Evaluating the Speed and Accuracy of Touch Input at the Edge of a Table

by

Nikhita Joshi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2019

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis includes first-author content from the following peer-reviewed conference publication:

- Nikhita Joshi and Daniel Vogel. 2019. **An Evaluation of Touch Input at the Edge of a Table**. *In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Paper 246, 12 pages. DOI: https://doi.org/10.1145/3290605.3300476

The content from this paper has been adapted and extended for this thesis.

Abstract

Tables, desks, and counters are often nearby, motivating their use as interactive surfaces. However, they are typically cluttered with objects. As an alternative, we explore touch input along the 'edge' of table-like surfaces. The performance of tapping, crossing, and dragging is tested along the two ridges and front face of a table edge. Results show top ridge movement time is comparable to the top face when tapping or dragging with the index finger. When crossing, both ridges are at least 11% faster than the top face. Effective width analysis is used to model performance and provide recommended target sizes. Based on observed user behaviour, we explore top and bottom ridge crossing using a "braced" thumb and provide design recommendations with example applications.

Acknowledgements

I would like to thank my amazing advisor, Dan Vogel, for his support and guidance. There is no doubt that Dan is an incredibly smart and talented professor, but what sets him apart is how much he cares about his students and seeing them succeed. I am so thankful to have gotten the chance to work with him these past couple of years and I am thrilled to continue working with him during my PhD. Who knew that a day at UROC and a shared love for Sackville, New Brunswick would result in all of this! Thanks for everything you do, Dan.

I was lucky to have worked with talented engineers and researchers during my two internships. Thanks to the entire prototyping team at North Inc. and the UIR team at Autodesk Research for their mentorship. I also thank Jian Zhao and Mark Hancock for their insightful comments as readers of this thesis.

The HCI lab is filled with brilliant faculty and students, and I am thankful to have had the opportunity to work alongside them. Together, we fostered a great community, both on a professional and personal level. I have a lot of fond memories with you all, including, but certainly not limited to, our walks to Starbucks, weekend dinners, and late nights (that definitely got weird) during peak CHI crunch. Thank you for making these past two years so fun. A special thanks to Keiko, Matthew, Antony, and Jeremy for allowing me to annoy them with questions and to Greg, Johann, and Lisa for always being incredible friends and sharing the DEETS about statistics and life.

Finally, none of this would have been possible without the support of my family: Radhika, Vinod, and Janaki. Even though they know nothing about computer science and HCI, they always try to learn so they can support (and brag about) me. Thank you for always being there and for believing in me. You are my pillars of strength.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With Spatial Augmented Reality (SAR) [32] any surface in an environment can act as an interactive display, and a large, flat, horizontal surface like a table, counter, or desk is often nearby. This availability motivates their use as an interactive touch surface [27, 6, 49, 40, 15] and led to several touch performance evaluations [31, 35, 18] within the human-computer interaction (HCI) community. However, tables, counters, and desks are often cluttered [10]. As such, using a large portion of the top for interaction may not be practical.

There are other unobstructed table surfaces that could support interaction. The bottom face is an obvious option [43], but a table edge has a vertical face and two one-dimensional ridges which may support common interaction techniques. Researchers have examined interaction on other types of surfaces, such as convex hemispheres [33] and raised ridges [20, 24, 48, 47, 46], but the specific surfaces located around the edge of a table have not been investigated. As the region nearest the user on a touch tabletop is faster and easier to use [8], table edge surfaces may have similar advantages.

1.1 Contributions

We compare the speed and accuracy of touch input when tapping, crossing, and dragging on three surfaces that form a table edge: a vertical *front face*, a *top ridge*, and a *bottom ridge* (Figure 1.1). We use a 1D Fitts' law [17] task and compare performance to using the *top face* and *bottom face*. An initial experiment explored using the index finger since it is most natural and common, and a second experiment extended our inquiry to using the thumb with a "braced" posture [12] (Figure 1.2).

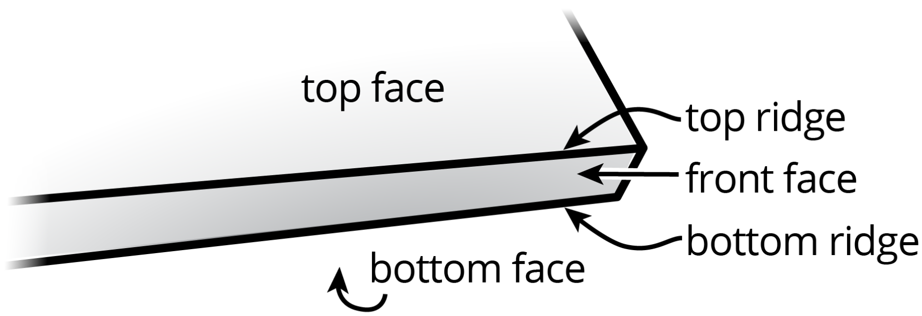
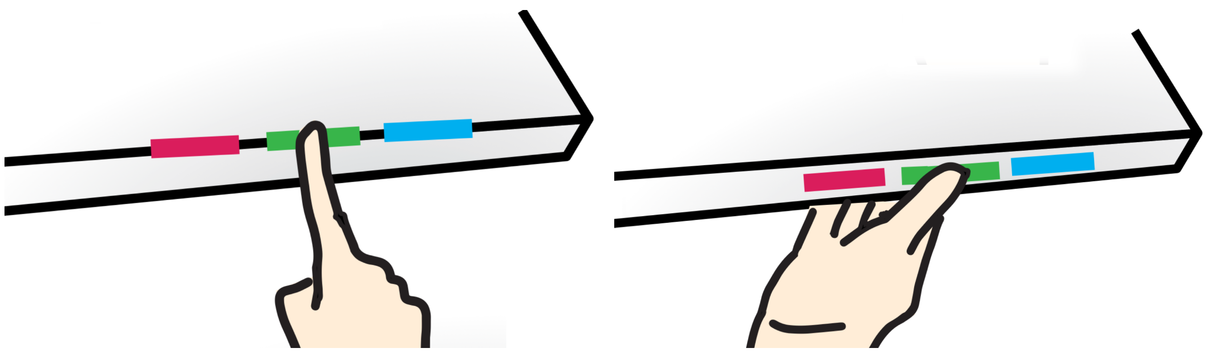


Figure 1.1: Different table edge surfaces



(a) The index finger on the top ridge.

(b) A "braced" thumb on the front face.

Figure 1.2: The two postures used in our experiments.

We contribute empirical results that validate the potential for touch input along the edge of a table. Top ridge interaction is often preferred, and its movement time is comparable to the top face when tapping or dragging with an index finger. When crossing, both ridges are at least 11% faster than the top face, and errors are below 6% for all surfaces. Effective width analysis shows minimum target widths are 7.4 to 19.2 mm. A "braced" thumb [12] has poorer performance than the index finger.

Our work sets a foundation for designers and researchers to adopt and explore new edge-of-table interactions.

1.2 Outline

This thesis is organized as follows:

- Chapter 2 outlines previous work that may influence touch input along a table edge, including touch along flat and irregular surfaces.
- Chapter 3 describes the potential design space and use cases for edge-of-table interactions.
- Chapter 4 describes the prototype system we developed to track touch along the edge of a standard, rectangular table.
- Chapter 5 describes our initial experiment, where participants used their index finger to interact with five different table surfaces.
- Chapter 6 describes our second experiment, where participants used a "braced" thumb to interact with the three table edge surfaces.
- Chapter 7 describes our design recommendations, proposed example applications for table edge interactions, and possible limitations in our methods.
- Chapter 8 concludes by summarizing our work and discussing possible avenues for future work.

Chapter 2

Background and Related Work

This work relates to Fitts' law, touch input tasks, and interactions along flat and non-planar surfaces. Given this breadth of topics, we focus on those most relevant to our study design or factors that might influence touch input on a table edge.

2.1 Fitts' Law

Fitts' law [17] is a predictive model commonly used to evaluate the performance of target selections. It models the time to acquire targets of varying widths (W) and different distances (D).

Designing Experiments

When designing an experiment that uses Fitts' law in its analyses, researchers must use many W and D values to create selection tasks of varying difficulty. This difficulty, referred to as the *index of difficulty* (ID), is calculated using D and W (Equation 2.1). It is recommended to use target selection tasks with many IDs between 2 and 8 to get representative task difficulties [36].

$$ID = \log_2 \left(\frac{D}{W} + 1 \right) \tag{2.1}$$

Conducting Experiments

To model target selection performance, participants are asked to perform specific movement tasks. Traditionally, this has consisted of one-dimensional selection tasks. A *discrete* task involves moving from a starting position to a single target with 'infinite' height and W width. The centre of this target is placed D distance away from the starting position. A *serial* task involves selecting targets repeatedly, moving back and forth between two targets of width W, with centres placed D distance apart.

These one-dimensional selection tasks are confounded by the angle of movement. As such, it is recommended that researchers implement multidirectional selection tasks [36], where square or circular targets of varying widths or diameters (W), are placed around the circumference of a circle with a diameter of D. Participants select targets in a circular pattern, moving across the diameter of the circle with every target selection. While we recognize a multidirectional selection task is considered best practice, the three edge surfaces are not large enough. As such, we use one-dimensional selection tasks.

Selection time is defined as the time moving from a starting position or the most recently acquired target to another target.

Adjusting for Accuracy

After collecting user data from selection tasks, it is recommended to adjust ID for accuracy, especially when observed error rates are high [36]. To do this, we calculate the *effective target width* (W_e), or the "true" width of the targets based on observed user data. The distribution of distances between the participant's contact point and the centre of the target is saved, and its standard deviation (σ) is multiplied by 4.133 (Equation 2.2). Note that 4.133 represents the maximum entropy of observations, and is based on a normal distribution with 4% of observations occurring in the tails [36]. Designers and researchers often use W_e to make recommendations for UI element sizes, like button sizes. This W_e is then used to calculate the *effective index of difficulty*, ID_e.

$$W_e = 4.133 \times \sigma \tag{2.2}$$

Using participants' average target selection time (MT) and ID_e, one must use a least-squares linear regression to calculate the intercept, a, and the slope, b (Equation 2.3). a and b characterize the input device or interaction technique being tested.

$$MT = a + b \times ID_e \tag{2.3}$$

Ideally, a should be a small, positive number [17, 36]. However, due to the variability of human performance, a can also be negative, but should be between -200 and 400ms [36]. a can be indicative of a constant time needed to react, or move from one input device to another (*homing time*) [36].

b describes the rate at which average selection time increases as a task becomes more difficult. As such, it provides better insight into performance than a (lower is better) [36]. The inverse of b gives *throughput* (TP, Equation 2.4). TP describes the rate of information transfer (bits/s) when using a particular input device or interaction technique. This metric combines speed and accuracy into a single measure (higher is better) and is used to understand task effectiveness [36].

$$TP = \frac{1}{b} \tag{2.4}$$

We use Fitts' law to calculate the throughput of touch input along the edge of a table using one-dimensional, serial selection tasks. We use one-dimensional, discrete selection tasks to estimate homing time (described above). Effective width analyses is used to make recommendations for minimum target width sizes for future researchers and designers.

