28 Frames Later: Predicting Screen Touches From Back-of-Device Grip Changes

Mohammad Faizuddin Mohd Noor[*†], **Andrew Ramsay**[*], **Stephen Hughes**[‡], **Simon Rogers**[*], **John Williamson**[*] **and Roderick Murray-Smith**[*]

[*]School of Computing Science, University of Glasgow
{noorm,adr,srogers,jhw,rod}@dcs.gla.ac.uk

[†]Universiti Kuala Lumpur
Malaysian Institute of Information Technology
mfaizuddin@miit.unikl.edu.my

[‡]SAMH Engineering
Blackrock, Ireland
stephenahughes@gmail.com

ABSTRACT
We demonstrate that front-of-screen targeting on mobile phones can be predicted from back-of-device grip manipulations. Using simple, low-resolution capacitive touch sensors placed around a standard phone, we outline a machine learning approach to modelling the grip modulation and inferring front-of-screen touch targets. We experimentally demonstrate that grip is a remarkably good predictor of touch, and we can predict touch position 200ms before contact with an accuracy of 18mm.

Author Keywords
capacitive; touch; back-of-device; machine learning;

ACM Classification Keywords
H.5.2. User Interfaces: Input devices and strategies

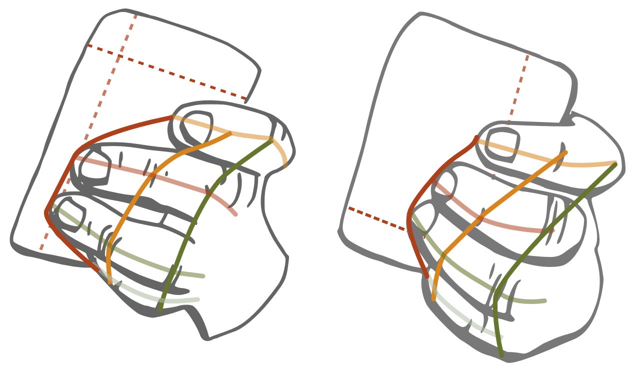


Figure 1. Grip changes as the thumb targets different areas. Thumb target shown as dashed crosshairs. Notice the change in the contours of the phalanges and finger tips (solid lines).

INTRODUCTION
Touch input has become the dominant form of interaction with mobiles. There have been a number of proposed enhancements to touch interaction recently described to overcome input space constraints and extend the capabilities of touch, including hover tracking, full finger pose estimation and back-of-device/around-device interaction. Standard front-of-device touch is, however, likely to remain the most common modality for the forseeable future, because of its direct link between control and display. In this paper we explore how back-of-device sensors can improve front-of-device interaction by predicting the contact of fingers *before* they reach the touchscreen. This is based on the observation that, when holding a phone single-handed, it is impossible to target with the thumb across the whole display without adjusting grip (Figure 1). This paper explores *implicit* back-of-device interaction for the purpose of estimating front touch position. We focus on finding structures in the hand grip modulations and correlating these with touch actions. We use standard machine learning techniques to do prediction, forming a regression model which predicts x, y position and expected time of contact t from a capacitive sensor time series. As well as being an interesting result in its own right, this could be applied to extend user interfaces (e.g. with mid-air "taps") or to improve error correction (e.g. as a more robust measure of finger "slip"), and it has immediate and compelling application to reducing latency in mobile applications.

CHI 2014, April 26–May 01, 2014, Toronto, ON, Canada.
Copyright © 2014 ACM 978-1-4503-2473-1/14/04...$15.00.
http://dx.doi.org/10.1145/2556288.2557148

Use case: Preloading content
Off-device, cloud-based processing offers many opportunities for mobile interaction. One of the key issues holding back cloud applications is extended UI latency. Retrieving content over a wireless link introduces a substantial necessary latency; even a fast connection may have latencies of 100-200ms. This level of delay is very noticeable, and can disrupt the rhythm of an interaction. Prediction of touch events could be used to identify interface components a user was about to touch, and preload the content associated just ahead of time The accuracy of this touch prediction determines how much content needs to be downloaded (e.g. just for one button push or for four nearby buttons?) and thus the feasibility of the preloading.

