Bimanual Input for Multiscale Navigation with Pressure and Touch Gestures

Sebastien Pelurson and Laurence Nigay
Univ. Grenoble Alpes, LIG, CNRS
F-38000 Grenoble, France
{sebastien.pelurson, laurence.nigay}@imag.fr

ABSTRACT

We explore the combination of touch modalities with pressure-based modalities for multiscale navigation in bifocal views. We investigate a two-hand mobile configuration in which: 1) The dominant hand is kept free for precise touch interaction at any scale of a bifocal view, and 2) The non-dominant hand is used for holding the device in landscape mode, keeping the thumb free for pressure input for navigation at the context scale. The pressure sensor is fixed to the front bezel. Our investigation of pressure-based modalities involves two design options: control (continuous or discrete) and inertia (with or without). The pressure-based modalities are compared to touch-only modalities: the well-known drag-flick and drag-drop modalities. The results show that continuous pressure-based modality without inertia is 1) the fastest one along with the drag-drop touch modality 2) is preferred by the users and 3) importantly minimizes screen occlusion during a phase that requires navigating a large part of the information space.

CCS Concepts

• **Human-centered computing~Interaction techniques** • *Human-centered computing~Graphical user interfaces* • *Human-centered computing~Mobile devices*

Keywords

Smartphone; multimodal/bimanual input; pressure input.

1. INTRODUCTION

1D data such as temporal information is increasingly used on smartphones. Examples include long lists of emails, contacts or music, financial data and even more recently health and wellness data. The need to handle and navigate such large information spaces in their entirety remains important and complementary to other query/recommender approaches. For the visualization of 1D/temporal information structure on smartphones, we adopt a bifocal display which is one type of Focus+Context display with two levels of detail. As illustrated in Figure 1, the intuitive layout of a detailed view (focus) and perspective/distorted panels on either side (context) is particularly suitable for maximizing the utilization of the available display area in landscape mode [12].

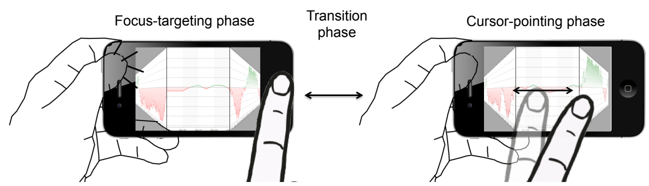


Figure 1: Interacting with a bifocal view of stock prices: Fast access to data at the context scale by using a pressure sensor with the non-dominant hand holding the smartphone (focus-targeting phase). Precise interaction at the focus scale using drag-flick gestures performed with the dominant hand (cursor-pointing phase).

Focusing on the generic task of multiscale navigation, there are two ways to interact as explained by Appert et al. [1]: accurate but slow navigation by interacting in the *focus space* (cursor-pointing phase in Figure 1) and fast but inaccurate navigation in the *context space* (focus-targeting phase in Figure 1). A movement of 1 pixel in the focus or context space corresponds to a movement respectively of 1 pixel at the focus space and of *mf* pixels at the focus scale, where *mf* is the magnification factor.

For multiscale navigation on smartphones in landscape mode, standard interaction using touch gestures (e.g. drag, flick) suffer two significant limitations. Firstly, the user will need to perform mode switches (via different types of gestures, different display areas for performing gestures or physical buttons) to perform navigation at the context scale or at the focus scale. Secondly, gestures will occlude a large part of the information space while navigating, seriously minimizing the added-value of the Focus+Context view.

The multimodal approach we propose alleviates these two problems. In terms of the CARE –Complementarity, Assignment, Redundancy and Equivalence- properties for multimodality [13], we assign a pressure-based modality to the non-dominant hand (NDH) for navigation in the context space and the standard touch modality to the dominant hand (DH) for navigation in the focus space. This bimanual approach is doubly consistent with Guiard's Kinematic Chain Model [4]: pressure-based modality with the NDH is used 1) before the DH 2) to define the frame of reference of the DH touch navigation. For fast navigation during the focus-targeting phase, we motivate a pressure-based modality using the NDH because it allows us 1) to use the NDH that holds the device in landscape mode for pressure input by fixing a pressure sensor to the front bezel (Figure 1) and 2) to avoid screen occlusion during a phase that requires navigating a large part of the information space.

We first give an overview of related work. Next we expose the two orthogonal design options explored for pressure input: control (continuous or discrete) and inertia (with or without). We then report on a laboratory study that compares the four designed pressure-based modalities with two touch modalities. For touch modalities we consider the standard drag-flick touch interaction as well as the drag-drop touch interaction that takes full advantage of the benefits of a Focus+Context visualization. The experiment shows that the continuous pressure-based modality without inertia and drag-drop touch modality are the fastest ones with no significant difference between them; Continuous pressure input without inertia is also faster than the well-known drag-flick touch interaction and preferred by the users. Considering that the pressure-based modalities minimize screen occlusion in comparison with touch modalities, these results show that the continuous pressure-based modality is a good candidate for multiscale navigation on a smartphone with minimal modification of hardware implementation.

2. BACKGROUND AND MOTIVATION

In the large body of literature on pressure-based interaction, we focus on input modalities on mobile devices (tablets and smartphones) for the task of navigation. But pressure-based modalities on mobile devices have been used for other tasks including text entry [2, 9] or menu selection [9, 18].

2.1 Single-Handed Navigation

The Force Gestures prototype [5] augments touch screen gestures using force by adding a sensor frame with 12 pressure sensors. The captured normal and tangential force enriches traditional touch gestures, while the authors report that some gestures including dragging cause fatigue. As for commercially available force-sensitive screens (e.g. on Apple and Huawei smartphones), such an approach does not prevent occlusion of the display, an important usability issue while navigating and exploring the displayed information space. We then decided to add pressure sensors on the bezel of the device in order to avoid touch interaction on screen that causes visual occlusion.

GraspZoom [10] uses a pressure sensor on the back of the phone used in portrait mode. This technique enables the user to scroll continuously by pushing operation and to switch direction by tiny thumb gestures on screen. This technique shows the feasibility of combining pressure-based modalities with touch modalities. From GraspZoom we reuse the method for defining the direction by touch and use pressure to control the navigation.

