GestureSleeve: Using Touch Sensitive Fabrics for Gestural Input on the Forearm for Controlling Smartwatches

Stefan Schneegass
VIS, University of Stuttgart
stefan.schneegass@vis.uni-stuttgart.de

Alexandra Voit
VIS, University of Stuttgart
alexandra.voit@vis.uni-stuttgart.de

ABSTRACT
Smartwatches provide quick and easy access to information. Due to their wearable nature, users can perceive the information while being stationary or on the go. The main drawback of smartwatches, however, is the limited input possibility. They use similar input methods as smartphones but thereby suffer from a smaller form factor. To extend the input space of smartwatches, we present GestureSleeve, a sleeve made out of touch enabled textile. It is capable of detecting different gestures such as stroke based gestures or taps. With these gestures, the user can control various smartwatch applications. Exploring the performance of the GestureSleeve approach, we conducted a user study with a running application as use case. In this study, we show that input using the GestureSleeve outperforms touch input on the smartwatch. In the future the GestureSleeve can be integrated into regular clothing and be used for controlling various smart devices.

Author Keywords
Wearable Computing; Smart Textiles; Smartwatch; Gesture Input.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation(e.g. HCI): Miscellaneous

INTRODUCTION
Smartwatches have become a common part of our personal device infrastructure. They offer quick information access and can be controlled while on the move. Especially during sports activities, smartwatches provide benefit that smartphones cannot due to their fixed location at the user‘s wrist. The user can still perceive content without the necessity of getting the phone out of their pocket. This allows reading incoming messages and performance measures from fitness applications. However, current input techniques are not perfectly suited for interacting on the move. Current smartwatches have small touch-enabled displays similar to smartphones. Due to the reduced size of the displays, touch-based interaction becomes even more cumbersome on the move compared to smartphones.

ISWC '16, September 12-16 2016, Heidelberg, Germany

ACM 978-1-4503-4460-9/16/09$15.00
DOI: http://dx.doi.org/10.1145/2971763.2971797

Figure 1. A user wearing the GestureSleeve at the forearm made out of touch-enabled textile. The GestureSleeve is capable of detecting gestures extending the input space of smartwatches.

At the same time, touch enabled textiles are gaining importance (e.g., Project Jacquard [16]). These textiles detect touch input similar to touch screens. Thus, they are capable of detecting taps as well as gestures. In contrast to touch screens, the touch enabled textiles still have similar properties as regular textiles. They are flexible just as comfortable to wear. By including patches made from such textiles in our everyday life clothing, novel input mechanism can be created. This includes simple taps as well as gesture input.

In this work, we introduce GestureSleeve, a novel input system for smartwatches using touch enabled textile at the forearm (cf., Figure 1). The textile has much more input space than a smartwatch on its own. At the same time, the smartwatch provides output as well as processing power that is not integrated in the textile itself. To show the feasibility of GestureSleeve, we implemented a fitness tracker on the user‘s smartwatch. The application can be controlled with touch gestures performed on the touch enabled textile on the forearm and with touch input on the smartwatch.

The contribution of our work is twofold. First, we introduce the idea of combining touch enabled textiles with smartwatches. Thereby we build a prototype that is capable of detecting taps and stroke gestures. Second, we report on a user study showing that touch gesture based input on smart textile outperforms touch input on the smartwatch.

RELATED WORK

The basic interaction techniques of nowadays smartwatches have been adopted from mobile phones. The screen is touch-enabled and allows direct touch as well as small gestures. One way to deal with the small display size is adopting the interface. This has mainly be done for text input. Zoomboard, for example, uses multiple taps for entering characters [12]. With the first tap, a broad region on a qwerty keyboard is selected and, with the second tap, the actual character is selected out of a zoomed-in part of the keyboard. In addition, Funk et al. developed a touch-sensitive wristband for text input on a smartwatch [4]. Moving the interaction beyond the touch screen, Partridge et al. proposes adding tilt movements to ease up the text input [13]. Text is entered by tilting in a certain direction and pressing a button. Moving this concept even further, Xiao et al. use tilting in addition to panning, twisting, and clicking to control watches [31]. They show different example applications that can be controlled using these operations without occluding the screen.

Different approaches for gesture based input that utilize the space around the smartwatch are explored. This is realized using simple depth sensors [29], cameras [24], or magnetic field sensors [1, 6]. Due to the placement at the wrist, smartwatches are capable of detecting wrist and hand gestures of the hand the watch is placed on by augmenting the watch-strap with sensors. Examples include the capacitive wristband by Rekimoto [19] which is capable of sensing the movement of the wrist and fingers. Similarly, Zhang and Harrison use electrical impedance tomography detecting similar movements [32].

