EdgeSelect: Smartwatch Data Interaction with Minimal Screen Occlusion

Ali Neshati
aneshati@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

Aaron Salo
saloamj@myumanitoba.ca
University of Manitoba
Winnipeg, Manitoba, Canada

Shariff AM Faleel
amshamoh@student.ubc.ca
University of British Columbia
Kelowna, British Columbia, Canada

Ziming Li
ziming.li17@student.xjtlu.edu.cn
Xi'an Jiaotong-Liverpool University
Suzhou, China

Hai-Ning Liang
haining.liang@xjtlu.edu.cn
Xi'an Jiaotong-Liverpool University
Suzhou, China

Celine Latulipe
celine.latulipe@umanitoba.ca
University of Manitoba
Winnipeg, Manitoba, Canada

Pourang Irani
pourang.irani@ubc.ca
University of British Columbia
Kelowna, British Columbia, Canada

ABSTRACT

We present EdgeSelect, a linear target selection interaction technique that utilizes a small portion of the smartwatch display, explicitly designed to mitigate the 'fat finger' and screen occlusion problems, two of the most common and well-known challenges when interacting with small displays. To design our technique, we first conducted a user study to answer which segments of the smartwatch display have the least screen occlusion while users are interacting with it. We use results from the first experiment to introduce EdgeSelect, a three-layer non-linear interaction technique, which can be used to interact with multiple co-adjacent graphs on the smartwatch by using a region that is the least prone to finger occlusion. In a second experiment, we explore the density limits of the targets possible with EdgeSelect. Finally, we demonstrate the generalizability of EdgeSelect to interact with various types of content.

CCS CONCEPTS

• **Human-centered computing → Empirical studies in interaction design**.

KEYWORDS

smartwatch; target selection; visualization; interaction technique

ACM Reference Format:
Ali Neshati, Aaron Salo, Shariff AM Faleel, Ziming Li, Hai-Ning Liang, Celine Latulipe, and Pourang Irani. 2022. EdgeSelect: Smartwatch Data Interaction with Minimal Screen Occlusion. In *INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION (ICMI '22), November 7–11, 2022, Bengaluru, India.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3536221.3556586

ICMI '22, November 7–11, 2022, Bengaluru, India

ACM ISBN 978-1-4503-9390-4/22/11…$15.00
https://doi.org/10.1145/3536221.3556586

1 INTRODUCTION

Smartwatches collect data through both sensors and network sources and then provide data-dependent information representations to users through their small displays [1]. Because many of the collected data are inter-related (e.g., heart rate, body temperature, and sleep pattern), representing multiple graphs that visualize these multiple data sources is common to help smartwatches users engage in sense-making [9]. By displaying graphs together, smartwatch application designers can provide an overview of multiple sets of related data and therefore allow users to access information with convenience directly on such wearable displays, without having to resort to their smartphone. As a result, smartwatch users can enhance their self-awareness and adjust their behaviour to achieve their captured health-related goals [14]. For example, visualizing heart rate data, sleep patterns, and body temperature can help smartwatch users can get a glimpse of their sleep quality [2], without having to resort to elaborate analytic tools. Figure 1 shows examples of existing applications using multi-visualizations to represent multiple inter-related data on smartwatch applications. As an example, figure 1-d shows how representing multiple vital information about a baby during pregnancy (heart rate and oxygen saturation which are interrelated data) can increase mothers' awareness of their baby in real-time. Figure 1-b also shows information about the user's running/walking speed and heart rate. These two pieces of information can be helpful for athletes in how to adjust their speeds and heart rate to achieve a specific level of performance.

Due to the limited space on smartwatch displays, visualization techniques are usually simplified. This means that most of such visual tools are designed to show an overview of the data without providing details [5]. Removing either the x- and/or y-axis, gridlines, the graph's legend, or removing data point values are the most common ways to simplify a graph. Although excluding such details helps smartwatch users more clearly see an overview of the collected data, it forces users to access the data to other devices

Figure 1: Smartwatch applications using multiple visualizations to represent inter-related data. a,b) wHealth dashboard, representing consumed water, burned calories, steps taken, walking/running speed, and heart rate. c) built-in health app in GOQII smartwatch representing steps, burned calories and heart rate. d) Sense4Baby, a smartwatch application representing multiple vital information about the baby during pregnancy.

(e.g., cell phone or computer) to explore the details [17], which is bothersome and unnecessary for many instances (such as a quick glance to understand how events are related) [18]. Therefore, interacting with multiple data visualizations on a smartwatch can help the users explore the details of graphs right on their wrists without using other devices [9]. For instance, smartwatch users can interact with multiple graphs to explore their heart rate, speed, and cadence during a workout for potential adjustment.

Designing an interaction technique for smartwatch displays is challenging due to the limited available space. Direct interaction with graphs on the smartwatch can block the content under the user's finger, known as the 'fat finger' problem [23]. Screen occlusion is another similar issue that may happen while interacting with small touch displays [8]. Screen occlusion may happen even with indirect interaction with the content. For instance, wearing the smartwatch on the left wrist and interacting with the specific segments of the smartwatch bezel with the right hand can block a significant portion of the display [16]. Therefore, 'fat finger' and screen occlusion issues are two primary considerations for any interaction technique developed for a smartwatch display.

Researchers have proposed many interaction techniques to mitigate screen occlusion and the 'fat finger' problem [6, 7, 10, 11, 21, 26, 30]. However, most of these techniques are not specifically designed to interact with multiple visualization techniques. Partial BezelGlid (PBG) [16] is a technique designed for users to interact with graphs on smartwatches using the smartwatch bezel, while avoiding the 'fat finger' and screen occlusion issues. However, PBG has two significant disadvantages: users can only interact with one visualization at a time, and the technique does not work well for interacting with dense graphs. Our technique, EdgeSelect, addresses these issues by using a small portion of the smartwatch display as an interactive area to explore multiple graphs with different density levels.

In this paper, we examine the possibility of minimizing the interactive region of a smartwatch display, to mitigate screen occlusion while exploring multiple graphs. This small interactive region, is designed to occupy less than 10 percent of the smartwatch display without hindering the density of graphs displayed on the watch. Our two main objectives summarized as our contributions include: 1) quantifying the screen occlusion resulting from interacting with the entire smartwatch display; and 2) taking advantage of segments with minimum screen occlusion to design an interaction technique to interact with multiple graphs.

2 RELATED WORK

We describe previous research related to data visualization and interaction with smartwatches, as well as techniques that attempt to address the 'fat finger' and screen occlusion problems.

