

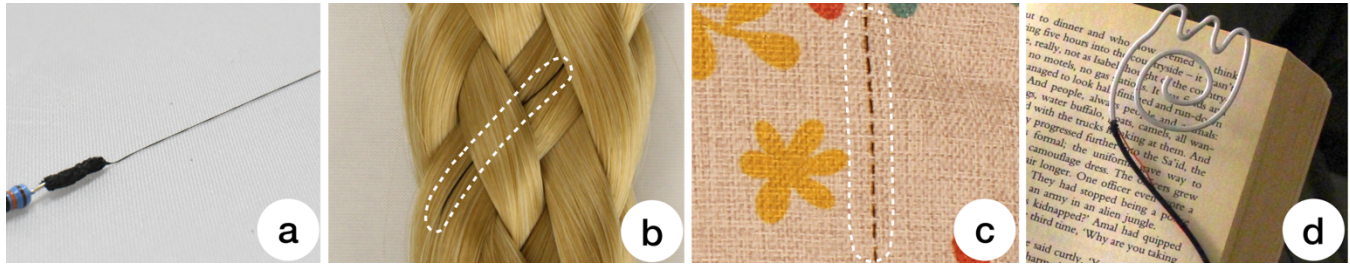
# ThreadSense: Locating Touch on an Extremely Thin Interactive Thread

Pin-Sung Ku<sup>1,2</sup>, Qijia Shao<sup>1</sup>, Te-Yen Wu<sup>1</sup>, Jun Gong<sup>1</sup>, Ziyang Zhu<sup>1,3</sup>, Xia Zhou<sup>1</sup>, Xing-Dong Yang<sup>1</sup>

Dartmouth College<sup>1</sup>, National Taiwan University<sup>2</sup>, University of Science and Technology of China<sup>3</sup>

scott201222@gmail.com

{qijia.shao.gr; te-yen.wu.gr; jun.gong.gr; xia.zhou;  
xing-dong.yang; }@dartmouth.edu



**Figure 1** ThreadSense prototype in (a) a thin thread of less than 0.4 mm thick. Our sensing technique can locate up to two finger touches on extremely thin objects, found in (b) braided hair band, (c) embroidery, and (d) wire craft.

## ABSTRACT

We propose a new sensing technique for one-dimensional touch input workable on an interactive thread of less than 0.4 mm thick. Our technique locates up to two touches using impedance sensing with a spacing resolution unachievable by the existing methods. Our approach is also unique in that it locates a touch based on a mathematical model describing the change in thread impedance in relation to the touch locations. This allows the system to be easily calibrated by the user touching a known location(s) on the thread. The system can thus quickly adapt to various environmental settings and users. A system evaluation showed that our system could track the slide motion of a finger with an average error distance of 6.13 mm and 4.16 mm using one and five touches for calibration, respectively. The system could also distinguish between single touch and two concurrent touches with an accuracy of 99% and could track two concurrent touches with an average error distance of 8.55 mm. We demonstrate new interactions enabled by our sensing approach in several unique applications.

## Author Keywords

Touch input; impedance sensing, thread, fabric

## CSS Concepts

• Human-centered computing ~ Interaction devices

## INTRODUCTION

Today's computing technologies are "weaving themselves into the fabric of everyday life", as Mark Weiser envisioned

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.  
CHI '20, April 25–30, 2020, Honolulu, HI, USA.  
© 2020 Association for Computing Machinery.  
ACM ISBN 978-1-4503-6708-0/20/04...\$15.00.  
DOI: <https://doi.org/10.1145/3313831.3376779>

28 years ago [26]. Innovations in materials [4, 28], sensors [15, 31], and soft electronics [13, 17] are quickly changing the way people interact with computers and what people are interacting with. Touch input, for example, has gone from the rigid body devices like touch screens and found its way in one-dimensional (1D) soft sensors (e.g., stripe, cords) that can bring interactivities to everyday objects such as drawstrings [10, 12] and headphone wires [12, 23].

In this paper, we introduce a new approach for locating up to two concurrent finger touches on a thin thread made of conductive material using the principle of impedance sensing. This approach works on cords, stripes, and more importantly, *very thin* threads (less than 0.4 mm thick) with a simple structure (a single line of resistive material) that cannot be supported by existing methods using pressure [13], capacitance [23], or time domain reflectometry [27]. Thus, our new approach extends touch input to everyday *thin* threads, such as tinsels, braids, wire crafts, or embroidery (Figure 1b, 1c, 1d).

Our method is also unique in that it employs a model-based approach, which locates touches based on a mathematical model describing the change in the impedance of the thread in relation to touch locations. Unlike many of the existing techniques also using impedance sensing [29–31], our system does not need to be trained. The user only needs to perform a quick calibration by simply touching a known location(s) on the thread to inform the system of their finger impedance, needed later for locating a touch.

The simplicity of the calibration allows the system to quickly adapt to various environments and users. Environmental noises such as AC voltage can vary across environments (e.g., kitchen, cars, offices) because of different types of surrounding objects (e.g., power lines, metal objects). Such noises cause ground coupling with the thread and can affect

impedance sensing. Additionally, finger impedance can vary across users because of differences in their body impedance. It can even vary for the same user with different fingers or finger conditions (e.g., wet or dry). Prior designs on impedance sensing would likely require repeated training with tedious data collection in each different settings [31]. Our system, however, can adapt to these variations by only requiring the user to touch the thread one or a few times depending on the accuracy needed for an application.

We demonstrate the effectiveness of our approach through a proof-of-concept prototype (called ThreadSense) developed in the form of a thin, thermally extruded monofilament via a conventional FDM 3D printer (Figure 1(a)). In a controlled experiment with ten participants, we tested the tracking accuracy of our sensing approach in distinguishing and locating touch positions of one versus two fingers. Our study revealed that the system could sense touch location with an average error distance of 6.13 mm with a single calibration point. Increasing the number of calibration points to five reduced the error distance to 4.16 mm. The system could distinguish single touches from two-touches with an accuracy of 99% and could track two concurrent touches with an average error distance of 8.55 mm.

Our contributions are: (1) a sensing technique for locating touch input on an extremely thin thread; (2) a model-based approach that enables the system to work based on a simple calibration process; (3) the result of experiments to demonstrate the effectiveness of our approaches.

## RELATED WORK

This work builds and extends upon prior work in a number of domains, including capacitive and impedance sensing and 1D touch input on interactive stripe, cord, and fabric.

### Touch Input on Cords and Stripes

User input on a thin stripe or cord is mainly achieved through touch [12, 22, 23] and deformation [8, 12, 22, 23, 25]. For example, I/O braid [12] allows a user to press or twist a cable to perform input. TactileTape [8] allows the user to use press as input. Both Cord Input [23] and Cord UIs [22] can sense touch and deformation of the cable but the resolution of touch input is limited by the number of capacitive sensing unit. With StretchEBand [25], the user can perform input by stretching a stripe. The works from Sousa, *et al.* [24] and Klamka, *et al.* [10] allow the user to perform input on a cord using a sliding bead. Our work focuses on 1D touch input.

Existing 1D touch sensing techniques are mainly based on capacitive sensing [10, 12, 22, 23]. Although most of them are effective, spatial sensing resolution is limited because the wirings for the large number of electrodes needed for a high spatial resolution are extremely hard to arrange in a very thin form factor. Techniques based on time domain reflectometry (TDR) [27] need a pair of electrodes to be placed side-by-side with a gap of a certain distance (e.g., 0.5 mm in the authors' best example), which limits how thin the thread can be made and its applications. TDR is also sensitive to the

deformation of the sensor, which makes it less reliable for touch input on soft objects (e.g., hair tinsels). The techniques using the charging time of capacitance [6] can potentially locate two touch points on a single line of electrode but the technique has not been explored on a thin thread.