Use case: Feedback Responsiveness Enhancement
Another potential application of touch contact prediction is enhancing auditory and tactile touch feedback. Existing feedback solutions can increase user confidence that touches have been registered, but introduce latency of their own (e.g. audio buffering delay). A delay of just 30ms between touch and response is clearly apparent. By predicting touch contact

times, audio or vibrotactile feedback can be queued to trigger exactly on the predicted touch time. This requires high predictive accuracy, but only at the fraction of a second immediately preceeding a touch.

RELATED WORK

In order to overcome occlusion problem, new interaction technique using back of the device has been proposed. This is by using a see through mobile device that allows direct touch input to be made precisely [11]. Apart from the occlusion problem, back of device interaction also has shown to be useful in increasing privacy by preventing *shoulder-surfing* [7], and to overcome *fat finger* problem in small devices [1]. Back of device interaction also allows the creation of grasp-based technique that could predict users' intention by the way they hold the device [6]. The Bar of Soap is a multifunction prototype device that used grasp interaction to switch between several hand-held modes [10]. Similarly, HandSense discriminates between different ways of grasping a device which can be used as interaction cues in both explicit and implicit ways [12]. The use of back of device sensing also allows mobile devices to be more adaptive to the dynamic nature of user interaction such as soft keyboard positioning in iGrasp [3] and screen orientation in iRotate [2]. Besides capacitive technology, users' hand postures also can be inferred using combination of built-in sensors found on most commodity mobile phones [4]. Alternatively using active acoustic sensing, rough positions of touch and different postures of touch on solid object can also be estimated [8].

EXPERIMENTAL SETUP

(a) Back view (b) Front side view (c) PCB view

Figure 2. Overview of the prototype device used in the experiment. The sensor pads are marked in yellow in (c).

Current smartphones do not typically have grip sensing around the device, and so we fabricated a custom prototype system (shown in Figure 2a). The prototype based around a Nokia N9, which has been modified to include around device sensing using a 0.1mm thick flexible PCB, interfaced directly to the phone's internal I2C bus with custom electronics. The prototype has 24 capacitive sensors distributed around the back and sides of the device (Figure 2c) to capture user's hand grip. We use 2x AD 7147 programmable touch controllers. The total size of this prototype is fractionally larger than the device itself, with dimensions of 116.5 mm x 61.2 mm x 12.1 mm and a weight 135 g. The N9 prototype has a screen density of 251 pixels per inch (ppi). Capacitive sensing technology is used because it is a well proven touch sensing technology which is practically implementable on mobile devices. The flexible PCB solution gives us a prototype which is almost identical in form factor to a standard mobile device. The prototype is configured to sample data at $50Hz$. The capacitive sensing has a raw bit conversion depth of 16 bits. It is subsequently filtered, offset removed and scaled to fit in an 8 bit container in software. A Python application was developed on the prototype to coordinate the data acquisition.

Data acquisition

In order to collect touch grip samples, 20 users were recruited locally (12 male and 8 female, age $25 - 40$). For each user, we recorded 250 unique touch targets with each hand, while seated on a chair, in front of their desk. We are not interested in how the users initially pick up the phone, therefore the recordings begin when the phone is held by the users. We used 5 sessions for each hand, each with 50 targets, for 500 targets in total, alternating hand between each session, for 250 targets for each hand. Each hand therefore has an equal number of touches. This is to ensure that we are not observing only a single grip pattern, but a range of plausible grips for each user. It is worth mentioning that each session was separated by a 5 minute break to minimise the repetition effects. The experiment required the user to touch random targets distributed randomly on the prototype screen using their thumb, while holding the phone single handed. A half second delay is used between targets to encourage the user to return to a rest pose before next target is shown. Audio feedback is given if the user touches the target correctly. A legitimate touch requires a stable thumb contact within the minimum target area for at least 60ms. The target area used in our setup is 1 cm in diameter or 98.8 pixels on our device. We recorded timestamps, both target and touch coordinates (x, y) in pixels and capacitive readings from the back of the device into the prototype's internal storage for subsequent off-line analysis.