In contrast to mid-air gestures, 2D gestures on the user‘s body can also be used to control smartwatches. These gestures are more socially acceptable compared to mid-air gestures because they are less expressiveness [20]. While using 2D gestures to control smartwatches is sparse, different approaches for on-body gestures have been explored. Skinput, for example, is capable of detecting taps on the arm by measuring acoustic signals inside the body [7]. Weigel et al. used carbon-doped electrodes and a combination of resistive and capacitive sensors for the detection of touch inputs on the skin [28]. However, they are mainly focusing of detecting taps rather than more sophisticated gestures. Garment-based sensors, in contrast, allow a variety of different touch gestures integrated into the clothing of the user. Using simple stitched buttons, Komor et al. explored textile based input on the strap of a messenger bag [10]. They present different layouts and analyze their performance while swiping over or pressing the buttons. Focusing on interaction with smart glasses, Dobbelstein et al. propose performing swipe gestures on the belt [3]. However, the approach could also be extended to smartwatches. In addition to using button based approaches, fabrics with similar functionality as touch screens have been proposed with various spatial resolutions and refresh rates (cf., Zhou et al. for an overview [33]). An early example using these fabrics is GesturePad [19], a textile touchpad that can be integrated in clothes of the user. Randell et al. embedded smart textiles into a jacket [18]. They recognized affective gesture with this jacket and used them for communicating with pre-defined messages to a special partner. Similarly, Heller et al. used a touch sensitive fabric at the tights showing the influence of different activities such as walking, sitting, and standing on input performance [8]. In contrast, we use a fabric at the forearm of the user which offers an easy to reach input space while looking at the smartwatch‘s display.

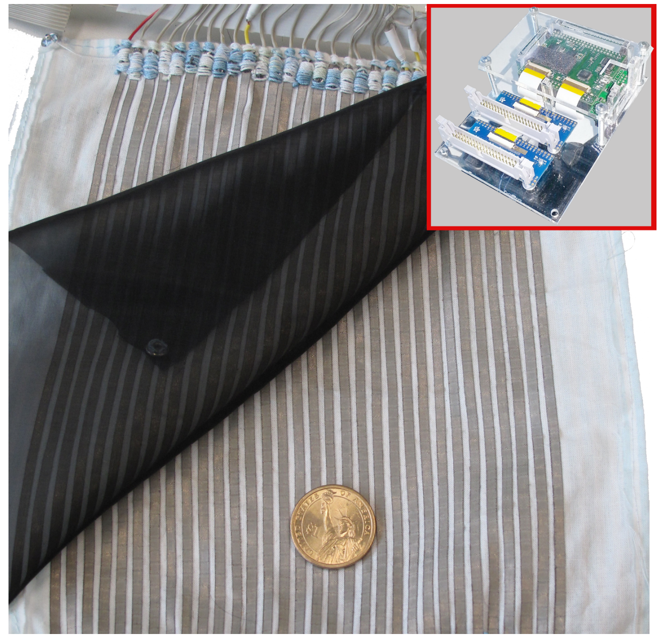

Figure 2. The touch enabled textile used for creating the Gesture Sleeve. Two layers of stripe electrodes placed perpendicular to each other with a force sensitive fabric in between. The processing electronics is shown in the top right.

THE GESTURESLEEVE

Performing gesture input on the forearm provides a large input area. Different touch enabled fabrics have been proposed such as the work of Zhou et al. [33] or Project Jacquard [16]. These fabrics are similar to regular, non-interactive fabrics and allow manufacturing clothes with similar comfort and wearability (cf., [5]). We present GestureSleeve which augments the forearm with touch functionality. Using this functionality, we can detect various kind of input such as taps or stroke gestures. We envision using this input as a means to control smartwatches. Thereby, GestureSleeve fills the blanks between touch input on devices with small form factor and complex and not always socially accepted, mid-air gestures.

Hardware Setup

We use a touch enabled fabric with the size of 16cm x 16cm (cf., Figure 2). The fabric consists of three layers. On top and bottom, groups of 32 parallel stripe electrodes of 3mm width and 2mm spacing between two electrodes are attached to the fabric. Both fabrics with electrodes are placed perpendicular to each other. A force sensitive fabric is placed between both layers changing the resistance based on the applied vertical pressure. The final fabric is fixed with Velcro tape around the arm of the user. The fabric is connected via cables to a small processing board (cf., Figure 2 – top right) measuring the resistance for each of the 32 by 32 (i.e., overall 1024 pressure sensors) points where two stripe sensors overlap. The sampling rate is 50Hz. The measured values are transfered via wireless LAN to the smartwatch. As a smart swatch, we use the Simvalley AW 414.GO running Android 4.0.