### Touch Input on Interactive Fabric

In addition to stripes and cords, touch input has also been developed on soft fabrics. Existing techniques for touch input on interactive fabric can be divided into the ones based on capacitance sensing [16, 17] and those based on resistance sensing [13, 14, 21].

The class of work utilizing capacitive touch sensing is based on fabric capacitors made of conductive materials acting as electrode plates. On a piece of fabric, the electrodes can be created using conductive threads or inks. Musical Jacket [16], is an example of early explorations in this field. The authors used stainless-steel yarns to embroider a capacitive touch keypad on denim. A more recent work, Project Jacquard [17], describes the design and fabrication of a new type of conductive yarn that can be woven into textiles using standard looms at scale.

The approaches using resistive touch sensing are based on fabric resistors. A sensor structure commonly seen in this category of work has two conductor-layers separated by a semi-conductive middle layer. For example, GestureSleeve [21] is an interactive sleeve that allows a user to perform touch gestures on the forearm. New methods are also under development to improve the resistive sensing technology. For example, Parzer *et al.*'s method can reduce the complexity of the sensor using a new type of yarn comprised of a metallic thread with a resistive coating [13].

These prior works focus on sensing touch on a surface and mostly rely on a grid of threads. Instead, our work aims to sensing 1D touch input with a single thread-form sensor, which is therefore not achievable with previous approaches.

### Impedance Sensing

Electrical Impedance Tomography (EIT) has been employed in many recent work for sensing hand postures [30, 32] and 2D touch locations [28, 29, 31]. For example, Tomo [30] is a wrist-worn device that senses hand postures using EIT. The technique measures interior impedance geometry with eight electrodes inside a wristband to recognize gross hand and thumb-to-finger pinch gestures. Electrick [31] enabled touch input on a wide variety of irregular objects and surfaces using EIT. A technique based on a similar sensing principle has been used to enable the track of finger and pen movements on an interactive paper [29]. iSoft [28] utilized EIT to track real-time continuous touch input and deformation in a flexible sensor.

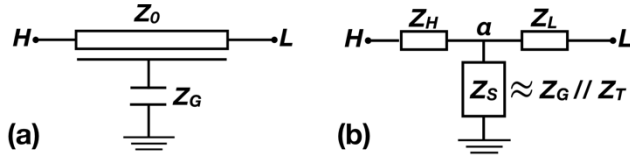
While most of the work mentioned above uses a single AC current frequency, Swept Frequency Capacitive Sensing (SFCS) was shown to provide richer information in sensing hand gestures on everyday objects [19] and differentiating users of a touch-screen device [7]. A more recent work,

Zensei [20], used SFCS to support implicit and ubiquitous user recognition on mobile devices, furniture, and in many of the indoor environments.

In comparison to above works, our work differs in that it provides a new approach to locating touch in an extremely thin 1D space of an interactive thread. Additionally, all the existing impedance-based touch sensing techniques require training and tedious data collection. They use machine learning to classify touch input, thus these systems would likely need to be retrained under various environments and for different users. Our model-based approach replaces training with a simple calibration process, which only requires the user to touch a known location(s) on the thread. Such simple calibration allows the system to quickly adapt.

### SENSING PRINCIPLE

ThreadSense enables a thread made of evenly distributed conductive material to locate user touches. The key rationale is to sense the impact of user touches on the impedance of the thread and infer touch positions. Specifically, when each end of the thread is connected to an electrode, the impedance of the thread can be measured by inserting a small AC current between the electrodes. When a finger touches the thread, it shunts a small amount of current to ground (known as "shunt mode"), which increases the thread's impedance measured at a certain AC frequency. More importantly, when the resistance of the thread is evenly distributed along the length of the thread, the measured impedance is dependent on the touch position on the thread. Thus, one can measure the thread's impedance to locate the finger touch.



**Figure 2** Equivalent circuit (a) without a finger touch and (b) with a finger touching at a location  $\alpha$  on the thread.

To further illustrate the rationale, Figure 2 demonstrates the equivalent circuit with and without a finger touch. Here  $H$  and  $L$  are the high and low voltage endpoints connecting to the electrodes, respectively.  $Z_0$  is the known impedance of the thread and  $Z_G$  is the impedance caused by ground coupling due to environmental noises. When a finger touches the thread at a location, described using a *location coefficient*  $\alpha$  (expressed as a ratio from 0 to 1, with the  $H$  end as 0 and  $L$  end as 1),  $Z_s$  is introduced by the integration of the shunting impedance of the ground ( $Z_G$ ) and finger touch ( $Z_T$ ), which comprises the impedance of the finger and the contact impedance of the finger touch. Note that  $Z_s$  remains constant regardless of the finger touch location. Finally, the impedance of the thread segment separated by the touch location is denoted as  $Z_H$  and  $Z_L$ .

When  $Z_0$  and  $Z_s$  are known, the impedance of the thread  $Z(\alpha)$  (with the finger at location  $\alpha$ ) can be calculated (or estimated) as below:

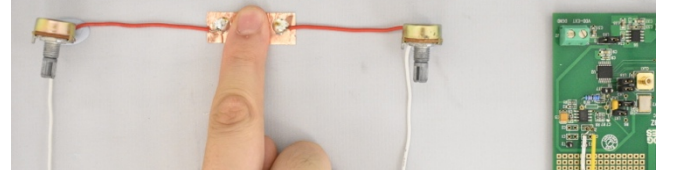
$$Z(\alpha) = Z_0 + \alpha \cdot (1 - \alpha) \cdot \frac{Z_0^2}{Z_s} \quad (1)$$

because the circuit in Figure 2(b) can be converted into a delta network using a star-to-delta transform [9].

Since thread resistance is evenly distributed,  $\alpha$  represents the proportion of the resistance of the thread segment separated by the touch location. For a fixed  $Z_0$  and  $Z_s$ , different touch location  $\alpha$  results in different thread impedance, which ideally should equal to the estimation  $Z(\alpha)$ . This relationship forms the basis for locating touch position.

### Experimental Validation

To validate the relationship between measured impedance and the estimation using Eq. (1), we use a pair of variable resistors to emulate the variations in  $Z_H$ , and  $Z_L$ , caused by a finger touching the sensor. Thus, the ratio of the resistance values of the variable resistors determines the location coefficient  $\alpha$ . We keep the summation of the resistance of them (or  $Z_0$ ) as 1 M $\Omega$ . As shown in Figure 3, the variable resistors are placed on a breadboard and connected via a copper tape representing the touch position.



**Figure 3** Setup of our experimental validation.

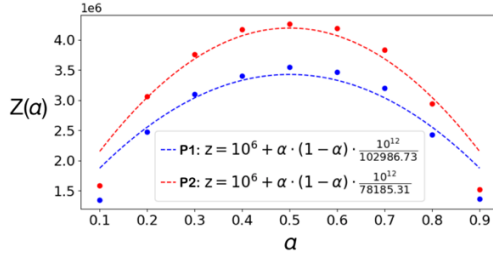
Two participants were recruited in the test and were asked to press their index finger on the copper tape. We measured the resulting impedance at 80 kHz using the AD5933 impedance analyzer chip. We repeated the measurement by varying  $\alpha$  at a step size of 0.1. Figure 4 plots the measured impedance of the participants as  $\alpha$  varies. We fitted the data using a least square fit into Eq. (1) to find out  $Z_s$  and plot the fitted function curve. The data fitted well with the curves, indicating that Eq. (1) is an effective estimation of the effect of  $\alpha$  on the measured impedance. Therefore, if  $Z_T$ ,  $Z_G$ , and measured impedance are known,  $\alpha$  can be calculated to estimate the touch location. Figure 4 also shows that the curves differ between the two participants, suggesting that  $Z_s$  needs to be calibrated individually.

