Grips and Gestures on a Multi-Touch Pen

Hyunyoung Song[1,3,4], Hrvoje Benko[1], Francois Guimbretiere[4], Shahram Izadi[2], Xiang Cao[2], Ken Hinckley[1]

[1]Microsoft Research
One Microsoft Way
Redmond WA 98052

[2]Microsoft Research Cambridge
7 JJ Thomson Avenue
Cambridge CB3 0FB, UK

{benko,shahrami,xiangc,kenh}@microsoft.com

[3]University of Maryland
Computer Science
College Park, MD 20742
hsong@cs.umd.edu

[4]Cornell University
Information Science
Ithaca NY, 14850
francois@cs.cornell.edu

ABSTRACT

This paper explores the interactive possibilities enabled when the barrel of a digital pen is augmented with a multi-touch sensor. We present a novel multi-touch pen (*MTPen*) prototype and discuss its alternate uses beyond those of a standard stylus, such as allowing new touch *gestures* to be performed using the index finger or thumb and detecting how users *grip* the device as a mechanism for mode switching. We also discuss the hardware and software implementation challenges in realizing our prototype, and showcase how one can combine different grips (tripod, relaxed tripod, sketch, wrap) and gestures (swipe and double tap) to enable new interaction techniques with the MTPen in a prototype drawing application. One specific aim is the elimination of some of the comfort problems associated with existing auxiliary controls on digital pens, such as barrel buttons and barrel scroll wheels; such mechanical controls work best in only a few specific hand grips and pen rotations. Comparatively, our gestures can be successfully and comfortably performed regardless of the rotation of the pen or how the user grips it, offering greater flexibility in use. We describe a formal evaluation comparing MTPen gestures against the use of a barrel button for mode switching. This study shows that both swipe and double tap gestures are significantly faster than commonly employed barrel buttons after acquisition time is taken into consideration.

Author Keywords

Digital stylus, digital pen, multi-touch, grip detection.

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces- Graphical user interfaces.

INTRODUCTION

Digital pens and styli are popular input devices for digital tasks, such as annotating and drawing, due to their ability to leverage the fine precision of a pen tip and the dexterity of users' hands. Beyond this fine control of a physical pen, digital pens also support common interface tasks such as mode switching and scrolling. However, current commercial digital pens offer limited additional input capabilities, mainly consisting of mechanical buttons or scroll wheels mounted on the barrel of the pen [16, 27]. These options provide *explicit* controls and are simple to operate, but are limited in number and fixed in location and size. The use of mechanical buttons and wheels also leads to many usability challenges: Due to different hand sizes or different tasks, users often wish to hold their pens in different ways, and the need to access these mechanical controls greatly restricts the variety of grips that can be used. In addition to these grip restrictions, some grips can result in users mistakenly triggering buttons when simply wanting to ink, or having to rotate or reposition the pen to acquire a physical button on the stylus when wanting to access on-screen context menus.

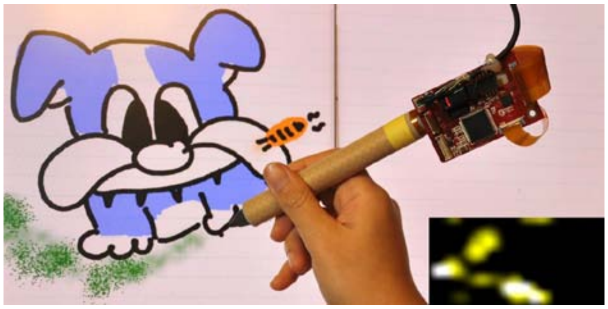


Figure 1.Our MTPen prototype. Inset image shows touch data imaged by the multi-touch sensor around the barrel.

Previous studies suggest using other dimensions of a digital pen such as rolling [4] or tilting [25] as alternative explicit controls [4, 23, 25] required by digital interfaces. Additionally, researchers have demonstrated that grip can be leveraged as an *implicit* dimension that conveys some user intention (or state) as users interact with a device [14, 24, 29]. This can also be applied to a pen: There are many instances where the way in which the pen is being held can convey information. For example, when a user holds the pen using the conventional tripod grip [13] (Figure 1), this may indicate that she requires precision for a task such as inking or drawing. Clearly the use of grips could become an important cue in increasing the input vocabulary of pen users.

In this paper we explore the possibilities of broadening the gestural language of the digital pen through multi-touch sensing along the pen barrel. We demonstrate how this novel combination allows for sensing of both the handgrips, for implicit controls, and finger-based touch

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
Copyright 2011 ACM 978-1-4503-0267-8/11/05....$10.00.

gestures, for explicit controls, all while preserving the symmetric nature of a normal pen. We provide the following four contributions:

First, we present a custom-built and novel digital pen prototype called *MTPen* with a capacitive multi-touch sensor wrapped around an off-the-shelf stylus (Figure 1).

Second, we demonstrate the implementation of a recognizer that can distinguish different hand grips using the contact data from fingers resting on the multi-touch sensor. This recognizer also detects finger-based touch gestures such as horizontal or vertical swipes and double taps with the thumb or index finger. We further discuss the challenges in building such a recognizer, in particular interpreting contact data on a pen device of curved cylindrical nature coupled with the close proximity of fingers resulting in many merged contact points.

Third, we demonstrate how the ability to sense grips and gestures on the MTPen device can be combined into novel interaction techniques to provide the ability to switch modes, adjust continuous parameters, and issue customized commands within a custom drawing application.

Finally, to confirm the viability of our gesture approach, we conduct a formal evaluation to test our pen performance with compare to an existing mode-switching technique, the pen barrel button. We found that the performance of our proposed finger gestures, swipe or double tap, are comparable to mechanical barrel button without its disadvantages.

RELATED WORK

Our work builds upon three distinct areas of prior research that are covered in turn in this section: The first is the body of work that investigated adding additional input controls to the digital pen; the second is work that explored the use of touch sensing on the surface of existing input devices to extend interaction vocabulary; the third covers studies in grips and pen design in ergonomics.

Additional Input Controls on Digital Pens

Digital pens have been augmented with auxiliary hardware controls such as one or two buttons or a scroll wheel (e.g. Wacom's Airbrush) to manipulate the pen's properties or on-screen widgets. However, Li et al. [15] performed a comparative study of five mode switching techniques and concluded that clicking a mechanical button on a digital stylus was not the optimal solution. Inspired by their study, we use the same methodology to compare our new gestures on MTPen – the swipe and double tap – with the de facto standard barrel button implementation.

