E-textile Microinteractions: Augmenting Twist with Flick, Slide and Grasp Gestures for Soft Electronics

Alex Olwal **Thad Starner** **Gowa Mainini**
Google Research, Mountain View, CA 94043, USA
olwal@acm.org thadstarner@gmail.com gowa.wu@gmail.com

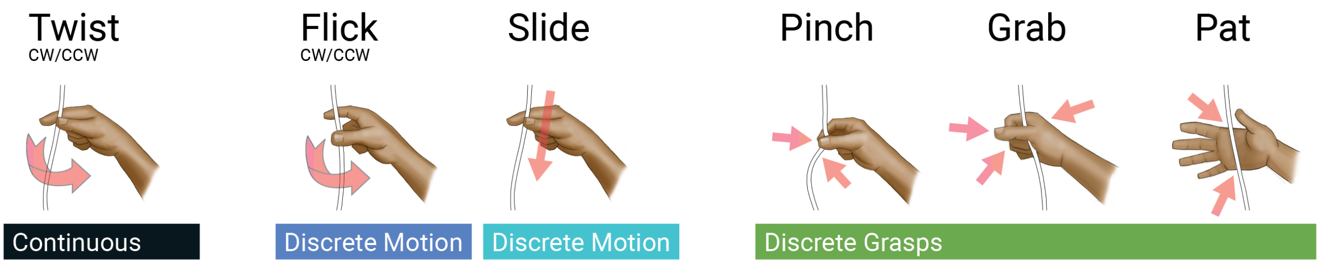


Figure 1. E-textile Microinteractions leverage the I/O Braid architecture for soft electronics to combine *continuous* twist sensing with casual, *discrete* gestures, such as Flick, Slide, Pinch, Grab and Pat. We demonstrate our hybrid interaction techniques through soft electronics that can be used for the control of consumer electronics, digital media and for interactive applications.

ABSTRACT
E-textile microinteractions advance cord-based interfaces by enabling the simultaneous use of precise *continuous* control and casual *discrete* gestures. We leverage the recently introduced I/O Braid sensing architecture to enable a series of user studies and experiments which help design suitable interactions and a real-time gesture recognition pipeline.

Informed by a gesture elicitation study with 36 participants, we developed a user-dependent classifier for eight discrete gestures with 94% accuracy for 12 participants.

In a formal evaluation we show that we can enable precise manipulation with the same architecture. Our quantitative targeting experiment suggests that twisting is faster than existing headphone button controls and is comparable in speed to a capacitive touch surface. Qualitative interview feedback indicates a preference for I/O Braid's interaction over that of in-line headphone controls.

Our applications demonstrate how *continuous* and *discrete* gestures can be combined to form new, integrated e-textile microinteraction techniques for real-time continuous control, discrete actions and mode switching.

Author Keywords
E-textile; electronic textile; smart textile; interactive fabric; wearables; soft electronics; microinteractions; gestures

CHI '20, April 25–30, 2020, Honolulu, HI, USA

ACM ISBN 978-1-4503-6708-0/20/04.
https://doi.org/10.1145/3313831.3376236

CSS Concepts
• **Human-centered computing~Human computer interaction (HCI)**; Interaction Devices; Interaction techniques; User interface design; Gestural input

INTRODUCTION
Integrating capabilities for sensing, feedback and display in everyday objects is part of the vision of both ubiquitous and wearable computing. It is particularly attractive to overcome the boundaries between traditionally rigid devices and soft fabric garments, textiles and furniture to enable technology that can comfortably co-exist with human-facing materials. Recent developments in fabrication, soft electronics and miniaturized computation have been instrumental in advancing interactive textile concepts and applications.

Many examples exist that leverage textile topologies and electronics to integrate input capabilities. Early commercial efforts focused on adding discrete mechanical or touch-sensitive switches to garments, such as snowboarding jackets or gloves with integrated music control [3].

With the mass-adoption of multi-touch capacitive sensing in mobile devices, there has been significant attention to how to embed more expressive interaction. Many recent approaches focus primarily on surface patches that enable 2D interaction [15][19][20][26][27][28][29][31] or 2.5D deformation gestures [10][21][17]. These solutions allow absolute 2D positioning and gesture interfaces similar to multi-touch devices, such as phones or tablets. The ability to track fingers enables both mousing and swipes as well as more complex gestures, such as pinch-to-zoom.

However, interfaces that depend on 2D touch surfaces are not always ideal. Wearable and ubiquitous computing allow computation to be more widely integrated with everyday materials such that our interactions can be more casual and eyes-free. Input devices with affordances that match the context of use enable users that may be situationally impaired, under high cognitive load, or otherwise require fast, unambiguous and efficient input with limited attention or effort.

This work advances recent cord-based concepts, hardware and e-textile interfaces, by enabling the combination of both precise continuous control and casual discrete gestures. See Figure 1. We leverage a recently introduced braided sensing architecture to enable a series of user studies, which help us design suitable casual gestures and a real-time gesture recognition pipeline. To validate the potential for precise interactions, we evaluate the performance and stability of continuous twisting in a controlled study. We show the new capabilities by combining the continuous and discrete gestures into hybrid cord interaction techniques that we demonstrate in a set of applications.

CONTRIBUTIONS

Our contributions include:

- **Hybrid e-textile interaction techniques** that combine precise and continuous control with casual and discrete gestures in a compact textile cord interface.
- **User-dependent classification of *discrete* gestures** with real-time recognition (94% accuracy) for eight gestures (12 participants) informed by our elicitation study (36 participants).
- **Quantified performance of user-independent *continuous* twisting** for relative input, demonstrating benefits over inline remotes (speed, accuracy and preference) and similar performance to state-of-the-art trackpads, based on a formal study (12 participants).
- **Three applications** that show how *continuous twist* and *discrete flick, pinch, grab, pat* and *slide* gestures could be used in a cord for microinteractions with devices, digital media, and entertainment.

RELATED WORK

This work builds on a large body of sensing and interaction techniques developed for electronic textiles.

Interactive Textiles

Since the work by Post et al. [25] on “e-broidery,” much research in interactive textiles has focused on integrating conductive threads into 2D interactive patches to enable capacitive sensing. Gilliland et al.’s Textile Interface Swatchbook [9] attempted to re-invent components of the graphical user interface in such 2D patches. Flexible touch matrices have been created through the weaving of conductive thread [19][20][26], multi-layer conductive fabric [15][20][21][27][31][34], pressure-sensitive textile optical fibers [29], plastic film over sensing electrodes [28], a piezoresistive elastomer-based soft sensor using electrical impedance tomography [42], embroidery [9][25][40], fabric screen-printing [41], and metal foils [16]. Other interactive textiles are designed to be deformed during interaction [17]. Pinstripe [10] utilizes conductive thread to create 1D textile interfaces that are manipulated by pinching the fabric. More related to this work, some structured textiles rise above the 2D plane to create new affordances such as pleats and beads [9][40] that can be stroked or manipulated.

Cords

The affordances of textile-based cords are of particular interest as such devices can be quickly grasped and manipulated without visual attention. Schwarz et al. [33] add sensors to cords to detect touch, pull, and twist; however, these sensors are added to the end of the cord and not incorporated into the structure. Schoessler et al. [32] augment cords with bend sensors to detect knots, piezo copolymer coaxial cables to detect kinks, conductive polymer sandwiched between two sheets of copper foil to detect pressure, and a resistive rubber to detect stretch. They use resistance for touch and pressure in a headphone cord using conductive yarn woven into the fabric of braided cable sleeving.

