Unimanual Pen+Touch Input Using Variations of Precision Grip Postures

Drini Cami[1], **Fabrice Matulic**[2,1], **Richard G. Calland**[2], **Brian Vogel**[2], **Daniel Vogel**[1]
[1]School of Computer Science, University of Waterloo, [2]Preferred Networks Inc
{dcami, dvogel}@uwaterloo.ca, {fmatulic, calland, vogel}@preferred.jp

ABSTRACT
We introduce a new pen input space by forming postures with the same hand that also grips the pen while writing, drawing, or selecting. The postures contact the multitouch surface around the pen to enable detection without special sensors. A formative study investigates the effectiveness, accuracy, and comfort of 33 candidate postures in controlled tasks. The results indicate a useful subset of postures. Using raw capacitive sensor data captured in the study, a convolutional neural network is trained to recognize 10 postures in real time. This recognizer is used to create application demonstrations for pen-based document annotation and vector drawing. A small usability study shows the approach is feasible.

Author Keywords
pen input; touch input; interaction techniques

INTRODUCTION
Pen interaction makes drawing and writing natural and precise, but current applications still require frequent use of graphical user interface (GUI) buttons, menus, and widgets for actions like switching direct manipulation tools (e.g. selecting, inking, or highlighting), setting attributes (e.g. stroke colour or thickness), and issuing commands (e.g. using contextual menus or pen gestures). However, GUI menus and toolbars take up space and can be error prone to use [36], so increasing the pen input space to reduce GUI use is an important goal.

Researchers have proposed and evaluated many pen input techniques including pen-only methods like special stroke shapes [13, 18], barrel buttons [18] and manipulating the pen in detectable ways [33]. When a pen is combined with multitouch (typically called "pen+touch"), touches with the non-dominant hand have been used for gestures [21], postures [41], or to indicate object context [15, 26] and change the mode of the pen held in the other hand. Our work combines pen input and multitouch differently: We use the tablet touch sensor to detect user-controllable hand postures while that same hand grips the pen to write, draw, or manipulate graphical objects.

UIST '18, October 14–17, 2018, Berlin, Germany

ISBN 978-1-4503-5948-1/18/10...$15.00
DOI: https://doi.org/10.1145/3242587.3242652

Our idea stems from insights into how people hold writing tools. Most adults use a variation of a dynamic tripod grip [31, 28, 6] where the thumb and index fingers work in opposition, and a third finger (typically the middle) provides extra stabilization. This is a type of precision grip [19] which balances requirements for a firm hold of the pen with the ability for its independent manipulation [25]. The ideal writing posture rests part of the hand or fingers on the writing surface [31], but people can control a pen with the hand in the air, especially for less precise pen input tasks [39, 20]. A detailed examination of hand postures while using digital pens also showed diversity of grips [37]. These observations suggest that there is room for people to modify their hand posture, in terms of how individual fingers or hand contact the surface, while still maintaining a precise grip with the index and thumb. However, there was an open question whether intentional touches can be made with the same hand while using the pen, and how such touches can be effectively used for input.

Our proposed *unimanual pen+touch* input space is created by detecting when users consciously adjust how their hand posture contacts a surface while maintaining a precise pen grip. The palm can touch with its side, heel, or just float; the index and thumb can slide down to touch the surface beside the pen tip, and the middle, ring, and pinky fingers can contact the surface outside or inside the palm area. These combinations suggest 324 theoretical postures, but most are impractical. We identify a candidate set of 33 postures to evaluate.

We evaluate these postures in terms of pen control and subjective preference using a representative set of controlled tasks with 15 people. Our results show there is a large subset of practical postures. During the evaluation, we also logged all touch input data, including frame images capturing the raw capacitive signal. We use this to train a deep neural network to recognize 10 postures in real time with 91.4% average accuracy (96% best case). Using the recognizer, we illustrate

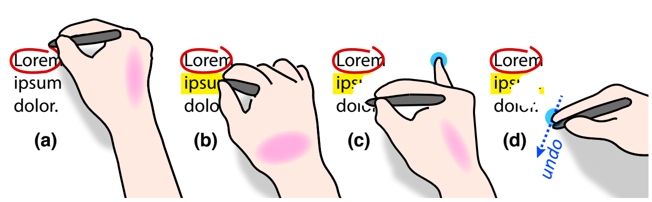

Figure 1: Varying pen grip posture to change input modes: (a) resting side of palm writes; (b) resting heel of palm highlights; (c) contacting extended pinky erases; (d) touch index beside pen for gesture commands. Pink and blue regions show where palm and fingers contact the surface.

how the postures can be applied to document annotation and vector drawing, two common pen applications with frequent mode switching. Our prototype is designed as a system service that runs in the background to inject commands into real applications using a configurable mapping of postures to short-cut keys and other actions or widgets. This enables the technique to trigger tools, make attribute selections, and invoke commands. A final study uses the applications to validate the usability and usefulness of the interaction space and interactive recognizer performance.

Our contribution is a new pen input space validated in terms of basic usability, usefulness, and feasibility. Since our unimanual techniques do not involve the non-dominant hand, they are compatible with bimanual pen+touch techniques, and with some limitations, also work with pen-only mode switching methods such as using an eraser button or pressure. Furthermore, unimanual pen+touch likely requires less screen space than bimanual methods, thus making it more suitable for small tablets and perhaps even smartphones. The objective is not to replace existing mode-switching techniques, but to propose a pen input space that complements and extends previous methods. Our design space exploration and recognition method show the unimanual pen+touch concept is feasible, and sets the stage for future comparative studies.

RELATED WORK

Mode-switching and command activation are critical to interaction, and many pen input techniques have been proposed for these tasks. However, using different postures performed while maintaining and using the standard tripod precision pen grip posture has not been previously explored.

Pen Input Techniques

Given how critical mode-switching is, it is not be surprising that researchers have examined it in detail [18], and proposed techniques and approaches to make mode-switching, and pen input in general, more powerful.

One family of techniques can be thought of as bringing toolbars closer to the pen. These include Tracking Menus [8], Springboard [14], and the Trailing Widget [9]. These still require screen space, but reduce movement time. Another family of techniques use gestures to hide or eliminate toolbars, such as Marking Menus [17], Hover Widgets [11], or the Scriboli delimiter [13]. Some are fast and some are slow, and since a gesture is defined by movement, it can be hard to smoothly differentiate it from movement for direct manipulation. Regardless of possible issues with these families of techniques, our method is compatible with them. For example, a Springboard, Tracking Menu, or Marking Menu could be shown only when using a certain hand posture since most gestures are independent of the hand posture used to grip the pen.

A third family of techniques exploits other pen input channels, such as pressure [27], rolling the pen barrel [2], tilting [43]. A fourth family of techniques changes modes based on how the pen is manipulated. This is as simple as inverting the pen to use the "eraser" end or pressing a barrel button [18], to bending the pen shaft [7], and creating a new pen-like device with 26 unique ways to contact the drawing surface [37]. Our technique may be less compatible with these techniques since they require non-standard pen manipulations.

Related to the last set is Song et al.'s grip detection pen [33]. By wrapping a capacitive sensor around the barrel and adding motion sensors, the pen can detect how it is held (e.g. a power grip) or how it is manipulated (e.g. shaken, pointed). Our input space is different: we do not sense how the pen is gripped, we look for ways to vary the hand posture around a standard precision grip; we only modify behaviour when the pen is used as a pen; and our technique requires no special hardware.

Combining Pen and Touch

With the rise of multitouch input, multiple ways to combine pen and touch have been proposed. Yee's early experiments [44] followed by Brandl et al.'s [3] showed how bimanual pen and touch can work as two independent input sources. Wu et al. [41] used the shape of the non-dominant hand on the surface to set a mode for non-dominant hand pen input. Matulic et al. [21] expanded this idea to using non-dominant hand touch gestures to set the pen mode, and performed experimental studies of its effectiveness [20].

Hinckley et al.'s Pen+Touch interaction vocabulary combines non-dominant hand touch with dominant hand pen in a more integrated and context-sensitive way. Their principle is "the pen writes, touch manipulates, and the combination of pen + touch yields new tools". Combined pen and touch actions trigger a mode-switch based on the context of a graphical object. For example, dragging the pen off a photo held by the other hand triggers a copy mode. Pfeuffer et al. [26] show how the pen+touch concept can be applied to mobile tablets with smaller screens using only the non-dominant thumb. Since our technique only uses a single hand, it is compatible with bimanual pen+touch techniques.