Note that  $\alpha$  is symmetric around the center of the thread. As such, touching the left or right side of the thread is indistinguishable from each other (Figure 4). This issue can be resolved by connecting the thread in series with a resistor of the same resistance (*coupling resistor*). The coupling resistor cannot be replaced by a potentiometer as the internal structure of the potentiometer may introduce noises.

### Challenges

Though we have proved the feasibility of locating a single touch on a thread, challenges exist for detecting two concurrent touch and for the system to work robustly against environmental noises. Detecting multi-touch is challenging

because a single touch may cause a change in the measured impedance similar to that caused by two concurrent touches at different locations. For example, at 84 kHz, the impedance measured with two touch locations at  $\alpha_1 = 0.56, \alpha_2 = 0.64$  is close to that of a single touch with location coefficient  $\alpha = 0.7$ . Further, environmental noises may also affect the measured impedance, making it less reliable for locating touch. We present our design to address these challenges.



**Figure 4** The fitted impedance curve shown in relation to the measured impedance. The impedance increases differently under the influence of different participant's touch, but both fit in with Eq. (1).

### SYSTEM DESIGN

To enable the ability of multi-touch detection and enhance the system's robustness against environmental noises, we consider a *frequency sweeping* approach, where thread impedance measurements are collected as we sweep the frequency of AC current. Since the thread impedance varies under different AC frequencies, frequency sweeping allows us to collect a spectrum of impedance measurements across different AC frequencies. These measurements collectively can be more robust against environmental noises and present distinguishable features between single touch and multiple touches. Frequency sweeping has been used in prior works [19] to distinguish users or recognize hand poses. We are the first to apply the technique to locate touches. Next, we discuss the model-based touch localization method and system calibration.

#### Model-based touch localization

We extend our discussion of sensing principle from a single AC frequency to  $N$  AC frequencies, with each frequency denoted by  $f_i$ , where  $i = 1, 2, \dots, N$ . Since different AC frequencies lead to different thread impedances, we denote the measured impedance for a single touch under  $f_i$ ,  $Z_{f_i}$ . Similarly, we denote the impedance estimated using Eq. (1) under  $f_i$ ,  $Z(\alpha, f_i)$ . Here the problem of locating a touch position is to seek an estimated location coefficient ( $\alpha^*$ ) that minimizes the summation of all differences between the measured and estimated impedance values, as shown below:

$$\alpha^* = \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^N (Z_{f_i} - Z(\alpha, f_i))^2 \quad (2)$$

To search for  $\alpha^*$ , we used the Trust Region Reflective algorithm [11]. As mentioned earlier, to compute  $Z(\alpha, f_i)$ , the system must know the corresponding  $Z_G$  (the impedance caused by ground coupling from the environmental noises) and  $Z_T$  (the impedance of the finger and the contact impedance caused by the finger touching the thread) at

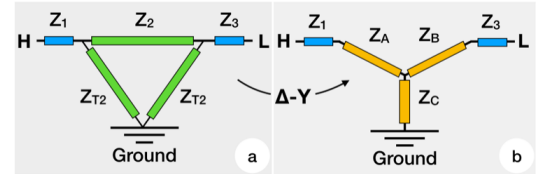
frequency  $f_i$ . We will discuss how this information is acquired shortly in the system calibration section.

A similar approach can be used to locate multiple concurrent touches at different locations. Here we use two touches as an example for the sake of simplicity. When a user touches the thread using two fingers, an equivalent circuit can be described in the form shown in Figure 5a. This circuit can be simplified using a star-to-delta transformations, and the resulting circuit is similar to the one time transformed single touch circuit (Figure 5b). This step is useful as it allows us to calculate the ideal thread impedance for two touches  $Z(\alpha_1, \alpha_2, f_i)$  at different locations and AC frequencies using the method described in Sensing Principle. We show the equation as follow.

$$Z(\alpha_1, \alpha_2, f_i) = \frac{2 \cdot Z_0 \cdot (\alpha_2 - \alpha_1) \cdot Z_{T2,i}}{2 \cdot Z_{T2,i} + Z_0 \cdot (\alpha_2 - \alpha_1)} + Z_0 \cdot \alpha_1 + Z_0 \cdot (1 - \alpha_2) \\ + \frac{(Z_{G,i} + \frac{Z_{T2,i}^2}{2 \cdot Z_{T2,i} + Z_2}) \cdot (Z_0 \cdot \alpha_1 + \frac{Z_2 \cdot Z_{T2,i}}{2 \cdot Z_{T2,i} + Z_2}) \cdot (Z_0 \cdot (1 - \alpha_2) + \frac{Z_2 \cdot Z_{T2,i}}{2 \cdot Z_{T2,i} + Z_2})}{Z_{G,i} \cdot \frac{Z_{T2,i}^2}{2 \cdot Z_{T2,i} + Z_2}} \quad (3)$$

Finally, the estimated location coefficients  $\alpha_1^*, \alpha_2^*$  for a measured impedance spectrum can be computed as:

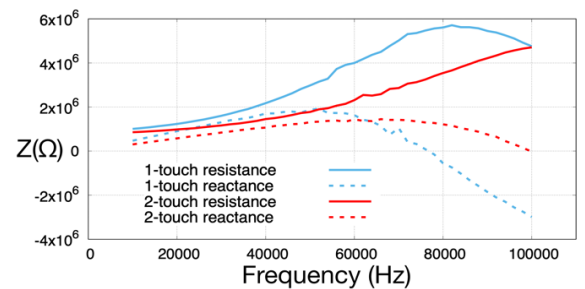
$$< \alpha_1^*, \alpha_2^* > = \underset{\alpha_1, \alpha_2}{\operatorname{argmin}} \sum_{i=1}^N (Z_{f_i} - Z(\alpha_1, \alpha_2, f_i))^2 \quad (4)$$



**Figure 5** Diagram of the equivalent circuit of two concurrent touches simplified into a circuit similar to one touch.

#### Distinguishing Between 1 and 2 Concurrent Touches

Once  $\alpha^*$  and  $(\alpha_1^*, \alpha_2^*)$  are determined, our system is now able to tell which one is more likely the cause of the change in the measured thread impedance. In particular, for a given spectrum of measured impedance across  $N$  AC frequencies,  $\vec{Z}_f$ , whether the occurrence of a touch event is caused by one or two fingers can be determined by choosing between  $\alpha^*$  and  $(\alpha_1^*, \alpha_2^*)$  based on which one has an estimated impedance spectrum closer to  $\vec{Z}_f$ . Figure 6 shows the measured impedance profiles caused by one and two concurrent touches, respectively.



**Figure 6** Measured impedance of a single touch ( $\alpha = 0.68$ ) and two touches ( $\alpha_1 = 0.66, \alpha_2 = 0.74$ ), shown in complex numbers with a real (resistance) and imaginary part (reactance).



Although we only demonstrate the detection and localization of two touches, our approach can be extended to more than two concurrent touches. We leave it for future research.

### Calibration

The goal of the system calibration is to acquire all the necessary information (e.g.,  $Z_G$  and  $Z_T$ ) needed to calculate the estimated impedance using Eq. (1). We calibrate  $Z_G$  and  $Z_T$  separately, because it allows the system to be aware of environment change based on the change in  $Z_G$  alone, enabling new types of interactions (details later). Due to frequency sweeping, we use  $\vec{Z}_G$  to denote the impedance spectrum of the ground coupling effects measured under all the  $N$  frequencies (or *background profile*). We use  $\vec{Z}_{T1}$  and  $\vec{Z}_{T2}$ , to denote the spectrum for the finger and contact impedances for one and two touches respectively (or *touch profile*). Finally,  $Z_0$  is the impedance of the thread without touch (or *thread impedance*), which can be estimated using resistance, measurable using a multimeter, by ignoring the small effect of capacitance and inductance.