Some cord interfaces use retractable strings or embedded sensors for interaction [4][5][12][24]. Detecting deployed cord length and relative angle allows a 2D or 3D interaction space. Wimmer and Baudisch [38] create a touch-sensitive cord with absolute positioning using time domain reflectometry and, as an example, demonstrate headphone volume adjustment. Sousa and Oakley [36] sense the position of a conductive bead that slides along a cord. I/O Braid [18] describes how capacitive sensing with only 8 electrodes can detect twists and other gestures performed on a cord braided with insulated conductive threads [26] that form a repeated trackpad along its length.

Gesture Elicitation and Grasping

As we investigate cord-based interaction as an alternative to 2D surface interaction, it becomes critical to also consider how the input device will be grasped and manipulated. Human grasping is a well-studied topic in robotics. Cutkosky’s taxonomy from 1989 [6] is based on a study of human machinists to inform robotic grasping and manipulation. The GRASP taxonomy of human grasp types [8] provides a comprehensive characterization of 33 static grasp types, involving hand interactions with rigid objects. Bell [2] shows many examples of how robotic mechanisms could be extended to textiles and strings. The grasping of flexible and deformable objects is, however, a challenging research area.

Wobbrock et al.’s work with surface gestures [39], and Ruiz et al.’s work with mobile motion gestures [30], demonstrate the benefits of involving end users to inform gesture sets for new interaction devices, especially when best-practices are still being formed. Lee et al. [13] use this approach to provide insights into bimanual interaction with deformable displays. Inspired by these strategies, we ground our single-handed, cord interaction technique design in existing taxonomies and a gesture elicitation study.

Microinteractions

Ashbrook [1] describes microinteractions as requiring less than four seconds to initiate and complete. They are typically designed to minimize visual, manual and mental attention. This reduced distraction benefits wearable computing and ubiquitous computing in particular. Cord interfaces are often motivated by their suitability to such non-primary and micro-interaction tasks [5][12][18][24][32][33][38].

Recently, Sharma et al. [35] systematically explored single-handed grasping microgestures, informed by an elicitation study with over 2400 gestures performed with 12 handheld objects. Directional and continuous gestures were the most popular in Sharma et al.'s study, which partially inspired our emphasis on discrete flicks and precise continuous twist.

DESIGN OPPORTUNITIES: CORD MICROINTERACTIONS

To advance cord user interfaces, we wish to leverage the unique qualities of capacitive sensing textile cords. We identified three opportunities that frame our design for expressive microinteractions:

- **Casual gestures.** With minimal attention or effort, and preferably eyes-free, the user should be able to trigger different basic functionality with one hand. We can avoid the need to acquire a small input device, since previous work shows how the whole cord can be made sensitive.
- **Precise manipulation.** In a similar manner, it is desirable to support precise control of at least one continuous parameter.
- **Leveraging affordances.** Cord stiffness resists twisting and can provide implicit feedback to the user as to the amount of provided input. Interfaces should leverage those tangible characteristics for implicit user feedback.

DISCRETE GESTURES: ELICITATION TO INFORM SET

In order to design an expanded gesture set of discrete gestures we wanted to gather insights into participants' imagined interactions with a gestural cord interface. Our first experiment is thus an elicitation study that explores which gestures users may expect from textile cord interaction without instruction. To provide a relatable task, we focused on controlling earphones.

Participants

We recruited 36 participants (11 females, six left-handed, one ambidextrous) from our institution who were compensated with a $25 gift card for products or services.

Apparatus

The participants were outfitted with a wireless earphones prototype where the two ear pieces are tethered with a braided cord. This device was intended to emulate the design of a common form factor for wireless earphones (e.g., Jakan [14] or Pixel Buds [23]). In order not to influence the participants' imaginations, the device was not functional and did not provide feedback.

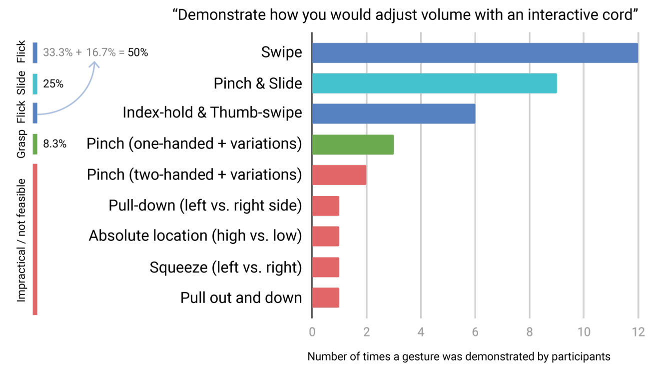


Figure 2. Gesture elicitation with imagined touch sensing. Three derived classes cover 83.3% of the elicited gestures.

Task and Procedure

Participants were asked to demonstrate how they would change the volume using the cord and to describe and explain their interactions and reasoning for choosing them.

Results

Participants suggested a wide range of interaction styles, including swiping, sliding, flicking, holding, pinching, pulling and squeezing. We group them into different classes, described below and shown in Figure 2.

Flick (50% of participants)

Flicks are quick directional gestures orthogonal to or along the cord. As half of the participants proposed some variation of "Swipe" or "Index-Hold & Thumb Swipe", this was the most popular class.

Slide (25% of participants)

Slides are gestures where the fingers move along the length of the cord. Nine participants performed a variant of a "Pinch & Slide" gesture.

Grasp, Single-touch (8.3% of participants)

Three participants performed variations of single-touch pinching, making contact with the cord in different ways.

Impractical: Multi-touch, Absolute Position, Pulling force

The five least popular categories had only one or two participants performing the gesture. These gestures were also not practical to implement with the current architecture and were thus not considered for the gesture set.

Discussion

The GRASP taxonomy [8] classifies gestures based on thumb opposition type, grasp type (power, intermediate or precision) and finger coordination. For gestures with the thumb abducted, we observed precision gestures with the finger pads, either using palmar pinches (thumb and finger), or with a prismatic finger arrangement (2 fingers and thumb). Coarser gestures used palm contact, while wrapping the fingers around the cord. Certain gestures combined a grip with subsequent manipulation, where the thumb was abducted such that it could interact further with the cord.

Going beyond static pose, the Grasping Microgestures work maps gestures performed *during* grasping [35]. Their study