2.2 Touch Input Tasks

While there are many touch input tasks, tapping and dragging are considered elemental interactions within the HCI community [11] and serve as a foundation for many commonly used operations, such as clicking a button and direct manipulation. Furthermore, research on crossing-based selections (e.g. [29]) suggests crossing is a suitable target acquisition method. However, table edge surfaces have varying orientations, sizes, and textures. As such, common touch input tasks may be better suited for certain edge surfaces. We discuss factors that might impact tapping, crossing, and dragging along a table edge.

Luo and Vogel [29] explore the effectiveness of different crossing-based selection tasks performed using direct touch input. Using a capacitive touch display placed on a tabletop angled towards the user, participants performed target selection tasks using targets of varying widths and heights. They found no significant difference in speed and error rate between crossing through horizontal and vertical targets. However, this work only evaluated one surface angle that was neither horizontal nor vertical. It is possible that the varying orientations of different table edge surfaces could make different crossing orientations more or less effective for touch input. Two edge surfaces are one-dimensional ridges

(Figure 1.1). As such, it is impossible to test targets of varying heights since the "height" of the ridge is essentially zero. Therefore, we primarily use horizontal crossing targets in our experiments with some evaluation for horizontal versus vertical targets on compatible table edge surfaces.

Cockburn et al. [11] study the effects of three different input methods, a mouse, a stylus, and the finger, when selecting and dragging targets. They show the friction between fingers and the touchscreen makes dragging slower than tapping. While dragging is considered a fundamental interaction within HCI, friction may also negatively affect performance when dragging along the larger top, bottom, and front faces. However, dragging along thin table ridges may create less friction.

Guerreiro et al. [24] explore the effects of different touch interactions for users with motor impairments. Participants performed target selection tasks using tapping, crossing, and crossing along the edge of the phone screen. Overall, participants preferred tapping over crossing, but crossing along the edge of the phone screen can help increase task precision [46, 48, 47, 20, 24]. The natural, one-dimensional affordance of table ridges may be more suitable for crossing, where the ridge itself becomes the crossing target.

2.3 Interacting with Large, Flat Surfaces

Previous work has focused on interacting with a single, large horizontal surface [18, 34]. A formative study for Bi et al.'s Magic Desk [8] compared touch performance over different regions of a tabletop. One-handed tasks performed in the region nearest the user were generally faster, however, they also report that this region can be occluded by the user's hands. The table edge is composed of surfaces closest to the user, so a balance between speed and hand occlusion may be a challenge. In addition, Bi et al. [8] studied two-handed tasks, and found both near and far regions were faster and easier overall. We focus on one-handed input in our experiments, but we include two-handed input in our larger table edge interaction design space (Chapter 3.2).

In addition, work has compared horizontal and vertical surfaces. Pedersen and Hornbæk [31] study the effects of differently-oriented surfaces when tapping and dragging. They show tapping is 5% faster on vertical surfaces, and dragging is 5% faster and less error-prone on horizontal surfaces. Bachynskyi et al. [4] evaluate the performance and ergonomics of different touch surfaces, including large horizontal tabletops and vertical displays. They find tapping on a tabletop yields 14% more throughput than a vertical display. The varying orientations of the table edge surfaces may impact task performance.

Forlines et al. [18] note that one-finger contact is similar at any location of a vertical touchscreen and other fingers are often "tucked" away. When interacting with targets further away on horizontal surfaces, other fingers can come in closer contact with the touchscreen and cause accidental touch input. Locations closer to the user, such as the table edge, may be more accurate for touch input on horizontal surfaces.

The area underneath the table (bottom face) is another large, horizontal surface that is often nearby but cannot be cluttered. Wigdor et al. [43] explore touch input along the bottom face. Using bimanual interaction, they found larger targets may be selected from underneath the table without visual feedback. However, related work examining mobile back-of-device interaction suggests visual feedback is preferred, if not necessary, to interact with hidden surfaces [5, 42, 46]. It is unclear how touch performance along the bottom face compares to that of the three edge surfaces. In addition, when interacting with targets on the bottom face, users must adjust the orientation of their hand, so their palm is facing the table. This posture may feel uncomfortable; interacting with the edge may feel more natural. Given these unknowns, we also evaluate touch input on the bottom face to perform meaningful baseline comparisons, but include visual feedback.

2.4 Interacting with Non-Planar Surfaces

Previous work on non-planar touch input has focused on curved devices, such as Sphere [7] and Mouse 2.0 [38]. Roudaut et al. [33] explore touch on smooth surface curvatures, and find convex surfaces increases pointing accuracy. A table edge's ridges can be considered "convex," although not smooth nor curved. It is unclear whether the table edge could also be used to improve pointing accuracy. Curve [45] and BendDesk [39] combine horizontal and vertical touch surfaces using a curved concave edge. However, Weiss et al. [39] note dragging along this curve is slower. A table edge's ridges could also be used to drag content across two differently-oriented surfaces.

Others have used raised physical ridges to increase task precision [46, 48, 47, 20, 24]. EdgeWrite [48] is a text entry technique that allows users to write letters along the ridges of a small square hole. When comparing touch to joystick input, Wobbrock et al. [47] show using EdgeWrite with touch input is preferred among participants and leads to faster interaction times. EdgeWrite is faster using the index finger on the front of a touchscreen for one- and two-handed interactions.

Although touch input along a table edge has not been examined, previous work suggests it may be a strong candidate for meaningful interactions due to its physical properties.

Chapter 3

Design Space

Table-like objects are almost always around us. We work at desks, eat at tables, gather around bars, prepare food on counters, reach for objects on shelves, and use tools at workbenches. The near-future vision for surface-mapped SAR [32] combined with high quality touch input throughout the environment using depth cameras [49, 44] or flexible capacitive sensors [23] means ubiquitous surfaces can also be used as interactive digital displays (Figure 3.1). Systems like Digital Desk [40], ambientROOM [25], LightWidgets [16], Bonfire [27], and WorldKit [49] demonstrate different applications for touch-enabled SAR, like a pervasive calculator, ambient displays, a volume control, viewing notifications, and dimming lights.

Table-like surfaces are often cluttered. Malone's [30] interviews with office workers found tables piled with documents. In Cheng et al.'s [10] survey to understand the objects people have on their desk, participants never worked at an empty table. Tabard et al. [37] report physical object occlusion as a primary issue for their eLabBench tabletop system. Others have acknowledged clutter indirectly, by proposing display techniques to accommodate a cluttered surface [28, 14, 19]. We examine interactions around a table edge as table-like objects are ubiquitous, they can be augmented using SAR and touch sensors, and they cannot be cluttered.

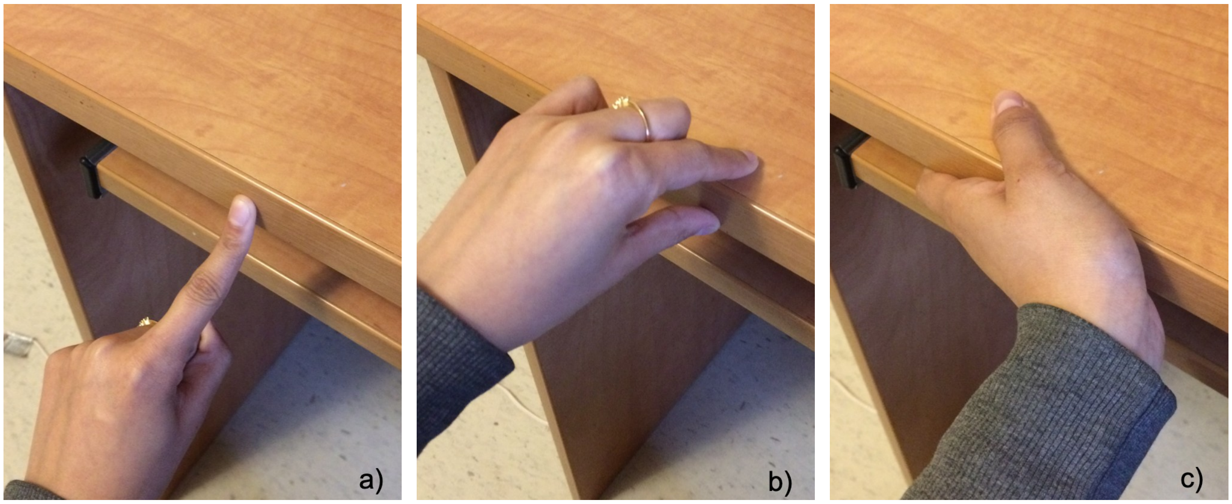
3.1 Potential Applications

The edge of a table-like object has many interesting physical properties such as faces, ridges, and corners of varying shapes, sizes, and textures that could support touch input.

Figure 3.1: Using the table edge to control remote SAR content.

Corners could replicate menu selection buttons; sliding along a straight table ridge using one or two fingers could control volume, 'flicking' the bottom ridge with one finger could trigger discrete events, like dismissing notifications; grabbing the table edge with the whole hand could pause a video player; and 'pinching' the index and thumb along a table's front and top faces, like a cross surface pinch-to-zoom, could create an input dimension that spans multiple surfaces (Figure 3.2).

Compared to the tabletop, table edge surfaces may be more easily grasped eyes-free. Shorter users, like young children, could interact with the front face due to its reachable height and access more useful IOT functions like dimming the lights in a kitchen. For users with motor impairments, touch input along a ridge may help stabilize their movements [46, 47].



(a) Single surface with 1 finger. (b) Multiple fingers 'pinching' two surfaces. (c) Grab the entire edge.

Figure 3.2: Some possible ways of interacting with the edge.

Given this large, mostly unexplored design space, we focus on touch interactions along the *edge* of a rectangular table, which consists of a front face and two ridges. Ridges are one-dimensional, but we believe they could still support meaningful interaction as sliders or crossing targets. As we are not proposing using the table's edge as a general purpose 2D touch pad, this physical property is not an issue.

There are several touch operations such as 2D rotation, scaling, and translation. Many input strategies exist: multi-touch [3, 41], bimanual [43] input, and input with different fingers [22] or finger orientations [21]. We focus on three common operations: tap, cross, and drag with the index finger and explore a "braced" [12] thumb in a second study.

As ridges are raised surfaces, the tactile feedback when tapping may seem 'button-like,' and improve performance. Ridges are also 1D in nature, which may pair well with crossing, where the ridge *becomes* the crossing target. It is unclear if dragging is suitable for edge-of-table interactions.