Some researchers have proposed solutions to mode switching such as using a distinct gesture in the hover zone above the tablet to change the modality [8, 22], and redesigning the entire pen interface to use ink crossing [1] or pigtail strokes [10].

Another line of research for pen control is to utilize other pen parameters such as rolling [4], tilting [25], pressure [19], and combined rolling and motion [23] that can be used for changing modes or providing further single handed input mechanisms. Pen rolling [3] at a resolution of 10 degrees around its longitudinal axis can be used to control a mode change and multi-parameter input. The *Tilt Menu* [25] is a technique to generate secondary input by tilting the angle of the pen to select one of the eight pie menus. *Pressure widgets* [19] demonstrated how six levels of pressure can, with appropriate visual feedback, improve selection tasks. Suzuki et al. [23] explored pen rolling and shaking as an interaction technique. Our approach in enabling finger touch gestures on a pen adds a new perspective to this body of research.

Extending Device Capabilities through Touch Sensing

While the MTPen is the first prototype to utilize a multi-touch sensor on the outer barrel of a pen, multi-touch sensing technologies have been demonstrated on many other input devices to extend interaction capabilities. Most recently, *Mouse 2.0* [26] showed a series of re-designs of the traditional mouse to support touch sensing on the outer casing and beyond.

Researchers have also attempted to understand the user's intention (or mode switch) by capturing the *implicit input* that occurs in the ways people hold their devices [14, 24, 29]. For example, *Graspables* [24] demonstrated how a coarse grid of capacitive sensors can be wrapped around various form-factors and shapes to allow different grips to be recognized and utilized for input. Our work extends this approach and uses a much finer grained multi-touch sensor to understand the movements of the user's fingers while grasping a pen.

Combining a multi-touch surface with pen-based usage scenario has also been explored in recent years [5, 12, 31]. These projects explored bimanual techniques that support multi-touch input using the non-dominant hand, whilst supporting pen-based interaction using the dominant. Such interactions have been employed for drawing [5], annotation in active reading environment [12], and even solving math problems [31]. In comparison, our MTPen provides touch directly on the barrel of the pen and explores a novel way for touch and ink to be combined.

Grips and Pens in Ergonomics

Fields such as ergonomics and surgery have been exploring the dynamics of the hand and grips for digital [30] and physical pen design [20], particularly in the context of drawing and writing. Goonetilleke [20] investigated how different barrel shapes (triangular, square, elliptical, hexagonal, octagonal, circular) of the pen can offer improved drawing and writing performance.

In prior studies, Napier [18] provided valuable observations that we found our work on. First, according to Napier, *"during a purposive prehensile action, the posture of the hand bears a constant relationship to the nature of that activity"*. Napier also lays the foundation for our postulation that while griping the pen, the majority of the hand is used to stabilize the grasp while only certain fingers remain free to carry out a gesture. These insights have been valuable in designing and developing our research prototype.

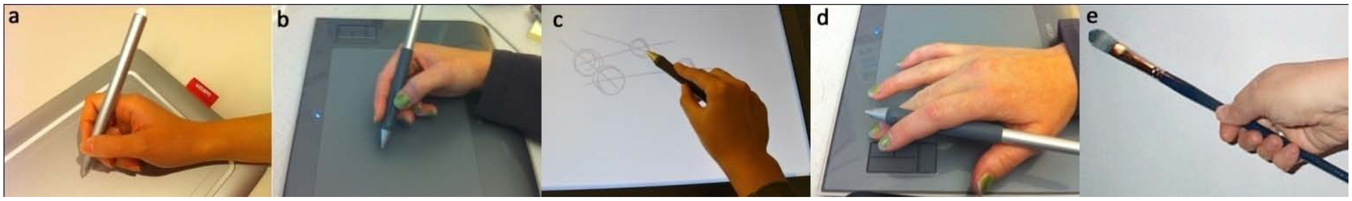


Figure 2. Different grips observed in our pilot investigation: (a) Tripod grip is used for precise writing or drawing, (b) Relaxed tripod grip is frequently used for less precise tasks, (c) Sketch grip, (d) Tuck grip, and (e) Wrap grip is an example of a grip not frequently used with digital pens but is often used with physical tools.

MOTIVATING INVESTIGATION

To help better understand the limitations of current digital pens in everyday computing scenarios, we conducted an observational design study with expert users of digital pens. We interviewed four regular digital pen users (2 male, 2 female) aged 27–35. Each used a digital stylus for more than half of their computing time on a regular basis. The participants were long-term pen users (ranging 5–15 years) and were industrial designers and artists by profession. All participants used Intuos/Cintiq Grip pens, which feature two buttons on the barrel, a pressure sensitive tip, and a pressure sensitive eraser (Figure 2). We asked them to fill out a questionnaire (with ratings on a 7 point Likert scale) before the interview, which asked about their everyday pen usage. We then observed how they use their digital pen in their work environment and conducted a one-on-one interview that focused on understanding two issues: a) the extent to which the current pen design supports changing the pen grip while inking, gesturing, or relaxing, and b) how users access the mechanical buttons on the pen.

While all our users expressed that they would prefer to hold the pen in different ways depending on the task, they commented that the fixed button location makes changing the grip difficult. As a result, three of our participants adapted their grips in order to be able to activate the buttons, e.g., we observed that P3 – who has been using digital pens for over 15 years – had developed an awkward style for griping the Wacom pen so that both buttons were always underneath his finger. P4 commented, "*While digital drawing applications excel in replicating other physical drawing tool properties, the design of digital stylus is not only limited to replicating a pen but even constrains how we actually hold the pen.*"

When asked about the different pen grips that the users currently employ or would like to use, the most common grip for all our four designers was the *tripod* grip [13] (Figure 2a). It is a type of a precision grip [18] that is most commonly used in writing and drawing. P2 demonstrated a relaxed version, i.e., a *relaxed tripod* grip (Figure 2b), used not only to draw and write but also manipulate their general software controls. P1 mentioned that she would relax her tripod grip when less precision is required such as using a marker for highlighting documents. P3 demonstrated a *sketch* grip which is used to create long and consistent strokes especially when the tablet was angled (Figure 2c). When transitioning between different input controls, P1 adjusted their grip to balance the pen in between the fingers such that the pen was tucked underneath the middle finger (*tuck* grip, Figure 2d). Finally, P4 showed a *wrap* grip that they would use to hold a crayon, a stamp, or a brush, or when a drawing tool is being transported (Figure 2e).