Sensor Placement

Even though related work suggests placing the touch enabled textiles on the thigh performs slightly more intuitively compared to the user‘s lower arm [9, 26], we decided to use the lower arm because our system is designed to be used while on the move. The thigh might not be easily reachable due to the movement, especially during sport activities such as running. In contrast, the lower arm is reachable most of the time. The proximity to the smartwatch further recommends the placement at the forearm. Users can observe the feedback on the watch while entering commands on the sleeve. Additionally, Speir et al. investigated if the users prefer one- or two-handed interactions for wearable remote controls worn on the wrist or the hand area [25]. The results of their study show that two-handed interactions on a wristband are significantly faster than one-handed interactions on a glove. Also the participants stated that the two-handed interaction on the wristband is easier than one-handed interactions on the glove. In addition, Profita et al. investigated the social acceptance of inputs on smart clothes [17] and found out that interactions on the forearm and the wrist are mostly social acceptable. While we used a patch of touch enabled fabric for our prototype, we envision the full sleeve being touch-sensitive so that the user does not need to find the touch sensitive area.

Gesture Detection

Due to the placement of the GestureSleeve on the forearm and its continuous movement, the sensor data is noisy. When the user shakes his or her arm, the GestureSleeve reacts on this movement in terms of changing the values of some of the 1024 pressure sensors. Tackling this, we included a threshold defining a minimum pressure value counting as an intended input. To prevent folds in the fabric from generating unintended input, we dismiss all sensor values exceeding the threshold which do not have at least 6 neighbors that also exceed the threshold. Even though the user just taps the fabric, the resistance of the adjacent sensors also exceed the threshold. As soon as an intended input is detected, we instantly start a new gesture. Since most of the time the pressure value changes for more than a single sensor, we always use the sensor with the highest pressure as intended input. The position of this sensor in the 32x32 matrix (i.e., the ”pixel” position as known from touch screens) is added to a list that stores the currently performed gesture. We add the sensor position with the current highest value to the list until no further input is detected for at least 10 frames (i.e., 200 ms). Afterwards, the gesture detection is started with the recorded list.

In the initial version of GestureSleeve, we focus on detecting taps as well as stroke gestures. For detecting the stroke gestures, we used the $P algorithm [27] with $N = 64$ points. A gesture is recognized if the sum of all Euclidean distances between the points of the performed and a respective template gesture is smaller than 7 pixels. For detecting taps, we extended the gesture recognizer. It detects a tap when the length

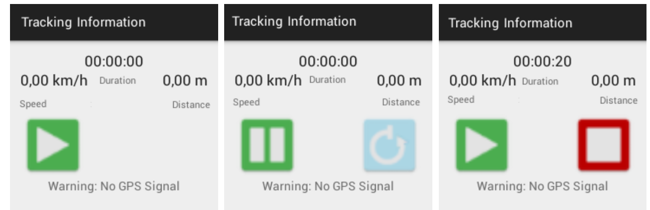

Figure 3. The running application used in the user studies. The user interface at the beginning (left), after the participant started the recording (middle), and after pressing pause (right).

of a gesture is between 10 and 50 points and the Euclidean Distance is between all points smaller than 2 pixels.

EVALUATION: SPORTS TRACKING APPLICATION

We conducted a user study to evaluate our GestureSleeve using a sports tracking application as the use case. Interacting during running activities gains more and more importance. In addition to controlling sport tracking applications, other use-cases for interacting while running are proposed. Wozniak et al. present an approach for remote cheering for the runner [30]. They use a watch like device, providing visual and haptic feedback and a single button to ask for and acknowledge remote cheering. In contrast, Smus and Kostakos used foot gestures for controlling music player while running [23]. Nevertheless, we focus on basic features of running applications.

Sports Tracking Application

We developed a prototype of a sport tracking application that is capable of tracking jogging activities (cf., Figure 3). The application can be controlled either via touch buttons on the smartwatch or via gestures on the GestureSleeve. In a first step, we investigated how current sports application are designed. We analyzed the user interface of the top three Android applications (Endomondo[1], Runtastic[2], and SportsTracker[3]) offering the functionality we wanted to use in the study (i.e., start the tracking, pause the tracking, stop the tracking, and initiate a new training lap). Then, we derived our user interface from these applications. We used similar layout of the interface elements and chose the size of each element similar to the ones of the commercial applications. The *start button* is placed on the left and turns into the *pause button* as soon as the user starts the training. The button for starting the *next lap* is placed on the right and appears as soon as the training is started. When the user pauses the training, the *next lap* button turns into the *stop button*. The design is deliberately chosen to be minimalistic so that the interface does not distract the user while running.

