Contact-sensing Input Device Manipulation and Recall

by

Lisa Anne Elkin

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, 2018

I hereby declare that I am the sole author of this 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.

Abstract

We study a cuboid tangible pen-like input device similar to Vogel and Casiez's Conté. A conductive 3D-printed Conté device enables touch sensing on a capacitive display, and orientation data from an enclosed inertial measurement unit (IMU) reliably distinguishes all 26 corners, edges, and sides. The device's size is constrained by hardware required for sensing. We evaluate the impact of size form-factor on manipulation times for contact-to-contact transitions. A controlled experiment logs manipulation times performed with three sizes of 3D printed mock-ups of the device. Computer vision techniques reliably distinguish between all 26 possible contacts, and a resistive touch sensor provides accurate timing information. In addition, a transition to touch input is tested, and a mock-up of a digital pen is included as a baseline comparison. Results show larger devices are faster, contact-to-contact transition time increases with distance between contacts, but transitions to barrel edges can be slower than some end-over-end transitions. A comparison with a pen-shaped baseline indicates no loss in transition speed for most equivalent transitions. Based on our results, we discuss ideal device sizes and improvements to the simple extruded-rectangle form-factor. Subsequently, we evaluate learning and recall of commands located on physical landmarks on the exterior of a 3D tangible input device in comparison with a 2D spatial interface. Each of the 26 contacts is a physical spatial landmark on the exterior of Conté. A pilot study compares command learning and recall for Conté with a 2D grid interface, using small and large commands sets. To facilitate novice learning, an on-screen model of Conté replicates the physical device's orientation and displays icons representing commands on the corresponding landmarks. Results show there is likely no difference between 2D and 3D spatial interface recall for a small command set and high recall is possible with large command sets. Applications illustrating possible use cases are discussed as well as possible improvements to the on-screen guide based on our results.

Acknowledgements

I would like to thank my advisor Dan Vogel not only for his incredible guidance and mentorship but also for taking a chance on a student without a computer science background. Dan, I will always be grateful that you believed in me. I'm not crying as I write this, I promise.

Thank you to the HCI lab for being most amazing people I could have asked to spend these three years with. Thank you for tolerating my weirdness and pretending you thought it was funny. You're such a brilliant, supportive, hilarious group of people and I'm so lucky to have gotten to spend this time with all of you.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The mode of an application determines how input is interpreted. For example, a stroke drawn in a drawing application may be green and dashed, or purple and connected. The action of drawing the stroke is the same but the mode determines how it looks. It is common for interfaces to have many modes and the ability to quickly switch between them is an important aspect of usability. In order to facilitate easy mode-switching on a touchscreen device while harnessing the precision of pen input, Vogel and Casiez created Conté: a pen-like tangible input device with 26 different contacts [39]. The 26 different contacts include 8 corners, 12 edges, and 6 sides (Figure 1.1). Vogel and Casiez's prototype did not realize the full potential of their design and some important questions were left unanswered. Their prototype contains internal infrared LEDs and a battery and is used on a diffuse illumination table. Diffuse illumination tables are not common outside the research community, and their system could only sense 10 of the possible 26 distinct contacts. Furthermore, the authors claim it would be easy and fast to switch application modes by changing which corner, edge, or side of the device is in contact with a touchscreen. But this claim was never tested, and it is unknown how fast these kinds of contact-to-contact transitions are in practice. Finally, it is not known if users can learn all 26 command locations, or even a small subset of them, and an effective teaching method was not proposed.

We create a conductive Conté device that can be used on any capacitive display and classifies data from an internal inertial measurement unit (IMU) to uniquely distinguish each of the 26 contacts from each other. We study contact-to-contact transition times and specifically the impact of size constraints, posed by hardware, on contact-to-contact transition times. Through its ability to uniquely classify each of the 26 contacts, our conductive prototype enables us to study recall. Using Conté as an example of a three-dimensional spatial rehearsal-based interface, we study learning and recall in such interfaces in comparison to their two-dimensional counterpart.

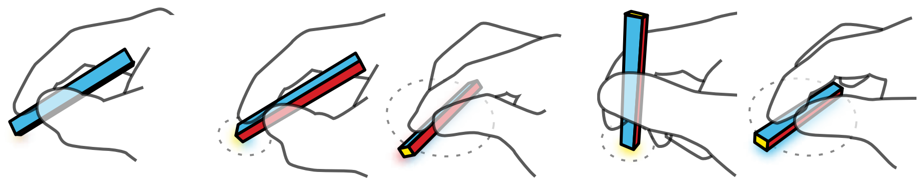


Figure 1.1: Conté contact types: corners, edges, and sides.

Previous research of pen-like manipulation has focused on validating sensing methods and interaction techniques [8, 9, 31]. One study specifically investigated the time to use a pen eraser as a mode-switch [20]. A study of pen mode-switching time is related, but different than our focus on manipulation time for a tangible pen-like device. Physical manipulation of Conté is one aspect of mode-switching. Before transitioning from one contact to another to execute the desired command, the user must first remember where the command is located. Spatial interfaces leverage physical landmarks to improve command-location learning and recall [6]. Large command sets have been learned successfully in rehearsal-based interfaces [40], and learning of rehearsal-based spatial interfaces has been successful in the two-dimensional case [5], particularly with imposed time constraints [18].

We built a conductive prototype of Conté containing a wifi-enabled microcontroller, 9-axis IMU, and lithium ion polymer (LiPo) battery. IMU data is streamed over wifi and classified using a decision tree to uniquely distinguish each of the 26 different contacts. The conductive case was 3D printed on a consumer-level 3D printer, and its corners were engineered to compress and contact the screen with a large surface area to enable recognition on a capacitive display without access to its underlying capacitive image.

We conducted an experiment to measure manipulation time when transitioning between different contact points. We created three different Conté form-factor mock-ups: *small* is the same size as a real artist's Conté crayon; *medium* is the size of Vogel and Casiez's Conté; and *large* is the size of our capacitive prototype. As a baseline, we also created a mock-up of a standard digital stylus. Each mock-up is constructed so the type of contact can be reliably sensed using a hue-based computer vision algorithm, and accurately timed using a resistive touch overlay.

Our results show larger devices are faster overall, contact-to-contact transition time increases with distance between contacts, but transitions to barrel edges can be slower than some end-over-end transitions. We also found Conté transitions can be just as fast as equivalent ones with a pen. These results validate the original Conté idea, and can guide designers when choosing contact-to-command mappings, and hardware designers choosing device form factors.

To study learning and recall, we implement an on-screen visual guide to facilitate novice use

(Figure 1.2).

In Gutwin et al.'s Shape Slicer game [5], a square with an icon representing a command falls from the top of the screen, and the user's goal is to execute the command before the square reached the bottom of the screen. We augment Shape Slicer to be compatible with Conté and implement a 2D spatial interface inspired by FastTap [6] for baseline comparison.

We conduct a between-subjects experiment to measure adoption of expert selection, selection errors, selection time, and recall, using two selection techniques: *Conté*, and *grid* inspired by *FastTap*, and compare both using two command sets: *large* the full set of 26 commands, and *small* a 9 command subset. This is an initial exploratory study with two participants per condition.

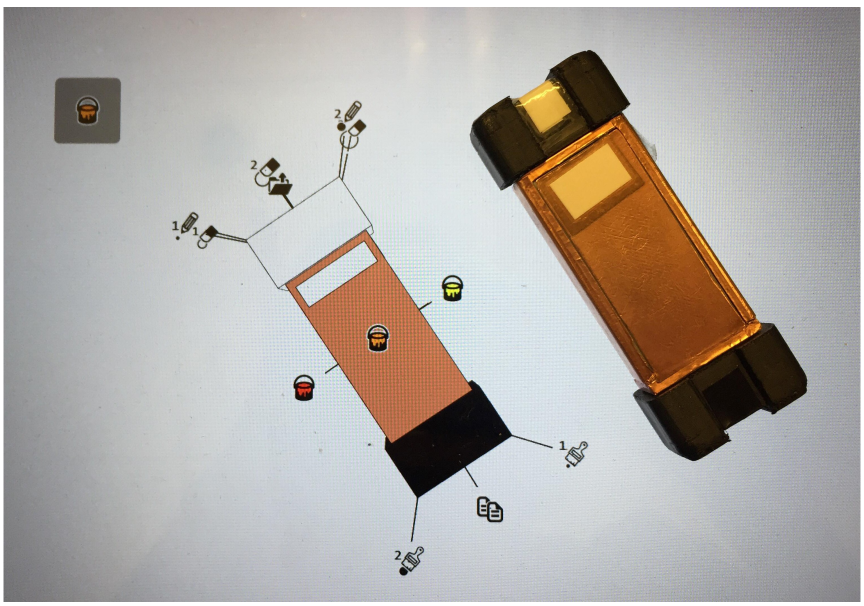


Figure 1.2: Conductive Conté prototype with on-screen guide in game.

Results show it is possible to learn command-contact mappings for both selection techniques and command set sizes, and learning of *Conté* with the *large* command set varies between participants. The results validate the feasibility of the original design space of Conté and show that commands located on spatial landmarks on cuboid-shaped pen-like input devices can be learned, particularly through a rehearsal-based interface. To show a wide variety of possible real-world use cases we discuss three demo applications.

1.1 Contributions

This thesis makes the following contributions:

- Design and fabrication of a conductive prototype of Vogel and Casiez's Conté.
- Implementation of software synthesizing touch and orientation data to output Conté event messages usable on any platform to create applications with Conté.
- An experiment comparing contact-to-contact transition times on three different sizes of Conté mockups.
- A design and implementation of an on-screen guide to facilitate learning of commands located on the exterior of a 3D tangible input device.
- An experiment comparing learning and recall between a 3D and 2D spatial interface for two sizes of command sets.

1.2 Organization

The remainder of this thesis is organized as follows:

- Chapter 2 describes previous and related work on modes and mode-switching, contact-based multimodal input devices with particular attention to Vogel and Casiez's Conté, the impact of form-factor on device manipulation, rehearsal interfaces, and spatial interfaces.
- Chapter 3 describes the design, fabrication, and implementation of our conductive Conté device.
- Chapter 4 describes the design and empirical evaluation of a study of the impact of size form-factor on contact-to-contact transition times.
- Chapter 5 describes the design and empirical evaluation of an exploratory study of learning and recall of command-contact mappings.
- Chapter 6 discusses three applications that demonstrate the use of Conté in real-world scenarios.
- Chapter 7 summarizes the conclusions of the previous chapters and concludes this thesis.

Chapter 2

Related Work

This work relates to modes and mode-switching, contact-based multimodal input devices with particular attention to Vogel and Casiez's Conté, the impact of form-factor on device manipulation, rehearsal interfaces, and spatial interfaces.

2.1 Modes and Mode-switching

Hinckley et al. define modes as simply "using one input device to do multiple things" [7]. Although Tesler argues for the aspirational goal of a "modeless" interface [34], Sellen et al. point out that modes are unavoidable in even moderately complex graphical applications [30]. Hinckley et al. enumerate common modes in simple pen-based note taking applications, such as inking, gesturing, selecting, erasing, highlighting, and navigating. Most current pen applications use toolbar buttons to switch modes, but this takes up screen space and creates temporal modes [1] that often cause users to make "mode errors" [30]. Alternative mode-switching techniques exit a selected mode at a logical end-of-input event, like when the pen is lifted up (called a quasimode [27] or spring-loaded mode [30]). These work well when there is a well-defined primary mode, like inking, and temporary secondary modes, like deleting. An even better mode-switching strategy is entering and exiting a mode based on some identifiable action performed with the input device, such as using the eraser end of the pen [20], or any of the contacts on multimodal input devices such as Conté [39].

2.2 Contact-based Multimodal Input Devices

Vogel and Casiez use of the term multimodal input device to mean an input device whose primary purpose is to facilitate easy mode-switching [39]. Multimodal interfaces more commonly refer to systems that combine multiple input modalities or methods, like speech and pen, to make interaction more natural [23, 3]. We use Vogel and Casiez's terminology in this work.

Fitzmaurice et al.'s Bricks demonstrate the use of extruded-rectangles as tangible, physical handles for manipulating on-screen objects [4]. Bricks are Lego-sized and sit on a large horizontal display. A brick placed on a virtual object becomes its handle, and translating or rotating the brick translates or rotates the virtual object. Multiple bricks are used to stretch or deform an object. Claims about the advantages of their device include: taking advantage of spatial reasoning, leveraging existing skills for manipulating physical objects, and offering a one-to-one mapping between control and controller.

With the goal of removing menus and toolbars from the screen, Rekimoto and Sciammarella created Toolstone [28]. Toolstone a is cordless, $25 \times 40 \times 50$ mm block-shaped device, used in the non-dominant hand for mode-switching, and is intended to be used in conjunction with a mouse or stylus. A WACOM tablet that emits magneto-electro signals was used, and three coils from WACOM pens were embedded in three edges of Toolstone. When one face of Toolstone touches the tablet, only the coil in its adjacent edge is close enough to receive and respond to the signal. The resonance response pattern is different for each coil and is used to identify the coil and measure its position on the tablet and its angle. Once the responding coil is identified, its angle indicates which of its two adjacent faces is touching the tablet. Toolstone is only a mode-switching device and can only detect which face contacts the surface.

Van Laerhoven et al. [38] created a cube-shaped object with embedded hardware that can sense which of its six sides is facing up as well as the which of the other four sides in facing the user. The cube system always assumes one of the six sides is facing upwards and can only accurately determine its orientation if it is rotated in 90 degree increments and never rotated parallel to the ground. In one demonstration application, each side of the cube was used to represent a preset profile in an audio mixing application. Visual indicators on each of the cube were semantically related to the profile they represented.

A digital pen can be used as a mode-switching device as well. Many commercially available pens have an "eraser button" to enable single-handed mode-switching [11, 21] and several include non-dominant hand devices to extend the number of modes available [12, 21]. The limited number of contact points in pens means other properties, including roll [2], pressure [10, 26], and tilt [21, 35] must be leveraged to actuate modes.

2.3 Vogel and Casiez's Conté

Vogel and Casiez's Conté [39] is a one-handed input device intended to facilitate easy mode-switching, shaped as an extruded-rectangle , and inspired by artists' Conté crayons. A command is performed when one of the 26 contacts (8 corners, 12 edges, and 6 sides) touches the screen. They interview artists to shape the design space, but do not empirically test contact-to-contact transition times. The novelty of the device is that each contact could perform a different command if they were all uniquely identifiable. Furthermore, their prototype uses infrared light on a diffuse illumination table and uses computer vision techniques on infrared images to determine which contact touches the table. They could not distinguish between contacts that look similar. For example, they could distinguish between a long edge and a short edge, but not between two corners. As a result, they could only identify 10 of the 26 possible contacts and could not study recall of a 26 contact-point input device.

2.4 Form-Factor and Device Manipulation

Many factors of an object, including form-factor, can influence how users manipulate it. Olafsdottir et al. investigate how users adapt their grip of virtual object on a multitouch tabletop when asked to manipulate it [22]. They find grip is influenced by starting position, target position, and anticipated rotation. They do not study the impact of size. Perelman et al. create the Roly-Poly mouse to combine 3D translation, 3D rotation, and 2D pointing into a single device [24]. In an informal test they find an 80 mm diameter prototype is easier to handle and translate than one with a 60 mm diameter or a 100 mm diameter. Using an 80 mm diameter prototype, they find 3D pointing tasks are faster with Roly-Poly mouse than a similar but differently-shaped commercial device.

A standard pen-like device only has two contact points: the nib and the end, but the interaction space of a pen is expanded when combined with touch and additional sensors. Different natural grip styles and poses of pen-like input devices used in conjunction with touch input have been explored by Hinckley et al. [8]. Hinckley et al. create a new design space using pen and touch together, and argue pen is best for writing and a pen+touch combination is best for tools [9]. Song et al. explore the use of multitouch sensors on a pen barrel to sense grip for mode-switching [31]. Sun et al. augment a digital pen and tablet with sensors to enhance the naturalness of digital drawing [32]. However, none of these empirically evaluate manipulation times for their techniques, and the tangible manipulation space is much smaller than what we investigate. Perhaps the closest empirical evaluation is Li et al., who investigate five techniques for pen mode-switching. They find that pressing a button with the non-preferred hand is the fastest method and flipping to an "eraser" end is among the slowest [20]. Their analysis is only for binary mode selection and does not include an investigation of the impact of device shape on performance. Mode-switching time has also been investigated for touch input [33] by Surale et al.. They use an experimental protocol based on Li et al. [20] in which a user crosses targets, alternating between baseline input and a mode-switching technique. Mode-switching with Conté is performed by touching the screen with a new contact, and not all contacts are used for input. We do not use the same protocol since it is not applicable to Conté mode-switching. We investigate the impact of size form-factor on mode-switching time where a mode-switch is accomplished through a contact-to-contact transition on an extruded-rectangle tangible input device.

2.5 Rehearsal Interfaces and Spatial Interfaces

On a desktop system, commands are typically selected from menus, particularly when a user is new to a specific piece of software (i.e. the user is a novice). Keyboard shortcuts are commonly used by experts, but the physical action of performing a shortcut is completely different from the action of selecting from a menu. The novice technique does not help a user learn the expert technique. Kurtenbach et al. introduce Marking Menus and the idea of a rehearsal-based interface: an interface in which the physical action of making an expert selection is rehearsed through novice selections [14, 17]. In Marking Menus, command selection is performed by drawing a scale-independent stroke through a radial menu. Labels in the menu appear after a delay to aid novices, but a command can be selected before the labels appear. A lab study [15] and a more realistic case study [16] validate the effectiveness of Marking Menus and the rehearsal technique.

Rehearsal-based interfaces have become a widely studied topic and have been used in spatial interface learning. FastTap [6] is a 2D grid-based selection technique that exploits spatial memory to outperform the speed of command selection in Marking Menus. An on-screen grid has a command in each cell, and the corners and edges of the screen act as physical spatial landmarks. Commands are selected by pressing a command button the bottom-left corner of the screen and tapping the desired cell. FastTap is a rehearsal interface: icons representing commands are displayed in their respective cells after a short delay, but users do not have to wait for the delay to pass, they can even select a command and activate the command button simultaneously. Delay time is not specified in this work but other FastTap interfaces use a 150 ms delay [5], and a 250 ms delay [18].

Gutwin et al. compare the adoption of expert selection in FastTap in a game called Shape Slicer and a drawing application [5]. The game rewards rapid selection while the drawing application does not have time constraints. They found that in a time-constrained game, users quickly transitioned to, and sustained use of, the expert selection method in a 12-item FastTap interface.

However, while using a 24-item FastTap interface in a relaxed drawing application in a ten week longitudinal study, users rarely adopted the expert selection method at all. Using a similar game and 12-item FastTap menu, Lafreniere et al. [18] show that users who predominantly use expert selection by the end of game continue to use expert selection in a drawing application over the course of week. This suggests that rehearsal-based learning occurs best in time-constrained tasks, but transfers well to tasks without these constraints.

Lafreniere et al. [19] investigate interaction techniques on rectangular touch screen watches. Their techniques rely on spatial memory, and they study a variety of sizes of 2D grids. To provide visual guidance for their system, they use faint transparent grid-marks. For 3×3 grids, they found the natural landmarks provided by the corners and edges of the watch face rendered the use of the grid-marks unnecessary since each command, aside from the center, is located at one of these landmarks.

Physical landmarks on the exterior of a screen become ineffective in large grid-based interfaces. It is easy to remember that a command is located one block to the right and one block above the bottom left corner, but difficult to remember that a command it is located four blocks to the right and seven blocks above the bottom left corners. There are simply too few physical landmarks for the number of commands in a large grid. To mitigate this, Uddin et al. [36] added artificial landmarks to a large grid-based interface, and they significantly increased users' ability to remember command locations. Their results also show that simple landmarks like a small number of shaded rectangles in a grid improve spatial memory more than detailed landmarks, like a transparent image on top of the grid. Instead of adding artificial landmarks to a 2D interface, 3D interfaces like Conté add another dimension to increase the number of physical landmarks; this could have a similar effect.

