PocketTouch: Through-Fabric Capacitive Touch Input

T. Scott Saponas[1] *Chris Harrison*[1,2] *Hrvoje Benko*[1]

[1]Microsoft Research
One Microsoft Way
Redmond, WA 98052
{ssaponas, benko}@microsoft.com

[2]Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
chris.harrison@cs.cmu.edu

ABSTRACT
PocketTouch is a capacitive sensing prototype that enables eyes-free multitouch input on a handheld device without having to remove the device from the pocket of one‘s pants, shirt, bag, or purse. *PocketTouch* enables a rich set of gesture interactions, ranging from simple touch strokes to full alphanumeric text entry. Our prototype device consists of a custom multitouch capacitive sensor mounted on the back of a smartphone. Similar capabilities could be enabled on most existing capacitive touchscreens through low-level access to the capacitive sensor. We demonstrate how touch strokes can be used to initialize the device for interaction and how strokes can be processed to enable text recognition of characters written over the same physical area. We also contribute a comparative study that empirically measures how different fabrics attenuate touch inputs, providing insight for future investigations. Our results suggest that *PocketTouch* will work reliably with a wide variety of fabrics used in today‘s garments, and is a viable input method for quick eyes-free operation of devices in pockets.

ACM Classification: H.5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces; Input devices and strategies.

General terms: Algorithms, Design, Human Factors

Keywords: Touch, gestures, eyes-free, touch-stroke text entry, always-available input, wearable computing.

INTRODUCTION
Modern mobile devices are sophisticated computing platforms, enabling people to handle phone calls, listen to music, surf the web, reply to emails, compose text messages, and much more. These devices are often stored in pockets or bags, requiring users to remove them in order to access even basic functionality. This demands a high level of attention - both cognitively and visually - and is often socially disruptive. Further, physically retrieving the device incurs a non-trivial time cost, and can constitute a significant fraction of a simple operation‘s total time [2].

UIST'11, October 16–19, 2011, Santa Barbara, CA, USA.

Figure 1. PocketTouch can sense multitouch finger input through a variety of garments.

To address these issues, several techniques have been implemented or proposed. Many consumer mobile devices already include one mechanism for quick, through-pocket interaction: a physical button. For example, one can use a physical switch to quickly silence an incoming phone call. Although simple, this feature is extremely valuable. However, supporting in-pocket operation using physical buttons is limiting, since one can only support a few discrete actions. In response, several research projects have sought to expand the interaction space by taking advantage of richer input channels. Of note are Tap Input [11] and Whack Gestures [7]. Both of these techniques support a small vocabulary of through-pocket gestures using a device‘s accelerometer (directional taps and coarse whacks, respectively). An entirely different approach is to embed sensing for interaction directly into textiles [8,9]. However, this requires wearing custom garments equipped with sensing.

In this work, we present a novel method for through-pocket interaction, called *PocketTouch*. Our approach utilizes capacitive sensing to detect finger-strokes through fabric (Figure 1). Sensing strokes through pockets enables a wide variety of interaction possibilities in situations where taking

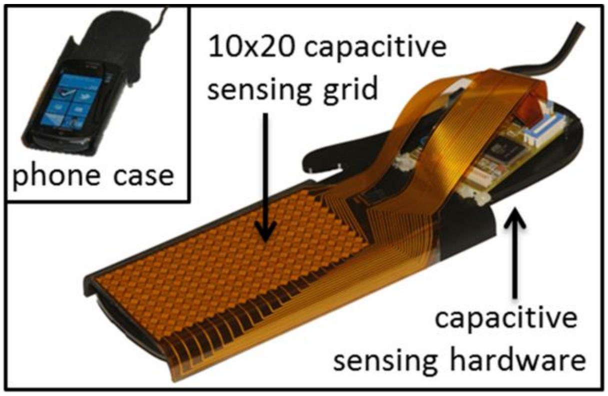


Figure 2. PocketTouch prototype. Inset shows PocketTouch mounted on the back of a smartphone.

out a device is inconvenient or impossible. Our technique contributes to the domain of eyes-free and always-available interfaces [12]. In this paper, we demonstrate two uses of through-pocket stroke detection. First, we show that single and multi-stroke gestures can be used to issue commands to mobile devices such as ―next song” or ―new message.” Second, we illustrate how through-pocket strokes can be used for eyes-free text input on an enclosed device.

