A Personal Touch - Recognizing Users Based on Touch Screen Behavior

Sarah Martina Kolly
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
kollyma@student.ethz.ch

Roger Wattenhofer
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
wattenhofer@tik.ee.ethz.ch

Samuel Welten
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
swelten@tik.ee.ethz.ch

ABSTRACT

The spread of touch screen based smart phones has been constantly increasing over the last few years. However, there are still many open research questions concerning the basic input properties of these devices. We performed a large scale study to research the users' touch screen behavior on standard UI elements. To do so we programmed and published a quiz game that logs touch data and sends it back for evaluation purposes. Over 14,000 persons have played this game so far and sent back statistical data. We use the collected data to present basic touch properties, such as mean hold time and pressure dynamics, and to show that touch screen based input is individual for each person and that one can identify a specific user in a set of 5 users with a precision of about 80%, based on just a few touch events.

Keywords

behavioral biometrics, touch input, large scale user studies, crowdsourcing

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

Experimentation, Human Factors

1. INTRODUCTION

Although touch screen devices are known since the early 1970s, only recently they gained popularity with the raise of touch screen based smart phones and tablets.

It has been shown for popular input methods such as keyboard and mouse that the way people interact with it are individual and one can recognize who is using it, by analyzing input patterns. Even though touch screen based devices have become the primary means of user interaction of personal smart phones and touch screen based devices are often shared among several persons, little is known about the individuality of general touch screen input.

PhoneSense'12, November 6, 2012, Toronto, ON, Canada.

1.1 Contribution.

In this paper we present a large scale touch screen behavior study on smart phones. Towards this end, we collected touch events of over 14,000 users from around the globe, a total of about ten Gigabytes raw touch data. We study whether users have "a touch of identity", i.e. whether we can distinguish users regarding their touch screen behavior, that is, to what extent their behavior is characteristic. For instance, when choosing five of our users randomly, we can correctly identify a user with a probability of about 80% after just the touch of ten buttons. To the best of our knowledge, this is the first study that tries to differentiate users according to input characteristics on touch screens.

In contrast to other large scale studies about touch behavior, we paid attention to use only standard UI elements in the game, which allows to generalize our observations to other applications that use standard buttons and lists.

2. RELATED WORK

The raising popularity of touch screen smart phones has lead to an increased effort to understand and enhance touch screen based input methods. In the following we give a compact overview on relevant recent approaches in this area.

2.1 Describing and Understanding Touch Input.

Research in touch screen based input methods has revealed two basic problems: poor accuracy and missing haptic feedback. The deficient accuracy was attributed a long time to the so called *fat finger* problem [28]. It states that users are not able to point precisely on a touch screen because the softness of the skin leads to a big, hardly controllable touch area and that the target is not visible because it is occluded by the finger. Recently, Holz et al. have instead proposed a generalization of the *perceived input point* model that should be used instead of the *fat finger* model, which explains the inaccuracy through the different angles in which the screen is touched and most importantly the mental model a user has of acquiring the target [9]. In that paper they clearly state: "the inaccuracy of touch is primarily the result of failure to distinguish between different users and finger postures, rather than the fat finger problem". In a follow-up work they further show that using the *perceived input point* model, one can achieve much better pointing accuracy, if the user and his input point model is known [10]. Similar results were reported Findlater et al., who observed that typing patterns on touch screens vary widely among users but were highly reliable for an individual user [6]. The touch events collected in our study seem to confirm this reported individual position offset on buttons.

Stewart et al. investigated the role of pressure on mobile devices and stated that humans are not that good at consciously controlling the applied pressure, which also points to the direction that pressure as a touch characteristic could be individual for a user [25].

2.2 HCI Input Analysis.

It was well known for a long time that the way a person interacts with a technical device is specific to this person and highly individual. Back in 1899 Bryan and Harter published their observations on telegraph operators. They found out that telegraph operators had distinctive patterns of keying messages over telegraph lines and that often operators recognized who was transmitting information, just by listening to the characteristic pattern [14]. With the increasing proliferation of personal computers, a considerable area of research analyzing keystroke patterns evolved. Spillane et al. first described the use of keystrokes for the purpose of user identification [24]. In the eighties and nineties this became a popular topic. Most research groups reported that the inter-key time was the most distinctive feature [11, 18, 27]. However, Robinson et al. stated that in their experiments the hold times were even more useful to distinguish users [21]. Attempts to perform user classification on smart phones, based on keyboard input, had less success than most of the PC counterparts, due to the limited computation power available [2, 3]. A good overview on the different approaches can be found in [4, 19]. Also the dynamics of the computer mouse were used for biometric authentication (eg. [1, 17]).