Analysis methods

From the recorded samples, we performed Principal Component Analysis (PCA) to visualise the structure of the capacitive signal coming from the back of the device. In particular, we are interested to see whether there is a correlation between grip (back of device) and touch target (front of device). We used Canonical Correlation Analysis (CCA) to study this relationship. Drawing the results from CCA, we performed regression to see if touch target predictions can be made from the way the device is being grasped.

Canonical Correlations Analysis (CCA)

CCA [5] measures linear correlation between two multidimensional datasets. In our case, we have a 24-dimensional vector describing the capacitive sensor values at a given time point, $\mathbf{s}$ and a 2-dimensional vector defining the target the user was aiming for $\mathbf{x}$. CCA finds projection vectors, $\mathbf{a}$ and $\mathbf{b}$ such that for a set of $n = 1 \dots N$ observations (sensor-target pairs), $u_n = \mathbf{a}^T\mathbf{s}_n$ is maximally correlated with $v_n = \mathbf{b}^T\mathbf{x}_n$. Typically, CCA finds M pairs of projection vectors, $(\mathbf{a}_1, \mathbf{b}_1), \dots, (\mathbf{a}_M, \mathbf{b}_M)$ where M is equal to the dimensionality of the smaller data space (in our case, 2). The first pair

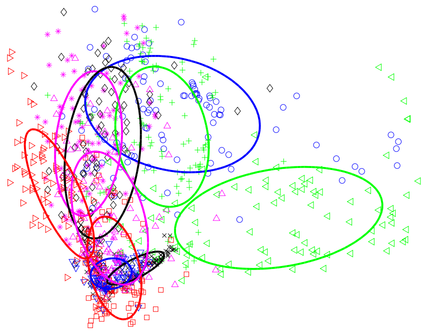


Figure 3. Principal component analysis of capacitive sensor values, s from 10 users for right hand during the touch. Each colour/symbol combination represents one user. For simplicity, only 100 samples are shown from each user. It is clear that users have quite distinct grip patterns.

of projection vectors provide the most correlated linear combinations and the second pair define the most correlated linear combinations that are orthogonal to the first, etc.

Gaussian Process Regression

Our final goal is to predict the intended target location $\mathbf{x}$ using the back of device sensors $\mathbf{s}$, which is naturally viewed as a regression task. Gaussian Process Regression (GP) [9] is a flexible, non-parametric approach to regression analysis. To define a GP, we define a prior mean regression function (in our case; $f(\mathbf{s}) = 0$) and a prior covariance function that defines the smoothness of the regression function. We train a separate, independent GP for each co-ordinate axis. In this work, we use the popular Gaussian covariance function. We used the `gpml` GP package for Matlab. In all experiments, the data are split into independent training and test sets. The hyper-parameters are optimised by maximising the marginal likelihood on the training data (see [9] for details).

RESULTS

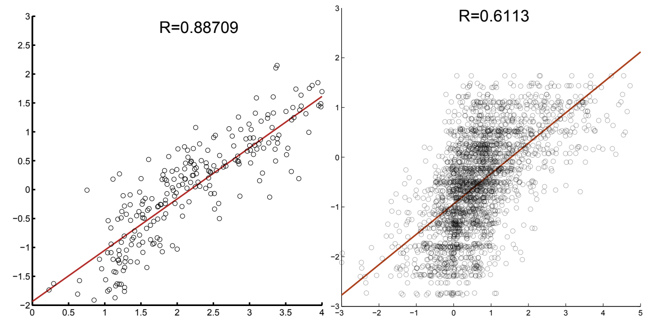
We start by identifying the structure of hand grip data during the touch. We used Principal Components Analysis (PCA) on hand grip to project the 24-dimensional capacitive values, $\mathbf{s}$ to two-dimensional space. This allows us to observe patterns in the data. Figure 3, shows the first two components from right hand data from all users, and we can see that most of the users have different ways of holding the phone during touches. This diversity suggests that any model based from hand grip may have poor generalisation ability and is likely to be user-specific.