SidePress [14] uses two pressure sensors co-located on one side of the mobile device used in portrait mode. For scrolling a long document, the two sensors provide continuous rate-based control by capturing the pressure intensity and bi-directional navigation capabilities. The experimental comparison of SidePress with the standard drag-flick touch interaction shows that SidePress can be faster than touch for distant targets. While providing one-hand interaction in portrait mode, one limitation of SidePress is related to the need to switch between navigation and selection performed using touch: this requires users to change the hand grip in order to comfortably touch the screen with the thumb. Identified as future work in [14] for the landscape mode, one hand can be dedicated to navigation using pressure while the other hand performs touch gestures on screen. Our study explores this context of use with two-handed interaction (Figure 1) in landscape mode.

2.2 Two-Handed Navigation

Two-handed interaction is particularly suited when the device is used in landscape mode (Figure 1) [7, 8, 9, 12, 14]. The study performed by Pelurson and Nigay [12] on a smartphone compared 6 modalities (touch, tilt, pressure and peephole modalities) for fast navigation in a bifocal view, each modality being followed by drag-flick gestures for precise pointing. The study considered two pressure sensors co-located on one side of the mobile device as SidePress [14]. Since in [12] the mobile device was used in landscape mode, the two sensors were fixed on the top side of the device. The results showed good performance and subjective preferences for pressure-based interaction.

McLahlan and Brewster [8] also studied two-handed navigation using pressure and touch on a tablet. They explored pressure used as a NDH modality and touch as a DH modality. Although the results showed that a screen slider controlled by the NDH is faster than the pressure sensor, they conclude that splitting the control of scrolling speed (pressure, NDH) and scrolling direction (touch, DH) across two hands is a viable way to scroll on a tablet. MacLahlan et al. [7] further extended this work by studying another type of combination of the two modalities: pressure is used as a transient NDH modality, while touch is used as a DH modality to control the selection. As opposed to the previous study [8], pressure is used as an auxiliary input modality, supplementing but not replacing touch.

Previous work mentioned in this section provides useful insights for pressure-based interaction for navigation. First the users can use the pressure-based modality accurately enough for navigation using their DH on a smartphone [14] and their NDH on a smartphone [12] and on a tablet [7]. Second pressure can be controlled by the NDH holding the tablet, enabling touch interaction with the DH [7, 8]. We extend this work by focusing on multiscale navigation on a smartphone. In contrast to [7] we consider pressure as a main NDH modality for fast navigation (in the context space of the bifocal view) and touch as a main DH modality for precise navigation (in the focus space). More closely related to the bi-manual interaction study on tablet [8], we focus on interaction on a smartphone for which the user's hand posture and involved muscles for controlling pressure input can be different. Moreover we explore different parameters of the pressure-based NDH modality.

3. PRESSURE-BASED MODALITIES

3.1 Hardware Design

We build a prototype using an Arduino Micro Board and one pressure sensor Interlink Elektroniks Force-Sensitive Resistor (FSR) 400. We fix the pressure sensor to the front bezel (Figures 1 and 2). This position allows the user (1) to comfortably hold the smartphone and apply pressure with the non-dominant hand and (2) to interact on the touch screen with the dominant hand. We did not consider two pressure sensors fixed to the left and right bezels of the device in landscape mode since the transition phase (Figure 1) from the pressure-based modality to the touch modality will reduce the overall performance as shown in [12]. Furthermore with one pressure sensor, the holding position allows two-sided "grip" interaction with the thumb on front and the index finger on the back (Figure 1). This device pose works best for handheld pressure input [16]. We apply rate-based control for input pressure. Rate-based control allows for more precise, faster and less mentally demanding control of pressure input than positional control [14, 17]. Moreover positional control for navigation in a large 1D information space would imply a very large pressure range: it has been shown that above 8 or 10 levels of pressure, accuracy begins to decline [17].

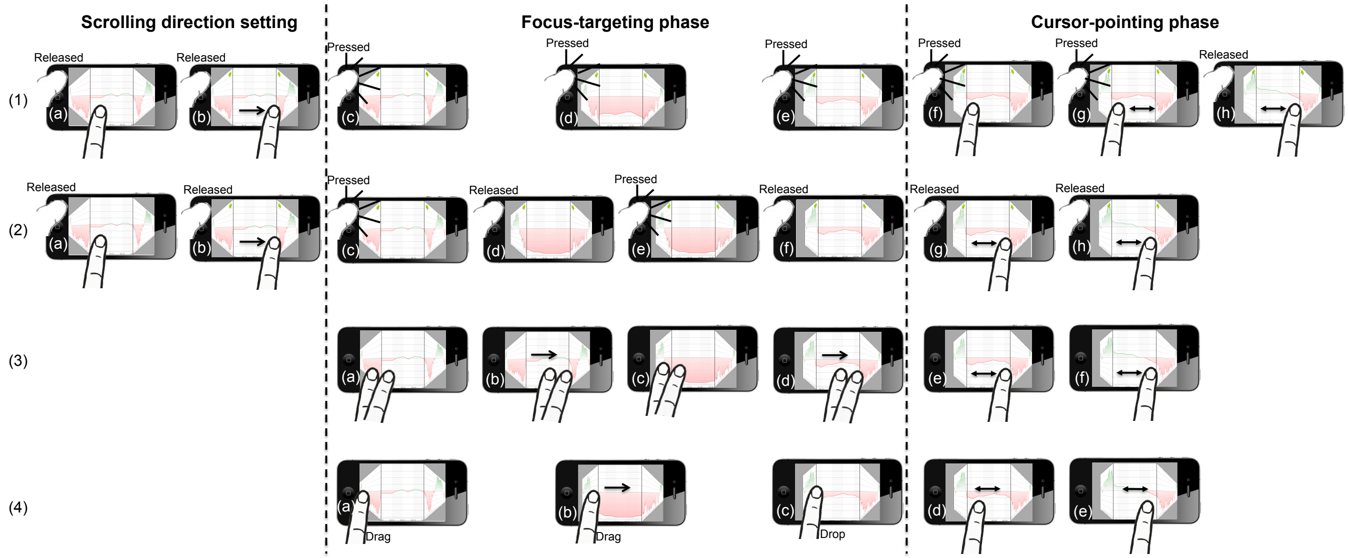


Figure 2: Examples of interaction paths.

