. Vargas-Hernandez, J. D. Summers, and S. Kulkarni. Collaborative sketching (c-sketch) - an idea generation technique for engineering design. *The Journal of Creative Behavior*, 35(3):168–198, 2007.

[20] J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 159–168, New York, NY, USA, 2007. ACM.