Printing Multi-Key Touch Interfaces

Çağdaş Karataş
Rutgers University
North Brunswick, NJ 08902,
USA
cagdas@winlab.rutgers.edu

Marco Gruteser
Rutgers University
North Brunswick, NJ 08902,
USA
gruteser@winlab.rutgers.edu

ABSTRACT

We present a technique for creating multi-key conductive ink touch user interfaces that can be printed on paper in a single pass. While 3D printing and open-source electronics platforms have led to enormous creativity in creating smart objects, the means for user interaction with such objects are often limited and require remote interaction through a smartphone app. Paper-based touch circuits are a convenient medium for exploring custom touch sensors that can be attached to numerous objects in our environment. The challenge lies in creating a reliable and customizable touch circuit that is easy to produce. Specifically, it should not require assembly of multiple layers and it should support multiple touch points without needing separate connections to a microcontroller for each touch point.

We address this through a resistive touch sensor that exploits the inherently high resistance of printed traces to create multiple detectable touch points. The finger closes the circuit when in contact with the touch point and the sensor uses a polarity-switching technique to cancel out the effect of the unknown skin resistance. We evaluated the touch sensor using keypads with 10, 15 and 20 touch points and achieved 99.6%, 93.5%, and 91% touch detection accuracy, respectively. We also observed touch detection rates of up to 154 touches per minute.

Author Keywords
Paper Keyboard; Conductive Ink Circuits; Flexible Input Interfaces;

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation : Miscellaneous; H.4 Information Systems Applications: Miscellaneous

INTRODUCTION
Embedding an ever increasing number of smart devices into our surroundings will create a need for more ubiquitous user interfaces that let us interact with these devices. As we have transitioned from PCs and laptops to small mobile devices, touch sensing has proven itself to be an especially compelling means of interaction with smaller devices. What if we could touch-enable a much richer set of objects and surfaces in our environment? One could imagine custom buttons on car dashboards that control devices designed long after the car was manufactured, remote control functions in clothing or couch armrests, lighting control and telepresence functions through buttons on the conference table, or new generations of interactive children's books that do not require bulky reading devices.

Challenges. The main challenges lie in creating touch sensors that can be easily produced and customized for different applications. The touch sensor design should enable exploration in this area, similar to how 3D printing and open electronics platforms have led to enormous creativity in the design of smart objects. In particular, touch sensors should be customizable in size and shape, so that they can easily be affixed to objects of different proportions. Moreover, a touch sensor should accommodate various number of distinguishable touch points to allow creation of interfaces with multiple buttons or keys. Design, production, and particularly reproduction of these keyboards should not require specialized hardware knowledge. They should be straightforward to integrate with custom electronics or smartphone apps such as IFTTT [1].

Existing solutions. Earlier research has studied a variety of technologies for touch-enabling surfaces in our environment. First, projection and camera systems have been used to create virtual touch points on surfaces [12, 19]. This usually requires careful setup and alignment of the camera system and is susceptible to occlusion. Second, touch surfaces have been embedded into objects with custom circuitry. Examples are wearable keyboards [4] or touch points on tables. This again requires careful engineering and instrumentation for a specific object. Third, audio sensing of touches has been used to create a virtual keyboard on a table[18] but it requires careful fingerprinting of the audio environment during initialization time and the process will need to be repeated if the nearby objects are moved. Only recently, researchers have explored printing of touch sensors through conductive ink. The existing work is limited to a single button interface [16], or requires assembly of multiple printed layers and connection of a large number of ADC lines to an electronic circuit [5]. Either the level of customization or the ease of production and reproduction is therefore limited in these approaches.

Approach. To address these challenges, we propose a resistive polarity switched touch sensor design that can be printed in one pass on paper using conductive ink. It separates the
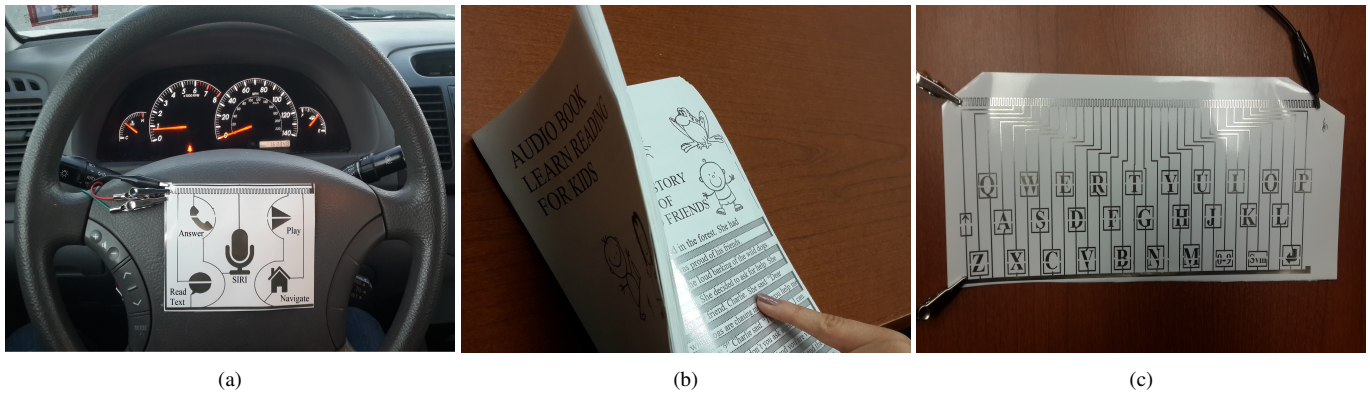
<center>(a) (b) (c)</center>

Figure 1: Example prototypes of printed touch interfaces.

necessary readout electronics components into a small attachable clip device in order to avoid mounting electronic components on paper. Single layer printing is possible through a resistive touch sensor design wherein the finger acts to close the circuit. A key advantage of the resistive polarity switched design is that the layout and number of touch points can be modified, without requiring any redesign of the clip device or a change in the number of interface lines between the device and the paper. Specifically, the current prototype allows the creation of up to 10-15 reliable touch points with the three GPIO pins on the clip device. This is in contrast to other more common touch sensor designs, where adding more touch points requires more GPIO pins on the readout interface. The approach supports multiple touch points by exploiting the inherently high resistivity of printed traces ($0.19\Omega/\square$ in our configuration)[1] to create a voltage divider circuit layout that allows distinguishing touch points based on different read out voltages. This requires a sensing technique that is independent of the unknown skin resistance at the finger, however. Polarity switched sensing achieves this by implicitly canceling out the effect of finger resistance.

