MultiScroll: Using Multitouch Input to Disambiguate Relative and Absolute Mobile Scroll Modes

Stephen Fitchett
Department of Computer Science
University of Canterbury
Christchurch, New Zealand
saf75@cosc.canterbury.ac.nz

Andy Cockburn
Department of Computer Science
University of Canterbury
Christchurch, New Zealand
andy@cosc.canterbury.ac.nz

ABSTRACT

We propose MultiScroll, a general purpose hybrid scrolling technique that uses multitouch input to allow for a combination of rate based scrolling for navigating short and medium distances and zero-order scrolling for navigating large distances. The design challenges of supporting both scrolling modes on mobile devices are discussed, including the use of 'drift zones' and 'edge proximity warnings' to resolve potential problems of touch controlled mobile rate-based scrolling. Evaluations with participants both stationary and walking show the complimentary benefits of the techniques over flick scrolling across a variety of scrolling tasks.

Categories and Subject Descriptors

H5.2 [**User Interfaces**]: Input devices and strategies (e.g., mouse, touchscreen)

Keywords

Mobile device, scrolling, multitouch

1. INTRODUCTION

Mobile devices are everywhere, and their capabilities are impressive: they have gigabytes of data storage, abundant applications, graphics processors, and high bandwidth wireless communications. In many ways, they are yesterday's desktop computer in your palm. But many of the interface controls for desktop computers are inappropriate or awkward in a mobile setting, raising new challenges for mobile designs. In particular, the small screenspace of mobile devices means that data spaces must be heavily partitioned, either through explicit hierarchies or through scrolling. It is therefore critical that mobile scrolling interfaces are effective.

Through decades of iterative refinement, desktop scrolling is well supported by a variety of on-screen widgets (e.g., the scrollbar) and by specialised input devices (e.g., mouse scroll wheels and trackpad scroll regions). Contemporary touchscreen mobile devices, however, cannot dedicate precious screen real estate to scrolling widgets, and external devices are unavailable.

Flick scrolling, as used on the Apple iPhone, is a popular scroll solution for mobile devices. Users pan the view by dragging their finger across the display. If the dragging is sufficiently quick, the view continues moving at the drag velocity when the finger is lifted from the screen, gradually decelerating due to 'friction' [2]. Researchers have proposed iterative refinements to flick scrolling that slightly improve its performance [1], but it is rarely the most efficient interface in a particular scenario. For example, tilt scrolling is faster for reading and analysis tasks while stationary [5], and scrollbars are faster for long documents [1]. Additionally, flick scrolling provides no shortcut for returning to an item in a well known location (such as a well known 'weather' link on a frequently visited webpage) – instead users must reproduce many flicks to gradually move through the page. We contend that flick scrolling is 'good enough' in most situations but the best in few.

To better understand the relative merits of different mobile scrolling techniques, and to explore the design space of refined scrolling designs, in this paper we design and evaluate a hybrid mobile scrolling technique for general purpose use. It uses multitouch to combine two complementary techniques – rate based and zero order – to work well in a wide variety of scenarios. We then evaluate the two components of this hybrid technique for reading and revisitation tasks.

2. DESIGN OF MULTISCROLL

Typical mobile scrolling tasks include looking for a piece of information on a webpage, reading an email, or finding a song or contact in a list. Each of these represent different methods of interaction; the first is an example of a visual search task, the second is an analysis task involving slowly processing an entire document and the third involves searching for an item in a structured setting in which the item's location can be partially predicted.

Designing a technique that is suitable for a diverse range of tasks is challenging. For visual search tasks, a desirable interface will allow for quick movement, a clear view of the document's contents and preferably random access. An example would be speed-dependent automatic zooming (SDAZ), which allows for variable scrolling rates while automatically zooming the document to reduce motion blur [8, 4, 9]. For reading and analysis tasks, users will typically slowly process a document. Smooth scrolling that does not interfere with the task is important, whereas quick scrolling speeds and random access are not. In fact, paging can be more efficient than line by line scrolling on mobile phones since overall less time is spent scrolling, even though each scrolling action can be quite disruptive [10]. Finally, interfaces that aid revisitation tasks must contain features to either explicitly return to previous locations or to provide cues as to where these locations are. A general purpose technique would ideally contain desirable properties for all types of tasks, but conflicts between them mean that this is often impossible.

Our design combines the benefits of rate-based scrolling for supporting smooth visual search activities with benefits of absolute scrolling for direct access to salient document regions. The two scrolling modes are disambiguated by using the multitouch sensing capabilities of most mobile input surfaces.