The core contribution of this paper is demonstrating the feasibility of using finger-strokes to control a mobile device equipped with capacitive sensing *through* a pocket‘s enclosing material. In our prototype, we built a custom capacitive sensor that fits on the back of a mobile phone. However, we envision future mobile devices having capacitive-sensing ability on both sides, enabling *PocketTouch* interaction regardless of a device‘s in-pocket orientation.

Finally, for brevity, we use the term ―pocket.” However, we consider this to encompass a variety of possible storage locations, including pants, shirts, jackets, purses, outside pockets on backpacks, and many other garment placements.

PROTOTYPE HARDWARE

Enabling through-pocket use of capacitive touch screens requires adaptively adjusting the low-level sensitivity settings of the sensor not typically exposed by a capacitive touch controller‘s APIs. Our custom unit allowed us full programmability of the sensor (Cypress Semiconductor CY8CTMA3), but similar capabilities can be enabled on many existing capacitive touchscreens through low-level access to the sensor. Our input area measures 46x97mm.

We mounted our touch sensitive surface to a 3D-printed base 11.5mm thick, with a flat top, simulating a smart phone form factor (see Figure 2). The touch controller board is also attached to the base. The controller is tethered over USB to a host machine and all touch processing described in this paper is performed on a PC host. To further explore interactions in the mobile domain, our form supports being mounted on the back of an existing smartphone (Figure 2, inset). This configuration is for mockup purposes at present; exploration of through-pocket smartphone interactions remains future work.

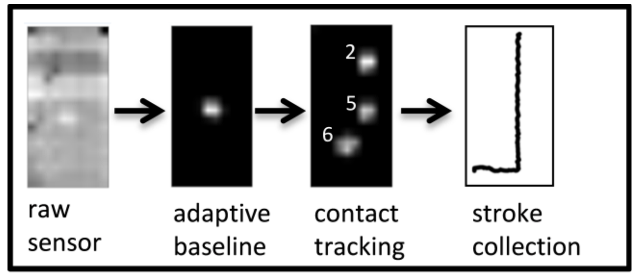


Figure 3. PocketTouch processing pipeline.

TOUCH PROCESSING

To enable touch gesture detection in the pocket, it is important to understand the underlying touch processing that extracts and tracks individual contacts. Our capacitive system samples a 10x20 grid of electrodes at approximately 100 Hz. For simplicity, it is easier to consider this matrix as a 10x20 image of capacitive pixels where the intensity of each pixel corresponds to the capacitive signal at that location. In each frame, we segment finger contacts by subtracting the current signal from a baseline image. We then search for connected components, labeling each connected component as a contact, and track each contact over time thus producing strokes. This touch tracking process is similar to many camera-based sensing prototypes (e.g., [10]) or other capacitive sensing devices [13,14].

What is different about our approach is that inserting the device into a pocket dramatically changes the operating conditions of the senor. Most touch sensors are calibrated to work in a direct-contact manner and to reject signal that is weaker than the baseline. However, with *PocketTouch*, the touch signal is attenuated by the fabric requiring the software to continuously calibrate the sensor, thus effectively normalizing the signal and adapting it to the current conditions. Our baseline is created by sampling each pixel at startup and continuously adapting the baseline by adding 0.05% of the current value at each pixel to 99.95% of the baseline value. This acts as a slow high-pass filter to adapt to changing conditions.

To avoid including noise or inadvertent touches as strokes, we first discard any contacts that are only visible for a short period of time or do not move far during their existence. We also smooth strokes with gaps or that eclipse the boundary of the sensor by merging together strokes where the end of one stroke is temporally and spatially near the beginning of the next stroke.

GESTURE AND TEXT RECOGNITION

One very useful application of stroke detection through a pocket is stroke-based gesture input. This style of input can be quick, expressive, and eyes-free. In our prototype system, we detect single and multi-stroke gestures using an implementation of the $N gesture recognizer [1].