2.1 Multi-graph Visualization on Smartwatches

Due to the limited available space on the small displays of smartwatches, novel visualization techniques are needed to present complex, dense data [15]. Many of these visualization techniques are explicitly designed to take advantage of every small piece of available space on the smartwatch display, know as space-efficient or space-filling visualization techniques [4, 25]. For example, the Space-Filling Line Graph (SFLG) [15] was introduced to simplify complex, dense line graphs and add more available space around the main graph. The additional space can be used to visualize other interrelated data. As the outcome of SFLG, there will be multiple graphs representing information about multiple related data sources.

G-Spark [18] and Sparklines [24] are two other space-efficient visualization techniques that can help smartwatch app designers to add more graphs and charts to the primary visualization technique on the smartwatch display. These two techniques compress the main graph on the y- and x-axis without affecting the glanceability of the graph, so there will be more available space to add auxiliary charts. The additional information can help the users glean deeper insights from the data and increase their self-awareness. As smartwatches are capable of collecting many health-related data points, this could be important for smartwatch users. In addition to these space-efficient visualization techniques, a survey paper investigated how smartwatch users utilize information visualization on their smartwatch watch face [9]. The result of this paper showed that more than 80% of smartwatch users use at least two different pieces of information on the watch face simultaneously. They also showed that graphs are among the most common information types displayed on smartwatches, along with icons and text.

2.2 Screen Occlusion Limitations

Due to the limited available space on smartwatch displays, presenting information [1], and interacting with content are both challenging [19]. The 'fat finger' issue is one of the main problems when a smartwatch user interacts with the small display directly [23], blocking the content underneath the user's finger and preventing the user from interacting with the content properly. Screen occlusion is a similar issue, but can occur even without direct interaction with the display [8, 16].

Screen occlusion is a general issue that may happen in other devices, even with large displays. For instance, Vogel et al. [27, 28] investigated the effect of screen occlusion while users interact with tabletop displays. They showed that the position of the user's hand could significantly block the content preventing the user from interacting with the content. However, the fat finger problem is widespread on small displays, as the size of the target is usually smaller than the fingertip [19], and direct interaction with this content can block most or all of the content the user is trying to interact with. One of the ways to fix the fat finger problem is by designing interaction techniques that provide indirect interaction with the targets. However, indirect interaction techniques may not address the screen occlusion problem. For instance, Full BezelGlide (FBG) [16] is a bezel interaction technique to interact with graphs on smartwatches. However, interacting with specific segments of the smartwatch bezel may cause screen occlusion, blocking the content underneath the body of the user's finger. This is the main reason the smartwatch user needs to tilt their head or the smartwatch in order to see the content.

Many novel interaction techniques have been proposed by researchers to solve the fat finger problem and the screen occlusion issue [6, 7, 10, 11, 21, 26, 30]. However, many of these interaction techniques require advanced hardware, which may cause additional limitations. Additional hardware usually means bigger and heavier form factors that can affect the usability of wearable devices such as a smartwatch. Using the existing smartwatch hardware such as the smartwatch bezel [3], and the accelerometer [13] to interact with the smartwatch display will avoid the potential side effects of adding new hardware to the smartwatch.

2.3 Techniques Mitigating Screen Occlusion

The smartwatch bezel provides an excellent potential solution to fix the screen occlusion and fat finger problem when interacting with smartwatch content [3, 22, 31–33]. Smartwatch bezel overcomes these two issues by moving the interaction space to the rim of the display. However, most existing bezel interaction techniques are not designed to interact with graphs and charts. For example, [32] and [20] are bezel interaction techniques that utilize the smartwatch bezel for text entry. In COMPASS [32], the user can use the entire smartwatch bezel to interact with the display to select letters to perform the text entry task. This means that the interaction area is not restricted to a specific space on the smartwatch bezel, meaning that the screen occlusion may still happen while using the smartwatch bezel to interact with the display. Screen occlusion is also an issue in SwipeRing [20]. Using this bezel input technique, the user must move their finger from one segment on the smartwatch bezel to the other across the display, entirely blocking the content beneath the moving finger.

Bezel-Initiated (BI) [29] and Bezel-to-Bezel (B2B) [12] are techniques explicitly designed as eyes-free interaction techniques with the smartwatch display.

Neshati et al. introduced Partial BezelGlide (PBG) and Full BezelGlide (FBG) as two bezel interaction techniques to interact with graphs on smartwatches. However, these two proposed bezel interaction techniques have major constraints. The thin width of the smartwatch bezel does not allow the user to use the width of the smartwatch bezel as an input slider [16]. This is the main reason BezelGlidee is a linear interaction technique that allows the user to interact with only one graph at a time, which means it does not allow the user to interact with multiple graphs. In addition, it is unclear how the number of data points and density of the graph can affect the performance of BezelGlide.

3 STUDY 1: MEASURING SCREEN OCCLUSION OF SMARTWATCH DISPLAYS

The main goal of this study is to measure the screen visibility of the smartwatch display while participants are interacting with the entire smartwatch display. In other words, we will quantify the minimum area the user needs to interact, to gain the maximum screen visibility on such small displays. The difference between this study and the screen visibility study of Ali et al. [16] is the interaction area. In BezelGlide,the authors only focus on interacting with the smartwatch bezel. However, our screen occlusion experiment is further generalized to assess the degree of screen occlusion when interacting with the entire smartwatch display, including the screen bezel.

3.1 Experimental Design

In this experiment, we divide the screen display into 88 targets. The location and the number of segments on the right/left and up/down half of the display was the same (symmetric). We picked 88 targets, as this was the largest number of targets we could include, without making touch cumbersome. Using a larger number of subdivisions would reduce the number of targets and make it tedious to select each cell. The system highlighted a cell or target (red colour) on each trial, randomly selected from 88 targets. Only the highlighted target was visible to the participants, and they had to hit the target to be able to move on to the next target. To make sure they hit the target, we provided them with auditory and visual feedback, changing the background colour from white to light green after landing on the target. We also asked them to hold their finger on the target for three seconds to ensure that our camera recorded the display and their finger position on the smartwatch display. Participants selected each of the 88 targets three (x3) times per block. Increasing the number of samples would result in a more accurate result but needed to be balanced with the total overall time to collect the data. To measure the screen occlusion, we placed a camera on a head strap designed to hold an active camera at the center of the participant's forehead. Unlike the work done by Neshati et al. [16], we decided to record videos as this makes the recording process easier. To minimize the offsets between the camera lens angle and participant's point-of-view, we followed the same guidelines explained by previous papers [16, 28, 28] by placing the camera in the middle of the forehead and as close as possible to the eyes without interfering with the participant's eyesight and affecting their performance on the task.