Canonical correlation analysis

In order to understand the correlation between grip and touch, we use CCA to measure the linear relationship between capacitive sensors, $\mathbf{s}$ and touch targets, $\mathbf{x}$. CCA provides 2 bases, one for each variable, that are optimal with respect to correlation. The plot of correlation coefficients in Figure 4 shows that the two variables are correlated.

Prediction of touch targets

Based on the touch-grip examples, we train the GP to predict touch targets before finger contact. We use root-mean-square error (RMSE) in millimetres to evaluate prediction error (Figure 5) and compare our results with a baseline defined by RMSE of always guessing the centre of the screen (an uninformed guess). To predict touch target before time of contact, we train the GP using grip data prior to the touch contact and measure the RMSE of the prediction on a separate test set. Figure 6 show the error against time before contact.



(a) Individual **(b) Pooled**

Figure 4. (a) shows an example of single user canonical correlation analysis. (b) Analysis based on 100 random samples pooled from every user. x and y axes correspond to first canonical components of s and x respectively. There is clear correlation between the back-of-device sensor values and the touch position.

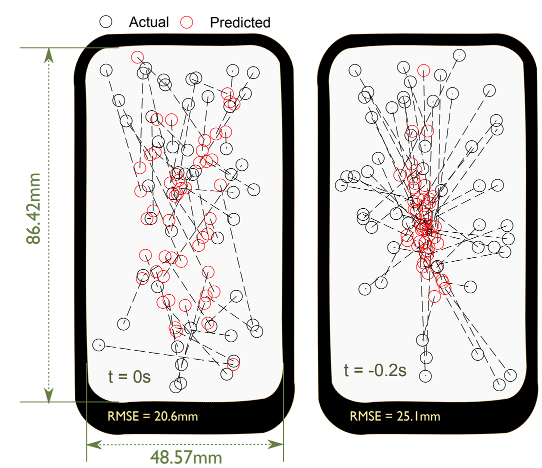


Figure 5. Example of touch target predictions for a random user, right hand at $t = 0$s and $t = -0.2$s. Black markers correspond to real targets and red markers correspond to predicted targets.

Prediction of contact time

To predict the time the finger will make contact with the display, we extend the feature vector to include the first time derivative (estimated using an order-4 Savitzky-Golay filter). We train a new GP with the resulting 48-dimensional feature vector. As Figure 7 shows, we can estimate time of contact accurately just before touch, with reasonable estimates up to 0.5 seconds before contact.

DISCUSSION

The results show that there is a surprisingly strong correlation between grip modulation and touch target, and we can predict touch contact position reasonably well several hundred milliseconds before touch. This is accurate enough to estimate

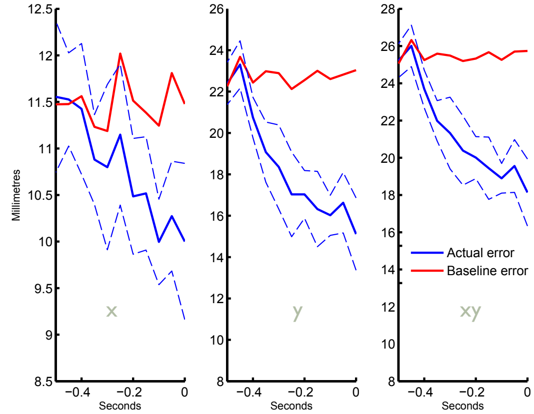


Figure 6. RMSE of touch target predictions including ± standard error before touch contact for right hand averaged across 10 users. Left and middle panels correspond to prediction error for x and y axes and right panel corresponds to combination error of x and y axes.