For the gesture based input, we chose four different gestures – one for each command. These gestures are derived from the icons that are shown in the user interface of the different sports applications (cf., Figure 6). The training is started with

[1] https://play.google.com/store/apps/details?id=com.endomondo.android

[2] https://play.google.com/store/apps/details?id=com.runtastic.android

[3] https://play.google.com/store/apps/details?id=com.stt.android

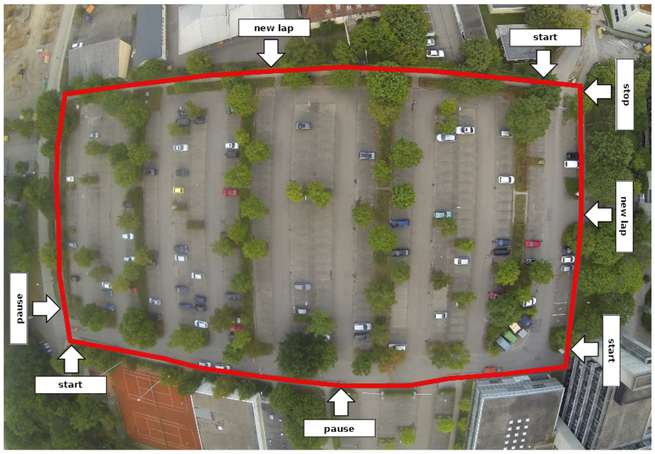

Figure 4. The trail used for the user study of approximately 400m. The positions of the signs are indicated where the participants performed a certain task.

an arrow gesture (derived from the triangle symbol of the running app‘s play button), paused by a stroke (derived from the pause symbol with but with a single line), and stopped by a simple tap (similar to a square of the stop button). A new lap is started by drawing a circle indicating the lap in a stadium. We defined templates for the gesture detection and asked 14 persons to perform each of the gestures 15 times to train our system. None of them took part in the user study afterwards.

Participants and Procedure

We invited 16 participants (6 female, 10 male) aged between 21 and 38 years ($M = 27.3$, $SD = 4.6$) through university mailing lists. Each of the participants received €10 as renumeration. After participants arrived in our lab they filled in a consent form and we equipped them with a smartwatch and the GestureSleeve. The processing board was placed in a back pack with an external battery pack. Next, we explained the GestureSleeve and the four gestures as well as the smartwatch application and the touch interface. We allowed the participants to get familiar with both interfaces. Before each condition we repeated this instruction so that participants knew which gesture to perform, how the gesture look like, and how the application was controlled using the touch screen.

We designed our study as a within subject study and, thus, each participant took part in both condition, namely controlling the smartwatch application with gestures and with touch input. The order of the conditions was alternated. We prepared a jogging trail of about 400 meters (cf., Figure 4). Along the trail, we distributed paper signs with commands (e.g., pause). We instructed the participants to jog along the trail and perform the commands seen on the signs as soon as they reach the line in front of the signs. We also told them, not to pause for executing the commands. In total, each participant should perform start three times, new lap and pause twice, and stop once per condition. We deliberately chose the pause commands in areas were the participants needed to cross the street and we instructed them to carefully cross the street.

We logged the user interaction with the smartwatch and the GestureSleeve (i.e., the raw pressure sensor values and the detected gestures). Further, we videotaped the whole study for post-hoc video annotation of interaction times and to understand issues during the interaction. We used high quality video setting with a frame rate of 60 fps for the videotaping. We selected the video frame in which the user's hand starts moving into the direction of the GestureSleeve or the touch screen of the smartwatch and the one the participant lifted the finger again from the input device (cf., Figure 5). We deliberately chose this method because we wanted to investigate the whole interaction time including the time the user needs to select the input areas on the smart fabric or the buttons on the touch-screen of the smartwatch. Therefore, we measured both the time needed to perform the gesture and the time the button was pressed.

Results

We analyzed objective measurements (task completion time, error rate) and then conducted semi-structured interviews with all participants after they performed both conditions.