The only truly unimanual pen+touch technique we are aware of are two demonstrations with Conté [37], an unconventional pen-like device. In one example, laying the pen-like device flat on the surface enables a mouse mode with 'clicks' enabled by same-hand touch input. Another example controls guideline placement using the thumb touches while the same hand holds the device on the surface. Conté is not a conventional pen, and neither of these techniques explores same hand touch patterns while holding a pen in a precision grip.

Our work combines pen input and multitouch differently from previous pen+touch research: We use the tablet touch sensor to detect user-controllable hand postures while the same hand grips the pen and performs input actions like writing, drawing, or object manipulation. This potentially creates a very large input space, which we detail and discuss next.

UNIMANUAL PEN+TOUCH INPUT SPACE

Our focus is on the interaction space created when standard pen input is combined with touch input performed at the same time with the same hand that holds the pen. This *simultaneous* unimanual pen+touch, is different from alternating between pen and touch input with the same hand, such as tucking the pen in the palm when touching the surface (called "palming" [15, 37]). Finally, we focus on interactions where the pen is manipulated with a precision grip. This is different from using nearby touches while the device lays on the surface [37].

To make implementation practical, we constrain postures to those identifiable by the pattern created by surface contacts, an approach demonstrated for touch-only input [22]. Although additional pen grip postures could be recognized with an instrumented pen [33], or cameras capturing the hand above the surface, we define postures detectable on current touch sensitive devices. One way to think about this approach is to use touch data otherwise discarded by palm rejection [30]: Before running a palm rejection pipeline, check to see if a unimanual posture is recognized. If so, use the posture as input then reject all associated touches. Otherwise, simply process with standard palm rejection pipeline.

Input Degrees of Freedom and Notation

Although precision pen grips often use part of the middle finger for a third stabilizing part of the grip (forming a "tripod"), not all adults use a third finger [31] and we found it is possible to maintain a precision grip with only the index and thumb by using the index finger side for stabilization. We test the validity of this in the controlled study to follow.

To describe and reason about possible hand postures achievable as variations of the tripod grip, we consider six degrees of freedom defined as the types of touch contact made by the palm and the five fingers (Figure 2):

- *Palm Contact (Side, Heel, Float)*: While writing, the palm can contact the surface on its side, near the wrist (heel), or float with no contact. This creates 3 variations.
- *Grip Finger Contact (Thumb, Index)* $\times$ *(touching, not touching)*: The two primary tripod grip fingers, the index and thumb, can be independently slid down the barrel of the pen to touch the surface immediately beside the pen nib. Two fingers with two independent states create 4 variations.
- *Non-Grip Finger Contact (Middle, Ring, Pinky)* $\times$ *(In, Out, not touching)*: The other three fingers can independently touch the surface when extended outside the hand or curled inside the hand, or not touch the surface at all. Three fingers and three states create 27 variations.

Theoretically, this allows for 324 possible postures ($3 \times 4 \times 27$) with at least a basic level of precision since the tripod grip remains minimally altered.

To make referring to postures more concise in text, figures, and tables, we introduce standard notation. In written text, postures are named as a compound set of words: the palm contact type is always given (*Float, Side, Heel*); if the grip finger is named (*Index, Thumb*), then it is touching; and the name of the non-grip fingers are given with the postfix *In* or *Out* when touching (e.g. *MiddleIn, MiddleOut, ...*). For example: *Side-Index-PinkyOut* means the palm is contacting on the side, the index is touching just beside the nib, and the pinky is touching outside the main hand contact. In addition to this long form, we also use a condensed notation of initial letters: `S`, `H`, or `F` for palm state; `T` and `I` if grip a finger is touching; `M`, `R`, or `P` if the non-grip finger is touching outside the palm; `m`, `r`, or `p` if the non-grip finger is touching inside the palm. If a finger is not touching, a dash - is used. For example, the condensed form of *Side-Index-PinkyOut* is (`S-I--P`). Figure 2 provides more examples of long and short notation.

Reduced Set of Candidate Postures

Although this large set of postures might be physically achievable, many are clearly uncomfortable or difficult to perform due to individual flexibility, dexterity, and hand anatomy. A 3-person pilot tested all finger states across all palm states using a simplified version of the experiment protocol described in the following section. This led to a set of rules: (1) In general, grip finger and non-grip finger states should not be combined. If an index or thumb finger is touching, then the middle, ring, and pinky must not be touching, and vice versa. (2) The ring finger should move with the pinky or middle finger, since independent ring finger movement is difficult. (3) Splitting non-grip finger positions to be both *Out* and *In* should be avoided in most cases.

Using these rules, we reduced the 324 possible postures to 33 candidates for further investigation. The first 30 candidates are created by combining the 10 specific grip and non-grip finger states (Table 1 top) with all three palm states. The remaining 3 candidates are special postures that include specific palm states (Table 1 bottom). *Float-MiddleOut-RingIn*: Having the middle and ring finger spread is generally uncomfortable, but less difficult with the floating palm state. *Float-MiddleOut-RingOut*: We found this comfortable with floating palm, but difficult with other palm states. *Side-Thumb-Index-MiddleOut-RingOut-PinkyOut*: This was one case where combining grip fingers and non-grip seemed feasible.

Finger States Combined with all Palm States	
no fingers touching (`*-----`)	*Index* (`*-I---`)
Thumb (`*T----`)	*Thumb-Index* (`*TI---`)
PinkyIn (`*----p`)	*PinkyOut* (`*----P`)
MiddleOut (`*--M--`)	*RingIn-PinkyIn* (`*---rp`)
MiddleOut-RingOut-PinkyOut (`*--MRP`)	*RingOut-PinkyOut* (`*---RP`)
Additional Postures using Specific Palm States	
Float-MiddleOut-RingIn (`F--Mr-`)	
Float-MiddleOut-RingOut (`F--MR-`)	
Side-Thumb-Index-MiddleOut-RingOut-PinkyOut (`STIMRP`)	

Table 1: Reduced Set of 33 Candidate Postures: the 10 finger states in the top are each combined with 3 palm states to create 30 different postures; the 3 postures in the bottom include specific palm states.

EXPERIMENT: POSTURE SUITABILITY

The primary goal of this experiment is to evaluate the candidate set of postures in terms of subjective preference and pen control. We do this by asking participants to complete a set of synthetic, but representative pen input tasks using each posture, during which we measure accuracy relative to a target stimulus as well as movement time. We then ask for a posture preference rating that considers comfort and control. The results are used to create design guidelines for unimanual pen+touch postures for specific types of interactions in applications. An additional goal of this experiment is to collect data to train a posture recognizer using machine learning.

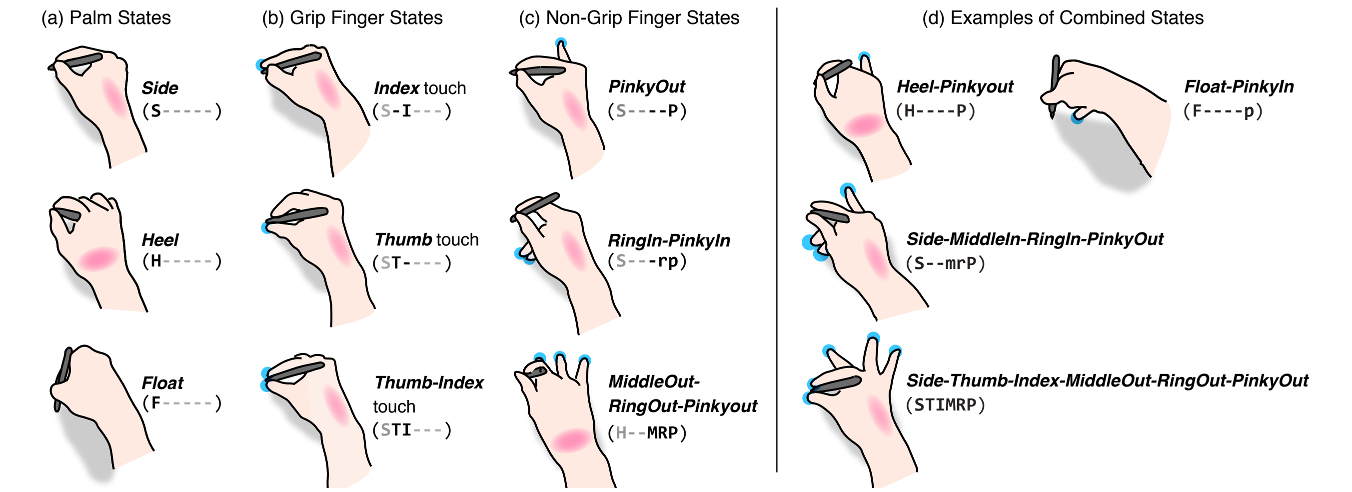

Figure 2: Input space examples and notation: (a) side, heel, or floating palm contact; (c) touching pen grip fingers to the surface near the pen tip; (b) extending and touching non-grip fingers outside or inside the palm; (d) examples of complete postures. The dark pink regions show where the palm contacts the surface and cyan circles show where fingers contact the surface.