#### Calibrating Environmental Noises ( $\vec{Z}_G$ )

As suggested by [1], the effect of ground coupling can be approximated using a distributed capacitance, which can be calculated using the following formula:

$$\vec{Z}_G = \frac{k \cdot (1 - k) \cdot \vec{Z}_0^2}{\Delta \vec{Z}_C} \quad (5)$$

where  $k$  is a constant between 0 and 1 and  $\Delta \vec{Z}_C$  is the increase in the measured impedance caused by the coupling effect. As suggested in [1], we used 0.21 for  $k$  in our calculation. Although this approach overlooks the other parasitic components that may introduce noises, such as the impedance of electrodes at the two ends, we found these noises have negligible impact on system performance in our later experiments.

Note that calibrating the background noises does not need to be carried out manually by a user. Instead, our system is designed to be re-calibrated automatically and periodically (e.g., every 5 minutes). By cross comparing every new background profile with the ones stored in the database, the system can identify if the user is in an unknown environment. This allows the system to notify the user for recalibration if the user is in a new environment.

#### Calibrating Finger and Contact Impedances ( $\vec{Z}_{T1}$ and $\vec{Z}_{T2}$ )

This step of calibration requires a user to touch a predefined location on the thread using the index finger. This is to allow the system to calibrate  $\vec{Z}_S$ . Again,  $\vec{Z}_S$  approximates  $Z_G$  and  $Z_T$  connected in parallel, which can be calculated using Eq. (6), with a pre-determined  $\alpha$ . In our case,  $\alpha = 0.5$  as we asked the user to touch the middle of the thread.

$$\vec{Z}_S = \alpha \cdot (1 - \alpha) \cdot \frac{\vec{Z}_0^2}{\vec{Z}_f - Z_0} \quad (6)$$

Since  $\vec{Z}_S$  approximates  $Z_G$  and  $Z_T$  connected in parallel, when both  $\vec{Z}_S$  and  $\vec{Z}_G$  are known, a single-touch profile  $\vec{Z}_{T1}$  can be calculated using Eq. (7):

$$\vec{Z}_{T1} = \frac{\vec{Z}_G \vec{Z}_S}{\vec{Z}_G - \vec{Z}_S} \quad (7)$$

Like the background profile, user profile can be stored in a database for repeated usages. The touch profile for two fingers can be retrieved in the same manner but with two predefined location coefficients ( $\alpha_1, \alpha_2$ ). We assume that  $Z(Z_{T2,i}, \alpha_1, \alpha_2, f_i)$  (specified in Eq. (3)) equals to the measured impedance  $Z_{fi}$  under every frequency  $f_i$ :

$$\forall i \in \{1, 2, \dots, N\} \rightarrow Z(Z_{T2,i}, \alpha_1, \alpha_2, f_i) = Z_{fi} \quad (8)$$

By calculating the  $Z_{T2,i}$  for every frequency  $f_i$ ,  $\vec{Z}_{T2}$  can be obtained by:

$$\vec{Z}_{T2} = (Z_{T2,1}, Z_{T2,2}, \dots, Z_{T2,N}) \quad (9)$$

Finally, we show in Figure 7 the touch profiles of a user touching the thread using one vs. two fingers.

#### Calibrating Sensor Impedances ( $Z_0$ )

An important requirement of our system is the uniformity of the resistance across the length of the sensor. However, this requirement may not always be satisfied (e.g., due to issues in the manufacturing process). As such, extra calibration can be used to obtain a more accurate impedance curve. The calibration can be performed by a user touching extra pre-determined locations (e.g., three evenly spaced points across the sensor). Once the extra calibration data is collected, a precise mapping can be established between the calculated location coefficient  $\alpha$ 's and the ground truth positions. The curve segments between the calibration points can be interpolated linearly.

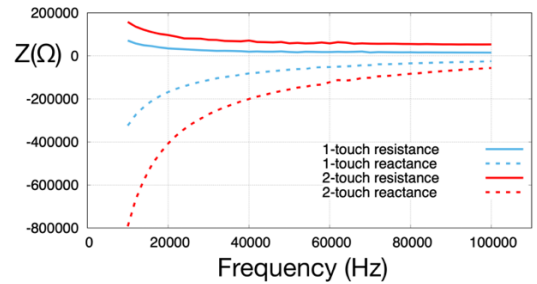


Figure 7 Touch profiles of a user touching the thread using one ( $\alpha = 0.68$ ) vs two fingers ( $\alpha_1 = 0.66, \alpha_2 = 0.74$ ).

### IMPLEMENTATION

To evaluate our approach, we developed a proof-of-concept prototype using a FDM 3D printer and off-the-shelf hardware and software. This section presents our implementation details.

#### Thread Prototype

Like in the previous work [31], the resistivity of the conductive thread cannot be too high or too low. This is because if the resistivity is too high, the electric field will be

too weak to sense the signal. However, if the resistivity is too low, the change in the difference in the impedance caused by the finger touching the thread can be too small to be detected. We found resistances between 500 k $\Omega$  and 1 M $\Omega$  work the best for our impedance analyzer chip (AD5933) from Analog Devices [5]. This requirement unfortunately makes most, if not all, the commercially available conductive threads unqualified for our need. Therefore, we had to create our proof-of-concept prototype using a FDM 3D printer.

The thread was a thin line of conductive filament in 11.5 cm long with a 0.1 mm  $\times$  0.4 mm rectangle cross section (dimension limited by our FDM 3D printer. The thread has a resistance of 432 K $\Omega$ , coupled with a 470k $\Omega$  resistor to extend the sensing range to approximately the length of the sensor. Ideally, the resistance of the thread and coupling resistor need to match or sensing range or accuracy may be affected. The thread was made using a thermally extruded monofilament using a Ultimaker 3 printer [3], with the carbon filament CDP12805 from Proto-pasta [18]. The two ends of the thread are connected to our sensing board using copper wire (Figure 8).



**Figure 8: Our thread sensor prototype, coupled with a 470k  $\Omega$  resistor and connected to the AD5933 impedance sensing board.**

### Impedance Sensing Board

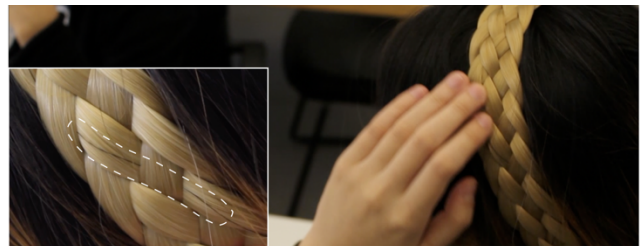
The main components of our sensing board are an impedance analyzer chip (AD5933 from Analog Devices) [2], a 1MSPS analog to digital converter (ADC), and a digital signal processor (DSP), which runs discrete Fourier transform (DFT) and returns the real and imaginary parts of the measured impedance of the thread at a desired frequency. The AD5933 allows impedance measurement from 1 k $\Omega$  to 10 M $\Omega$  with a system error of around 0.5%. The AC frequencies of our sensing board were from 10 kHz to 100 kHz with a 2 kHz interval. This leads to a total 46 samples per cycle, which takes around 10ms to finish. When the sensor is working, a small AC voltage is applied between the two electrodes, and the current flow through the thread is measured using an auto balancing bridge circuit. The impedance between the two ends of the thread is then calculated as the ratio between the input voltage and the output current measured by the sensor. Ideally, the resistor should have the same resistance as the sensor, we were unable to find one on the market. Although this affected sensing accuracy, the implementation was sufficient to demonstrate usage scenarios. Finally, sensor data is sent to a Macbook Pro through USB for further computation. It takes an average of 25 ms and 26 ms to locate a single and two touches respectively.