Line (1): Continuous press walkthrough. (a-b) Swipe gesture to define the direction (c-e) Sensor pressed and continuous scrolling (motion at the context scale) (f) Sensor not released but touch stopped the scrolling (g-h) Precise drag-flick gestures performed everywhere on screen (motion at the focus scale).

Line (2): Discrete press walkthrough. (a-b) Swipe gesture to define the direction (c-d) Strong [press and release] provoking a long displacement in the dataset (motion at the context scale) (e-f) Light [press and release] provoking a short displacement in the dataset (g-h) Precise drag-flick gestures performed everywhere on screen (motion at the focus scale).

Line (3): Flick walkthrough. The non-dominant hand (not represented in the figure) is holding the device. Touch actions are performed with the dominant hand. (a-d) Two flick gestures with two fingers requiring clutching (motion at the context scale). (e-f) Precise drag-flick gestures with one finger performed everywhere on screen (motion at the focus scale).

Line (4): DirectTouch walkthough. (a-b-c) Drag and drop (motion at the context scale). (d-e) Precise drag-flick gestures with one finger performed in the focus area (motion at the focus scale).

The output voltage of the FSR does not linearly change with the applied force. For obtaining a linear mapping function for pressure input (better than a quadratic mapping function [16]), we adopt the solution of [16] also used for SidePress [14] by using an op-amp-based current to voltage converter.

3.2 Pressure-Based Interaction: Two Design Options

The pressure-based modalities are dedicated to fast navigation at the context scale of the bifocal view using the non-dominant hand, which also holds the smartphone. The pressure-based modalities are combined with touch-based modalities for precise interaction at the focus scale of the bifocal view (Figures 1 and 2). Moreover as in [8] for the case of a tablet, the scrolling direction is defined by a swipe gesture with the dominant hand (Figure 2 lines 1 and 2) and the scrolling speed is defined by pressure input with the non-dominant hand. As stated in the introduction, this combination of the pressure and touch modalities is consistent with Guiard's Kinematic Chain Model [4] of bimanual interaction.

Making the parallel between the pressure-based modality and the well-known touch modality (drag, flick) we consider two options for the design of the pressure-based modalities: Continuous/Discrete control and inertia.

Inspired by the drag gesture, a first design solution is to map the applied pressure to the scrolling speed for continuous rate-based control when navigating (Figure 2 line 1). The harder the users press the sensor using the non-dominant hand, the faster the scrolling speed is. Continuous visual feedback [14, 18] for pressure is provided and any touch on screen with the dominant hand will stop the navigation: this allows the users to control the pressure-based continuous navigation. Instead of continuous navigation we also explore discrete navigation. This implies discrete actions on the sensor and thus saves the users from maintaining a force on the sensor. The harder the users press the sensor, the larger the displacement is in the information space. For navigating, the users will perform a series of discrete [press-release] actions on the sensor applying different pressures according to the current distance to the target (Figure 2 line 2). The scrolling motion on screen starts when the user releases the sensor. This allows the users to control the pressure-based navigation. The series of actions on the sensor for discrete navigation is similar to the series of touch actions when performing flicking gestures with clutching. As explained by Nancel et al. [11], clutch-less movements are harder to perform. One possible cause is that clutching allows us to decompose a difficult movement into simpler controllable chunks. This parallel between pressure inputs and touch flicking/clutching actions led us to consider a second design option: adding inertia to pressure-based navigation, since inertia scrolling is provided when performing flick gestures with the touch modality Pressure-based navigation as drag-flick touch navigation can be enhanced by kinetic scrolling. The system then simulates inertia. The inertial scrolling motion is computed at the release of the pressure sensor and according to the last five pressure values before the pressure has reached 0. Any touch on screen with the dominant hand will stop the inertial scrolling.

The two design issues, namely control and inertia, define four pressure-based modalities. The next section describes their implementation.

3.3 Implemented Pressure-Based Modalities

The pressure-based modalities are developed on an iPhone 4 with a screen resolution of 960x640 pixels. The information space we used is 57600 pixel wide. This size matches with concrete application cases: for instance the visualization of a 10 year stock chart with a one-month period displayed in the focus: 480 pixels (focus size) * 10 years * 12 months = 57600 pixels (from [12]).

The drag-flick gestures take precedence over the pressure sensor as described by the state machine of Figure 3. Thus, users can stop the navigation triggered by pressure inputs by touching the screen. In the same way, the pressure sensor events are ignored if there is a finger on the screen when the users press the sensor (states “Pressure ignored” and “Pressure ignored&Inertia” in Figure 3).

3.3.1 Modality: Continuous

As soon as the sensor is pressed, the scrolling motion starts. Then, the harder the sensor is pressed, the faster the navigation is. When the sensor is released, the scrolling motion stops. The pressure/speed conversion is linear, allowing navigation speeds between 0 and 2800 pixels per second in the context space (respectively 0 and 33600 pixels per second in the focus space). This maximum speed has been defined after a pilot study comparing different gain factors for the linear mapping function.

3.3.2 Modality: ContinuousInertia

With the *ContinuousInertia* modality, when the sensor is released, an inertial effect is added. This effect consists of decelerating the scrolling motion over time. The velocity used for the inertial effect is the mean of the last five pressure values before the pressure has reached 0. The scrolling speed then decelerates over time until it reaches 0 using the following equation: $V_{t+1} = V_t \times 0.997^{dt}$ where *dt* is the time interval between two calculations (30ms). This equation has been established by conducting a pilot study comparing three values for the inertial effect: we chose the highest value for the inertial effect. Below 1 pixel per second, the scrolling stops. Thus, the faster the scrolling speed is when the sensor is released, the longer the deceleration phase is. At any time the inertial scrolling can be stopped with touch interaction (Figure 3).