3.2 Apparatus

We used a GoPro Hero 9 camera to record the experiment. This action camera can capture high-quality videos with a built-in stabilizer, which ensures clear videos for image/video processing.

3.3 Participants

We recruited 13 participants, all students from a local university (5 female, 8 male, $M_{age} = 21.29, SD = 1.5$). Participants were all right-handed, and none were colour blind (based on Ishihara color

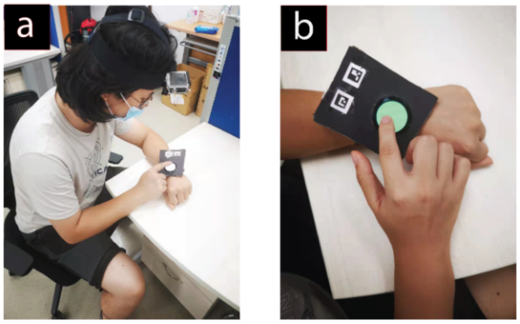

Figure 2: Study setup. a) this image shows the placement of head strap and the camera in Study 1. Participants had to put their wrist on the table to prevent fatigue. b) shows the location and orientation of QR indicators that were used in our image/video processing. Participants had to hold their finger on the target for 3 seconds after hitting the target.

blindness test). Participants were compensated with a $15 gift card for their time.

3.4 Procedures

This study was conducted during the COVID-19 pandemic, so we followed the guidelines of the university ethics board to conduct the study safely for both the participants and the researchers. At the beginning of each experiment session, the researcher ensured that the smartwatch and all devices, including the camera and head strap, were appropriately sanitized. We also asked all of our participants to answer a set of health-related questions to make sure they did not have COVID-related symptoms. We also made sure that they did not travel outside of the province for the past two weeks. Most importantly, we tried to minimize direct interaction with participants during the experiment to maintain social distancing.

Upon arrival, and after following COVID-related health guidelines, we explained the primary goal of the experiment to our participants, and asked them to read and sign the consent form to start the experiment. To maintain consistency, we asked our participants to wear the smartwatch on their left hand and interact with the smartwatch display with their right index finger. This kept the image processing simple and consistent across all participants.

In the next step, we asked our participants to wear the head strap with the mounted camera. We ensured the strap was comfortable but also tightly secured. The camera was placed at the center of the participant's forehead and toward the smartwatch display to capture the interaction with the screen following the guidelines from previous papers [16, 28, 28].

3.5 Video Processing

We used a similar approach to Neshati et al. [16] to detect the occluded area of the smartwatch. In this section, we describe the steps taken that are different from Neshati et al. [16] approach that addresses some of the limitations of that approach. In lieu of capturing static images of the participant interacting with the screen, a video of the complete session was recorded. Note that the screen turns green when the participant touches the targets and remains so for 3 seconds. This screen colour change is used to synchronize the events generated on the smartwatch with the recorded video.

Extracting multiple frames for each touch event from the video allows removing frames that have artifacts such as motion blur by comparing the calculations from a series of frames and removing the frames that are outliers. Choosing green (Figure 4a & Figure 3b) as the colour for the background also helps with differentiating the edge pixels of the smartwatch screen from the edge pixels of the finger. To further differentiate the two types of edge pixels, the calculated visible screen area was made black (Figure 3d) prior to calculating which type each edge pixel is. This ensures the edge pixels of the smartwatch (red edge in Figure 4b) have much lower values on the red channel since these pixels will be surrounded mainly by black pixels. While the edge pixels of the finger (blue edge in Figure 3b) would have more pixels with higher values on the red channel. These adjustments allowed the parameters of image processing to be less sensitive and therefore more robust across participants and trials.

Another issue with the previous approach was the different lighting conditions and location of the smartwatch on a given frame. The former was rectified by more closely managing the lighting of the environment and allowing time for the camera's auto colour balance to stabilize. The latter was caused by how different participants hold their hands when interacting with the smartwatch. Compared to asking the participant to hold their hand in a particular position or using a stand to ensure the same posture is used by different participants, capturing the occlusion under natural interactions is beneficial to the following analysis as it would be more representative of how participants would interact with the smartwatch in the real world. An assumption we make in designing this study is that for a given hand posture, the occlusion by moving only the head with the gaze fixed on the smartwatch would not vary much. To reduce the manual intervention needed to calculate the position of the smartwatch on each frame, fiducial markers were used. A frame for the smartwatch containing the markers was designed and attached to the smartwatch (see Figure 3a). The frame was placed such that the markers would always be on the opposite side from which the participants would interact with the screen. The frame essentially extends outward in one direction, and the markers were attached on the frame such that they were aligned to the direction in which the frame is being extended. Since the actual size of the markers was known, an offset from a given corner of any marker to the center of the screen can also be calculated. Since the markers are placed so that they are aligned to the direction in which the frame is extending, and the watch is circular, this offset would remain the same for any orientation in which the frame is attached. These measurements could then be translated to pixels, which would inform the center of the screen on the frame as well as provide a scale used to calculate the size of the image to crop from the frame prior to performing the occlusion calculations (the blue square on Figure 4a represents the cropping region on the original frame, Figure 3b-d are frames cropped from Figure 3a). Multiple different markers are placed on the frame to improve robustness. As the tracking of the fiducial markers ignores frames that are not clear, it also functions as another measure to discard frames on top of the outliers being calculated as described above.

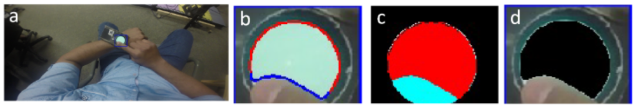

Figure 3: a) Frame as captured by the camera. The blue rectangle represents the region that will be cropped for further processing based on the fiducial markers, b) Extracted edges of the screen colour-coded based on the type: Red for edge pixels of the smartwatch, Blue for edge pixels of the finger, c) Fitted ellipse and occluded region, and d) Image used to calculate the edge pixel type. The detected screen region is replaced by black.

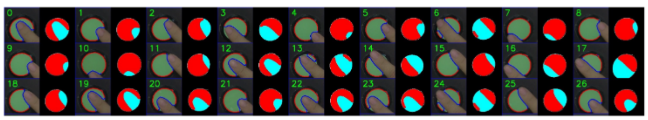

Figure 4: The result of our video/image processing algorithm for one of the participants. Samples 1,2,5,8,9,10 show that the interaction with segments on the right, bottom and bottom-right side of the smartwatch display have minimum screen occlusion. On the other hand, samples 0,6,13,14,19,and 24 show that interaction with segments on the top-left corner of the smartwatch leads to the highest screen occlusion levels

3.6 Results

Figure 5 illustrates the results of the screen occlusion study in the form of a heatmap. Our analysis shows that interaction with the segments on the top-left quarter of the smartwatch display has the worst screen visibility with an average of 44.35% screen visibility compared to 74.22%, 73.31%, and 86.04% for bottom-left, top-right and bottom-right quarters respectively. In addition, 96% and 35% are the best and worst screen visibility, which belong to the top-left and bottom-right quarters of the smartwatch display, respectively. Figure 5 shows that interacting with the outermost segments (smartwatch bezel) on the right, bottom, and bottom-right side of the smartwatch display provide the best screen visibility, with a minimum of 89% and a maximum of 96% screen visibility. Segments on the smartwatch bezel on the top-left corner of the display provide the least screen visibility with a minimum of 38% and a maximum of 53% screen visibility.