Performance Measures

In general, the GestureSleeve performed well and every participant was able to use the system to control the smartwatch. Examples of the detected gestures are depicted in Figure 6. We excluded two participants for technical reasons. One participant interacted while pressing the arm against his body so that we could not identify the starting point of interaction using the video recording. The other participant took a shortcut of the trail and, thus, did not perform all commands. We first compared the task completion time. We extracted the task completion time by manually encoding the start and end time of each interaction from the video data (cf., Figure 5). As soon as a participant moved his or her arm towards either the GestureSleeve or the touch screen of the smartwatch, we started measuring the interaction time. The end point was defined as the video frame in which the participant first lifts the finger.

We conducted a dependent t test comparing both interaction techniques with regards to the task completion time and error rate. For the task completion time, the results show that participants controlled the smartwatch application significantly faster using the GestureSleeve ($M = 1.50s$, SD = 0.09) compared to touch input ($M = 1.85s$, SD = 0.12), $t(13) = -3.583$, $p = .003$, $r = .78$. The error rate for using the GestureSleeve ($M = 0.28$, SD = 0.37) was higher compared to touch input ($M = 0.17$, SD = 0.27). The t test, however, could not show any statistically significant differences, $t(13) = 1.649$, $p = .123$, $r = .33$.

Qualitative Feedback

In the semi-structured interviews, we ask the participants question about the GestureSleeve and the perceived performance. The participants stated using gestures is *fun, novel* [P2], and *easy* [P5]. However, they also noted that they would have needed more time to perfectly master the gesture input [P6]. One participant acknowledged that he needed to look at the sleeve for performing the gestures but is confident that

Figure 5. Video recordings of a user performing three different inputs. The user performs input on the smartwatch using touch (top) and two different gestures on the GestureSleeve (middle and bottom).

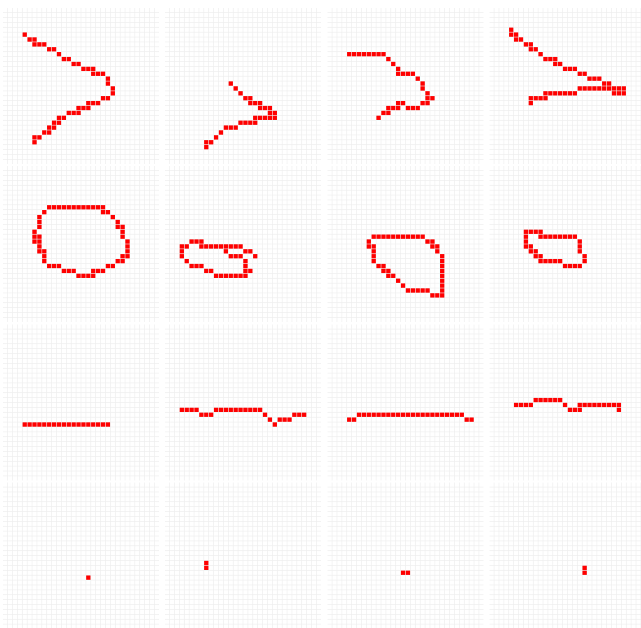

Figure 6. Examples of the performed start, new round, pause, and stop gestures of four different participants recorded during the user study.

this would not be necessary with more practice [P14]. Additionally, participants agreed on the fact that the ease of input is mainly influenced by the type of gestures. Tap and stroke gestures were easier to perform compared to circles. Especially when the fabric was not tightly fitted to the arm, the circle gesture was not easy to perform. Furthermore, participants noted that performing gestures without looking at the GestureSleeve was possible which was not the case for touch input on the smartwatch.

DISCUSSION

GestureSleeve Performance

The evaluation of the GestureSleeve yields promising results. We showed users are capable of faster entering commands compared to touch input on the smartwatch's display. The error rate is slightly higher which could be caused by the fact that gestures have the inherent drawback that they need to be learned and remembered. There is no cue reminding user's which gesture needs to be performed to fulfill the desired task. This is also supported by statements of the participants during the interviews. Even though we derived the gestures from the well known icon set of known running applications, participants needed to think about which gesture is mapped to which command (as stated by, for example, [P6, P14]). By giving participants more time to practice the gestures, we believe that the error rate will be further reduced and eventually match or even surpass the error of the smartwatch interface. Further, the smartwatch used in this study is an off-the-shelf product which we compared to our prototype of a GestureSleeve. A more mature version of the GestureSleeve would most likely perform even better.

Interaction Location

We decided to focus on interacting on the arm due to the close proximity between input and output medium. In the summer, however, wearing short sleeved shirts is common in many regions. While a similar gesture-based interaction could be applied using the skin as input surface (cf., Skinput [7] or iSkin [28]), using other parts of the body can also enhance the interaction with smartwatches. As related work suggests [9, 26] the thighs are another promising area for entering commands. Situations in which thighs are especially useful include, for example, sitting on a chair in a meeting or watching movies on a sofa. The user is then able to easily enter commands on the thighs. Thus, the GestureSleeve concept could be applied to the thighs as well.