3.3.3 Modality: Discrete

With the *Discrete* modality, nothing happens when the sensor is pressed, until it is released. When the sensor is released, the mean of the last five pressure values defines the displacement length in the dataset. The pressure/displacement conversion is linear and allows displacement from 0 to 11000 pixels. This enables the users to navigate through the 57600 pixel wide dataset with approximately 5 pressure gestures. This corresponds to the number of drag/flick gestures necessary to navigate the same distance with the touch modality (see description in section 4.1).

3.3.4 Modality: DiscreteInertia

As for the previous one, the *DiscreteInertia* modality does not trigger scrolling navigation until the pressure sensor is released. Rather than triggering a displacement in the dataset as for *Discrete*, *DiscreteInertia* creates an inertial effect. When the sensor is released, the mean of the last five pressure values is used as the initial velocity of the inertial scrolling. The deceleration equation is the same as for *ContinuousInertia*. The speed range is from 0 to 2800 pixels per second in the context space (respectively 0 and 33600 pixels per second in the focus space). If the users press and release the sensor again during the inertial deceleration phase, the velocities are cumulated: $V_{t+1} = V_t + V_p$,where V_p is the velocity provided by a pressure click (i.e. [press-release] on the pressure sensor) and V_t the current scrolling velocity., The inertial scrolling can be stopped with touch interaction at any time (Figure 3).

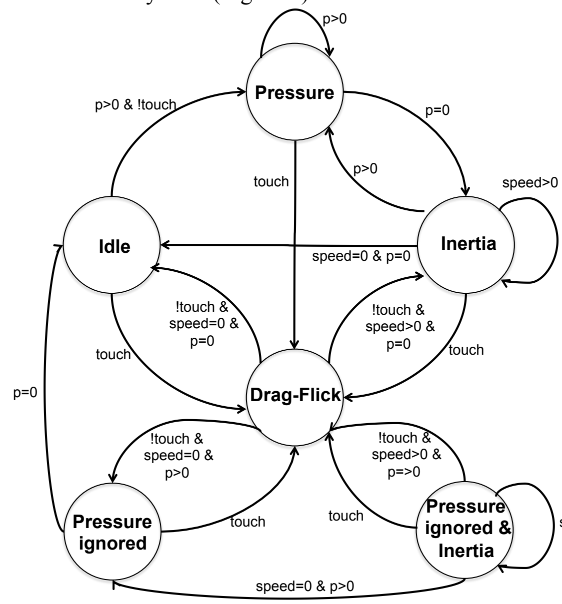


Figure 3: The four pressure-based modalities and their combinations with touch described by a single state machine.

4. EXPERIMENT

We conducted a controlled, within-subject experiment to compare the four pressure-based modalities (*Continuous*, *ContinuousInertia*, *Discrete*, *DiscreteInertia*) with two touch modalities that use the well-known drag-flick and drag-drop gestures (*Flick*, *DirectTouch*). The experiment enables us to investigate the following research questions that are related to the design rationale of the pressure-based modalities:

Q1: Can pressure-based modalities outperform touch modalities?

Q2: Is there a systematic advantage to using continuous pressure-based modalities instead of discrete ones?

Q3: What are the effects of inertia on focus-targeting and transition phases?

4.1 Baseline Touch Modalities

The first modality, namely *Flick*, uses drag-flick gestures in the context space with two fingers, to distinguish them from the drag-flick gestures in the focus space performed with one finger during the cursor-pointing phase (Figure 2 line 3). Thus, users can easily switch between the two modalities and perform flick gestures everywhere on the screen. A displacement with two fingers of 1 pixel on the screen causes a displacement of 1 pixel in the context space, and therefore a displacement of *mf* pixels in the focus space (*mf* being the magnification factor). The directions of gestures are those usually used on mobile phones with touchscreens: drag gestures to the left (respectively right) move data to the left (respectively right).

The second modality, namely *DirectTouch*, uses drag-drop touch actions. Users directly select and then drag a target in the context view to drop it into the focus view. By doing so the modality fully exploits the possibilities of a Focus+Context view. As drag gestures in the context area define drag-drop actions, users can only perform drag-flick gestures of the cursor-pointing phase in the focus area (Figure 2 line 4).

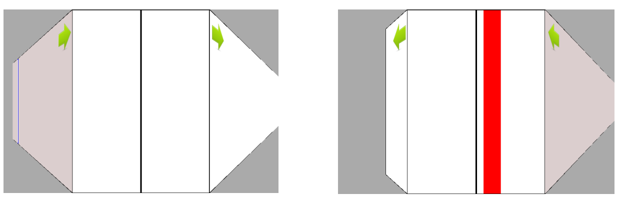
The two touch modalities define two different ways to distinguish between the two navigation phases (Figure 1): *Flick* is based on different gestures (one/two fingers) while *DirectTouch* involves similar gestures in different areas of the screen. Pressure-based modalities with the non-dominant hand enrich the vocabulary of events and enable us to assign a distinct modality per phase.

4.2 Participants & Apparatus

Eighteen unpaid volunteers (eight females), aged 15 to 38 years old (average 25, median 27), participated in the experiment. They were all experts in touch interaction on a smartphone since they were regular users of a smartphone with a touchscreen. We conducted the experiment on an iPhone 4 (with a screen resolution of 960x640 pixels) in landscape mode. Thus the program was fully iOS coded, except for a C program that receives pressure sensor values from the Arduino card and sends them to the phone using WiFi.

4.3 Experimental Task

As the main task using the bifocal view is to move points of interest into the focus area, we chose to study performance realizing pointing tasks. The task is illustrated by Figure 4 and is similar to [1, 12].



(a) (b)

Figure 4: Experimental task: The arrows indicate the scrolling direction. (a) The target is in the left panel of the context view. (b) The target is in the focus view, on the right of the cursor displayed at the center (black line).