2.1 Rate Based Scrolling

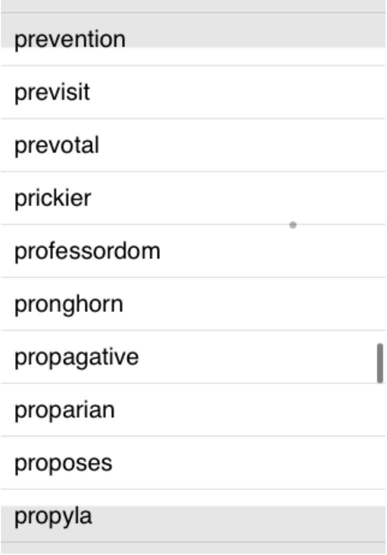
Rate based scrolling [14] (also called ‘Autoscroll’) allows the user direct control over the scroll velocity. It is supported in many desktop applications, and can typically be controlled by dragging with the middle mouse button. Our implementation works by first recording the location of the initial touch, and then calculating the scrolling speed based on the distance between the current touch location and the initial touch location, according to Equation (1):

$$s = (|p_1 - p_0|)^{1.3} \tag{1}$$

Here, s is the scrolling speed in pixels per second, p_1 is the current touch location and p_0 is the initial touch location. The view is scrolled in the opposite direction to Apple's flick scrolling implementation; moving the finger down scrolls the view down. An exponent of 1.3 was devised empirically to allow for both fine control with small distances and high scrolling speeds with large distances. In our prototype we supported only one dimensional scrolling, in effect ignoring the x components of p_0 and p_1, however this control method is easily scaled to two dimensions.

2.1.1 Drift Zones

A difficulty with this implementation is that if the initial touch location is close to the edge of the view, the scrolling speed is limited to a low speed in one dimension. For example if the user touches the display near the top of the view and wishes to scroll upwards, they can only move their finger up a small amount whilst keeping it on the display. To solve this, we introduce a new concept called drift zones. Drift zones appear at the top and bottom of the view and are displayed as subtle translucent rectangles, shown in Figure 1. When the current touch location is within a drift zone, the initial touch location is gradually moved away from the touched end of the view, up to a maximum of 25% of the view's height away from the opposite end of the view. The speed at which the initial touch location moves is based on how long the finger has been in the drift zone and increases over time. It is calculated using Equation (2), where t is the time in seconds since entering the drift zone and s is the speed at which the initial touch location is moving, in pixels per second.



prevention
previsit
prevotal
prickier
professordom
pronghorn
propagative
proparian
proposes
propyla

Figure 1: Rate based scrolling while active, with drift zones displayed at the top and bottom

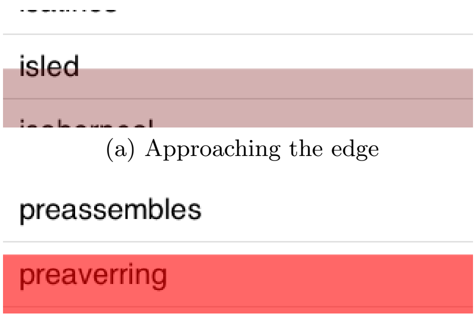
$$s = 25 + 10t \tag{2}$$

This technique allows for a distance between the initial and current touch locations of up to 345 pixels regardless of where the display was initially touched, which corresponds to a maximum scrolling speed of approximately 2000 pixels per second using Equation (1).

2.1.2 Edge Proximity Warnings

Mobile touch sensitive screens (such as the iPhone) do not distinguish being a finger being lifted off the device's display and the finger sliding off the edge of the display onto the bezel; both are interpreted as *touch ended* events. Similarly, sliding the finger from the bezel onto the display is interpreted as a *touch began* event. This is problematic due to the use of drift zones at the edges of the display: sliding the finger too far during a scroll operation will result in a *touch ended* event that stops the scrolling. Sliding the finger back onto the display does not rectify the problem as the initial touch location will not be in its original location. While other researchers have attempted to detect swipes onto the bezel (eg Bezel Swipe [13]), these techniques will never be completely accurate without additional hardware.

We alert the user to this problem by implementing proximity warnings when the touch location is close to the edge of the display during a scroll operation. When the finger is in the outer half of a drift zone (that is, the 20 pixels adjacent to the top and bottom of the display), the drift zones are shaded red to indicate edge proximity. The shading is graduated so that the closer to the edge the finger is, the redder the shading. A value p representing the proximity to the



(a) Approaching the edge

(b) Very close to the edge

Figure 2: Edge proximity warnings in the drift zones

edge is calculated using Equation 3, where y is the distance, in pixels, from the edge of the display. The proximity value, which is between zero (20 or more pixels from the edge) and one (right on the edge), is then used in Equation 4 to calculate the drift zone colour. Note that we are representing colours in the form {red, green, blue, alpha} with each component in the range $(0, 1)$. The edge proximity warning can be seen in Figure 2.