The approach builds on advances in conductive ink printing which now allow for printing of conductive traces on plain photographic paper with standard home-use inkjet printers at relatively low cost (ink prices of a few hundred US dollars per 100ml). This makes it possible to create touch circuits on a paper substrate that is flexible and thus able to conform to objects that are not flat. We have also developed layout design software that can create the necessary printed traces based on a simple specification of the location of the touch points. The keystroke recognition method is also resource-efficient in terms of computing power and hardware, allowing the design of long-lasting, small form factor clip devices that wirelessly connect to the devices they control (e.g., a smartphone or light switch).

Example applications. To illustrate some of the possible applications for prototype printed touch sensors, we have presented three sample prototypes in Figure 1. Many printed cir-

cuits replace or extend the capabilities of existing input solutions, such as car dashboards, bulky tablet add-on keyboards, or DJ controllers. Users may want to create their own physical shortcut keys to be placed in a convenient location. This is illustrated through the customized interface for a driver's phone on the steering wheel 1(a). While some cars already include certain functions to control the phone over Bluetooth, printed interfaces allow for a more personalized experience and allowing older cars to keep up with newer phone functions. A touch sensor in an interactive children's book is demonstrated in Fig 1(b). Other examples in this class may include touch-sensing surveys and ballots. Fig 1(c) stretches the support for multiple touch points to create an entire rollable and foldable keyboard. These examples also illustrate how it can be convenient to use the conductive ink to print touch points in the form of recognizable icons. While not recommended by the manufacturer of the conductive ink we used, other inks and printing processes may allow for overlaying the touch circuits with printed imagery using regular ink.

In summary, the salient contributions of our work are itemized below:

- simplifying the use of conductive ink printing for multi-button custom touch input interfaces that augment objects in our environment.

- introducing a touch circuit design that can be printed in one pass and supports multiple touch points (i.e., buttons) without changes to the external device's hardware or its 3 pin interface to the paper circuit.

- designing a resistive polarity switching touch detection technique to improve touch detection accuracy and allow scaling to 20 touch points with the three pin connection from the paper to the external hardware module.

- developing a prototype readout circuit and layout guidance to create custom paper user interfaces.

- evaluating the accuracy of touch detection and its dependence on several factors such as skin resistance and size of the touch point.

---

[1] The standard unit for sheet resistance is $\Omega/\square$ which is dimensionally equal to Ω. However, $\square$ is used to discriminate it from regular resistance.

Voltage Divider

(a) Basic voltage divider touch circuit.

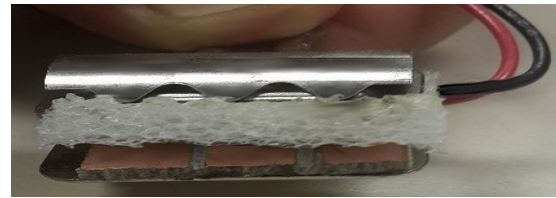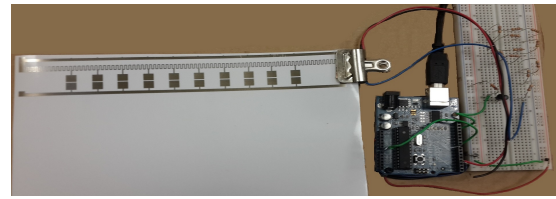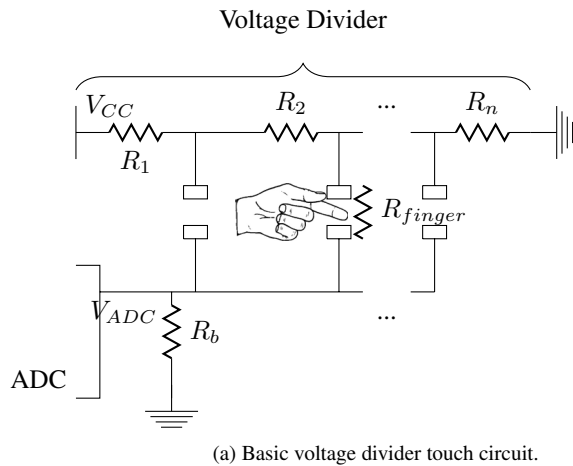(b) Printed voltage divider with padstacks for buttons.

(c) Connector clips

Figure 2: Implemented circuit.

BACKGROUND AND RELATED WORK

The need for reliable and accurate input interfaces has led to a tremendous effort in the research community. The research in this field has resulted in great improvements in touch-screen technology including analog resistive, surface capacitive, projected capacitive, surface acoustic wave, infrared(IR) and optical technology, to mention a few. However, power and portability requirements limit the screen size of mobile devices and the interaction surfaces on them.

Therefore, there has been a great amount of work put into moving touch interfaces out of the smartphone footprint [12, 4, 3]. In Canesta keyboard[12], users input text by pressing keys on a projected image of a keyboard, and a sensor module captures the intersection of fingers with an IR light plane emitted from IR light source. Several works proposed wearable keyboards (e.g., [4]) that detect key presses from kinematic sensors placed on fingers. Moreover, Elfekey and the colleagues [3] proposed a 4-layer thin keyboard that exploits the AC hum phenomenon that leads to a small AC current on the human body due to AC noise from the environment. However, sufficient AC noise may not exist in all locations.

Analog resistive keypads and touchscreens have been widely used by the industry [15] mainly because their recognition approach enables detecting multiple keys with a single sensing line. In a typical circuit, a reference voltage is applied across a voltage divider resistor with multiple stages in which pressing a key closes the circuit and brings a stage in contact with the sensing line. Since, the voltage is different at every stage (V_i at i^{th} stage), the key that is pressed can be found by measuring the voltage on the ADC line. A similar principle is applied to touch screens to find the 2D position of the finger by using two resistive sheets separated by spacers. Unfortunately, mechanical switches and two layer approaches require extra assembly and production effort.

To provide ubiquitous keyboards and touch surfaces, researchers have also explored vision and audio sensing approaches. Visual panel [19] uses computer vision techniques to detect a paper keyboard and the relative position of a fin-

ger tapping a key on the paper. This method, however, is computationally intensive and requires a carefully positioned camera. Wang and colleagues proposed using [18] using two microphones employed in a smartphone and detecting a keystroke's location from multipath fading features of the audio signals. While this technique could enable a new method of human-computer interaction, it requires keyboard software training with every movement of the keyboard or nearby objects. Overall, none of these approaches lends themselves well to instrumenting our environment with touch surfaces.