## DEMO APPLICATIONS

To showcase ThreadSense and its capabilities, we created four demo applications and highlight various usage scenarios. Each application demonstrates the use case of one or more of interactions enabled by our sensing approach.

### Interactive Hair

We implemented our sensing technique on a braided headband to allow gestural input to be carried out as if the user is touching or scratching the hair (Figure 9). Although touch input on the hair has been explored in prior research [5], the state-of-the-art methods can only sense the occurrence of touch. With ThreadSense, the systems can also detect swipe gestures for an extended input vocabulary. The use case for this type of input is broad. In particular, we see its benefit of being less obtrusive in social settings. For example, in the situations where repeated interacting with a smartphone or watch (e.g., checking voicemail messages) can be considered inappropriate. Swiping the hairband is less interruptive to other people since the motion is ambiguous about whether the user is actively using technology or just touching the head. In our implementation, a single touch on the sensor hangs up the phone call, and a swiping gesture changes the phone to mute mode. Although there is only one thread sensor in our current implementation, we foresee that the entire headband can be augmented with our sensing technique in the future with more research efforts. This will enable a much richer set of interactions via the hair.



**Figure 9 A user performs a subtle swipe gesture on a braided head band to interact with a computing device.**



**Figure 10 A user control the TV program using touch input on ThreadSense, hand-sewed on a cushion.**

### Interactive Embroidery

The next application is an interactive embroidery that allows the user to perform input on a soft object covered or made by interactive fabrics. In our implementation, we manually sewed our thread sensor onto an owl embroidered cushion cover (Figure 10). Touching different locations on the thread triggers different actions on a smart IoT device. For example,

touching somewhere near the first owl disables/enables the microphone of the Alexa. Tapping near the second owl using one finger turn on the music. Tapping near the same location using two fingers plays news. Swiping near the same location navigates the menu. This way, the cushion becomes the user's always-available remote controller at the couch and can be used when voice input is not desirable.

### Interactive Wire Crafts

Another example of bringing rich interactivity to objects that are traditionally passive is through wire crafts. In our third application, we developed an interactive tulip bookmark. When in use, the bookmark has a new function that allows the user to perform continuous touch input to control the brightness of the ambient light while reading. For example, the user can slide the finger to increase or decrease the brightness of a lamp (Figure 11).



Figure 11 When reading, a user uses the interactive tulip bookmark to control ambient light.

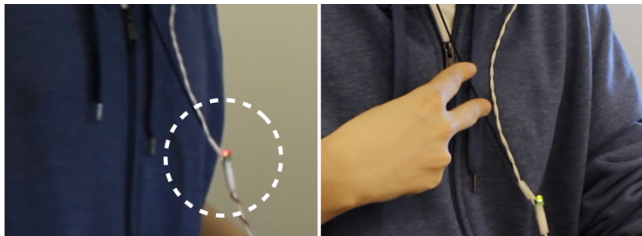


Figure 12 Left: a red light indicates that a user enters a new environment, so that a recalibration is needed. Right: two concurrent touches on the earphone cord to start recording.

### Interactive Headphone Cable

Finally, we demonstrate that all the interactions can be integrated into a single device of a headphone cable to extend input capability of the existing work that cannot sense touch locations [12]. With our implementation, the user can use a single tap to quickly pick up a call. The user can also record a call by tapping the cable using two fingers. Further, the user can use slide with a single finger to control the volume or two fingers to navigate the contact list. A unique feature of our system is that it can recognize the change in the user's environment through the auto-calibration process. When the system detects that the current  $\vec{Z}_G$  is significantly different from the ones in its database, it shines an red light using an LED to notify the user to recalibrate  $\vec{Z}_{T1}$  or  $\vec{Z}_{T2}$ . Once the new environment is calibrated, the profile is stored in the system for later use, and the light turns green (Figure 12).

## EVALUATION

We conducted a system evaluation to measure the accuracy of our sensing approach in locating touch positions and distinguishing touches between one vs. two fingers.

### Participants

Ten right-handed participants (5 female) between the ages of 20 and 25 participated in the study. The average width of participants' index fingers is 14.47 mm; s.e. = 0.18).

### Apparatus

As the sensor is extremely thin and bendable, keeping it straight is important for us to accurately record the touch position(s) as the ground truth. For the study only, we printed the sensor straight on a supporting structure made of non-conductive material. Additionally, we printed the sensor twice as long as its original size. The extra piece was used to replace the coupling resistor to ensure a precise measurement of our sensing accuracy. During the study, the sensor was placed on an empty wood desk in front of the participant, who performed the task in a seating position.

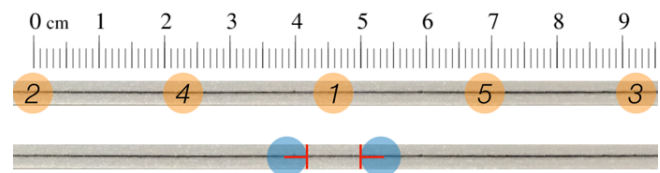


Figure 13 Calibration locations for one (top) vs two fingers (bottom; 8 mm space between the fingers). The numbers indicate the order in which the calibration points were added for analysis on the effect of the number of the calibration points.

### Calibration

Prior to the study, the system was calibrated for each participant using one and two fingers. In the one-finger condition, the calibration data was collected at five locations evenly apart from each other across the sensor (0 mm, 23 mm, 46 mm, 69 mm, 92 mm from the left end) (Figure 13 top). The location distances are calculated using the end near the coupling resistor as the origin. The calibration data was used later to investigate whether and how well the number of touch points involved in the calibration may improve sensing accuracy. In the two-finger condition, the calibration data was collected with the participant touching the sensor using the index and middle finger, approximately 8 mm apart from each other. Participants were asked to center a target position at 46 mm (from the left end) in the middle of the two fingers (Figure 13 bottom). Calibration for two fingers was only performed at a single location.

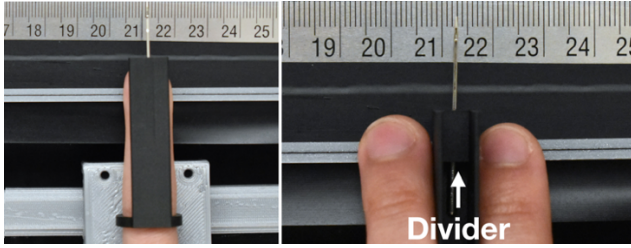
### Data Collection

*One finger.* For the touches using one finger, participants slide their index fingers against the narrow edge of the sensor (0.1 mm) from 0 mm to 92 mm two times, with a sliding distance of 2 mm each. The start and end positions were chosen to avoid them mistakenly touching outside the sensor's sensing region. Participant's right index finger was held in a ring mounted on a slider placed in parallel to the sensor. This allowed them to precisely control the position of



the finger on the sensor. Participants stopped every 2 mm, and the experimenter recorded the ground truth, measured using a ruler (Figure 14). A computer recorded the predicted location.