Udin et al.'s HandMark menu is another technique that adds spatial landmarks to a large display [37]. In a HandMark menu, hands and fingers are used as spatial landmarks when touching the screen. Items in the menu are either placed in the open spaces between fingers or in a grid placed between the thumb and index finger. Novice to expert learning is facilitated by rehearsal. When the hand is placed on the screen, users can tap the location of an item immediately or wait for 500 ms until icons representing commands are displayed. On a 42-command HandMark menu, command selection is 0.6 seconds faster than standard tabs with similar errors and greater user preference.

Schramm et al. [29] evaluate designs of hidden toolbars that require fewer steps for command selection than hidden toolbars found in common commercial mobile interfaces. Two of their selection designs provide shortcuts that leverage users' spatial memory. The two designs perform 700 ms faster than standard hidden toolbar designs. Colours were used as artificial landmarks for all commands, but the results show that commands on and around the device's physical landmarks were selected approximately 200 ms faster and with more accuracy.

In Perrault et al.'s Physical Loci [25], commands are associated with objects in a room and invoked by pointing at the object. Novice to expert learning is facilitated with an optional on-screen guide: a TV screen displayed an image of the room showing the command-object mappings on the user's request. This 3D interface relies on spatial memory, in addition to object and semantic memory, and is related to Conté learning. However, commands are placed on objects *inside* a 3D space (the room), unlike Conté where commands are placed on the *exterior* of the device. Participants remembered 47/48 command locations one week later.

Zhai and Kristensson [40] found that users with no previous experience using an ATOMIK stylus keyboard, learn an average of 58.67 out of 100 distinct keyboard gestures in five sessions with a minimum of one day between them. The ATOMIK keyboard layout is optimized for stylus input and is different than the commonly used QWERTY keyboard. A gesture for a word is the movement pattern that results from tapping the word; tapping is rehearsal for using a gesture. This is not a spatial interface, but is an example of rehearsal-based learning on a large command set. The 58.67 commands learned is significantly more than the maximum 26 new commands on Conté and sets a precedent for users learning a large number of new commands over a short period of time in a rehearsal-based interface.

To the best of our knowledge, there is no work investigating spatial learning of commands using physical landmarks on a 3D input device like Conté. We build on the success of rehearsal-based learning in the 2D case, particularly Gutwin et al.'s FastTap [6] since its use of corners and edges on the perimeter of a touch screen is the closest 2D analog to corners, edges, and sides on the exterior of a cuboid. Lafreniere et al.'s [18] tuning of Gutwin et al.'s Shape Slicer game [5] proved to be an effective method for expert adoption of FastTap and also for transfer of this expert adoption to a relaxed task. We build on this as well.

Chapter 3

Device Prototype

Our device follows Vogel and Casiez's design objectives that Conté must work with an unaltered touch device and must be simple enough for researchers and hobbyists to recreate. We also added our own constraint that prototyping Conté applications should be possible on any platform. On the software side, this meant we could not use blob size and/or shape for contact classification since this requires software to be tailored to specific operating systems and in some cases hacking the OS all-together. On the hardware and fabrication side, we did not print custom PCBs and we used a consumer-level Ultimaker 2+ 3D printer for the outer shell. Vogel and Casiez's prototype emitted and reflected infrared light and classification used computer vision techniques on images captured by a diffuse illumination table. This meant they could distinguish contacts that looked different but not those that looked the same; they could distinguish a short edge from a long edge, but not corner 1 from corner 2. As a result, they could only identify 10 of the possible 26 contacts. In contrast, our conductive IMU-based prototype reliably detects all 26 contacts.

3.1 High-level Description and Terminology

Conté is shaped as an extruded rectangle (or cuboid). It has 26 *contacts*: 8 *corners*, 12 *edges*, and 6 *sides*. A different *command* or *mode* can be assigned to each one, and touching the screen with a *contact* activates its associated *command* or *mode*. The corners are small and can be used for precise input like a stylus. Edges and sides can be used for input as well. For example, an edge can be used to create a straight line on screen, and a side can be used to stamp a signature on a page. All contacts can also be used to change modes like 'change draw colour to green', or execute a command like 'undo'.

Large conductive objects touching a capacitive screen can be ignored or interpreted as multiple objects by the system. To address this problem, large *contacts* are broken down into multiple *points* (see Figure 3.1). *Contact* refers to which corner, edge, or side is used while *point* refers to the physical material on the prototype that touches the screen. There are single-point contacts, two-point contacts, and four-point contacts. Two-point contacts and four-point contacts are referred to as multi-point contacts.

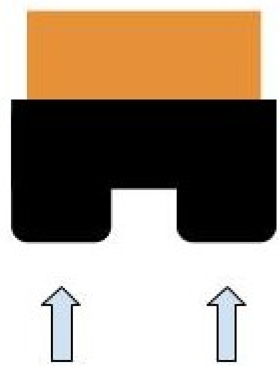


Figure 3.1: Two-point contact.

We give each *contact* a unique name for easy reference (see Figure 3.2). One end of Conté has white markings on it; we refer to this as *white end* and the other end as *black end*. Contacts on *white end* have names beginning with "W", *black end* begin with "K", and contacts along the body or barrel begin with "B". "C" denotes a corner, "E" and edge, and "S" a side.

3.2 Fabrication

Kratz et al. explain that capacitive touch screens have a grid of capacitors at discrete locations on the screen. Since a user is typically grounded with respect to the screen, a touch modifies the capacitance at the touch location which enables the touch to be detected [13]. This works because humans are conductive. We create a conductive Conté device that acts as a conductive extension of the human hand when touching a capacitive screen and is registered as a normal touch.

Our prototype is 85 $\times$ 30 $\times$ 15 mm at its largest points. To create multi-point contacts, the body is recessed inwards by 1.5 mm in width and height (see Figure 3.3). The outer shell of our prototype was 3D modelled in 3ds Max. It consists of four parts: top case, bottom case, and

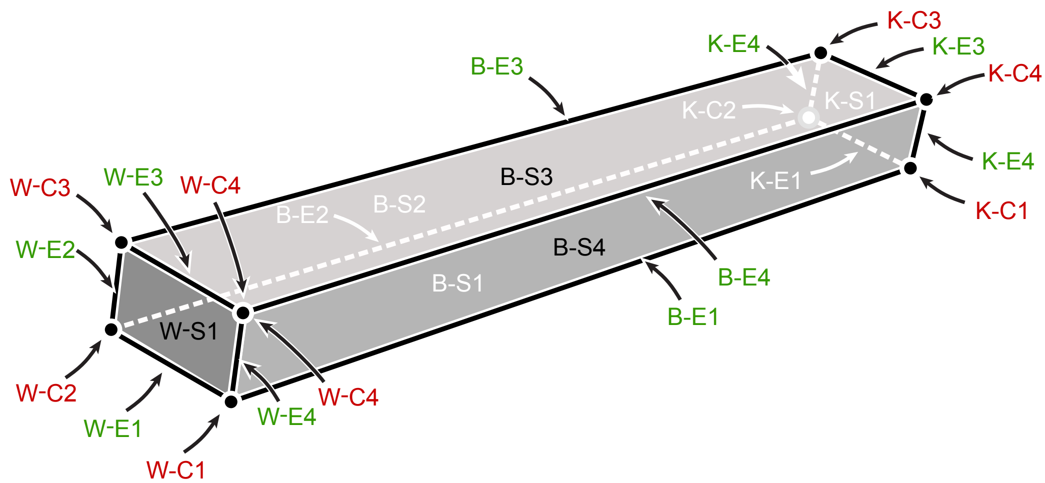


Figure 3.2: Contact names.

two identical end points. The top and bottom case slide together and snap shut so that they are secure while being used but can be opened easily to change the battery or upload new code to the microcontroller. The case pieces were 3D printed using polylactic acid (PLA) filament (non-conductive hard plastic), and the exterior was covered in copper tape for conductivity. Since the case is not as wide as the ends and never touches the capactive screen, we were not concerned with scratching from the copper tape. A white marker was added to W-S1, B-S3, and B-S2 so that every Conté contact is visually distinguishable from the others. Since we are studying the use spatial landmarks in learning, we only added visual markers to disambiguate the spatial landmarks and chose not to add extra visual landmarks.

The end point were modelled with rounded edges to prevent screen damage. The middle of the endpoints is recessed inwards by 1.5 mm and sits flush with the case. To ensure Conté's corners are not dismissed as conductive noise, they need to contact the screen with a large enough surface area, but making them too large decreases their input precision and makes it difficult to feel the differences between a corner and an edge. Another solution is to make them squishy so that their surface area increases when used. The end points are printed from conductive 95 Shore A thermoplastic polyurethane (TPU) [1], which is squishy, but not quite squishy enough. In order to combat this, we built small air pockets into the corners of the model. To do this we made copies of the corners, shrunk them slightly and placed inside the 3d-model's corners with a small gap between the actual outer corners and the new interior corners. Walls were added to the

[1] http://rubber3dprinting.com/pi-etpu-95-250-carbon-black/

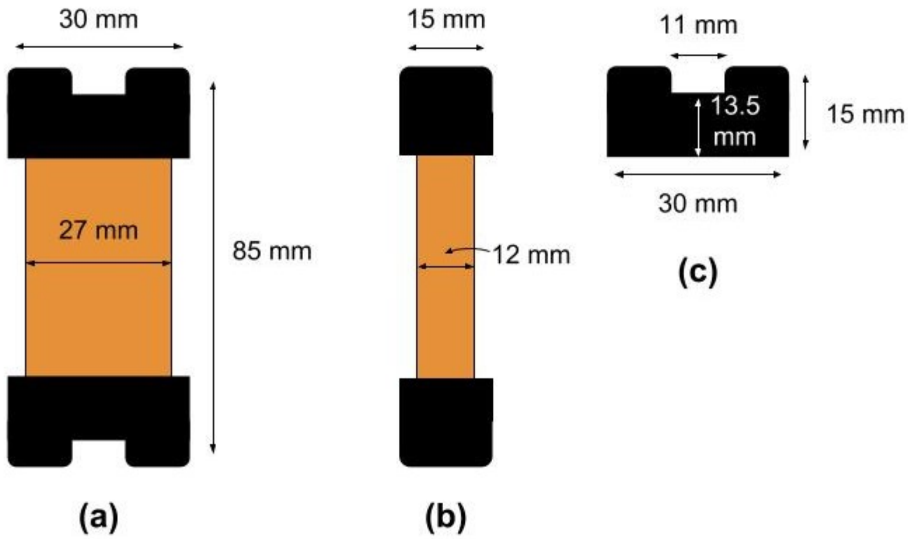


Figure 3.3: Conté dimensions: (a) front; (b) side; (c) end point front.

interior corners to close them off, and their normals were flipped to stop the printer from adding filament inside the pockets (Figure 3.4).

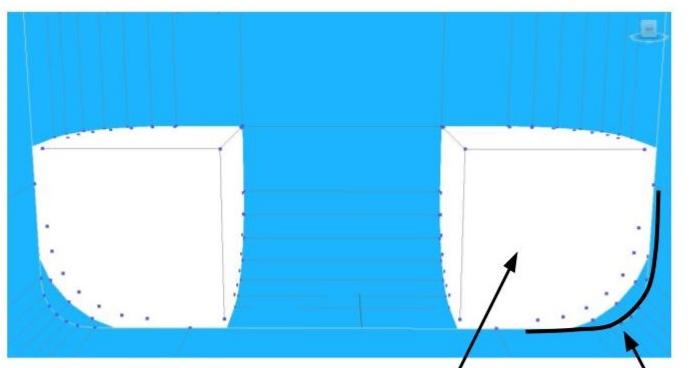


Figure 3.4: Inside corner 3D model.

Each endpoint is $30 \times 15 \times 15$ mm. A middle section 11 mm wide is recessed inwards by 1.5 mm to create two bumps used for two-point and four-point contacts (see Figure 3.3). To help users feel the difference between a corner and an edge, it was important that corners feel squishy and edges feel sturdy. To achieve this, we designed bumps to be large enough to each contain an air pocket but still have space outside the air pocket filled with material. The distance between the two bumps had to be large enough that it would be possible use distance thresholds to smooth touch data from each bump separately. Skipping is discussed in detail below. It was also important to keep the total endpoint size as small as possible since corner input precision would suffer if the ends were too big and bulky. To satisfy all of these constraints, bump sizes and distances between them were determined through iteration.

3.3 System Design

An IMU inside the Conté case senses orientation and acceleration and transmits it to an attached microcontroller which encodes it into User Datagram Protocol (UDP) messages that are sent over wifi. A host application on a touch-screen device sends out TUIO touch events. A computer running Conté software receives the orientation data and TUIO touch events. It processes this information and sends out OSC messages containing Conté events (see Figure 3.5). The host application and Conté software can run on the same machine but do not need to. We discuss the hardware components inside Conté and Conté software. A host application is separate and can be written on any capacitive touch device that can send and receive OSC messages.

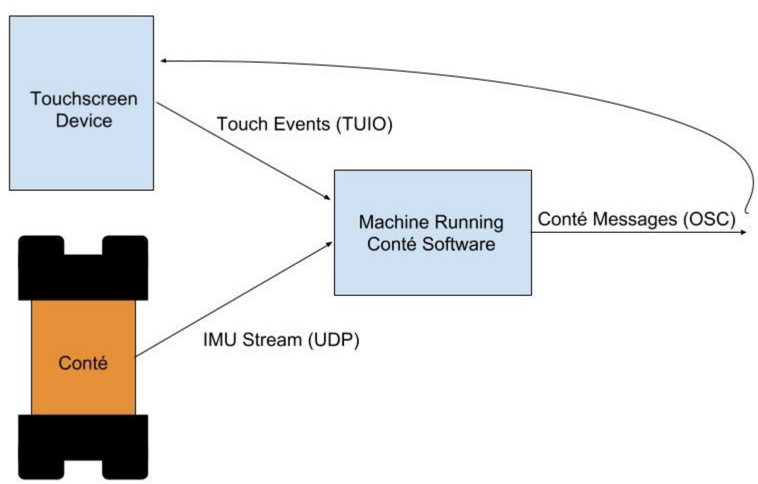


Figure 3.5: Communication pipeline.

3.3.1 Hardware

The case contains hardware to enable sensing and communication (see Figure 3.6):

- Microcontroller: Wifi-capable ESP8285[2] measuring 17.78 $\times$ 25.4 mm. Programmed through Arduino environment.
- IMU: "Ultimate Sensor Fusion Solution"[3]. A 12.7 $\times$ 17.78 mm 9 DoF IMU (3 DoF accelerometer, 3 Dof gyroscope, 3 Dof magnetometer) with a dedicated on-board chip to perform sensor-fusion.
- Battery: 3.7V, 150mAh LiPo battery [4].

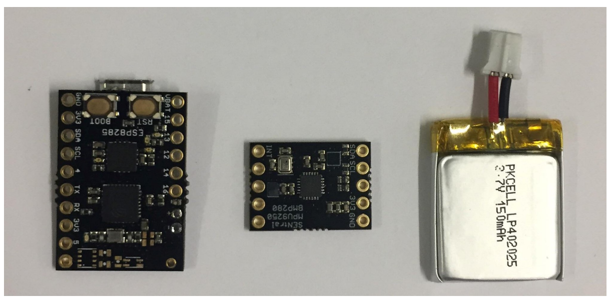


Figure 3.6: Left to right: microcontroller, IMU, LiPo battery.

We soldered a 20 mm wire to a through-hole at the end of the ESP8285 board's antenna trace and soldered wires connected to a 2-pin male JST connector for our battery. Wires attached to the battery were cut down, and we added a new female JST connector so it could fit in our case and connect to the microcontroller (see Figure 3.7). Euler yaw, pitch, and roll, and acceleration in x, y, and z, are transmitted from the IMU to the ESP8285 over soldered wires and from the ESP8285 to a laptop over wifi (see Figure 3.8).

3.3.2 Software

A program written in Processing combines touch and orientation data into Conté events. Processing is built on top of Java and allows Java libraries to be imported. We use the TUIO protocol to

[2] https://www.tindie.com/products/onehorse/esp8285-development-board/
[3] https://www.tindie.com/products/onehorse/ultimate-sensor-fusion-solution/
[4] https://www.adafruit.com/product/258

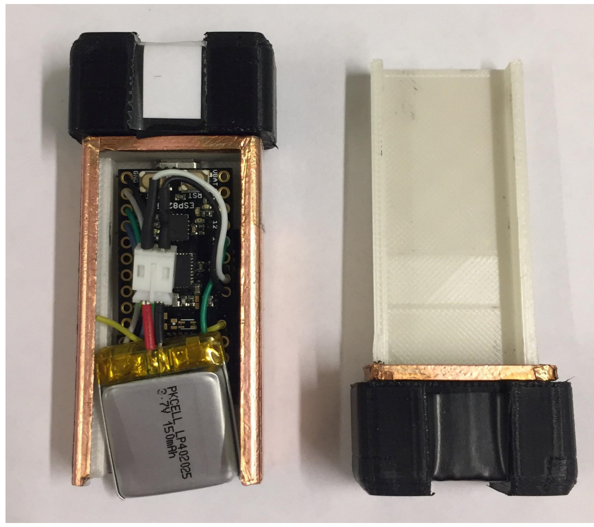


Figure 3.7: Conté with internal hardware.

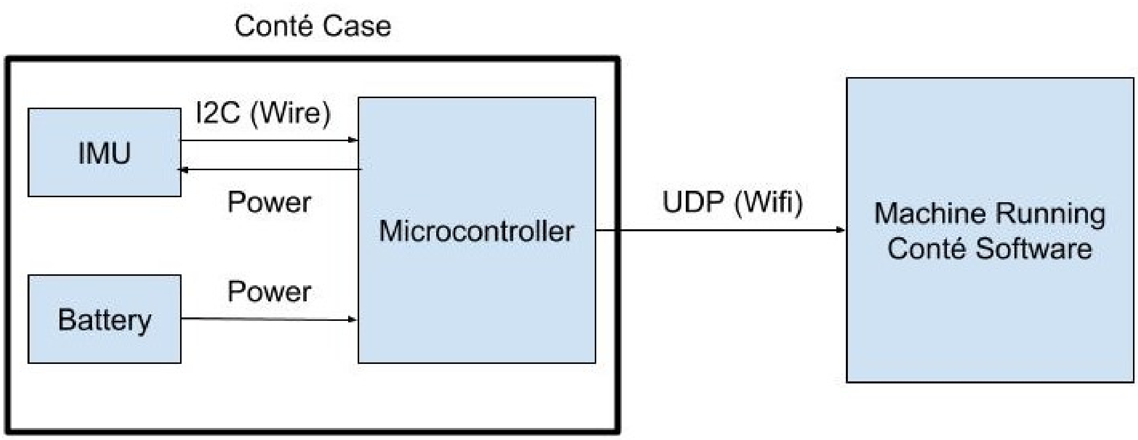


Figure 3.8: Conté schematic.

receive touch events. TUIO is a touch-specific protocol built using Open Sound Control (OSC) which is a multimedia communication protocol. TUIO and OSC libraries exist for most programming languages. IMU data is sent from hardware inside Conté, encoded as UDP messages. After processing, Conté events are output as OSC messages. Any capacitive touch device device that can send TUIO messages and receive OSC messages can run a host application that can use Conté.