More recent work has also explored conductive ink to print special conductive patterns onto a paper to detect finger touch for creating touch user interfaces. Touch events on the pattern can be detected with capacitive [13, 16, 10, 11] sensing techniques as well as through resistive graphs [7]. Resistive graphs use a similar approach as our method and they allow touch sensing in 2D. Although these techniques come into prominence with different features such as integrated displays [11], they all require special multi-layer substrates to find the exact touch position and, therefore, the ease of production and the level of customization are limited. Our polarity-switching technique, in comparison, enables finger touch detection in single layer, such as on a sheet of paper printed on one side. An alternative capacitive sensing technique [5, 8] uses similar conductive ink and a single-layer paper substrate as in our proposed interface. Gong and colleagues [5] are also able to sense pressure by using skin resistivity. However, it needs a separate set of input pins from the microcontroller for every touch point to detect the position of the finger. This requirement means connecting a large number of tracks from the paper to the off-paper microcontroller, which has implications on the ease of connection and the size of the off-paper device. Hodges and colleagues [6] achieved limiting the number of the tracks by connecting touch points with one-wire and I2C bus-based protocols. Still, this requires additional electrical components for extra touch points, which limits scalability. We describe how we address the limitations of these earlier works next.
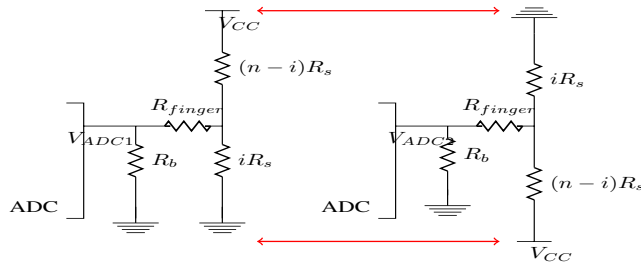
Figure 3: Key recognition and Finger resistance calculation by switching voltage divider reference voltage

Figure 4: Histogram of finger resistance values taken from 15 people at different times, $27M\Omega$ and $65M\Omega$ are taken from the same person.

SYSTEM OVERVIEW

In this section, we detail the challenges unique to the printing medium and introduce the polarity-switched resistive touch sensor system as a way to overcome these challenges.

Challenges

Designing customizable and easily printable touch sensors involves several challenges due to restrictions in the printing medium.

Single Layer. The touch sensing circuit should be in one layer so that it can be printed from an inkjet printer in one pass. Several layers would introduce additional complexity in combining these layers. In particular, connections across layers are difficult since vias and drilling are not easily achieved on a paper substrate.

No mounting of circuit components. The sensor should not require mounting of external circuit components, which consumes excess time, restricts flexibility (the paper becomes less bendable), and introduces protrusions that may be easily ripped off during use.

Multiple touch points or keys. The touch sensor design should support printing multiple distinguishable touch points or button on the same circuit. This will allow, for example, creating the five icons with different functions as depicted in Fig. 1(a).

Easy connectivity. The touch sensor design should allow for straightforward connectivity with mobile devices or devices in our environment. Existing printable touch sensors often connect printed touch points to an off-paper readout and communication circuit to accomplish this task. This off-paper device can include radios to connect to other devices, but the connection between the paper sensor and the off-paper device introduces a new scalability challenge. As the number of touch points increases, the number of required sensing lines from the paper to the off-paper device goes up. The touch sensor design should minimize this complexity.

Approach

To eliminate mounting of circuit components, we have moved all other necessary items, such as the microcontroller, the amplifying transistors, and the high-value resistors, into a separate off-paper module. This module could be designed
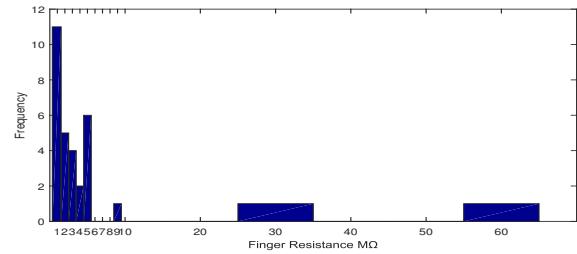
in the form of a binder clip that can be attached directly to the printed tracks. Unfortunately, a typical resistive touch sensor design requires two layers of electrodes, which are brought into contact through the pressure from the finger. To allow for single layer printing, we designed a resistive approach where the sensing circuit is open at the touch point and closed through the finger when touched. While straightforward in principle, the challenge lies in allowing for multiple distinguishable touch points without increasing the number of printed tracks that need to be connected back to the off-paper module. We address this through two key ideas.

Touch Detection with Polarity Switching

First, we exploit the inherently relatively high resistivity of printed tracks to create a voltage divider circuit. The sheet resistance ($0.19\Omega/\square$) of the conductive inks is sufficiently low enough to use as a conductive trace while still being higher than regular copper tracks. While often considered a disadvantage, this allows us to print the basic voltage divider touch circuit shown in Fig. 2a without requiring the mounting of external resistor components. We only need to ensure that the printed tracks are long enough to create the desired resistances R_1 to R_n. To elongate the tracks, we printed a snaking track, as shown in Fig. 2b (note the square-wave-like pattern at the top). Given constant skin resistance, this basic voltage divider circuit will allow for measurement of distinguishable voltages at the ADC line, depending on which touch point was pressed. This means that multiple touch points can be created with only three tracks to the off-paper module (Vcc, GND, and ADC). In practice, however, skin resistance varies significantly over time, from person to person, and for different tapping gestures.[2]

Second, we therefore designed a polarity-switching technique that can cancel out the effect of skin resistance. Its key idea is that the ratio of two voltage measurements, the regular one and the one where Vcc and GND are swapped, is independent of the skin resistance. We will detail this technique further below.