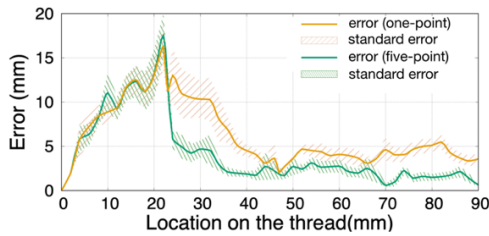
**Two fingers.** For the touches using two fingers, participants repeated the same task used in calibration but at three locations: 23 mm, 46 mm, and 69 mm. At each location, they were asked to center the target location in the middle of the index and middle finger, with the fingers set apart in a distance of 0 mm, 4 mm, 8 mm, 12 mm, or 16 mm from each other (Figure 13). This allowed us to systematically examine the impact of finger distance on sensing accuracy. We used a 3D-printed divider to precisely control the distance between the fingers (Figure 14 right), except in the zero distance condition, where the two fingers were touching each other. Finally, each trial was repeated twice. In total, we collected 920 and 300 samples for the touch events using one and two fingers respectively.



**Figure 14: Study apparatus. Left: one finger condition. Right: two-finger condition, in which the space between the fingers were controlled using 3D printed dividers.**

### Single Touch Accuracy

We used average error distance ( $ED_{avg}$ ) to measure the tracking accuracy of our approach. The  $ED_{avg}$  per participant is defined as  $\frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$ , where  $\hat{y}_i$  is the predicted location, and  $y_i$  is the ground truth, and  $n$  is the total number of trials per location per participant (e.g., 46 locations  $\times$  2 repetitions). We then average the  $ED_{avg}$  across all participants as the final accuracy metric.



**Figure 15 The  $ED_{avg}$  at different touch location under the configuration of one vs five points.**

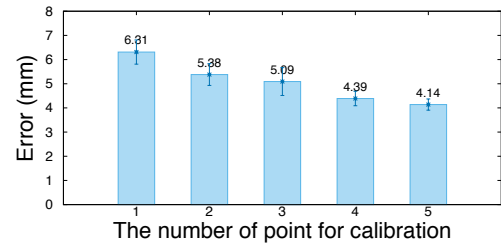
### Sensing Accuracy with Mid-Point Calibration

We first looked at the sensing accuracy achieved using a single calibration point in the middle of the sensing region. Figure 15 plots the  $ED_{avg}$  for the 46 tested locations (0 mm to 92mm, stopped every 2 mm) and the region covered by the stand error. We observe that  $ED_{avg}$  was 6.13 mm (s.e. = 0.26 mm) across all locations. Additionally, the average error

distance is higher in the first quarter of the sensing region ( $ED_{avg} = 5.21$  mm; s.e. = 0.26) than the remaining part ( $ED_{avg} = 8.88$  mm; s.e. = 0.43). Since half of the thread is used as a coupling resistor, this first quarter of the sensing region is close to the half point that separate the thread into two equilibrium impedance ( $\alpha = 0.5$ ). As shown in Figure. 4, impedance change caused by the touch location is much less significant when the finger touches the thread center (i.e., near the peak of the curve in Figure 4). It results into slightly coarser-grained granularity in differentiating touch locations and thus larger location errors.

### Effect of Calibration Points

We further compared the sensing accuracy while varying the number of touch points used for calibration. More specifically, with the two-point calibration, the calibration points were picked at the middle and start position; the three-point calibration was similar but with one additional point picked at the right end of the sensor; the four-point calibration was based on the three-point version with one additional point picked between point 1 and 2. (Figure 16).



**Figure 16 The  $ED_{avg}$  of single vs. up to five calibration points. Error Bars show  $\pm 2$  SE in all figures**

As shown in Figure 16, sensing accuracy improved with the increase of the number of calibration points along the sensor. Note that error may take place on either side of the touch location. Therefore, a touch sensed within the distance of  $ED_{avg}$  to both side of a touch button should be considered a successful hit on the target. Similarly, the distance between two touch buttons should be at least twice as big as  $ED_{avg}$ . We were able to reduce the  $ED_{avg}$  to 4.16 mm with five calibration points. With this level of sensing accuracy, continuous input through sliding is possible but a filter needs to be used to reduce noise. The improvement brought by the calibration points near the fourth quarter (point 3 & 5) is smaller than that brought by the calibration points near the second quarter (point 2 & 4). We expect that point 3 & 5 is not needed if the resistance of the thread is evenly distributed.

We further examine the location accuracy across participants. Figure 17 plots the average location error and the standard error for each participant. We observe that the  $ED_{avg}$  for most participants are around 5 to 8 mm. Participant 4 has the largest error because the width of this participant's finger (17.69 mm) is much larger than the average width (14.47 mm), which influenced the ground truth reading. Overall, our results suggest that touch controllers designed for ThreadSense are likely generalizable among different users.



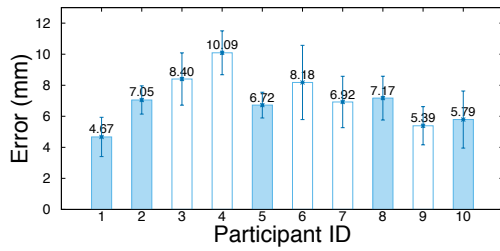


Figure 17 The  $ED_{avg}$  using the mid-point calibration for the ten participants.

### Two-Finger Touch Accuracy

Our result shows that  $ED_{avg}$  is 8.55 mm (s.e. = 2.17 mm) across the three locations and 15 touch points (Figure 18). It is 2.42 mm higher than the error of one-finger touch achieved using the mid-point calibration. Error is again higher near the first quarter of the sensing region with the  $ED_{avg}$  of 10.29 mm (s.e. = 1.95 mm), 7.95 mm (s.e. = 2.36 mm), and 7.41 mm (s.e. = 0.39 mm) for the tested locations at 23 mm, 46 mm, and 69 mm. As shown the Figure 18, error also increased with the increase of the distance between the finger, except at location 46mm where the calibration point (8mm spacing) has the lowest  $ED_{avg}$ . We suspect that it is because the touching area of the finger changes with the opening of fingers, which lead to small deviation in the  $\vec{Z}_{T2}$  from the calibrated one.

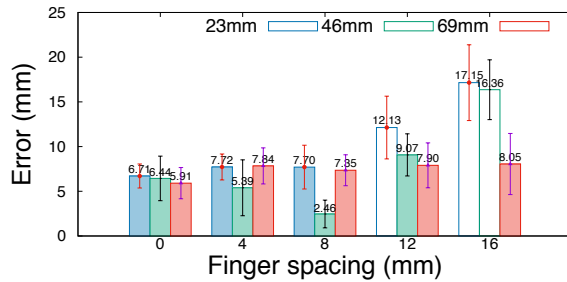


Figure 18  $ED_{avg}$  of two concurrent touches by location and spacing.

### Distinguishing One vs. Two Fingers

Finally, we also investigated how well our system can distinguish one versus two concurrent touches. We used the same data in the accuracy evaluation but fitting the data using the models for both one and two fingers. The recognition was carried out by comparing the similarity between the estimated and measured impedance spectrums. Out of 920 samples collected using one finger, three were recognized as two fingers (error rate: 0.32%). The errors occurred at 69 mm, 85.1 mm, and 89.7 mm from left. Out of 300 samples collected using two fingers, three were recognized as one finger (error rate: 1%). One error occurred at 16.1 mm, 29.9 mm and the other two occurred at 52.9 mm, 85.1 mm. Most of the errors occurred near the end of the sensor, where the increase in impedance caused by touch is less observable.

### SUPPLEMENTARY STUDYS

We conducted three supplementary studies to preliminarily evaluate how well the system can performance under

different environmental noises, finger conditions, and distinguish between different environments.

### Environmental Noise

The goal of this study was to measure the robustness of the system against different environmental noises. This study was carried out with a single participant (male, right-handed, 24 years old).