Chapter 4

Prototyping System

We simulate future display and touch sensing at the edge of a table using SAR projection and motion tracking.

4.1 SAR Projection Display

The system is built around a 61 cm $\times$ 122 cm rectangular table (height 72 cm and 3 cm thick top). A single 1280 $\times$ 720 resolution projector mounted to the right of the table is angled to project onto the top face, top ridge, front face, and bottom ridge. One limitation of this approach is that the hand may occlude part of the projected content. In practice, we found the size of a finger minimized occlusion of the immediate area, and people visually acquire targets before motor movement ends. The RoomAlive toolkit [26] is used to calibrate projector transformations to enable projection mapping to the table surfaces. A server (Windows 10, Core i7-6850K) runs a custom Unity3D 5.6.1 application, which processes tracked finger positions and renders projection-mapped content to the projector at 60 FPS using a GeForce GTX 1080 8G GPU (see below).

4.2 Motion Tracked Touch

A 10-camera Vicon system tracks a finger using a single 9.5 mm marker (Figure 4.1). Ten cameras track a 140 cm $\times$ 170 cm area around the front edge of the table (Figure 4.2).

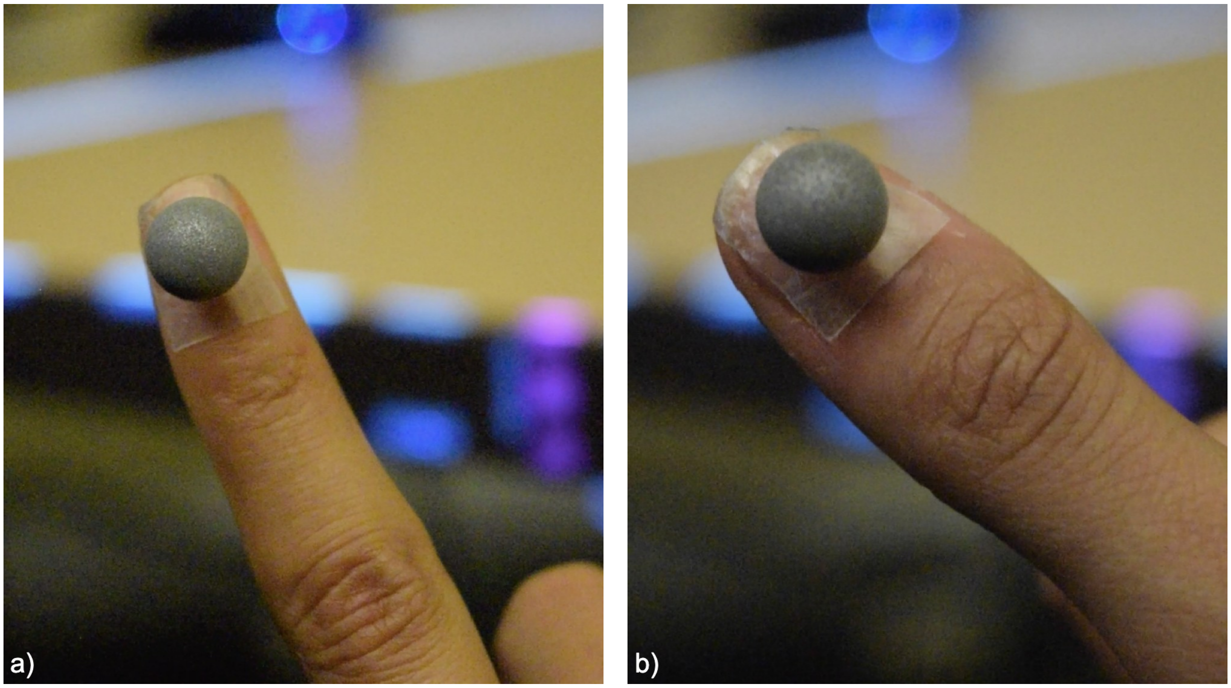


Figure 4.1: A single marker is placed on the participant's (a) index finger or (b) thumb.

We use Unity and the depth map produced by RoomAlive to create a virtual table. Its position corresponds to the position of the real, physical table placed in the space (Figure 4.3). Some manual tuning was required to precisely align the 3D model of the table to the real table. A virtual marker follows the participant's real finger movements tracked by the Vicon system. Collisions between the virtual marker and the virtual table are used to track table touch events in the real world.

To account for finger thickness and contact angle, a per-user calibration is used. For each surface, the user places their finger at the centre of two small pink targets spaced 52 cm apart. At each position, the 3D offset vector from the finger marker to the target centre is recorded. During system operation, a linear interpolation between the two 3D vectors captured for the closest surface is used to find the position of the finger relative to the tracked marker.

Tracking quality is quite good, but due to the user's hand and the table's large solid surface, the marker can be momentarily occluded causing its position to "flicker." To compensate for this, touch down events are detected when the calibrated finger position is less than 3 mm away from a surface, and touch up events when it is at least 10 mm away.

Figure 4.2: System used for testing edge-of-table interactions

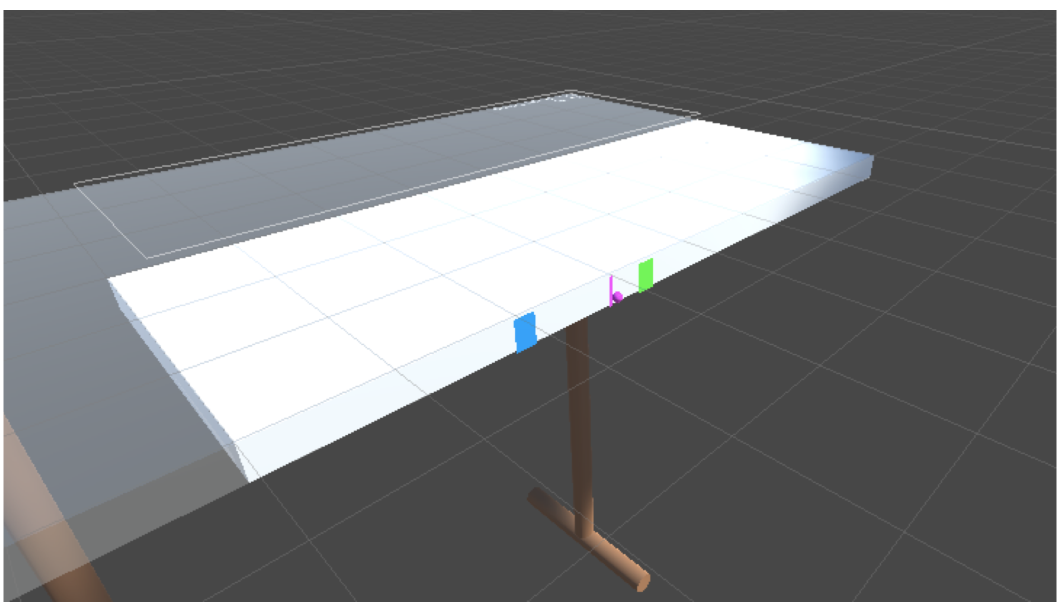


Figure 4.3: A virtual table and marker are used to track touches on a real, physical table.

Chapter 5

Experiment 1 - Index Finger Input

The goal of this experiment is to measure the speed and accuracy of one-dimensional, single-touch tapping, dragging, and crossing using the two ridges and face of a table edge. The top and bottom faces are included to make baseline comparisons. The results are used to calculate throughput and establish minimum target widths using effective width analysis.

5.1 Participants and Apparatus

We recruited 15 participants, ages 23 to 39 (M = 27, SD = 4.13), 5 identified as female, 10 as male, and 1 was left-handed. All reported previous experience using touchscreen devices. Remuneration was $15.

Participants sat in a standard office chair facing the table system (described in Chapter 4), with their body centered on the interactive area. The tracking marker was attached to the index finger using hypoallergenic tape (Figure 4.1). The touch offset calibration method was performed for each participant on all tested surfaces. To reduce friction caused by humidity, participants applied talcum powder to their finger as needed.

5.2 Tasks

Three tasks represent standard interactions (Figure 5.1):

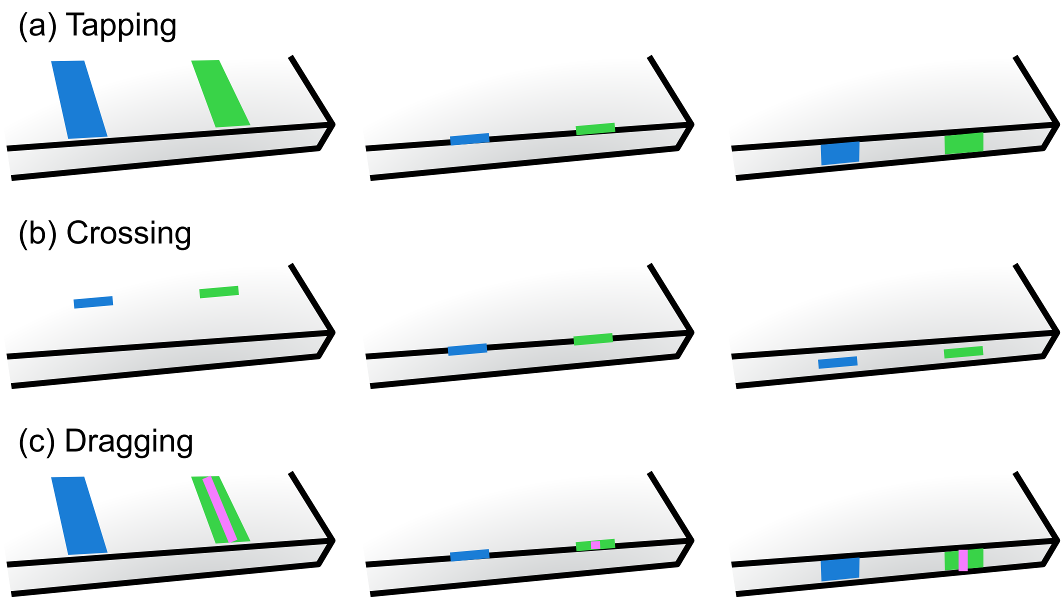


Figure 5.1: Targets for the top face, top ridge, and front face.

Tapping — The participant touched and released their finger within the bounds of a target.

Crossing — The participant touched above or below a horizontal line, moved their finger through the line, and released once they had passed through. *Discrete* crossing was used [29, 1, 2] as it is most similar to tap and drag. We also test crossing through vertical targets on all compatible surfaces (top, front, and bottom faces), which we discuss in a separate analysis.

Dragging — The participant acquired a pink line at the centre of a target by touching down within the target's bounds. The line was dragged by maintaining contact with the table until a docking target was reached and the finger lifted. The line snapped to the centre of the docking target.