An Arduino program on the microcontroller inside Conté receives orientation and acceleration data from an attached IMU. Gyroscopes measure angular velocity and *sensor fusion* combines data from a gyroscope, accelerometer, and magnetometer to compute angle. The IMU we used has a dedicated chip that uses proprietary methods to perform sensor fusion.

3.4 Processing and Classification

Conté software receives touch and IMU data. It processes and filters touch data, classifies IMU data to determine which Conté contact is touching the screen, combines touch and classification data together to create Conté events, encodes events as OSC messages, and outputs them so they can be used in a host application.

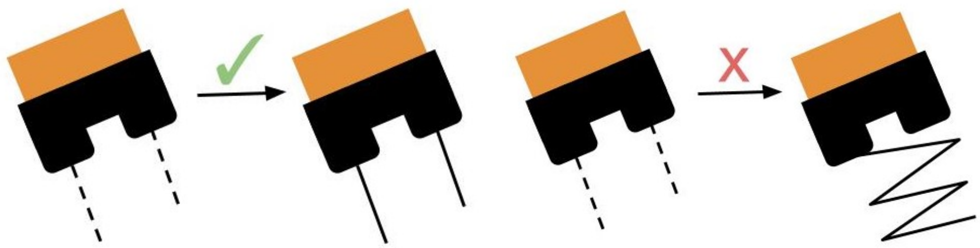
3.4.1 Touch Data Processing and Filtering

Our conductive endpoints are 3D printed using the only commercially available squishy conductive filament (to the best of our knowledge). We have already discussed our technique for corner recognition however the material's hardness also causes touch data to skip if suitable pressure is not applied to the touch-screen using Conté. Touch data is processed to eliminate skipping by looking for down events in close temporal and physical proximity to recent up events. If one is found within 300 ms and a heuristically determined distance, we ignore the up event and merge data from the two points so that our program's output shows continuous movement from a single point. A 45.97 mm threshold is used for *single-point*. A 15.24 mm threshold is used for *two-point* and *four-point* contacts. In practice this is the largest threshold that reliably interprets multiple touch points as separate touch points and smooths strokes from individual points separately (see Figure 3.9).

The proportion of the device's acceleration along the axis perpendicular to and pointed away from the screen is calculated using data from the IMU, and if it at least 75% of the total acceleration and its absolute value is above a heuristically determined threshold then any up events still within the 300 ms limit are immediately processed as actual up events.

3.4.2 Contact Classification

Consider any one Conté contact and assume a touch screen device is flat on a table. When Conté is held so that this contact touches the screen, its roll and pitch must be within a fixed



(a) **(b)**

Figure 3.9: Multi-point smoothing: (a) correct; (b) incorrect.

range. Holding Conté at an angle outside this range, along either axis, would change the contact touching the screen. We train a decision tree on roll and pitch data from the IMU inside Conté to determine these ranges for each contact and classify roll and pitch in real time to determine which contact touches the screen.

A decision tree was trained on 120 data points per contact collected from the first author (see Figures 3.10 and 3.11). Forty trials per contact were conducted, and the first three IMU readings after Conté touches the screen were used from each trial. We collected samples using the full range of angles we deemed to be acceptable for each contact in pitch and roll. The resulting decision tree was trained in Weka with 10 fold cross-validation and is 99.16% accurate. In practice, the two small sides of conté were often misclassified as adjacent edges. Both sides have pitch values closer to +-90 degrees than the other contacts so pitch was used to classify these two sides before the decision tree. After recalibrating the gyroscope and accelerometer, one medium edge was often misclassified as its adjacent large side; pitch was used to adjust this result after classification.

We chose not to train on individual participants because classification is only based on the angle Conté is held at when touching the screen with each contact, and since we captured the full acceptable range in our training data, we did not think this would vary greatly between users.

3.4.3 Combining Touch and IMU Data

Every reading from the IMU is classified by the decision tree. To combine touch and IMU data, each incoming TUIO message votes between the first three IMU readings received after it. In the case of a tie, the first reading immediately after the TUIO message is received is used. This is point classification. Each point is also assigned a Conté classification. The contact classifier was trained on data captured immediately after Conté touches the screen. In order to

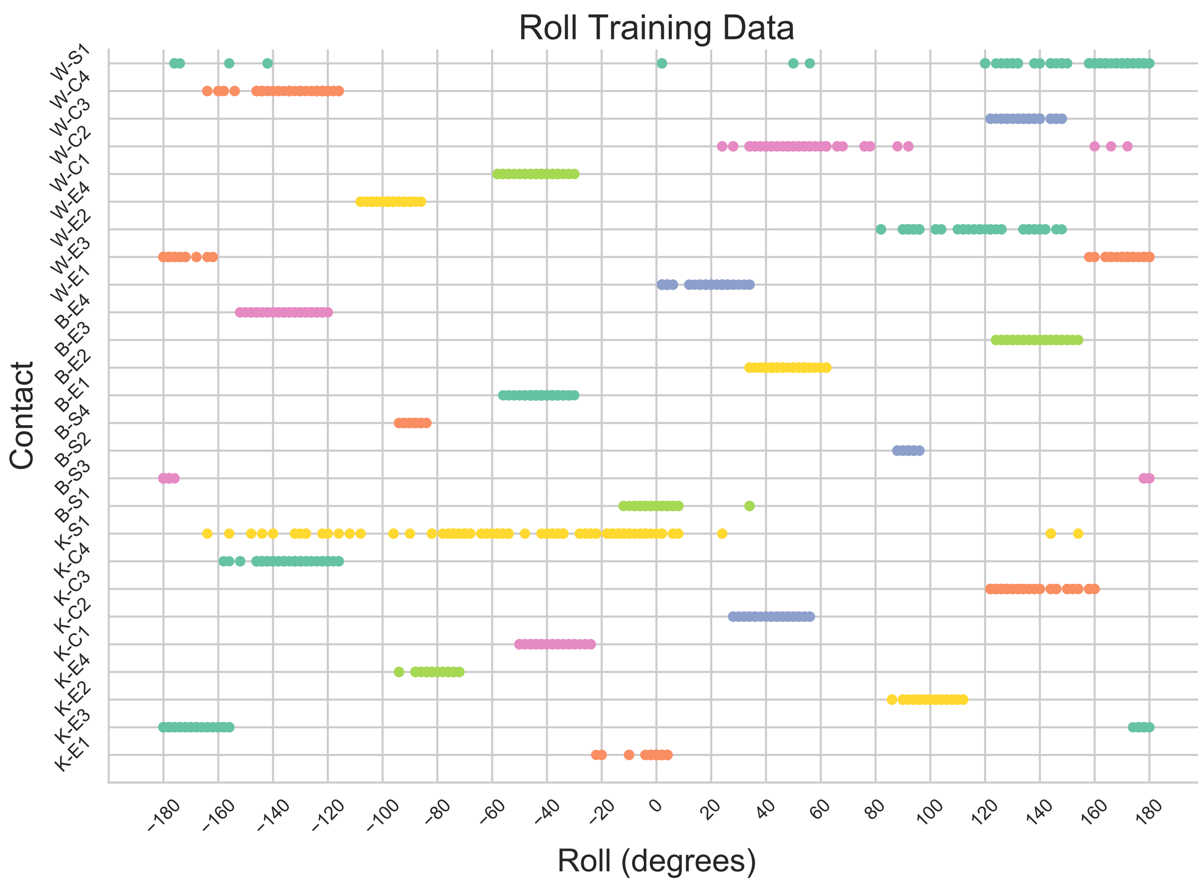


Figure 3.10: Roll training data.

accurately determine which contact is touching the screen, the Conté classification is the voted-on classification for the first *down* message received immediately when Conté touches the screen and is not changed until Conté is removed from the screen entirely. This classification is used by touch smoothing. Output messages for each point contain both the Conté classification and the point classification. When Conté is not touching the screen, each IMU reading is still classified and assigned a location of (-1,-1). IMU data is also used to filter TUIO *up* messages as discussed above.

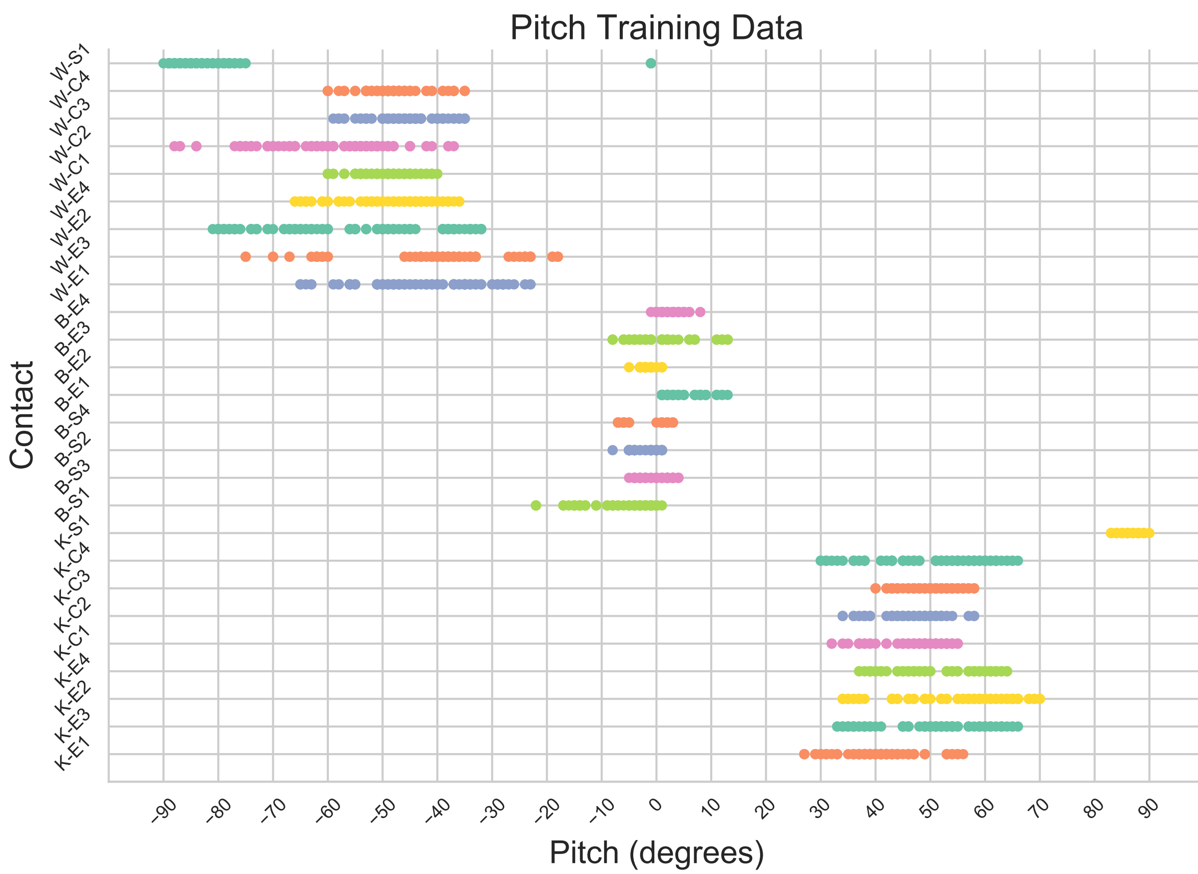


Figure 3.11: Pitch training data.

3.4.4 Output Messages

Our Processing programs outputs four types of OSC messages: down, move, up, and air. Each message contains a combination of touch, IMU data, and vote-decided classification. With regards to *two-point* and *four-point* contacts, a down message is only sent for the first down event, up only for the last up event, and the rest are sent as move events. Independent ids are maintained for different *points* in the same *contact* so that the receiving application can handle them separately if desired. Air messages are sent when Conté is not touching the screen.

3.5 Summary

In this chapter we described the device, next we investigate and validate that manipulation time is reasonable with its larger size. Afterwards, we use the device to evaluate how well people can memorize command locations and demonstrate how the device was used to create a wide range of applications and interactions.

Chapter 4

Manipulation Study

Conté is designed to facilitate quick and easy mode-switching by changing which corner, edge, or side contacts a touchscreen display. However, mode-switching speed was never tested. Furthermore, contact-to-contact transition times are likely associated with device form-factor, particularly size. In practice, size is primarily constrained by embedded hardware necessary for functionality. We have now discussed three sizes of Conté devices in decreasing order of size: our conductive device, Vogel and Casiez's infrared prototype, and a real artist's Conté crayon which Vogel and Casiez argue represents the ideal size. The goal of this experiment is to investigate the impact of size form-factor on manipulation times for contact-to-contact transitions. Mock-ups of all three device sizes are tested. Conté's rectangular shape increases the number of contacts in comparison to a digital pen, however it is not known if this causes a loss in contact-to-contact transition time. To study the impact of shape form-factor on manipulation times we test a mock-up of a digital pen as well. We expect manipulation time to increase with distance between contacts. Since distance between contacts increases with size, we expect transition time will increase with size that and our conductive prototype will be slowest. However, we expect manipulation time with our larger device size to be reasonable in comparison with other sizes. Recall that Vogel and Casiez's prototype was only able to sense 10 of the 26 possible contacts and our conductive device senses all 26.

4.1 Participants

We recruited 12 participants, (mean age 29 SD = 7.6, 6 women, all right-handed). Participants received $15 for their time. Manipulation of contact-to-contact transition uses different hand movements in right-handed and left-handed users. For example, a transition from a corner to an edge to its right is a movement outwards for a right-handed participant, but inwards for a left-handed participant. To ensure these difference did not affect the results, only right-handed participants were studied.

4.2 Apparatus

The experiment was performed on a custom 44 × 40 × 17 cm light box enclosing two lightbulbs and a Logitech camera. A sheet of acrylic and a resistive touch sensor were placed above the camera. Except for a small 8 × 3 cm opening through which video was captured, the acrylic was covered with paper (see Figure 4.1). This reduced the amount of light obstructing participants' view but was large enough for the camera to see a significant portion of all mockups so that colour-classification could be performed. A large monitor was placed behind the lightbox.

Three sizes of Conté mock-ups were 3D printed from polylactic acid (PLA) filament and each face was painted with a different colour. The relative locations of the colours was identical across all three mock-ups. A digital pen mockup was made from a paper stump (derwentart.com) and painted with three colours, one on each of the two ends and one along the barrel. Images from a Logitech C930e camera were processed by a colour classifier written in OpenCV. Hue and saturation thresholds were determined heuristically for each of the six colours used. A unique combination of colours is visible for each distinct contact. The colour-contents of a frame were analyzed to determine which contact was being used. The first frame immediately following a touch on the resistive touch sensor was analyzed; a touch within 100 ms of a previous touch was ignored.

4.3 Tested Device Sizes

Three SIZES of Conté mock-ups were tested: SMALL (63 × 6 × 6 mm), MEDIUM (84 × 11 × 8 mm), LARGE (85 × 30 × 15 mm), and a mock-up of a digital PEN (7 mm diameter, 110 mm length).

SMALL is the size of an artist's Conté crayon. MEDIUM is the size created by Vogel and Casiez. LARGE is the size of our conductive device: the smallest prototype we determined can accommodate a microcontroller, wifi module, antenna, IMU, LiPo battery, and conductive end points that can be sensed on a standard capacitive display. Since Conté is intended to combine inking and mode-switching into a single device, a PEN was tested for baseline comparison.

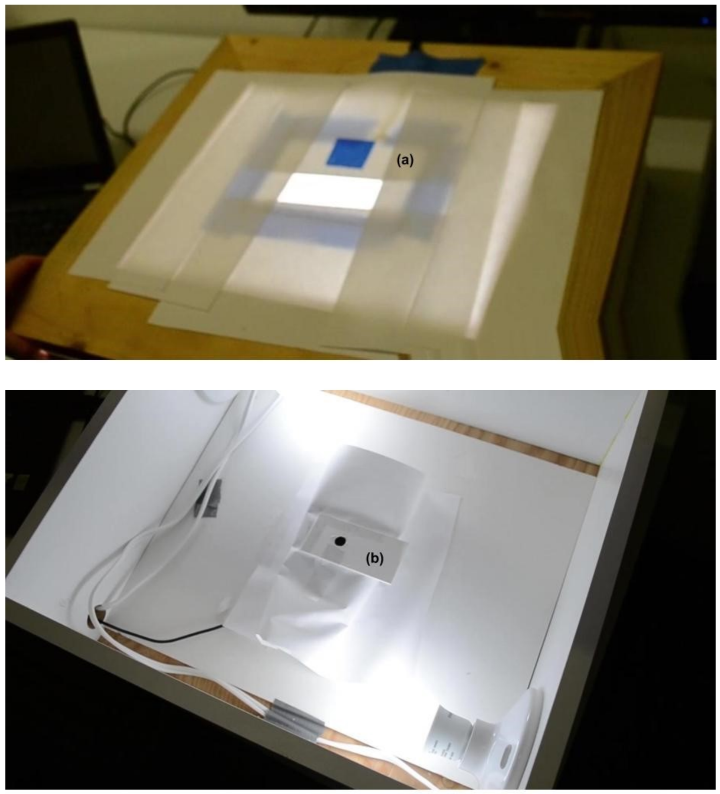


Figure 4.1: Lightbox: (a) resistive sensor on closed top; (b) camera inside.

4.4 Contact Transitions

Recall that Conté contacts are the set of all 26 CORNERS, EDGES, and SIDES of the extruded-rectangle. A TRANSITION is the movement from one contact to another. Since Conté is a pen-like input device, one corner could be used for inking (*start-corner*), with all other contacts used for mode-switching. In this paradigm, a ROUND-TRIP mode-switch consists of a TRANSITION from *start-corner* to any other contact point and a TRANSITION back to *start-corner*. Always using the same corner as *start-corner*, we investigate TRANSITIONS in both directions between *start-corner* and all other contacts. Note that SMALL Conté is an extruded square, so nine TRANSITIONS are omitted due to symmetry. A TRANSITION to touch input is also tested for each SIZE, after the ”palming” action used in Pen+Touch [9].

There are four TYPES of TRANSITIONS: CORNER, EDGE, SIDE, and TOUCH. Since all TRANSITIONS involve *start-corner*, the TYPE of a TRANSITION is the TYPE of the other con-

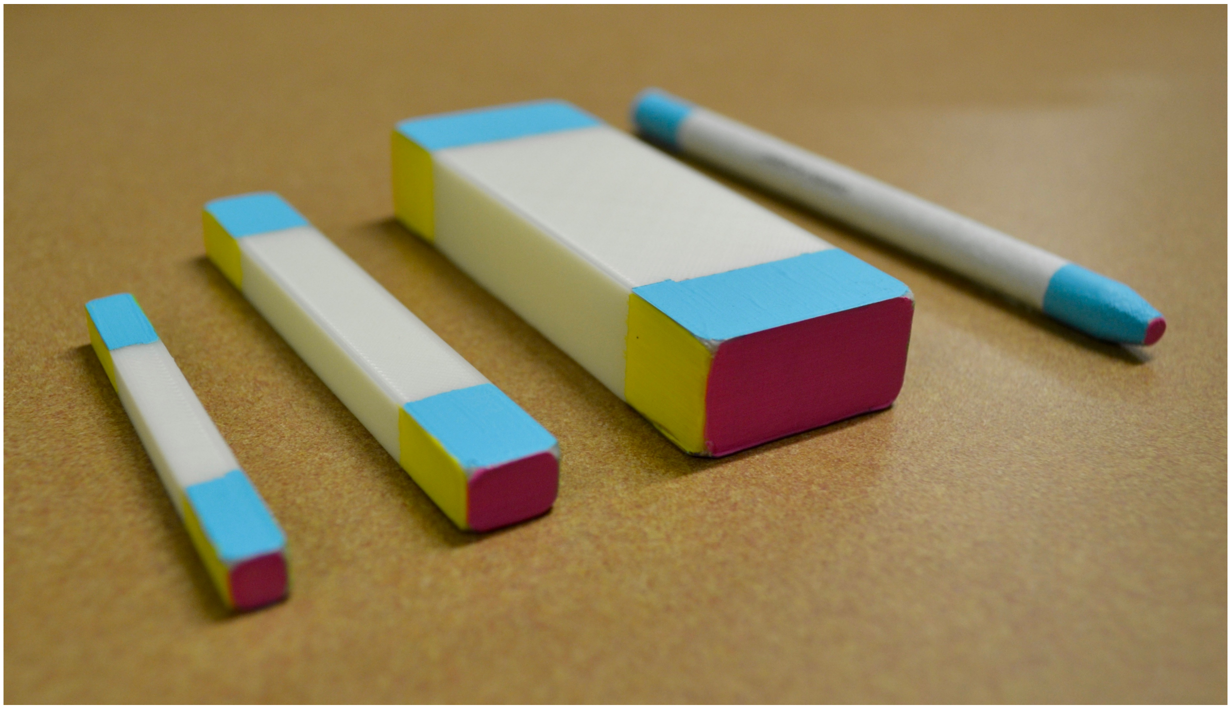


Figure 4.2: Left to right: SMALL, MEDIUM, LARGE, PEN.

tact. TOUCH is a transition from *start-corner* to finger.