### Data Collection

The study included four daily environments: (1) a running car, where the sensor was placed on the back seat; (2) outdoor, where the sensor was placed on a wood desk in an open space; (3) kitchen, where the sensor was placed on a dining table, surrounded by kitchen appliances, including a refrigerator and a microwave; and (4) workplace, where the sensor was placed on a wood desk, full of computers and electrical cables. In each environment, the participants taped the sensor using the index finger at five locations: 0 mm, 23 mm, 46 mm, 69 mm, 92 mm from the left. Each trial was repeated five times. In total, we collected 100 samples for data analysis. During the study, the sensor board was powered through a USB cable connected to a laptop.

We first measured the sensing accuracy with  $\vec{Z}_{T1}$  calibrated in each tested environment. We then measured the sensing accuracy with  $\vec{Z}_{T1}$  calibrated in an environment outside the tested ones – in a clean lab desk. This was to investigate whether and how well the system works in a new environment without recalibrating finger impedance.

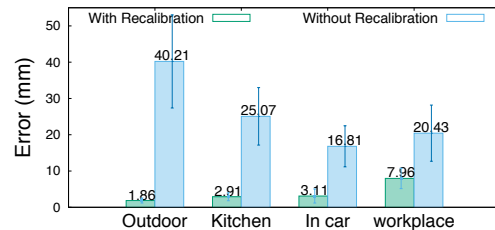


Figure 19: Sensing accuracy with and without calibration in different environments.

### Result

The data was analyzed using a repeated measures ANOVA with *Environment* and *Calibration* as independent variables. Our result showed that the  $ED_{avg}$  across all the four environments was 3.96 mm with the system recalibrated separately for each environment. ANOVA showed no significant different between the four environments in the recalibration condition ( $p = 0.93$ ). In contrast, the  $ED_{avg}$  increased to 25.63 mm without recalibration. ANOVA yielded a significant different between the condition with and without recalibration ( $F_{1,76} = 13.65$ ,  $p < 0.001$ ). The most noticeable difference between the two calibration conditions was found in the outdoor environment (Figure 19). This is mainly because of the large difference in the grounding effect caused by the earth (outdoor) and the floor of our building.

The  $ED_{avg}$  from this participant was lower than the average sensing accuracy found in the main evaluation. This is fine

as sensing accuracy varies among users. For this study, we were only interested in the difference between the tested conditions.

### Distinguishing Environments

The goal of this evaluation was to measure how well the system can distinguish between different environments based on the difference in the current background profile ( $\vec{Z}_G$ ) and the ones existing in the system database. We defined the similarity of two profiles using the Fréchet distance between them. The smaller the distance is the more similar the two profiles are. Our database contains 100 samples for each of the four environments tested in the previous study. To calculate the recognition threshold for each environment, we first calculated 100 distance scores by calculating the similarity between the each of the database profile and the average of all the 100 samples (average profile). The recognition threshold is thus the sum of the average and standard deviation of the distance scores. If the distance between the testing and the average profile is smaller than the recognition threshold, the environment is recognized. Otherwise, we tag it as a new environment. Our testing data included five new samples from each of the four tested environments, collected in a 20-minute interval. Our result showed a 100% recognition accuracy with no false positive.

### Finger Conditions

Finally, we conducted another quick test to investigate if the sensing accuracy is affected by a sweaty finger. We repeated the test in the lab environment with the same participant performing the task with a dry or sweaty finger. To create a sweaty finger, we asked the participants to jog for thirty minutes. We collected 50 samples for data analysis. Our result showed that  $ED_{avg}$  was 2.99 mm in the dry finger condition but increased to 4.41 in the sweaty finger condition. This is expected as body and touch impedance change when the finger sweats. A more careful study is needed to better understand how system performance may vary with the change of body condition.

### DISCUSSION AND FUTURE WORK

We discuss insights gained from this research, propose future research, and acknowledge the limitations of our work.

*Overhead of Recalibration.* As suggested by our study, recalibration may be required upon the change in user or environment. However, our calibration process only requires a single touch from the user (e.g., a long one to distinguish from input) with a workload similar to unlocking an iPhone via the home button. It is simpler than the existing approaches based on machine learning, involving the collection of training data in various settings.

*Spacing Resolution.* The sensing resolution of our approach is affected by the operating frequency of the system. As suggested by Sato *et al.* [19], a wider operating frequency may likely introduce more observable changes in the signal spectrum in response to the user input. The spacing resolution can also be improved with an increased impedance

per sensor length. This allows the signal caused by a small displacement of the finger to be more noticeable. Note that the sensor board used in our current implementation only supports an operating frequency of up to 100 kHz and maximum impedance measurement of 10 M $\Omega$ . We expect that the sensing accuracy of our approach can be improved with better hardware.

*Evaluation.* We presented a technical evaluation that demonstrated promising results for sensing accuracy for our approach. The result should be considered as a high bar of what our sensing technique could achieve since the samples were collected under controlled conditions. Future evaluations should look at whether and how sensing performance may be affected if the sensor is bent. ThreadSense could also benefit from formal user studies to understand how the device and the proposed interactions would be used by end users in the proposed scenarios. For example, a user study may help us find preferred gestures for these applications.

*Sensor Fabrication.* Although the sensor in our current implementation was created using an FDM 3D printer, we believe that it can be manufactured using fibers. One possible approach is to coat a fiber thread using conductive ink. Another possibility would be utilizing materials like conductive polycarbonate to create the thread. The challenge of course is in the requirement for the conductive to be evenly distributed along the thread, which requires a precise manufacturing process to overcome.

*Possibility of User Recognition.* The validation result shown in Figure 4 suggests that the impedance profiles differ across the users. This is primarily due to the difference in their body impedances. Such individual differences can potentially be useful for differentiating input from different people. The key challenge, however, lies in the mixture of the effects on measured impedance caused by user's body and the finger touching at different locations. We plan for future work to investigate methods that can separate these effects for user differentiation.

### CONCLUSION

In this paper, we discussed an approach to enabling 1D touch sensing on an interactive thread based on impedance sensing. Our technique can locate up to two touches with a spacing resolution unachievable by the existing methods. Our system is also unique in that it employs a model-based approach, which locates touches based on a mathematical model describing the change in stripe impedance in relation to the touch location. By sweeping the frequency of injected current during impedance measurement, the system requires only a quick calibration without training. The simplicity of the calibration allows the system to quickly adapt to various environments and users. We foresee that the proposed approach can go beyond 1D touch sensing and serve as an important groundwork for future investigations in sensing techniques on extremely thin and soft objects.