The current target was green, while the other was blue. A 'click' or 'beep' sound signaled if the task was successful, or not. Once the finger was lifted after a trial, the target colours changed, and the next trial continued. As recommended [5, 42, 46], a small red cursor, projected on the top face, tracks the finger's movements on the bottom face (Figure 5.2). Its diameter is 3 mm when the finger contacts the table, but becomes larger and more transparent as the finger is lifted.

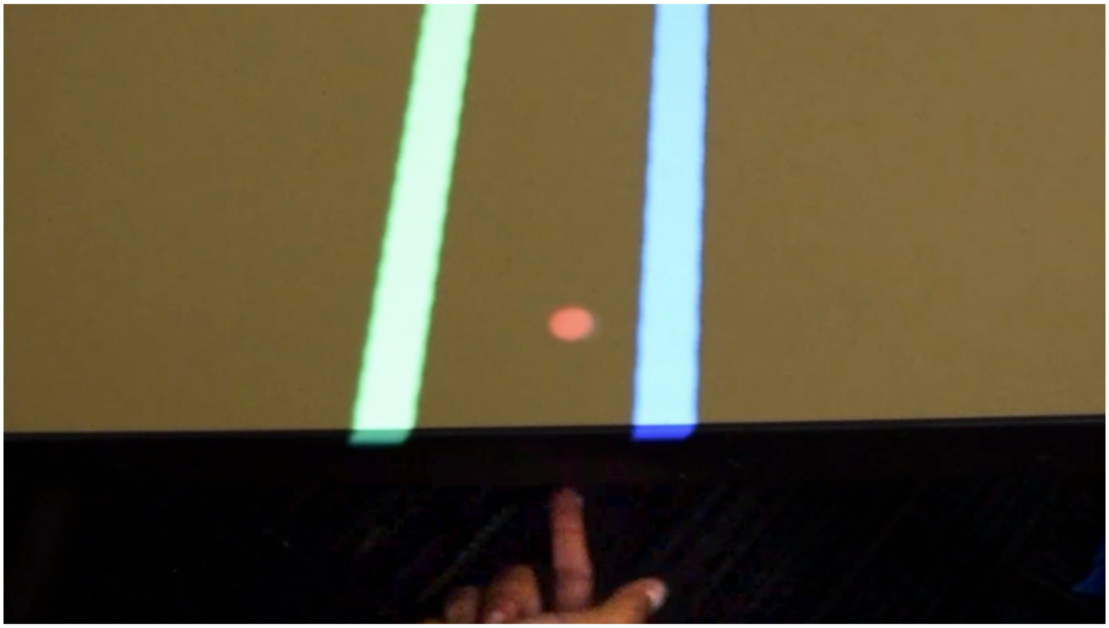


Figure 5.2: The cursor the participant saw as they interacted with the bottom face.

For each task, the target width and distance was experimentally controlled. The participant completed a set of 5 reciprocal task trials (back-and-forth between two targets), using the same width and distance. We use one-dimensional, serial tasks since the top ridge, front face, and bottom ridge are not large enough to support two-dimensional tasks.

'False' errors could occur if marker tracking is momentarily lost. To mitigate this, we define a 2 cm region on both sides of the target, and consider any errors detected outside this range to be caused by a motion tracking error. An error sound is played, and the participant must restart the trial. Errors inside the 2 cm region around the target are counted as 'true' errors and are used for the error rate metric. Note 2 cm is the width of the largest target we test.

5.3 Procedure

Each participant used all surfaces for each task. Before each combination of a specific task and surface, they completed a practice block containing a set of the easiest and most difficult combinations of target distance and width.

After, they completed five sets of trials, each set a single combination of target distance

ID	W (mm)	D (mm)
2.58	20	100
3.46	18	175
4.14	15	250
4.75	13	325
5.36	10	400

Table 5.1: The Fitts' law parameters used

and width. This was done thrice to create three blocks of sets for a surface, and the process repeated for all surfaces. After each set, the participant pressed a large 'continue' button rendered 15 cm from the back of the top face. This provided a place to take a break.

The task changed once all surfaces were used for the current task. At this time, the participant ranked the five surfaces for ease of use, ease of learning, comfort, and overall preference. The study lasted approximately 1 hour.

5.4 Design

This is a within-subjects design with two primary independent variables: TASK with 3 levels (TAP, CROSS, DRAG); and SURFACE with 5 levels (T-FACE, T-RIDGE, F-FACE, B-RIDGE, B-FACE). BLOCK and ID are secondary independent variables, where ID is the index of difficulty described in Fitts' law [36].

There were 5 ID variations for each task (Table 5.1), and every ID was calculated using a target width (W) and distance (D). Each ID variation was presented in random order. We selected these W and D values as ID should be between 2 and 8 [36], W should be large enough to see and touch, and D should not exceed the length of the table.

The order for TASK was counter-balanced using a 3 × 3 Latin square. For crossing, the vertical and horizontal orientations occurred one after another using a random ordering.

The primary measures are *Selection Time* and *Error Rate*. *Selection Time* is the time from the previous target selection until the current target selection. *Error Rate* is the proportion of trials that had one or more task errors. There are also 12 subjective rankings. Excluding the extra horizontal crossing condition, there are:

$$\begin{array}{rl} & 3 \text{ TASKS} \\ \times & 5 \text{ SURFACES} \\ \times & 3 \text{ BLOCKS} \\ \times & 5 \text{ IDS} \\ \times & 5 \text{ repetitions} \\ \hline \end{array}$$

1125 data points per participant.

5.5 Results

For every combination of TASK, SURFACE, and BLOCK, trial times more than 3 standard deviations from the mean were excluded as outliers. An additional 8 trials were excluded due to tracking errors. Overall, 218 trials (1.3%) were removed.

An implementation bug caused some crossing trials to be falsely categorized as errors. During the experiment, target intersections were tested using a series of points. This is correct for detecting a point inside a tapping or dragging target, but fast movements sometimes passed through thin crossing targets without detection. To correct this, trials are retroactively marked successful if any line segments in the series of contact points intersect a crossing target. This corrected 1903 false error trials (26% of crossing trials). Participants heard a warning sound when a valid cross was falsely detected as an error, which may have influenced task preference rankings. Because we are primarily focused on comparing surfaces, this is an acceptable limitation.

Analysis Methods

In the analysis, a SURFACE × TASK × BLOCK ANOVA with Tukey HSD post hoc tests was used, unless noted. When sphericity was violated, degrees of freedom were corrected using Greenhouse-Geisser ($\epsilon < 0.75$) or Huynh-Feldt ($\epsilon \geq 0.75$). As residuals for *Selection Time* were not normally distributed, log transformed values were used. For readability, we use short surface names in figures and tables: TF = T-FACE, TR = T-RIDGE, FF = F-FACE, BR = B-RIDGE, and BF = B-FACE. Error bars in all graphs are 95% confidence.

Learning Effect

We are interested in practised performance, so we examine if earlier blocks took longer and should be removed (as recommended for Fitts' studies [36]). There is a small, but significant effect of BLOCK on *Selection Time* ($F_{2,28} = 42.28$, $p < .001$, $\eta^2_G = .02$), but no interactions involving BLOCK. Post hoc tests found BLOCK 3 faster than BLOCK 1 ($p < .001$) and BLOCK 2 ($p < .01$). The difference between BLOCK 2 and 3 is small, only 30ms (2.5%).

We noticed participants switching to optimal postures or movement strategies as the experiment progressed. Given this, and the significant difference, we use block 3 data in all subsequent analysis to be more representative of practised performance. Analysis using more blocks yields similar results, and any differences are noted in our discussion.

Selection Time

Considering all tasks together, using the top ridge is fastest. A significant main effect of SURFACE on *Selection Time* ($F_{4,56} = 34.56$, $p < .001$, $\eta^2_G = .27$) with post hoc tests reveals differences between all surfaces (all $p < .05$): T-RIDGE (990ms) is fastest, followed by T-FACE (1039ms), F-FACE (1137ms), B-RIDGE (1172ms), and B-FACE (1311ms).

When considering specific tasks, the top ridge is always one of the fastest surfaces. There is a significant interaction involving TASK and SURFACE on *Selection Time* ($F_{8,112} = 8.32$, $p < .001$, $\eta^2_G = .12$). Post hoc tests examine differences between each SURFACE, for every TASK (Figure 5.3):

- For TAP, there is no significant difference between T-FACE and T-RIDGE. Other differences are significant (all $p < .001$). T-FACE (692ms) and T-RIDGE (710ms) are faster than F-FACE (768ms), B-RIDGE (958ms), and B-FACE (1075ms).

- For CROSS, there is no significant difference between T-FACE and F-FACE, but other differences are significant (all $p < .01$). T-RIDGE (821ms) is fastest (19% faster than T-FACE), followed by B-RIDGE (902ms) (11% faster than T-FACE), F-FACE (981ms) and T-FACE (1013ms), and B-FACE (1150ms).

- For DRAG, there is no significant difference between B-FACE, F-FACE, and B-RIDGE, or T-FACE and T-RIDGE, but all other differences are significant (all $p < .001$). T-FACE (1409ms) and T-RIDGE (1434ms) are fastest and B-RIDGE (1658ms), F-FACE (1685ms) and B-FACE (1708ms) are slowest.

In addition, post hoc tests between TASK, for every SURFACE find tapping is always the fastest task, except for the bottom ridge, where crossing is fastest. For all surfaces, there is a significant difference between all tasks (all $p < .001$), except between TAP and CROSS for B-RIDGE which is borderline ($p = .04$). TAP is the fastest across surfaces, except for B-RIDGE, where CROSS (902ms) is faster than TAP (958ms).