Although segments of the smartwatch bezel on the right, bottom-right and bottom size of the display have the best screen visibility, adjacent segments to the smartwatch bezel (inner segments on the smartwatch display) have high screen visibility as well, making them potentially good options for an interactive region to facilitate explore graphs on the smartwatch display.

3.7 Discussion

While earlier related work examined screen occlusion when interacting with the smartwatch bezel [16], we offer a broader set of results to include the entire smartwatch display. Our analysis confirmed the result of screen occlusion reported by Neshati et al. [16], showing that interaction with the outermost segments

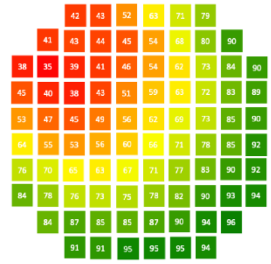

Figure 5: Screen visibility of smartwatch display while participants were interacting with different segments. The number on each segment represents the average percentage of screen visibility when participants hit the target on that specific segment of the display.

(smartwatch bezel) on the right, bottom-right and bottom side of the smartwatch display ensures the highest level of screen visibility. Interacting with the top left corner of the smartwatch bezel provides least screen visibility. Our results suggest that we can use a sufficiently large region, taking less than 10% of the entire screen space to interact with all contents on the display. This provides us an opportunity to facilitate interacting with multiple graphs using minimal finger movement around the display. We next demonstrate how we can use our selected regions with more than 90% screen visibility to interact with graphs on the small smartwatch display.

4 INTERACTION TECHNIQUE

This section will introduce EdgeSelect, a linear target selection interaction technique that can interact with multiple graphs, as an application, on small screens of smartwatches with minimal screen occlusion. The result of the first experiment revealed how interaction with different segments of the smartwatch could affect the screen visibility from the user's perspective. The first row of Figure 6 shows all the segments of the smartwatch with at least 90%, 85% and 80% screen visibility. To design our interaction technique, we will only focus on the segments with at least 85% and 90% screen visibility. We will call these two areas, Large and Small interaction areas, respectively, throughout this paper. We excluded segments with at least 80% screen visibility because they occupy more than 10% of the smartwatch display, which means the interaction area is significantly larger than two other interaction areas.

The border of selected segments for both Large and Small interaction areas is jagged (Figure 6 first row). This makes it hard and less intuitive for the smartwatch users to move and keep their fingers within the interaction area. Because of this, we smoothed the border of the interaction area for both the Large and Small interaction areas (Figure 6 second row). The curved shape of the interaction area helps the smartwatch users to move their fingers along the curved display bezel, making the interaction more intuitive for smartwatches with circular displays.

Unlike BezelGlid, the main goal of EdgeSelect is to take advantage of the width of the interaction area as a selection tool to select one of the multiple graphs on the smartwatch display. Therefore, by dividing the width of the interaction area into a number of bands (Figure 6 second row), each band can be used to interact with one graph. Our pilot study showed that three is the optimum number of bands for the interaction area in both Large and Small interaction areas. The Inner-band is the layer that is closest to the center of the display. The Middle-band is the layer in the middle and the Outer-band is the layer closest to the smartwatch bezel.

Each of the three bands of the interaction area can be used to interact with one graph. A smartwatch user can start interacting with a specific graph by hitting the correct band of the interaction area. As soon as the user interacts with a band, the outer border of the corresponding graph will be highlighted to show the selected graph. If the user hits a wrong band, the user can slide his/her finger to reach the correct band. Sliding the finger within a specific band will allow the user to interact with the data points of the corresponding graph, see Figure 7.

Figure 6: First row: represents all the segments of the smartwatch display with at least 90%, 85% and 80% screen visibility. Second row: three different interaction areas corresponding to each screen visibility level. The width of the interaction area is divided into three bands. Each band will be dedicated to interact with one graph

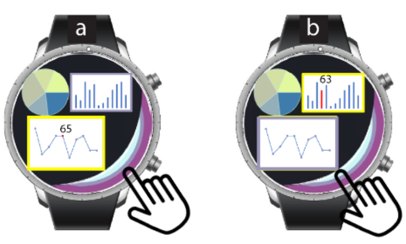

Figure 7: a) Interacting with line graph using the Outer-band of interaction area b) Sliding the index finger into the Middle-band to interact with bar chart. Moving from one band to another band changes the highlighted graph to make it easier for the user to identify the graph they are interacting with

5 STUDY 2: SIZE OF THE INTERACTION AREA AND THE NUMBER OF DATA POINTS

The size of the interaction area in EdgeSelect is inversely proportional to the size of the visible area for displaying content on the screen. Thus Study 2 involved a series of target acquisition tasks designed to evaluate participants' performance while interacting with two different sizes of interaction areas: Small (area with screen visibility of 90% or higher) and Large (area with screen visibility of 85% or higher). We measured how each of the three interaction area bands could be used to interact with graphs of differing densities, in terms of target selection time, for each size of the interaction region. We also measured the maximum number of data points per graph that users can effectively interact with using EdgeSelect.

In order to measure the maximum number of data points, we used three linear arrays of targets (Figure 8) instead of actual graphs. This enabled us to have precise control over the number and position of targets for target acquisition measurement. Each array represents a graph, and the targets represent graph data points.