Participants had to select targets in the context view. They had to put the target on a cursor displayed in the focus view and maintain it for 1 second: the red target became green when positioned on the cursor. After that, a new target appeared. Targets were generated to the left and to the right. Targets were displayed as 60 pixel wide red areas. The cursor was displayed at the center of the focus view as a 6 pixel wide black line (10 times smaller than the targets). Thus, targets can be easily selected, as the concrete task is to move interesting data points in the focus view in order to obtain details. Our goal is thus not to evaluate the accuracy of drag-flick modality in the focus view (cursor-pointing phase of Figure 1). We therefore make this phase easy to perform in terms of accuracy. Because of the perspective effect and the magnification factor, the target in the context view was not clearly visible. First, the background color of the context side containing the target was modified (Figure 4-a) in order to indicate where the target is and thus to minimize the desert fog effect [6]: participants always knew in which direction they had to navigate. Second, the target was surrounded by a blue rectangle in the context view to be always perceptible.

We used a fixed target width (W=60 pixels) and considered 4 different distances: D1=4800 pixels, D2=9600 pixels, D3=19200 pixels and D4=38400 pixels. Those distances in the focus space respectively correspond to 400, 800, 1600, and 3200 pixels in the context space. D1 is reachable with only one flick gesture in the context space. The other distances are simply D1 multiplied by 2, 4 and 8 (D2=2*D1, D3=2*D2, D4=2*D3). Thus these distances define four levels of task difficulty ($IDi=\log_2(Di/60 + 1)$), respectively ID1= 6.33, ID2=7.33, ID3=8.32 and ID4=9.32.

4.4 Procedure

The participants were sitting and holding the smartphone as shown in Figure 1. The mobile condition was not considered. First Wilson et al. [17] showed that pressure interaction remains usable while walking. Second navigating and exploring a long list of items corresponds to a task that the users rather do standing or sitting but not walking (e.g., list of emails, music or products of a shopping website while standing or sitting in the tramway back home).

We first explained the principle of the bifocal view by presenting to participants a concrete example with stock market data. Then, the evaluation was divided into 6 blocks, one per modality. The order of blocks was counterbalanced with a Latin square. For each block, we first described the modality, and participants had to realize 6 training tasks. They could ask any question during this step. They then performed 32 pointing tasks (8 tasks per ID). They finally completed a SUS questionnaire [3] on the modality they just experienced in order to collect the participant's subjective point of view on its usability. Lastly, after the 6 blocks, participants were asked to rank the 6 modalities in order of preference and to explain their choices. The entire experiment lasted approximately 45 minutes for each participant.

4.5 Time Analysis per Phase

For each modality, we define the focus-targeting phase as the time spent using the modality dedicated to fast navigation (at the context scale). For pressure-based modalities it starts when the users touch the screen to set the direction. For the modalities with inertia, it ends when the inertia animation ends or when the users touch the screen to use the precise modality. For modalities without inertia, the focus-targeting phase ends when the users remove their finger from the pressure sensor (for pressure-based modalities) or from the screen (for touch modalities).

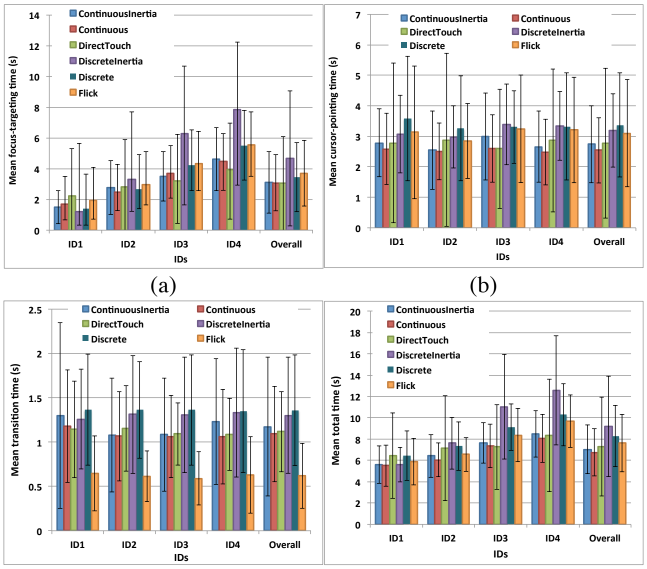
The cursor-pointing phase is defined as the time spent using the precise modality (at the focus scale). It starts when the users touch the screen and ends when they remove their finger from the screen (drag gesture) or when the inertia effect stops (flick gesture).

The transition phase is defined as the time spent between these two phases. Thus, if the users put their finger on the screen while the pressure sensor is pressed, the transition phase is null. Otherwise it is defined as the time between when the navigation triggered by the pressure sensor stops, and when the users put their finger on the screen.

4.6 Results

4.6.1 Performance: Execution Time

For each task, we logged all events triggered by users (touch and pressure). A post analysis of the generated log files allowed us to know which modality was used and for how long. We checked the normality of our data using the Shapiro-Wilk test. It revealed a right skew in the data distribution and none of the transformations (log, square root typically used with this type of deviation) normalized the data. We then performed the non-parametric Friedman test in order to test the significance effect of each factor (Modality and Index of Difficulty-ID) on each dependent variable (execution time for each phase) and of the factors combination (Modality x Index of Difficulty-ID) on each dependent variable. We also used Wilcoxon T test with Bonferroni correction for pairwise comparisons. We did not use a multi-factor analysis while our goal was to compare each of the six designed modalities with one another.



(a) (b) (c) (d)

Figure 5: (a) Focus-targeting mean time (b) Cursor-pointing mean time (c) Transition mean time (d) Total mean time (with 95% confidence intervals)

Focus-targeting phase time: The Friedman test revealed a significant effect of Modality factor on this phase time. ($\chi^2(5)=185$, $p<0.0001$). A Wilcoxon T pairwise comparison with Bonferroni correction showed that *Flick* (mean=3.70s) and *DiscreteInertia* (mean=4.66s) are significantly slower than all other modalities ($p<0.05$). *Discrete* (mean=3.45s) is significantly slower ($p<0.0001$) than *DirectTouch* (mean=3.04s), *Continuous* (mean=3.08s) and *ContinuousInertia* (mean=3.09s). *DirectTouch* is significantly faster ($p<0.005$) than *Continuous* and *ContinuousInertia*.