An interesting recent approach was published by Saevanee et al., who compared the input of users typing a PIN on a touchpad [23]. Additionally to the keystroke patterns, they reported that the touch pressure information was more discriminative than the timing.

Another research direction in HCI input analysis is the prediction of the emotional state of a user, based on the user's input. An experiment with 12 subjects revealed that it is possible to recognize different emotional states by looking at the keystroke patterns [5]. Similarly, the touchpad characteristics were used to detect negative affect in two different projects. Mentis et al. concluded that it might be possible to recognize negative affect of a user, but did so based on a user study of only 3 participants [16]. McLaughlin et al. presented a user study with 9 participants in which they were not able to recognize frustration reliably, due to large differences among the users [15].

Recent works that used crowdsourcing to understand the users touch behavior in games and on soft keyboards are the publications of Henze et al. and Dmitry et al. Henze et al. used a game for smart phones to record touch events of about 100,000 users. The game consisted of quickly touching circles of different size that appeared at different positions on the screen. They showed that the pointing error rate can be reduced by considering the systematical skew of touch positions [8]. In contrast to their work, our goal was to research the touch behavior of standard UI elements in everyday situations. Therefore, we focused on standard UI touch elements, such as buttons, lists, and radio buttons. Moreover, our game did not put time pressure on the participants in order to receive touch events that are as close as possible to the ones of a everyday usage of the device. Dmitry et al. presented a text-typing game for touch screen based smart phones that offered adaptive key resizing, based on the pointing precision of a user on a on-screen keyboard [22]. Their results indicate that the systematic pointing offset might be individual for a user, but is not very pronounced. Therefore, this work motivates the research question about the individuality of touch gestures.

2.3 Crowdsourcing User Studies.

In the last years, crowdsourcing has become more and more popular. Tools like Amazon's Mechanical Turk or Yahoo Answers provide an easy to use platform to take advantage of the idle capacities of millions of Internet users. Kittur et al. describe how Amazon's Mechanical Turk can be used to explicitly perform user studies [12]. In contrast to this, there are the *games with a purpose*, as presented by Law et al. [13]. They present the concept of using games that rely on input agreement to collect labeled data. While these platforms were first mainly used for a lot of human computation tasks, recently some research groups have started utilizing the potential of crowdsourcing HCI user studies [20, 26]. Examples are the iPhone app of Zhai et al. or Henze et al., who both used the mobile app stores to distribute application for HCI studies [29, 7].

3. METHOD

Even though touch screen based smart phones are becoming standard, little is known about their basic input properties and the user-specific differences. It is therefore the goal of our research to shed a light on the characteristics of this input method. In this section we describe the approach we took to gain insights about the touch input behavior of a large number of smart phone users.

3.1 Recording Touch Data.

Opposed to other smartphone operating systems, the Android platform allows to retrieve detailed information about a touch event, which was the reason for us to develop the touch-recording application for Android. As soon as a person starts to touch the display, the Android OS delivers touch events approximately every 20 milliseconds. An event consists of a timestamp (unix time in milliseconds), the type of the event (up/down/move), the coordinates of the event (x/y in pixels), the normalized pressure value (range $[0, 1]$), and the normalized area value (range $[0, 1]$). First experiments revealed that the reported touch-area value of Android is currently unusable, due to a poor and device-dependent precision. We therefore ignored the area value for the rest of our work.