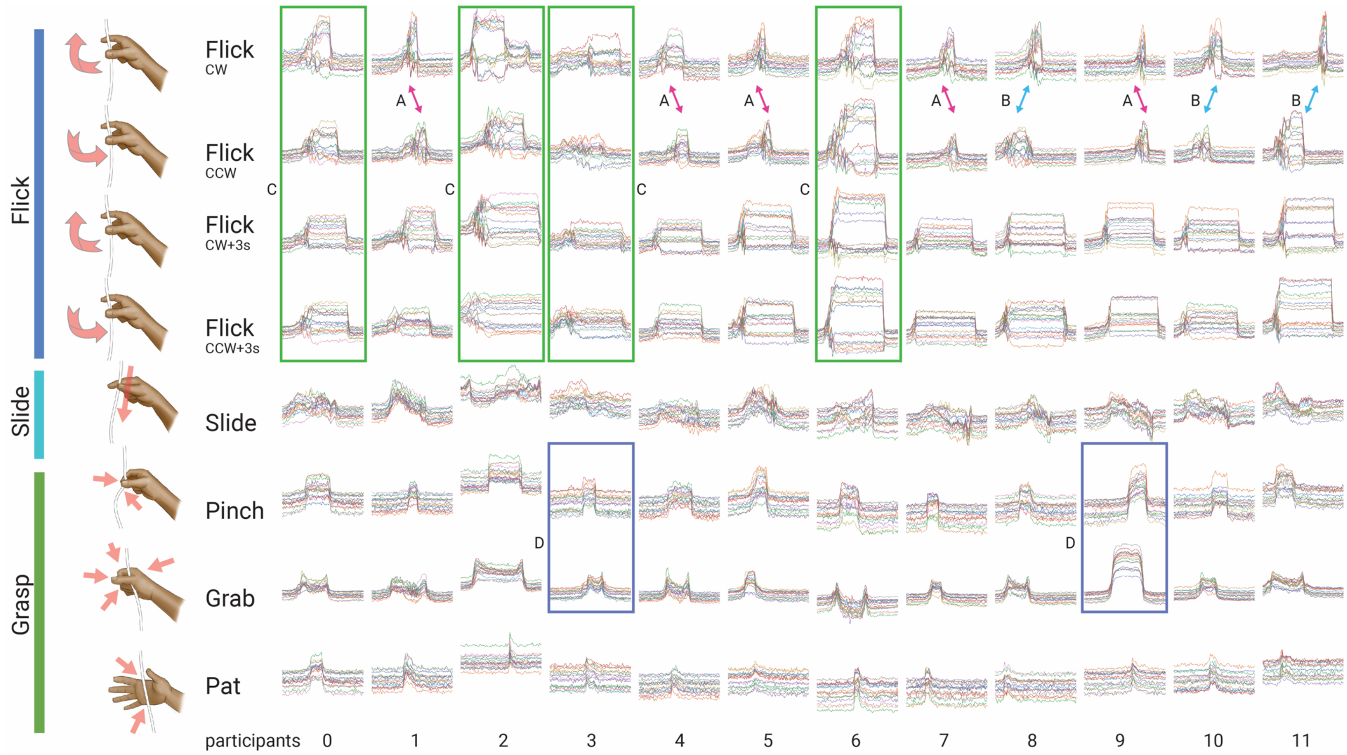


Figure 3: Based on the gesture elicitation, we choose to support three gesture classes (Flick, Slide, Grasp), which represent 83.3% of the elicited gestures. The plot shows data from one repetition (out of nine) for the 12 participants (horizontal axis) for the eight gestures (vertical axis). Each sub-image shows a plot of 16 overlaid feature vectors, which has been interpolated to 80 observations over time. Participants performed gestures without feedback and in their own style, which required user-dependent classification. Some potential issues can be seen in the time series:
(A/B) Temporal variations between Flick directions differ between participant group A and B.
(C) Flick vs. Flick→hold 3s was potentially less distinguishable for some participants, compared to group A/B.
(D) Participants with examples of very similar Pinch and Grab gestures.

finds that directional and continuous swipes are most popular for microinteractions among their participants, which matches our results for Flicks and Slides. Their second most popular gesture, Tap, also aligns with the single-touch gestures that we observed. It is encouraging that despite different object types and device form factors, we reach a similar observation; that one-handed gestures based on relative motion seem well-suited for microinteractions.

To inform continued development, it is important to not prioritize solely based on popularity. We must also account for desired expressivity and technical feasibility, especially since elicitation studies typically avoid limiting participants' imagination by providing technical constraints, such as sensing limitations or mechanical stability.

In our study, we clustered impractical gestures into a dedicated "exclusion class." While multi-touch and absolute position are intriguing to explore in future work, our current sensing approach does not disambiguate between multiple simultaneous contacts [18]. Future hardware extensions, such as time-domain reflectometry [38], could enable capabilities beyond relative motion. We also believe that the demonstrated pulling-force gestures are impractical for cord-based interaction, given potential mechanical instabilities and disconnects. Furthermore, the limited attention to these gestures in our study aligns with previous work [35].

DISCRETE GESTURES: USER-DEPENDENT CLASSIFIER

The results from the elicitation experiment and I/O Braid's capabilities inspired us to focus on three classes, Flick, Slide and Grasp, which would cover 83.3% of the gestures.

In the next session, we collected data from another set of participants performing our candidate gestures to guide the development of a machine learning pipeline. Our goal was to expand the expressivity of cord interaction through per-user trained classifiers to allow a broad set of casual gestures based on the three classes.

Five New Discrete Motion Gestures: Slide and Flicks

We decided to investigate five *motion gestures,* where the user's changing contact with the cord triggers a discrete action. First, we designed variations of a *flick gesture*, inspired by the most popular suggested swiping gesture (18/36 participants). Second, we designed a *slide gesture*, inspired by the second most popular style (9/36 participants).

- **Flick × 2**: CW ⟳ and CCW ⟲
- **Flick→hold 3s × 2**: CW ⟳ and CCW ⟲
- **Slide down**

Three Grasping Styles: Pinch, Grab and Pat

Inspired by the single-touch variations and the Grasping Microgestures work's positive feedback on taps [35], we include three *discrete single-touch* events for user-dependent detection of contact (pinch, grab and pat):

- **Pinch** (thumb and index finger)
- **Grab** (grab in a fist)
- **Pat** (tap with open hand)

Participants

We recruited 13 participants (seven females) from our institution who were compensated with a $25 gift card for products or services. Our institution only allows collection of age ranges. Participants were between 18–24 (n=2), 25–34 (n=8) and 35–44 (n=3) years old. All were right-hand dominant and performed the tasks in a standing position with their dominant hand. We excluded one participant from analysis due to corrupt data.

Apparatus

The experiment used an Apple MacBook Air 13" (MacBookAir7,2; 1440×900 pixel display resolution; 1 pixel = 0.1945 mm). For our data collection, we used the I/O Braid development kit [18], which provides 16 integer values from the 4×4 repeating capacitive sensing matrices along the braided textile cord. We used a braid that was ~500 mm long with ⌀ 4 mm.

Task and Procedure

Participants performed 10 repetitions for the eight discrete gestures. We removed the first repetition from our analysis and classification.

For each gesture set, the experimenter demonstrated the gesture and let the participant practice up to five times. When ready, the experimenter started the data collection for that gesture. Participants saw a 2s countdown on the screen, after which they made contact with the cord and performed the gesture. Immediately after completion, they released the cord and pressed the space bar, which started the countdown for the next repetition. The data collection took approximately 5 minutes per participant.

We continuously recorded the 16 raw capacitance values along with metadata (e.g., participant #, gesture type, repetition # and time stamps). We thus used 8 gestures × 9 repetitions × 12 participants = 864 samples for our analysis.

Classification

We implemented a Python-based toolchain, using machine learning (scikit-learn [22] and tslearn [37]) for time series analysis and classification.

Sample length varies according to the time it took to perform the gesture in a repetition. We resample each gesture time series with linear interpolation. Figure 3 shows 96 samples (12 participants × 8 gestures) with each having 16 features linearly interpolated to 80 observations over time.