$$p = \left(1 - \min\left(\frac{y}{20}, 1\right)\right)^2 \tag{3}$$

$$c = \left\{p, 0, 0, 0.05 + \frac{p}{2}\right\} \tag{4}$$

2.2 Zero-Order Scrolling

While the rate based scrolling design described above is useful for scrolling short to medium distances, it is slow for long ones. Zero-order scroll control maps the finger position directly to the view position [14]; as the finger position moves, the view position changes proportionally. While Apple's flick scrolling implementation uses zero-order control when dragging a finger, the control-display gain is such that a single gesture can scroll the view by a maximum of the height of the view. We chose a control-display gain such that it is possible to scroll to any position of the document immediately: moving the controlling fingers to the top of the screen scrolls to the top of the document, and moving the fingers to the bottom of the screen scrolls to the end of the document. More generally, touching y% down the display scrolls y% down the view, much like directly controlling the scroll thumb in a scroll bar. This mode of scrolling is disambiguated from rate based scrolling by using two fingers. Although our implementation did not directly support multitouch zooming, the multitouch scroll method can be disambiguated from zooming by the relative movement of the two points of contact – when the two points move in opposite directions, the action is interpreted as a zoom.

Once activated, zero-order scrolling continues until all fingers are released, making it easier for the user to refine their location (necessary because the two fingers do not generally define exactly the same y positions). An undesirable consequence of this is that when releasing both fingers at a similar time, the view may scroll to the location of the last finger to be released. To avoid this problem, we implemented a 0.1 second delay before scrolling. If all fingers are released in this time, any scrolling done while doing so is cancelled.

3. EVALUATIONS

MultiScroll contains two techniques within it, disambiguated by movement with one or two fingers on the screen: rate based scrolling, primarily aimed at smoothly navigating short to medium distances (such as when reading a document), and zero-order scrolling, aimed at rapid long distance movements (e.g., finding an entry in a list). We evaluated these two interaction styles independently on an iPod Touch, using flick scrolling as the control.

3.1 Reading Evaluation

We compared our rate based scrolling technique to flick scrolling for reading tasks, with the participants both stationary and walking.

3.1.1 Participants and Apparatus

20 computer science students (four female) with a mean age of 25 participated in the evaluation. Eight had had previous experience with an iPod touch or iPhone. Participants were given a $10 shopping voucher for participating in the experiment.

The evaluation was performed on a second generation iPod touch running iPhone OS 2.2.1. The display's resolution was 480 × 320 pixels and it was always oriented in portrait.

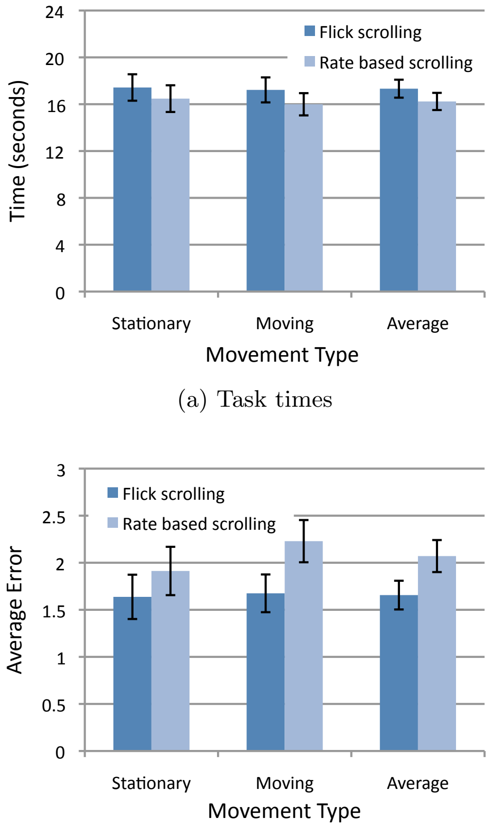
3.1.2 Procedure and Design

The experiment was designed as a 2 × 2 repeated measures analyses of variance (ANOVA) for factors *interface* (rate based and flick scrolling) and *movement type* (stationary and walking). Both factors were counterbalanced. Dependent variables were task times and error rates. Percentage preferred walking speed (PPWS) [11, 12] was also analysed for walking tasks.

Participants were given a brief introduction to the experiment before they carried out a preferred walking speed calibration to determine their normal walking speed by doing three laps, weaving between three chairs evenly spaced along an eight metre line.

Next, they were shown an example task. Tasks were based on Fitchett and Cockburn [5] and involved counting the number of occurrences of a cued five letter word within a text field made up of 120 five letter English words (typically 25 or 26 lines long). Although artificial, these tasks are intended to expose difference in the smoothness with which users can read and assimilate information from the interfaces. A smooth and continuous scrolling interface should allow users to count the words more accurately and faster than a jerky and hard to control one.