The PEN is treated differently since it only has five contacts and is a different shape. TRANSITIONS using the PEN begin at the ”edge” between the nib and barrel to mimic a natural inking position. TRANSITIONS to NIB, BACK, BARREL, BACK EDGE (edge between back and barrel), and TOUCH are tested.

Due to the high number of TRANSITIONS, we categorize them into three discrete GROUPS: (SAME END, BARREL, and OPPOSITE END). SAME END transitions are to CORNERS, EDGES, and SIDES on the same end as *start-corner*. BARREL transitions are to EDGES and SIDES along the middle. OPPOSITE END TRANSITIONS are to CORNERS, EDGES, and SIDE on the opposite end from *start-corner*. These TRANSITION GROUPS can be thought of as a discrete set of three transition distances. GROUPS do not apply to PEN.

4.5 Task

For each TRANSITION, two side-by-side images of contacts are presented on screen with a green rectangle surrounding the active contact. Text indicating the contact TYPE is displayed above the image. The mock-up is used to tap the resistive touch sensor with the active contact, alternating between the two. A TRIAL for a given TRANSITION begins with one of the contacts in the TRANSITION touching the resistive touch sensor, and ends with the other contact touching the sensor. For each TRANSITION, the participant completes as many TRIALS as required until five successful consecutive TRIALS are completed. The last four of the five good TRIALS are used to create two ROUND-TRIP measurements. Sound effects are used to communicate successful and unsuccessful TRIALS to the participant.

Time and location were recorded for each down and up reading from the resistive touch sensor. Time, active contact, classification, and whether contact was correct were recorded for each TRIAL.

4.6 Design and Procedure

The experiment design is within-subject, repeated measures, and full factorial. SIZE order was counter-balanced using a 4x4 balanced Latin square. Within a given SIZE, TRANSITIONS were ordered randomly.

Each participant was given brief instructions on how to interpret on-screen images and how to complete a TRANSITION. Participants were instruction to hold Conté like a pen when possible. They were told to think about the best way to complete a TRANSITION and were allowed to practice on the table before beginning. To obtain practised TRANSITION performance, participants were told to go as fast as possible. Participants were allowed to rest between SIZES but this was not enforced. No other instructions were given.

In summary: 2 ROUND-TRIPS using last 4 TRIALS × (26 LARGE TRANSITIONS + 26 MEDIUM TRANSITIONS + 17 SMALL TRANSITIONS + 5 PEN TRANSITIONS) = 148 data points per participant. The experiment took between 60 mins and 90 mins to complete.

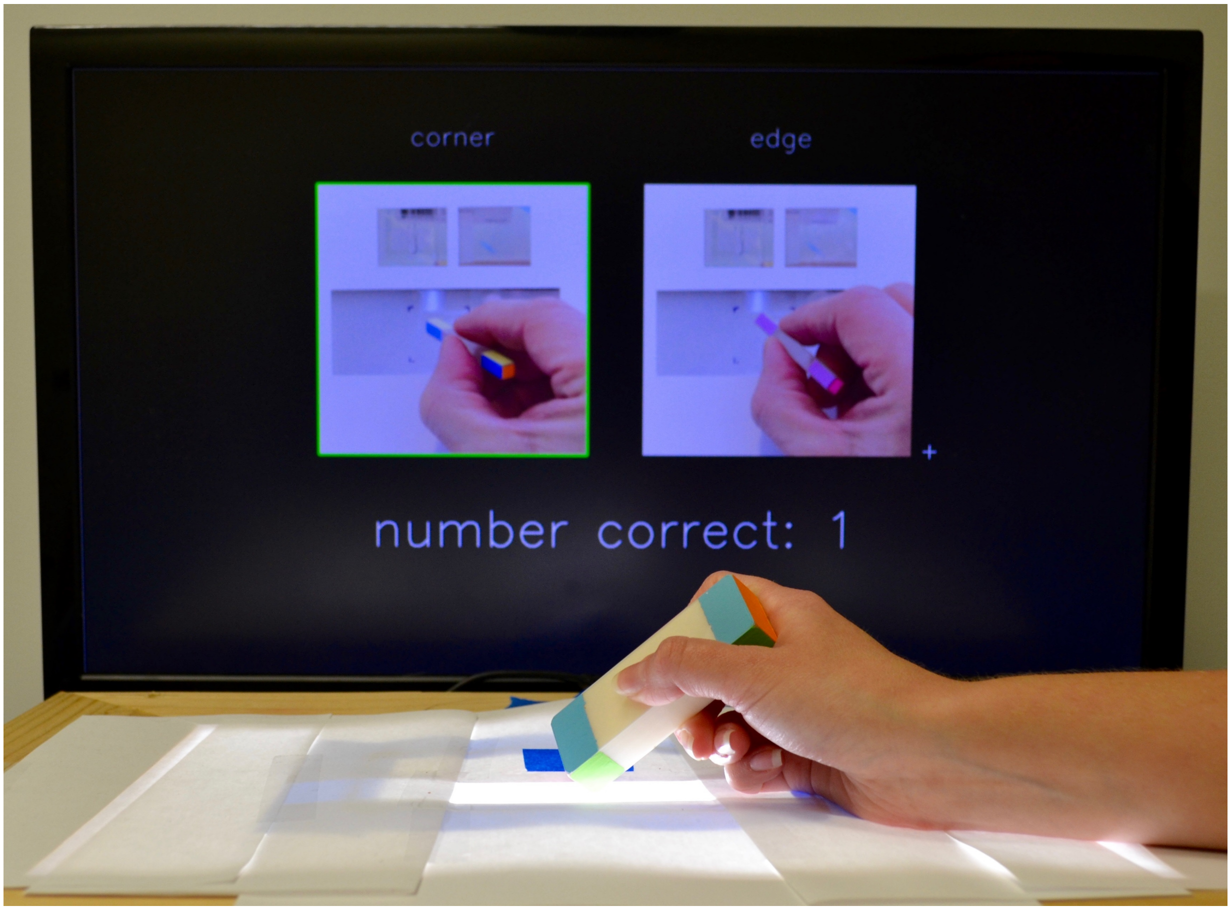


Figure 4.3: Experiment setup and task.

4.7 Results

Repeated measures ANOVA and pairwise t-tests with Holm correction were used for all measures[1]. Time data is aggregated using the mean.

[1]When the assumption of sphericity was violated, we corrected the degrees of freedom using Greenhouse-Geisser (Greenhouse-Geisser's $\varepsilon < 0.75$) or Huynh-Feldt (Greenhouse-Geisser's $\varepsilon \geq 0.75$).

4.7.1 Data Pre-Processing

Recall that SMALL Conté is an extruded square, so nine TRANSITIONS were omitted due to symmetry. For analysis purposes, data from the symmetric counterparts of these nine TRANSITIONS was included twice. We examined transition times for the five consecutive error-free TRIALS to identify outliers more than 3 standard deviations from the mean for each SIZE. This removed 101 of all 4680 TRIALS (2.16%). Removed TRIALS were then imputed using mean substitution, where the mean was calculated from all other TRIALS with the same SIZE and TRANSITION. Imputation was necessary to conduct analysis of the learning effect. Of the five consecutive error-free TRIALS only an even number could be included in round-trips. The first TRIAL was dropped and round-trip times were computed from the last four consecutive error-free TRIALS resulting in: ROUND-TRIP 1 and ROUND-TRIP 2.

4.7.2 Learning Effect

To determine if transition times changed during the four consecutive successful TRIALS, we investigated round-trip time differences between ROUND-TRIP 1 and ROUND-TRIP 2.

Overall, ROUND-TRIP had a significant effect on round-trip time ($F_{1,11} = 23.51$, $p < .001$, $\eta_p^2 = .0036$) with ROUND-TRIP 1 (2.90 s) being significantly slower than ROUND-TRIP 2 (2.80 s). In all subsequent analysis, we use only ROUND-TRIP 2 for the best estimation of total practised round-trip time.

4.7.3 Time

Device Size

Since distance between contacts increases with SIZE, we expected round-trip time to increase with SIZE. Surprisingly, SMALL was the slowest SIZE (mean 3.31 s), followed by MEDIUM (mean 2.68 s), then LARGE (mean 2.41 s). There was a significant main effect of SIZE on round-trip time ($F_{2,22} = 11.42$, $p < .001$, $\eta_p^2 = .1477$). Post hoc tests found SMALL significantly different than MEDIUM and LARGE (both $p < .0001$) and MEDIUM significantly different than LARGE ($p < .01$).

Transition Group

To investigate this further, we compare round-trip times across SIZES when broken down by GROUP. TOUCH transitions are omitted. Round-trip times were aggregated by GROUP per participant and SIZE. We found a significant interaction effect of GROUP × SIZE on round-trip time ($F_{4,44} = 3.26$, $p < .05$, $\eta_p^2 = .0256$). In each GROUP, SMALL was the slowest, followed by MEDIUM, then LARGE, although differences were not always significant. Post hoc tests found no pairwise difference for comparisons involving SAME END and SIZE. For BARREL transitions, there was a significant difference between SMALL and LARGE ($p < .01$) and between all other pairs of SIZES ($p < .05$). OPPOSITE END transitions were significantly slower for SMALL compared to MEDIUM and LARGE ($p < .05$).

Our initial hypothesis was that round-trip time would increase with distance from *start-corner*. Increasing SIZE increases this distance for each TRANSITION, but as discussed above, round-trip times did not increase with SIZE. We suspect this is due to easier physical manipulation of larger devices and investigate the impact of increasing distance within each SIZE separately. Within a given SIZE, distance from *start-corner* increases with GROUP. In ascending order by "distance" from *start-corner*, GROUP order is SAME END, BARREL, OPPOSITE END. We use GROUP as a discrete measure of distance. When broken down by SIZE, round-trip time increased with GROUP and hence distance (see Figure 4.4).

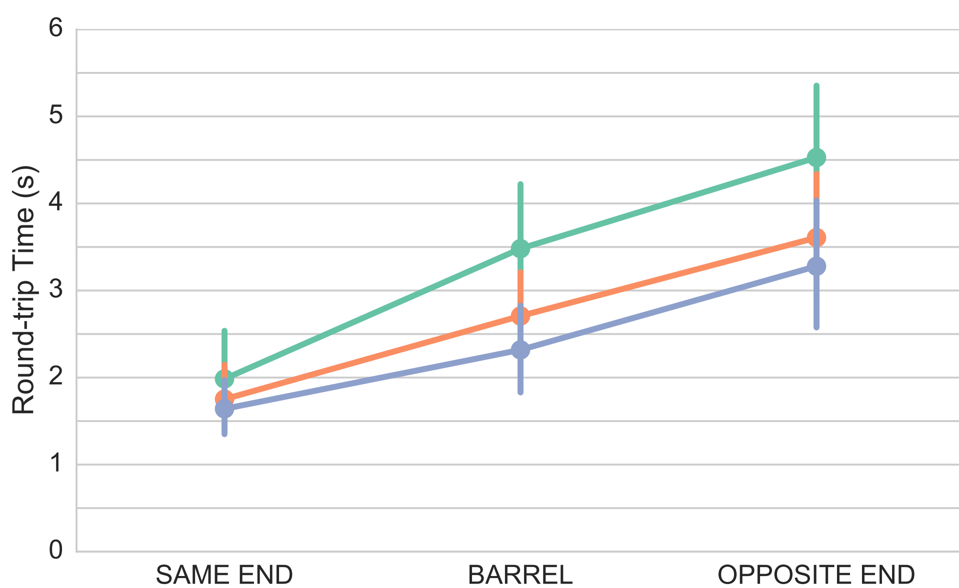


Figure 4.4: Round-trip time by GROUP and SIZE (all error bars 95% CI).

Contact Type

Round-trip times were aggregated by participant, per SIZE and TYPE. A two-way ANOVA for TYPE × SIZE on round-trip time revealed a significant main effect of TYPE ($F_{2,22} = 24.07$, $p < .0001$, $\eta_p^2 = .1761$) on round-trip time. Post hoc tests found SIDE significantly faster than EDGE and CORNER (both $p < .0001$) and EDGE significantly faster than CORNER ($p < .01$). The significant main effect of SIZE on round-trip time has already been discussed and there was no significant interaction effect of TYPE × SIZE on round-trip time.

To investigate this further, we compare the effect of TYPE × GROUP on round-trip time. Note that not all TYPE × GROUP combinations are possible (BARREL does not have any CORNERS). It is interesting to note that while OPPOSITE END was slower than BARREL for all SIZES, TRANSITIONS to SIDE on OPPOSITE END (S: 2.65, M: 2.68, L: 2.03 s) were faster than TRANSITIONS to EDGE on BARREL (S: 4.23, M: 3.21, L: 2.69 s). For SMALL Conté, TRANSITIONS to SIDE on OPPOSITE END were also faster than TRANSITIONS to SIDE on BARREL (2.73 s). All other GROUP orderings were preserved when broken down by TYPE.

Transition

The prefix of a TRANSITION name denotes the GROUP: 'S' means the contact is at the *same end* as *start-corner*, 'B' means the contact is located along the *barrel*, and 'O' means the contact is located at the *other end* relative to *start-corner*. The suffix denotes the contact TYPE and number: 'C' for *corner*, 'E' for *edge*, 'S' for *side*. 'T' denotes TOUCH. See Figure 4.5. For example, 'O-E3' is a transition from *start-corner* to an edge on the opposite end labelled as edge 3, and 'B-S1' is a transition from *start-corner* to a side on the barrel labelled as side 1.

Across all three SIZES, six of the eight fastest and slowest TRANSITIONS are the same. While the order of round-trip time by TRANSITION is not identical between all three SIZES, Figure 4.6 illustrates a shared pattern of round-trip time.

Touch

For TOUCH TRANSITIONS, there was no significant effect of SIZE on round-trip time.

Compared to all other TRANSITIONS, TOUCH is fourth fastest for SMALL (mean 1.52 s), MEDIUM (mean 1.42 s), and LARGE (mean 1.45 s). Aggregating round-trip times by SIZE and participant, TOUCH is the fourth fastest TRANSITION overall (mean 1.47 s).

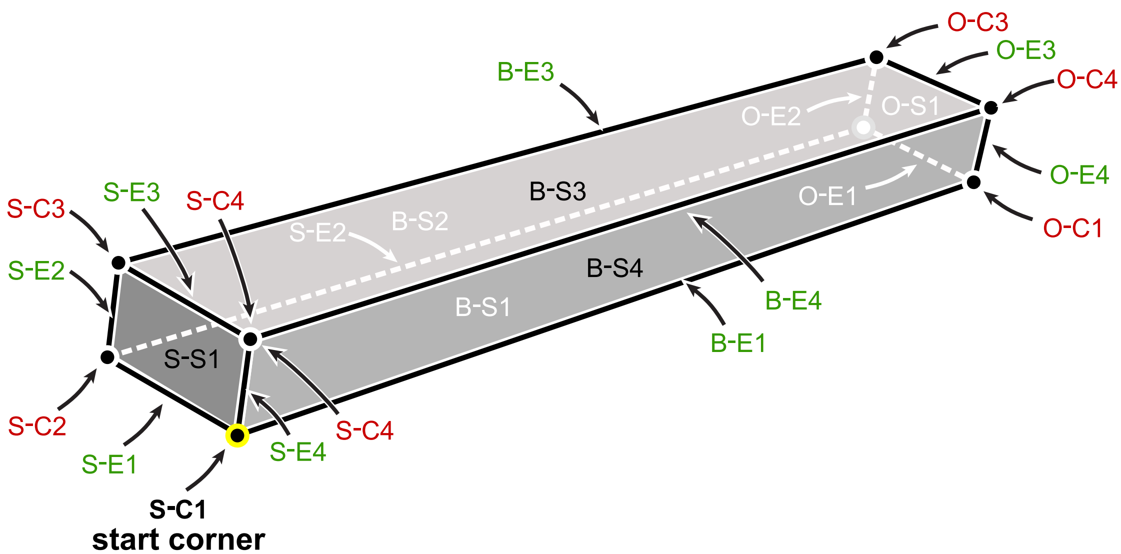


Figure 4.5: Contact names and groups. The prefix denotes GROUP: 'S' means the contact is at the *same end* as the start corner, 'B' means the contact is located along the *barrel*, and 'O' means the contact is located at the *other end* relative to to the start corner. The suffix denotes the contact type and number: 'C' for *corner*, 'E' for *edge*, 'S' for *side*. 'T' denotes TOUCH.

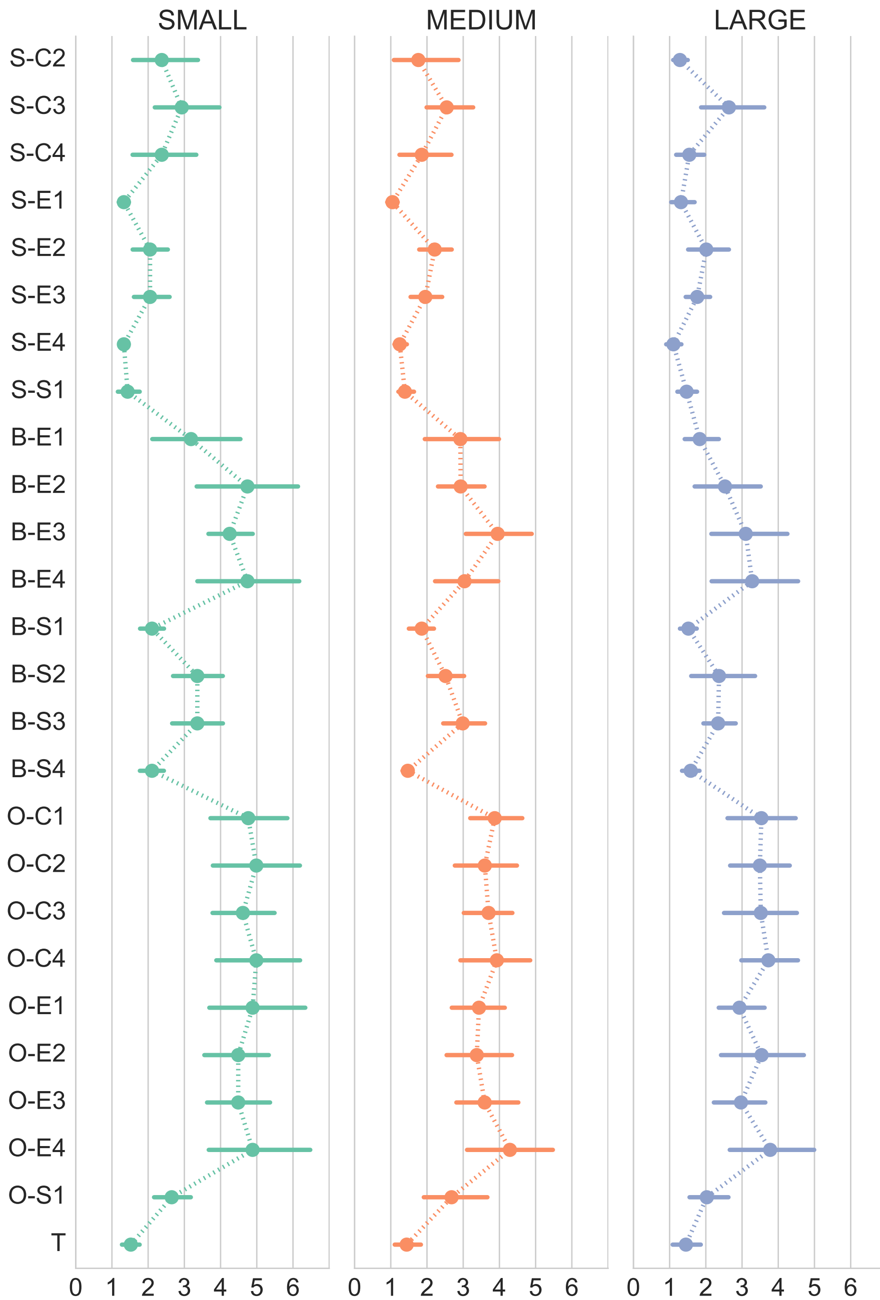


Figure 4.6: Round-trip time by TRANSITION. Note TRANSITION is a categorical variable, dashed lines connecting TRANSITIONS used for readability only. T is a special TRANSITION to touching the display.