Based on the data set size and characteristics we decided to use a time-series specific support vector classifier with a

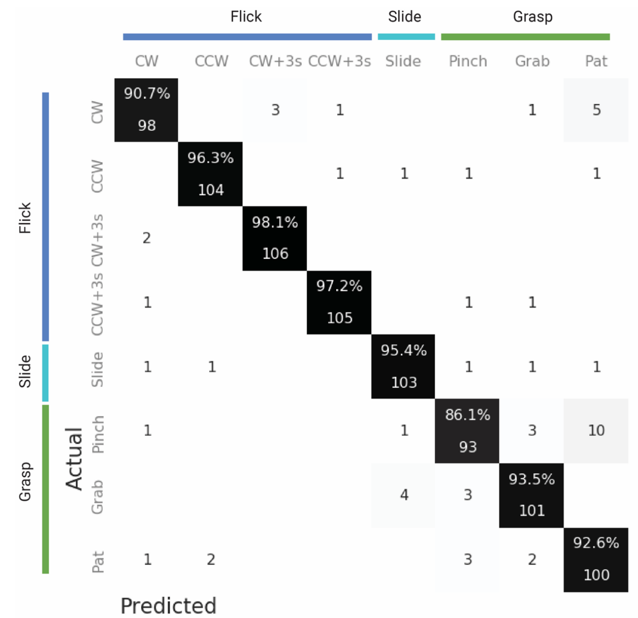
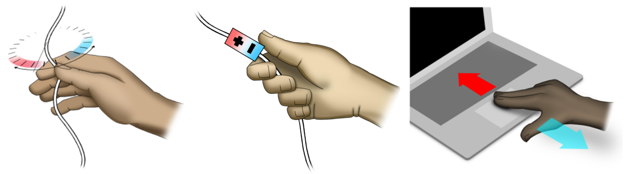


Figure 4. Confusion matrix from 9-fold cross-validation for 12 participants (9 classifications/gesture/participant), with a resulting maximum of 108 correct classifications for each gesture. Average recognition accuracy is 93.8%.

global alignment kernel [7] using the implementation available in tslearn. We ran a 9-fold leave-one-repetition-out cross-validation for each user across the gestures (train on 8 repetitions, test on 1 repetition × 9 permutations).

Discussion

The average recognition accuracy was 93.8%. All gestures are recognized with an average accuracy exceeding 90% except *pinch* which has a recognition accuracy of 86.1% due to confusion with *grab* and *pat*, as shown in Figure 4. These numbers are encouraging, though there is limited ecological validity in such a laboratory study where participants may perform the gesture more consistently than they would in the wild. One drawback to using this recognition approach is that the user must make the full gesture and release it before recognition occurs, possibly slowing interaction speed. In the future, audio or visual feedback could assist the user in performing the gesture properly since real-time sensing is available in parallel. It is encouraging that such a low-resolution sensor matrix (eight electrodes) can enable additional gestural expressivity and demonstrated robustness beyond what was demonstrated in I/O Braid [18]. Notable here is that there are inherent relationships in the repeated sensing matrices that are well-suited for machine learning classification. The support vector classifier enables quick training with limited data, which makes a user-dependent interaction system reasonable. Training for a typical gesture should take ~30s = (~2s pause + 1s gesture) × 10 repetitions, which is comparable to the amount of time required to train a fingerprint sensor.



a) I/O Braid **b) Inline remote** **c) Touchpad**

Figure 5. Our formal evaluation (12 participants) suggests that I/O Braid's user-independent *continuous* twist interaction is faster than an inline remote control and on par with state-of-the-art trackpads.

User-independent Classification

Participants were allowed to freely perform the eight gestures in their own style without feedback as we wanted to accommodate individual differences since the classification of grips is highly dependent on user style ("contact"), preference ("how to pinch/grab") and anatomy (e.g., hand size). Our gesture pipeline was thus designed to require user-dependent training as this resulted in more consistency within each user's data, but various differences across participants. Examples of potential differences are highlighted in Figure 3:

- Group A (five participants) seem to perform CW/CCW differently from group B participants (three participants)
- Flick and Flick→hold 3s seem less distinguishable for group C (four participants), compared to group A and B.
- Two participants in group D seem to perform Pinch vs. Grab in a very similar manner.

Unsurprisingly, these differences result in low average accuracy in leave-one-user-out cross validation analysis. In future work, user-independent results could be improved with stricter instructions to ensure consistency across users, capture of data from a larger population, and in more ecologically diverse scenarios. Additionally, users could be clustered into similar groups which are then used to create independent per-group recognizers. Real-time feedback will also help mitigate differences as the user generally learns to adjust their behavior to achieve better results.

QUANTIFYING PERFORMANCE OF CONTINUOUS TWIST FOR PRECISION AND CONTROL

The *per-user trained* gesture recognition enabled eight new *discrete gestures*, which shows how a variety of actions could be triggered from the textile cord. For continuous interactions, however, we also wanted to quantify how well the previously introduced *user-independent, continuous twist* [18] performs for precision tasks, such as controlling music volume.

Participants

The same 12 subjects as in the previous experiment participated without additional compensation.

Task and Procedure

To evaluate performance, participants used three different input devices (Figure 5) for 1D movement to match a target position that alternated between the left and right sides of the screen. The interface displayed the target as a rectangular outline, which the participant was instructed to "fill" by using the different input devices to expand/shrink a solid rectangle (Figure 6).

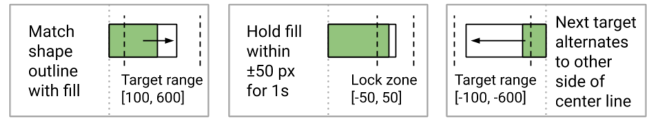


Figure 6. Experimental targeting task to assess input device performance. Participants move cursor to adjust a shape in order to match the outline of a target shape. Reciprocal alternation between left and right sides of the display ensures that they cross the center of the screen in each trial.

The target position was randomized in each trial between 100 to 600 pixels (right side) and -100 to -600 (left side). Thus, in each trial the target could appear in a range of 500 pixels, offset by at least 100 pixels from the vertical center line. The reciprocal alternation of the target location ensured that participants were forced to cross the center of the screen in each trial. Target times were calculated from the crossing of the center line to the completion of the trial. There was no way to fail a trial. Instead, to complete each trial and progress to the next, participants had to reach the target zone (target ±50 pixels) and remain inside for a specific time (1000 ms). The timer was reset if they left the zone before 1000 ms had passed. Our software logged all interactions and event times.

Apparatus: I/O Braid, Buttons and Scroll

To contextualize our results, we compare I/O Braid with two baselines, a trackpad and the common volume remote control box on headphone cords.

I/O Braid: Continuous Twist

We used the same I/O Braid hardware as in the previous experiment using previously described algorithms [18] to track the phase relationships across the matrix to derive clockwise (CW) or counterclockwise (CCW) twist. The relative motion across the touch matrix is accumulated into a positive or negative angle while the user is gripping or twisting the device. Upon release, the device re-centers at 0 (similar to an elastic joystick) and the fill resets to centerline on the screen.

Buttons: In-line Remote with Buttons (Baseline 1)

As a first baseline, we assess performance relative to an in-line remote with mechanical buttons. We use a microcontroller with a SparkFun TRRS 3.5 mm four-ring breakout jack to interface with a pair of Samsung Galaxy 8 earbuds. In our implementation, each button press on plus or minus moves 10 pixels to the left or right, respectively. The user can long-press for (an empirically derived) non-linearly accelerated change, which scales the increment by 5%:

$$x_0 = 0, \quad v_0 = \pm 10, \quad x_{t+1} = x_t + v_t, \quad v_{t\,t+1} = 1.05 * v_t,$$

where x_t is the cursor position at time t, and v_t is the value used to increment the cursor position at each time step. This method was designed to simulate a comparable behavior to that of typical volume control on smart phones.