## REFERENCES

- [1] Carles Aliau-Bonet and Ramon Pallas-Areny. 2014. Effects of stray capacitance to ground in bipolar material impedance measurements based on direct-contact electrodes. *IEEE Transactions on Instrumentation and Measurement*, 63 (10). 2414-2421. DOI: <https://doi.org/10.1109/TIM.2014.2310033>
- [2] Analog Devices. 1 MSPS, 12-Bit Impedance Converter, Network Analyzer. AD5933. Retrieved January 5, 2020 from <https://www.analog.com/media/en/technical-documentation/data-sheets/AD5933.pdf>
- [3] Ultimaker BV. Ultimaker 3. Retrieved January 5, 2020 from <https://ultimaker.com/3d-printers/ultimaker-3>
- [4] Laura Devendorf, Joanne Lo, Noura Howell, Jung Lin Lee, Nan-Wei Gong, M Emre Karagozler, Shiho Fukuhara, Ivan Poupyrev, Eric Paulos and Kimiko Ryokai. 2016. I don't want to wear a screen: probing perceptions of and possibilities for dynamic displays on clothing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing System* (CHI '16), 6028-6039. DOI: <https://doi.org/10.1145/2858036.2858192>
- [5] Christine Dierk, Sarah Sterman, Molly Jane Pearce Nicholas and Eric Paulos. 2018. Häiriö: Human hair as interactive material. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '18), 148-157. DOI: <https://doi.org/10.1145/3173225.3173232>
- [6] Chuhan Gao, Xinyu Zhang and Suman Banerjee. 2018. Conductive Inkjet Printed Passive 2D TrackPad for VR Interaction. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (MobiCom '18), 83-98. DOI: <https://doi.org/10.1145/3241539.3241546>
- [7] Chris Harrison, Munehiko Sato and Ivan Poupyrev. 2012. Capacitive fingerprinting: exploring user differentiation by sensing electrical properties of the human body. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 537-544. DOI: <https://doi.org/10.1145/2380116.2380183>
- [8] David Holman and Roel Vertegaal. 2011. TactileTape: low-cost touch sensing on curved surfaces. In *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*, 17-18. DOI: <https://doi.org/10.1145/2046396.2046406>
- [9] Arthur Edwin Kennelly. 1899. The equivalence of triangles and three-pointed stars in conducting networks. *Electrical world and engineer*, 34 (12). 413-414.
- [10] Konstantin Klamka and Raimund Dachselt. 2018. ARCoRD: Visually Augmented Interactive Cords for Mobile Interaction. In *CHI '18 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '18), LBW623. DOI: <https://doi.org/10.1145/3170427.3188456>
- [11] Yuying Li. 1993. Centering, Trust Region, Reflective Techniques for Nonlinear Minimization Subject to Bounds. Technical Report. Cornell University, USA.
- [12] Alex Olwal, Jon Moeller, Greg Priest-Dorman, Thad Starner and Ben Carroll. 2018. I/O Braid: Scalable Touch-Sensitive Lighted Cords Using Spiraling, Repeating Sensing Textiles and Fiber Optics. In *Proceedings of The 31st Annual ACM Symposium on User Interface Software and Technology*, 485-497. DOI: <https://doi.org/10.1145/3242587.3242638>
- [13] Patrick Parzer, Florian Perteneder, Kathrin Probst, Christian Rendl, Joanne Leong, Sarah Schuetz, Anita Vogl, Reinhard Schwoedlauer, Martin Kaltenbrunner and Siegfried Bauer. 2018. RESi: A Highly Flexible, Pressure-Sensitive, Imperceptible Textile Interface Based on Resistive Yarns. In *Proceedings of The 31st Annual ACM Symposium on User Interface Software and Technology*, 745-756. DOI: <https://doi.org/10.1145/3242587.3242664>
- [14] Patrick Parzer, Kathrin Probst, Teo Babic, Christian Rendl, Anita Vogl, Alex Olwal and Michael Haller. 2016. FlexTiles: a flexible, stretchable, formable, pressure-sensitive, tactile input sensor. In *CHI '16 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '16), 3754-3757. DOI: <https://doi.org/10.1145/2851581.2890253>
- [15] Patrick Parzer, Adwait Sharma, Anita Vogl, Jürgen Steimle, Alex Olwal and Michael Haller. 2017. SmartSleeve: real-time sensing of surface and deformation gestures on flexible, interactive textiles, using a hybrid gesture detection pipeline. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 565-577. DOI: <https://doi.org/10.1145/3126594.3126652>
- [16] Alex Pentland. 2000. Looking at People: Sensing for Ubiquitous and Wearable Computing. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 107-119. DOI: <https://doi.org/10.1109/34.824823>
- [17] Ivan Poupyrev, Nan-Wei Gong, Shiho Fukuhara, Mustafa Emre Karagozler, Carsten Schwesig and Karen E Robinson. 2016. Project Jacquard: interactive digital textiles at scale. In *Proceedings of the SIGCHI Conference on Human Factors in Computing System* (CHI '16), 4216-4227. DOI: <https://doi.org/10.1145/2858036.2858176>
- [18] Proto-pasta. Electrically Conductive Composite PLA. CDP11705. Retrieved January 5, 2020 from <https://www.proto-pasta.com/products/conductive-pla>
- [19] M. Sato, I. Poupyrev and C. Harrison. 2012. Touché: Enhancing Touch Interaction on Humans, Screens,



- Liquids, and EverydayObjects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing System* (CHI '12), 483–492.  
DOI: <https://doi.org/10.1145/2207676.2207743>
- [20] Munehiko Sato, Rohan S Puri, Alex Olwal, Yosuke Ushigome, Lukas Franciszkiewicz, Deepak Chandra, Ivan Poupyrev and Ramesh Raskar. 2017. Zensei: Embedded, multi-electrode bioimpedance sensing for implicit, ubiquitous user recognition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing System* (CHI '17), 3972–3985.  
DOI: <https://doi.org/10.1145/3025453.3025536>
- [21] Stefan Schneegass and Alexandra Voit. 2016. GestureSleeve: using touch sensitive fabrics for gestural input on the forearm for controlling smartwatches. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers* (ISWC '16), 108–115.  
DOI: <https://doi.org/10.1145/2971763.2971797>
- [22] Philipp Schoessler, Sang-won Leigh, Krithika Jagannath, Patrick van Hoof and Hiroshi Ishii. 2015. Cord UIs: controlling devices with augmented cables. in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '15), 395–398.  
DOI: <https://doi.org/10.1145/2677199.2680601>
- [23] Julia Schwarz, Chris Harrison, Scott Hudson and Jennifer Mankoff. 2010. Cord input: an intuitive, high-accuracy, multi-degree-of-freedom input method for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing System* (CHI '10) 1657–1660.  
DOI: <https://doi.org/10.1145/1753326.1753573>
- [24] Cátia Sousa and Ian Oakley. 2011. Integrating feedback into wearable controls. In *IFIP Conference on Human-Computer Interaction*, 556–559.  
DOI: [https://doi.org/10.1007/978-3-642-23768-3\\_81](https://doi.org/10.1007/978-3-642-23768-3_81)
- [25] Anita Vogl, Patrick Parzer, Teo Babic, Joanne Leong, Alex Olwal, and Michael Haller. 2017. StretchEBand: Enabling Fabric-based Interactions through Rapid Fabrication of Textile Stretch Sensors. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17).  
DOI: <https://doi.org/10.1145/3025453.3025938>
- [26] Mark Weiser. 1999. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.* 3, 3–11.  
DOI: <https://doi.org/10.1145/329124.329126>
- [27] Raphael Wimmer and Patrick Baudisch. 2011. Modular and deformable touch-sensitive surfaces based on time domain reflectometry. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 517–526.  
DOI: <https://doi.org/10.1145/2047196.2047264>
- [28] Sang Ho Yoon, Ke Huo, Yunbo Zhang, Guiming Chen, Luis Paredes, Subramanian Chidambaram and Karthik Ramani. 2017. iSoft: a customizable soft sensor with real-time continuous contact and stretching sensing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 665–678.  
DOI: <https://doi.org/10.1145/3126594.3126654>
- [29] Yang Zhang and Chris Harrison. 2018. Pulp nonfiction: Low-cost touch tracking for paper. In *Proceedings of the SIGCHI Conference on Human Factors in Computing System* (CHI '18), 117.  
DOI: <https://doi.org/10.1145/3173574.3173691>
- [30] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, 67–173.  
DOI: <https://doi.org/10.1145/2807442.2807480>
- [31] Yang Zhang, Gierad Laput and Chris Harrison. 2017. Electrick: Low-cost touch sensing using electric field tomography. In *Proceedings of the SIGCHI Conference on Human Factors in Computing System* (CHI '17), 1–14.  
DOI: <https://doi.org/10.1145/3173574.3173691>
- [32] Yang Zhang, Robert Xiao and Chris Harrison. 2016. Advancing hand gesture recognition with high resolution electrical impedance tomography. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 843–85  
DOI: <https://doi.org/10.1145/2984511.2984574>