To demonstrate the expressivity of *PocketTouch*, we built a text entry recognizer. This is more complex than gesture recognition given the small size of mobile device screens.

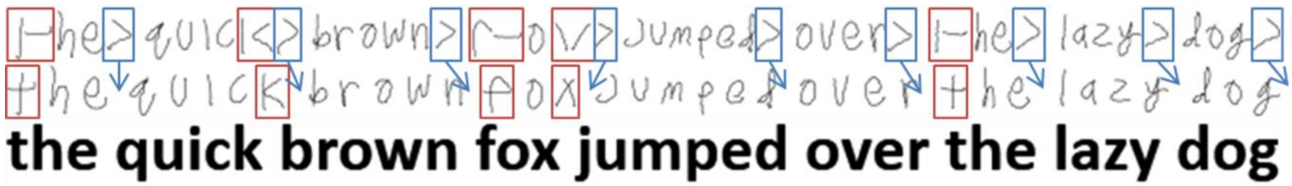


Figure 4. A sequence of strokes (top) can be formed into characters, words, and ultimately sentences (bottom). A '>' character acts like a space bar (blue highlight). Strokes from multistroke characters (t, k, f, x) are spatially merged (red highlight).

Indeed, it is near impossible to accurately stroke an average word in the confines of a small handheld screen. However, an individual character can easily be drawn in the space of a mobile phone. We utilize this ability to implement a through-pocket text entry system.

When characters are drawn in sequence they are written on top of one another. To address this, we use an additional stage of stroke processing. To separate words, we use the > gesture, which acts like a space bar. Single stroke letters are easily segmented, and can be arranged in temporal order to form words. To handle lowercase letters composed of two strokes (k, t, f and x), we take advantage of the fact that the second stroke is unique. Specifically, if a < — \ or / gesture is encountered, we simply merge the stroke with the previously entered character (Figure 4). Our approach makes it impossible to enter these reserved characters, but we feel that the brief ad hoc operation in pocket will not be impeded much by missing these few characters.

Once an entire word is entered, followed by a > gesture, we lay out the character strokes in 2D (Figure 4). We then pass this composite stroke collection to the Microsoft Ink API, which uses a language model for word recognition. Importantly, this step also resolves ambiguities between characters such as i and l (e.g., ―llt" is recognized as ―ilt").

The main benefit of our approach is that it does not require the user to learn a new gesture alphabet, but simply to draw characters on top of the device. However, other stroke-based text entry approaches are also possible (e.g. Palm OS' Graffiti). Additionally, if the touch sensitive area incorporated a physical border, EdgeWrite offers several advantages including physical stability and tactility [15].

EXAMPLE USES

An always-available input mechanism for mobile phones that does not require removing the phone from the pocket is useful in many scenarios. For example, using strokes as the basis for simple gestures, a user could send commands to an application running on their phone through their pocket. Some simple examples are dismissing an incoming call, switching among alert modes, or controlling a music application. A more complex application of through-pocket gestures is playing an audio-based trivia game where a user might respond with a multiple choice answer of 1, 2, or 3. Alphanumeric text entry combined with gesture can also enable scenarios such as following a dismissed phone call with a short text message reply to the caller. *PocketTouch* also enables lightweight querying. For example, a user could search for unread text messages, emails, or voice mails. Feedback could be provided through a Bluetooth headset, headphones, audio from the device [3], near-eye display, or through a vibrotactile actuator [4].

FABRIC ATTENUATION OF CAPACTIVE TOUCH

An obvious concern (and perhaps the most important challenge) for *PocketTouch* is under what circumstances can strokes be reliably detected through fabric. If there is sufficient signal to reliably segment strokes, applications such as gesture recognition and text entry are easily achieved. However, if the signal is too faint, there will be significant gaps in strokes (i.e., tracking loss). Consequently, the system will not be able to disambiguate purposeful strokes from inadvertent touches or noise.