3.2 Large Scale Deployment.

We decided to crowdsource the user study by designing a quiz game, because it is comparably easy to program and allows to place standard UI elements unobtrusively in the game. Moreover, by choosing a popular topic, a game can quickly attract a large number of users. The quiz consists of questions about the Harry Potter novel and film series. Different types of questions were used to allow the placement of various UI elements:

- **Guesstimates.** This type of question requires to guess or estimate a number related to Harry Potter. A seek bar is offered to enter the answer.
- **Puzzles.** Users have to uncover a picture behind a wall by removing as little bricks as possible. Bricks can be removed by double-clicking on them. Once they think they know the right answer they can select it from a list.
- **Multiple Choice.** Users can answer a question by selecting one of several choices and pressing the *ok* button.

These different questions types were alternatively repeated in 7 difficulty levels. Each level consists of three questions of each type. Additionally, the touch events of all buttons in the menus and between the questions were logged.

All sizes and shapes of the control elements in the game are the same as in common productive Android applications. The only

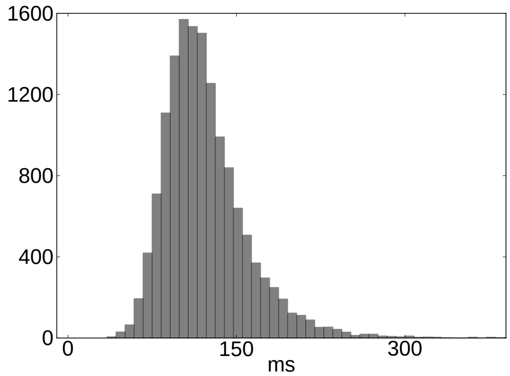

Figure 1: The distribution of the mean touch time of a button per user over all users. The mean touch time over all users is 122ms and the mean standard deviation over all users is 47ms.

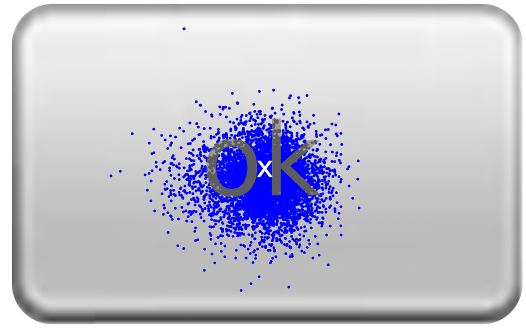

Figure 2: The plot shows the mean landing positions on a button (each point is the mean of one user). One can see the concentration around the center of the button and the slight offset towards the bottom.

visual adjustment we applied to the UI elements is that we changed the color of the buttons to match the Harry Potter theme.

4. USER STATISTICS

This section is based on the touch recordings of 14,890 users, who generated a total of over 1 million button touch events and over 2 million list touch events while playing the Harry Potter game and navigating in the menu. The analysis sheds a light on three basic properties of touch: the timing, the pressure, and the position relative to the target. These three elements can be measured in most touch interactions.

4.1 Timing.

Research in keyboard input analysis has revealed how characteristic the timing of a user's input can be. The two main properties in this context are hold time and inter-key time. Considering touch screen input, two comparable features can be found: the contact time on the screen (hold time) and the inter-touch time in double-clicks. We have computed the mean hold time for each user, which results in the distribution of mean hold times as seen in the plot in Figure 1.

The standard deviation of the mean hold times of all users (36 ms) is in the same order of magnitude as the mean standard deviation of a user (47 ms), which means that this feature is well suited to differentiate individuals.

The mean double click frequency is 4.2Hz with a standard deviation of 1.4Hz per user. With the same argument as for the hold time, the double click frequency is well suited to differentiate users. However, as for touch screens the double click is a less common UI element, the recognition based on the double click frequency might be less relevant in practice.

4.2 Position.

Another property of touch events that is very interesting to observe is the exact position, where a touch is registered by the hardware. To account for different device sizes and display densities we computed the statistics about the touch position only for a subset of devices (5616 devices) that all have the same pixel density and display resolution.

The *perceived input point model* states that the offset of a user's actual touch position to the target is based on a mental model, that might be individual for each user. Therefore, we evaluated the landing position of each button touch. The distribution of the touch positions on a button of 5616 users can be seen in Figure 2. These different means of touch points, confirm the assumption behind the *perceived input point* model, that users have an individual and quite constant offset in horizontal and vertical direction. However, if one looks at the horizontal and the vertical offset separately, differences become visible. The mean offset of a user in horizontal direction is 0.57 density-independent-pixels (dp) to the left and the mean offset in vertical direction is 2 dp to the bottom. This clear tendency to a vertical offset is also visible in Figure 1. Moreover, the average standard deviation of a user over his touch events is smaller in vertical direction than in horizontal direction. This indicates that the offset in vertical direction is more systematic and individual than in horizontal direction.