Text was displayed in left aligned 22 point Helvetica font. One target word occurred multiple times, with the number of occurrences following a Binomial distribution with $p = 0.05$ and $n = 120$, meaning that each generated word has a 5% probability that it will be the target word and a 95% chance that it will be different, resulting in an expected count of 6



(a) Task times

(b) Average error per task

Figure 3: Task times and error rates for the reading evaluation. Error bars show standard error.

target word occurrences (on average) per trial. This procedure results in random placement of the target words, and allows the possibility of clustering. After each trial, participants entered the number of times they counted the word. Once five trials were completed, they moved between walking and stationary conditions (counter-balanced), and then completed NASA Task Load Index (TLX) worksheets [6] for the interface. They then moved to their second interface (again, counter-balanced).

3.1.3 Results

Task Times. There was a significant main effect for *interface* ($F_{1,19} = 6.28, p < 0.05$), with rate based scrolling (mean: 16.2 seconds) faster than flick scrolling (mean: 17.3 seconds) in both stationary and moving tasks (see Figure 3a). There was no significant main effect of *movement type* and no significant *interface*×*movement type* interaction.

Errors. Error count, summarised in Figure 3b, is the difference between the participant's word count and the actual count. It showed a significant main effect for *interface* ($F_{1,19} = 6.26, p < 0.05$), with higher errors for rate based scrolling (mean: 2.1) than for flick scrolling (mean: 1.7). This indicates that participants chose a different speed-accuracy tradeoff when using rate based scrolling and suggests that it may be well suited to skim reading tasks where speed is valued over accuracy. There was no main effect for *movement type* nor a significant interaction.

Percentage Preferred Walking Speed (PPWS). There was a marginal main effect for interface ($F_{1,19} = 3.23, p = 0.088$), with participants walking slightly faster during rate based scrolling tasks (71.2% of preferred walking speed) than for flick scrolling tasks (68.7%).

Preferences. Participants were split over which interface they preferred. Overall, eight preferred rate based, nine preferred flick scrolling, and three were neutral. Five preferred rate based scrolling for both moving and stationary tasks, while seven thought the same about flick scrolling. Four preferred rate based scrolling for moving tasks and flick scrolling for stationary tasks, while another four thought the exact opposite. Finally, ten participants agreed that they would like support for rate based scrolling as the main scrolling technique in web browser and other applications on the iPod touch and iPhone, while six disagreed and four were neutral.

The main criticisms of rate based scrolling included difficulty in controlling the rate of scrolling and occlusion problems, although several participants stated that they thought they would improve with practice. Positive comments included that it was intuitive, easier to control, provided a consistent scrolling speed and required less thinking. Others commented that it was good for long documents and when just skimming, corroborating the above conclusions about the different speed-accuracy tradeoff for rate based scrolling. The mean NASA-TLX responses were similar for both interfaces on all measures, with no significant differences.

3.2 Recall Evaluation

Hinckley et al. [7] proposed a quantitative methodology for evaluating scrolling techniques that involves repeatedly navigating between two points in a document, varying the scrolling distance and the tolerance of scrolling. This methodology allows us to analysis which scrolling distances zero-order scrolling works well for as well as how precisely it can be used. We adapted it to suit the iPod touch interface and to reduce the time requirements and performed an evaluation comparing zero-order scrolling with flick scrolling.

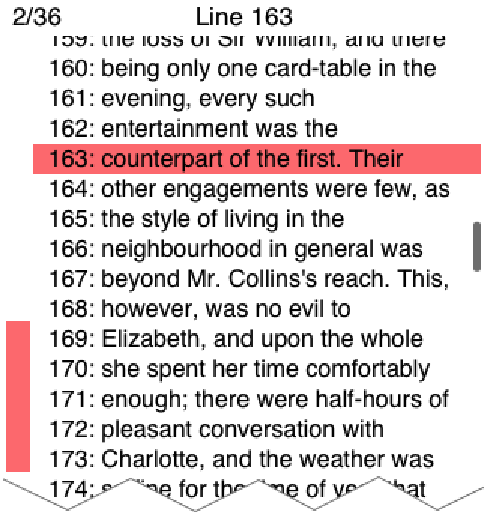
3.2.1 Participants and Apparatus

12 computer science students (three female) with a mean age of 25 participated in the evaluation. Five had had previous experience with an iPod touch or iPhone. Participants were given a $10 shopping voucher for participating.

The evaluation was performed on a second generation iPod touch running iPhone OS 2.2.1. The display's resolution was 480 × 320 pixels and it was always oriented in portrait.

3.2.2 Task

Tasks consist of a vertically scrolling view containing 600 lines of English text from a public domain book. The text within the view is rendered in 16 point Helvetica, allowing approximately 23 lines of text to be visible at any time. Lines are numbered to simulate scrolling in a familiar document, as in the original methodology and first conceived by



2/36 Line 163

159: the loss of Sir William, and there

160: being only one card-table in the

161: evening, every such

162: entertainment was the

163: counterpart of the first. Their

164: other engagements were few, as

165: the style of living in the

166: neighbourhood in general was

167: beyond Mr. Collins's reach. This,

168: however, was no evil to

169: Elizabeth, and upon the whole

170: she spent her time comfortably

171: enough; there were half-hours of

172: pleasant conversation with

173: Charlotte, and the weather was

174: s ne for th e of ve hat

Figure 4: The task interface for the zero-order scrolling evaluation (trimmed at the bottom). The user must try to align the target line (line 163) inside the frame (shown on left).