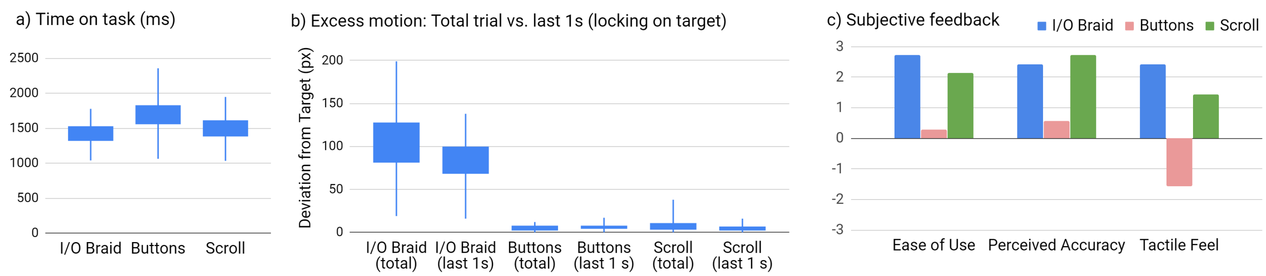


Figure 7. a) Task completion times for the three input devices. I/O Braid was faster than Buttons (statistical significance). b) I/O Braid had more excess motion compared to Buttons and Scroll. The boxplots show median values with quartiles and min/max extent. c) Weighted average subjective feedback. We mapped the 7-point Likert scale to a score in the range [-3, 3]. We multiplied the score by the number of times the technique received that rating and computed an average for all the scores.

Scroll: Non-textile, Rigid Trackpad (Baseline 2)

As a second baseline, we use a gold standard touch input device. We initially considered state-of-the-art touch- or pressure-sensitive textile matrices [16][19][26] but wanted an idealized input device without the trade-offs from different textile sensor implementations and form factors. Thus, as our ideal sensor we use a state-of-the-art laptop trackpad for its responsiveness and precision. Further, its rigidity allows us to focus on pure touch sliding performance, without confounding variations in deformability, texture and stiffness in different textile sensing topologies.

Design

We performed a repeated-measures, within-subjects study with three input devices (I/O Braid, Buttons, and Scroll) and target locations 100 to 600 pixels from the vertical center line and starting point for each trial. The order in which the techniques were presented was counterbalanced across participants. The participant started with a practice block of 10 trials, followed by a pause and then a test block with 50 trials. These 60 trials were performed for each technique, for a total of 180 trials per participant. After each block of practice and test trials, participants rated the technique with regards to ease of use, accuracy, and tactile feel and were invited to provide additional feedback through comments. After using all input devices, they were interviewed about their final overall preference and reasoning.

Analysis

Input device (I/O Braid, Buttons and Scroll) is our independent variable, and we have three dependent variables; time on task (milliseconds), total motion, and motion during end-of-trial. We captured all participant interaction in order to compute motion (pixels) as a measure of stability for each technique.

We analyze cursor motion during the entire trial, as well as cursor motion during the last 1000 ms of each trial and in the range ±50 pixels of the target position (participants were required to stay within this target zone for 1000 ms).

Excess motion was measured as the difference between target distance and actual pixel travel. Similarly, excess motion during end-of-trial, was computed as total distance traveled while in range ±50 pixels from target and a target value of 50. If participants traveled exactly 50 pixels to target value, it would indicate perfect motion. The excess motion would increase with undershooting, overshooting or signal jitter. Buttons have no jitter due to their discrete input and trackpads are designed with filters to help reduce jitter. We used no filtering for I/O Braid.

To visualize performance variations across different target distances we divided the data into 50-pixel intervals for 10 ranges (100–149, …, 550–599) and calculated mean values for each dependent variable in these ranges. Each interval corresponds to approximately 10 mm.

Hypotheses

Prior to the experiment we formulated three hypotheses:

H_1: I/O Braid will be faster than Buttons, since the continuous control provides both fine and fast manipulations with its analog-style rate control.

H_2: Scroll will be faster than I/O Braid, given that a rigid, state-of-the-art touch sensor affords more robust and consistent manipulation.

H_3: I/O Braid will have more excess motion than Buttons and Scroll, given their mechanical stability and filtering.

Task Completion Time

There was a statistically significant difference between input devices as determined by a one-way ANOVA ($F(2,1942) = 78.437$, $p < 0.001$). Effect size is 0.93. A Tukey post hoc test revealed that I/O Braid (1654.3 ± 594.3) was significantly faster than Buttons (2033.2 ± 762.1 $p < 0.001$) but no effect was found when compared with Scroll (1681.4 ± 417.0, $p = 0.703$). See Figures 7a and 8a. These results support hypothesis H_1, but we reject H_2.

Motion

Motion in Entire Trial

For accumulated excess motion in matching target distances, there was a statistically significant difference between input devices, as determined by a one-way ANOVA ($F(2,1942) = 119.297$, $p < 0.001$). The effect size is 0.89. A Tukey post hoc test revealed that I/O Braid (197.7 ± 370.2) had significantly more motion than Buttons (18.1 ± 47.7, $p <$

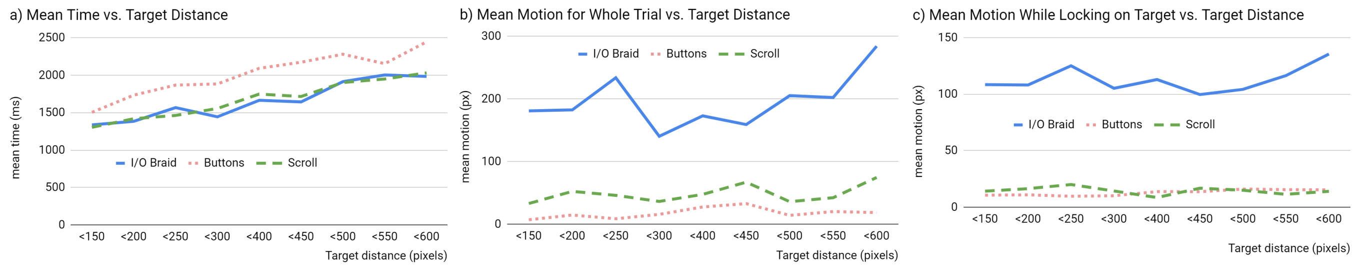


Figure 8. a) Mean completion times for target distances show that Buttons was consistently slower. b) Mean motion (pixels) vs. target distance. I/O Braid has more motion across all target distances. We also observe a potential increase in motion for the largest distances (550-600 pixels). c) Mean motion (pixels) for different target distances while locking on target. I/O Braid has consistently more motion.

0.001) and Scroll (48.8 ± 105.9, $p < 0.001$). These results support hypothesis H3. See Figures 7b and 8b.

Motion in End of Trial, While Locking on Target

Further, when isolating motion during the last 1000 ms of each trial and in range of the target (±50 pixels), the means of excess motion were higher for I/O Braid (Figures 7b and 8c). There was a statistically significant difference between input devices as determined by a one-way ANOVA ($F(2,1942) = 1268.863$, $p < 0.001$). Effect size is 0.57. A Tukey post hoc test revealed that I/O Braid (111.5 ± 65.9) had significantly more excess motion than Buttons (13.0 ± 11.8, p < 0.001) and Scroll (14.4 ± 20.3, p < 0.001). These results support hypothesis H3.