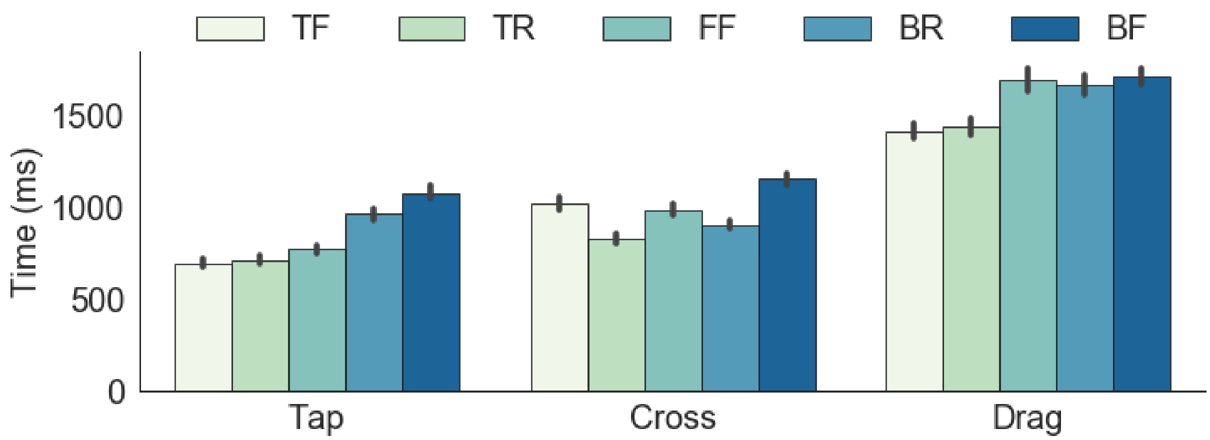


Figure 5.3: Selection Time (ms) by TASK for each SURFACE

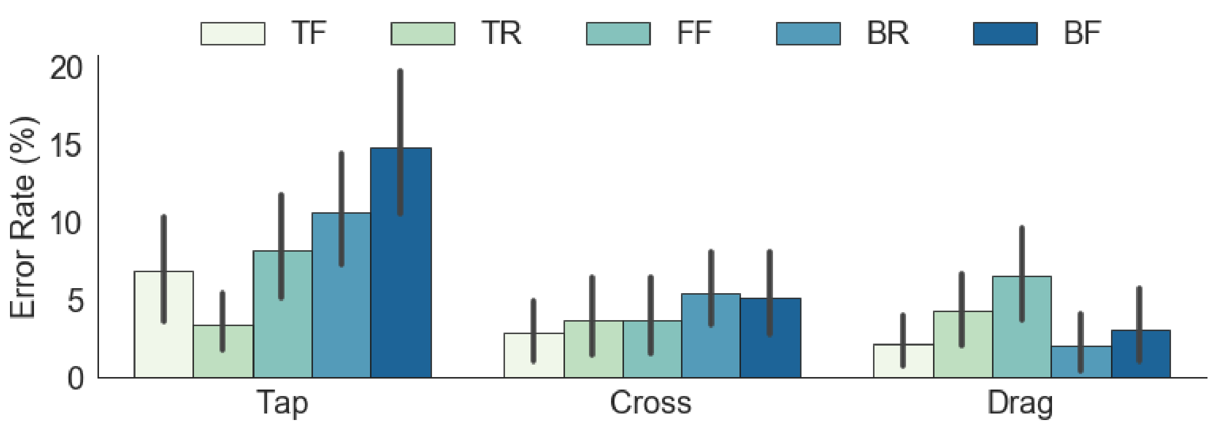


Figure 5.4: Error Rate (%) by TASK for each SURFACE

Error Rate

Considering all tasks, top face and top ridge are less error prone than the bottom face. A significant main effect of SURFACE on *Error Rate* ($F_{4,56} = 5.16$, $p < .01$, $\eta^2_G = .07$) with post hoc tests reveals significant differences between T-RIDGE and B-FACE, and T-FACE and B-FACE (both $p < .05$). Overall, T-FACE and T-RIDGE, F-FACE and B-RIDGE, and B-FACE have average error rates of 4%, 6%, and 8%, respectively.

Surface does not affect error rates when crossing or dragging (Figure 5.4). There is a significant interaction between TASK and SURFACE ($F_{8,112} = 4.21$, $p < .001$, $\eta^2_G = .12$) on

Error Rate. Post hoc tests reveal differences between each TASK, across every SURFACE. For TAP, there are differences between T-RIDGE (3%) and B-RIDGE (11%) ($p < 0.05$); T-RIDGE and B-FACE (15%) ($p < .001$); and T-FACE (7%) and B-FACE ($p < .05$). For CROSS and DRAG, there are no significant differences.

Error rate is consistent across tasks the for top ridge and front face. Post hoc tests reveal differences in *Error Rate* between TAP and CROSS for T-FACE, B-RIDGE (both $p < 0.05$), and B-FACE ($p < .001$). There is a difference between TAP and DRAG for T-FACE ($p < .05$), B-RIDGE and B-FACE (both $p < .001$).

Preference Scores

After completing a single TASK, participants ranked surfaces from best to worst for ease of use, ease of learning, comfort, and overall preference. Ties are allowed. We assign a Condorcet rank [50] to each condition. A condition ranked first is the condition that defeats all others in pairwise comparisons (Condorcet winner), while the condition ranked second defeats all others save the rank one condition, and so on. A Condorcet winner or strict ordering may not exist. If an ordering exists, each condition strictly dominates the others in preference. As strict ordering is not important, we allow for equal Condorcet rankings. Table 5.2 presents all Condorcet rankings. The top face and top ridge are consistently highly ranked across all measures whereas the other surfaces are ranked lower. The bottom ridge is only highly ranked (1 or 2) for crossing.

Crossing Orientation

The main study focused on the horizontal target orientation (vertical crossing), but we also gathered data for vertical crossing targets (horizontal crossing) with the compatible top, front, and bottom faces. For this analysis we define a 3-level factor SURFACE∗ for the three surfaces, and a 2-level factor CROSSTYPE∗, defined by crossing *type* rather than target orientation: VERT and HORZ.

Overall, vertical crossing is significantly faster than horizontal crossing. A significant main effect of CROSSTYPE∗ on *Selection Time* with post hoc tests reveals VERT (1048ms) is significantly faster than HORZ (1151ms), which differs from Luo and Vogel's previous results [29]. This may be due to differences in surface orientation or texture.

Overall, front face is fastest. A main effect of SURFACE∗ on *Selection Time* ($F_{2,28} = 6.68$, $p < .01$, $\eta^2_G = .07$) and post hoc tests show differences between all surfaces (all $p < .05$).

	Ease of Use			Ease of Learning			Comfort			Overall		
	Tap	*Cross*	*Drag*	*Tap*	*Cross*	*Drag*	*Tap*	*Cross*	*Drag*	*Tap*	*Cross*	*Drag*
TF	1	2	1	1	3	1	1	2	1	1	4	2
TR	2	1	2	2	1	1	2	1	2	2	1	1
FF	3	3	3	3	2	2	4	4	4	4	3	4
BR	4	2	4	4	1	4	4	2	5	5	2	5
BF	5	4	3	5	4	3	3	3	3	3	5	3

Table 5.2: Condorcet preference rankings for Experiment 1

F-FACE (1037ms) is faster than T-FACE (1087ms) and B-FACE (1173ms). There are no interactions.

Homing Time

Our experiment was not designed to measure the time spent moving from a primary input device, like a keyboard, to an edge surface. However, we can estimate this *Homing Time* measure using the time span between pressing the 'continue' button (rendered 15 cm from the back of the top face) and the event starting (or ending) each set of trials.

After removing 212 outliers (2.5%), we find top face homing time is only 150ms less than top ridge or front face. A significant main effect of SURFACE ($F_{4,56} = 35.34$, $p < .001$, $\eta^2_G = .20$) with post hoc tests shows T-FACE is fastest (1176ms), followed by T-RIDGE (1327ms) and F-FACE (1325ms), and B-RIDGE (1475ms) and B-FACE (1442ms) (all $p < .001$).

These times are slower than the keyboard-to-mouse homing factor, H (400ms), in the keystroke-level model [9]. This is likely due to a reaction time when our participants located the first task target in a set, or realize the set ended.

Effective Width and Fitts' Modelling

The distance between the centre of the target and the participant's contact point is recorded for every trial, and we calculate the *effective target width* by multiplying the standard deviation of these distances by 4.133 (Equation 2.2). To make recommendations for minimum target width [36], we examine the average effective widths using the smallest target size (10 mm) for every task and surface. Results (Table 5.3) show crossing and dragging targets on the front face have the smallest (7.4 mm) and largest (19.2 mm) minimum target widths, respectively.

	Tap	**Cross**	**Drag**
TF	12.5	10.0	12.8
TR	10.9	8.3	15.1
FF	16.6	7.4	19.2
BR	15.9	10.4	15.2
BF	14.9	8.3	13.4

Table 5.3: Minimum target width (mm) sizes for Experiment 1

	Tap			**Cross**			**Drag**		
	a	b	R^2	a	b	R^2	a	b	R^2
TF	76	159	0.95	444	133	0.97	322	278	0.98
TR	145	140	0.99	322	115	0.96	-75	421	0.94
FF	122	169	0.91	384	136	0.98	235	423	0.92
BR	14	245	0.98	372	129	0.98	388	332	0.95
BF	262	231	1.00	507	155	1.00	414	328	0.98

Table 5.4: Fitts' model parameters for Experiment 1

Given the high tapping error rates, we use ID_e, the effective index of difficulty [36], to model average speed using Fitts' law. Fitts' law has strong predictive power for all table edge surfaces: all 15 combinations of TASK and SURFACE are modelled well, with all fitness values above 90% (Table 5.4). All models except T-RIDGE dragging (-75) have positive intercepts. T-FACE and B-RIDGE have the lowest TAP intercepts. For crossing, the intercepts for all edge surfaces are less than those for the top and bottom faces. T-RIDGE (-75) and F-FACE (235) have the lowest intercepts for DRAG.

The inverse of b gives average throughput [36]. For every surface, CROSS and DRAG have the highest and lowest throughput, respectively (Table 5.5). T-RIDGE produces the highest throughput for TAP (7.15 bits/s) and CROSS (8.68 bits/s), and T-FACE gives higher throughput for DRAG (3.60 bits/s).

5.6 Discussion

While the top ridge is consistently fast with low error rates and high subjective ratings, the bottom ridge is only fast and preferred for crossing. All three edge surfaces have crossing

	Tap	Cross	Drag
TF	6.30	7.54	3.60
TR	7.15	8.68	2.37
FF	5.92	7.35	2.37
BR	4.09	7.75	3.01
BF	4.34	6.45	3.05

Table 5.5: Throughput (bits/s) for Experiment 1

error rates below 6%. Crossing minimum target widths are smaller than those for tapping and dragging across all edge surfaces. Given these results, the top ridge may be the best surface for interaction techniques, and crossing tasks may generally be better suited for edge-of-table interactions.

Learning Effect

There are many factors that could have influenced learning. The crossing and dragging tasks may have been more difficult for participants to understand, since they required participants to touch the table, move their finger while maintaining contact, and release once they had passed through or reached the target. As participants were not told to use a particular hand posture or position, it is possible they took time to explore optimal strategies for each task.

Recall our analysis only uses block 3 due to a significant learning effect. Using more blocks for analysis reveals more details about the learning effect. Analysis with blocks 2 and 3 yields the same results with one exception: for TAP, there is a 37ms difference in *Selection Time* between T-FACE (699ms) and T-RIDGE (736ms) ($p < .01$). Using all three blocks yields one more minor difference: for DRAG, T-FACE (1433ms) is 70ms faster than T-RIDGE (1503ms) ($p < .05$). This suggests participants were learning more efficient methods of tapping and dragging with T-RIDGE during the first two blocks.