Transition		Mean Round-trip Time (s)			
Conte	Pen	Small	Medium	Large	Pen
S-S1	NIB	1.43	1.39	1.47	1.28
B-S4	BARREL	2.11	1.47	1.58	1.78
O-S1	BACK	2.65	2.68	2.03	1.92
O-C1	BACK EDGE	4.76	3.87	3.53	2.01
T	TOUCH	1.52	1.44	1.45	1.29

Table 4.1: Round-trip times for "pen-like" TRANSITION pairs

4.7.4 Pen

We analyze pen separately because it only has five TRANSITIONS (including touch) and is a different shape than the three Conté mockups.

Round-trip times were aggregated by participant, per SIZE. Overall, SIZE +PEN had a significant effect on round-trip time ($F_{3,33} = 21.45$, $p < .0001$, $\eta_p^2 = .3479$). PEN round-trip time (mean 1.66 s) was faster than SMALL (mean 3.31 s), MEDIUM (mean 2.68 s), and LARGE (mean 2.41 s). When comparing all SIZES with PEN, post hoc tests found PEN significantly different than MEDIUM and LARGE (both $p < .01$) and SMALL ($p < .001$).

To make a more direct comparison, we compare round-trip times for all three SIZES with PEN using the five most "pen-like" TRANSITIONS. A TRANSITION is determined to be "pen-like" if the TRANSITION is similar in distance and TYPE to the PEN TRANSITION, with distance prioritized over TYPE. See Table 4.1 for pairing of Conté TRANSITIONS and pen TRANSITIONS and their round-trip times.

A two-way ANOVA for (SIZE + PEN) × "pen-like" TRANSITION on round-trip time revealed a significant main effect of TRANSITION on round-trip time ($F_{1.38,15.20} = 8.23$, $p < .0001$, $\eta_p^2 = .1013$) and a significant main effect of (SIZE + PEN) ($F_{3,33} = 8.23$, $p < .001$, $\eta_p^2 = .1013$) on round-trip time. Post hoc tests found PEN significantly different than SMALL ($p < .01$) and MEDIUM and LARGE (both $p < .05$). There was also a significant (SIZE + PEN) × TRANSITION interaction effect on round-trip time ($F_{12,132} = 6.81$, $p < .0001$, $\eta_p^2 = .1516$). Post hoc tests found a significant difference for (O-C1, BACK EDGE) TRANSITIONS between PEN and SMALL and MEDIUM (both $p < .001$), and LARGE ($p < .05$). There are no significant pairwise differences between PEN and any SIZE for any other "pen-like" TRANSITION. This provides evidence that Conté is not significantly slower than PEN for four of the five "pen-like" TRANSITIONS, and can be faster in some cases.

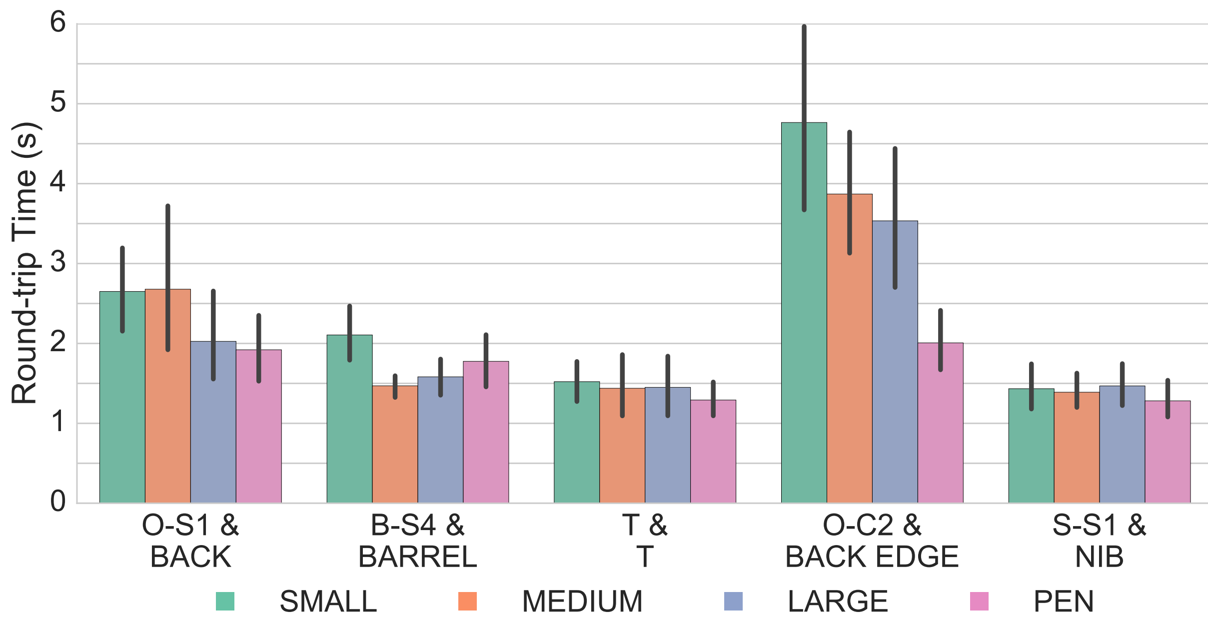


Figure 4.7: Round-trip time by "pen-like" TRANSITION and (SIZE + PEN)

4.8 Discussion

Vogel and Casiez suggested the ideal Conté size is an artist's Conté crayon like our small mock-up. We also expected that transition times would be lower for our smaller mock-up since the "distances" are smaller between contacts. Our results found the exact opposite. We believe the reason is because smaller tangible devices are more difficult to physically manipulate, and this had a larger impact on transition time than distance between contacts. An important design implication follows from the fastest performance of the large mockup: our conductive Conté device, the size of the large mockup, does not have any loss in manipulation performance.

It is interesting that within any size of Conté, transition time increases with group when considering it as a discrete measure of distance (in increasing order: same end, barrel, opposite end). This indicates that distance does impact transition time, just not as significantly as overall size. We found one exception across all sizes: transitions to the opposite end side are faster than transitions to an edge on the barrel. This strengthens the argument that grip impacts transition time more than distance. It can be difficult to hold an extruded rectangle along the barrel while maintaining enough control to use an edge. In contrast, sides provide more area to grip and require less fine control.

It is encouraging that round-trip times for individual transitions are similar across all three sizes. An optimal matching of commands to contact points on all three sizes could result in good command-contact pairings regardless of sizes. This would simplify a designer's task if faced with supporting a range of Conté sizes. Also, the large range of round-trip times for different transitions means an optimal command-contact mapping would not be arbitrary.

Perhaps most exciting, is that Conté was as good or better for four of five equivalent transitions with a pen. Some Conté sizes even out-performed the pen mock-up. There is little transition time lost with Conté, and given the increase in usable contacts, this seems like a worthwhile trade-off.

4.8.1 Limitations

Round-trip times were calculated from two consecutive correct transitions as a close representation of practised performance. It is possible that real expert users would be faster, however there is no indication that relative differences would change. We also limited our investigation to transitions originating from a corner, but Vogel and Casiez demonstrate interaction techniques using transitions between non-corner contacts. Our choice to focus on a corner follows from Conté as primarily a "pen-like device", so transitions from an "inking corner" are a realistic representation of use. Regardless, even accounting for symmetry, testing all contact pairs would have required participants to perform 193 transitions compared to the 74, which would take approximately 3 times our 1.5 hour experiment. Finally, only three sizes of extruded rectangles were tested. We chose our three sizes carefully, based on previous work and practical constraints like embedded hardware. Our results show you would certainly not want anything smaller, and likely do not need something larger for a real device. Other shapes, like prisms, could also be used for pen-like tangible devices like Conté, and could be tested in the future.

4.9 Summary

This chapter measures manipulation time, but another important aspect of mode-switching is how well people can remember command-contact pairings. We explore this next.

Chapter 5

Recall Study

Conté was created to facilitate easy mode-switching by combining inking and mode-switching into a single device. We saw in the previous chapter that users are able to manipulate a mock-up the size of our conductive Conté device faster than other sizes tested. In order to get a true sense of Conté's usability as a mode-switching device, we must also investigate whether people can learn locations of commands. Learning and recall of commands in spatial interfaces that use physical landmarks has been studied extensively in 2D [5, 18, 19, 29], but to the best of our knowledge has not been investigated in 3D. The goal of this experiment is to investigate command location learning and recall using our Conté prototype in comparison to a 2D spatial interface using a large and small command set, and evaluate if the game we adapted and on-screen guide we created are effective tools for teaching command-contact mappings. We expect small command sets to perform well with both techniques. We anticipate the 2D technique to perform better than the 3D technique with the large command set because it is rotationally stable. Conté combines inking and mode-switching into a single device while 2D spatial techniques are separate from input, and we believe the benefits of this outweigh the downsides of a small difference in recall.

5.1 Participants

We recruited 8 participants (mean age 26.5, SD = 5.6, 1 woman, 1 left-handed). The experiment is mixed-design. Two participants performed each of the four conditions. Participants received $20 for conditions that took two hours (large commands), and $10 for conditions that took one hour (small commands).

5.2 Apparatus

The experiment was performed on a Lenovo Yoga 2 touch-screen laptop (1.6 GHz i5 CPU with 8 GB RAM) running Windows 10. The laptop's 13.3" display has a resolution of 1920 $\times$ 1080 px and a density of 165 PPI. The laptop's screen was secured to a wooden board with velcro straps and the board was placed on a table. To prevent users from having to lean over the laptop's keyboard, the laptop was used upside down with its display flipped so that it appeared right-side up. An external keyboard was used to prevent users from having to lean over the laptop's screen (see Figure 5.1). The experiment code was written in Processing and ran on the same laptop as the Conté software. Our Conté device was used as well.

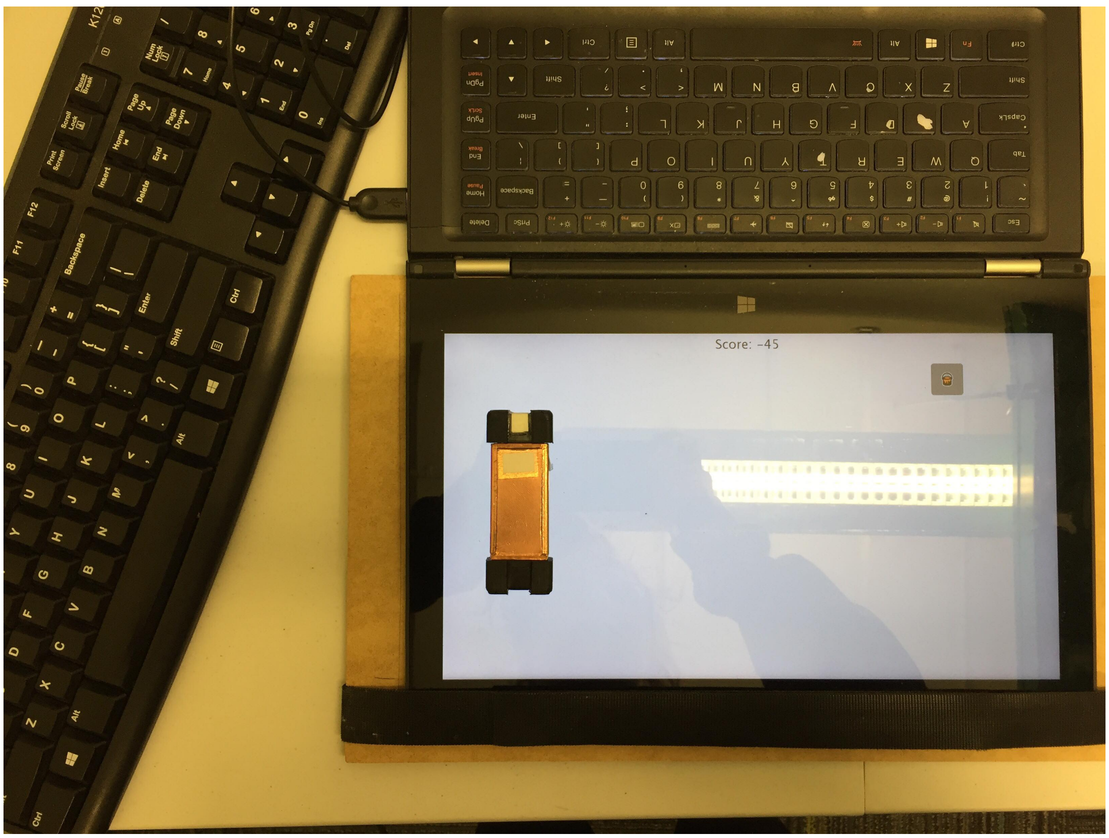


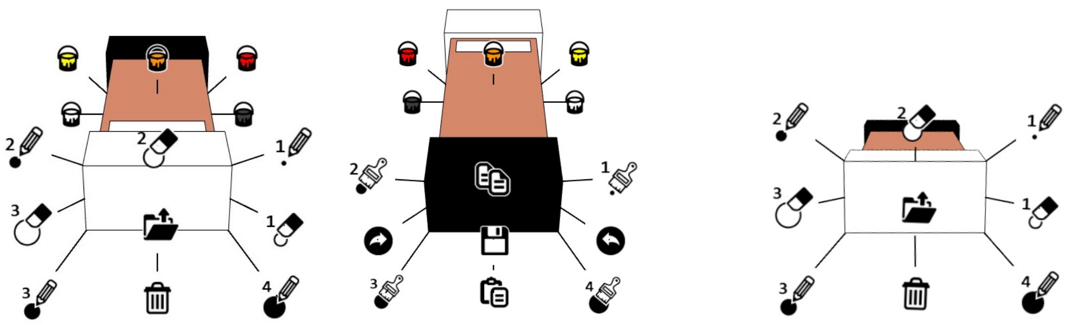
Figure 5.1: Laptop attached to wooden board with velcro. External keyboard attached.

5.3 Command Selection Techniques

We compare two selection TECHNIQUES: CONTÉ and GRID, each with two COMMAND SET SIZES: LARGE (all 26 commands), and SMALL (a 9-command subset).

5.3.1 Conté

The CONTÉ TECHNIQUE was performed using our conductive Conté device. For LARGE COMMANDS, a command was placed on each of the 26 Conté contacts. For SMALL COMMANDS, a command was placed on each nine contacts on *white end* (see Figure 5.2).



(a) (b)

Figure 5.2: Conté guides: (a) LARGE guide; (b) SMALL guide.

To facilitate novice to expert learning [14], we implemented a guide: a digital 3D model of Conté displaying a short line from each contact and an icon representing its corresponding command at the end of the line (see Figure 5.2). The on-screen model rotates with the physical prototype along all three axes. White marks on Conté are also shown on the 3D model (see Figure 5.3). Only icons closer to the user are displayed for better visibility. The guide is displayed while the ”m” key on the external keyboard is held, after a 500 ms delay. To mitigate yaw drift – a common problem in IMUs – pressing the ”n” key while the white end of Conté faces away from the user resets yaw and solves drift. The ”m” key was used because of the word ”menu” and participants’ non-dominant hand typically remained on the ”m” and ”n” keys during the experiment. Outside of an experiment, a Conté guide could be invoked by a gesture like shaking or tapping the side of Conté with a finger. The focus of this experiment is not on guide activation techniques so the keyboard is used to avoid confounds.

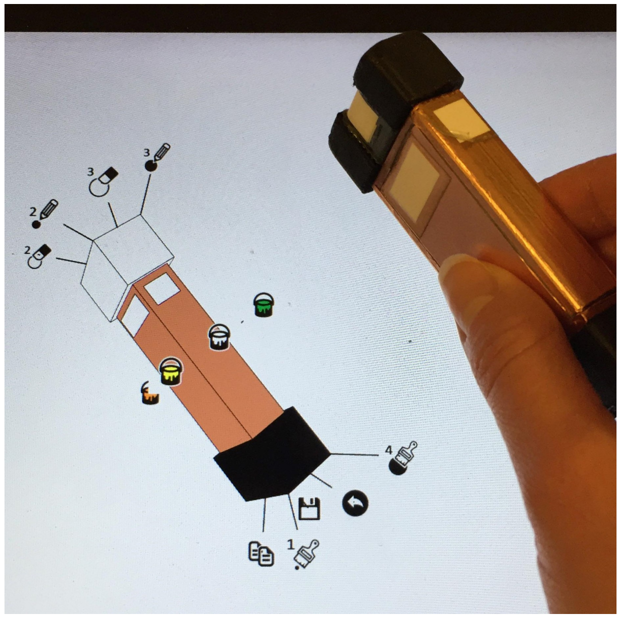


Figure 5.3: White marks replicated on on-screen guide.

There are two types of COMMANDS: TAP and DRAW. For commands that are drawing tools in real life, (4 pencil, 3 eraser, and 4 paint brush), the user must draw a stroke to execute the command. The style of the stroke is different for each DRAW COMMAND, and is used as feedback. All other commands are TAP COMMANDS (8 colours, delete, open, save, undo, redo, copy, paste). The user can tap anywhere on the screen and feedback (an icon corresponding to the command) is shown on screen for 2 seconds at the centroid of the Conté contact that was used to make the selection. We use two types of COMMANDS because Conté can be used for inking (represented by DRAW COMMANDS), and command-selection (represented by TAP COMMANDS).

Novice selection is performed by first viewing the guide and subsequently executing a command. A command cannot be executed while the guide is visible. Expert selection is performed by executing a command without viewing the guide. The physical action of command execution is identical whether novice or expert selection is used.