Subjective results

Likert Scales

Participants rated each input device on 7-point Likert scales for ease of use, accuracy and tactile feel. Ratings for I/O Braid and Scroll were overall positive as opposed to Buttons, as shown in Figure 7c. The result suggests that participants regarded I/O Braid to be on par with Scroll. Notably, while participants more frequently rated higher levels of accuracy with I/O Braid than Buttons (10 vs. 6 participants), their cursors moved much more with I/O Braid. We interpret that the movement did not bother them since they were still able to complete the task faster than Buttons and on par with Scroll.

Ratings and T-tests

We compared subjective ratings and found significant differences for ease of use and tactile feel. I/O Braid was rated significantly easier to use than Buttons, $t(12) = 3.282$, $p < 0.01$, but no effect was found with Scroll $t(12) = 0.671$, $p = 0.515$. I/O Braid was also rated as having significantly better tactile feel than Buttons, $t(12) = 3.671$, $p < 0.01$, but no effect was found with Scroll $t(12) = 0.940$, $p = 0.366$.

Interviews

Based on interviews, 8 out of 13 (62%) participants preferred using I/O Braid when compared with the other techniques because they felt it was easier to use, had finer control, and was natural (e.g., like turning a knob). Some noted that I/O Braid was especially good for micro-adjustments and allowed them to reach their target with finer control and accuracy. However, they also stated that the hyper-sensitivity of the twist caused overshooting. Specifically, some participants commented that the mapping between the amount of twist and pixel distance was difficult to learn and understand. Participants were also concerned about accidental activation, particularly if worn on the body.

Discussion

Depending on the context of use, each input device offers its strengths and weaknesses. Findings from statistical analyses indicate I/O Braid to be faster than conventional buttons on a volume remote and surprisingly comparable to the rigorous standards of a laptop trackpad.

These results are particularly remarkable given that I/O Braid was more sensitive and induced more motion for target matching tasks, compared to the rigid input devices, as illustrated in Figure 7. These results can be explained by noting that using the I/O Braid is comparable to holding a trackpad by its sensing surface while using it; motion was registered and accumulated due to the interaction of the surface and the supporting fingers. Filtering, as is done on trackpads, could reduce the amount of motion as well as help reduce overshooting. In fact, Figures 8b and 8c suggest that the I/O Braid has the most noise when the twisting motion is maximized, which could reflect the tension due to the stiffness in the cord as it is held in a highly twisted state. These results could potentially be used to create an adaptive filter that changes parameters based on how twisted the cord is. Even so, the results suggest that even with the most level of twist, participants were able to acquire and lock onto the target. It is remarkable that even without tuning to compensate for this extra motion, I/O Braid is performing so well compared to the other methods. That outcome may be due to the expressiveness of the interface; the user can quickly or slowly twist the cord depending on the distance target distance, and the actions are easy to reverse. For future tasks that require more accuracy, smoothing and high-pass filters will help improve precision.

For more casual interactions, where exact targets are not necessarily the goal (such as for volume control), I/O Braid seems to be a viable and effective option if the targeting tolerance is matched to the required precision. Another advantage is its form factor because users can place their fingers on any location for actuation with little attention when compared to Buttons. Use of conventional button

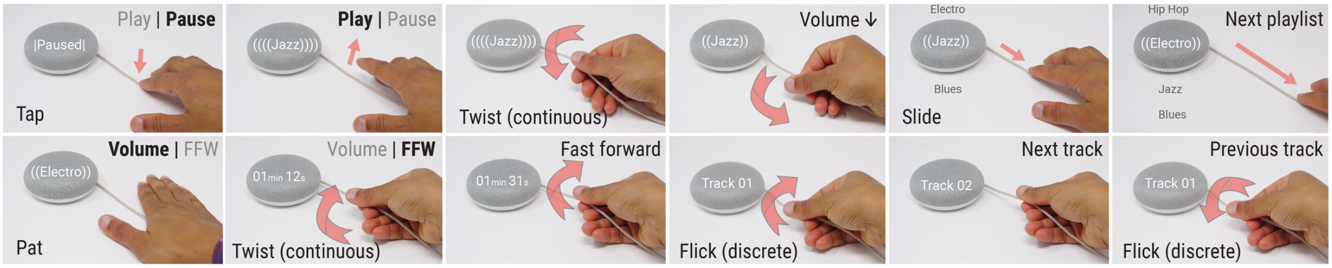


Figure 9. Augmenting continuous twist control with discrete gestures for an interactive speaker cord. Discrete actions: Tap for play/pause, flicks for next/previous track. Slide advances to the next playlist. Remapping: Pat toggles between volume or fastforward control. Continuous twist provides fine control over volume or fastforwarding.

remote controls in earphones require users to find their location on the cord, and change grip for each different action. This method adds a high cost to pressing the wrong button, whereas the twisting gesture is symmetric and reversible. To reject false positives, we currently use a high-pass filter based on the capacitive sensing, which limits activation from accidental skin contact. Further work is needed, however, to develop more robust mechanisms against accidental contact and evaluate the overall performance in actual contexts of use (e.g., volume control, dimmer switch, etc.) and over longer periods of time.

CORD GESTURES AND INTERACTION TECHNIQUES

I/O Braid's ability for parallel sensing of *continuous* twisting and *discrete* gestures provides new building blocks for interactive applications that can be controlled with a single textile sensor.

Continuous Control: Twist

In this paper, we quantified the performance of *continuous* twist to confirm its suitability for fast and precise control of continuous parameters.

Discrete Actions: Flick, Pinch, Grab, Pat and Slide

The machine learning-based pipeline enables classification of discrete gestures, which can be triggered in parallel with continuous interaction, for use as shortcuts or to trigger commands.

Accelerator Gesture: Flick Accelerates Twisting Effect

The *flick* gesture can be performed as a complementary action to accelerate the effect of *continuous twisting*. This approach is analogous to touch-screen dragging and swiping to, e.g., transition from smooth scrolling to jumping a page.

Remapping Input: Switching Modes

We may wish to increase/decrease more than one continuous parameter. We can therefore leverage discrete gestures to cycle across multiple parameters to control. This mechanism also makes it possible to reconfigure the input mapping if we wish to change how we control the interface (e.g., using discrete instead of continuous control of a parameter).

APPLICATIONS: E-TEXTILE MICROINTERACTIONS COMBINING CONTINUOUS AND DISCRETE GESTURES

We implemented an interactive, real-time end-to-end pipeline in Python. The pipeline provides a UDP interface that expects a delineated sequence of 16 capacitance values. It returns a sorted list of gestures with classification probabilities. We trained the pipeline for a subset of our original gesture set, to focus on *flick (CW/CCW)*, *slide down*, *pinch* and *grab*. A 9-fold leave-one-sample-out cross-validation for each of the 12 participants in Experiment 3 resulted in a 95.6% average accuracy for the subset. The pipeline operates in real-time and in parallel with continuous twist and touch tracking. We implemented a set of Java applications to explore how the new interaction techniques of continuous and discrete gestures could enable different expressivity for the user.

Speaker Cord: Controlling Tracks, Volume and Rate

We envision an interactive speaker cord, which augments an existing power or audio cable with interactive gestures for quick and casual control.

We use *pinch* (*or tap*) for play/pause and *grab/pat* to toggle between controlling volume or playback position. *Continuous twist* thus allows us to smoothly change the volume or fastforward the track. A quick *flick* changes to the next/previous track, while *slide* advances to the next playlist. See Figure 9.