Participants

We recruited 12 right-handed participants, ages 20 to 36, of which 5 were female. The right-handed requirement reduced variance due to handedness (we also ran 3 left-handed participants which we discuss later). Participants were recruited using on-campus flyers and word-of-mouth, and received $20 for successful completion of the study.

Apparatus

A Wacom Cintiq 22HD Touch pen tablet (1920×1080 px, 47.5×26.7 cm, 4.04 px-per-mm) was connected to an Intel NUC (Windows 10, Core i7 3.50 GHz 8GB RAM) to run the C# (.NET) application. Care was taken so computation and logging did not introduce any noticeable latency.

The Wintab API provided logging of pen tip coordinates, pen hover distance, pen pressure, and pen orientation at 140 Hz. The Wacom Feel™ Multi-Touch API provided raw capacitive data as 122×70 px grayscale bitmaps as well as 'finger' input events as an array of (max 10) touch ellipsoids at 100 Hz. The Wacom API did not send raw capacitive data when only the palm was touching the screen. As a workaround, we placed a capacitive presence in the top left corner to simulate a finger.

Tasks

We designed a set of generic pen input tasks indirectly derived from those originally suggested by Buxton [4]. The tasks are categorized as *constrained* or *unconstrained*, based on how restricted the pen movements are for task completion. The accompanying video also provides task demonstrations.

Constrained

The constrained tasks (Figure 3) simulated different atomic patterns of pen interaction with the aim of getting quantitative data on accuracy. Each task is presented as a pattern of grey 'dots' (tapping tasks) or 'paths' (tracing tasks) rendered on a black background. All dots are 4mm in diameter and all paths are 4mm thick. A green dot indicates the next dot to tap or the start of the next stroke. Paths also had a red 'cap' to indicate the end of the stroke.

Since one objective is to measure how accurate taps or strokes are compared to visual targets, the 4mm size functions only as a stimulus, not a strict boundary. Liberal acceptance thresholds ensured participants tapped or stroked on the expected target. Any tap less than 10mm from the edge of the current dot was accepted. For lines, any stroke that began within 10mm from the start, ended less than 15mm from the end, and had a length within 33% of the line length was accepted.

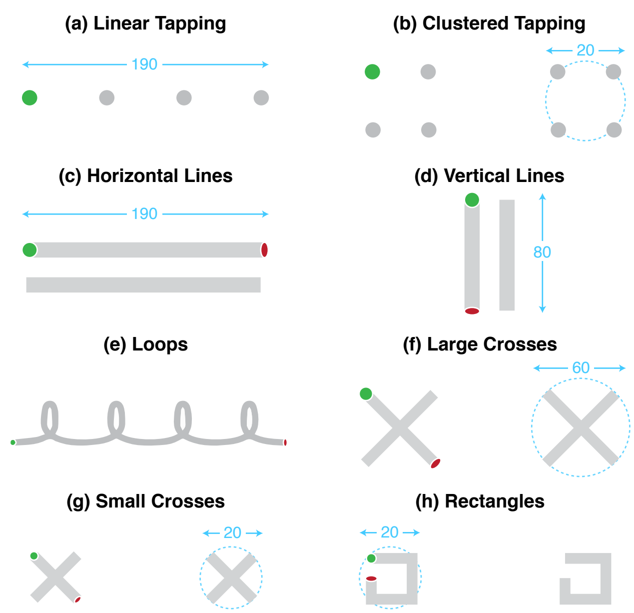

Figure 3: Constrained Tasks (see text). Colours altered for print.

The seven constrained tasks are:

Linear Tapping: Tapping left-to-right on 4 dots evenly spaced along a 19 cm horizontal line. Represents short tasks like tapping buttons with larger hand movement between.

Clustered Tapping: Tapping on two clusters of 4 dots spaced 15cm apart. Represents pushing a sequence of buttons on a menu or tool palette.

Horizontal Lines: Stroking two 19cm horizontal lines in both directions. Represents long strokes requiring large hand movement, like dragging an object across the screen.

Vertical Lines: Stroking along two 8cm vertical straight lines, in both directions. Represents long strokes requiring large hand movement, similar to the above.

Loops: Tracing a path with 4 loops, left-to-right, with each loop being 3cm tall, and the total path being 19cm wide. Represents long non-linear movements, like drawing or writing.

Big Crosses: Tracing two diagonal paths forming a cross, with both paths fitting in a bounding 6cm diameter circle, spaced 7cm apart. The top-left stroke was completed first. Represents larger off-axis pen movements with small amounts of palm motion, such a large stroke-based menu or manipulating objects (e.g. scaling or translating).

Small Crosses: As above, but the two diagonal crosses fit in a bounding 2cm diameter circle and were 15cm apart. Represents small pen tip motions with no palm movement, such as gestures or fine object manipulation.

Rectangles: Tracing two rectangular paths that fit in bounding 2cm diameter circles, 15cm apart. The direction is clockwise starting at the top-left. Similar to small crosses, but requires sharp pen tip direction changes.

Unconstrained

The unconstrained tasks (Figure 4) represent more common, integrative pen motions. These tasks were accepted as complete when the user pressed a "done" button.

Drawing: Copying a smiley face presented on the left side of the display into a 73×73 mm square. The experimenter monitored the participant to ensure they drew all parts (head, eyes, mouth, nose, ears). The same image was used across all postures for direct comparison.

Writing: Writing “important” on a 24cm baseline. The word was chosen as common 9-letter word with a good variety of letters and typographic diversity.

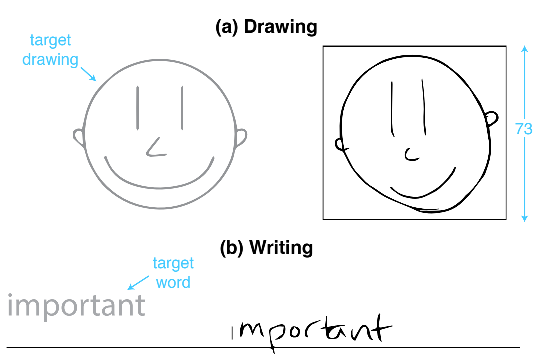

Figure 4: Unconstrained Tasks (see text). Colours altered for print.

Postures

We evaluate 35 postures; all 33 candidate postures in Table 1 (10 finger states for each of the 3 palm contact states) plus 3 other specific postures. In addition, we use two postures as upper and lower baselines to normalize the range of the subjective ratings across participants. We choose *Normal* (however the participant held the pen naturally) as an upper bound, and *Side-Thumb-Index-MiddleIn-PinkyIn* as the lower bound, as it received the lowest rating in the pilot test among the postures which all participants were able to complete.

Design and Protocol

The primary independent variable is POSTURE with 33 categorical levels representing each posture to test. To make the experiment easier for participants, we present all postures using each palm contact as three GROUPS: 10 heel, 11 side, and 12 floating. The order of the 3 GROUPS was determined using a balanced Latin square. Within each GROUP, the order of POSTURE was randomized. At least a 30 second break was required between sections, and the participant could stop and relax or stretch their hand in between any task. The two baseline postures were performed at the start of the experiment.