5.3.2 Grid Technique

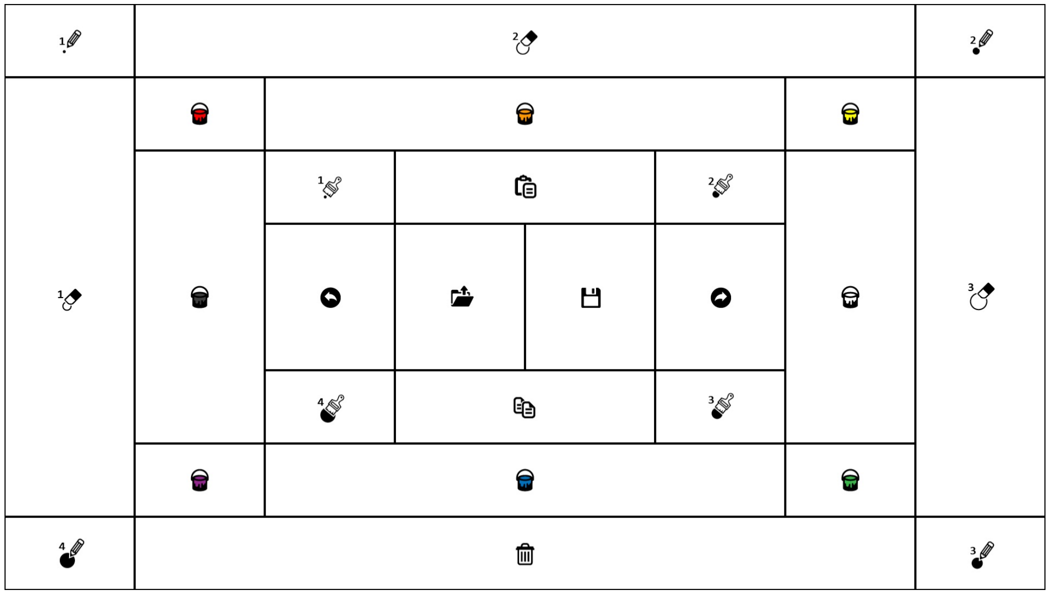
Conté is a spatial interface with commands located on physical landmarks (corners, edges, and sides) on the exterior of a 3D object. A grid interface uses of corners and edges on the perimeter of a touch screen as physical spatial landmarks. We compare against a grid interface because it is the closest 2D analog to our 3D interface.

The GRID TECHNIQUE is performed using a 2D spatial interface inspired by the multi-touch FastTap [6]. In FastTap, commands are located in an on-screen grid; one command in each rectangle. Successful FastTap grid sizes have ranges from 12-16 items ([6, 5, 18]). A 24 item FastTap interface was used in a longitudinal study of a drawing application and adoption of expert selection was poor [5]. Selection is performed by holding the rectangle in the bottom left-hand corner of the screen (invocation) and simultaneously tapping the rectangle with the desired command. After invocation, there is a short delay (150 ms - 250 ms in different versions of FastTap) and icons representing commands appear in the rectangles after invocation. Expert selection is performed by selecting a command before the timeout and novice selection by selecting a command after the timeout.

We depart slightly from a grid layout to replicate the organization of commands on Conté. For LARGE, four nested layers of rectangles are used. Beginning from the edges of the screen, the first layer corresponds to commands on corners and edges of *white end*. The second layer corresponds to commands on *barrel*. The third layer corresponds to commands on corners and edges of *black end*. The command on W-S1 (small side on *white end*) K-S1 (small side on *black end*) are in the fourth layer (see Figure 5.4 (top)). For SMALL, the first layer is used and the command on W-S1 is the only command in the second layer (see Figure 5.4 (bottom)).

In each of the first, second, and third layers, commands follow the same circular order as they do on Conté. In the first and third layer, commands on Conté corners are in the corners of the layer and commands on Conté edges are along the edges of the layer. In the second layer, commands on Conté sides are along the edges of the layer and commands on Conté edges are in corners of the layer. Note that copy and paste were unintentionally switched with each other, but since they only relate to each other this should not have an impact on learning.

Unlike CONTÉ, using GRID always takes two steps: selection and execution. Since GRID is only a selection method and not an input method, selection and execution must be performed in two separate steps. This is consistent with FastTap. Selection mode is active while the ”m” key is held on the external keyboard. Novice selection is performed by waiting 500 ms until icons representing each command are displayed in their corresponding rectangles; the user then taps the rectangle to select a command and releases the ”m” key. Novice delays in FastTap interfaces have ranged from 150 - 250 ms. We increase the novice selection delay time so that expert selections are possible on our larger screen and with our larger command set. Expert selection is similar to novice except that the rectangle is selected within the first 500 ms, before icons appear. In both cases, a blue outline appears around the selected rectangle while the ”m” key is held. Once a command is selected, it can be executed. Similar to the CONTÉ TECHNIQUE, the user strokes



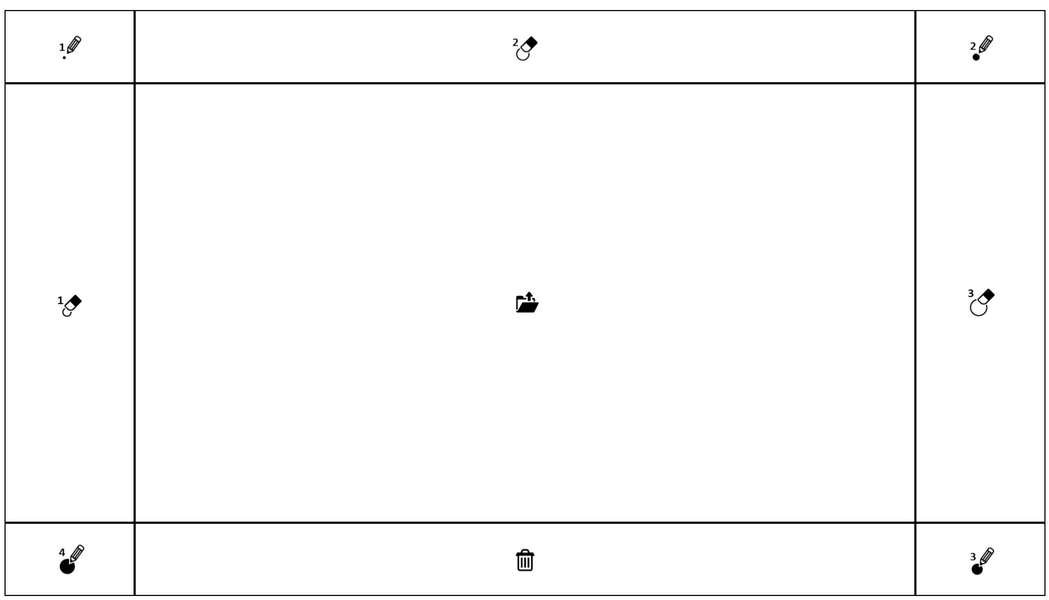


Figure 5.4: GRID guides: LARGE (top), and SMALL (bottom).

through the block for DRAW COMMANDS and taps anywhere on the screen for TAP COMMANDS. Feedback is the same as the CONTÉ TECHNIQUE. Since we are studying differences in 2D and 3D spatial interfaces with regards to learning and recall and not comparing differences between pen-like input and touch input, participants are only allowed to touch the screen with Conté, and the Conté software is still used for touch data even though its resulting classifications are not.

For the purposes of the experiment, the mapping of commands to contacts on Conté was fixed, as were command locations in grid.

5.4 Game Task

We base our task on Gutwin et al.'s Shape Slicer game [5] (see Figure 5.5). Shape Slicer is designed to facilitate novice to expert transition through increasing performance requirements. Lafreniere et al. [18] found that users who acquired an expert technique in their 2D grid-based FastTap menu while playing Shape Slicer maintained the expert technique in a relaxed drawing application.

For each trial, a single square with a command icon falls from the top of the screen, at a random horizontal location. These icons match those in CONTÉ and GRID guides. The participant must select the command before the square falls off the bottom of the screen. An error occurs when an incorrect command is selected or if the square falls off the bottom of the screen. For DRAW COMMANDS, an error also occurs if a stroke is too short, however this was not counted as an error in analysis. Participants were told to draw through the square for DRAW COMMANDS but this was not enforced programatically. A red border is displayed on the falling square for two seconds when an error occurs, and a green border, displayed for one second, indicates a correct selection (see Figure 5.6).

Gutwin et al.'s Shape Slicer game [5] rewarded participants with a score. One point per correct selection, and no penalty for errors, novice selections, or missed selections. Lafreniere et al. [18] did not use a score. To increase users' incentive to select correct commands and use expert selections, we integrate a fine-grained scoring system. For the duration of the game, the score is displayed at the top center of the screen. To deter guessing, 15 points are deducted for each incorrect selection in a trial up to a maximum of 45 points per trial. Once this maximum is achieved, no deductions are taken from the score for subsequent incorrect selections in the same trial. A square that falls off the bottom of the screen is treated as an error and 15 points are also deducted. Ten points are deducted for the first use of the guide in a trial. Subsequent uses in the same trial are not deducted since the participant is already making a novice selection. Note that this ten point deduction is less than the penalty for an error, so there is incentive to make a

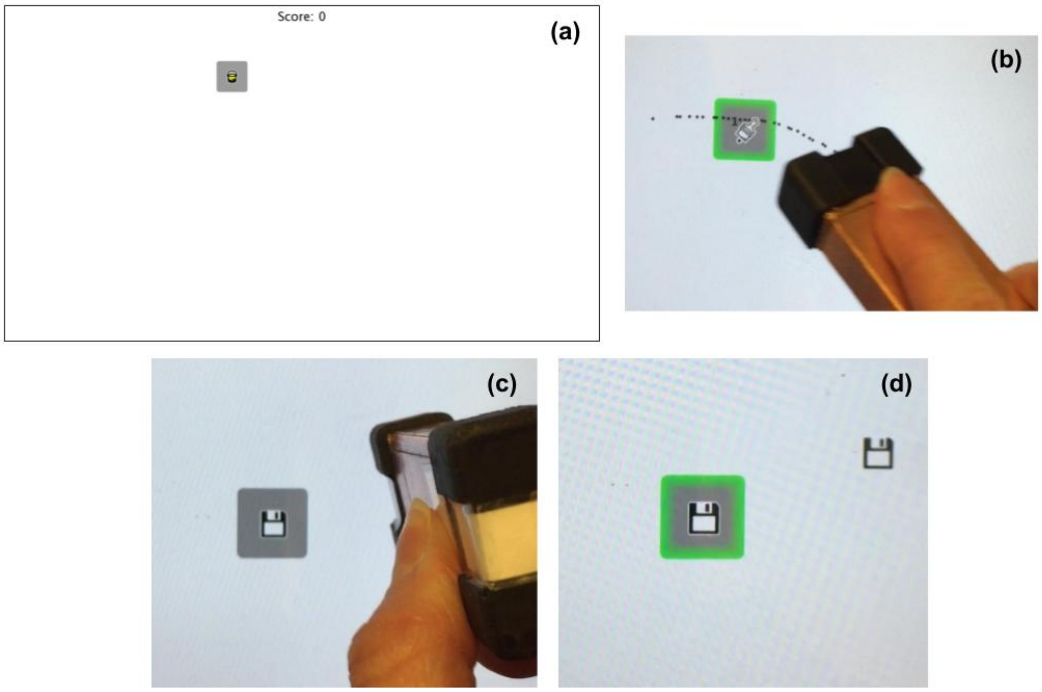


Figure 5.5: Game screenshots: (a) game screen; (b) draw command execution and feedback; (c) tap command execution; (d) tap command feedback.

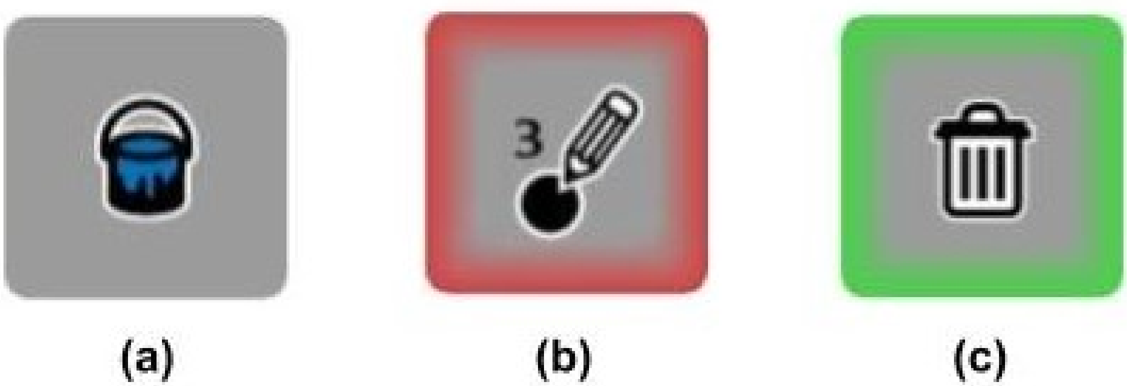


Figure 5.6: Falling squares: (a) normal; (b) error; (c) correct.

novice selection instead of guessing. Fifty point are awarded for each successful trial. This is slightly more than the maximum possible 45 point deduction for errors in a trial, so the net score still increases in a trial that ends in a correct selection even if multiple errors occur before. For each trial, we record time and location of: trial start, all errors, all openings and closings of the guide, guide selections for GRID TECHNIQUE, all correct selections, and trial end.

Retention tests are conducted using a special version of the game. In a retention trial, a single square with an icon for one command is displayed in the center of the screen. It does not move and there is no time limit. The guide is disabled and no feedback is displayed (DRAW COMMANDS do not draw anything, icons are not displayed for TAP COMMANDS, the square's outline does not change colour to indicate correctness, and there is no score). After a selection is made, the square disappears for one second and a new square with a different command's icon is displayed.

5.5 Design and Procedure

The experiment is mixed-design. Two participants completed each of the four combinations of TECHNIQUE and COMMAND SIZE. Each participant played a single game. A game consists of six stages. Each stage has a single speed: slow (20s), medium (10s), and fast (5s), these times are the number of seconds from the time the square is visible at the top of the screen until the time it fully exits the bottom of the screen. Speeds are all slower than Lafreniere et al. [18] due to our larger screen size, larger command set, and different TECHNIQUE. Stage order is: slow, medium, fast, medium, fast, medium. This is consistent with the most effective stage order for skill transfer found by Lafreniere et al.. Their rationale is that participants would experience some fast blocks early on but have the opportunity increase skill in subsequent medium blocks. Each stage consists of four blocks. Each block consists of one trial per command. Command order is random within a block. There is a one second delay between each trial in a single stage. Three retention tests were conducted after completion of the game. One immediately, one ten minutes later, and one approximately 24 hours later. A retention test consists of a single block.

Each participant was given brief instructions on the game and score, how to perform both novice and expert selection with their given TECHNIQUE, and execute commands with TAP or DRAW. Participants using CONTÉ were shown the white markings on Conté that align with white markings on the on-screen guide. They were also told to place Conté on the table with the white end facing away from them and press ”n” if they found the guide was out of sync. Participants using GRID were told to use a corner or short edge of Conté. Participants were informed about the retention tests. They were given the option to take a break between stages. For LARGE, participants were required to take a short break between each stage while the battery was changed. For SMALL, the battery was changed once after the third stage. Participants were not given any time to look at the guide or refresh their memory before retention tests. Participants completed the game, did the first retention tests, completed a post-experiment survey during a ten minute break, did the second retention tests, and completed the third retention test the following day.

In summary: SMALL: 9 commands × 4 blocks per stage × 6 stages = 216 experiment trials per participant. 9 commands × 3 retention tests = 27 retention trials per participant.
LARGE: 26 commands × 4 blocks per stage × 6 stages = 624 trials per participant. 26 commands × 3 retention tests = 78 retention trials per participant.

5.6 Results

We collected data from 2 participants per condition and consider an initial study to pilot an entire experiment. We plan to collect data from 8 participants per condition in total in the near future. Due to the small number of participants, we cannot run meaningful statistical tests and instead discuss trends observed in the results.

5.7 Novice vs. Expert Selection

Percentage of expert use was computed per participant for each block as the number of trials using expert selection in a block divided by the total number of trials in a block. In CONTÉ, there is an increasing trend in expert selection throughout the game and it is used exclusively by the end. Expert use of GRID with LARGE increases throughout the game as well although it is never completely adopted and GRID with SMALL reaches peak use at the beginning of the final fast stage and subsequently decreases (Figure 5.7).

To investigate this further, we compare rates of expert selection in the final stage. The final stage is a medium stage, so participants have time to use novice selection unlike fast stages. Expert selection rates in this stage are the closest approximation of what they would be in a relaxed task after the game. CONTÉ LARGE and CONTÉ SMALL both have 100% expert selections, following by GRID LARGE (mean 82%), then GRID SMALL (mean 17%) (Figure 5.8). Even at the top of the 95% confidence interval, GRID LARGE still has lower expert selection than both CONTÉ conditions. GRID SMALL has a large confidence interval suggesting expert selection may vary significantly between the two participants, however at the top of its 95% confidence interval, expert selection is still lower than the other three conditions. It is interesting to note that COMMAND SIZE appears to have a large impact on expert selection for GRID, but not for CONTÉ.

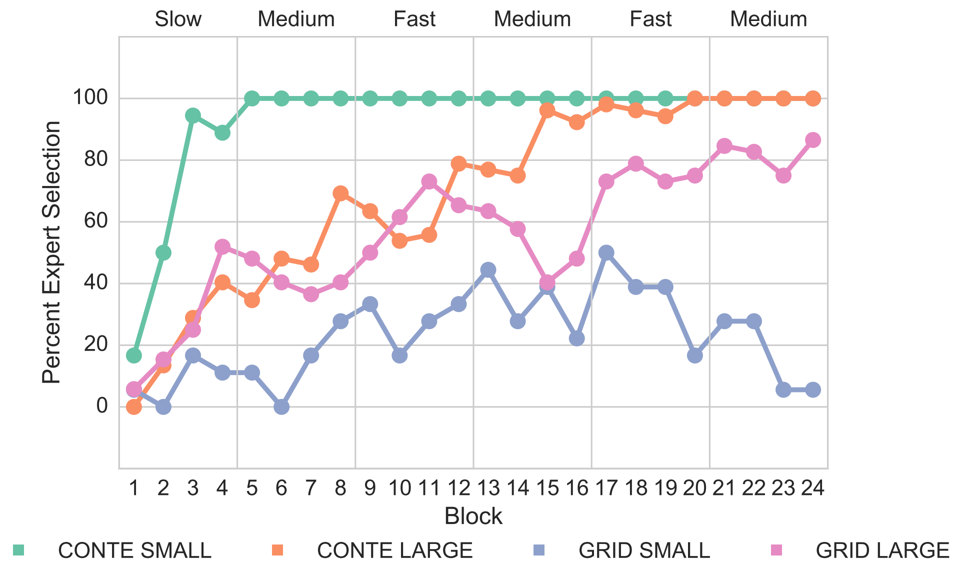


Figure 5.7: Percent expert selection (error bars omitted for readability).

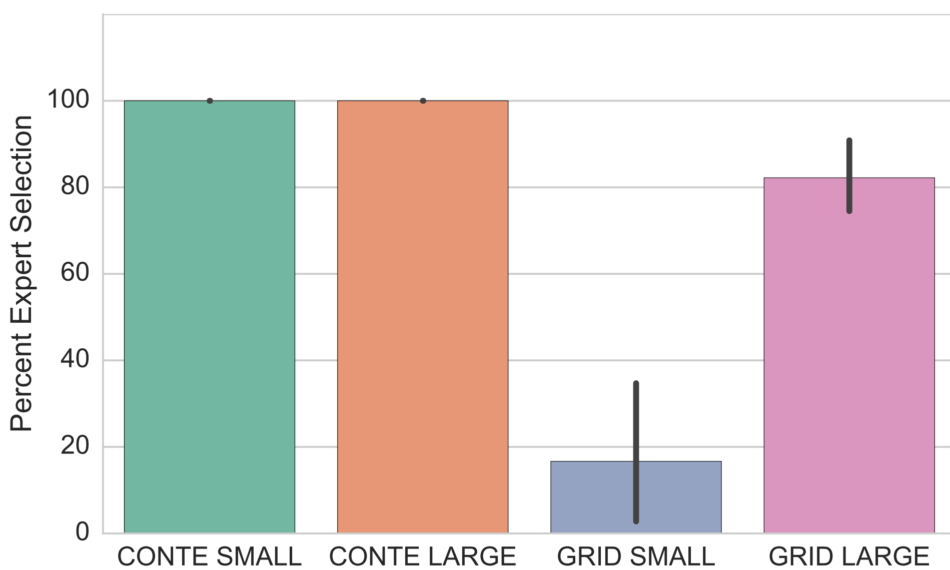


Figure 5.8: Percent expert selection final stage (all error bars 95% CI).