Buxton and Myers [3]. Tasks require participants to move back and forth between two target lines, which are highlighted in either red or blue for the first and second targets respectively. Only the next target is highlighted at any one time; on acquisition of a target the next target is highlighted and the highlighting for the current target is removed.

The scrolling view is 300 pixels wide and 456 pixels high, with the status bar hidden. A "frame" is shown on the left of the view, sized based on the tolerance of the particular task. The frame remains stationary while scrolling, with participants aiming to scroll until the target line is completely within the range indicated by the frame. The top 24 pixels of the display shows task information, including the line number of the current target line and the participant's progress through the experiment. The task interface is shown in Figure 4.

For timing purposes, task times begin as soon as the previous target is acquired, rather than when the first scrolling action takes place (although we record both times). This is done since there may be a difference in cognitive preparation time for scrolling between the two interfaces, for example zero-order scrolling could conceivably have longer preparation times if participants consider what position in the document they wish to scroll to before touching the display.

We differed from Hinckley et al.'s [7] original methodology in determining when a target was acquired. Rather than requiring participants to hit the caps lock key to confirm their selection, we automatically detected tasks' completion when three criteria were met: first, no fingers on the screen; second, the display is stationary (e.g., inertia is not moving the view with flick scrolling); third, the target is completely enclosed within the target frame. If just the first two criteria were satisfied but the target was on screen, this was classified as an error.

3.2.3 Procedure and Design

The evaluation was analysed as a 2 × 3 × 2 repeated measures analyses of variance (ANOVA) for factors *interface* (flick scrolling and zero-order scrolling), *distance* (D, the distance between targets, 20, 80 and 320 lines) and *target width* (W, 5 and 10 lines) with task time and error rate as dependent variables. The *interface* factor was counterbalanced, while the other factors were randomised.

For each interface, participants performed a block of practice trials followed by two blocks of timed trials. Each block consisted of a trial of each of the six combinations of scrolling distance and target width in random order. Trials consisted of six phases of reciprocal movement between two targets in practice blocks and 10 phases in timed blocks, with the first phase starting after scrolling to the first target and ending after scrolling to the second target. The first two phases of each trial were excluded from task time analysis. There were therefore 16 recorded phases for each distance-width-interface combination.

Participants began the experiment by providing basic demographic information and reading a brief overview of the evaluation. They were then given a demonstration of one interface. Next, they completed the one practice block and two timed blocks for the interface, as described above, before filling out NASA TLX worksheets. This was then repeated for the second interface. Participants were then asked several questions about comparisons between the two interfaces.

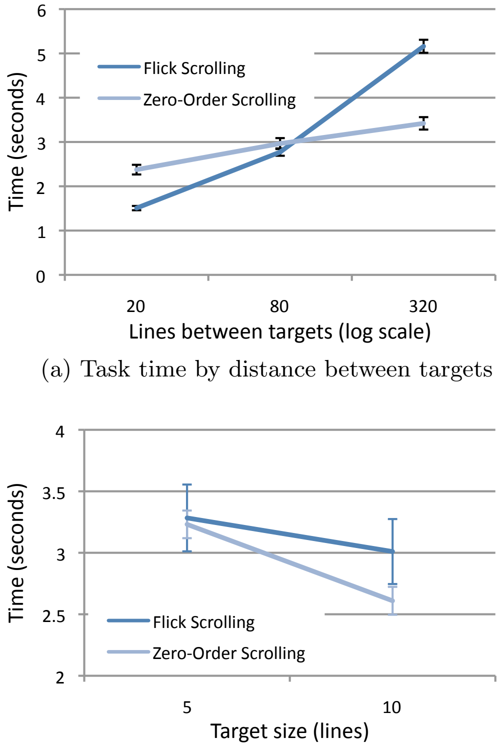
3.2.4 Results

Data for phases which had task times greater than three standard deviations away from the mean for the relevant *interface* × *distance* × *target width* combination were discarded as outliers. This amounted to 71 phases, or approximately 3.1% of all non-practice phases. The outliers were spread evenly across the factors.

Task Times. There was a main effect for *interface* ($F_{1,11} = 6.13, p < 0.05$), with zero-order (mean: 2.92 seconds) faster than flick scrolling (mean: 3.15 seconds). There were also significant main effects for *distance* ($F_{2,22} = 205.3, p < 0.001$) and *target size* ($F_{1,11} = 336.0, p < 0.001$), with longer distances and smaller target sizes respectively resulting in longer task times.