To explore this issue, we collected through-fabric sensor data using 25 different materials. We chose fabrics from typical apparel and purposefully included a range of thicknesses, natural and synthetic fibers, men's and women's garments, types of garments, and pocket location (see Appendix for photos and fabric composition). For each fabric, we measured the relative signal strength of a finger contact. Our method for calculating signal strength was to measure the total capacitance signal from baseline for a contact at four predefined locations on the sensor surface. We gathered 500 samples (approximately five seconds) of data from each location and took the average sum of signal for the contact. To ensure we were touching the same four locations, we used an acrylic guide with four holes that magnetically aligned to our sensor through the fabric (Figure 5). Further, to ensure a consistent signal source, we used a solid metal cylinder with a similar capacitance to a finger.

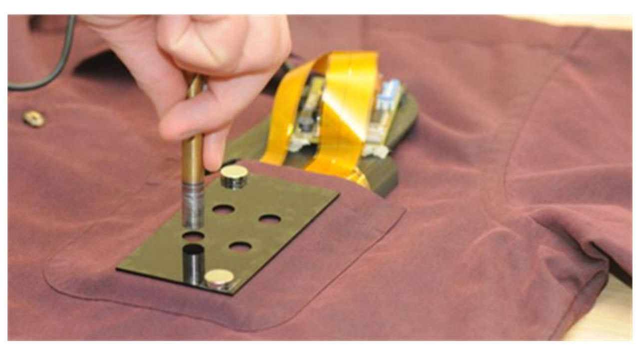


Figure 5. Experimental setup for material attenuation study.

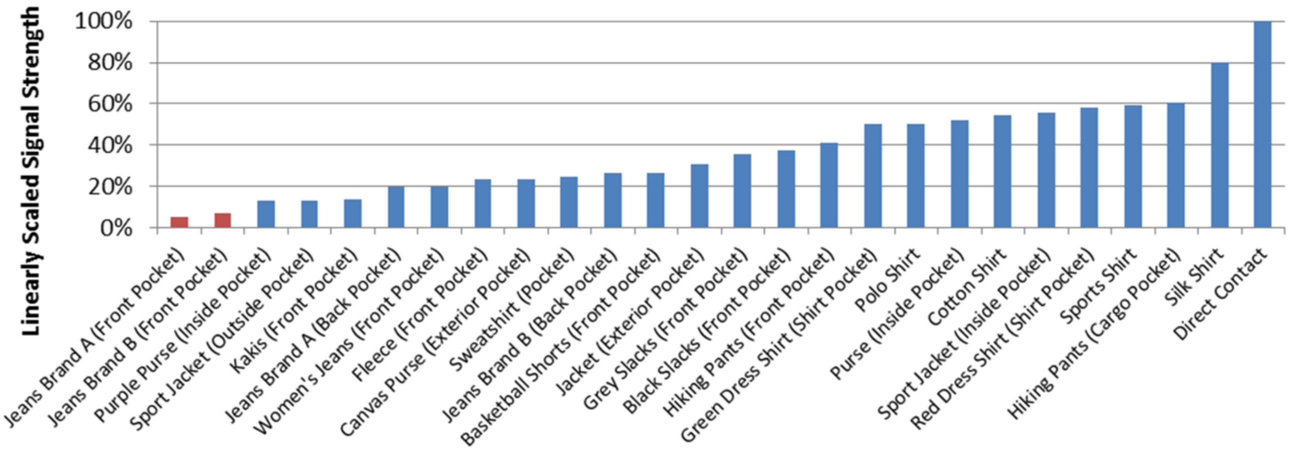


Figure 6. Capacitive signal strength through common fabrics. Signal strength is normalized relative to direct contact. Materials in red failed our text entry test.

This data collection provides a relative ordering of fabrics and a linearly scaled value of signal strength for each fabric (Figure 6). To relate these values to the viability of our prototype, we attempted to enter text through different fabrics. We purposely selected text entry as it is a high bar for evaluation - if such a complex operation can be achieved, a wide variety of applications are possible. Encouragingly, our investigation shows that text entry is robust on all fabrics with signal strength of at least 12.95% - 23 of our 25 test fabrics. On all these fabrics, text entry looks similar to that seen in Figure 4.

Two fabrics proved unreliable for text entry: the front pockets of two pairs of jeans. We were surprised these pockets performed poorly when the back pockets of jeans and much thicker materials (e.g., fleece) performed well. One possible explanation is that the air between the pocket lining and the outer denim fabric attenuates the capacitive signal more than the thickness or the composition of the fabric itself. This serves to demonstrate that the capacitive behaviors of fabrics are a complex interplay of material, thickness, density, layering, and many other factors.