In this work, we used a prototypical version of the GestureSleeve which we designed as an add-on patch to the normal clothing of the user. We believe that in the future, clothing will be produced using touch enabled fabric [2]. Thus, the user can perform gestures on the whole sleeve and is not restricted to a certain patch. However, since we used a patch of 16 by 16 cm, we believe that the size did not influence the results of our study. The forearm of the participants was always completely covered by the GestureSleeve.

Additional Interaction Possibilities

We focused on gestures performed on the sleeve because gestures are not influenced by a decoupled input and output space. However, additional types of input are also possible with our system. One example could be mapping different parts of the GestureSleeve to parts of the smartwatch (e.g., the four quarters of the display space). Thus, a touch event on the upper left quarter of the GestureSleeve is mapped to an input on the upper left quarter of the smartwatch. More fine grained direct touch input (e.g., mapping a QWERTY keyboard to the touch-sensitive textile) would probably require a visual feedback using textile display elements [14].

Additional Application Scenarios

In addition to controlling the smartwatch while running, GestureSleeve has the potential to be used for various applications beyond the fitness domain. One example could be using the gestural input to start applications for smartphones as proposed by Poppinga et al. [15]. By performing stroke gestures linked to certain applications, the user has quick access to these applications. Furthermore, pre-defined answers to received text messages could be defined. When the user receives a message, he or she could perform a gesture similar to a tick mark to send a quick reply. Even though we deliberately chose to enrich the input of smartwatches, GestureSleeve can also be used in combination with other smart devices such as eyewear computers.

Extended Devices

While we focus in this work on the extension of smartwatches, GestureSleeve can also be useful for extending the input capability of other mobile and wearable devices. The user can quickly gesture on the sleeve with the phone in the pocket to enter commands such as accepting calls or increasing the volume of a music player. Particularly devices that currently do not provide touch input such as smart glasses can benefit. The user enters commands on the GestureSleeve whereas the the output is provided by the near-eye display. This overcomes one of the main drawbacks of smart glasses. Further, the GestureSleeve could be extended with an on-body display [22] or mid-air display [21] working at a stand-alone wearable system.

Integration into Clothing

In the user study, the GestureSleeve is attached to the arm using Velcro tape making it easy to remove and reusable for multiple participants. However, the additional layer of Velcro makes the device bulkier. When integrating the GestureSleeve into clothing, the Velcro tape will not be necessary anymore reducing the size of the device. The electronic board is currently connected using wires which will be substituted with conductive thread and a connector between textile and electronics (e.g, a ball-grid-connector [11]). The electronics will then be slided into a pocket which will further reduce the size of the device. Thus, the GestureSleeve will become almost indistinguishable from regular clothes having similar wearability properties.

Limitation

The performed study used a jogging trail of 400 meters and presenting dedicated commands to the participants. Allowing the participants using a jogging distance they normally use and to perform the commands they actually would perform for measuring their performance could have increased the ecologic validity of the study. However, we believe that for an initial evaluation of our GestureSleeve concept, the usage of a more controlled setup is appropriate. Additional aspects we did not investigated are the environmental conditions. We conducted the study in the summer during days of sunshine. We did not evaluate how the GestureSleeve performs during rain or snow. we also did not evaluate how gloves impact the interaction.

CONCLUSION

In this paper, we present GestureSleeve a gesture input device for smartwatches. By providing a large input area, GestureSleeve helps overcome the drawback of the limited input space of smartwatches. To evaluate our approach, we developed a fitness application and conducted a user study in which we compared gestural input on the GestureSleeve and touch input on the smartwatch. Our results show that the GestureSleeve outperforms touch input with regards to the task completion time. While our prototypical version is built as an add on to the normal clothing of the user, we believe that in the future, sleeves of regular clothing can incorporate similar interaction possibilities.

In future work, we plan to further extend our GestureSleeve concept. We will more closely investigate how well gestures can be entered on the sleeve. Further, we will explore more application scenarios such as controlling list-based user interfaces or social network applications with the *GestureSleeve*.

Acknowledgements

This work is supported by the EU FP7 under grant agreement no. 323849 and by the German ministry of education and research within the DAAN project (13N13481).