While participants all agreed that they would appreciate more controls (buttons or scroll wheels) for interacting with their pen (*I would like more control while holding the pen* averaged 6 - on the 7 point Likert scale) they all complained that the current barrel button was difficult to access (*The button is the right size* averaged - 2.25 on the Likert scale, and *the right location* averaged – 3). This is likely due to the one-size-fits-all design of the barrel button which does not account for the diversity in hand sizes and grips. P1 and P4 pointed out that they put down and re-grasp their pens when they have to change input devices (e.g. keyboard or other auxiliary controls on the tablet). This frequent transition makes the acquisition of the button difficult for them, as the button is often not directly under their finger when they retrieve the pen. Other participants resonated highly with that sentiment in their answers (*I change my pen grip to access the barrel button* – 6.75, *I rotate the pen to access the barrel button* – 6.75). At the same time, users would often "hide" the button by rotating the pen when drawing to reduce accidental activations, then rotate the pen back in order to use the button again later (*I sometimes press the barrel button accidently when I don't intend to* – 5.75).

In addition to these hardware design limitations, our participants complained that a single (or two) button input is very limiting given that their drawing software is overloaded with numerous options and controls. While the scroll wheel on P4's Wacom Airbrush pen affords a bit more in terms of continuous input, she expressed the desire to manipulate more controls while inking, which is currently difficult even with additional inputs using the non-dominant hand (e.g., keyboard or side buttons on the tablet). All of these observations demonstrate a need to increase the interactive potential of the pen by supporting different grips and novel input techniques.

MULTI-TOUCH PEN IMPLEMENTATION

Motivated by the feedback from the experts, we chose to redesign the standard digital pen to activate mode switching regardless of user hand pose, while allowing for a diversity of user grips. We designed a custom pen which does not employ discrete mechanical buttons, but instead uses a touch-sensitive barrel as a means to explore a new approach to the problem. MTPen allows for sensing of different hand *grips*, and enables auxiliary controls via touch *gestures* along the barrel. In addition, our design opens up possibilities to combine grips and gestures that allow access to greater number of interactions than are available using the standard mechanical button models.

Hardware Design

Our prototype is built around a standard Wacom Intuos/Cintiq grip pen which is augmented with a custom capacitive sensor (Figure 3). We designed a plastic cylindrical enclosure around the pen onto which we overlaid a flexible matrix of capacitive-sensing electrodes to track the image of user's hand. Our enclosure eliminates access to the barrel button and makes the pen completely smooth and cylindrical. The length of the pen is 175 mm with a diameter of 16mm.

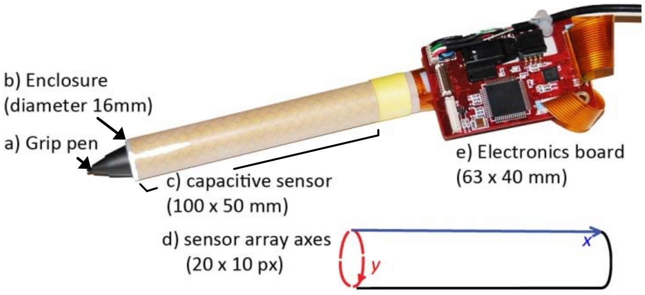


Figure 3. MTPen hardware prototype

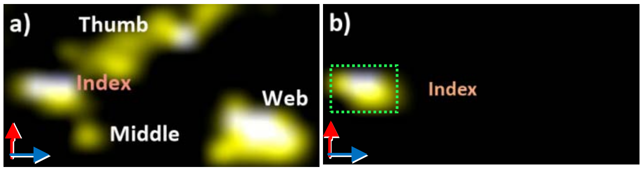


Figure 4. Touch and gesture sensing. (a) Raw sensor image captures the ambiguous hand parts. (b) We segment *the dynamic contacts* (Index) from the rest of the *static contacts*. Notice that the red arrow that corresponds to the longitudinal (x) axis and the blue arrow that corresponds to the circumference (y) axis of the pen from Figure 3.

The sensor is positioned 10mm from the tip of the pen, and covers an actual sensed area of 50mm x 100mm which is roughly 2/3 of the pen's barrel area. This is sufficient to sense many of the typical hand contacts while using the pen. The capacitive sensor is covered with a thin protective coating (architectural velum) to prevent direct electrical contact between the user's fingers and the sensor. We chose architectural velum as its surface properties made it comfortable to both hold the pen and to slide the fingers to perform gestures.

The sensor is capable of tracking multiple simultaneous contacts on a 20x10 grid of sensing elements printed on a flexible substrate using conductive ink. The raw capacitive sensor values are reported as 20x10 pixel grayscale images (Figure 1) at 100Hz, and processed to extract and track individual touch contacts. Note that the figures show interpolated grayscale images for clarity and smoothness; however, all processing is performed on the 20x10 grid.

We affixed the PCB (63mm x 40mm) at the rear end of the pen, which controls the sensor, captures raw capacitance values, and communicates these to a computer via USB. While the electronics board is exposed in our current hardware prototype, integrating it into the pen itself and using wireless communication should ensure symmetry and make the pen easier to handle.

Sensor Data Processing

Compared to the typical fingertip use of the traditional touch-screen, our multi-touch sensor captures contacts from different sections of the hand (e.g. finger tips and sides, web of the thumb, palm, etc.). As such, we share similar problems as described in the Mouse 2.0 project [26] in that our device needs to be held while interacting. Thus, any touch-gesture recognition needs to focus not only on specific contacts and their movement, but also to tolerate the existence of numerous additional contacts in order to support the user grasping the device.

In addition, due to the small size of the pen and the nature of the hand grip, contacts are closely grouped together and often merged into larger connected blobs. Hence, we altered standard computer vision touch processing techniques (e.g., Rekimoto's Holowall [17]) that determine separate connected components in the image and track them frame-to-frame.

In response to these challenges, we classify contacts into two categories based on their temporal characteristics: *static* and *dynamic*. The static contacts (Figure 4a) do not exhibit much movement and mostly remain of constant size and location. These static contacts are usually mapped to the parts of the hand sustaining the majority of the pen weight (e.g. thumb, middle finger, and web of the thumb). The dynamic contacts (Figure 4b) exhibit more movement and usually correspond to the agile finger (e.g. index, thumb) that is used for versatile purposes (e.g. gesturing, precision control, or pushing a button).