There was a significant *interface*×*distance* interaction ($F_{2,22} = 135.3, p < 0.001$), with faster task times for flick scrolling for short distances (20 lines) and zero-order scrolling for long distances (320 lines). Notably, times for flick scrolling increased greatly for larger distances, while they increased slowly with zero-order scrolling. This matched expectations since flick scrolling has a limited maximum speed, while zero-order scrolling allows users to take advantage of their knowledge about the location of the target and 'jump' direcly to it. For short distances, it was relatively simple for participants to move the short relative distance with flick scrolling, but with zero-order scrolling they had to also consider where they were in the document. Task times for different interfaces and distances are shown in Figure 5a.

There was also a significant *interface*×*target size* interaction ($F_{1,11} = 17.51, p < 0.005$). When the target size was 5 lines,



(a) Task time by distance between targets

(b) Task time by target size.

Figure 5: Task times by distance and target size for the recall evaluation. Error bars show standard error.

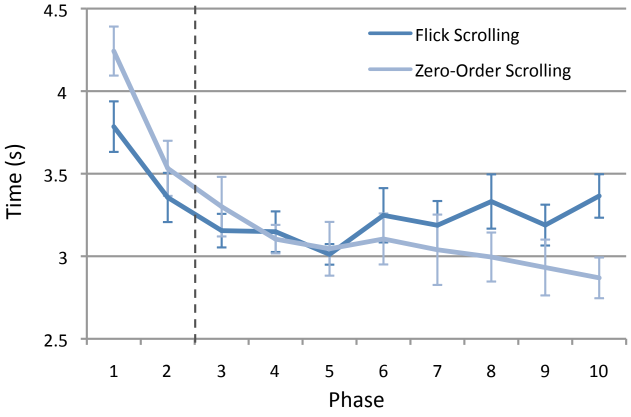


Figure 6: Task times by phase for the recall evaluation. Phase 1 corresponds to scrolling to the second target after acquiring the first target for the first time. Phases on the left of the dotted line were not included in analyses. Note the non-zero y origin. Error bars show standard error.

the two interfaces performed similarly on average. When the target size was 10 lines, zero-order scrolling performed better than flick scrolling. This result indicates that zero-order scrolling has a weakness for making precise selections and is more suited to scrolling to approximate locations. Task times for different target sizes are shown in Figure 5b.

Learning Effects. Average task times for each phase are shown in Figure 6, which clearly reveals a learning effect with zero-order scrolling as participants became familiar with the targets' locations. Zero-order scrolling times closely follow a power law curve ($R^2 = 0.92$), while those with flick scrolling do not ($R^2 = 0.32$). This can be explained with similar reasoning to the *interface*×*distance* interaction above; for flick scrolling, it is easy to reach the practical speed limit of scrolling, so even when the participant knew where the target was they could not get there any faster with practice. For zero-order scrolling, on the other hand, additional practice led to greater precision in estimating the target's location in the document, allowing for faster target acquisition.

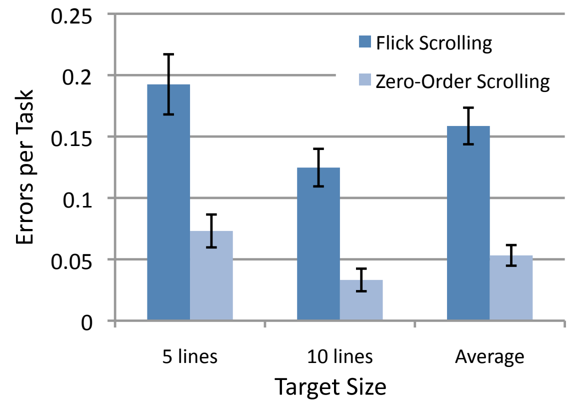
Although the *interface*×*phase* interaction is clear by the crossover effect in Figure 6, we also confirmed it by analysing the data as a 2 × 10 repeated measures analyses of variance for factors *interface* and *phase*. This shows a significant *interface*×*phase* interaction ($F_{10,110} = 3.53, p < 0.001$).

Error Rates. There was a main effect for interface ($F_{1,11} = 32.4, p < 0.001$), with zero-order scrolling resulting in fewer errors per task (mean: 0.053) than rate based scrolling (mean: 0.159). Caution should be taken interpreting this result, however, since there may have been false positives for flick scrolling. For example, if the view stopped scrolling near the target after a participant released their finger but before they placed it again, it would have been counted as an error, however it may just have been the participant being slow to start the next scroll action. On the other hand, it could be an indication of a real difference. This could potentially be explained by imprecision caused by uncertainty about how much scrolling will occur after a flick motion ends, or by greater care taken to correctly acquire targets with zero-order scrolling since the cost of an error is much greater than with flick scrolling.