Preferences and Strategies for Touch Input

It is possible participants took time to explore optimal strategies for each task. For example, some participants positioned their fingers horizontally (P4, P5) and others rotated their chairs (P9) to interact with top ridge targets from the side. Some participants (P9, P12) crossed through top ridge targets by carefully maintaining contact with the top and front faces. Others ‘flicked’ through the top ridge upward (P3). With practice, most participants ‘flicked’ top ridge crossing targets in a downward motion in block 3, suggesting this is a preferred crossing strategy for index finger input.

The front and bottom faces are never preferred. Some participants (P2, P5, P6, P9) felt the front face was placed in a strange position, as lifting required them to move their entire arm rather than just their finger or wrist. Others (P2, P4, P9, P14) felt the bottom ridge and face were awkward due to poor hand or body positioning. P2 and P5 felt it was strange to lower their fingers after making contact with the table as they associated ‘lifting’ with an upward movement. P2 and P6 liked positioning their hands palm-up.

The bottom face was the slowest for all tasks, and more error prone than the top face for tapping tasks. The difference in tapping error rate reinforces Wigdor et al.’s results [43]. For tap and cross, all three edge surfaces are faster than the bottom face. For drag, the top ridge is the only edge surface that outperforms the bottom face. While some participants (P6, P8, P9) enjoyed seeing a cursor follow their finger movements, P2 and P4 felt it was disorienting.

Participants tried several hand postures and techniques for each task, so it seems there is an opportunity for different postures and perhaps using different fingers [22]. As the top and bottom ridges had the highest rankings, fastest times, and low errors ($< 6\%$) for crossing, we explore crossing variations using the thumb along these two surfaces in a follow-up study. We include tapping to perform baseline comparisons. Participants’ feedback on front face interaction was valuable. We try to improve front face interaction in the same follow-up study by introducing a “braced” hand posture [12] with less arm movement and occlusion.

Chapter 6

Experiment 2 - Using a Braced Thumb

We try to improve front face interaction using a "braced" touch posture, and explore how this posture affects task performance along the top and bottom ridges.

6.1 Participants and Procedure

We recruited 8 participants, ages 22 to 42 (M = 28, SD = 4.13), 2 identified as female, 6 as male, and 1 was left-handed. All reported experience using touchscreen devices. Remuneration was $5. The same prototyping system as Experiment 1 was used.

The tasks, tapping and crossing, were the same as those used in Experiment 1. However, instead of using the index finger, the participant used their thumb. We explore thumb input since it was suggested by Cockburn et al. [12, 13] and data on minimum target width for thumb input would give more options for designers. As a variation on Cockburn et al.'s "braced" posture, other fingers were placed underneath the table to provide additional stability and minimize occlusion (Figure 1.2). Note our results from Experiment 1 suggest bracing from below is optimal, since the bottom face is less likely to be used for touch input due to its poor performance. The experimental procedure was identical to that of Experiment 1, except only two tasks were performed using three edge surfaces. The study lasted 20 minutes.

6.2 Design

This is a within-subjects design with two primary independent variables: TASK with 2 levels (TAP, CROSS) and SURFACE with 3 levels (T-RIDGE, F-FACE, B-RIDGE). BLOCK and ID are secondary independent variables, with 3 and 5 levels respectively. All target widths and distances are the same as those used in Experiment 1 (Table 5.1). Each ID variation was presented in random order. There were 3 blocks per combination of TASK and SURFACE, and the order for TASK was counter-balanced.

Primary measures include *Selection Time* and *Error Rate*. *Selection Time* is the time from the previous target selection until the current target selection, and *Error Rate* is the proportion of trials with task errors. There are also 8 subjective rankings. In total:

$$
\begin{array}{rl}
 & 2 \text{ TASKS} \\
\times & 3 \text{ SURFACES} \\
\times & 3 \text{ BLOCKS} \\
\times & 5 \text{ IDS} \\
\times & 5 \text{ repetitions} \\
\hline
\end{array}
$$

450 data points per participant.

6.3 Results

For every combination of TASK, SURFACE, and BLOCK, trial times more than 3 standard deviations from the mean were excluded as outliers. In total, 62 trials (1.9%) were removed. As was done in the previous study, we retroactively mark any crossing trial as successful if any line segments formed by the series of generated contact points intersects with the targets. A total of 310 crossing trials (19%) were corrected.

Learning Effect

There is a significant main effect of BLOCK on *Selection Time* ($F_{2,16} = 21.35$, $p < .001$, $\eta^2_G = .04$), but no interaction effects. Post hoc tests found BLOCK 1 and BLOCK 2 are significantly slower than BLOCK 3 (both $p < .01$), though the difference between blocks 2 and 3 is small (42ms). We use block 3 data for all subsequent analysis.

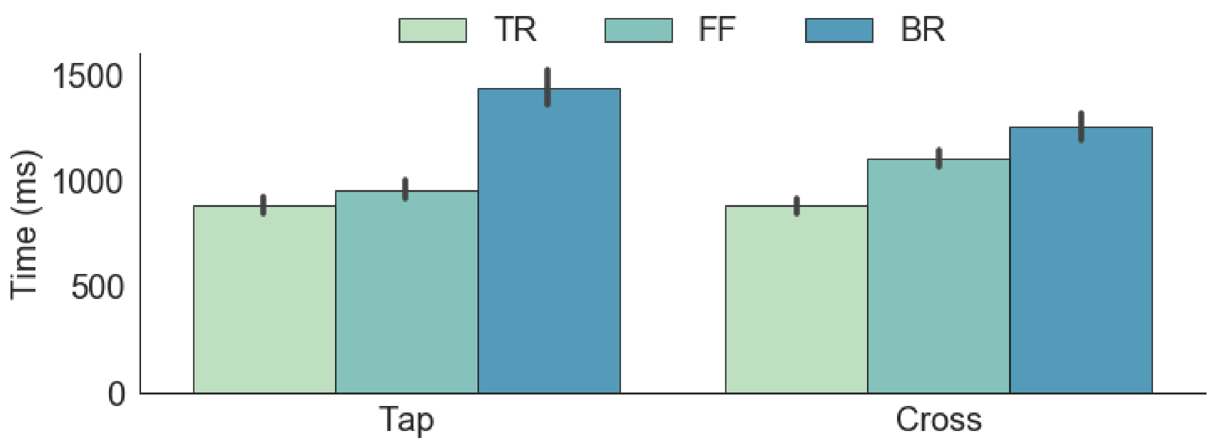


Figure 6.1: Selection Time (ms) by TASK for each SURFACE for Experiment 2

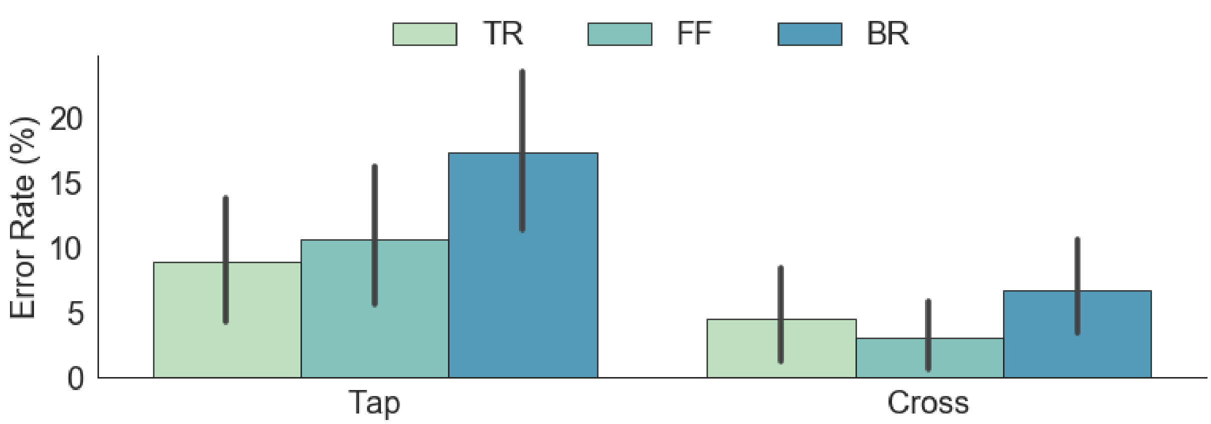


Figure 6.2: Error Rate (%) by TASK for each SURFACE for Experiment 2

	Tap	Cross
TR	10.1	7.4
FF	14.5	13.0
BR	15.5	13.6

Table 6.1: Minimum target width (mm) sizes for Experiment 2

Selection Time

Considering all tasks, top ridge is faster than the front face and bottom ridge. There is a main effect of SURFACE on *Selection Time* ($F_{2,16} = 43.16$, $p < .001$, $\eta^2_G = .43$), with post hoc tests showing significant differences between all three surfaces (all $p < .001$). Overall, T-RIDGE (879ms) is faster than F-FACE (1029ms) and B-RIDGE (1344ms). There are no significant interactions involving *Selection Time* (Figure 6.1).

Error Rate

Considering all surfaces, crossing is less error prone than tapping. There is a significant main effect of TASK on *Error Rate* ($F_{1,8} = 5.66$, $p < .05$, $\eta^2_G = .19$), where CROSS (5%) is less error prone than TAP (12%). There are no other significant main effects nor any significant interactions (Figure 6.2).

Preference Scores

After completing a single TASK, participants ranked each surface from best to worst using the same criteria as Experiment 1. We assign a Condorcet ranking to each condition. For both tasks, the rankings (T-RIDGE, F-FACE, B-RIDGE) are mostly identical across all four measures, with the exception of ease of learning for CROSS, where F-FACE and T-RIDGE are both ranked first.

Effective Width and Fitts' Modelling

Results (Table 6.1) show *effective target widths* for crossing are smaller than those for tapping. T-RIDGE crossing targets and B-RIDGE tapping targets have the smallest (7.4 mm) and largest (15.5 mm) minimum target widths, respectively.

	Tap			Cross		
	a	b	R^2	a	b	R^2
TR	-115	260	0.95	124	175	0.96
FF	123	224	0.93	421	167	0.95
BR	733	204	0.83	419	212	0.86

Table 6.2: Fitts' model parameters for Experiment 2

	Tap	Cross
TR	3.84	5.73
FF	4.47	5.98
BR	4.90	4.71

Table 6.3: Throughput (bits/s) for Experiment 2

ID_e has strong predictive power for the top ridge and front face, where $R^2 > 0.9$ (Table 6.2), but is not as strong for the bottom ridge ($R^2 > 0.8$). T-RIDGE has a negative intercept for TAP (-115), and the lowest intercept for CROSS (124). TAP has lower intercepts than CROSS for T-RIDGE and F-FACE, but the intercept is lower for CROSS with B-RIDGE.