We use these two segmented images to detect grips and gestures on the surface of the pen. First, we segment out the dynamic contacts and process them to detect gestures. Then, we process the static sensor image to detect the handgrip.

$$\text{DynamicImage}(x,y) = \text{Signal}(x,y) - \text{Baseline}(x,y); \quad \textbf{(1)}$$
$$\text{Baseline}(x,y) = \text{Signal}(x,y) * \alpha + \text{Baseline}(x,y) * (1 - \alpha);$$

To obtain the dynamic image, we employ a decay function (Equation 1) over the entire image, which preserves only the moving elements, while the static contact areas are effectively removed. To compute this *DynamicImage*, the capacitive signal value is compared with the baseline value at each pixel location (x,y), which in turn is updated given a decay value (α = 0.03). We track the connected components in this decayed image and thus report only moving contact points for the gesture recognition.

This simple approach works well in practice to segment moving contacts from the rest of the hand while the user maintains a stable grip; however, when the user acquires the pen, puts the pen down, or simply repositions the pen in their hand, many dynamic contacts can temporarily occur on the surface. We detect these re-griping events by observing the size of the bounding box fitted around all the pixels present in the dynamic image (Figure 4b, green rectangle). During the grip changes, the size of this bounding box is fairly large since many contacts are moving (Figure 5). When only a single finger is moving, the bounding box is considerably smaller. By focusing on small overall movements, we can isolate the movement of the single finger and use that information to detect finger gestures regardless of how the user holds the pen. The overall amount of contact pixels (i.e., their pixel sum) can be used to easily detect when the pen is being held or not.

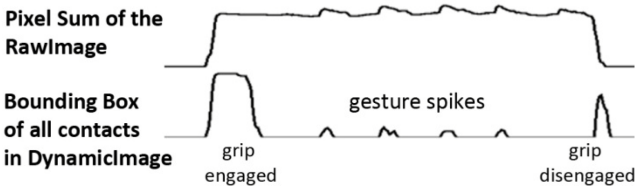


Figure 5.Temporal sequence of acquiring the pen, performing four finger swipes followed by the release of the pen.

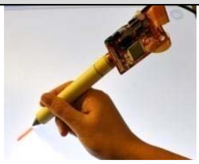
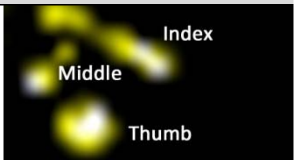
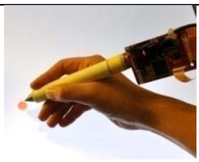
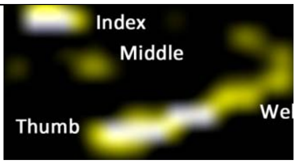
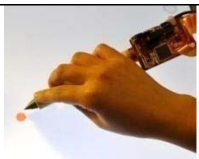
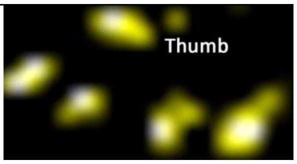
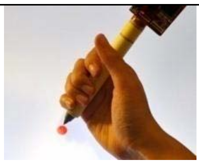
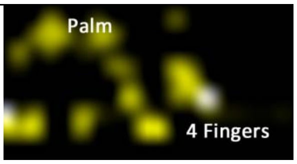
Grip	Hand Image	Sensor Image
Tripod		
Relax		
Sketch		
Wrap		

Figure 6. Different grips we recognize and their corresponding static sensor images.

Grip Recognition

To support the recognition of different hand grips, we track the entire shape of the hand using the static contact sensor image. When users are maintaining a stable grip, we match it against a grip image database using the Naïve Bayes classifier. Taylor and Bove have showed that this classifier provides a good trade-off between accuracy and speed of classification [24].

Since we need to detect hand grips regardless of how the user orients the pen in their hand, we extract features that are rotationally invariant along the pen's cylindrical barrel. The position (center of contacts) of contacts is not rotationally invariant, so instead we use features that include the number of the contacts, their sizes, orientation, and eccentricity in our recognition algorithm. We are currently able to disambiguate between four different grips: tripod, relaxed tripod, sketch, and wrap (Figure 6).

We used a traditional *training and testing* approach to gauge the effectiveness of our recognizer. We trained the recognizer for each individual user while adding more training images for faulty grips. We then collected data from ten pilot users who each provided 2000 recordings per grip. The recognition rate for four grips was 87% when using each user's training data individually. The wrap grip was the most problematic grip to recognize due to the great variability of the grip between grasps of the same user. Removing the wrap grip and reducing the set to three grips improves our overall recognition to 94% success rate.

Gesture Recognition

In addition to different grips, we recognize two types of finger gestures on the MTPen: *double tap* and *swipe*. Both gestures are simple to perform, and simple to detect in our dynamic contact image (as described previously).

Double tap: If two consecutive taps are reported within a threshold distance (d<5mm), and time span (t<200ms), they are recognized as a *double tap*. These thresholds were chosen after testing distinguishable thresholds on several pilot users with different hand sizes. Our recognizer also reports the location of each double tap.

Swipe: This gesture is performed when the user slides their finger along the barrel of the pen. The distance of the trail (> 0.5 mm), and the ratio of the width to the height of the bounding box (>3) are used to recognize a swipe. The recognizer reports the coordinate of the starting point, the current point, the direction, and the distance. As such, swipe can be used to control a simple mode switch or a continuous-parameter.

MTPEN INTERACTION TECHNIQUES

Combining our gestures and grips in interesting ways opens up a larger set of possible interactions for the MTPen users. We highlight these possibilities in a drawing application. Throughout these interactions, we assume that the base grip is the tripod grip, and note when other grips are required.

Automatic Focus Control using Engagement of the Pen

MTPen can detect whether the pen is being held or engaged. Observed in our interviews, designers frequently alternate back and forth between pen and other input devices (keyboard, mouse, etc). When MTPen detects that the user's hand is in contact with the pen, the input focus changes to the drawing area. When the user puts down their pen, the focus is passed onto non-pen-based interface components. Similar “engagement” interaction has previously been suggested by Hinckley and Sinclair [11].

Double Tap Color Picker

Once the double tap is detected, the color picker marking menu is brought up to easily change the color of the pen. Since changing the color of the ink is one of the most frequent tasks, we decided to map it to the double tap (Figure 7a). However, as double tapping is a simple mode switch, it can also be used to trigger other interactions such as mouse right click or bringing up other menu widgets.

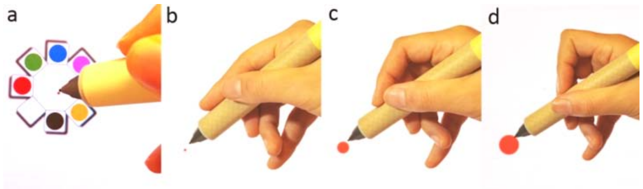


Figure 7. User controls mapped to finger gestures: (a) Control widget (e.g., color palette) pops up when double tap gesture occurs. (b-d) Continuous parameter control of the pen tip size using our swipe gesture.