5.1 Experimental Task

In order to determine the maximum number of points a user can interact with on a single graph, we varied the number of points on the graphs dynamically, depending on the success or failure of the participant. **Selection was triggered upon finger lift-off.** If a participant was able to select the specific highlighted target on a graph within a specified time limit (5 seconds), we considered that trial a success, and increased the number of points in that graph. A failed trial occurred when the participant failed to select the highlighter point in the time allotted or if the wrong iem was selected. In the event of a failed trial we decreased the number of points in that graph. We adapted the graphs to generate a point randomly in four segments of the graph before increasing or decreasing the number of points (Figure 8). Generating targets in 4 segments forced the participant to select a point in 4 different regions of the interaction area, one far to the left, two in the middle, and one to the far right. This ensured that every part of the interaction area was tested. In general, as the number of points increased, the difficulty in selecting the point increased. Over a series of trials, the participant would converge on an optimal maximum number of points in each graph, failing when the graph is too dense, and succeeding when the graph was below or at the optimal maximum density. If the participant selected targets successfully in 3 or more segments, the number of points in the graph would be increased. Otherwise, the number of points would be decreased (Figure 8). The experiment ended if the number of points decreased 3 times to the same level. For example, if the participant was seeing 10 points, failed target acquisition twice, and the number of points decreased to 9, then succeeded at target acquisition 3 times, and increased the number of points back to 10, but then failed again, causing the number of points to decrease to 9 twice more (total of 3 times decreasing to 9), the experiment would end.

5.2 Participants

We recruited 12 (new) participants (8 males and 4 females, $M_{age} = 22.08$, $SD = 3.50$) in Study 2. Our participants were all right-handed and used their watches on their left wrist. Four of these participants

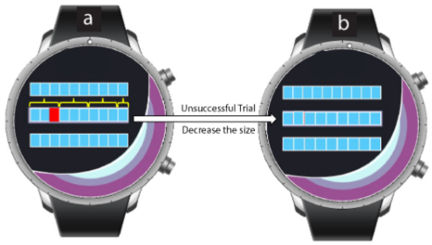

Figure 8: a) This image represents three arrays of targets. Our participants had to hit the highlighted target. In this experiment, the Inner-band interacts with the top array, the Middle-band interacts with the middle array, and the Outer-band interacts with the bottom array. To ensure participants interact with all the segments of each band, we divided the arrays into 4 major segments and picked one random target from each segment (indicated by yellow brackets in the middle array). b) If participants successfully perform the target selection task, we increased the number of targets in that specific array.

used smartwatches regularly, two occasionally, and four rarely used smartwatches. We compensated our participants with a $15 gift card for their time.

5.3 Apparatus

As in Study 1, we used the Samsung Galaxy Watch Active 2 smartwatch. The smartwatch-based interaction software was implemented aa a web app, using HTML, JavaScript, and CSS and ran natively on the watch.

5.4 Procedure

Similar to the previous experiment, we followed all COVID-related health guidelines to ensure the safety of participants and researchers. After participants arrived, we explained the objectives of the experiment. After signing the consent form, one of the researchers explained the experiment task and study progression to the participants. They were asked to practice with the interaction area and three bands to hit the random targets for as long as needed to feel comfortable (Figure 8). They were told that accuracy and response time were the two critical measurements, so they had to perform as quickly *and* as accurately as possible. We had two main conditions in this study: Large and Small interaction areas and, accordingly, different sizes of bands (Figure 6). We used a within-subject design, so all participants interacted with both the Large and Small interaction areas.

Similar to the first study, we asked our participants to put their hands on a table to prevent fatigue. They could also have a break between trials if needed. When they finished the experiment, we asked them to answer some open-ended questions regarding interacting with EdgeSelect and using different bands.

5.5 Results

In this section we describe the outcome of this experiment.

5.5.1 Response Time. The result of a Shapiro-Wilk test revealed that response time data was not normally distributed ($p < 0.05$). Accordingly, to identify the significant differences between conditions, Friedman and Wilcoxon signed-ranked tests were conducted. Bonferroni correction was applied to minimize Type 1 error.

The result of the Wilcoxon test showed that there was a significant difference between the the Large and Small interaction areas (Figure 9-left, $p < 0.001$, Large; Mdn = 1788ms, Small; Mdn = 1754ms). This shows that the smaller interaction area results in a faster response time. Our further data analysis showed that there was a significant difference between Large and Small interaction areas for the Inner-band condition (Figure 9-middle, $p < 0.002$, Large-Inner-band; Mdn = 1884ms, Small-Inner-band; Mdn = 1795ms), Middle-band (Figure 9-middle, $p = 0.038$, Large-Middle-band; Mdn = 1734ms, Small-Middle-band; Mdn = 1765ms), and the Outer-band (Figure 9-middle, $p < 0.001$, Large-Outer-band; Mdn = 1769ms, Small-Outer-band; Mdn = 1682ms). The result of Friedman test showed that there was not a significant difference between Inner-, Middle- and Outer-bands across both Large and Small interaction areas Figure 9-right.

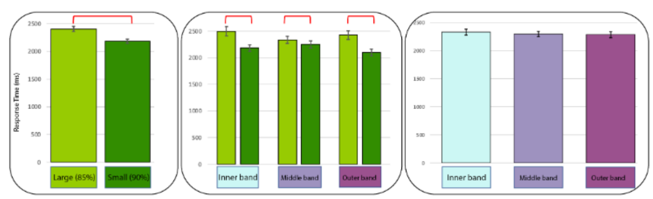

Figure 9: Comparing the response time of participants interacting with Large and Small interaction areas (left), and different bands of interaction area (right). The middle chart shows the performance of participants interacting with all combinations of bands and interaction areas. The error bar shows 95% confidence interval.

5.5.2 Number of Points. In this section, we will analyze the general pattern, as well as the minimum and the maximum number of points our participants reached using EdgeSelect. We will analyze how the size of the interaction area and each of the three bands could affect the participants' performance.

Our data analysis showed that there is a significant difference between the Large and Small interaction area across all trials (Figure 10-left, $p < 0.001$, Large-overall; Mdn = 10 points, Small-overall; Mdn = 11 points). Deeper analysis showed that there is a significant difference between the minimum number of points our participants reached, between the Large and Small interaction areas (Figure 10-left, $p < 0.001$, Large-min; Mdn = 7 points, Small-min; Mdn = 9 points). A similar result captured for maximum number of points our participants achieved, between two Large and Small conditions (Figure 10-left, $p < 0.001$, Large-max; Mdn = 11 data points, Small-max; Mdn = 12 data points). The result from a Friedman test revealed that there was a significant difference between the Inner-band, Middle-band, and Outer-band in terms of number of points. By comparing the conditions we observed a significant difference between Inner-band and Middle-band conditions (Figure 10-right, $p < 0.001$, Inner-band; Mdn = 10 points, Middle-band; Mdn = 11 points) and between Middle-band and Outer-band conditions (Figure 10-right, $p = 0.02$, Outer-band; Mdn = 10 points, Middle-band; Mdn = 11 points).