Together, these techniques allow us to create multiple touch points by printing only a single layer and clipping a device onto the paper so that it connects to the three interface tracks. It thus addresses the challenges above. The number of tracks

---

[2]We have observed skin resistances between 0.5 - 65 $M\Omega$.
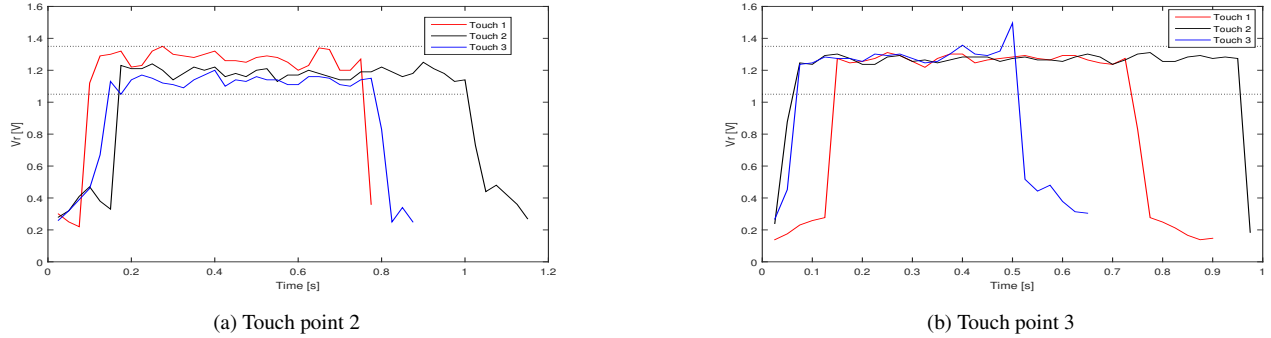
(a) Touch point 2

(b) Touch point 3

Figure 5: Voltage measurements

on the connection interface remains the same, regardless of the layout and number of touch points on the printed circuit, which allows for the designing of different touch sensors without changing the clip hardware. We have also experimented with capacitive sensing but ultimately chose the resistive approach because we found it to be more robust with multiple touch points and only require a single sensing line to the ADC.

In our solution, we use the finger as a circuit element in place of mechanical switches. The finger acts as an inconsistent and high-value resistor R_{finger} that connects the stage of the voltage divider to the ADC line. Both the inconsistency and high-value characteristics of R_{finger} create challenges. First, even a small current drawn from the analog to digital converter (ADC) pin of the microcontroller will create a significant voltage drop across the finger due to the high resistance of the finger. Therefore, the voltage drop is proportional with finger resistance. The measured voltage is also affected by the pull-down resistor R_b, which is used to ensure zero voltage at the ADC when there is no finger connection. The measured voltage by the ADC is thus equal to $V_{ADC} = \frac{V_i R_b}{R_{finger} + R_b}$ where V_i is the voltage at the ith stage of the voltage divider. To illustrate this, Figure 5 shows the voltage measurements when two different touch points are each pressed three times. We can observe that the voltage for touch point 2 varies between 1.05 and 1.35 volts. Unfortunately, the voltages for touch point 3 are within the same range. This means the changing finger resistance makes it difficult to distinguish the touch points based on these voltage measurements alone.

Estimating finger resistance is difficult since there are many psychological [9], physiological [17], and environmental factors that affect skin resistance. During our experiments we encountered a wide range of finger resistance values. In Figure 4, we show the histogram of finger resistances encountered when about 15 students have touched our keypad at different times. As we can see, more than 93% of the finger resistance values were less than $10M\Omega$ but two outlier measurements of $27M\Omega$ and $65M\Omega$ were observed (from the same person incidentally). Therefore, we extend the method to better accommodate different skin resistances.

Our method eliminates the dependence on skin resistance by switching the polarity of the reference voltage lines that feed the voltage divider and taking a second measurement. The measurements are time multiplexed during a single touch. Using these two measurements it is possible to determine the touch point independent of finger resistance. We use this approach since simply calibration R_{finger} with a separate independent circuit suffers from two problems. First, the resistance measuring circuit should not affect the measurements of the primary touch point detection circuit. Second, there is no room to route a second independent circuit on a single layer paper.

Figure 3 illustrates the polarity switching approach. It shows two copies of the circuit from Figure 2a with opposite polarities. Note that the voltage divider is abstracted here for the case where the ith touch point is pressed on a n touch point input interface. If we neglect the current drawn by the ADC and assume that the finger is not grounded, we can represent the measured voltages as follows.

$$V_{ADC1} = \frac{V_i R_b}{R_{finger} + R_b} \qquad (1)$$

Similarly, we obtain the following from the second measurement.

$$V_{ADC2} = \frac{V_{n-i} R_b}{R_{finger} + R_b} \qquad (2)$$

Combining and simplifying the equations (1) and (2), yields

$$\frac{V_{ADC1}}{V_{ADC2}} = \frac{i}{n-i} \qquad (3)$$

This ratio of the two ADC voltage readings only depends on the point that the finger touches and eliminates the dependence on skin resistance and pull up resistor. To illustrate this Figure 6 shows the $\frac{V_{ADC1}}{V_{ADC2}}$ ratio for the voltage measurements from Figure 5. Note that the ratio values are considerable more consistent across touches and remain between
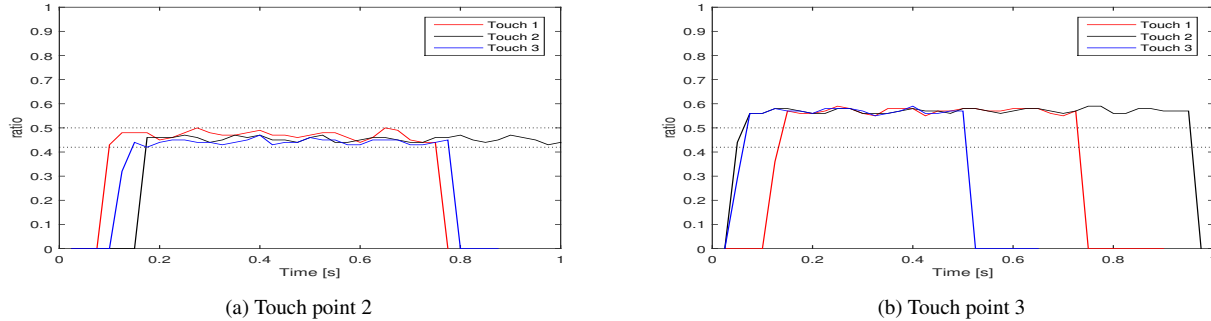
173

(a) Touch point 2

(b) Touch point 3

Figure 6: Corresponding ratio values of consecutive voltage measurements.

0.4 and 0.5 for touch point 2. Note also that these values can now clearly be distinguished from those at touch point 3.

Algorithms

We first use a threshold detection algorithm to detect when a finger touches any touch point. Specifically, we detect a touch if $V_{ADC1} + V_{ADC2} > t(V_{cc} - 2V_{be})$, where t is a threshold parameter that we chose as 0.8 based on empirical data and V_{be} is the voltage drop across the transistor. In theory, the following should hold.

$$V_{ADC1} + V_{ADC2} = V_{cc}\frac{R_b}{R_{finger}} - 2V_{be}$$

The conservative threshold therefore accounts for noise and the simplified model due to unknown finger resistance. When a touch is detected we consider V_{ADC1} and V_{ADC2} a valid sample pair and process it further.