For each POSTURE, the participant was first presented with a training section lasting typically 20 to 140s. This began with the experimenter describing and demonstrating the posture as the participant practised on an empty drawing canvas. This was followed by a subset of the constrained tasks.

The 8 constrained tasks were presented first in random order, followed by the 2 unconstrained tasks, also in random order. The directions of the horizontal and vertical lines were also randomized for each participant (i.e. some drew left-to-right first, some drew right-to-left first). The same random orders were used for each participant across all tasks and postures to make the sequence predictable and to reduce unnecessary mental effort. Once all the tasks were completed, the participant was asked to consider the comfort and control of the posture for the tasks and provided a single preference score from 1 to 7 (step 0.5). We considered asking for separate ratings for metrics like fatigue and complexity, but our pilots suggested this was too much in an already long study. The entire experiment took under 2 hours.

Results

Given the large number of posture conditions, we interpret the results based on visual inspection of the trends evident in graphs of 6 key metrics (Figure 5).

Preference

For each posture, the participant rated the comfort and control of the posture as a single subjective preference score from 1 to 7 (most preferred) with step size 0.5. Note this is not a Likert-type scale, but a continuous interval measure. As explained above, two postures served as upper and lower baselines to help normalize this subjective score (**N** and **STIm-p**).

Examining the pattern of preference by posture (Figure 5a), we see a clear preference for the side palm (**S-----**) and floating palm (**F-----**) with scores approaching the upper baseline normal posture. For side palm, postures with the pinky out

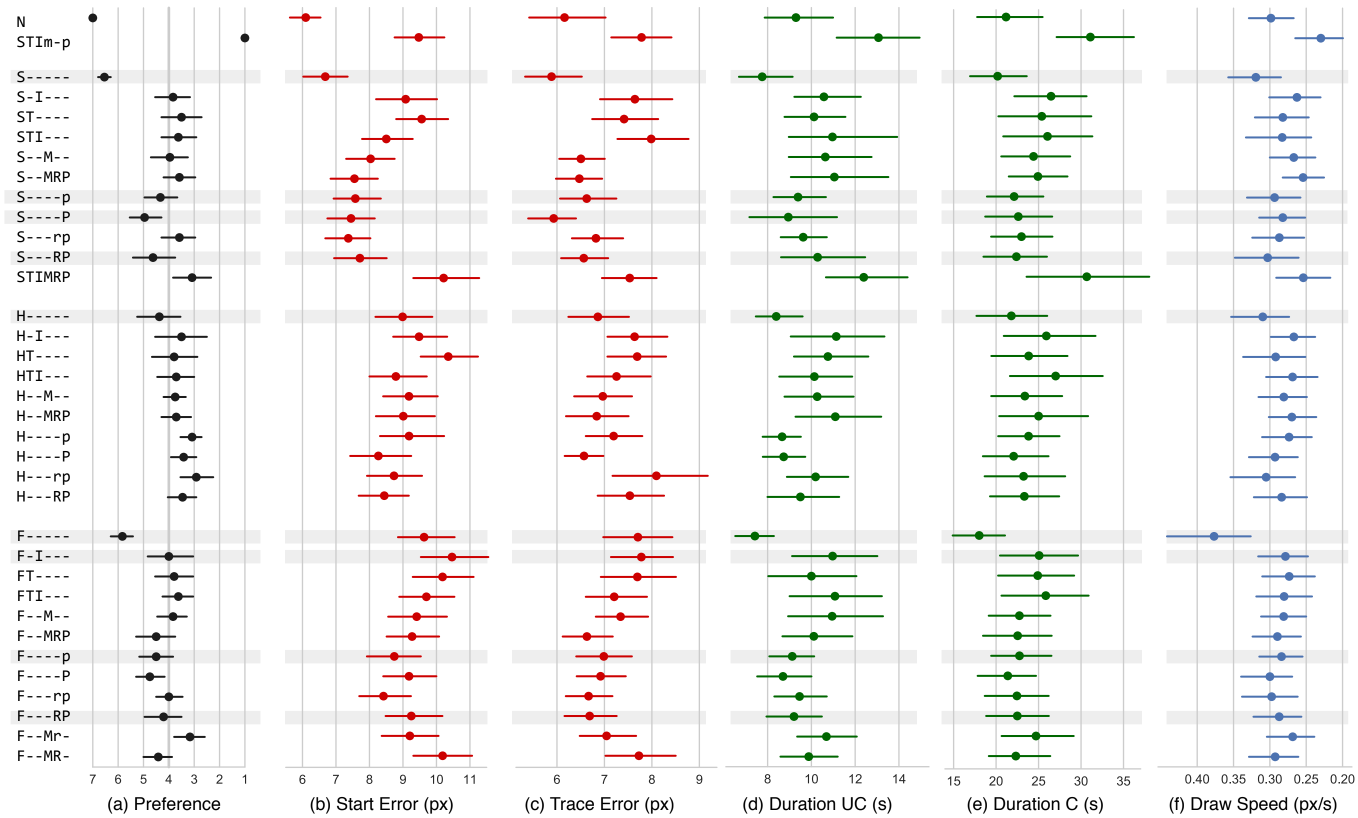

Figure 5: Comparison of postures by main metrics. Some scales inverted to make comparison easier, left-most points in each sub-graph are better (Error bars are 95% confidence interval). The 10 highlighted postures are those used in the recognizer and final application demonstrations.

(**S----P**), pinky in (**S----p**), or both ring and pinky out (**S---RP**) had above neutral preference. The same finger combinations were above neutral for floating palm, but it also added middle, ring, and pinky out (**F--MRP**) and middle and ring out (**S--MR-**). The simple heel palm only posture (**S-----**) was above a neutral score of 4, but heel and finger combinations were below neutral. Postures using grip finger states were less preferred, but index touching with a floating palm (**F-I---**) or side palm (**S-I---**) were both borderline neutral.

It is important to note that the lower baseline posture (**STIm-p**) received a mean preference of 1, but all other gestures received a mean preference of 3 or more, suggesting that no candidate postures were strongly disliked.

Error

We calculated two types of error metrics for constrained tasks similar to Matulic and Norrie's pen and touch tracing experiment [20]: *Start Error* applies to tapping and tracing tasks. It is the distance in pixels from the first contact point of the pen to the dot or line start. *Trace Error* applies only to line tracing tasks. It is the mean of the minimum distance in pixels from each pen stroke position to the target line.

Examining the pattern of errors (Figure 5b,c), the plain side palm (**S-----**) and side palm with non-grip finger combinations generally have lower error, especially for *Start Error*. Floating and heel postures generally have greater error, but of interest is how dropping a ring or pinky finger (e.g. **F----p** and **F---rp**) provides floating palm stability to reduce error.

Side palm with both grip fingers contacting and non-grip fingers out (**STIMRP**) had among the highest *Start Error*, and heel palm with ring and pinky in (**H---rp**) has the greatest *Trace Error*. These error rates exceed those of the baseline posture.

Duration and Speed

The time and speed to complete tasks indicate overall confidence and articulation ability. We calculated two time duration metrics: total stroking time for unconstrained drawing and writing tasks (*Duration UC* in Figure 5d) and total time to complete all constrained tapping and tracing tasks (*Duration C* in Figure 5e). Plain side, heel and floating palms all exhibit low durations. The side palm with both grip fingers and all non-grip fingers out (**STIMRP**) was notably slow, comparable to the low baseline posture.

We also examined the average stroke speed during tracing tasks (Figure 5f). The floating palm posture stands out as a fast posture which explains the higher error. Other postures span the speeds of the upper and lower baseline postures, with a few postures, like side palm and heel palm, slightly exceeding the upper baseline.

DISCUSSION AND DESIGN IMPLICATIONS

The pattern of preferences and other metrics suggest that most postures are reasonable to use. There is some divergence in how higher rated postures perform for error and time metrics.

For example, preferred floating palm postures exhibit higher error, but lower duration with faster movement. To consider this more closely, and to summarize our findings as groups of postures to recommend or avoid, we cluster the experiment data using combinations of metrics.