Regarding the ID factor (Figure 5-a), the Wilcoxon test showed that *DiscreteInertia* is significantly slower than all other modalities ($p<0.005$) for ID3 and ID4 while *DirectTouch* is significantly faster than all other modalities ($p<0.05$) for the same IDs. *Flick* and *Discrete* are significantly slower than *ContinuousInertia* for ID3 and ID4 ($p<0.05$) while they are significantly slower than *Continuous* only for ID4 ($p<0.0001$). For ID2, *DiscreteInertia* is significantly slower than all other modalities ($p<0.05$) except *Flick*. Finally for ID1, Discrete and DiscreteInertia are significantly faster than Continuous, Flick and DirectTouch ($p<0.05$).

Cursor-pointing phase time: The Friedman test revealed a significance effect of Modality factor on time ($\chi^2(5)=150$, $p<0.0001$). A Wilcoxon T pairwise comparison with Bonferroni correction revealed that *Flick* (mean=3.10s), *Discrete* (mean=3.36s) and *DiscreteInertia* (mean=3.19s) are significantly slower ($p<0.05$) than *DirectTouch* (mean=2.79s), *Continuous* (mean=2.74s) and *ContinuousInertia* (mean=2.54s). *ContinuousInertia* is significantly faster than *Continuous* ($p<0.005$).

Regarding the ID factor (Figure 5-b), the Wilcoxon test showed that *Discrete* is significantly slower than *DirectTouch*, *Continuous*, ContinuousInertia ($p<0.01$) for ID1 while *DiscreteInertia* is significantly slower than *DirectTouch* and *Continuous* ($p<0.01$) for the same distance. For ID2, *Discrete* is significantly slower than *DirectTouch*, *Continuous*, *ContinuousInertia* ($p<0.01$) while *DiscreteInertia* is significantly slower than *ContinuousInertia* and *Continuous* ($p<0.01$). For ID3, *DirectTouch* significantly faster than *ContinuousInertia*, *Discrete*, *DiscreteInertia* and *Flick* ($p<0.05$), while *Continuous* is significantly faster than *Discrete*, *DiscreteInertia* and *Flick* ($p<0.05$). Finally for ID4, *Discrete* and *DiscreteInertia* are significantly slower than *DirectTouch*, *Continuous* and *ContinuousInertia* ($p<0.05$). *Continuous* is significantly faster than Flick ($p<0.05$).

Transition time: The Friedman test revealed a significance effect of Modality factor on transition time ($\chi^2(5)=679$, $p<0.0001$). A Wilcoxon T pairwise comparison with Bonferroni correction showed that Flick (mean=0.62s) is faster than all other modalities ($p<0.0001$). DirectTouch (mean=0.83s), Continuous (mean=1.09s) and ContinuousInertia (mean=1.17s) are significantly faster ($p<0.0001$) than Discrete (mean=1.37s) and DiscreteInertia (mean=1.30s).

Regarding the ID factor (Figure 5-c) the Wilcoxon test showed that *Flick* is faster than all modalities for all IDs ($p<0.0001$). *Discrete* is significantly slower than *Continuous* and *DirectTouch* for ID1. For ID2 and ID3, *Discrete* is significantly slower than *DirectTouch* ($p<0.005$), *Continuous* and *ContinuousInertia* while *DiscreteInertia* is significantly slower than *Continuous* and *ContinuousInertia* ($p<0.05$). For ID4, *Discrete* and *DiscreteInertia* and significantly slower than *DirectTouch* and *Continuous* ($p<0.005$).

Total execution time: The Friedman test revealed a significance effect of Modality factor on total execution time ($\chi^2(5)=312$, $p<0.0001$). A Wilcoxon T pairwise comparison with Bonferroni correction showed that *DirectTouch* (mean=7.29s), *Continuous* (mean=7.02s) and *ContinuousInertia* (mean=6.74s) are significantly faster ($p<0.001$) than *Discrete* (mean=8.29s), *DiscreteInertia* (mean=9.20s) and *Flick* (mean=7.61s). *DiscreteInertia* is significantly the slowest modality ($p<0.0001$) while *Flick* is significantly faster than *Discrete* ($p<0.0001$).

Table 1: Summary of performance results: Ranking of modalities per phase

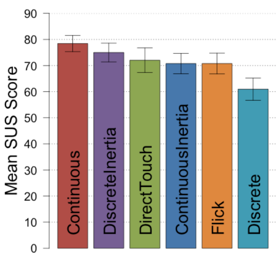
	Ranking
Focus-targeting phase	*DirectTouch > Continuous = ContinuousInertia > Discrete > Flick = DiscreteInertia*
Cursor-Pointing phase	*DirectTouch = ContinuousInertia > Continuous > Flick = Discrete = DiscreteInertia*
Transition phase	*Flick > DirectTouch = ContinuousInertia = Continuous > Discrete = DiscreteInertia*
Overall	*DirectTouch = Continuous = ContinuousInertia > Flick > Discrete > DiscreteInertia*

Regarding the ID factor (Figure 5-d), the Wilcoxon test showed that *Discrete* is significantly slower than *Continuous*, *ContinuousInertia* and *DiscreteInertia* for ID1 ($p<0.05$). For ID2, *DiscreteInertia* is significantly slower than *Flick*, *Continuous*, *ContinuousInertia* and *DirectTouch* ($p<0.01$) while *Discrete* is significantly slower than *Continuous* and *ContinuousInertia* ($p<0.001$). For ID3, *Discrete* and *DiscreteInertia* are significantly slower than all other modalities ($p<0.05$). *Flick* is significantly slower than *Continuous* and *DirectTouch* ($p<0.05$) while *ContinuousInertia* is significantly slower than *DirectTouch* ($p<0.05$). Finally for ID4, *Discrete* is significantly the slowest modality ($p<0.0001$), while *DiscreteInertia* and *Flick* are significantly slower than *DirectTouch*, *Continuous* and *ContinuousInertia* ($p<0.01$).