Continuous Pen Size Control using Swipe Gesture

Our swiping gesture can also provide the capability to control continuous parameters. The users can change the size of the pen tip by moving their index or thumb not only during the hover state in between inking, but also between a PenDown and PenUp event. In this way a user can create a single stroke of varying width.

Although pressure of the pen tip is frequently mapped to the width of the pen stroke, users can neither explicitly control the parameter using the pressure nor lock it at a certain value. Comparatively, our approach provides a scroll wheel-like control that can be accessed regardless of where and how the user is holding a pen (Figure 7b-d).

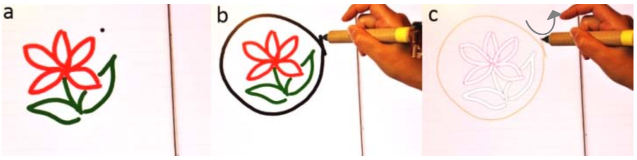


Figure 8. Swipe lasso selection: When the index finger (or thumb) swipes the barrel of the pen after a self-intersecting ink trace, the ink turns into a gesture ink, hence selecting the contents inside.

Swipe Lasso Selection

Another frequent operation that occurs in a drawing application is selection and repositioning of objects on the canvas. If a swipe gesture is detected after a self-intersecting ink stoke, the stroke turns into a lasso selection [10] (Figure 8), allowing the specified object to be moved on the canvas.

In this specific example we demonstrate how *ink strokes* can be combined with *finger gestures* (e.g. the swipe) to distinguish between a swipe meant to activate the resizing of the pen tip and one meant to lasso a selection: When the swipe occurs without a self-intersecting stroke, the magnitude of the finger swipe is used to change the pen tip width, while when the swipe occurs after a self-intersection, it is indicates a lasso selection.

Note that the finger gesture mode switch occurs after the pen is down. When a user swipes to change the current ink to become a selection ink, the mode is retroactively applied to the entirety of the stroke. Coordinating controls and inks at the same time is a well identified problem in pen-based interfaces [15], and our approach is similar to that of Hinckley et al. where the synchronization is relaxed between ink and a mode switch [9]. In our prototype, we relax the time period in which users can “mode” the current ink via finger gestures. Such design alleviates the burden of ensuring that every decision is made before starting their ink.

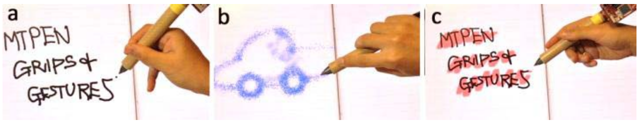


Figure 9. Pen selection based on grip: (a) Pen ink for the tripod grip, (b) paintbrush for the sketch grip, and (c) highlighter for the relaxed tripod grip.

Grip-based Pen Selection

In our painting demonstration application, we apply different pen grips to different pen types: tripod grip is mapped to a normal pencil, sketch grip is mapped to a paintbrush, and relaxed tripod grip is mapped to a highlighter (Figure 9). Thus the user can implicitly change modes, simply by changing their grip.

Page Selection using Wrap Grip and Swipe Gesture

Touch gestures on the pen’s barrel can be detected and used when the pen is not directly in contact with the tablet surface. One example of this is to perform page flipping (forwards and backwards) by holding the pen in the wrap grip outside of sensing range of the tablet and using the thumb to swipe against the top of barrel. Depending on the direction of the horizontal swipe, the notebook flips to the previous or next page (see Figure 10).

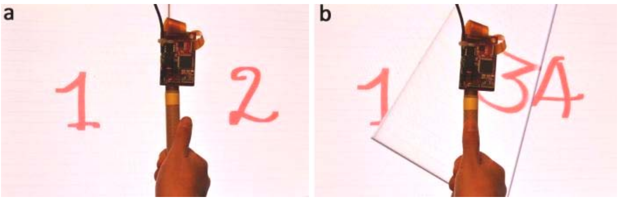


Figure 10. Page flipping interaction is triggered when user holds the pen in a wrap grip and performs a horizontal swipe (right to left in this image).

USER EXPERIMENT

In the previous section, we showed how our gestures and grips can be used to design novel interactions using the MTPen. We ran a formal experiment to test how *quickly* and *reliably* the gestures can be used for mode switching. In particular, we were interested in comparing our gestures to the conventional barrel button. In particular, we introduced a lower bound and an upper bound for the barrel button conditions based on our observation from earlier interviews. These confirmed that barrel button activation time should factor in the time to acquire a button, in addition to activating a button. We were also interested in the false positive and the false negative rate of our gesture recognizer. To answer these questions, we used a setting similar to Li [15] in which we created a pie crossing task that requires users to alternate between ink and command stroke to identify the cost of mode switching.

Mode Switching Techniques

We compare the swiping (SW) and the double tap (DT) techniques described above to two barrel-button baseline conditions. In the first baseline condition (BB1), the users switch the mode using a barrel button, but we do not force the user to acquire the button (i.e. users could have their finger directly over the button for the duration of the experiment). This is the standard setting used by Li [15].

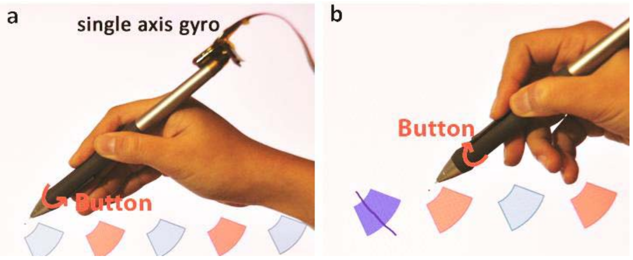


Figure 11.The BB2 condition requires rotation of the pen in order to acquire the button: (a) Button faces down when inking. (b) Button faces up in order to activate it. The arrows show the button location.

Second barrel button (BB2) condition reflects the cost of button acquisition (similar to homing in the KLM model [6]). In this condition users are forced to acquire the button before clicking it. To simulate 90–180 degrees rotation that occurs often before clicking the button, we require the users to place the button facing down as an initial setup. When we require the user to click the button, users must rotate the pen so that the button is facing up, to activate the button. To ensure the pen rotation in our trials, we installed a single axis gyro to the rear end of the regular pen (Figure 11). Our software required the user to rotate the pen by 120 degrees before the button could be activated.