Recommendations Based on Clustering

K-means is used to group postures into five clusters to represent bands from those that are top choices to use to those that should likely be avoided or used for infrequent actions. Since there is some divergence when considering error and time-related measures, we create two sets of cluster postures based on two feature vectors. The first set focuses on error in 3 dimensions: *Preference*, *Start Error*, and *Trace Errors*. The second set focuses on time-related measures in 4 dimensions: *Preference*, *Duration UC*, *Duration C*, and *Draw Speed*. In both cases, the preference dimension is included given its importance, and mean cluster preference is used to establish a relative group ordering.

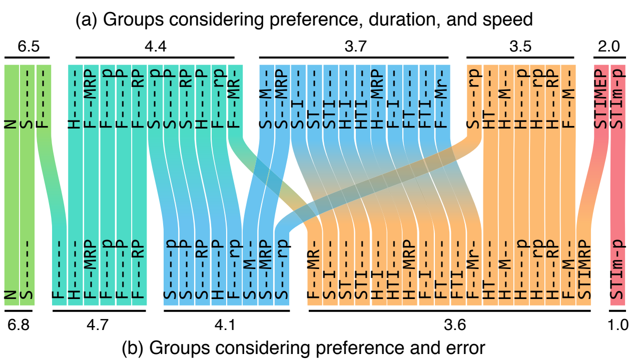

Figure 6: Posture groups: (a) group clusters focused on time; (b) group clusters focused on error. Mean preference shown for each group.

The clustering results are shown in Figure 6. As expected, upper and lower baseline postures appear in the highest and lowest groups. Plain side, heel and floating palms appear in the top two groups. For time-related metrics, there is some cluster separation between pen-finger postures and heel non-pen postures, suggesting the former are used more confidently. The side palm with ring and pinky in shows the largest shift between ordered groups, but all other postures shift no more than one adjacent group.

Left-Handed Participant Pilot

We also conducted the complete protocol with three left-handed participants finding similar results in terms of scale and pattern to those above. Notable exceptions include greater preference for the plain heel posture, less preference for plain floating posture, and a strong dislike for heel with ring and pinky in. Regarding errors, heel postures with any grip-fingers touching show a more pronounced increase in error, and there was an even clearer separation in high drawing speed for floating palm. In spite of some differences, the general pattern is similar, and we believe the interaction technique and posture recommendations hold for left-handed users as well.

Design Implications

Overall, our results show designers should favour side and floating palm postures over those using the heel, with exception of plain heel (`H-----`) and heel with pinky out (`H----P`) postures. Postures with non-grip fingers should be preferred, and within this group, postures using only the pinky (`*----P` or `*----p`) or a matching ring and pinky combination (`*---RP` or `*---rp`) are good choices. Placing all non-grip fingers out with a floating palm (`F--MRP`) is also a reasonable option. Although postures with touching grip fingers are not ranked as high, the clustering exercise suggests they may be good candidates for less precise fast actions.

These results do not consider how reliably different postures can be recognized, an aspect that will have implications on practical implementation and real-world usability. We investigate this in the following section.

RECOGNITION

Our hand posture detection exploits the touch contact pattern of the full hand on the screen surface instead of just the fingertips. Prior work has also looked at that extended touch input space with recognizers either using the hand contour when raw touch data is available [29, 22], contact ellipses [38, 40], or simply the touch points [12]. In most cases, these techniques use classifiers based on simple features and heuristics that may only work well for a small number of very distinct contact patterns. Recent deep learning methods applied on the grey touch image itself have the potential to yield higher recognition performance [23]. Therefore, to recognize the different postures we use a classifier based on a deep neural network that is trained on the pen and touch data recorded in the experiment above. This recognizer is trained for a 10-posture set (see Table 2) selected for the application demonstrations described later. In practice the same methodology can be used to train any set of postures.

Training Data and Recognition Context

The recognizer can be triggered upon or around pen down to determine the mode of the whole pen action, change mode continuously during input [20], or invoke in-place widgets like menus. Many devices, like the Wacom tablet, support hover detection, which provides pen coordinates when the pen is near the surface. The number of hovering input frames just before the pen contacts the surface depends on the sensing hardware and the speed of the hand motion. With our system and data, an average of 5.7 initial pen hovering frames (sd=2.0) are available to be used in the classifier.

Classification

Raw capacitive touch images lend themselves well to neural networks used for natural image classification such as CNNs [23], so we adopt a similar approach. Since we have pen data in addition to the raw touch input, we augment the single-channel image of each touch frame with two additional channels capturing pen position and contact state. Specifically, we draw round blobs centred at the pen coordinates in the second channel if hovering and third channel if touching. Figure 7 shows examples of three-channel images resulting from that data-combining operation.

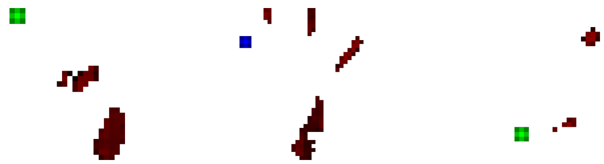

Figure 7: Pen and touch data combined in separate image channels. Represented postures from left to right: `H---rp`, `STIMRP`, and `F--Mr-`

Postures are classified using a deep neural network based on the VGG16 model [32] with convolutional layers pre-trained on ImageNet [5]. This pre-training on natural images allows the network to converge after only a few epochs when fed with other images to classify. Furthermore, VGG is well known and thus reproducible and may be compared with other work.

The features of the network are extracted after the max-pooling operation following the final convolution, which are then fed through a fully connected layer with 1024 units, and finally through another fully connected layer which has output size equal to the number of classes. The first linear layer uses the ReLU [24] activation and dropout [34] with a ratio of 0.4. The network is trained using the Adam [16] optimizer with a learning rate of 0.001, a batch size of 128 and with weight decay set to 0.001. Our neural network architecture was implemented in Python using the Chainer framework [1].

Training and Validation

To train our VGG network, we used the three-channel images of the combined pen and touch data contained within a 200ms window centred on pen down since that is when most of the posture-classification decisions will be made. Only images with actual pen data were used, meaning the pen was either in detectable range for hovering or touching the surface. We split our participant data into 15 training and 3 validations sets (12 experiment participants, 3 left-handed pilot participants, 3 other pilot participants). The data of left-handed participants was mirrored. We did not include the plain floating palm posture for classification, since it has no touch data and therefore is easy to distinguish. The number of frames per posture were approximately 9,000 to 12,000 for the training sets and 1,300 to 2,500 for the test sets. We artificially doubled those samples through data augmentation by applying random translations and light scaling and rotation transforms. Those operations add diversity to the data and fill in screen areas not sufficiently covered by the experiment tasks.

We performed 30 runs of repeated random sub-sampling validation with our leave-3-out scheme for our two posture sets. For each run, we recorded the maximum overall accuracy and lowest loss (softmax cross-entropy) within 5 epochs and computed the average and maximum accuracies and losses obtained over the 30 runs. We also saved the neural network model associated with the max accuracy and minimum loss of each run for detailed analysis.

Results

For the 34-postures set, we obtained an average overall accuracy of 62.2% with associated loss of 1.82, with the best performing model registering a 70% accuracy and a 1.63 loss. The confusion matrix of mean accuracy of the best models in each run is shown in Figure 8.

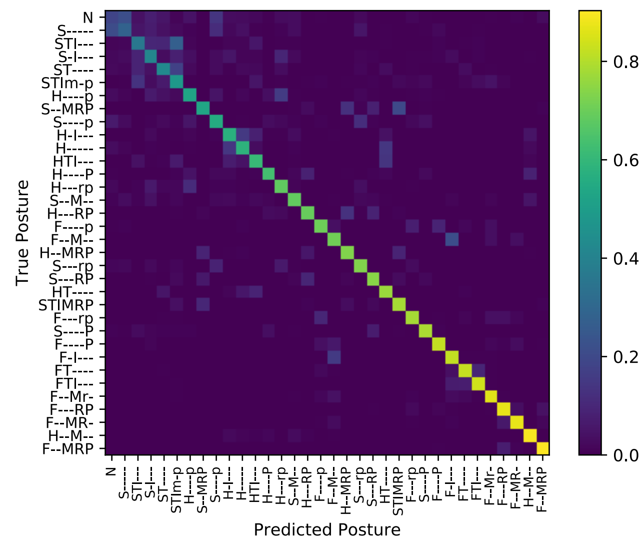

Figure 8: Confusion matrix for all postures except Floating using the best neural network model

The matrix shows that normal and side palm postures are often confused, which is to be expected as people rest on their hand edge when writing normally. Side palm postures with index or thumb touches around the nib also show poor recognition accuracy, perhaps because they are too close and their touch print is not distinct enough. Generally, there seems to be ambiguity between side and heel-based postures, which is also not surprising as the palm base inevitably rolls when positioning the fingers and dragging the hand on the surface. Floating palm-based postures do not suffer from that confusion and hence generally score higher.