5.8 Errors

We compute each participant's percent error in a block as the number of trials that had errors errors divided by the total number of trials in the block. CONTÉ SMALL and both GRID conditions have low error rates for the duration of the game, have some spikes, but otherwise are below 20%. CONTÉ LARGE has higher error rates throughout the entire game, and has noticeably higher error rates during the fast stages (see Figure 5.9). During the experiment, one of the two participants who completed CONTÉ LARGE (P7) said they thought it was faster to guess than to use the guide. Consequently, this participant made many incorrect selections and error rates are higher for this condition.

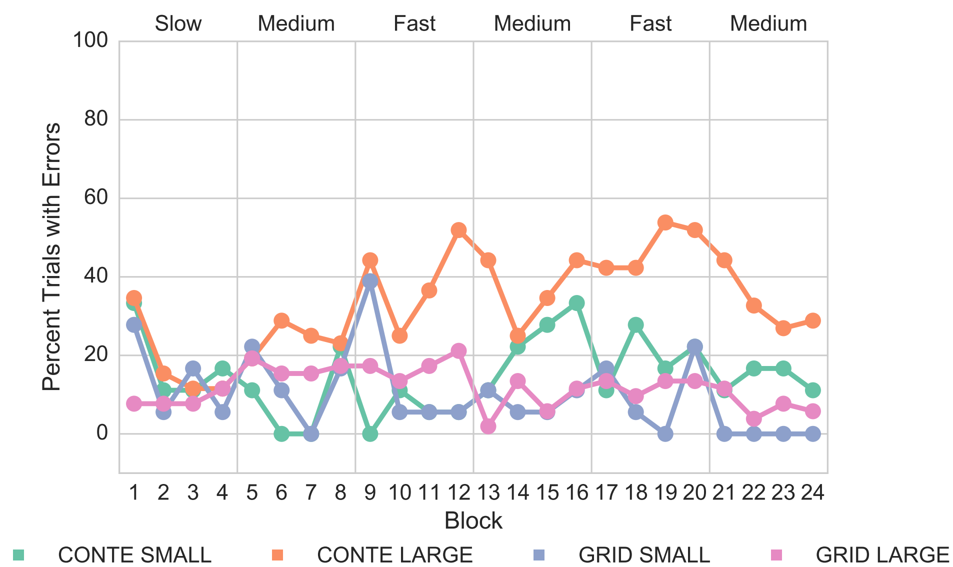


Figure 5.9: Percent trials with errors (error bars omitted for readability).

The final stage of the game is the best representation of practiced performance. We compare error rates in the final stage, and find GRID SMALL has the lowest error rate (mean 0%), following by GRID LARGE (mean 7.2%), then CONTÉ SMALL (mean 13.9%), and finally CONTÉ LARGE (mean 33.1%) (see Figure 5.10). The 95% confidence interval for GRID LARGE intersects with the 95% confidence interval for CONTÉ SMALL, and the 95% confidence interval for CONTÉ SMALL intersects with the 95% confidence interval for CONTÉ LARGE. For both TECHNIQUES, LARGE has more errors, but GRID LARGE has fewer errors than CONTÉ SMALL.

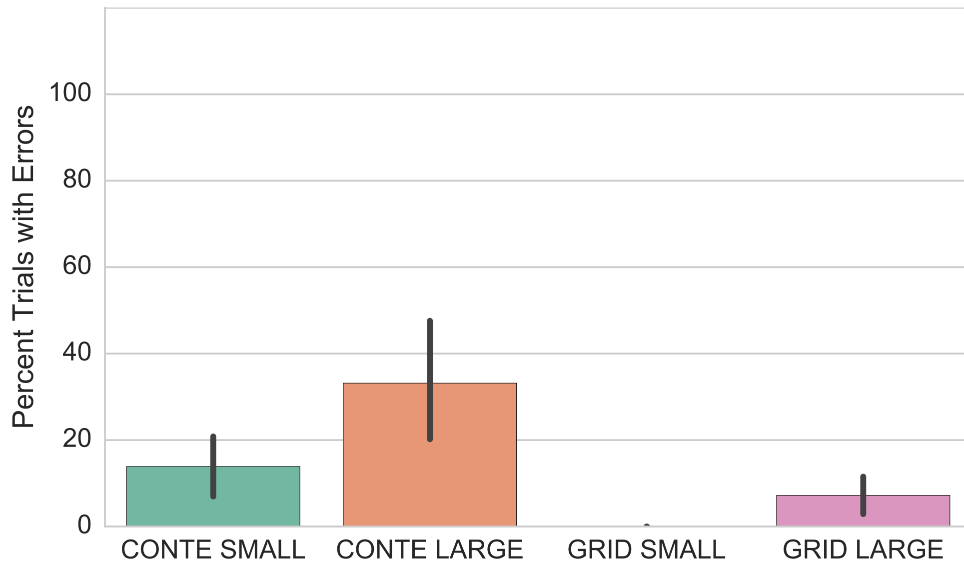


Figure 5.10: Percent trials with errors final stage (all error bars 95% CI).

5.9 Time

To investigate mean trial completion time, we only consider trials in which the correct command was selected, even after multiple errors. If the correct command was not selected in a trial, it ended at its pre-determined time limit. Novice and expert selection times both include the motor action to issue the command. Novice selection also includes visual search for the command's icon. Expert selection includes time to recall the command's location and visual search for the spatial landmark. Trial completion times were aggregated by block per participant using the mean. All conditions immediately decrease completion time in the initial slow stage, continue to decrease in the subsequent medium stage, then decrease in fast stages, and increase slightly in medium stages (see Figure 5.11). Maximum allowed completion times were slow (20 s), medium (10 s), fast (5 s).

In the final medium stage, CONTÉ SMALL was fastest (mean 2.88 s), followed by GRID LARGE (mean 3.65 s), then GRID SMALL (mean 3.71 s), and finally CONTÉ LARGE (mean 4.46 s). While means for GRID SMALL and GRID LARGE are very close, there is no overlap between 95% confidence intervals for CONTÉ SMALL and CONTÉ LARGE. This suggests COMMAND SIZE may have an impact on completion time for CONTÉ, but not GRID.

In order from fastest to slowest, the fastest trial for each condition was: CONTÉ SMALL 1.300 s, GRID LARGE 1.642 s, CONTÉ LARGE 1.749 s, and GRID SMALL 1.849 s.

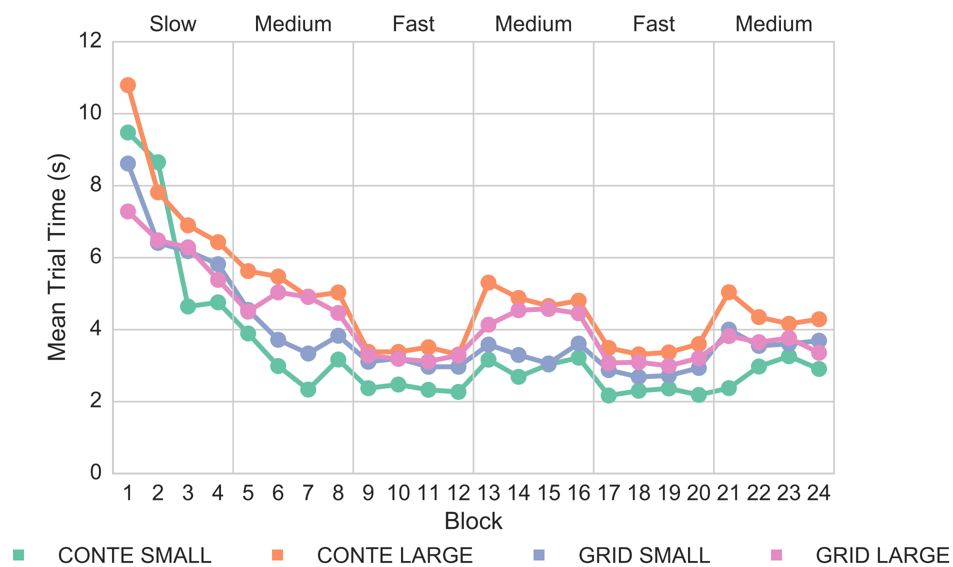


Figure 5.11: Mean trial completion time (s) (error bars omitted for readability).

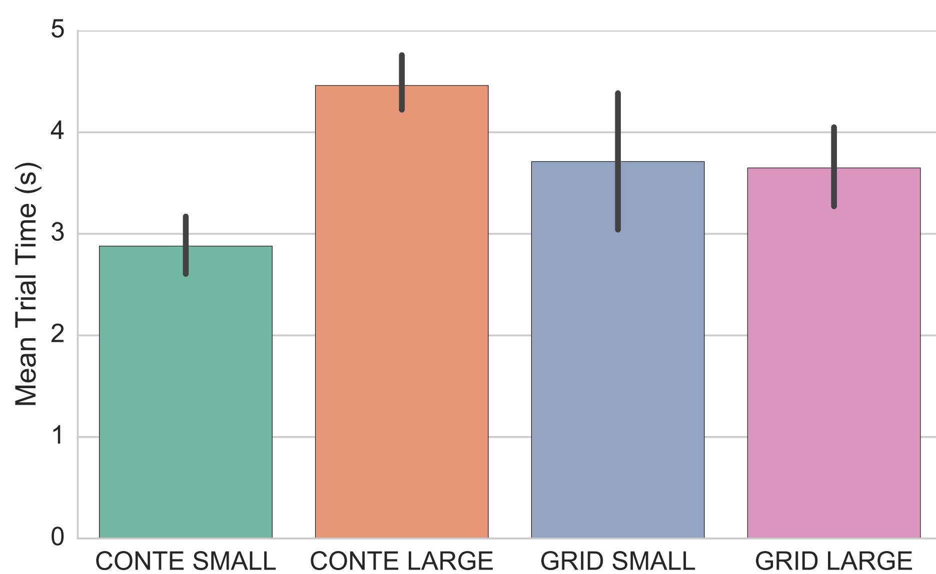


Figure 5.12: Mean trial completion time (s) final stage (all error bars 95% CI).

5.10 Retention

A retention test consists of one trial for each command in random order. Percent recall in a retention test is calculated as the number of correct selections divided by the total number of commands. To keep interaction consistent, participants were instructed to stroke through DRAW commands and tap for TAP commands in retention tests. The selection was only labelled as an error if the wrong command was used and was labelled correct for a DRAW command even if the stroke drawn was too short or did not intersect with the square.

For all retention tests, CONTÉ LARGE had the lowest mean percent recall (76.9 % 0 mins, 71.2 % 10 mins, 69.2 % 24 hours), followed by GRID LARGE (94.2 % 0 mins, 94.2 % 10 mins, 80.8 % 24 hours). For the 0 minute retention test GRID SMALL had the highest mean percent recall (100 %) followed by CONTÉ SMALL (94.4 %), and GRID SMALL and CONTÉ SMALL had the highest (and identical) mean percent recall for 10 minute (94.4 %) and 24 hour (100 %) retention tests (see Figure 5.13).

SMALL recall appears to have increased after 24 hours. This is certainly true for GRID, but CONTÉ recall may appear lower than it actually is due to classification errors in 0 minute tests and 10 minute tests. Note that incorrect classifications typically misclassify a contact as its adjacent contact. From observing retention tests, participants usually remember the *type* of contact used for a command but sometimes forget *which* contact. For example, participants tend to remember pencils are on corners but forget which corner pencil 1 is on. Contacts of the same *type* are never adjacent, thus classification errors rarely result in false positives. The 100% recall at 24 hours is likely correct as classification errors should not have resulted in higher recall, but it is possible that they caused it to be lower in earlier retention tests.

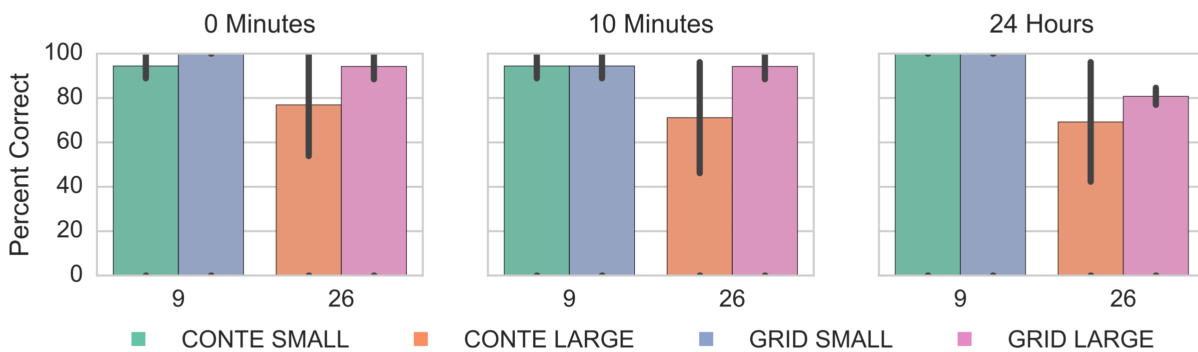


Figure 5.13: Retention tests (all error bars 95% CI).

To gain further insight, we consider participants' individual recall studies (see Figure 5.14).

GRID LARGE outperformed CONTÉ LARGE when considering mean performance however, there is a large difference in performance between individual participants for CONTÉ LARGE. Participant 7 has low scores (53.8% at 0 mins, 46.2% at 10 mins, 42.3% at 24 hours), while participant 14 has near perfect scores (100% 0 mins, 96.2% 10 mins, 96.2% 24 hours). In the 24 hour test, participant 14 outperformed both participant 6 (76.9%) and 12 (84.6%) who completed the GRID LARGE condition. Individual data for SMALL is similar to what the aggregate data showed.

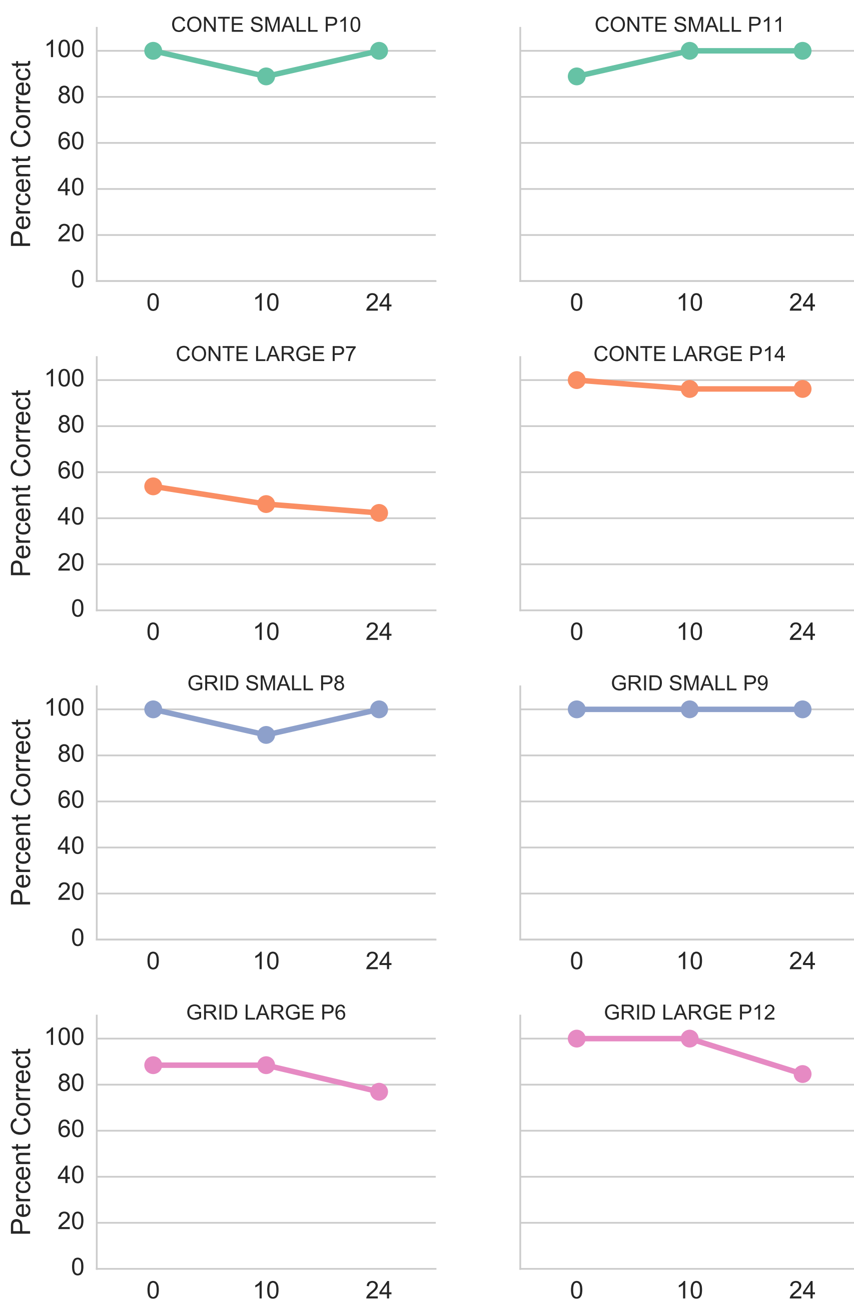


Figure 5.14: Retention tests by participant.

5.11 Discussion and Limitations

5.11.1 Recall and Implications of Digital Guide

The retention tests are perhaps the most important and interesting results. Recall of the small command set is almost perfect when using both Conté and grid selection. However for the large command set, the grid technique appears to outperform Conté, at least when considering the aggregate results. A closer look at individual participant performance reveals a drastic difference between the two participants who used Conté with all 26 commands (P7 and P14). Given that we only collected data from two participants per condition, it is difficult to say whether P7 or P14 is an outlier, or perhaps a larger sample size would have bimodal results. It is clear though that all 26 command locations on Conté can be learned, but that it can also be challenging.

While performing the experiment, P7 verbalized frustration and noted that the mental rotation required to use the guide was difficult. We believe the guide can be improved. Since the guide shows commands that are *closer* to the user and hides those that are further away, when the user finds the command they would like to perform, the corresponding Conté contact is facing *towards* them. They then have to rotate Conté so that the contact is facing the screen in order to use it. That P7 had poor recall, and P14 had great recall, could be a result of P14 being better at mental rotation than P7. Cutting the guide in half and showing commands that are *further* from the user would remove this problem and we believe this could significantly improve recall of all 26 Conté commands. We are working on this and will test this new guide in the near future.

When using Conté, recall of the 9 command subset was close to 100% on all tests. This shows that people can learn command locations on Conté. Grid recall was high in the 26 command case, which shows that people can learn 26 commands in a spatial interface. Combining these two ideas, we are fairly confident that it is possible to learn all command locations on Conté, and are hopeful that we will be able to demonstrate this with changes to the guide.

It is also interesting that Conté performance remains relatively constant between the three retention tests while grid performance decreases drastically in the 24 hour test. We did not test beyond 24 hours but it would be interesting to conduct a seven day test as well and see if these trends continue.