B-RIDGE and F-FACE have the highest throughputs for TAP (4.90 bits/s) and CROSS (5.98 bits/s), respectively. For T-RIDGE and F-FACE, throughput is higher for CROSS than TAP (Table 6.3).

Comparing Thumb to Index Finger Input

In Experiment 1, participants used their index finger to interact with targets. In Experiment 2, participants used their thumb and rested their other fingers underneath the table to stabilize arm movements [12] and minimize occlusion. To compare postures, we define a 3-level factor SURFACE∗ for the three edge surfaces (T-RIDGE, F-FACE, B-RIDGE), a 2-level factor TASK∗ (TAP, CROSS), and a new, 2-level factor POSTURE (POINTING, BRACED). We use the same analysis methods as before, however, we use a mixed-factorial ANOVA where POSTURE is a between-subjects factor and TASK and SURFACE are repeated measures.

Index finger input is always faster than braced touch, with the difference as low as 77ms for crossing on the top ridge to 478ms for tapping on the bottom ridge. A significant main effect of POSTURE on *Selection Time* ($F_{1,22} = 13.67$, $p < .001$, $\eta^2_G = .22$) with post hoc tests shows POINTING (857ms) is faster than BRACED (1084ms). There is also an interaction between POSTURE and SURFACE ($F_{2,44} = 5.66$, $p < .001$, $\eta^2_G = .05$). Post hoc tests show POINTING is always faster than BRACED (all $p < .001$). For TAP, BRACED is 168ms (24%), 186ms (24%), and 478ms (50%) slower than POINTING for T-RIDGE, F-FACE, and B-RIDGE, respectively. For CROSS, it is 77ms (9%), 124ms (13%), and 354ms (40%) slower than POINTING for T-RIDGE, F-FACE, and B-RIDGE, respectively. This is similar to Cockburn et al.'s results for the "static" condition [12].

Index finger input is less error prone than braced touch. Considering all tasks and surfaces, a significant main effect of POSTURE on *Error Rate* ($F_{1,22} = 6.65$, $p < .05$, $\eta^2_G = .04$) shows POINTING (6%) is less error prone than BRACED (8%). There are no significant interactions involving POSTURE. For TAP, BRACED is 7%, 2%, and 5% more error prone than POINTING for T-RIDGE, F-FACE, and B-RIDGE, respectively. For CROSS, it is 1% more error prone than POINTING for all edge surfaces.

Minimum target widths for tapping are always smaller using the thumb. For the front face, the difference in target width is over 2 mm. Crossing widths are smaller with index finger interaction, with the exception of crossing on the top ridge. Index finger input yields higher tapping throughput for the front face and top ridge. Tapping on the bottom ridge yields higher throughput with thumb input. Crossing throughput is always better using the index finger.

6.4 Discussion

For both tasks, top ridge and bottom ridge are the fastest and slowest surfaces, respectively. Cross is less error prone than tap across all surfaces. A pointing posture may be better suited for edge-of-table interactions. While the braced touch posture is slower, the difference in error rate is small for crossing. This posture could still be used for infrequent crossing actions.

Preferences and Strategies for Touch Input

Most participants (P1, P2, P3, P5, P6) agreed the braced posture was awkward for bottom ridge interactions. P1 felt it was difficult to reach targets on her non-dominant-hand side, as she had to cross her arm over her body. Others (P2, P6) re-positioned the chair to reach these targets. Braced touch interactions should be performed along the user's dominant hand-side.

While we introduced a braced posture to stabilize arm movement, some participants (P1, P6) felt the posture was tiring along the front face as well due to awkward wrist placement. Participants may have experienced discomfort from trying to maintain contact with the bottom face using the entire hand. Others may not have adjusted wrist rotation based on target location. Braced interactions may be more comfortable if the wrist-forearm angle is minimized, or if the hand is braced on the top face.

Chapter 7

General Discussion

We rephrase the most relevant results as design recommendations, discuss example applications for table edge interaction, and finally, discuss possible limitations in our methods.

7.1 Design Recommendations

Hand Postures

A pointing posture should be prioritized over a braced hand posture due to its faster speeds and lower error rates. A braced hand posture could be used to trigger infrequent commands that do not require speed. If a braced hand posture is to be used, user interfaces should be placed along the user's dominant-hand side to minimize arm and wrist discomfort. A braced hand posture should only be used along the top ridge and front face for crossing tasks; the bottom ridge performs too poorly to justify its use. Other braced postures (like bracing on the top face) may result in better performance, especially along the bottom ridge.

Tasks and Surfaces

The top ridge is fast with low error rates, and is a good candidate for tapping, crossing, and dragging. Interaction along the top ridge should always be prioritized. The front face can support all three tasks, but due to slower speeds and physical discomfort, should not be used as often. Interactions along the front face should minimize arm and wrist movement.

As top ridge and front face error rates are consistent across tasks, we recommend tapping tasks to be prioritized, as they are faster. Crossing is a comparable alternative. Dragging should be used when speed is not necessary. The bottom ridge should only be used for crossing tasks.

Minimum Target Size

For a braced thumb tap, targets should be at least 10.1 mm wide for top ridge and 14.5 mm wide for front face. Horizontal crossing targets should be 7.4 mm along the top ridge and 13 mm wide on the front face. With a pointing posture, top ridge targets should be at least 10.9 mm, 8.3 mm, and 15.1 mm wide for tap, cross, and drag. Targets placed along the front face should be at least 16.6 mm, 7.4 mm, and 19.2 mm wide for tap, cross, and drag. Bottom ridge crossing targets should be at least 10.4 mm wide.

7.2 Example Applications

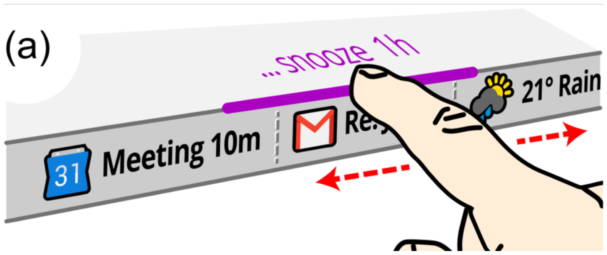
Notification Tray — The front face becomes a notification tray for emails, tweets, weather updates, and meeting reminders. Different edge interactions can: adjust notification settings like “snooze” time; dismiss a notification; access shortcut actions such as “liking” a tweet or sending a “canned” email reply; and viewing notification details on a computer or nearby wall display using SAR (Figure 7.1).

Smart Home Control — The table edge becomes a menu to adjust smart home functions, like heating, lights, security, and media. Available functions appear along the top ridge and crossing through an item selects it. Sliders, toggles, and buttons appear on the front face for users to adjust settings.

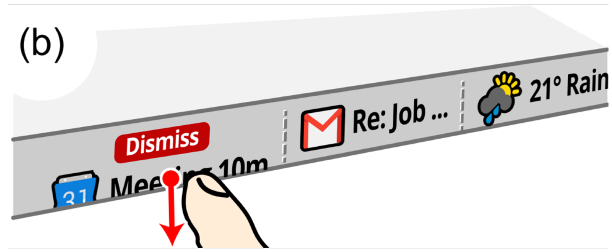
7.3 Limitations

Touch Pitch and Roll

We mimic a touchscreen by using a marker’s location to establish a single point of contact between the table and finger. While this technique works well for the area *underneath* the finger, participants may be tempted to interact with content using the *side* of their finger. This input works with touchscreens, but our system may categorize it as an error. Previous

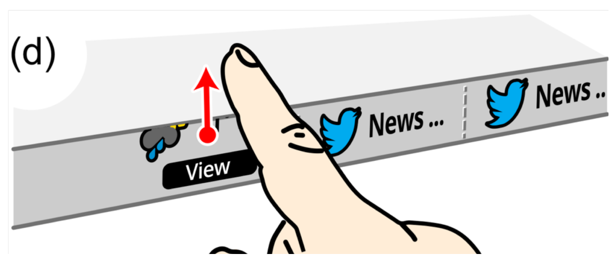


(a) Drag along the top ridge to set a snooze time.



(b) Cross bottom ridge to dismiss a notification.

(c) Braced thumb front face tap reveals shortcuts, like email quick replies. Tap the top ridge to select an action.



(d) Cross top ridge to view notification details on desktop or nearby wall

Figure 7.1: Example edge-of-table notification application.

work [21] shows touch pitch and roll distributions are similar for tap and drag. Given this, and low cross and drag error rates, this limitation is minor.

Occlusion

Many participants from Experiment 1 agreed targets were occluded on the top face (P4, P6, P7, P9, P14, P15), but only two participants (P4, P7) ranked it as the worst or second worst surface for this reason. As occlusion did not seem to affect most participants' rankings or performance, this is a minor issue. Multi-projector SAR systems or tables with built-in capacitive sensing could help reduce this effect.

External Validity

We used reciprocal Fitts' law task trials to measure repeated touch input performance and provide recommendations for minimum target width, a reasonable approach for our initial exploration. While we do not believe edge-of-table input is strictly for single actions interleaved with a primary device, examining homing time in more detail may provide additional insight into performance.

We focused on interactions along a standard, rectangular table edge in this work and note how other physical properties could be exploited in Chapter 3. However, we recognize some of our methods and results may not generalize to other types of tables. For example, it is unclear whether crossing would be fast along a curved or chamfered edge. Furthermore, Fitts' law may not be suitable for modelling speed when interacting with a round table due to the spherical coordinate space of the edge surfaces. Future work could explore identifying the best approaches for modelling touch along other types of surfaces.

Chapter 8

Conclusion

We explored the possibilities of touch input along the edge of table-like surfaces. Results from a one-finger input user study show all edge surfaces could support interaction. The top and bottom ridges are the fastest surfaces for crossing and all edge surfaces have error rates below 6%. Braced touch input has poorer performance. Effective width analysis shows minimum target width is 7.4 mm to 19.2 mm and 7.4 mm to 15.5 mm for the index finger and thumb, respectively.

8.1 Future Work

Our work suggests new topics for future research. Our prototyping system could be improved by using multiple projectors to minimize target occlusion. Capacitive sensors placed on a table could track finger position, creating a "3D surface touchscreen." Smaller prototyping systems, such as those used in [33] could also be used.

Perhaps most exciting, are the many other possibilities for edge-tailored postures and gestures. For example, multitouch input could be explored, such as two-finger crossing on ridges. A subtle variation for the braced thumb posture is resting the fingers on the top face. This may be slightly more comfortable, but likely causes more occlusion. The thumb and middle fingers could also brace using the top and bottom faces simultaneously, with the free index finger used for front face input.

This simultaneous top and bottom brace could be extended into a "grab" posture, where multi-surface thumb and index taps, crosses, and drags work as advanced gestures distinct from single-surface equivalents. The affordance of a table edge naturally suggests this type of posture.