Digital Magazines: Navigation with Twists and Flicks

The Digital Magazine prototype leverages smooth *continuous twist*, analogous to a jog dial, to scroll up or down with varying speeds. A *flick* is an accelerator for page down or up. Similar to how touch-screen interfaces use drag and swipe, this interaction combines fine manipulation, rate control, and acceleration in a single mode. Further, the user can *pinch* the cord to toggle between a list of articles and to focus on a specific article. The *slide* gesture cycles to the next magazine section.

We imagine that this interface could be used for reading on a mobile device while wearing headphones. It allows the reader to control the essentials of a reading experience without having to touch the display. See Figure 10.

Game Controller with Optional Mode Switching

Finally, we wanted to explore an experience that requires time sensitive interactive control, and we chose the game of Tetris. Here, we use two modes, which the user can alternate between using the *grab* gesture. In *Twist mode*, *continuous twists* move the block left/right, and *pinch* rotates the block. In the *Flick mode*, *discrete flicks* move left/right, pinch

Figure 10. Interactions with a digital magazine. Continuous twist provides fine scrolling up/down. Discrete actions: Pinch to enter/exit article, flicks accelerate to next/previous page. Slide advances to next magazine section.

rotates the block, and *slide down* drops the block. Here, we demonstrate two strategies that the user can toggle between effortlessly. The more sensitive continuous twist is faster, but has risks of overshooting, as discussed in our performance quantification experiment. The discrete *flicks* require more effort but provide more consistent control.

LIMITATIONS AND FUTURE WORK

Our interaction studies were conducted in a lab setting in a standing posture and may not match or simulate real-life settings while in motion or in different postures. For the data collection used in our machine learning, we were not able to accommodate formal location- and time-independent verification with participants, even if the authors did in fact train and use the system regularly across locations, over time and using different apparatus.

Future studies can improve ecological validity by using the interface and the live gesture recognizer in a variety of contexts, such as on a wearable and attached to a device, to collect more representative data and to compare participant performance. The live recognizer also allows inquiry into how users adapt their gestures over time to improve recognition. Moreover, explorations into locations (e.g., stationary/mobile, indoor/outdoor, public spaces, in vehicle) and varying postures and movements can inform how the technique scales to everyday use where attentional limitations exist. Durability and practicality should be tested in longitudinal studies where the impact of long-term use can be examined and quantified. We currently use a basic threshold-based low pass filter to ensure that only skin contact can activate the cord. Future work should apply more advanced adaptive algorithms with time-series based activation signatures to increase robustness and reject false positives.

While a button provides a single point and orientation of activation, I/O Braid can offer designers greater flexibility as activation can be supported from any position. Future work will investigate the best location and length of the I/O Braid as visibility and manual access may influence its design for a given application.

Studies focusing on various feedback modalities (e.g., sound, light) can help to determine optimal scales for auditory and visual perception of outputs.

Lastly, we have assumed user-dependent gesture recognition for the machine learning-based classifier. Future experiments on user-independent recognition and user-adapted recognition, where the user provides increasing numbers of gesture examples to help adapt a user-independent model to her gestures, would be helpful in determining how best to introduce I/O Braid to new users.

CONCLUSIONS

Recent work has introduced novel ways to embed touch-sensitive electronics in textile, fabric and garments. We build on recent cord sensing techniques to enable hybrid microinteractions for casual and precise interactions using a minimal interactive textile.

First, we contribute a machine learning pipeline to complement previous *continuous twisting* techniques with *discrete flicks, pinches, pats, grabs* and *slide* gestures. The gesture design was informed by proposed gestures from 36 participants in an elicitation study. We trained user-dependent models for 12 participants with 94% accuracy for eight gestures.

Second, we contribute a quantitative targeting experiment that shows how *continuous twisting* is significantly faster than button-based controls and comparable in speed to state-of-the-art non-textile trackpads. Our qualitative interviews indicate a preference for I/O Braid and trackpads over in-line remote controls.

Third, we demonstrate how the *continuous* and *discrete* gestures can be combined to form new e-textile interfaces for discrete actions, continuous parameter control, accelerators and mode switching through a single textile cord that uses just eight electrodes for capacitive sensing. We apply these techniques in our implementation of an interactive speaker cord with music control, a digital magazine browser, and entertainment.

In conclusion, I/O Braid shows a viable approach to simultaneously enable both precise small-scale and large-scale motion in a compact form factor. With this work, we hope to advance textile user interfaces and inspire the use of microinteractions for future wearable interfaces and smart fabrics, where eyes-free access and casual, compact and efficient input is beneficial.

ACKNOWLEDGMENTS

We thank the Google Bio Interfaces team, Interaction Lab, Project Jacquard team, and Google Wearables. Special thanks to Mark Zarich for illustrations, Ben Carroll for UX research support, and Frank Li for data processing. We thank Mathieu Le Goc for valuable discussions and suggestions, and our reviewers for their useful feedback and perspectives.

REFERENCES

[1] Daniel L. Ashbrook. 2010. Enabling mobile microinteractions, Doctoral dissertation, Georgia Institute of Technology.

[2] Matthew Bell. 2010. Flexible Object Manipulation. Ph.D. Dissertation. Dartmouth College, Hanover, NH, USA. Advisor(s) Devin Balkcom. AAI3397916.

[3] Joanna Berzowska. 2005. Electronic Textiles: Wearable Computers, Reactive Fashion, and Soft Computation, TEXTILE, 3:1, 58-75, DOI: 10.2752/147597505778052639

[4] Ravin Balakrishnan, George Fitzmaurice, Gordon Kurtenbach, and Karan Singh. 1999. Exploring interactive curve and surface manipulation using a bend and twist sensitive input strip. In Proceedings of Interactive 3D Graphics (I3D '99), 111-118. http://dx.doi.org/10.1145/300523.300536

[5] Gabor Blasko, Chandra Narayanaswami, and Steven Feiner. 2006. Prototyping retractable string-based interaction techniques for dual-display mobile devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06), 369-372. http://dx.doi.org/10.1145/1124772.1124827

[6] Mark R. Cutkosky. 1989. On grasp choice, grasp models, and the design of hands for manufacturing tasks. In IEEE Transactions on Robotics and Automation, vol. 5, no. 3, pp. 269-279, June 1989. doi: 10.1109/70.34763

[7] Marco Cuturi. 2011. Fast global alignment kernels. In Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML'11), Lise Getoor and Tobias Scheffer (Eds.). Omnipress, USA, 929-936.

[8] Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M. Dollar, and Danica Kragic. 2016. The GRASP Taxonomy of Human Grasp Types. In IEEE Transactions on Human-Machine Systems, vol. 46, no. 1, pp. 66-77, Feb. 2016. doi: 10.1109/THMS.2015.2470657

[9] Scott Gilliland, Nicholas Komor, Thad Starner, and Clint Zeagler. 2010. The textile interface swatchbook: creating graphical user interface-like widgets with conductive embroidery. In Proceedings of the IEEE International Symposium on Wearable Computers (ISWC’10). 1-8. http://dx.doi.org/10.1109/ISWC.2010.5665876

[10] Thorsten Karrer, Moritz Wittenhagen, Leonhard Lichtschlag, Florian Heller, and Jan Borchers. 2011. Pinstripe: eyes-free continuous input on interactive clothing. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11), 1313-1322. https://doi.org/10.1145/1978942.1979137

[11] Sunyoung Kim, Eric Paulos, and Mark D. Gross. 2010. WearAir: expressive t-shirts for air quality sensing. In Proceedings of the SIGCHI Conference on Tangible, Embedded, and Embodied Interaction (TEI '10), 295-296. http://dx.doi.org/10.1145/1709886.1709949

[12] Erik Koch and Hendrik Witt. 2008. Prototyping a chest-worn string-based wearable input device. In Proceedings of the IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks,1-6. https://doi.org/10.1109/WOWMOM.2008.4594882

[13] Sang-Su Lee, Sohyun Kim, Bopil Jin, Eunji Choi, Boa Kim, Xu Jia, Daeeop Kim, and Kun-pyo Lee. 2010. How users manipulate deformable displays as input devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 1647-1656. DOI: https://doi.org/10.1145/1753326.1753572

[14] Jakan. 2020. https://urbanears.com/us/en/jakan.html. Urban Ears.