5.11.2 Expert Selections, Errors, and Time

Mean selection time in all medium blocks in all four conditions is below six seconds. This is slower than fast blocks (below four seconds), but the time limit in fast blocks is five seconds while the time limit in medium blocks is ten seconds. Gutwin et al. found that time constraints imposed by a game in a hard task encourage users to make expert selections and adoption of an expert technique is significantly higher than in a passive activity like a drawing application [5]. It is possible that the game creates an environment in which users self-impose more severe time constraints than the game itself, and this further encourages expert use.

In the final medium stage, expert selection using grid with the small command set is very low. This condition seemed quite easy to perform and participants did not seem rushed. Perhaps it was so easy that participants did not feel the need for quick expert selection. It is also interesting to note that recall was 100% at the 24 hour mark for this condition. So participants learned command locations despite primarily using novice selection. Furthermore, the mean final stage time is very similar between both grid conditions. Selection took the same amount of time even though users predominately made novice selections when given the smaller command set.

When using the large command set, grid was faster than Conté, but it also had better recall. Comparing both techniques using the small command set, they had similar recall and Conté was faster than grid. It is possible that this trend may hold when using the large command set as well and if we improve Conté recall, it may become faster than grid. However, when the full command set is used, two consecutive Conté selections may require an end-to-end transition, a barrel-to-end transition, or an end-to-barrel transition. In small commands, all consecutive Conté selections are within the same end, and our manipulation study shows that these are fastest.

5.11.3 Limitations

Our Conté prototype was built for research and is not perfect. Users would occasionally perform correct commands and the system would classify them as errors. As a result, users did not always believe the system when they were told they made a mistake and would keep trying to use the same contact to see if the error was a system error. The imperfect system had implications on the study, but we need to study the system before we invest time in making it perfect. In theory our classifier is 99.16% accurate however this is not the case in practice. We captured video footage of Conté retention tests, can obtain ground-truth classifications from it, and approximate the accuracy of our system in practice. We leave this for future work.

Another factor in keeping our study realistic is time. Recall using the small command set on Conté was better than large, and it would have been interesting to compare more command set sizes and get a more granular understanding of the relationship between command-set size and recall. We could have investigated using both ends of Conté but not the barrel, just the barrel, or even start with one end and slowly increase the number of commands. The large command studies took two hours and we did not think it was feasible to run longer studies or more variations. Instead, we studied the full command set and the smallest non-arbitrary subset and are hopeful that our proposed changes to the guide will improve recall for Conté using the large command set. Finally, we only study two participants per condition. We treat this as a pilot and gain very useful insight. We leave expanding our sample size to immediate future work and are excited to see the results.

Chapter 6

Demos

Three demo applications were created to show different ways Conté can be used in a realistic scenario. They were created by a developer [1] who used our Conté device and Conté software to create these applications in approximately two months. In addition to showing interesting ways Conté can used, these demos also validate that our device and system can be used by engineers.

Demos were created on an iPad Pro. Since Processing cannot run on an iPad, our Conté software was used on a MacBook Pro.

6.1 PDF Annotations

Annotating a PDF requires precise input commonly performed with a stylus and frequent mode-switching (see Figure 6.1). It is a natural application of Conté.

- Corners are used as a pen or highlighter. A long press reveals a menu that can be used to change the colour or style.
- Small edges scroll the page in any direction.
- Medium edges zoom in and out. Moving forward zooms in, backward zooms out.
- Long edges display a ruler. The ruler remains on screen when Conté is released so it can be used as a guide for drawing a straight line with any corner. The ruler disappears after a line is drawn.

[1] Undergraduate Research Assistant Jean-Baptiste Beau.

- Ends stamp. White end stamps a signature, black end stamps the current date.

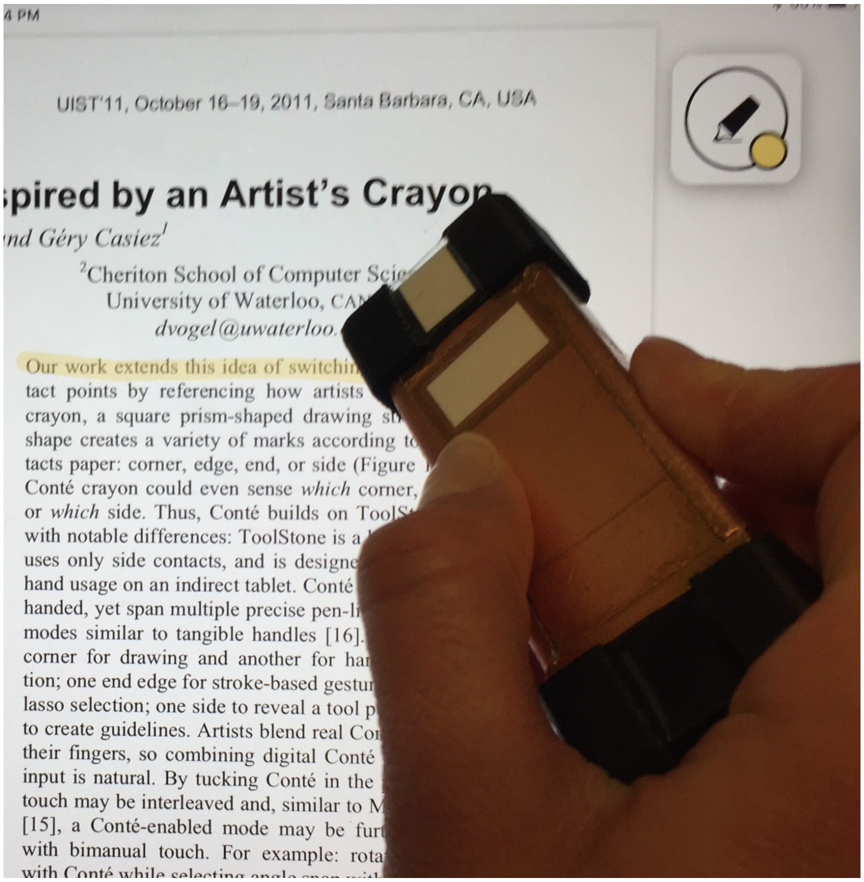


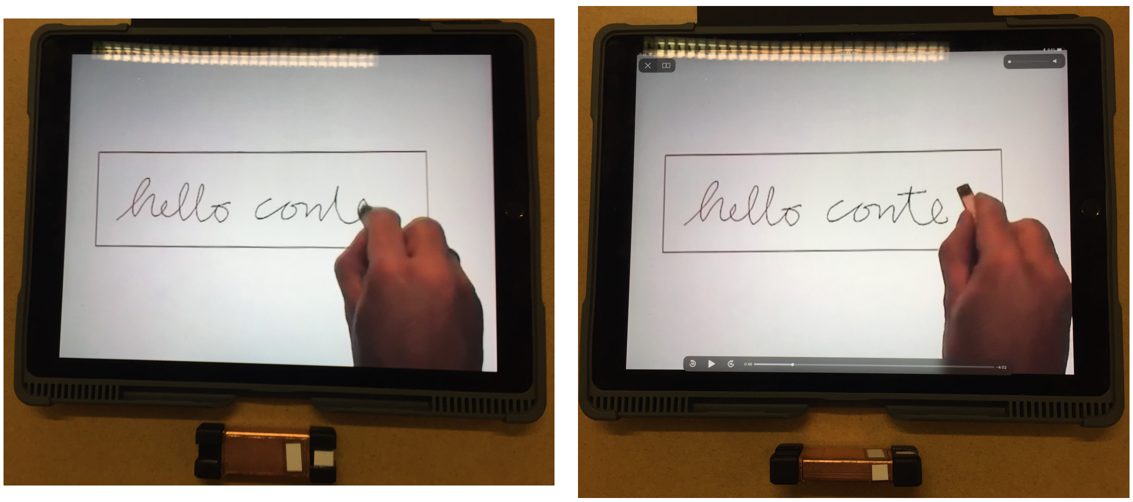
Figure 6.1: PDF annotations.

6.2 Video Player

We use Conté to control a video player (see Figure 6.2).

- Laying Conté flat on a large side plays a video.
- Turning onto a medium side pauses the video.

- Tilting left decreases volume.
- Titling right increases volume.
- Moving quickly to the right fasts forward.
- Moving quickly to the left rewinds.



(a) (b)

Figure 6.2: Video player: (a) play; (b) pause.

6.3 Bounce Game

We create a simple game in which a ball bounces around the screen and Conté is used to control it. By dividing the screen in half, two Conté devices can be used at the same time and users can pass the ball back and forth. Native iOS touch, not the Conté software, is used to determine the location of both Conté(s). When a large side of Conté is placed on the screen, its four points form a rectangle and the ball bounces off this rectangle (see Figure 6.3).

Events from Conté software are used for *special events*. In the two Conté case, only one Conté has internal hardware and uses Conté software. *Special events* are:

- Corners draw curves that the ball bounces off of. Curves disappears when Conté is removed from the screen.

- White end spawns a new ball.
- Black end adds a small square obstacle.

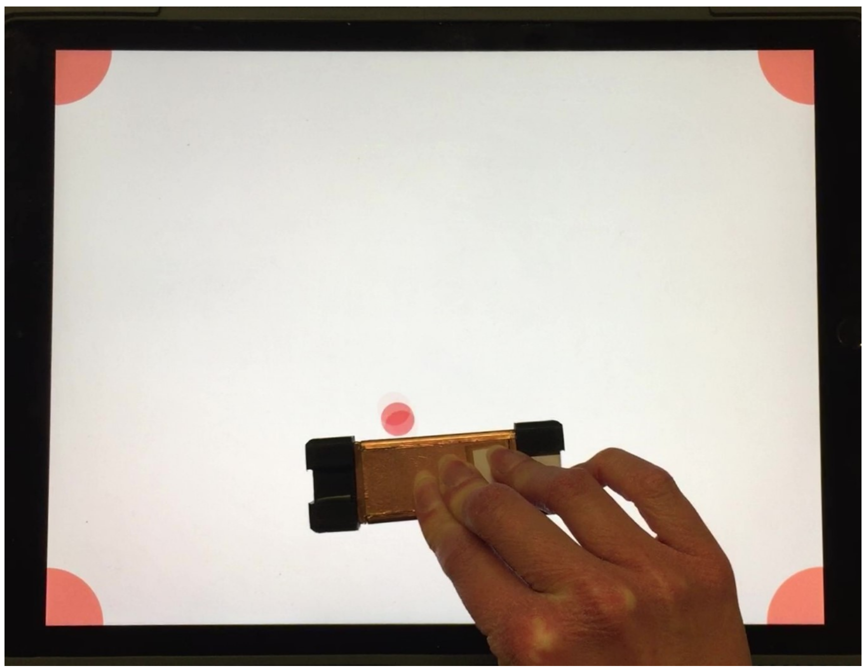


Figure 6.3: Ball bounces off Conté.

Chapter 7

Conclusion

We realize Vogel and Casiez's vision of a cuboid shaped multimodal input device with 26 contacts that can be uniquely identified and used for different commands. This was accomplished by 3D modelling and 3D printing a conductive case, embedding a wifi-capable microcontroller, IMU, and LiPo battery, training a decision tree on IMU data for contact classification, and combining this with touch data from a capacitive display. Future work could create a pipeline that facilitates custom, user-designed tangible input devices. Users could mold and 3D scan a clay model to create a digital 3D model, and contacts could be identified, sharpened, and assigned a unique range of angles using computer vision techniques. Similar to our conductive Conté device, internal sensing hardware could determine orientation, which in turn could determine which command to execute.

Our work compares contact-to-contact transition times for three different sizes of Conté: *small*, the size of an artist's Conté crayon; *medium*, the size of Vogel and Casiez's prototype; and *large*, the size of our conductive version. We find transition times decrease with larger device size, and increase with greater ”distance” between contacts. A comparison with a pen mock-up shows that Conté transition times are comparable. To expand on these results, a comparison of non-rectangular shapes, mock-ups with tactile feedback like bumps, and the effect of adjusting the centre of mass would be informative.

Using our conductive Conté prototype, we compare learning and recall of commands located on physical landmarks on the exterior of a 3D cuboid to a 2D grid-like interface. We test a large and small command set and find expert selection is adopted in a higher proportion of trials when using the Conté selection technique, recall is similar for both selection techniques when using a small command set, and high recall is possible for both techniques when using a large command set. This was a small pilot study with two participants per condition. The next step is to run the full study with eight participants per condition. Future work could enable users to choose command-contact mappings and study the impact of user-placed commands on recall.

Our conductive implementation realizes Vogel and Casiez's design, a manipulation study validates its size, and a recall study shows it is possible to learn command locations. Vogel and Casiez expressed high hopes for the impact of Conté, comparing its potential effect on HCI to the new style of art influenced by the invention of the artists' Conté crayon. We are certainly closer to achieving this goal.

References

[1] Michel Beaudouin-Lafon. Instrumental interaction: an interaction model for designing post-WIMP user interfaces. In *Proc. CHI*, CHI '00, pages 446–453, 2000.

[2] Xiaojun Bi, Tomer Moscovich, Gonzalo Ramos, Ravin Balakrishnan, and Ken Hinckley. An exploration of pen rolling for pen-based interaction. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 191–200. ACM, 2008.

[3] Bruno Dumas, Denis Lalanne, and Sharon Oviatt. *Multimodal Interfaces: A Survey of Principles, Models and Frameworks*, pages 3–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[4] George W Fitzmaurice, Hiroshi Ishii, and William AS Buxton. Bricks: laying the foundations for graspable user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442–449. ACM Press/Addison-Wesley Publishing Co., 1995.

[5] Carl Gutwin, Andy Cockburn, and Benjamin Lafreniere. Testing the rehearsal hypothesis with two fasttap interfaces. In *Proceedings of the 41st Graphics Interface Conference*, GI '15, pages 223–231, Toronto, Ont., Canada, Canada, 2015. Canadian Information Processing Society.

[6] Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. Faster command selection on tablets with fasttap. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 2617–2626, New York, NY, USA, 2014. ACM.

[7] Ken Hinckley, François Guimbretière, Patrick Baudisch, Raman Sarin, Maneesh Agrawala, and Edward Cutrell. The springboard: multiple modes in one spring-loaded control. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 181–190, Montréal, Québec, Canada, 2006. ACM.

[8] Ken Hinckley, Michel Pahud, Hrvoje Benko, Pourang Irani, Francois Guimbretiere, Marcel Gavriliu, Xiang'Anthony' Chen, Fabrice Matulic, William Buxton, and Andrew Wilson. Sensing techniques for tablet+ stylus interaction. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 605–614. ACM, 2014.

[9] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. Pen+ touch= new tools. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 27–36. ACM, 2010.

[10] Apple Inc. Apple pencil. http://www.apple.com/apple-pencil/, 2 2017.

[11] FiftyThree Inc. Pencil stylus for ipad, ipad pro & iphone. https://www.fiftythree.com/pencil, 2 2017.

[12] Adobe Systems Incorporated. Adobe ink & slide. https://helpx.adobe.com/ink-and-slide.html, 2 2017.

[13] Sven Kratz, Tilo Westermann, Michael Rohs, and Georg Essl. Capwidgets: Tangile widgets versus multi-touch controls on mobile devices. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pages 1351–1356, New York, NY, USA, 2011. ACM.

[14] Gordon Kurtenbach. *The design and evaluation of marking menus.* University of Toronto, 1993.

[15] Gordon Kurtenbach and William Buxton. The limits of expert performance using hierarchic marking menus. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 482–487. ACM, 1993.

[16] Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 258–264, New York, NY, USA, 1994. ACM.

[17] Gordon Kurtenbach, Abigail J Sellen, and William AS Buxton. An empirical evaluation of some articulatory and cognitive aspects of marking menus. *Human-Computer Interaction*, 8(1):1–23, 1993.

[18] Benjamin Lafreniere, Carl Gutwin, and Andy Cockburn. Investigating the post-training persistence of expert interaction techniques. *ACM Trans. Comput.-Hum. Interact.*, 24(4):29:1–29:46, August 2017.

[19] Benjamin Lafreniere, Carl Gutwin, Andy Cockburn, and Tovi Grossman. Faster command selection on touchscreen watches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4663–4674, New York, NY, USA, 2016. ACM.

[20] Yang Li, Ken Hinckley, Zhiwei Guan, and James Landay. Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces. *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 461–470, 2005.

[21] Microsoft. Microsoft surface pen. https://www.microsoft.com/en-us/surface/accessories/surface-pen?icid=MSCOMSearchTopResultSurfacePen, 9 2017.

[22] Halla B Olafsdottir, Theophanis Tsandilas, and Caroline Appert. Prospective motor control on tabletops: planning grasp for multitouch interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2893–2902. ACM, 2014.

[23] Sharon Oviatt. Advances in robust multimodal interface design. *IEEE computer graphics and applications*, 23(5):62–68, 2003.

[24] Gary Perelman, Marcos Serrano, Mathieu Raynal, Celia Picard, Mustapha Derras, and Emmanuel Dubois. The roly-poly mouse: Designing a rolling input device unifying 2d and 3d interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 327–336. ACM, 2015.

[25] Simon T. Perrault, Eric Lecolinet, Yoann Pascal Bourse, Shengdong Zhao, and Yves Guiard. Physical loci: Leveraging spatial, object and semantic memory for command selection. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 299–308, New York, NY, USA, 2015. ACM.

[26] Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. Pressure widgets. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 487–494. ACM, 2004.

[27] Jef Raskin. *The humane interface: new directions for designing interactive systems*. Addison-Wesley Professional, 2000.

[28] Jun Rekimoto and Eduardo Sciammarella. Toolstone: effective use of the physical manipulation vocabularies of input devices. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 109–117. ACM, 2000.

[29] Katherine Schramm, Carl Gutwin, and Andy Cockburn. Supporting transitions to expertise in hidden toolbars. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4687–4698, New York, NY, USA, 2016. ACM.

[30] Abigail Sellen, Gordon Kurtenbach, and William A. S. Buxton. The Prevention of Mode Errors Through Sensory Feedback. *Human-Computer Interaction*, 7(2):141–164, 1992.

[31] Hyunyoung Song, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. s. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1323–1332. ACM, 2011.

[32] Minghui Sun, Xiang Cao, Hyunyoung Song, Shahram Izadi, Hrvoje Benko, Francois Guimbretiere, Xiangshi Ren, and Ken Hinckley. Enhancing naturalness of pen-and-tablet drawing through context sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 83–86. ACM, 2011.

[33] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. Experimental analysis of mode switching techniques in touch-based user interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3267–3280. ACM, 2017.

[34] Larry Tesler. The smalltalk environment. *Byte*, 6(8), 1981.

[35] Feng Tian, Lishuang Xu, Hongan Wang, Xiaolong Zhang, Yuanyuan Liu, Vidya Setlur, and Guozhong Dai. Tilt menu: using the 3d orientation information of pen devices to extend the selection capability of pen-based user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1371–1380. ACM, 2008.

[36] Md. Sami Uddin, Carl Gutwin, and Andy Cockburn. The effects of artificial landmarks on learning and performance in spatial-memory interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3843–3855, New York, NY, USA, 2017. ACM.

[37] Md. Sami Uddin, Carl Gutwin, and Benjamin Lafreniere. Handmark menus: Rapid command selection and large command sets on multi-touch displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5836–5848, New York, NY, USA, 2016. ACM.

[38] Kristof Van Laerhoven, Nicolas Villar, Albrecht Schmidt, Gerd Kortuem, and Hans Gellersen. Using an autonomous cube for basic navigation and input. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, ICMI '03, pages 203–210, New York, NY, USA, 2003. ACM.

[39] Daniel Vogel and Géry Casiez. Conté: multimodal input inspired by an artist's crayon. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 357–366. ACM, 2011.

[40] Shumin Zhai and Per-Ola Kristensson. Shorthand writing on stylus keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 97–104, New York, NY, USA, 2003. ACM.