4.3 Pressure.

One important feature that separates keyboard based input from touch screen input is the ability to measure the pressure a user applies to the touch surface. Even though all modern capacitive smart phone touch screens are capable of sensing the pressure quite precisely, there are significant differences in the reporting of these values. Basically the Android API states that the measured pressure value is normalized to the range of zero to one. However, this does not implicate that all devices return the same value if the same physical pressure is applied. To account for that, we did not include devices that report obviously imprecise measurements for the pressure (i.e. that have only very few discrete values). Still, slight differences among devices may exist.

Despite these constraints, interesting observations concerning the pressure dynamics can be made. Figure 3 depicts how the pressure develops during a touch event. It shows the mean over all button touches from all users. To account for the different hold times of the touch events, the time axis has been normalized before averaging the pressure values. The plot shows, that the maximal pressure is normally reached at about $\frac{1}{3}$ of the total touch time. The high start value can be explained by the threshold that is used to recognize the start of a touch event.

5. USER RECOGNITION

As described in Section 2, a lot of researchers have successfully developed algorithms to identify users based on their keyboard typing patterns or their mouse input. It seems only natural to ask the question if the same can also be done for touch screen input. The

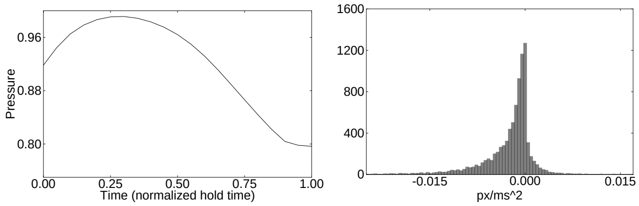

Figure 3: Left: The pressure development of a button click over time, averaged over all users. The high start value most certainly originates from the threshold Android uses to trigger a touch event. Right: The distribution of the mean of the second derivative in vertical direction, while scrolling in lists. Negative values indicate a curvature to the right and positive values a curvature to the left.

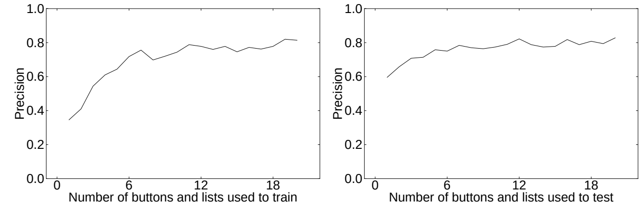

Figure 4: The graphs show the influence of the number of training samples and the number of events that are used to recognize a user, on the recognition rate of the naive Bayes classifier. On the one hand, one can see that for 5 users a recognition rate of about 80% can be achieved and on the other hand the recognition rate does not significantly profit from more than 10 training samples or more than 5 test samples.

previous section has shown that the mean of some of parameters in touch screen interaction differs from user to user. Based on these observations we conducted experiments to find out how well one can identify users based on their touch behavior and on which factors this depends.

As mentioned before, different device types might exhibit hardware-dependent differences, concerning the measurement of pressure. To prevent that the recognition is based on the difference of the hardware rather than on difference of the users, we decided to run the user identification experiments only on sets of identical devices, i.e., the recognition experiments only compare users with the same device.

All experiments described in the following are based on button and list touches only (i.e. no radio buttons or seek bars). This decreases on the one hand the recognition quote, but shows on the other hand how little touch information is needed to identify a person.

Moreover, in all experiments we have separated training and test data sets to avoid overfitting. If not mentioned differently the experiments are performed using ten button touches and five list touch event for training and five button/list touch events for classification.

5.1 Features.

In machine learning, the standard approach to supervised learning is to extract meaningful features from a labeled data source and to train a classifier on these features and labels. We used the observations of our large scale user statistics to derive the following set of promising features on which the classifier will work: The mean and the maximal pressure of a touch event, the point in time when the maximal pressure occurs, the minimal and the maximal gradient of the pressure, the hold time of a touch event, the mean X and Y position (relative to the center of the touch element), and the variance of the touch event in X and Y direction.