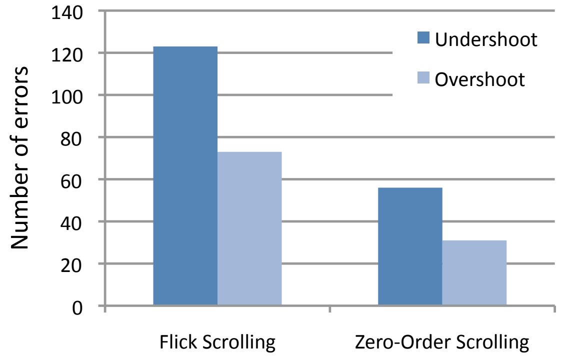
There was also a main effect for target size ($F_{1,11} = 11.29, p < 0.01$), with 5 line targets having a greater number of errors (mean: 0.133 per task) than 10 line targets (mean: 0.079). There was no main effect for distance and no significant interactions. Error rates for different interfaces and target sizes are shown in Figure 7a.

The type of each error, that is whether it was undershooting or overshooting the target, was also recorded. Figure 7b shows the frequency of each type of error for each interface. Undershoots were more common than overshoots for both interfaces. There was no apparent interaction between error type and interface, with undershoots being 63% of errors for flick scrolling and 64% of errors for zero-order scrolling.

Preferences. Of the 12 participants, nine thought that zero-order scrolling was quicker for getting to the approximate area of a target, while three thought the same about flick scrolling. All but one participant thought that flick scrolling was better for making precise selections. These preferences confirm the previously discussed *interface*×*target size* interaction, in that zero-order scrolling can be used to quickly get near a target but is not as good as flick scrolling for precise selections. Overall, nine participants preferred flick scrolling and three preferred zero-order scrolling.



(a) Error rates per task. Error bars show standard error.



(b) Frequency of error types.

Figure 7: Error rates and types for the recall evaluation

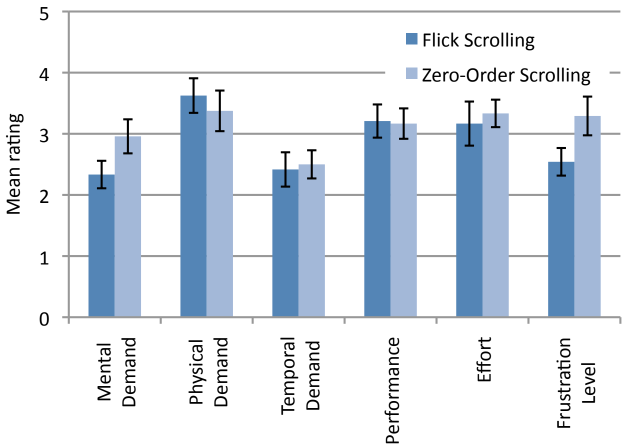


Figure 8: Mean NASA-TLX responses for the recall evaluation. Lower numbers are better except for performance. Errors bars show standard error.

Participants were also asked if they would like zero-order scrolling to be the main scrolling technique in web browser and other applications on the iPhone. Eight participants disagreed, while two agreed. Participants were then asked if they would like zero-order scrolling to be available in combination with a relative technique in these applications, and eight agreed (one disagreed).

The most common comment from participants was that they could not refine their position after releasing their finger when using zero-order scrolling. It was described as "very fiddly to control" by one participant and another commented that "flick scrolling felt a lot more natural". Many of the participants' issues with zero-order scrolling would be rectified if used in combination with a rate based approach. At the other end of the spectrum, several participants noted that zero-order scrolling was useful for scrolling long distances, with one even describing it as "fun" in such cases.

The mean NASA-TLX responses, shown in Figure 8, were similar for flick scrolling and zero-order scrolling for most measures, although flick scrolling was significantly better than zero-order scrolling for mental demand (Wilcoxon $z = 1.66, p < 0.05$).

4. DISCUSSION AND FURTHER WORK

We have shown that rate based scrolling achieves better task performance than flick scrolling for reading tasks, at the cost of a slight loss in accuracy, suggesting that rate based scrolling is well suited to skim reading tasks. When rated subjectively, rate based scrolling is competitive with flick scrolling. These results are promising considering that most participants had been previously exposed to flick scrolling, either in earlier evaluations or from past experience using an iPod touch or iPhone.

The recall evaluation for zero-order scrolling confirmed that the technique improves task performance when acquiring targets in known locations that are far away. However, it also highlighted its weaknesses for scrolling short distances and precisely selecting targets.

These results confirm our hypothesis that rate based scrolling and zero-order scrolling suit different, complementary tasks and a hybrid approach is worth pursuing. Adding rate based scrolling to a zero-order scrolling interface, for example, solves the problems zero-order scrolling has for scrolling short distances and for refining the current location. Alternatively, zero-order scrolling could be combined with flick scrolling, which is perhaps better suited for making precise selections than rate based scrolling, while rate based scrolling is best suited to slightly longer distances than flick scrolling. Further evaluations would be needed to compare hybrid approaches to standalone techniques and to find which hybrid approach works best.