In SW and DT condition, if users issue a finger gesture while inking, the gestural state is retroactive to the beginning of the stroke similar to our swipe-based selection technique. This recommendation was also supported by Li's study [15], which showed that relaxing synchronization between ink and mode switch improved the user's performance.

Compound and Baseline Task

Users were asked to complete two pie-crossing tasks (Figure 12). In the baseline task (Figure 12, top), the user does not need to switch modes, but simply cross through all five pie slices in the desired direction. When the corresponding stroke intersects both arcs (inner and outer) of each pie, the color becomes darker, allowing the user to progress onto the next slice.

In the compound task (Figure 12, bottom), the user needs to switch modes when progressing to each new pie slice. Participants were asked to cross blue slice with blue ink and red slices with red ink. In order to switch ink color, users have to change the mode using one of the four mode-switching techniques described previously. Blue ink is the default and the red ink is only available with a mode switch (a touch gesture on the MTPen or a barrel button press on a standard stylus). In the BB2 condition, users must rotate their pen before activating the button because blue slices must be crossed with the button facing down (Figure 11a), and red slices must be crossed with the button facing up while pressing the button (Figure 11b).

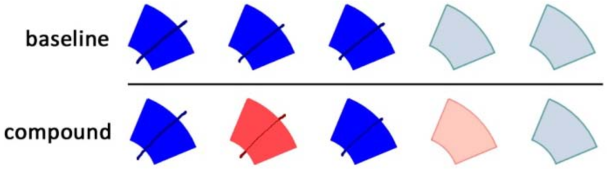


Figure 12.Baseline and compound pie crossing tasks are performed left to right. Successfully crossed slices are darker. Note: slices in this trial need to be crossed in NE direction.

To evaluate hand positions and wrist movement in different inking scenarios, each compound and baseline block is repeated in eight different directions (N, NE, E, SE, S, SW, W, and NW). A full set included all repetitions of both the baseline and the compound blocks in all eight different directions consisting of 80 pie crossings (2x5x8=80) in total.

Participants

Twelve participants (aged 22–35, right-handed, five male and seven female) were recruited from a university campus. Among them, one participant used a digital stylus on a regular basis, four participants claimed to have digital stylus experience in the past, while the rest had never used a digital stylus before.

Procedure

The study consisted of a training phase and an evaluation phase for each of the four techniques: BB1, BB2, DT, and SW. In the training phase, participants were asked to complete four sets of compound tasks. All participants completed the training set right before the evaluation phase for each of the conditions. The overall order of trials was counterbalanced using Latin square across users.

During the evaluation phase, participants had to go through four full sets (baseline and compound blocks in 8 directions). The first set was used to help users get used to the baseline and compound conditions, hence the data was discarded. The data from the next three sets was used for our analysis. The participants could take a break between sets. Since there are 16 mode switch cycles in a single set (eight directions, two mode-switch cycles per direction), users completed 64 mode switch cycles and 320 pie crossings (eight directions, two blocks, five pies per block) per condition (total of 1280 pie crossings in the evaluation stage).

Time Measurement

To give us a measure of how long it took our participants to mode switch in each of our conditions, we logged the timestamps of all pen-up events that occurred after successfully crossing a pie slice. Pen-up events are events generated when the pen loses contact with the surface.

Since the only difference between the baseline and compound tasks were the mode-switches, we computed the average mode-switching time by comparing the specific time intervals in both the compound and the baseline blocks. We illustrate this in Figure 13.

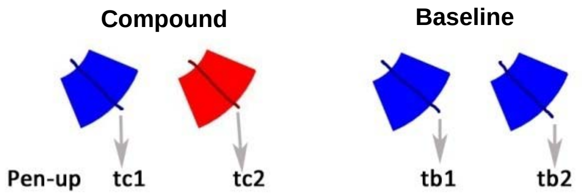


Figure 13. We measure the average time (ms) it took participants between two Pen-up events in both compound (tc2-tc1) and baseline tasks (tb2-tb1). The difference between these two timings is the mode switch time.

In essence, we measure the average time it took the participants to issue two specific pen-up events. In our example illustrated in Figure 13, that is accomplished by computing *tc2-tc1* for the compound task, and *tb2-tb1* in the baseline. The difference of those two timings provides the mode switching time. While Dillon [7] calculates the mode switching time using the pen-up of the previous stroke and the pen-down event of the current stroke, we use the pen-up events in both cases since our mode switching condition can occur before or during the entire stroke. Note that in the BB2 condition, the button acquisition time is captured within the net mode switching time.

Error Measurement

In addition to mode switching times, we logged two different types of errors: false negatives and false positives. The *false negatives* can occur only in the compound block when the mode-switch is not recognized (i.e. mode engagement failed). If a mode switch is detected in the baseline block, the gesture or the button is triggered accidently and is the cause of *false positives*.

Results

We analyzed our data using repeated measure ANOVA, and Bonferroni corrections were used for multiple comparisons.

Mode Switching Time Analysis

We first discuss the mode switching time. A one-way repeated measure ANOVA showed a significant main effect of technique ($F(3,33)=22.447$, $p<.001$, partial $\eta^2=.67$). Pairwise comparison revealed that the BB2 condition was significantly slower than all other conditions, while there were no significant differences in other conditions (**Error! Reference source not found.**). Overall, this result suggests that the MTPen gestures (especially SW) performed comparably to the best-case barrel button scenario (BB1) and drastically better than the sub-optimal use of the barrel button (BB2). Since BB1 (lower bound) and BB2 (upper bound) cover the two extreme ends of the pen button use spectrum, typical usage falls somewhere in between those two values.

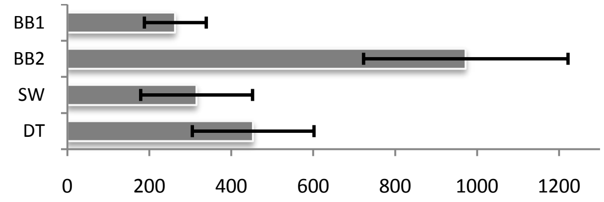


Figure 14. Mean mode switching time (ms) and confidence levels (α=.05) are shown in error bars.

Error Rate Analysis

We also analyzed the error rate. There was a significant main effect on *false negative errors* between different type of techniques ($F(3,33)=8.91$, $p<.001$, partial $\eta^2=.448$). Swipe (M=5.3%, SD=2.51%) and double tap (M=6.8%, SD=1.97%) suffered more from this error compared to both barrel button conditions (BB1: 0% and BB2: 1.3 %) mostly due to the gesture recognition errors.