4.6.2 User Preference

User preferences do not entirely reflect the performance results. In Figure 6-a, the modalities are ordered from left to right with decreasing order of mean SUS score. A Friedman test revealed a significant effect of Modality factor on SUS score ($\chi^2(5)=15$, $p<0.01$). A Wilcoxon test with a Bonferroni correction showed that *Discrete* has been ranked significantly lower than *Continuous* and *DiscreteInertia* ($p<0.05$). All other modalities are not significantly different.

The preference ranking score of Figure 6-b is not entirely similar to the mean SUS score. The preferred modalities are the fastest ones, but are different from the three most usable modalities. We computed the ranking score (S) of a modality using the formula $S = 3*1^{st} + 2*2^{nd} + 3^{rd}$, where 1^{st}, 2^{nd} and 3^{rd} were its ranking at the corresponding place. We checked that with larger coefficients (respectively 5, 3, 1), the results remained similar.



(a)

Technique	#1	#2	#3	Score
Continuous	7	6	3	36
DirectTouch	7	2	3	28
ContinuousInertia	1	3	7	16
DiscreteInertia	1	5	2	15
Flick	2	2	3	13
Discrete	0	0	0	0

(b)

Figure 6: (a) Mean SUS score with 95% confidence intervals (b) Preference ranking scores and the number of times the modalities were ranked 1st, 2nd or 3rd.

4.7 Discussion

4.7.1 Pressure vs Touch

Continuous and *DirectTouch* were the fastest modalities and obtained the best ranking scores. In a different context of use and task, this confirms results of previous studies [7, 13] showing that pressure-based modalities can be as effective or outperform touch modalities (respectively *DirectTouch* and *Flick*) while minimizing screen occlusion.

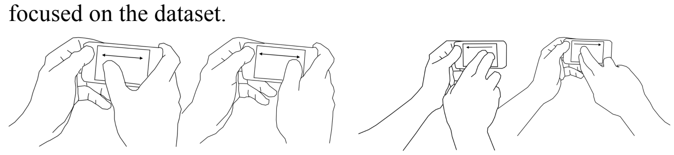
Flick has been described as uncomfortable. Indeed, smartphone users are used to interact with the touchscreen with their thumbs. This way, scrolling to the left or to the right simply involves thumb joint movements (Figure 7-a). The *Flick* modality involved using two fingers in order to perform fast navigation at the context scale. To do so, all the participants used the forefinger and the middle finger to interact. In addition most of them oriented their fingers in the scrolling direction: This forced them to perform a little wrist rotation (Figure 7-b) and created discomfort in using *Flick* reported by several participants. Moreover, we observed some participants having problems with touch recognition because of their nails as already reported by Spindler et al. [15]. Quite bad results obtained by the *Flick* can then be explained by this unusual position. Thus, we can conclude that the disambiguation approach using one or two fingers to perform *Flick* gestures really disturbs smartphone users and leads to bad performances. But constraining the users to perform flick gestures in particular areas of the screen (context and focus) would also lead to difficulties as we observed in a pilot study.

About the transition phase, the *Flick* modality was clearly shorter than the other modalities. However, participants have not perceived a significant enough difference worth commenting on. Nevertheless, discrete pressure-based modalities showed a longer transition phase. This can be explained by a disorientation effect due to the loss of control of the movements as explained by Pelurson and Nigay [12].

4.7.2 Continuous vs Discrete with and without Inertia

Continuous obtained better performances and was preferred to *ContinuousInertia*. *Discrete* obtained lower ranking score than *DiscreteInertia* despite better performances. We can then anticipate that a longer study enabling the users to better use the inertia effect may lead to different results, in particular for the discrete navigation mode. Indeed, *DiscreteInertia* seems to be a pressure version of the well-known drag-flick touch interaction and users provided positive feedback about it. Indeed, while some participants found the discrete pressure-based modalities cumbersome because of the repetitive pressure gestures they had to realize, some participants also reported that successive pressure gestures seem more "natural" than the continuous navigation mode. This can be explained by the fact that the participants are used to performing series of flicking and clutching when scrolling on their mobile phones, and discrete pressure-based modalities also involve a series of actions (i.e. series of [press-release] actions on the pressure sensor). Thus, some participants, because of the similarity with the Flick gesture, had appreciated using pressure sensors in a discrete mode. *DiscreteInertia* gave rise to good results for the shortest distance. This allows us to think that users are more comfortable with low-pressure levels which contradicts subjective results in [16].

Continuous and ContinuousInertia provided better results than Discrete and DiscreteInertia. Participants expressed that controlling the range of pressure values was difficult in discrete mode. Some of them said that they used those modalities randomly because they did not know which pressure they applied on the sensor. We think that this issue could be minimized after a longer learning period. The pressure control was easier with the continuous mode because of the continuous visual feedback. Continuous feedback mechanisms are highlighted as important in [14, 18]. One solution could be adding a visual feedback to indicate the applied force or the length of the jump, but this disturbs the user attention as reported in [5]. So we rather suggest adding a haptic feedback to the pressure sensor (improve hardware to better feel pressure applied). This would allow users to be aware of the pressure they are applying, while staying focused on the dataset.



(a) (b)

Figure 7: Flick modality: user's hand postures (a) with one finger and (b) two fingers.

4.7.3 Inertia vs No Inertia with Continuous and Discrete Mode

Unlike what we expected, inertia has not been very useful. It does not have any effect on the transition phase. One would expect that the inertia effect reduces this transition by providing a smoother focus-targeting phase ending. But performance results did not confirm this. Users were mitigated on its utility. On the one hand, most of them found that it increased the mental workload: they found it difficult to anticipate when releasing the sensor to better use the inertial scrolling. They thus had to be focused on this issue. On the other hand, several participants reported that the inertial scrolling allowed them to more easily switch to the precise modality (cursor-positioning phase) since they were not interacting with the device when the inertial scrolling was active. In addition most of the participants told us that they would probably be more effective with more experience. This is surprising since inertia is well-known by smartphone users. We performed a power law of practice on our data in order to study the learning curve of the modalities using inertial scrolling: we did not find any learning effect.