5.2 Classifiers.

In literature, a wide variation of classifiers are described for supervised learning problems. To get an impression of how well which classifier works, we used the *Weka machine learning toolkit* to compare a large amount of classifiers on a subset of our data from the large scale user study. Based on these preliminary experiments we concentrated on the naive Bayes classifier to differentiate multiple users and for anomaly detection, as it has a low computation complexity and performs not significantly worse than other common classifiers on our test data.

The restriction to use a classifier with low computation complexity allows to perform the user recognition also in the resource constrained environment of smart phones.

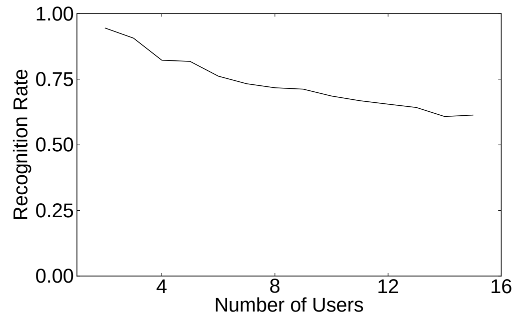

Figure 5: The influence of the number of users on the recognition rate on the naive bayes classifier. For two user the recognition rate is 94%, whereas for ten users it is at 68%.

5.3 Recognition Factors.

The difficulty of recognizing a user out of given set of users depends on several factors. In the following we highlight three of them (see Figure 4 and Figure 5): The number of touch events used to train the classifier, the number of test events used to classify a user, and the size of the set of user, out of which the users should be recognized. It can be seen in Figure 5 that the recognition rate depends highly on the number of users the classifier should be able to differentiate. For small groups of up to 5 persons, it works reasonably well (80% or more), whereas for larger groups the recognition rate drops significantly. Concerning the number of needed training samples, one can see that as little as ten button touch events and ten list scroll events are needed to train the classifier to recognize a user. Using more than five touch event to classify a user, does not significantly increase the recognition probability.

5.4 Anomaly Detection.

Another scenario of touch based input analysis is detecting if a different person than the true owner of the device, for example a thief, is using it. In comparison to just differentiating between a given set of users, as described above, this scenario needs other methods. In this case, one has to learn the distribution of the data in the normal case (i.e. the true owner), without knowing how the data would look like in an abnormal case (i.e. a thief). This problem is often referred to as anomaly detection. We have used the

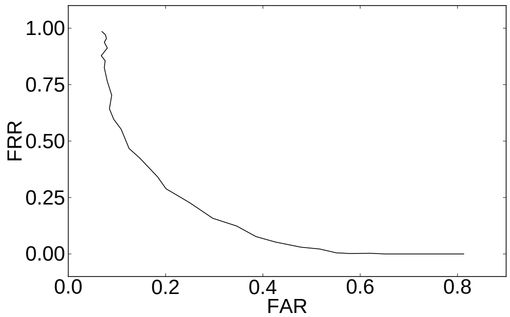

Figure 6: Simple anomaly detection using a naive Bayes classifier. The plot shows the false accept rate versus false reject rates over different thresholds.

collected usage data to test a simple touch based anomaly detection algorithm.

In our experiment, we use the features of five button touch events from the true owner to learn the distribution of the features. This is done by assuming a Gaussian distribution of the feature values and estimating the mean and the variance of each feature. Whenever a new touch event is observed, one can use these distributions to compute the probability that the observed event was generated by the true owner. A threshold is used to determine if a touch event should be considered as coming from the true owner or not.

We trained and tested this algorithm with random users from our data set and varied the probability threshold to generate the false-accept/false-reject plot in Figure 6.

One can see that the equal error rate is at approximately 30%, which is not very low, compared to other input based anomaly detection methods. However, this experiment should clearly be seen a proof of concept and the equal error rate could certainly be improved by using more elaborate anomaly detection algorithms and by including more touch dynamics.