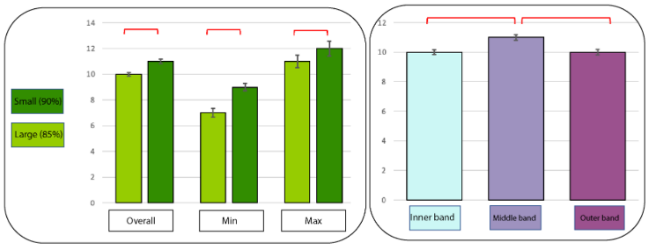

Figure 10: Left: Overall average, as well as the minimum and maximum number of points participants could select in this experiment using the Small and Large interaction areas. Right: Average number of points participants were able to select in each band across both Large and Small interaction areas. (red brackets indicate pairwise significance). The error bar represents 95% confidence interval.

Figure 11 shows all the consecutive (successful and unsuccessful) trials across all study participants, with each graph representing trials in a different interaction band. Each separately-coloured line represents the trials of one participant until they reached the end of the experiment. The y- and x-axis of all three graphs are scaled to 25 and 50 respectively, making it easier to compare the existing patterns. Comparing these three graphs shows that in general, trials in the Outer-band terminated faster than trials in the Inner-band and Middle-band (Figure 11). Visually comparing three graphs in figure 11, it shows that it took longer for our participants to finish the trials in the Middle-band condition compared to other bands. One potential reason could be that it was challenging for our participants to keep their finger precisely in the Middle-band and not overlapping with other Inner- and Outer-bands. Moving the finger out of the interaction area, Middle-band, increases the task completion time and decreases the accuracy of the target selection task. This could be the reason for some of participants and in general, it took longer for our participants to perform the tasks

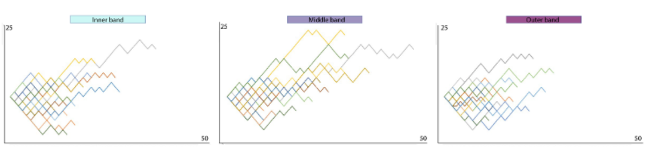

Figure 11: Number of points (y-axis) of all trials (x-axis) for each participant (each line).

5.5.3 Failed Trials. In this section, we will analyze how many times participants failed the trials. A failed trial occurred either when the user lifted their finger from the incorrect target, or when the 5 seconds elapsed without the user selecting a target. Our analysis showed that participants significantly failed more in the Large interaction area compared to the Small interaction area ($p < 0.001$, Large; Mdn = 9 fails, Small; Mdn = 7 fails). We also found that the number of failures was significantly different only between the Inner-band and Middle-band ($p < 0.038$, Inner-band; Mdn = 7 fails, Small; Mdn = 9 fails) across both Large and Small conditions.

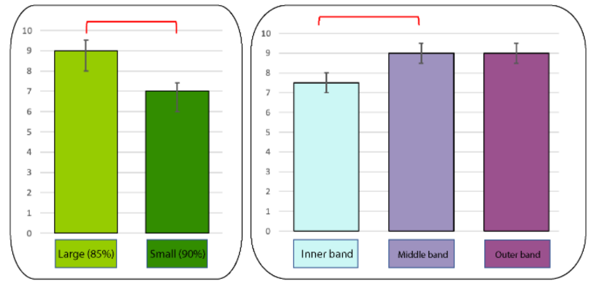

Figure 12: Number of failed trials split by size of the interaction area (left) and three bands EdgeSelect (right). The error bar shows 95% confidence interval

5.5.4 Qualitative Analysis. At the end of the experiment, we asked all of our participants to fill out a questionnaire about the interaction technique and experiment. We asked them to rank the interaction layers (Inner-, Middle-, and Outer-band) based on their preference and easiness. In addition, we asked our participants which interaction area size they preferred (Large or Small). Seventy-five percent of our participants chose the Large area. Many of our participants explained that it was easier to interact with compared to the Small interaction area. For instance, P6 commented, "Touch area is large and easy to touch", P5 and P3 also mentioned that they did not have any "accidental touch in the large area" and that it "feels more precise" in the large interaction area. In addition, nearly 60% of the participants preferred to use the Outer-band to interact with graphs (Figure 13). Only 8% of the participants picked the Inner-band as their favourite layer to interact with.

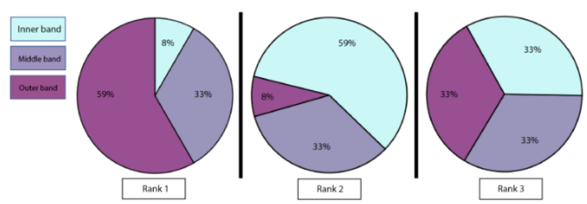

Figure 13: Proportion of each of the three bands' rankings. Outer-band was the most preferred layer of EdgeSelect. Nearly 60% of participants preferred to interact with this Outer-layer (left).

5.6 Discussion

Our results show that the Small interaction area enables better target selection performance than the Large interaction area in terms of response times and the maximum graph density that participants could interact with. Our data analysis showed that interaction with the Small area yields a significantly faster performance (almost 500ms faster) than the Large area. When the interaction area is smaller, it is faster to scroll and navigate through the interaction area, which means that the user can interact with more data points in a shorter period of time.

The average maximum number of targets our participants could reach, while interacting with the Small area was slightly, but significantly, higher than with the Large interaction area. The smaller size made the interaction faster to reach the targets, and that means even if participants made a mistake in target selection, they had time to correct their selection. In the Large interaction area, since it took longer for our participants to navigate through the interaction area, correcting a target selection error could take longer, which could be considered as a fail trial.

In addition, the minimum number of targets participants reached was significantly higher in the Small condition, which means they made fewer mistakes while interacting with the Small interaction area. Figure 10-left also shows that the number of failures was lower in the Small interaction area.

Although there was no difference between the three different layers of the EdgeSelect interaction area across the Small and Large trials, in terms of response time (Figure 9-right), the number of targets they could reach using the Middle-band was higher compared to other bands. This means that the Middle-band is a good option to interact with more dense graphs with a higher number of data points.

While EdgeSelect was designed and evaluated on a smartwatch with a circular form factor, a slightly modified version of EdgeSelect can be used on smartwatches with a rectangular display. The same three bands can be implemented as the interaction area on the smartwatch display's bottom right corner, including a portion of the right and bottom edge and the entire bottom-right corner. However, to make the interaction continuous and more natural (similar to a circular smartwatch), the interaction area on the bottom right corner can have a slight curve (not angular). This prevents the user from going to the watch's bottom right corner and suddenly changing direction, making it a two-step interaction technique.

6 GENERAL DISCUSSION

In this section we discuss the critical findings and potential applications, and limitations of the EdgeSelect technique.