For the 10-posture set, the overall average accuracy was 91.4% with a loss of 0.50 and the best performing model had an accuracy of 96% with a loss of 0.20. Those results show that our chosen postures can—in theory—be reliably detected in a real application context. We use the best model as the basis for the recognizer of the applications used in the second study.

APPLICATION DEMONSTRATIONS

To illustrate how unimanual pen+touch input can be applied in a real application context, we created a posture-based interface for two representative types of tasks: document annotation (using Foxit Reader) and vector drawing (using Inkscape). The former is a simpler and more intuitive application because of its real-world parallel. The latter is more complex and requires a dense toolset to use effectively. We use desktop applications for rapid prototyping, but the general concept generalizes to purpose-built tablet applications.

We created three widgets designed specifically for general pen input: radial menus, gesture input, and handwriting recognition for text input (Figure 9). We use radial menus for selecting colours and changing drawing attributes. Gesture input uses six standard Microsoft Application gestures: left and right

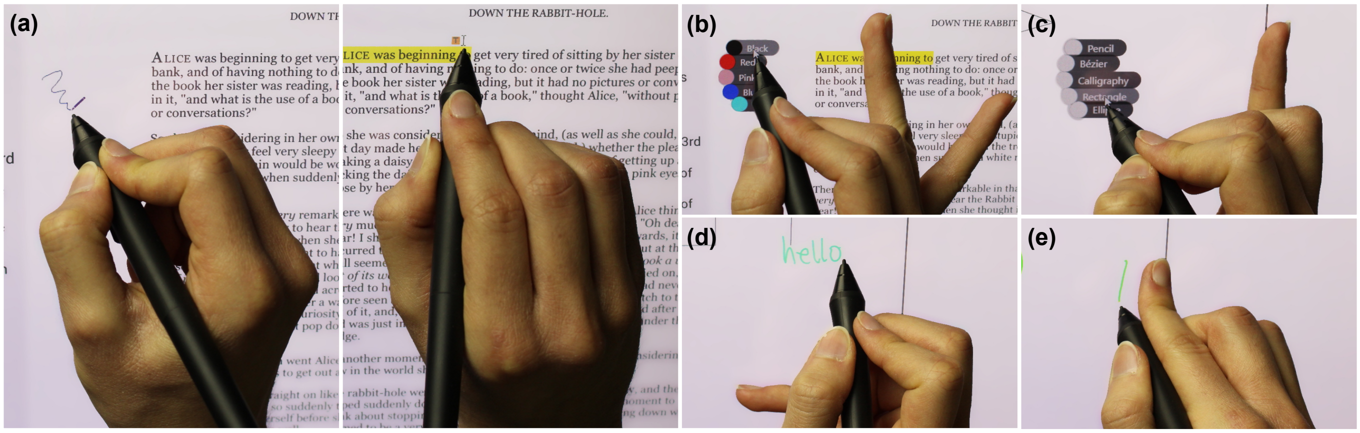

Figure 9: Application highlights: (a) switching from pencil to highlighter with *Side* and *Heel* postures (document annotation); (b) choosing pen colour from radial menu with *Side-RingOut-PinkyOut* (document annotation); (c) object creation menu with *Side-PinkyOut* (vector drawing); (d) using *Side-PinkyIn* to use handwriting recognition for creating a text object (vector drawing); (e) gesture command mode using *Float-Index* (vector drawing).

semi-circles for undo and redo; up and down flicks for copy and paste; and a "scratch out" zig-zag for delete.

Posture Mapping and Interfaces

Using a set of 10 postures, we designed the mappings and interfaces to optimize semantic proximity and posture-action suitability (Table 2). The results of the suitability experiment were taken into account when choosing the posture set and mapping postures to application commands. The video provides full application demonstrations.

Semantic proximity: This was achieved by mapping similar actions to related postures. In both applications, side palm based postures were associated with creation: drawing, erasing, creating shapes etc., whereas floating palm based postures were associated with more "macro" interactions: selecting, transforming, and styling in the vector editor; navigation and search in the document annotator.

Posture-action suitability: This was achieved by mapping common actions to more preferred and less restrictive postures. In both applications, the most common, most precision-demanding tool is mapped to *Side* while the next most common is mapped to *Side-PinkyOut*. Both of these postures had high preference scores and accuracy results. Menus are triggered by postures which allow for pen tip motion but restrict hand motion, i.e. those using *RingOut-PinkyOut*. In document annotation, another equally simple posture, *Heel*, is used for the common action of highlighting. Actions requiring less precision, like gesturing, were assigned to postures with lower accuracy results, *Float-Index*.

Technical Implementation

Our implementation works as a global service that runs the recognizer in the background and injects commands into the current application. Any application can be given a unimanual pen+touch interface by defining a YAML configuration file which is automatically activated based on the foreground window's title. The global service (C#, .NET, WPF) forwards raw touch and pen input data to the recognizer (Python) using sockets. To improve performance, the service triggers only one recognition request, consisting of the latest 8 frames of input, every 150ms. Everything is done asynchronously, so no software lag is introduced and there is no increased delay for pen or touch events. The maximum possible delay from when a posture change occurs to when the system recognizes it is 200ms. However, since the postures require some time to form, this delay is not noticeable in practice. Based on the recognized posture, the service then triggers actions to the foreground application by sending keyboard shortcuts, key strokes, or mouse events (using Window's `SendKeys` and `inputsimulator`), displaying pen-localized radial menus (using custom WPF windows), or by collecting strokes for gesture or text recognition (using Microsoft's InkCanvas APIs).

Posture	Annotation	Drawing
`S-----`	Pencil	Polygon Tool
`S----p`	–	Text Tool
`S----P`	Eraser	Creation Menu
`S---RP`	Colour Menu	Style Menu
`H-----`	Highlighter	Node Editor
`F-----`	Hand	Select Tool
`F-I---`	Gestures	Gestures
`F----p`	Search	–
`F----P`	–	–
`F---RP`	–	Style Menu

Table 2: Posture mapping for application demonstrations

STUDY: USABILITY IN PRACTICE

The goal of this study is to test general usability of unimanual pen+touch input in a more realistic setting. Using the demonstration applications, we examine if people can accomplish realistic tasks using the techniques, whether they can remember posture-to-command mappings, and also get a sense for in situ recognizer accuracy.

Participants and Apparatus

We recruited 5 right-handed participants, ages 22 to 30, of which 2 were female. Recruiting was done by word-of-mouth and each participant received $10 for successful completion of the 90-minute study. Previous pen-tablet experience was preferred, since the problem of pen mode-switching would be better understood: 4 of the participants had such experience.

The same Wacom tablet was used, but with an HP Envy (Windows 10, Core i5 2.60 GHz, 8GB RAM) because of its GPU (GeForce GT 740M) to improve recognition speed.

Design and Protocol

The session began with approximately 10 minutes of posture training. This required the participant to complete a reduced task set from Study 1 with all 10 postures used by the demo applications. Unlike Study 1, the *recognized* posture was displayed so the participant knew if the recognizer successfully identified the performed posture. After posture training, the participant completed a set of training and test tasks for document annotation, followed by a set of training and test tasks for vector drawing. During this training session, a document was loaded into an application that included instructions to perform certain tasks along with the postures to use. During the trials, they were only given written instructions, although they were allowed to ask the experimenter if they forgot the corresponding posture. The tasks for each application were chosen to be non-expert, relatively common, and representative of the possible postures. A post-study interview was conducted after all application tasks were completed.