Finally, other table surfaces could be explored, such as corners and legs. Corners, in particular, may be suitable as 'buttons,' but it is unclear how users could take advantage of proprioception to acquire small corner targets when seated at a table. Physical properties of surfaces could be exploited: ridges can be chamfered, rounded, or sharp, and faces can be made in different profiles and textures. Other table-like objects could be studied, like cupboards and window ledges.

Any unused surfaces in the environment could become meaningful interactive displays. Our work extends the scope of touch interaction and presents empirical evidence on the types of edge-of-table interactions people enjoy, and are capable of performing. We believe these results can help designers and researchers create more meaningful interactions that can be implemented and adopted in our everyday lives.

References

[1] Johnny Accot and Shumin Zhai. Beyond fitts' law: Models for trajectory-based hci tasks. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 295–302, New York, NY, USA, 1997. ACM.

[2] Johnny Accot and Shumin Zhai. More than dotting the i's — foundations for crossing-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 73–80, New York, NY, USA, 2002. ACM.

[3] Michelle Annett, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. Medusa: A proximity-aware multi-touch tabletop. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 337–346, New York, NY, USA, 2011. ACM.

[4] Myroslav Bachynskyi, Gregorio Palmas, Antti Oulasvirta, Jürgen Steimle, and Tino Weinkauf. Performance and ergonomics of touch surfaces: A comparative study using biomechanical simulation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1817–1826, New York, NY, USA, 2015. ACM.

[5] Patrick Baudisch and Gerry Chu. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1923–1932, New York, NY, USA, 2009. ACM.

[6] Hrvoje Benko, Ricardo Jota, and Andrew Wilson. Miragetable: Freehand interaction on a projected augmented reality tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 199–208, New York, NY, USA, 2012. ACM.

[7] Hrvoje Benko, Andrew D. Wilson, and Ravin Balakrishnan. Sphere: Multi-touch interactions on a spherical display. In *Proceedings of the 21st Annual ACM Symposium*

on User Interface Software and Technology, UIST '08, pages 77–86, New York, NY, USA, 2008. ACM.

[8] Xiaojun Bi, Tovi Grossman, Justin Matejka, and George Fitzmaurice. Magic desk: Bringing multi-touch surfaces into desktop work. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2511–2520, New York, NY, USA, 2011. ACM.

[9] Stuart K. Card, Thomas P. Moran, and Allen Newell. The keystroke-level model for user performance time with interactive systems. *Commun. ACM*, 23(7):396–410, July 1980.

[10] Kai-Yin Cheng, Rong-Hao Liang, Bing-Yu Chen, Rung-Huei Laing, and Sy-Yen Kuo. icon: Utilizing everyday objects as additional, auxiliary and instant tabletop controllers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1155–1164, New York, NY, USA, 2010. ACM.

[11] A. Cockburn, D. Ahlström, and C. Gutwin. Understanding performance in touch selections: Tap, drag and radial pointing drag with finger, stylus and mouse. *Int. J. Hum.-Comput. Stud.*, 70(3):218–233, March 2012.

[12] A. Cockburn, D. Masson, C. Gutwin, P. Palanque, A. Goguey, M. Yung, C. Gris, and C. Trask. Design and evaluation of braced touch for touchscreen input stabilisation. *International Journal of Human-Computer Studies*, 122:21 – 37, 2019.

[13] Andy Cockburn, Carl Gutwin, Philippe Palanque, Yannick Deleris, Catherine Trask, Ashley Coveney, Marcus Yung, and Karon MacLean. Turbulent touch: Touchscreen input for cockpit flight displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 6742–6753, New York, NY, USA, 2017. ACM.

[14] Daniel Cotting, Markus Gross, and Markus Gross. Interactive environment-aware display bubbles. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 245–254, New York, NY, USA, 2006. ACM.

[15] Paul Dietz and Darren Leigh. Diamondtouch: A multi-user touch technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 219–226, New York, NY, USA, 2001. ACM.

[16] Jerry Alan Fails and Dan Olsen Jr. Light widgets: Interacting in every-day spaces. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, IUI '02, pages 63–69, New York, NY, USA, 2002. ACM.

[17] Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement., 1954.

[18] Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 647–656, New York, NY, USA, 2007. ACM.

[19] Euan Freeman and Stephen Brewster. Messy tabletops: Clearing up the occlusion problem. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 1515–1520, New York, NY, USA, 2013. ACM.

[20] Jon Froehlich, Jacob O. Wobbrock, and Shaun K. Kane. Barrier pointing: Using physical edges to assist target acquisition on mobile device touch screens. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '07, pages 19–26, New York, NY, USA, 2007. ACM.

[21] Alix Goguey, Géry Casiez, Daniel Vogel, and Carl Gutwin. Characterizing finger pitch and roll orientation during atomic touch actions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 589:1–589:12, New York, NY, USA, 2018. ACM.

[22] Alix Goguey, Mathieu Nancel, Géry Casiez, and Daniel Vogel. The performance and preference of different fingers and chords for pointing, dragging, and object transformation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4250–4261, New York, NY, USA, 2016. ACM.

[23] Nan-Wei Gong, Jürgen Steimle, Simon Olberding, Steve Hodges, Nicholas Edward Gillian, Yoshihiro Kawahara, and Joseph A. Paradiso. Printsense: A versatile sensing technique to support multimodal flexible surface interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1407–1410, New York, NY, USA, 2014. ACM.

[24] Tiago Guerreiro, Hugo Nicolau, Joaquim Jorge, and Daniel Gonçalves. Towards accessible touch interfaces. In *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '10, pages 19–26, New York, NY, USA, 2010. ACM.

[25] Hiroshi Ishii, Craig Wisneski, Scott Brave, Andrew Dahley, Matt Gorbet, Brygg Ullmer, and Paul Yarin. ambientroom: Integrating ambient media with architectural space. In *CHI 98 Conference Summary on Human Factors in Computing Systems*, CHI '98, pages 173–174, New York, NY, USA, 1998. ACM.

[26] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. Roomalive: Magical experiences enabled by scalable, adaptive projector-camera units. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 637–644, New York, NY, USA, 2014. ACM.

[27] Shaun K. Kane, Daniel Avrahami, Jacob O. Wobbrock, Beverly Harrison, Adam D. Rea, Matthai Philipose, and Anthony LaMarca. Bonfire: A nomadic system for hybrid laptop-tabletop interaction. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, pages 129–138, New York, NY, USA, 2009. ACM.

[28] Mohammadreza Khalilbeigi, Jürgen Steimle, Jan Riemann, Niloofar Dezfuli, Max Mühlhäuser, and James D. Hollan. Objectop: Occlusion awareness of physical objects on interactive tabletops. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 255–264, New York, NY, USA, 2013. ACM.

[29] Yuexing Luo and Daniel Vogel. Crossing-based selection with direct touch input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2627–2636, New York, NY, USA, 2014. ACM.

[30] Thomas W. Malone. How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst.*, 1(1):99–112, January 1983.

[31] Esben Warming Pedersen and Kasper Hornbæk. An experimental comparison of touch interaction on vertical and horizontal surfaces. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, NordiCHI '12, pages 370–379, New York, NY, USA, 2012. ACM.

[32] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 179–188, New York, NY, USA, 1998. ACM.

[33] Anne Roudaut, Henning Pohl, and Patrick Baudisch. Touch input on curved surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1011–1020, New York, NY, USA, 2011. ACM.

[34] Farzan Sasangohar, I. Scott MacKenzie, and Stacey D. Scott. Evaluation of mouse and touch input for a tabletop display using fitts' reciprocal tapping task. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 53(12):839–843, 2009.

[35] Andrew Sears and Ben Shneiderman. High precision touchscreens: Design strategies and comparisons with a mouse. *Int. J. Man-Mach. Stud.*, 34(4):593–613, April 1991.

[36] R. William Soukoreff and I. Scott MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *International Journal of Human-Computer Studies*, 61(6):751 – 789, 2004. Fitts' law 50 years later: applications and contributions from human-computer interaction.

[37] Aurelien Tabard, Simon Gurn, Andreas Butz, and Jakob Bardram. A case study of object and occlusion management on the elabbench, a mixed physical/digital tabletop. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 251–254, New York, NY, USA, 2013. ACM.

[38] Nicolas Villar, Shahram Izadi, Dan Rosenfeld, Hrvoje Benko, John Helmes, Jonathan Westhues, Steve Hodges, Eyal Ofek, Alex Butler, Xiang Cao, and Billy Chen. Mouse 2.0: Multi-touch meets the mouse. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, pages 33–42, New York, NY, USA, 2009. ACM.

[39] Malte Weiss, Simon Voelker, Christine Sutter, and Jan Borchers. Benddesk: Dragging across the curve. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 1–10, New York, NY, USA, 2010. ACM.

[40] Pierre Wellner. Interacting with paper on the digitaldesk. *Commun. ACM*, 36(7):87–96, July 1993.

[41] Daniel Wigdor, Hrvoje Benko, John Pella, Jarrod Lombardo, and Sarah Williams. Rock & rails: Extending multi-touch interactions with shape gestures to enable precise spatial manipulations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1581–1590, New York, NY, USA, 2011. ACM.

[42] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. Lucid touch: A see-through mobile device. In *Proceedings of the 20th Annual ACM*

Symposium on User Interface Software and Technology, UIST '07, pages 269–278, New York, NY, USA, 2007. ACM.

[43] Daniel Wigdor, Darren Leigh, Clifton Forlines, Samuel Shipman, John Barnwell, Ravin Balakrishnan, and Chia Shen. Under the table interaction. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 259–268, New York, NY, USA, 2006. ACM.

[44] Andrew D. Wilson. Using a depth camera as a touch sensor. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 69–72, New York, NY, USA, 2010. ACM.

[45] Raphael Wimmer, Fabian Hennecke, Florian Schulz, Sebastian Boring, Andreas Butz, and Heinrich Hußmann. Curve: Revisiting the digital desk. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 561–570, New York, NY, USA, 2010. ACM.

[46] Jacob O. Wobbrock, Brad A. Myers, and Htet Htet Aung. The performance of hand postures in front- and back-of-device interaction for mobile computing. *Int. J. Hum.-Comput. Stud.*, 66(12):857–875, December 2008.

[47] Jacob O. Wobbrock, Brad A. Myers, Htet Htet Aung, and Edmund F. LoPresti. Text entry from power wheelchairs: Edgewrite for joysticks and touchpads. *SIGACCESS Access. Comput.*, (77-78):110–117, September 2003.

[48] Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. Edgewrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 61–70, New York, NY, USA, 2003. ACM.

[49] Robert Xiao, Chris Harrison, and Scott E. Hudson. Worldkit: Rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 879–888, New York, NY, USA, 2013. ACM.

[50] H. P. Young. Condorcet's theory of voting. *American Political Science Review*, 82(4):1231–1244, 1988.