The second stage in our pipeline is a *touch point recognition* algorithm. Touch point recognition algorithm first estimates the touch point for each valid sample pair by finding the the key with the closest expected ratio value to the measured value. The expected ratio values are determined in calibration stage after the keyboard is printed. It further applies filtering over multiple samples to improve robustness. Estimates for invalid sample pairs are assigned to zero. The last n samples both estimates for both invalid sample pairs and estimates for valid sample pairs are held in a buffer. The touch recognition algorithm recognizes a key only if at least half of the members in the sample buffer are valid and agree on a key other than the previously recognized key. Therefore, the key recognition algorithm will not produce duplicate results when the user holds his or her finger on a key constantly.

The third stage applies temporal filtering. The temporal filter holds the output of the key recognition algorithm temporarily and releasing it only if no other key is recognized within a certain time period, the *temporal window*. If another key press is detected within this period, the earlier key detection is discarded and the new one is placed on hold. Therefore, the temporal filter ensures that only the last key recognized is released. This temporal filtering approach reduces spurious key detections at the beginning of a key press event. While the finger is still moving and the measured resistance is changing, false detections are especially likely. We have observed that
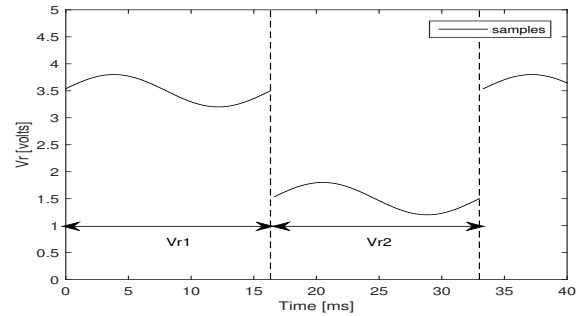

Figure 7: Signal at ADC and AC Hum noise

the last recognized key, when the finger has settled, is usually the correct one and hence the algorithm gives prioritizes the last detection.

The response time of the current system is affected by the measurement period, buffer size and temporal window. In our current settings, we have observed best results when the measurement period is $16.67ms$, the prediction buffer is 6 valid samples requiring at least 4 predictions to agree on a key, and the temporal window is $20ms$. Therefore the time it takes to recognize a touch point is $153ms$. In order to recognize another key, the prediction buffer needs to be flushed by the release of the finger and needs to be refilled for the values of the following touch event. In theory, the current system can recognize consecutive touch events every $286ms$. In practice, our experiments show a minimum of $389ms$.

Calibration

Unfortunately, the printed voltage divider pattern is quite nonlinear in terms of resistance. As a consequence, it is not possible to calculate precise expected voltages and the ratio values for each touch point at design time. These voltage ratio values need to be measured after the circuit is printed. Fortunately, the system needs to be calibrated only once. For this purpose, we have created a small calibration procedure for the microcontroller. During the calibration process, the user is instructed to short the sensing line with each touch point one after another. After ratio values for each touch point are calculated, consecutive ratios are averaged to create safe margins between predefined ratio values.
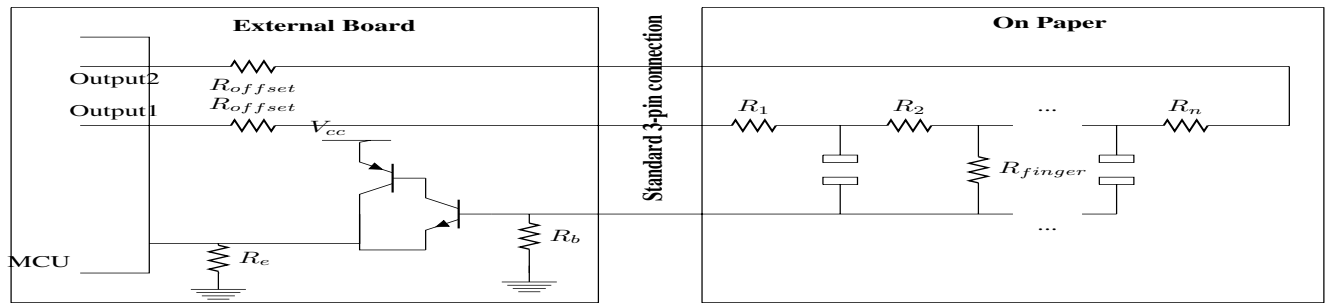
Figure 8: Final Circuit

Layout Considerations

Let us first consider placement of the voltage divider. The voltage divider is in reality a very long trace. To create such a long trace within a small area we have used a square wave pattern. We have chosen the top or bottom of the page for the voltage divider, since we found it less distracting to the rest of the design. It is also important to know the number of the keys that will be placed on the paper. The number of keys is used to find the most efficient strategy for dividing the paper into stages.

Due to the aforementioned voltage drop across the amplifier circuit, the first and last stage should output 0.5V in the first and second measurement periods, respectively. On the other hand, the nonlinear resistivity of a printed voltage divider pattern forces us to use a safer voltage of 0.7V. Since our reference voltage is 5V, most of the first and last 0.7/5 portions of the voltage divider are left as offset margins. The rest of the voltage divider is split equally based on the number of keys.

The last step of the design is placing touch points on the paper and connecting them to the voltage divider pattern. Key patterns must be composed of at least two terminals that will be connected with only a finger touch. We have observed that the minimum space between two terminals should be 0.88mm, otherwise a leakage current can occur between terminals after a touch due to moisture residue from the finger.

IMPLEMENTATION

Implementing this method requires addressing several practical issues that we ignored so far.

First, we assumed that R_{finger} does not change between consecutive voltage readings. This assumption holds when the finger remains steady on a touch point, but R_{finger} actually changes during finger movement due to the change in the contact area. For example, the contact area is very small at the beginning of a touch event, largest at full press and then decreases when finger is leaving the touch point, which continuously changes the resistance. To reduce the change, the time between the consecutive measurements should be minimized. The change of finger resistance between consecutive measurements can also be neglected if the change is relatively small with respect to $R_{finger} + R_b$. For that reason, we used a high-value resistor R_b of $100M\Omega$ in our circuit.