DISCUSSION

Enabling *PocketTouch* on consumer devices will require several considerations. Foremost, it will be necessary to vary the sensitivity of the screen given the physical context (e.g., default sensitivity when operating ―out of pocket” and increased sensitivity when operating ―in pocket”). There are several mechanisms that could be employed to detect whether a device is in-pocket including ambient light level, proximity detection, material detection [6], and self-capacitance measurements on the sensor itself, which could be tuned to detect proximity.

Another important consideration is being able to ―unlock” a device for operation in any orientation. The concerns here are that the system needs to be robust to inadvertent touch contact as well as be able to correctly interpret strokes for hand writing recognition. We suggest that this can be addressed by a single *orientation-defining unlock gesture*. For examples, a single stroke ‗L‘ is unlikely to be generated by inadvertent contact [5] and also implicitly indicates the device‘s orientation. This enables recognition of orientation dependent strokes (e.g. the characters ―d” and ―p”).

CONCLUSION

PocketTouch enables capacitive touch-based gesture control and text input on a mobile device *through* fabric enclosing the device (e.g., a pocket or bag). Our results demonstrate that with many of the textiles used in today‘s garments, *PocketTouch* is a viable input method for quick eyes-free operation of devices in your pocket.

REFERENCES

1. Anthony, L. & Wobbrock, J.O. 2010. A lightweight multistroke recognizer for user interface prototypes. In *Proc. Graphics Interface 2010* (GI '10). 245-252.
2. Ashbrook, D.L., Clawson, J.R., Lyons, K., Starner, T.E., & Patel, N. 2008. Quickdraw: the impact of mobility and on-body placement on device access time. In *Proc. CHI '08*. 219-222.
3. Blattner, M., Sumikawa, D. & Greenberg, R. 1989. Earcons and icons: Their structure and common design principles. *Human Comp. Interaction* 4, 1 (1989), 11-44.
4. Brewster, S. & Brown, L.M. 2004. Tactons: structured tactile messages for non-visual information display. In *Proc. AUIC '04*. 15-23.
5. Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M., Balakrishnan, R. 2006. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In *Proc. CHI '06*. 861-870.
6. Harrison, C. & Hudson, S.E. 2008. Lightweight Material Detection for Placement-Aware Mobile Computing. In *Proc. UIST '08*. 279-282.
7. Hudson, S.E., Harrison, C., Harrison, B.L., & LaMarca, A. 2010. Whack gestures: inexact and inattentive interaction with mobile devices. In *Proc. TEI '10*. 109-112.
8. Karrer, T., Wittenhagen, M., Lichtschlag, L., Heller, F., & Borchers, J. 2011. Pinstripe: eyes-free continuous input on interactive clothing. In *Proc. CHI '11*. 1313-1322.

9. Orth, M., Post, R., & Cooper, E. 1998. Fabric computing interfaces. In *Proc. CHI '98*. 331-332.

10. Rekimoto, J. 2002. SmartSkin: An Infrastructure for Free-hand Manipulation on Interactive Surfaces. In *Proc. CHI '02*. 113-120.