6. CONCLUSION

After many years of input analysis on keyboard patterns, we have demonstrated that the potential that lies in the analysis of touch screen events on smart phones is at least comparable to the one of keyboard input analysis. Many UI elements in touch interfaces lead to interesting dynamic usage patterns that, in some cases, differ widely among the users and result in a very individual usage profile.

7. REFERENCES

[1] A. Ahmed and I. Traore. A new biometric technology based on mouse dynamics. *IEEE TDSC*, 2007.

[2] A. Buchoux and N. Clarke. Deployment of keystroke analysis on a smartphone. In *Australian Conf. on Information Security & Management*, 2008.

[3] N. L. Clarke and S. M. Furnell. Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security*, 6(1), 2007.

[4] H. Crawford. Keystroke dynamics: Characteristics and opportunities. *PST*, 8, 2010.

[5] C. Epp, M. Lippold, and R. L. Mandryk. Identifying emotional states using keystroke dynamics. CHI, 2011.

[6] L. Findlater, J. O. Wobbrock, and D. Wigdor. Typing on flat glass: examining ten-finger expert typing patterns on touch surfaces. CHI, New York, NY, USA, 2011.

[7] N. Henze, M. Pielot, B. Poppinga, T. Schinke, and S. Boll. My app is an experiment: Experience from user studies in mobile app stores. *IJMHCI*, 2011.

[8] N. Henze, E. Rukzio, and S. Boll. 100,000,000 taps: Analysis and improvement of touch performance in the large. In *Proceedings of Mobile HCI 2011*. ACM, 2011.

[9] C. Holz and P. Baudisch. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. CHI, 2010.

[10] C. Holz and P. Baudisch. Understanding touch. CHI, 2011.

[11] R. Joyce and G. Gupta. Identity authentication based on keystroke latencies. *Commun. ACM*, 33, 1990.

[12] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. CHI, 2008.

[13] E. Law and L. von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. CHI, 2009.

[14] J. Leggett, G. Williams, M. Usnick, and M. Longnecker. Dynamic identity verification via keystroke characteristics. *IJMMS*, 1991.

[15] M. McLaughlin, Y. Chen, N. Park, W. Zhu, and H. Yoon. Recognizing user state from haptic data. In *ISAS*, 2004.

[16] H. M. Mentis. Using touchpad pressure to detect negative affect. ICMI, 2002.

[17] Y. Nakkabi, I. Traore, and A. Ahmed. Improving mouse dynamics biometric performance using variance reduction via extractors with separate features. *IEEE SMC*, 40(6), nov. 2010.

[18] M. Obaidat and B. Sadoun. Verification of computer users using keystroke dynamics. *IEEE SMC*, 1997.

[19] A. Peacock, X. Ke, and M. Wilkerson. Typing patterns: A key to user identification. *IEEE SP*, 2004.

[20] A. J. Quinn and B. B. Bederson. Human computation: a survey and taxonomy of a growing field. CHI, 2011.

[21] J. Robinson, V. Liang, J. Chambers, and C. MacKenzie. Computer user verification using login string keystroke dynamics. *IEEE SMC*, 28(2), mar 1998.

[22] D. Rudchenko, T. Paek, and E. Badger. Text text revolution: a game that improves text entry on mobile touchscreen keyboards. Pervasive, 2011.

[23] H. Saevanee and P. Bhattarakosol. Authenticating user using keystroke dynamics and finger pressure. In *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*. IEEE, 2009.

[24] R. Spillane. Keyboard apparatus for personal identification. *IBM Technical Disclosure Bulletin*, 1975.

[25] C. Stewart, M. Rohs, S. Kratz, and G. Essl. Characteristics of pressure-based input for mobile devices. CHI, 2010.

[26] B. Tidball and P. Stappers. Crowdsourcing contextual user insights for ucd. In *Workshop on Crowdsourcing and Human Computation*, CHI, 2011.

[27] D. Umphress and G. Williams. Identity verification through keyboard characteristics. *IJMMS*, 1985.

[28] D. Vogel and P. Baudisch. Shift: a technique for operating pen-based interfaces using touch. CHI, 2007.

[29] S. Zhai, P. O. Kristensson, P. Gong, M. Greiner, S. A. Peng, L. M. Liu, and A. Dunnigan. Shapewriter on the iphone: from the laboratory to the real world. CHI EA, 2009.