Second, our analysis neglected the current that the ADC draws. In order to minimize this current, we used a Sziklai pair transistor configuration with a current gain of approximately 10000. Therefore, the current that passes through the finger due to ADC is at the $20 - 30nA$ level. However, the voltage drop across the finger due to the ADC current can still be significant for very large finger resistances. The current drawn by the base of the Sziklai pair can be also taken into account by merging equivalent resistance seen from the base of the pair with R_b. To compensate the voltage drop across the Sziklai pair, a $0.5V$ bias voltage needs to be introduced to key voltages which can be done by connecting two serial resistors to both ends of the voltage divider. In our design, we preferred integrating these resistors into the voltage divider on the paper circuit. Therefore, the layout should start placing keys after leaving a certain margin at both ends of the voltage divider. Third, we assume the paper is non-conductive; however, the paper between terminals is highly resistive with a resistance around 500MΩ when they are open. This resistance of the paper becomes smaller and eventually comparable to the finger resistance as the gap between touch points decreases and the number of the keys increase. Therefore, the paper keypad design should maintain a minimum gap size and increase the gap between terminals as the number of keys are increased.

Finally, we assumed so far that the finger acts purely as a resistor rather than a voltage source. Unfortunately, the human body can act as an antenna and can introduce a voltage (noise) into the circuit. The primary source of noise is from the electricity mains which is also called *AC hum noise*. Figure 7 shows the voltage at the sensing line when a key is pressed. Note the clearly visible sine wave due to AC hum noise. In our design, we have solved this problem with a simple trick. By setting the polarity switching frequency to $60Hz$, we can observe exactly one full cycle of AC noise within our measurement period. Since the average of noise within a full cycle is zero, we can eliminate the AC hum noise by taking multiple samples within the measurement period and calculating the average of those samples. Our sampling frequency is $8160Hz$ therefore 136 samples are taken in each measurement period. Since AC hum was the primary noise source in our experiments, we did not use any filter. However, the design could be further improved with band-pass filtering to reduce noise at other frequencies.
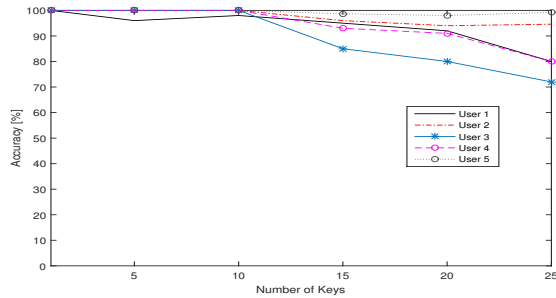
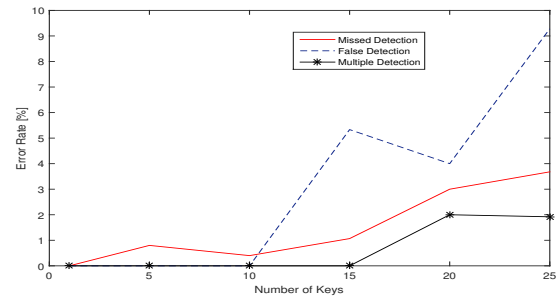Figure 9: The change in accuracy for different users and number of keys

Figure 10: The relationship between error rates and number of keys. The number of incorrectly detected keys becomes the dominant problem for keypads with more keys.

Hardware Platform
We have used an Arduino Duemilanove development board to run the touch point recognition algorithms. The development board has ATmega328 microcontroller and provides 6 ADC input pins and 14 Digital I/O pins. We use 2 of these digital I/O pins for polarity switching and 1 of the ADC pins as sensing line. The polarity switching is implemented by simply setting one of the digital I/O pins high while setting the other one low and then alternating them during the next measurement periods. The amplifier stage is implemented with a Sziklai pair. The Sziklai pair is a compound unit of two transistors of opposite polarities. The current gain of the Sziklai pair equals to the product of the gains of the two transistors. The main advantage of the Sziklai pair over the well-known, equivalent Darlington pair is that the base turn-on voltage is only half of the Darlington's nominal turn-on voltage. This advantage is very important for our application since the turn-on voltage is left as a margin in the $5V$ V_{cc} voltage.

In order to connect paper circuits with the Arduino, we implemented a clip-like connector as in Figure 2c. The connector is composed of a bulldog clip and a copper board and was first introduced in [14]. The copper board is scored with three separations for each connection and serves as a pad.

Printing
To print circuits we have used an Epson WF30 printer, NBSIJ-MU01 ink and resin coated paper from Mitsubishi Imaging. The printed voltage divider resistor (snaking line) is 0.5cm high and 19.5cm long and has approximately $3k\Omega$ resistance. Based on equation (3), however, the actual resistance of the printed voltage divider is not very important as long as it is lower than the resistance of the finger, which is typically on the order of $M\Omega$s. We print touch circuits with different numbers of keys by spacing keys equally along the entire voltage divider.

EVALUATION
The performance of the system is evaluated based on key factors, namely finger resistance, key patterns and their size, number of keys, as well as key press frequency. First, we study the limits of our touch detection circuitry in terms of the number of touch points it can support and the detection delay after touch. Next, we evaluate the detection accuracy with different touch point designs to provide guidance to printable

touch user interface designers. Finally, we validate the proposed method by testing it with different finger resistances and showing the range of finger resistances it can accommodate.

The experiments were carried out in a standard office environment and the paper interface was placed on a wooden desk. The experiments were performed by three male and two female users in their 20s and early 30s. During the experiments, users did not receive visual or audio feedback from the electronic device. They pressed the keys five times and are asked to touch the key fully.

Metrics. Generally, the experiments consider three types of errors. *Missed detection* errors happen when no key is detected after a key press. *False detection* errors happen when the wrong key is detected after a key press. In theory, false detections can also occur when a key press is reported even though no key is pressed. In practice, however, we have never observed this type of error in our experiments after proper calibration. All reported false detections are therefore errors where the wrong key was reported. Finally, *multiple detection* errors happen when the algorithm produces multiple keys after a single key press. We define the overall *accuracy* as the percentage of key touches correctly recognized, that is touches where none of these errors occur.

Number of keys
In order to understand the limits of the paper keypads in terms of number of keys, we have conducted experiments with printed circuits with 1, 5, 10, 15, 20, and 25 keys. The touch points use a rectangle pattern and the 10 key setup is shown in Figure 2b. Each user presses every key five times.