6.1 Key Findings

Result of the first study reveal that interacting with the segments at the bottom, bottom-right, and right side of the smartwatch display yields the best screen visibility, with an average of 86.7%, 90% and 85% screen visibility, compared to other segments of the smartwatch display. As we move toward the center of the smartwatch display and the top-left corner, the screen visibility decreases significantly. Interaction with the top-left corner of the smartwatch display has the worst screen visibility with an average of only 40.9% of the screen being visible. The outermost segments at the right-bottom corner of the display offer a 92% or higher screen visibility, confirming the result reported by [16], which measured the screen occlusion only for the segments of the smartwatch bezel. However, our experiment showed that interacting with inner segments of the smartwatch display can also provide a reasonable amount of visibility to users (more than 80% screen visibility). Including these specific areas of the smartwatch display, as the interaction area, without sacrificing the screen visibility, can help smartwatch app designers to use the width and length of EdgeSelect to select and interact with graphs or other types of content that requires target selection and continuous linear interaction.

In the second experiment, we showed that using segments of the smartwatch display with at least 90% screen visibility (Small interaction area) outperformed the interaction with the segments of the smartwatch screen with at least 85% screen visibility (Large interaction area), in terms of response time and errors. This means that the Small interaction area occupies less smartwatch screen real estate and enables faster and more accurate user interaction with graphs and data points. We also found that the Middle-band in the interaction area is more suitable to interact with graphs with a higher number of data points (e.g., line/bar graphs) compared to Inner- and Outer-band which are suitable to interact with less dense graphs (e.g., pie/donut chart).

6.2 Potential Applications

EdgeSelect can be used to interact with content other than graphs such as virtual keyboards (Figure 14-a), sliders (smartwatch setting, Figure 14-b), navigating a music track (Figure 14-c), and hierarchical menu selection (Figure 14-d). EdgeSelect can be used to interact with three-row virtual keyboards. Each row of the virtual keyboard can be mapped to each band of the interaction area. To make the interaction intuitive, the top-row of the virtual keyboard can be mapped to the inner-band, middle-row to middle-band and the bottom-row of the keyboard to the outer-band of the EdgeSelect. There are approximately nine letters per row (Figure 14-a). As the results of our second experiment suggest, using the EdgeSelect can effectively interact with such a number of points in each row of the virtual keyboard. By sliding the index finger on the right band, the smartwatch user can select the right letter they want to type. Selection can be made by moving the finger a bit higher than the smartwatch display or by holding the finger in that position for a short period of time (e.g., 2 seconds).

Interacting with sliders (e.g., increasing and decreasing smartwatch brightness and sound volume) and similarly navigating through a music player are two other examples of continuous interaction using EdgeSelect. EdgeSelect can also be used to interact with nested or hierarchical menus. The first layer of the interaction area can interact with the highest level of the menu. Then there will be items from the sub-menu (second row, Figure 14-d). The second band can be used to select the item from the sub-menu. The same process may happen with the third layer of the menu and the last band of the interaction area. For instance, the user can select a workout application from the apps menu, then the type of workout (e.g., running) and then the duration s/he wants to workout.

6.2.1 Limitations and Future Work. We decided to use arrays of targets in our target selection experiment as using actual graphs would add more complexity by adding too many confounding variables. Although we demonstrated that EdgeSelect was designed to interact with graphs with different densities (Figure 7), formal qualitative and quantitative evaluation could confirm the effectiveness of EdgeSelect on actual graphs. In the future, we will quantify the efficiency of using EdgeSelect to interact with various types of graphs with different levels of density.

Figure 14: a) Each of the three EdgeSelect bands can be used to interact with a three-row virtual keyboard on the smartwatch without blocking the letter while typing. b, c) Using EdgeSelect to interact with sliders such as adjusting settings and navigating through music tracks. d) EdgeSelect can also be used in hierarchical menus for item selection.

Another limitation of this work is that the two experiments were done in the lab. Smartwatch users employ their smartwatches on-the-go and in different mobility conditions. Running an in-the-wild study with participants running and being outdoors, would make data collection (such as measuring the screen visibility) very difficult and inaccurate. many factors, such as lighting, could affect the result of our video processing. To reduce these types of confounding factors, we excluded mobility conditions from our current studies and controlled the environment by conducting a lab based-study. However, in reality, environment factors are constantly changing. We will further explore EdgeSelect under different ecologically valid settings.

7 CONCLUSION

This paper introduces EdgeSelect, an interaction technique that facilitates linear target selection on small smartwatch screens. EdgeSelect was inspired by the need to interact with various graphs, representing multiple interrelated data sources, which can often be shown at the same time on a smartwatch. Our design was geared at mitigating 'fat finger' and screen occlusion effects, as the interacting finger largely overlaps the small display when examining content. One of our key design goals was to shift the interactive region to an area of the smartwatch resulting in minimal screen occlusion (about 10% in our study) while optimizing input across the entire smartwatch display. To design EdgeSelect, we first conducted a study to measure the screen occlusion caused by the finger interacting with the entire smartwatch display. Our results indicate that interacting with outer segments of the smartwatch display to the right, bottom-right and bottom sides of the display offers the best screen visibility. Based on this initial result, we designed a three-layer EdgeSelect interaction technique. Each of the three layers can be mapped to interact with different graphs, making EdgeSelect one of the first interaction techniques enabling the exploration of multiple data visualizations on a small smartwatch display.

In the second experiment, we examined EdgeSelect with two different sizes of the interaction area, Large and Small, which enable inversely proportional levels of screen visibility. Our results showed that participants performed faster and more accurately with the Small interaction area (with higher screen visibility) compared to the Large interaction area (with lower screen visibility), making EdgeSelect a suitable interaction method on the small smartwatch display. We also found that the Middle-band of EdgeSelect is more suitable to interact with denser graphs (e.g., line/bar chart). On the other hand, Outer- and Inner-bands are more suitable to interact with less dense graphs (pie/donut charts). Finally, we demonstrated that EdgeSelect enables interaction with multiple graphs of different types and densities, and also can be applied to interact with a wide range of other applications, including text-entry, menu selection, and adjusting sliders. In future work, we aim to deploy EdgeSelect in ecologically valid settings to examine the breadth of exploratory in-situ analysis afforded by such a technique.

REFERENCES

[1] Fereshteh Amini, Khalad Hasan, Andrea Bunt, and Pourang Irani. 2017. Data representations for in-situ exploration of health and fitness data. In *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare.* 163–172.

[2] Ranjini Aravind, Tanja Blascheck, and Petra Isenberg. 2019. A survey on sleep visualizations for fitness trackers. In *Posters of the European Conference on Visualization (EuroVis).*

[3] Emeline Brulé, Gilles Bailly, Marcos Serrano, Marc Teyssier, and Samuel Huron. 2017. Investigating the design space of smartwatches combining physical rotary inputs. In *Proceedings of the 29th Conference on l'Interaction Homme-Machine.* 13–20.