Document annotation tasks were provided as a list of operations to perform on a PDF document, such as“highlight the word ‘Alice’ in yellow” and “search for the word ‘Party”’. The complete set of document annotation tasks were: circling words or writing text with the pencil tool; erasing with the eraser tool; highlighting text; changing pencil or highlighter colour; performing gestures; searching the document.

Vector drawing tasks required the participant to draw a set of shapes to match a given drawing. These shapes were chosen to require multiple tool mode-switches. The complete set of vector drawing tasks were: drawing polygons with the polygon tool; using other creation tools (pencil, rectangle, ellipse); node editing; transforming objects; styling objects; performing gestures; entering text.

Results

All participants successfully completed the experiment in under 90 minutes. Due to scheduling issues, one participant (P1) did the study in 2 parts over 2 days. Four participants said they would use at least some postures in their personal work if the recognizer was more accurate. The fifth participant had experience with indirect pen tablets and preferred keyboard shortcuts for mode-switching.

While trying to recall corresponding posture actions during the “test” section, participants often tried to perform the posture, and watch the cursor icon to determine if they were in the correct mode. Participants had the most trouble with the colour menu of the document annotator. *Side-RingOut-PinkyOut* would bring up the colour menu for the pencil tool or the highlighter tool, depending on which tool was previously active. Although participants found the semantics of this useful: *“I’d just drop the 2 fingers regardless of palm”* (P3), they found it difficult to transition to it from *Side* without another posture being recognized during the transition. Three participants said there were too many postures: *“I would rather maybe only have a core posture and one that lets me change its tool”* (P4). This could be due to the novelty of the technique.

Three participants commented on discomfort with *RingOut-PinkyOut*, but some commented that slight adjustment addressed this *“It was initially difficult, but some slight modifications had a positive impact”* (P1). The short time of the study prevented all participants from finding comfortable posture variants which worked best for their hand. Two participants did comment positively on *Float-Index*. Participants were encouraged to and experimented with using different fingers for particular postures, such as the middle or ring fingers near the pen nib instead of the index finger.

The recognizer worked well with some misclassification more apparent for certain participants. For some participants, *Side* would be frequently misclassified as *Heel* or *Side-PinkyIn*. Despite data augmentation, the recognizer also exhibited poorer performance in regions not covered by the first experiment, causing us to verbally prompt the user to perform the postures closer to the centre of the screen.

DISCUSSION

The results of the first study suggest that a number of unimanual pen+touch postures may be performed relatively comfortably while maintaining pen control. This was further supported by all 5 participants completing the tasks in the second study.

Although the deep learning recognizer exhibited high accuracy with the controlled experiment data, the more nuanced results of the second study suggest improvements are needed for real application contexts. More data is necessary to determine if recognition errors are due to inherent posture similarity or to insufficient training data. Accuracy would likely increase by at least partially training the network on each target user, or by using a reduced set of say 4 to 5 robust postures.

The demonstrations show pre-existing applications can be augmented to create simple or complex unimanual pen+touch experiences. The core pen, highlighter, and eraser switching was positively received, as were the gesture and text input widgets. Posture-based menu interactions were less positively received, possibly because the colour menu could be difficult to trigger and proved a little unstable due to software issues.

Other applications that could make use of unimanual pen+touch postures are painting applications, with mappings to different brushes and colours, and spreadsheet editors, with posture-based switching between data entry (using handwriting recognition) and selection or data manipulation. Furthermore, our text input widget hints at general keyboard input, and even triggering keyboard shortcuts using postures. For example, writing the letter C using *Float-PinkyOut* could trigger CTRL-C. Scaling this to shortcuts with multiple modifiers would be a challenge, but could bring desktop shortcut methods to a pen-only environment.

Limitations and Opportunities

Applicability for users with reduced hand control — Older people, children, people with motor impairments, and people with hand injuries such as missing digits might not be able to fully utilize our technique. We are proposing a way to accelerate existing GUI operations, so redundant input methods may exist. Individual differences in hand control could be leveraged to create user-specific postures, especially when non-precision grips or irregular hand physiology is present.

Compatibility with other pen input methods — Some input methods, like pushing barrel buttons or simultaneous fine control of pen pressure, are likely harder to execute while maintaining some unimanual postures. But we believe many techniques, like using the eraser end of the pen, tilting, and rotating could be compatible. Since our input space includes the normal posture, all pen input techniques remain compatible with that grip, but perhaps not the diverse postures we propose. Unimanual postures are also compatible with marking menus and related command gestures, and our postures provide a solution to the "inking versus command" problem.

Discoverability — Admittedly these postures are not intuitive to users and must be taught, but a participant who did the second study over two days commented on improvement on the second day, suggesting that the learning curve is not steep. The second study further also suggests that cursor feedback is essential for users to discover posture actions once they know what postures are possible. Our succinct posture naming formats (`F---RP` or *Float-RingOut-PinkyOut*) could be displayed alongside menu items and tooltips, similar to how keyboard shortcuts are displayed in some applications. Using onscreen gesture guides for training [10] is another approach.

Cognitive Learning — Like any large input space, memorizing mappings between posture and action requires some effort. Future work can examine learning directly and investigate how a rehearsal-based interface or feed-forward technique could help transition from novice to expert performance.

Physical Learning — In addition, since pen grip is a fine motor skill, more time, or a night's sleep after exposure might have an impact on comfort and control, as suggested again by our two-day study participant, who also admitted to experiencing less discomfort on the second day.

Recognition Issues — Some systematic recognition issues were observed. Participants mentioned difficulty when transitioning between *Side* to *Side-RingOut-PinkyOut* or *Side-PinkyOut*. During the transition, the hand contact can erroneously trigger any of those three postures. Although not a problem if posture actions are non-destructive, those postures triggered menus, which proved jarring when not expected. One possible solution is to have a mandatory hold period when transitioning between similar postures. In general, false positives were minimal, though we noticed *Heel* was often misclassified as *Side*. We did observe a larger variety of *Heel* postures in the first study, which might have affected the recognizer. A stricter definition of that posture might reduce misclassifications.

Hardware — Our recognizer requires raw capacitive input, something all touch devices support, but vendors often do not expose it without some low level system work [42, 23]. Other simpler, more synthetic touch data (such as contact ellipses or even touch points) could possibly be sufficient to recognize a few very distinctive postures.

CONCLUSION

We introduced a novel unimanual pen+touch input space. The results of our first evaluation with 33 postures indicated many are reasonable in terms of subjective comfort and objective degree of control. Using data from the study, we trained a convolutional neural network using a pre-trained VGG architecture to recognize the postures with high accuracy. Two application demonstrations using a 10-posture interface show how the techniques work in practice, and a small usability study found positive feedback. We were surprised to find so many viable postures, with our demonstration applications showing that 10 postures are feasible. However, even a more limited system with 2 or 3 postures would be valuable for fast switching between frequent modes, for instance ink, highlighter, and eraser in a simple note-taking application.

A logical next step is to formally test the performance of switching between different postures using a standard mode-switching experiment protocol [18, 35]. Also, a direct comparison with techniques such as marking menus would establish performance benchmarks relative to popular baselines. We hypothesize that unimanual pen+touch postures more closely associate command activation with direct manipulation, but unlike marking menus, there is no obvious method to support novice-to-expert learning for our postures.

Other future directions include exploring a smaller set of postures for mobile tablets or other non-tripod grip postures (like a power grip) that can be sensed using only the capacitive pattern on the touch surface. Our hope is that our work contributes to making pen input more expressive, meaning the input space is increased and more nuanced. We remain in awe of how remarkable the human hand is, and what it is capable of when given the right digital support.

ACKNOWLEDGEMENTS

This work was made possible by funding from NSERC Discovery Grant (#402467).

REFERENCES

1. Chainer: A flexible framework for neural networks. `https://chainer.org/`
2. Xiaojun Bi, Tomer Moscovich, Gonzalo Ramos, Ravin Balakrishnan, and Ken Hinckley. 2008. An exploration of pen rolling for pen-based interaction. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. ACM, Monterey, CA, USA, 191–200.
3. Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. 2008. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proceedings of the working conference on Advanced visual interfaces*. ACM, Napoli, Italy, 154–161. DOI: `http://dx.doi.org/10.1145/1385569.1385595`
4. William Buxton. 1995. Touch, Gesture & Marking. In *Readings in Human Computer Interaction: Toward the Year 2000*. San Francisco: Morgan Kaufmann Publishers, Chapter 7, 469–482.

5. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. DOI: http://dx.doi.org/10.1109/CVPR.2009.5206848
6. John M. 1 Elliott and K. J Connolly. 1984. A classification of manipulative hand movements. *Developmental Medicine & Child Neurology. Vol 26(3)* (1984).
7. Nicholas Fellion, Thomas Pietrzak, and Audrey Girouard. 2017. FlexStylus: Leveraging Bend Input for Pen Interaction. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 375–385. DOI: http://dx.doi.org/10.1145/3126594.3126597
8. George Fitzmaurice, Azam Khan, Robert Pieké, William A. S. Buxton, and Gordon Kurtenbach. 2003. Tracking menus. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*. ACM, Vancouver, Canada, 71–79.
9. Clifton Forlines, Daniel Vogel, and Ravin Balakrishnan. 2006. HybridPointing: fluid switching between absolute and relative pointing with a direct input device. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. ACM, Montreux, Switzerland, 211–220.
10. Dustin Freeman, Hrvoje Benko, Meredith Ringel Morris, and Daniel Wigdor. 2009. ShadowGuides: Visualizations for In-situ Learning of Multi-touch and Whole-hand Gestures. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*. ACM, New York, NY, USA, 165–172. DOI: http://dx.doi.org/10.1145/1731903.1731935
11. Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. 2006. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, Montréal, Québec, Canada, 861–870.
12. Chris Harrison, Robert Xiao, Julia Schwarz, and Scott E. Hudson. 2014. TouchTools: Leveraging Familiarity and Skill with Physical Tools to Augment Touch Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2913–2916. DOI: http://dx.doi.org/10.1145/2556288.2557012
13. Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, and Francois Guimbretiere. 2005. Design and Analysis of Delimiters for Selection-action Pen Gesture Phrases in Scriboli. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 451–460. DOI: http://dx.doi.org/10.1145/1054972.1055035
14. Ken Hinckley, François Guimbretière, Patrick Baudisch, Raman Sarin, Maneesh Agrawala, and Edward Cutrell. 2006. The springboard: multiple modes in one spring-loaded control. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, Montréal, Québec, Canada, 181–190.
15. Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. 2010. Pen + touch = new tools. *Uist '10* (2010), 27. DOI: http://dx.doi.org/10.1145/1866029.1866036
16. Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).
17. Gordon Kurtenbach and William Buxton. 1994. User Learning and Performance with Marking Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*. ACM, New York, NY, USA, 258–264. DOI: http://dx.doi.org/10.1145/191666.191759
18. Yang Li, Ken Hinckley, Zhiwei Guan, and James a Landay. 2005. Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces. *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems* (2005), 461–470. DOI: http://dx.doi.org/10.1145/1054972.1055036
19. Christine L. MacKenzie and T. Iberall. 1994. *The Grasping Hand (Advances in Psychology)* (1 ed.). North Holland.
20. Fabrice Matulic and Moira Norrie. 2012. Empirical Evaluation of Uni- and Bimodal Pen and Touch Interaction Properties on Digital Tabletops. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces (ITS '12)*. ACM, New York, NY, USA, 143–152. DOI: http://dx.doi.org/10.1145/2396636.2396659
21. Fabrice Matulic and Moira C. Norrie. 2013. Pen and Touch Gestural Environment for Document Editing on Interactive Tabletops. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces (ITS '13)*. ACM, New York, NY, USA, 41–50. DOI:http://dx.doi.org/10.1145/2512349.2512802
22. Fabrice Matulic, Daniel Vogel, and Raimund Dachselt. 2017. Hand Contact Shape Recognition for Posture-Based Tabletop Widgets and Interaction. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. ACM, New York, NY, USA, 3–11. DOI: http://dx.doi.org/10.1145/3132272.3134126
23. Sven Mayer, Huy Viet Le, and Niels Henze. 2017. Estimating the Finger Orientation on Capacitive Touchscreens Using Convolutional Neural Networks. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. ACM, New York, NY, USA, 220–229. DOI: http://dx.doi.org/10.1145/3132272.3134130

24. Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10)*. Omnipress, USA, 807–814. http://dl.acm.org/citation.cfm?id=3104322.3104425

25. John R. Napier. 1956. The prehensile movements of the human hand. *The Journal of Bone and Joint Surgery. British Volume* 38-B, 4 (Nov. 1956), 902–913. http://www.ncbi.nlm.nih.gov/pubmed/13376678

26. Ken Pfeuffer, Ken Hinckley, Michel Pahud, Bill Buxton, and Interactive Systems. 2017. Thumb + Pen Interaction on Tablets. *Acm Chi* (2017), 3254–3266. DOI: http://dx.doi.org/10.1145/3025453.3025567

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

28. Rosemary Sassoon. 1993. *The Art and Science of Handwriting*. Intellect Books.

29. Dominik Schmidt, Ming Ki Chong, and Hans Gellersen. 2010. HandsDown: Hand-contour-based User Identification for Interactive Surfaces. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10)*. ACM, New York, NY, USA, 432–441. DOI: http://dx.doi.org/10.1145/1868914.1868964

30. Julia Schwarz, Robert Xiao, Jennifer Mankoff, Scott E. Hudson, and Chris Harrison. 2014. Probabilistic Palm Rejection Using Spatiotemporal Touch Features and Iterative Classification. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2009–2012. DOI: http://dx.doi.org/10.1145/2556288.2557056

31. Ann-Sofie Selin. 2003. *Pencil grip: A descriptive model and four empirical studies*. 140 pages.

32. Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

33. Hyunyoung Song, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. 2011. Grips and gestures on a multi-touch pen. *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11* (2011), 1323. DOI: http://dx.doi.org/10.1145/1978942.1979138

34. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* 15, 1 (Jan. 2014), 1929–1958. http://dl.acm.org/citation.cfm?id=2627435.2670313

35. Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 2017. Experimental Analysis of Mode Switching Techniques in Touch-based User Interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3267–3280. DOI: http://dx.doi.org/10.1145/3025453.3025865

36. Daniel Vogel and Ravin Balakrishnan. 2010. Direct pen interaction with a conventional graphical user interface. *Human–Computer Interaction* 25, 4 (2010), 324–388.

37. Daniel Vogel and Géry Casiez. 2011. Conté: Multimodal input inspired by an artist's crayon. *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11* c (2011), 357. DOI: http://dx.doi.org/10.1145/2047196.2047242

38. Daniel Wigdor, Hrvoje Benko, John Pella, Jarrod Lombardo, and Sarah Williams. 2011. Rock Rails: Extending Multi-touch Interactions with Shape Gestures to Enable Precise Spatial Manipulations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1581–1590. DOI: http://dx.doi.org/10.1145/1978942.1979173

39. Fong-Gong Wu and Shuyi Luo. 2006. Design and evaluation approach for increasing stability and performance of touch pens in screen handwriting tasks. *Applied Ergonomics [Kidlington]* 37, 3 (2006).

40. Mike Wu and Ravin Balakrishnan. 2003. Multi-finger and Whole Hand Gestural Interaction Techniques for Multi-user Tabletop Displays. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA, 193–202. DOI:http://dx.doi.org/10.1145/964696.964718

41. Mike Wu, Chia Shen, Kathy Ryall, Clifton Forlines, and Ravin Balakrishnan. 2006. Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces. In *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. IEEE Computer Society, Washington, DC, USA, 185–192. DOI: http://dx.doi.org/10.1109/TABLETOP.2006.19 ACM ID: 1110635.

42. Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, USA, 47–50. DOI: http://dx.doi.org/10.1145/2817721.2817737

43. Yizhong Xin, Xiaojun Bi, and Xiangshi Ren. 2011. Acquiring and Pointing: An Empirical Study of Pen-tilt-based Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 849–858. DOI: http://dx.doi.org/10.1145/1978942.1979066

44. Ka-Ping Yee. 2004. Two-handed interaction on a tablet display. In *CHI '04 extended abstracts on Human factors in computing systems*. ACM, Vienna, Austria, 1493–1496.