Additional further work may involve creating a two dimensional version of both rate based scrolling and zero-order scrolling. Since they currently use only one dimension for input, this should be relatively simple to accomplish. We can extend zero-order scrolling to map both the horizontal and vertical touch locations onto horizontal and vertical positions in the document. Additionally, for rate based scrolling we can take both the x and y offsets from the touch origin to determine the scrolling speed. Drift zones could be added to the left and right of the display, and moving the finger into a drift zone would move the touch origin away along the line between the current touch position and the touch origin. In both cases, dimensional stability should be considered; if a user wants to scroll down, for example, they are unlikely to be able to be precise enough to keep their finger in the same position horizontally, resulting in some unwanted horizontal scrolling. For rate based scrolling, this could be rectified by implementing threshold angles, such that the view only scrolls in a dimension if the movement from the touch origin in that dimension is sufficiently large relative to the movement in the other direction. For zero-order scrolling the problem is not as simple to solve since this technique would result in large jumps when the threshold is first exceeded, however a similar approach based on it could potentially be used.

5. CONCLUSIONS

We have designed and implemented MultiScroll for mobile devices, which contains a rate based scrolling technique as an alternative to flick scrolling and a zero-order scrolling technique that allows rapid shortcut scrolling to any point in the document. Design adaptations for controlling rate based scrolling on a touchscreen are described, including drift zones and edge proximity warnings. MultiScroll distinguishes between rate based and zero-order scroll modes by using the multitouch capabilities of contemporary touchscreens, with one finger movements controlling rate based scrolling and two finger movements controlling the zero-order absolute document position. Evaluations showed that the rate based scrolling technique is comparable with flick scrolling and that it is well suited to skimming tasks. They also showed that the zero-order technique is well suited for revisiting distant targets, outperforming flick scrolling. The results lend support to a hybrid approach using multitouch that gives the benefits of both techniques.

6. ACKNOWLEDGEMENTS

Thanks to the evaluation participants for their time and the anonymous reviewers for their comments. This work was partially funded by New Zealand Royal Society Marsden Grant 07-UOC-013.

7. REFERENCES

[1] D. Aliakseyeu, P. Irani, A. Lucero, and S. Subramanian. Multi-flick: an evaluation of flick-based scrolling techniques for pen interfaces. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1689–1698, New York, NY, USA, 2008. ACM.

[2] Apple Inc. iPhone How To – Scrolling. Available online: `http://www.apple.com/iphone/how-to/index.html#basics.scrolling`, last accessed 10 March 2010.

[3] W. Buxton and B. Myers. A study in two-handed input. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 321–326, New York, NY, USA, 1986. ACM.

[4] P. Eslambolchilar and R. Murray-smith. Tilt-based automatic zooming and scaling in mobile devices - a state-space implementation. In *Proceedings of Mobile HCI 2004*, pages 120–131. Springer, 2004.

[5] S. Fitchett and A. Cockburn. Evaluating reading and analysis tasks on mobile devices: A case study of tilt and flick scrolling. In *OZCHI '09: Proceedings of the 21st Australasian Conference on Computer-Human Interaction*, pages 225–232, New York, NY, USA, 2009. ACM.

[6] S. Hart and L. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In P. Hancock, editor, *Human Mental Workload*, pages 139–183, 1988.

[7] K. Hinckley, E. Cutrell, S. Bathiche, and T. Muss. Quantitative analysis of scrolling techniques. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 65–72, New York, NY, USA, 2002. ACM.

[8] T. Igarashi and K. Hinckley. Speed-dependent automatic zooming for browsing large documents. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 139–148, New York, NY, USA, 2000. ACM.

[9] S. Jones, M. Jones, G. Marsden, D. Patel, and A. Cockburn. An evaluation of integrated zooming and scrolling on small-screens. *International Journal of Human Computer Studies*, 63(3):271–303, 2005.

[10] G. Öquist and K. Lundin. Eye movement study of reading text on a mobile phone using paging, scrolling, leading, and rsvp. In *MUM '07: Proceedings of the 6th international conference on Mobile and ubiquitous multimedia*, pages 176–183, New York, NY, USA, 2007. ACM.

[11] H. Petrie, S. Furner, and T. Strothotte. Design lifecycles and wearable computers for users with disabilities. Presented at The First Workshop on Human-Computer Interaction With Mobile Devices, 1998.

[12] A. Pirhonen, S. Brewster, and C. Holguin. Gestural and audio metaphors as a means of control for mobile devices. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 291–298, New York, NY, USA, 2002. ACM.

[13] V. Roth and T. Turner. Bezel swipe: conflict-free scrolling and multiple selection on mobile touch screen devices. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1523–1526, New York, NY, USA, 2009. ACM.

[14] S. Zhai, B. A. Smith, and T. Selker. Improving browsing performance: A study of four input devices for scrolling and pointing tasks. In *INTERACT '97: Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction*, pages 286–293, London, UK, UK, 1997. Chapman & Hall, Ltd.