In terms of the swipe gesture, some users could not maintain finger contact when the pen tip came into contact with the screen. Since the swipe gesture requires a continuous touch trace, this had a negative impact on performance. With the double tap gesture, the second tap would sometimes shift location with respect to the first tap, resulting failure to detect the gesture. However, increasing the distance threshold for double tap wasn't an ideal solution either; for some users, neighboring fingers of the index (thumb or middle) would be registered as a second tap because the threshold was too big.

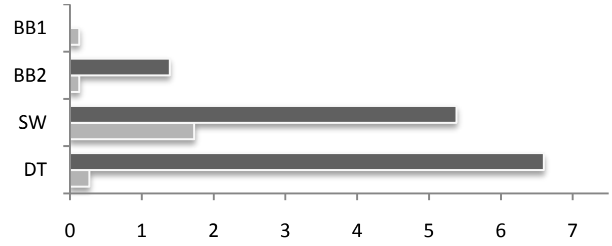


Figure 15.False negatives (black) and false positives (gray) for each condition. The values show percentages (%).

We also measured the *false positives* during the baseline block. Accidental double taps (1.7%) occurred more frequently than the swipe gesture (0.2%). Accidental double taps occurred when users mildly re-adjusted their entire hand for a better grip, contact points that are mapped to web of the thumb or the middle finger would rub against the sensor surface and trigger double tap. Similarly, we observed that for one user the web of the thumb would travel a long distance while readjusting the grip which triggered false positives of swipes. Figure 15 illustrates the false positives and false negatives errors.

Given the occurrence of false positive in the baseline condition, we analyzed the effect of each condition on the baseline performance. A one-way repeated measure ANOVA found a marginal effect of technique on the baseline time (F(3,33)=2.595, p=.069, partial η^2=.19) with BB1 (M=482.5, SD=142.18), BB2 (M=516.3, SD=112.0), SW (M=567.9, SD=152.2), DT (M=614.9, SD=152.1).This variation is partially attributable to a false positive trigger, which may have a slight impact on the mode switch time computation. However, several participants noted that the bulk of the prototype MTPen makes it more difficult to handle, which could account for some false positives. This issue equally affects the baseline and the compound conditions, and therefore does not influence the timing difference.

DISCUSSION

In this section, we reflect on the feedback from our study participants and from the four professional pen users that tried out our prototype drawing application.

Gestures

The ability to perform gestural mode switching on MTPen depends on a variety of ergonomic and technical issues. In terms of the ergonomics, the performance depends on how a user's hand is shaped (especially the index finger). For example, swipe gesture proved to be asymmetrical in terms of the swiping direction. Although we enabled swipe in both directions along the barrel, our participants used the flexion (curving the straight finger along the barrel) much more often than the extension (straightening the curved finger along the barrel).

When our professional designers used the swipe gesture to change the pen tip width, swipe gesture was better supported by our recognizer for the flexion than the extension. Since we modeled the contact points based on the centroid of their contact area, when the tip of the finger transitions to being flat as users extend their finger, the centroid actually travels in the opposite direction. As the result, it was harder to control the pen tip width during extension. From this observation, Wang's [28] touch point model that predicts the shape and orientation of the contact point was later added to replace use of the centroid with that of the tip of the finger. Our gesture recognizer also had limitations in detecting swipe when it was perpendicular to the pen's axis (e.g., in the page flipping interaction). As noted by Benko [3], the non-flat shape of the touch surface definitely presents challenges when doing touch interactions. In our case, the effects of the non-flat surface are especially noticed when using thumbs for the horizontal swipes. As thumbs are bound to the hand, they leave traces in an arc instead of straight lin.

We further discovered that non-finger part of the hand can also be used as a dynamic contact point: The possibility of using the web of the thumb as a dynamic contact point to control interactions was mentioned both during the user experiment and during designers' feedback sessions. While we did not design for it or demonstrate it to our users, some designers discovered this option and used it to control the pen width, instead of using their index fingers. Designers also mentioned that while transitioning between the tight tripod grip (Figure 2a) and a relaxed tripod grip (Figure 2b), the web of the thumb is bound to swipe across the barrel. As such, it is great way to capture squeezing movement of a hand which is yet another hidden dimension of the hand movement.

Hardware Design

When compared to the current industrial digital pens, MTPen's performance was generally hindered because of the larger diameter (16mm), extra weight, imbalance (from the electronics board and the wire), and its low-friction surface which negatively affected the stability required for gesturing. All of these issues are primarily due to the prototype nature of our device. However, even with such clear disadvantages, MTPen performed well in our user study despite that gesture-based input has higher error rate than mechanical input in general.

One of our expert designers also commented about extending sensor coverage area in both directions. The edge of the barrel, which starts tapering near the tip, plays a big role in pen interactions. If this front part was also covered with the sensor, finger tip movement would be better modeled. Similarly, extending the sensor to the rear end of the pen may open up opportunities for bimanual interaction with the device.

FUTURE WORK

Grip Recognition

We have started investigating ways to improve the grip recognition. In a different multi-touch hand-held device, adding an accelerometer has been shown to improve the grip detection by 10% [12, 21]. One of our preliminary experiments indicates that using additional sensor readings to reliably detect the rotation of the hand may improve grip recognition by reducing the task to a generic statistical pattern-matching problem. Lastly, combining our capacitive sensor with a resistive pressure sensor is another possibility [2] as each has its pros and cons in sensing hand movement. We observed, for example, that a resistive sensor [21] may be better at reliably detecting grips due to extra dimension that reflects the force of a grip.

Sensor Fusion

Another promising future work is to leverage the benefit of sensor fusion. In this work, our primary goal was to assess the benefit of a multi-touch sensor for recognition of grips and gestures. However, there is likely further contextual information to be gained through other sensors (e.g., roll [4], tilt [25], pressure [19], and motion [23]), that can be combined with the sensor image of a hand to enable new set of foreground interaction techniques. With respect to the sensor technology, we have already experimented with using a resistive sensor to sense pressure and enable new squeezing interactions.

Longitudinal Study

We also plan to extend the drawing application to observe how changes in grips and gestures affect the recognizer while users annotate and sketch. We intend to study the effect of false positives and false negatives of the recognizer on the users' pen-holding behavior.

Reaching beyond designers

This work was primarily motivated by our observations of the expert designers who use electronic pens on a daily basis in their workflow. Much work remains to be done to investigate specific interactions tailored to other domains. For example, MTPen can be combined with small slate computers, or a paper-based digital pen [16]. In addition, it may prove interesting to explore the use of MTPen in game scenarios where the pen would be used for writing or as a "magic" remote controller.