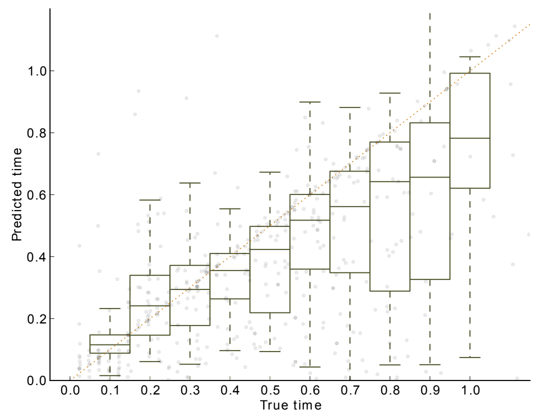


Figure 7. Predicted time of contact against actual time of contact, for all users, right hand, 1000 samples. The system can predict contact time with in a fairly small interval, even half a second before a touch.

the broad region of the screen user is targeting (e.g. to identify which cell in 3x3 division of the screen is being touched), and more than enough to enable effective preloading of content. Time-of-contact is also remarkably predictable. Prediction accuracy is similar for both left and right hand use, regardless of the user's handedness. We have focused on a specific targeting paradigm in this study – touching randomised abstract targets with one thumb. The grip dynamics of tasks with a known target positions (e.g. typing) may be different; this remains to be investigated. Other interaction poses, such as two-thumb interaction and single-finger tapping, are also likely to have substantially different grip models. Although our results suggest that the model may not be suitable for generalisation, however it could be possible to establish a group of people (based on clusters), and generalise the model based on this group.

CONCLUSION
The grip manipulations required to touch targets on a mobile touch screen have a distinct signature. Our methods are able to use this to predict finger contacts with a degree of accuracy that could enhance a wide range of mobile applications by reducing apparent latency. Gaussian process regression is efficient in learning a compact and robust mapping from a fairly low-resolution grip sensor to target positions and contact times. Although we used user-specific grip models a system using a pooled model combined with a small individual training sample may provide adequate performance without requiring a lengthy enrolment process. The use of back-of-device interaction for explicit interaction is a well explored area. Implicit interaction with whole device sensing offers opportunities to transparently enhance standard interaction techniques and build devices with responsiveness and precision beyond that which is possible from standard surface contact sensing.

ACKNOWLEDGEMENTS
We thank Nokia for funding the back of device hardware as part of the project "Human Emotional Communication in the field of Quality and Rapport".

REFERENCES

1. Baudisch, P., and Chu, G. Back-of-device interaction allows creating very small touch devices. *CHI '09* (2009), 1923.
2. Cheng, L.-P., Hsiao, F.-I., Liu, Y.-T., and Chen, M. Y. iRotate grasp. In *UIST Adjunct Proceedings '12* (2012), 15.
3. Cheng, L.-P., Liang, H.-S., Wu, C.-Y., and Chen, M. Y. iGrasp. In *CHI '13* (2013), 3037.
4. Goel, M., Wobbrock, J., and Patel, S. Gripsense: Using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *UIST '12* (2012), 545–554.
5. Hotelling, H. Relations between two sets of variates. *Biometrika 28* (1936), 321–377.
6. Kim, K., Chang, W., Cho, Sung-jung Shim, J., and Lee, H. Hand grip pattern recognition for mobile user interfaces. *IAAI '06* (2006), 1789–1794.
7. Luca, A. D., von Zezschwitz, E., Nguyen, N. D. H., Maurer, M.-E., Rubegni, E., Scipioni, M. P., and Langheinrich, M. Back-of-device authentication on smartphones. In *CHI '13* (2013), 2389.
8. Ono, M., Shizuki, B., and Tanaka, J. Touch and activate: Adding interactivity to existing objects using active acoustic sensing. In *UIST '13* (2013), 31–40.
9. Rasmussen, C. E., and Williams, C. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
10. Taylor, B. T., and Bove, V. M. The bar of soap: a grasp recognition system implemented in a multi-functional handheld device. *Ext. Abstracts CHI'08* (2008), 3459–3464.
11. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. Lucid touch. In *UIST '07* (2007), 269.
12. Wimmer, R., and Boring, S. HandSense: discriminating different ways of grasping and holding a tangible user interface. *TEI '09* (2009), 16–19.