[15] Joanne Leong, Patrick Parzer, Florian Perteneder, Teo Babic, Christian Rendl, Anita Vogl, Hubert Egger, Alex Olwal, and Michael Haller. 2016. proCover: sensory augmentation of prosthetic limbs using smart textile covers. In Proceedings of the SIGCHI Symposium on User Interface Software and Technology (UIST '16), 335-346. https://doi.org/10.1145/2984511.2984572

[16] Diana Marculescu, Radu Marculescu, Nicholas H. Zamora, Phillip Stanley-Marbell, Pradeep K. Khosla, Sungmee Park, Sundaresan Jayaraman, Stefan Jung, Christl Lauterbach, Werner Weber, Tünde Kirstein, Didier Cottet, Janusz Grzyb, Gerhard Troster, Mark Jones, Tom Martin, Zahi Nakad. 2003. Electronic textiles: a platform for pervasive computing. Proceedings of the IEEE, 91, 12: 1995–2018.

[17] Jussi Mikkonen and Riikka Townsend. 2019. Frequency-Based Design of Smart Textiles. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). ACM, New York, NY, USA, Paper 294, 12 pages. DOI: https://doi.org/10.1145/3290605.3300524

[18] 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 (UIST '18). ACM, New York, NY, USA, 485-497. DOI: https://doi.org/10.1145/3242587.3242638

[19] Patrick Parzer, Florian Perteneder, Kathrin Probst, Christian Rendl, Joanne Leong, Sarah Schuetz, Anita Vogl, Reinhard Schwoediauer, Martin Kaltenbrunner, Siegfried Bauer, and Michael Haller. 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 (UIST '18). ACM, New York, NY, USA, 745-756. DOI: https://doi.org/10.1145/3242587.3242664

[20] 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 Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems (CHI EA '16), 3754-3757. https://doi.org/10.1145/2851581.2890253

[21] 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 SIGCHI Symposium on User Interface Software and Technology (UIST '17), 565-57. https://doi.org/10.1145/3126594.3126652

[22] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12(October), pp. 2825-2830.

[23] Pixel Buds. 2020. https://www.blog.google/products/pixel/pixel-buds/. Google.

[24] Norman Pohl, Steve Hodges, John Helmes, Nicolas Villar, and Tim Paek. 2013. An interactive belt-worn badge with a retractable string-based input mechanism In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13), 1465-1468. https://doi.org/10.1145/2470654.2466194

[25] E. Rehmi Post, Maggie Orth, Peter R. Russo and Neil Gershenfeld. 2000. E-broidery: design and fabrication of textile-based computing. IBM Systems Journal, 39, 3-4: 840-860. http://dx.doi.org/10.1147/sj.393.0840

[26] 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 Systems (CHI'16), 4216-4227. https://doi.org/10.1145/2858036.2858176

[27] Cliff Randell, Ian Andersen, Henk Moore, Sharon Baurley. 2005. Sensor sleeve: sensing affective gestures. In Proceedings of the IEEE International Symposium on Wearable Computers - Workshop on On-Body Sensing, 117-123.

[28] Jun Rekimoto. 2002. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02), 113-120. http://dx.doi.org/10.1145/503376.503397

[29] Markus Rothmaier, Minh Phi Luong, and Frank Clemens. 2008. Textile pressure sensor made of flexible plastic optical fibers. Sensors 8, 7: 4318-4329. http://dx.doi.org/10.3390/s8074318

[30] Jaime Ruiz, Yang Li, and Edward Lank. 2011. User-defined motion gestures for mobile interaction. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 197-206. DOI: https://doi.org/10.1145/1978942.1978971

[31] Stefan Schneegass and Alexandra Voit. 2016. GestureSleeve: using touch sensitive fabrics for gestural input on the forearm for controlling smartwatches. In Proceedings of the ACM International Symposium on Wearable Computers (ISWC '16), 108-115. https://doi.org/10.1145/2971763.2971797

[32] 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 SIGCHI Conference on Tangible, Embedded, and Embodied Interaction (TEI '15), 395-398. https://doi.org/10.1145/2677199.2680601

[33] 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 Systems (CHI '10), 1657-1660. https://doi.org/10.1145/1753326.1753573

[34] Maximilian Sergio, Nicolo Manaresi, Marco Tartagni, Roberto Guerrieri and Roberto Canegallo. 2002. A textile based capacitive pressure sensor. IEEE Sensors, Orlando, FL, USA, 2002, pp. 1625-1630, vol.2. https://doi.org/10.1109/ICSENS.2002.1037367

[35] Adwait Sharma, Joan Sol Roo, and Jürgen Steimle. 2019. Grasping Microgestures: Eliciting Single-hand Microgestures for Handheld Objects. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). ACM, New York, NY, USA, Paper 402, 13 pages. DOI: https://doi.org/10.1145/3290605.3300632

[36] Cátia Sousa and Ian Oakley. 2011. Integrating feedback into wearable controls. In Human-Computer Interaction (INTERACT'11). Lecture Notes in Computer Science, vol 6949. Springer. https://doi.org/10.1007/978-3-642- 23768-3_81

[37] Romain Tavenard, Johann Faouzi and Gilles Vandewiele. 2017. tslearn: A machine learning toolkit

dedicated to time-series data. tslearn.readthedocs.io. Last accessed 2020 01 01.

[38] 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 (UIST '11). ACM, New York, NY, USA, 517-526. DOI: https://doi.org/10.1145/2047196.2047264

[39] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. 2009. User-defined gestures for surface computing. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09). ACM, New York, NY, USA, 1083-1092. DOI: https://doi.org/10.1145/1518701.1518866

[40] Clint Zeagler, Scott Gilliland, Haley Profita, and Thad Starner. 2012. Textile interfaces: embroidered jog-wheel, beaded tilt sensor, twisted pair ribbon, and sound sequins. In Proceedings of the IEEE International Symposium on Wearable Computers (ISWC’12), 60-63.

[41] Clint Zeagler, Scott Gilliland, Stephen Audy, and Thad Starner. 2013. Can i wash it?: the effect of washing conductive materials used in making textile based wearable electronic interfaces. In Proceedings of the 2013 International Symposium on Wearable Computers (ISWC '13). ACM, New York, NY, USA, 143-144. DOI: https://doi.org/10.1145/2493988.2494344

[42] Yang Zhang, Gierad Laput, and Chris Harrison. 2017. Electrick: Low-Cost Touch Sensing Using Electric Field Tomography. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). ACM, New York, NY, USA, 1-14. DOI: https://doi.org/10.1145/3025453.3025842