[4] Juri F Buchmüller, Udo Schlegel, Eren Cakmak, Daniel A Keim, and Evanthia Dimara. 2021. SpatialRugs: A compact visualization of space and time for analyzing collective movement data. *Computers & Graphics* (2021).

[5] Tak chung Fu, Fu lai Chung, Robert Luk, and Chak man Ng. 2007. Stock time series pattern matching: Template-based vs. rule-based approaches. *Engineering Applications of Artificial Intelligence* 20, 3 (2007), 347 – 364. https://doi.org/10.1016/j.engappai.2006.07.003

[6] YN Dimitrov, VP Aleksieva, and HG Valchanov. 2021. Comparative analysis of prototypes for two-touch finger interfaces of smartwatch. In *IOP Conference Series: Materials Science and Engineering*, Vol. 1032. IOP Publishing, 012019.

[7] Teng Han, Khalad Hasan, Keisuke Nakamura, Randy Gomez, and Pourang Irani. 2017. Soundcraft: Enabling spatial interactions on smartwatches using hand generated acoustics. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology.* 579–591.

[8] 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.* 121–124.

[9] Alaul Islam, Anastasia Bezerianos, Bongshin Lee, Tanja Blascheck, and Petra Isenberg. 2020. Visualizing information on watch faces: A survey with smartwatch users. In *2020 IEEE Visualization Conference (VIS).* IEEE, 156–160.

[10] Rhongho Jang, Changhun Jung, David Mohaisen, Kyunghee Lee, and Daehun Nyang. 2021. A One-Page Text Entry Method Optimized for Rectangle Smartwatches. *IEEE Transactions on Mobile Computing* (2021).

[11] Frederic Kerber, Tobias Kiefer, Markus Löchtefeld, and Antonio Krüger. 2017. Investigating current techniques for opposite-hand smartwatch interaction. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services.* 1–12.

[12] Yuki Kubo, Buntarou Shizuki, and Jiro Tanaka. 2016. B2B-Swipe: Swipe gesture for rectangular smartwatches from a bezel to a bezel. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems.* 3852–3856.

[13] Sakorn Mekruksavanich, Anuchit Jitpattanakul, Phichai Youplao, and Preecha Yupapin. 2020. Enhanced hand-oriented activity recognition based on smartwatch sensor data using lstms. *Symmetry* 12, 9 (2020), 1570.

[14] Vivian Genaro Motti. 2018. Smartwatch Applications for Mental Health: A Qualitative Analysis of the Users' Perspectives. *Jmir Ment. Health* (2018).

[15] Ali Neshati, Fouad Allalah, Bradley Rey, Yumiko Sakamoto, Marcos Serrano, and Pourang Irani. 2021. SF-LG: Space-Filling Line Graphs for Visualizing Interrelated Time-series Data on Smartwatches. In *Proceedings of the 23th International Conference on Human-Computer Interaction with Mobile Devices and Services.* 1–12.

[16] Ali Neshati, Bradley Rey, Ahmed Shariff Mohommed Faleel, Sandra Bardot, Celine Latulipe, and Pourang Irani. 2021. BezelGlide: Interacting with Graphs on Smartwatches with Minimal Screen Occlusion. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* 1–13.

[17] Ali Neshati, Yumiko Sakamoto, and Pourang Irani. 2019. Challenges in Displaying Health Data on Small Smartwatch Screens.. In *ITCH.* 325–332.

[18] Ali Neshati, Yumiko Sakamoto, Launa C Leboe-McGowan, Jason Leboe-McGowan, Marcos Serrano, and Pourang Irani. 2019. G-Sparks: Glanceable Sparklines on Smartwatches.. In *Graphics Interface.* 23–1.

[19] Ian Oakley, Carina Lindahl, Khanh Le, DoYoung Lee, and MD Rasel Islam. 2016. The flat finger: Exploring area touches on smartwatches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems.* 4238–4249.

[20] Gulnar Rakhmetulla and Ahmed Sabbir Arif. 2020. SwipeRing: Gesture typing on smartwatches using a segmented QWERTY around the bezel. In *Graphics Interface 2021.*

[21] Léa Saviot, Frederik Brudy, and Steven Houben. 2017. WRISTBAND. IO: expanding input and output spaces of a Smartwatch. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems.* 2025–2033.

[22] Marcos Serrano, Eric Lecolinet, and Yves Guiard. 2013. Bezel-Tap gestures: quick activation of commands from sleep mode on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 3027–3036.

[23] Katie A Siek, Yvonne Rogers, and Kay H Connelly. 2005. Fat finger worries: how older and younger users physically interact with PDAs. In *IFIP Conference on Human-Computer Interaction.* Springer, 267–280.

[24] Edward R Tufte. 2006. *Beautiful evidence.* Vol. 1. Graphics Press Cheshire, CT.

[25] Huub van de Wetering, Nico Klaassen, and Michael Burch. 2020. Space-reclaiming icicle plots. In *2020 IEEE Pacific Visualization Symposium (PacificVis).* IEEE, 121–130.

[26] Daniel Vogel and Patrick Baudisch. 2007. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI conference on Human factors in computing systems.* 657–666.

[27] Daniel Vogel and Géry Casiez. 2012. Hand occlusion on a multi-touch tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 2307–2316.

[28] Daniel Vogel, Matthew Cudmore, Géry Casiez, Ravin Balakrishnan, and Liam Keliher. 2009. Hand occlusion with tablet-sized direct pen input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 557–566.

[29] Pui Chung Wong, Kening Zhu, Xing-Dong Yang, and Hongbo Fu. 2020. Exploring Eyes-free Bezel-initiated Swipe on Round Smartwatches. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems.* 1–11.

[30] Haijun Xia, Tovi Grossman, and George Fitzmaurice. 2015. NanoStylus: Enhancing input on ultra-small displays with a finger-mounted stylus. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology.* 447–456.

[31] Hui-Shyong Yeo, Juyoung Lee, Andrea Bianchi, and Aaron Quigley. 2016. WatchMI: pressure touch, twist and pan gesture input on unmodified smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services.* 394–399.

[32] Xin Yi, Chun Yu, Weijie Xu, Xiaojun Bi, and Yuanchun Shi. 2017. Compass: Rotational keyboard on non-touch smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.* 705–715.

[33] Cheng Zhang, Junrui Yang, Caleb Southern, Thad E Starner, and Gregory D Abowd. 2016. WatchOut: extending interactions on a smartwatch with inertial sensing. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers.* 136–143.