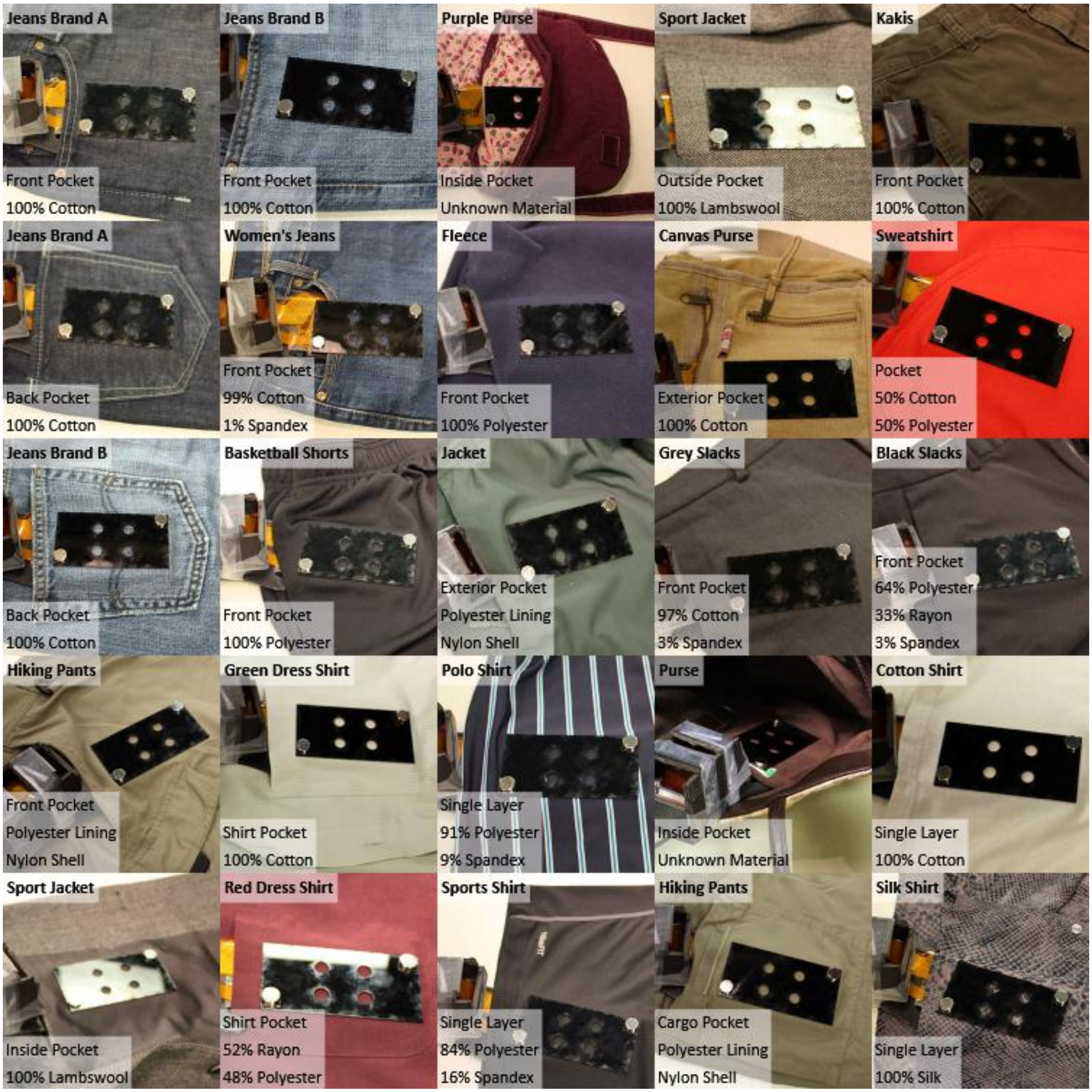
11. Ronkainen, S., Häkkilä, J., Kaleva, S., Colley, A., and Linjama, J. Tap input as an embedded interaction method for mobile devices. In *Proc. TEI '07*. 263-270.

12. Saponas, T. S., Tan, D. S., Morris, D., Balakrishnan, R., Turner, J., and Landay, J. A. Enabling always-available input with muscle-computer interfaces. In *Proc. UIST '09*. 167-176.

13. Song, H., Benko, H., Guimbretiere, F., Izadi, S., Cao, X., and Hinckley, K. Grips and Gestures on a Multi-Touch Pen. In *Proc. CHI '11*. 1323-1332.

14. Villar, N., Izadi, S., Rosenfeld, D., Benko, H., Helmes, J., Westhues, J., Hodges, S., Butler, A., Ofek, E., Cao, X., and Chen, B. 2009. "Mouse 2.0: Multi-touch Meets the Mouse". In *Proc. UIST '09*. 33-42.

15. Wobbrock, J.O., Myers, B.A. and Kembel, J.A. 2003. EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proc. UIST '03*. 61-70.



Appendix: The 25 materials from our fabric attenuation comparison, including test location and material composition.