CONCLUSIONS

In this paper, we presented a new input device that combines a multi-touch surface and a digital stylus. We describe the technical challenges and solutions in detecting gestures and grips on our custom-designed prototype. Hand grips and touch gestures can be combined to support a rich set of interaction techniques. Lastly, our user experiments showed that using MTPen finger gestures provides a mode switch comparable to mechanical barrel buttons without their limitations. We believe that the lessons learned from enabling and testing our MTPen prototype will open a large area for design of novel interactions with a digital pen.

ACKNOWLEDGEMENT

This research was supported in part by the National Science Foundation under Grants IIS-0447703. We thank Adam Bender for his comments and Otmar Hilliges for the enclosure design. Tom Yeh helped us improve the grip recognizer. Last, we thank our study participants and anonymous CHI reviewers for their feedback.

REFERENCES

1. Apitz, G. and F. Guimbretiere. CrossY: A Crossing-Based Drawing Application. In Proc. *UIST'04*. ACM Press. p. 3 - 12
2. Baudisch, P. and G. Chu. Back-of-device interaction allows creating very small touch devices. In Proc. *CHI'09*. ACM Press. p. 1923-1932
3. Benko, H. Beyond flat surface computing: challenges of depth-aware and curved interfaces. In Proc. *Multimedia'09*. ACM Press. p. 935-944
4. Bi, X., T. Moscovich, G. Ramos, R. Balakrishnan, and K. Hinckley. An exploration of pen rolling for pen-based interaction. In Proc. *UIST'08*. ACM Press. p. 191-200
5. Brandl, P., C. Forlines, D. Wigdor, M. Haller, and C. Shen. Combining and Measuring the Benefits of Bimanual Pen. In Proc. *AVI'08*. ACM Press. p. 154-161
6. Card, S.K., T.P. Moran, and A. Newell, The keystroke-level model for user performance time with interactive systems, in *Communications of the ACM*. 1980. p. 396 - 410.
7. Dillon, R.F., J.D. Edey, and J.W. Tombaugh. Measuring the true cost of command selection: techniques and results. In Proc. *CHI'90*. ACM Press Press. p. 19 - 26
8. Grossman, T., K. Hinckley, P. Baudisch, M. Agrawala, and R. Balakrishnan. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In Proc. *CHI'06*. ACM Press. p. 861-870
9. Guimbretiere, F., Fluid Interaction for High Resolution Wall-size Displays. 2002, Stanford University. p. 157.
10. Hinckley, K., P. Baudisch, and G.R.F. Guimbretière. Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in Scriboli. In Proc. *CHI'05*. ACM Press. p. 451-460
11. Hinckley, K. and M. Sinclair. Touch-sensing input devices. In Proc. *CHI'99*. ACM Press. p. 223-230
12. Hinckley, K., K. Yatani, M. Pahud, N. Coddington, J. Rodenhouse, A. Wilson, H. Benko, and B. Buxton. Pen + touch = new tools. In Proc. *UIST'10*. ACM Press. p. 27-36
13. Jones, L.A. and S.J. Lederman, Human hand function. 2006: Oxford University.
14. Kim, K.-E., W. Chang, S.-J. Cho, J. Shim, H. Lee, J. Park, Y. Lee, and S. Kim. Hand grip pattern recognition for mobile user interfaces. In Proc. *Innovative applications of artificial intelligence*. AAAI Press Press. p. 1789-1794
15. Li, Y., K. Hinckley, Z. Guan, and J.A. Landay. Experimental analysis of mode switching techniques in pen-based user interfaces. In Proc. *CHI'05*. ACM Press. p. 461-470
16. LiveScribe, http://www.livescribe.com/
17. Matsushita, N. and J. Rekimoto, HoloWall: designing a finger, hand, body, and object sensitive wall, in *UIST'97*. 1997, ACM: Banff, Alberta, Canada. p. 209-210.
18. Napier, J.R., The prehensile movements of the human hand. Journal of Bone and Joint Surgery. **38**(B): p. 902-904.
19. Ramos, G., M. Boulos, and R. Balakrishnan. Pressure Widgets. In Proc. *CHI'04*. ACM Press. p. 527 - 534
20. Ravindra S. Goonetilleke, Errol R.Hoffmann, and A. Luximon, Effects of pen design on drawing and writing performance. Applied Ergonomics 2009: p. 292-301.
21. Rosenberg, I. and K. Perlin, The UnMousePad: an interpolating multi-touch force-sensing input pad, in *SIGGRAPH 2009*. 2009, ACM. p. 1-9.
22. Subramanian, S., D. Aliakseyeu, A. Lucero. Multi-layer interaction for digital tables. In Proc. *UIST'06*. p. 269-272
23. Suzuki, Y., K. Misue, and J. Tanaka. Stylus Enhancement to Enrich Interaction with Computers. In Proc. *HCII'07*. LNCS 4551 Press. p. 133-142
24. Taylor, B.T. and J. V. Michael Bove. Graspables: grasp-recognition as a user interface. In Proc. *CHI'09*. p. 917-926
25. Tian, F., L. Xu, H. Wang, X. Zhang, Y. Liu, V. Setlur, and G. Dai. Tilt menu: using the 3D orientation information of pen devices to extend the selection capability of pen-based user interfaces. In Proc. *CHI'08*. ACM Press. p. 1371-1380
26. Villar, N., S. Izadi, D. Rosenfeld, H. Benko, J. Helmes, J. Westhues, S. Hodges, E. Ofek, A. Butler, X. Cao, and B. Chen. Mouse 2.0: multi-touch meets the mouse. In Proc. *UIST'09*. ACM Press. p. 33-42
27. Wacom Technologies, http://www.wacom.com/index.html
28. Wang, F. and X. Ren. Empirical evaluation for finger input properties in multi-touch interaction. In Proc. *CHI'09*. ACM Press. p. 1063-1072
29. Wimmer, R. and S. Boring. HandSense: discriminating different ways of grasping and holding a tangible user interface. In Proc. *TEI'09*. ACM Press. p. 359-362
30. Wu, F.-G. and S. Luo, Design and evaluation approach for increasing stability and performance of touch pens in screen handwriting tasks. Applied Ergonomics, 2006. **37**: p. 319-327.
31. Zeleznik, R., A. Bragdon, F. Adeputra, and H.-S. Ko. Hands-on math: a page-based multi-touch and pen desktop for technical work and problem solving. In Proc. *UIST'10*. ACM Press. p. 17-26