Moreover, the inertia had more effect on discrete pressure-based modalities than on continuous pressure-based ones regarding the focus-targeting phase. Indeed, *Continuous* and *ContinuousInertia* received similar performances while *Discrete* received significantly better performances than *DiscreteInertia*. Actually most of the users performed many low-level pressure actions, particularly for long distances (12.89 [press-release] actions average *for DiscreteInertia* and 9.78 average for *Discrete*), rather than fewer high-level pressure actions. As for the Flick modality, it allows an almost constant and quite slow navigation, but it gives the impression of a continuous navigation. This did not happen with the *Flick* modality because users are experienced using it. This usage of *DiscreteInertia* explains its bad performances being slower than *Discrete*. But this usage also decreases the number of overshoots due to the difficulty in controlling the discrete pressure-based modalities (see paragraph above). Indeed, there were more than twice as many overshoots with *Discrete* (0.45 average) than with *DiscreteInertia* (0.21 average) for long distances. Hence we think that a longer study should result in different performances for the modalities using inertial scrolling especially with *DiscreteInertia*.

5. CONCLUSION AND FUTURE WORK

Although many studies on mobile devices focused on one-hand interaction, there is still the need to interact with two hands and in particular in landscape mode. In this context of use, the laboratory study shows that continuous pressure interaction coupled with visual feedback with or without inertia (1) outperforms the well-known drag-flick interaction, (2) shows equivalent time performance with drag-drop interaction, while reducing screen occlusion for multiscale navigation in a bifocal view.

While continuous pressure input works well for navigation, we do not exclude discrete control of pressure for navigation. Discrete control implies a series of short pressure actions and simulates the well-known drag-flick using a pressure sensor. The laboratory study confirms previous results that advocate continuous feedback for pressure. This was observed both for discrete navigation for which the displacement is based on the applied pressure and for inertia based on the last applied pressure. Our on-going research focuses on discrete control of pressure with inertia 1) by providing tactile feedback to better feel the applied pressure 2) by starting the navigation while the sensor is pressed and by simulating the spiral plunger spinning top mechanism. Moreover, a longer study with realistic data and tasks will allow us to study if experience improves the performance. Concrete data could furthermore encourage users to use inertia to better explore and comprehend large information spaces.

6. ACKNOWLEDGEMENTS

We gratefully acknowledge the support of ANR (French National Research Agency ANR-11-LABX-0025-01), ANRT (French National Agency for Research and Technology) and the Streamdata.io software company.

7. REFERENCES

[1] Appert, C., Chapuis, O., and Pietriga, E. 2010. High-precision magnification lenses. In *Proc. of* CHI'10. ACM, 273-282.

[2] Brewster, S., and Hughes, M. 2009. Pressure-based text entry for mobile devices. In *Proc. of* MobileHCI'09. ACM, No. 9.

[3] Brooke, J. 1996. SUS: A Quick and Dirty Usability Scale. In *Usability Evaluation in Industry*. Taylor & Francis, 189-194.

[4] Guiard, Y. (1987). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behavior*, 19, 4 (1987), 486–517.

[5] Heo, S., and Lee, G. 2011. Force gestures: augmented touch screen gestures using normal and tangential force. In *Proc. of* UIST'11. ACM, 621-626.

[6] Jul, S., and Furnas, G. 1998. Critical zones in desert fog: aids to multiscale navigation. In *Proc. of* UIST'98. ACM, 97-106.

[7] Mclachlan, R., Boland, D., and Brewster, S. 2014. Transient and Transitional States: Pressure as an Auxiliary Input Modality for Bimanual Interaction. In *Proc. of* CHI'14. ACM, 401-410.

[8] McLachlan, R. and Brewster, S. 2013. A. Novel Modalities for Bimanual Scrolling on Tablet Devices. In *Proc. of* INTERACT'13, 229-246.

[9] McLachlan, R. and Brewster, S. 2015. Bimanual Input for Tablet Devices with Pressure and Multi-Touch Gestures. In *Proc. of* MobileHCI'15. ACM, 547-556.

[10] Miyaki, T. and Rekimoto, J. 2009. GraspZoom: zooming and scrolling control model for single-handed mobile interaction. In *Proc. of* MobileHCI'09. ACM, No. 11.

[11] Nancel, M., Vogel, D., and Lank, E. 2015. Clutching Is Not (Necessarily) the Enemy. In *Proc. of* CHI'15. ACM, 4199-4202.

[12] Pelurson, S., and Nigay, L. 2015. Multimodal Interaction with a Bifocal View on Mobile Devices. In *Proc. of* ICMI'15. ACM, 191-198.

[13] Serrano, M., Nigay, L. 2009. Temporal Aspects of CARE-based Multimodal Fusion. In *Proc. of* ICMI'09. ACM, 177-184.

[14] Spelmezan, D., Appert, C., Chapuis, O., and Pietriga, E. 2013. Side pressure for bidirectional navigation on small devices. In *Proc.* MobileHCI'13. ACM, 11-20.

[15] Spindler, M., Schuessler, M., Martsch, M., and Dachselt, R. 2014. Pinch - Drag - Flick vs. Spatial Input: Rethinking Zoom & Pan on Mobile Displays. In *Proc. of* CHI'14. ACM, 1113-1122.

[16] Stewart, C., Rohs, M., Kratz, S., and Essl G. 2010. Characteristics of pressure-based input for mobile devices. In *Proc. of* CHI'10. ACM, 801-810.

[17] Wilson, G., Brewster, S., Halvey, M., Crossan, A., and Stewart, C. 2011. The effects of walking, feedback and control method on pressure-based interaction. In *Proc. of* MobileHCI'11. ACM, 147-156.

[18] Wilson, G., Stewart, C., and Brewster, S. 2010. Pressure-based menu selection for mobile devices. In *Proc. of* MobileHCI'10. ACM, 181-190.