Due to performance variations between users, Figure 9 illustrates the effect of different number of keys on accuracy for different subjects. We observe that across all five users the accuracy remains above 96% for up to 10 keys. For larger numbers of keys, accuracy decreases to as low as 72% for some users but remains above 94% for two of the users.To understand the root cause of this accuracy decline, rates for different error types are given in Figure 10. As the number of keys is increased, false detection and multiple detection errors increase significantly. This is likely due to smaller volt-
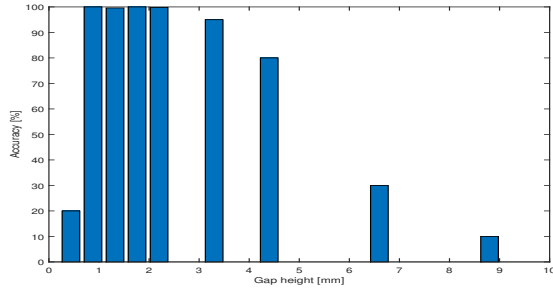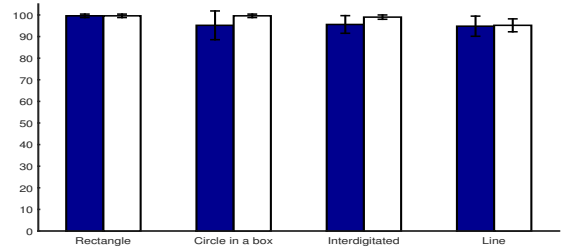
Figure 11: Accuracy for different gap heights

Figure 12: Accuracy for different key patterns and for small (dark blue bars) and large (white bars) pattern sizes. Key patterns are illustrated in Figure 13.

Figure 13: Different button patterns a) Rectangle pattern b) Circle-in-a-box pattern c) Interdigitated pattern d) Line pattern

age margins between the keys (or voltage divider stages) as the number of keys increases. As the finger's touch surface changes during a key press, the measured finger resistance varies and the prediction algorithm's output can swing between keys. As a consequence, the key recognition algorithm either recognizes the key incorrectly or outputs multiple keys for a single key press. We believe that the main reason for instantaneous changes in the measured finger resistance is the fingers motion during a key press.

When considering the average accuracy across all users, we obtain 100% for 1 key, 99.2% (standard deviation [SD] 1.6) for 5 keys, 99.6% (SD 0.8) for 10 keys, 93.5% (SD 4.64) for 15 keys, 91% (SD 6) for 20 keys and 85.1% (SD 10) for 25 keys.

Patterns
Next, let us examine how the shape of the keys affects the performance. We chose four shapes for comparison. The *rectangle key* pattern is chosen over a square pattern because it enables placing enough space between keys and still provides a decent contact area. The second pattern, which we call *circle-in-box* was chosen since its uniform shape ensures a connection between terminals. The third pattern, *interdigitated* pattern, is commonly used on printed circuit boards (PCB) for mechanical buttons. Finally, the last pattern, *line* pattern, can be used in designs where minimalist key patterns are desired. On the other hand, the gap between terminals had to be increased slightly for this pattern in order to make it distinguishable from regular traces. The patterns are illustrated in Figure 13. We also studied how performance changes with size. The aforementioned patterns are printed in small (4.4x8.8mm) and large (6.6x13.2mm) versions. The five users pressed every key 5 times on 10-key keypads.

The best performance attained with the rectangle key pattern for both sizes was 99.6% of the keys are correctly recognized and only two missed detection errors occurred over 500 key presses. The circle-in-a-box pattern and Interdigitated patterns show similar results for both small and large footprints. The circle-in-a-box yielded 95.2% (SD 6.65) in small size and 99.6% (SD 0.8) in large size for different people. The interdigitated pattern resulted in 95.6% (SD 4.08) and 99% (SD 1) accuracy for small and large sizes, respectively. We observed wrong key error in 0.4% of key presses and missed detection errors in 2.65% of the key presses. Although, the

line pattern had a similar shape to the rectangle pattern, it performed worst with 94.8% (SD 4.66) and 95.2 (SD 2.99)% accuracy for small and large key sizes, respectively. False detection errors occurred only three times and the rest of the errors were missed detection errors. The accuracy results are illustrated as a bar graph in Figure 12. We also experimented with the gap height factor on the rectangular pattern and its dependence on terminal size. For this purpose, we prepared 10-key touchpads with the following width, height, and gap height dimensions: 1.1mm to 5.5mm with 0.55mm increments for touchpad width, 1.1mm and 1.65mm for touchpad height, and 0.44, 0.88, 1.32, 1.76,2.2,3.3, 4.4,6.6 and 8.8mm for gap heights .The accuracy was not very sensitive to terminal width and height. For gap height, the average accuracy results are given in Figure 11 for the rectangular pattern.Note that the accuracy remains high for gaps between 0.88-2mm. However, when gaps become larger than about 2.5mm, the accuracy starts to decline and the system misses some touch events. This is expected since the chance increases that the finger no longer covers both terminals sufficiently as the gap gets larger. A second interesting observation is that for gap heights lower than 0.88mm the system no longer reliably detects the release of the finger. We believe that this occurs because of finger moisture left on the paper which compromises the isolating properties of the gap. Although the effect lasts only a few seconds, missed detection or multiple key errors occur during this time. In our experiments, this 0.88mm lower bound on the gap height was independent of the terminals height and width.

In summary, we observed the following. First, large patterns showed better performance than small patterns. Second, the first three patterns showed quite similar performance with over 99% accuracy in large patterns. Third, higher missed detection errors can occur as a consequence of the smaller contact area and larger gap between terminals, such as in the line pattern.

Typing Speed

In our tests, our 10-key keypad was able to detect up to 154 key presses per minute. We believe this rate is enough for most touch user interface applications and would even support keyboard typing at moderate speeds. The typing speed of an average smartphone user is 9.94 words per minute (WPM) and 75.85 WPM for a desktop QWERTY keyboard user [2].

Skin resistance

We also examined how effectively our method can cancel out changes in finger resistance. The experiments are performed by imitating the finger with resistors of a known value. In order to achieve this the resistor is connected to the sensing line in series, and the finger touch is simulated by a finger covered with aluminum foil and an isolation layer between the foil and the finger. The aluminum foil connects the two terminals of the key when pressed and the isolation layer eliminates any electrical noise that might leak from the body. We have used a keypad with 10 rectangular keys as in Figure 2a & b). This particular keypad was chosen because of its performance on our key pattern, size, and number of keys experiments. In this setup, every key is pressed 5 times, yielding a total of 50 touches, for each of the 10, 20, 30, 40, 50, 60, 70, and 80MΩ resistors. The results up to 50MΩ are shown in Table 1. Although we observed missed detection errors a few times for each resistance value, these errors occurred when the finger did not stay on the contact area long enough. The other error types are not as frequently observed during these experiments. Overall, the performance is not very sensitive to different finger resistance values up to $50M\Omega$. Above this threshold, however, the voltage drop across the finger becomes too significant and we were no longer able to detect touches for 60, 70, and 80MΩ.