REFERENCES

1. Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nenya: Subtle and Eyes-free Mobile Input with a Magnetically-tracked Finger Ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2043–2046. DOI: http://dx.doi.org/10.1145/1978942.1979238

2. Jingyuan Cheng, Paul Lukowicz, Niels Henze, Albrecht Schmidt, Oliver Amft, Giovanni A Salvatore, and Gerhard Troster. 2013. Smart Textiles: From Niche to Mainstream. *Pervasive Computing, IEEE* 12, 3 (2013), 81–84. DOI: http://dx.doi.org/10.1109/MPRV.2013.55
3. David Dobbelstein, Philipp Hock, and Enrico Rukzio. 2015. Belt: An Unobtrusive Touch Input Device for Head-worn Displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2135–2138. DOI: http://dx.doi.org/10.1145/2702123.2702450
4. Markus Funk, Alireza Sahami, Niels Henze, and Albrecht Schmidt. 2014. Using a Touch-sensitive Wristband for Text Entry on Smart Watches. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 2305–2310. DOI: http://dx.doi.org/10.1145/2559206.2581143
5. F. Gemperle, C. Kasabach, J. Stivoric, M. Bauer, and R. Martin. 1998. Design for wearability. In *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*. 116–122. DOI: http://dx.doi.org/10.1109/ISWC.1998.729537
6. Chris Harrison and Scott E. Hudson. 2009. Abracadabra: Wireless, High-precision, and Unpowered Finger Input for Very Small Mobile Devices. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 121–124. DOI: http://dx.doi.org/10.1145/1622176.1622199
7. Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: Appropriating the Body As an Input Surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 453–462. DOI: http://dx.doi.org/10.1145/1753326.1753394
8. Florian Heller, Stefan Ivanov, Chat Wacharamanotham, and Jan Borchers. 2014. FabriTouch: Exploring Flexible Touch Input on Textiles. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers (ISWC '14)*. ACM, New York, NY, USA, 59–62. DOI: http://dx.doi.org/10.1145/2634317.2634345
9. Paul Holleis, Albrecht Schmidt, Susanna Paasovaara, Arto Puikkonen, and Jonna Häkkilä. 2008. Evaluating capacitive touch input on clothes. *MobileHCI '08* (2008), 81–90. DOI: http://dx.doi.org/10.1145/1409240.1409250
10. N Komor, S Gilliland, J Clawson, M Bhardwaj, M Garg, C Zeagler, and T Starner. 2009. Is It Gropable? Assessing the Impact of Mobility on Textile Interfaces. In *Wearable Computers, 2009. ISWC '09. International Symposium on*. 71–74. DOI: http://dx.doi.org/10.1109/ISWC.2009.21
11. Andreas Mehmann, Matija Varga, Karl Gönner, and Gerhard Tröster. 2015. A Ball-grid-array-like Electronics-to-textile Pocket Connector for Wearable Electronics. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers (ISWC '15)*. ACM, New York, NY, USA, 57–60. DOI: http://dx.doi.org/10.1145/2802083.2802093
12. Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2799–2802. DOI: http://dx.doi.org/10.1145/2470654.2481387
13. Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello, and Roy Want. 2002. TiltType: Accelerometer-supported Text Entry for Very Small Devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST '02)*. ACM, New York, NY, USA, 201–204. DOI: http://dx.doi.org/10.1145/571985.572013
14. Roshan Lalintha Peiris, Adrian David Cheok, James Keng Soon Teh, Owen Noel Newton Fernando, Wen Yingqian, Andre Lim, Pan Yi, Doros Polydorou, Kian Peng Ong, and Mili Tharakan. 2009. AmbiKraf: An Embedded Non-emissive and Fast Changing Wearable Display. In *ACM SIGGRAPH 2009 Emerging Technologies (SIGGRAPH '09)*. ACM, New York, NY, USA, Article 1, 1 pages. DOI: http://dx.doi.org/10.1145/1597956.1597957
15. Benjamin Poppinga, Alireza Sahami Shirazi, Niels Henze, Wilko Heuten, and Susanne Boll. 2014. Understanding Shortcut Gestures on Mobile Touch Devices. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 173–182. DOI: http://dx.doi.org/10.1145/2628363.2628378
16. 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 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4216–4227. DOI: http://dx.doi.org/10.1145/2858036.2858176
17. Halley P. Profita, James Clawson, Scott Gilliland, Clint Zeagler, Thad Starner, Jim Budd, and Ellen Yi-Luen Do. 2013. Don'T Mind Me Touching My Wrist: A Case Study of Interacting with On-body Technology in Public. In *Proceedings of the 2013 International Symposium on Wearable Computers (ISWC '13)*. ACM, New York, NY, USA, 89–96. DOI: http://dx.doi.org/10.1145/2493988.2494331
18. Cliff Randell, Ian Anderson, Henk Muller, Andrew Moore, Philippa Brock, and Sharon Baurley. 2005. The Sensor Sleeve: Sensing Affective Gestures. In *Ninth International Symposium on Wearable Computers–Workshop on On-Body Sensing*. 117–123.

19. Jun Rekimoto. 2001. GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. In *Proceedings of the 5th IEEE International Symposium on Wearable Computers (ISWC '01)*. IEEE Computer Society, Washington, DC, USA, 21–. http://dl.acm.org/citation.cfm?id=580581.856565

20. Julie Rico and Stephen Brewster. 2010. Usable Gestures for Mobile Interfaces: Evaluating Social Acceptability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 887–896. DOI: http://dx.doi.org/10.1145/1753326.1753458

21. Stefan Schneegass, Florian Alt, Jürgen Scheible, Albrecht Schmidt, and Heifeng Su. 2014. Midair Displays: Exploring the Concept of Free-Floating Public Displays. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/2559206.2581190

22. Stefan Schneegass, Sophie Ogando, and Florian Alt. 2016. Using On-body Displays for Extending the Output of Wearable Devices. In *Proceedings of the 5th ACM International Symposium on Pervasive Displays (PerDis '16)*. ACM, New York, NY, USA, 67–74. DOI: http://dx.doi.org/10.1145/2914920.2915021

23. Boris Smus and Vassilis Kostakos. 2010. Running Gestures: Hands-free Interaction During Physical Activity. In *Proceedings of the 12th ACM International Conference Adjunct Papers on Ubiquitous Computing - Adjunct (UbiComp '10 Adjunct)*. ACM, New York, NY, USA, 433–434. DOI: http://dx.doi.org/10.1145/1864431.1864473

24. Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. 2014. In-air Gestures Around Unmodified Mobile Devices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 319–329. DOI: http://dx.doi.org/10.1145/2642918.2647373

25. Jessica Speir, Rufino R. Ansara, Colin Killby, Emily Walpole, and Audrey Girouard. 2014. Wearable Remote Control of a Mobile Device: Comparing One- and Two-handed Interaction. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 489–494. DOI: http://dx.doi.org/10.1145/2628363.2634221

26. B Thomas, Karen Grimmer, J Zucco, and Steve Milanese. 2002. Where does the mouse go? An investigation into the placement of a body-attached touchpad mouse for wearable computers. *Personal and Ubiquitous computing* 6, 2 (2002), 97–112.

27. Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2012. Gestures As Point Clouds: A $P Recognizer for User Interface Prototypes. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction (ICMI '12)*. ACM, New York, NY, USA, 273–280. DOI: http://dx.doi.org/10.1145/2388676.2388732

28. Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. 2015. iSkin: Flexible, Stretchable and Visually Customizable On-Body Touch Sensors for Mobile Computing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2991–3000. DOI: http://dx.doi.org/10.1145/2702123.2702391

29. Anusha Withana, Roshan Peiris, Nipuna Samarasekara, and Suranga Nanayakkara. 2015. zSense: Enabling Shallow Depth Gesture Recognition for Greater Input Expressivity on Smart Wearables. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3661–3670. DOI: http://dx.doi.org/10.1145/2702123.2702371

30. Paweł Woźniak, Kristina Knaving, Staffan Björk, and Morten Fjeld. 2015. RUFUS: Remote Supporter Feedback for Long-Distance Runners. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 115–124. DOI: http://dx.doi.org/10.1145/2785830.2785893

31. Robert Xiao, Gierad Laput, and Chris Harrison. 2014. Expanding the Input Expressivity of Smartwatches with Mechanical Pan, Twist, Tilt and Click. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 193–196. DOI: http://dx.doi.org/10.1145/2556288.2557017

32. 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 & Technology (UIST '15)*. ACM, New York, NY, USA, 167–173. DOI: http://dx.doi.org/10.1145/2807442.2807480

33. Bo Zhou, Jingyuan Cheng, Mathias Sundholm, and Paul Lukowicz. 2014. From Smart Clothing to Smart Table Cloth: Design and Implementation of a Large Scale, Textile Pressure Matrix Sensor. In *Architecture of Computing Systems - ARCS 2014*, Erik Maehle, Kay Roemer, Wolfgang Karl, and Eduardo Tovar (Eds.). Lecture Notes in Computer Science, Vol. 8350. Springer International Publishing, 159–170. DOI: http://dx.doi.org/10.1007/978-3-319-04891-8_14