LIMITATIONS AND FUTURE WORK

While we believe that this technique simplifies the construction of prototype touch interfaces, it also imposes several limitations and presents opportunities for further research.

Multi-touch. The current system is limited to a single touch at a time. If n touch points are pressed simultaneously, n finger resistance values and n stage values would need to be calculated. Such a system could potentially be solved with $2n$ linearly independent equations but obtaining independent measurements would require more sophisticated sensing techniques.

Special electrical conditions. As previously highlighted, the system cannot detect finger touch events when the finger acts like a voltage source, is grounded, or shows extremely high skin resistance. When the finger is grounded, the voltage readings on any key will be zero and therefore it will be impossible to identify the touch point. In addition, electrostatic discharge related problems can occur when the finger acts like a voltage source. We have also been unable to detect touches when the finger resistance exceeds $50M\Omega$, however this latter limit could be extended by using an ADC with lower sink current.

Finger Resistance [$M\Omega$]	Accuracy [%]
10	96
20	96
30	98
40	96
50	98

Table 1: Finger resistance and accuracy.

Robustness and ink issues. There are also substrate and ink related problems that may arise. The printed interfaces are not very durable to due to the nature of paper and ink. The paper interface can be easily broken by folding or tearing the paper and the ink can be scratched from the surface. Air moisture or residue from the finger may connect exposed terminals of touch points. However, the problem might be easily solved by increasing the gap height between terminals. Last but not least, potential health implications of long-term exposure to nanoparticle-based inks may need to be further studied.

These issues do not impede prototyping use, however.

CONCLUSIONS

We have demonstrated a method for quick prototyping of ubiquitous touch sensors by exploiting conductive ink printing. In particular, the method allows for multiple distinct touch points to be printed using a single layer circuit that can be printed in one pass. It eliminates the need for assembly or wire connections to a large number of conductive tracks on the paper. Instead, the printed touch sensor can be completed by simply attaching a device in the form of a binder clip to the paper.

We introduced a polarity-switched resistive touch identification technique that supports multiple touch points with only three connections to the binder clip device. Since it is able to cancel out the effect of changing finger resistance, it allows the fingers to be used to close the circuit at the touch point. This in turn enables printing the entire circuit in a single layer.

We have created a prototype of the readout circuit and printed custom touch sensor designs. Our experiments indicate that the sensor achieves touch detection accuracy above 99% with up to ten different touch points and above 90% with 15 different touch points. We found that the effect of the shape of the touch point is relatively small, as long as it is reasonably large enough to be touched with a finger and includes a gap of about 1-2mm. This enables a wide range of arbitrary shapes and touch sensor designs.

We hope that this technology inspires creativity in interaction design and touch-enables our environment, similar to how 3D printing and open electronics platforms have led to an abundance of smart object designs.

ACKNOWLEDGEMENT

REFERENCES

1. IFTTT. `https://ifttt.com/`.

2. Arif, A. S., and Stuerzlinger, W. Analysis of text entry performance metrics. In *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*, IEEE (2009), 100–105.

3. Elfekey, H., and Bastawrous, H. Design and implementation of a new thin cost effective ac hum based touch sensing keyboard. In *Consumer Electronics (ICCE), 2013 IEEE International Conf.e on*, IEEE (2013).

4. Fukumoto, M., and Tonomura, Y. body coupled fingerring: wireless wearable keyboard. In *Proc. of the ACM SIGCHI Conf. on Human factors in comp. systems*, ACM (1997).

5. Gong, N.-W., Steimle, J., Olberding, S., Hodges, S., Gillian, N. E., Kawahara, Y., and Paradiso, J. A. Printsense: a versatile sensing technique to support multimodal flexible surface interaction. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, ACM (2014), 1407–1410.

6. Hodges, S., Villar, N., Chen, N., Chugh, T., Qi, J., Nowacka, D., and Kawahara, Y. Circuit stickers: peel-and-stick construction of interactive electronic prototypes. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, ACM (2014), 1743–1746.

7. Holman, D., Fellion, N., and Vertegaal, R. Sensing touch using resistive graphs. In *Proceedings of the 2014 conference on Designing interactive systems*, ACM (2014), 195–198.

8. Kawahara, Y., Hodges, S., Cook, B. S., Zhang, C., and Abowd, G. D. Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of ubicomp devices. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, ACM (2013), 363–372.

9. Lazarus, R. S., Speisman, J. C., and Mordkoff, A. M. The relationship between autonomic indicators of psychological stress: Heart rate and skin conductance. *Psychosomatic Medicine 25*, 1 (1963), 19–30.

10. Olberding, S., Gong, N.-W., Tiab, J., Paradiso, J. A., and Steimle, J. A cuttable multi-touch sensor. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ACM (2013), 245–254.

11. Olberding, S., Wessely, M., and Steimle, J. Printscreen: fabricating highly customizable thin-film touch-displays. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, ACM (2014), 281–290.

12. Roeber, H., Bacus, J., and Tomasi, C. Typing in thin air: the canesta projection keyboard-a new method of interaction with electronic devices. In *CHI'03 extended abstracts on Human factors in comp. systems*, ACM (2003).

13. Savage, V., Zhang, X., and Hartmann, B. Midas: fabricating custom capacitive touch sensors to prototype interactive objects. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, ACM (2012), 579–588.

14. Shorter, M., Rogers, J., and McGhee, J. Practical notes on paper circuits. In *Proceedings of the 2014 conference on Designing interactive systems*, ACM (2014), 483–492.

15. Touch, C. Touch handbook. *Carroll Touch, PO Box 1309* (1989).

16. Unander, T., Nilsson, H.-E., and Oelmann, B. Printed touch sensor for interactive packaging and display. In *Polymers and Adhesives in Microelectronics and Photonics, 2007. Polytronic 2007. 6th International Conference on*, IEEE (2007), 12–17.

17. Venables, P., and Mitchell, D. The effects of age, sex and time of testing on skin conductance activity. *Biological psychology 43*, 2 (1996), 87–101.

18. Wang, J. e. a. Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *Proc. of the 12th annual international conf. on Mobile systems, applications, and services*, ACM (2014).

19. Zhang, Z. e. a. Visual panel: virtual mouse, keyboard and 3d controller with an ordinary piece of paper. In *Proc. of the 2001 workshop on Perceptive user interfaces*, ACM (2001).