ActiWare: Activity-aware Scrollbar for Display of Dynamic Interactions in Social Platforms

Osama Askoura
Electrical Engineering and Computer Science Department
York University, Toronto, Canada
oaskoura@cse.yorku.ca

We introduce activity-aware scrolling (ActiWare Scrolling), an approach that displays live activity on dynamic pages as they happen. ActiWare Scrolling makes use of the intrinsic display feature of the native scrollbar to indicate user interactions outside the current view. We present two versions of this scrollbar: clickable and screening. The screening version displays an indication for the user to scroll to; the clickable version allows the user to click on it for immediate teleportation. Our results show that users are able to save 95% of scrolling mileage and 50.9% in time finding recent activity using the clickable scrollbar - over the native scrollbar. The screening scrollbar also outperformed the native scrollbar with a 54.4% saving in scrolling mileage and a 40.5% saving on time.

1. INTRODUCTION

The modern scrollbar is a soft control element. It is a convenient tool for navigating large documents on small screens. This is because of its intrinsic feature as a display and a control. As a control, it allows for changing the view and location in a document or window. As a display, its size indicates the view size relative to the entire document. And its position indicates the view location in the document. Figure 1 illustrates a typical scrollbar and the notion of control-display.

![Figure 1: Anatomy of a typical scrollbar. The trough shows the scrollable region. The thumb indicates the current position and the size of the visible portion relative to the size of the document. The arrows move the thumb up and down in a fixed step-size.](image)

This ability of the scrollbar to give a birds-eye view of an entire document was exploited by Byrd [3] to visualize the presence of query matches in the context of text-searching. McCrickard and Catrambone [1], and Hill, et al. [5] have also expanded on this intrinsic feature to display the document’s text density in the scrollbar region. Video scrubbing – as experienced on Youtube - also exploits the video’s timeline sliding scrollbar to display frame snapshots on-hover. Matejka et al. [4] enhanced the video scrubbing technique for large videos by displaying a group of available frames in a form of grid.

In light of the scrollbar’s intrinsic features and prior work [1, 3, 4, 5], we propose ActiWare, an activity-aware scrollbar to visualize user interactions and dynamic social activity outside the user’s current view. The activity-aware scrollbar shows real-time indications of postings outside a user’s current view that might be of interest to the user.

As seen in Figure 2, our proposed widget displays visible red-popping boxes to indicate interesting social activity and interactions outside current view. This indication is a recommendation for the user to scroll to.

![Figure 2: Preview of the proposed activity-aware scrollbar – indicating activity outside current view.](image)
In the following section, we review related work. This is followed by details of our implementation and methodology. Finally, results are presented and discussed in the end of this paper.

2. RELATED WORK

Research in this area can be broadly divided into three approaches:

- To study or improve the control device e.g. isometric joystick [7] or scroll-wheel [8]
- To study or improve the soft-control e.g. Edit/Read Wear [5] or Swifter: video thumbnail scrolling [4]
- To study or improve the scrolling methods [2]

Our research focuses on the second approach of improving the soft-control, and augmenting the scrollbar soft-control with meaningful information.

Byrd 1999 [3] presented a color-coded visualization of query-matching tokens on the scrollbar. The user-study compared the developed color-augmented scrollbar with a native scrollbar to complete a task of text-searching as seen in Figure 3. Byrd’s results did not yield any statistical significance, likely due to the small number of participants used. His results and analysis played a role in our choice of the number of participants.

Content-aware scrolling (CAS) was presented by Ishak and Feiner [6] as a widget that varies direction, speed and zoom based on document content. In their CAS implementation, when a user scrolls, content is not treated as a single undifferentiated layer of information. Instead, scrolling could follow a different path for two-columned documents: to scroll back up as the user finishes reading the first column. Their design allowed for mode-based paths or manually added paths. In their paper, they presented a mural bar that displays the document’s text density in the scrollbar trough. The other technique is the pile bar that displays blocks of document texts as piles on a desk with labels, colors and sizes as identifiers. Both bars are exemplified in Figure 4.

In 2013, Matejka et al. [4] presented another variation of a scrollbar for videos – Swifter. Swifter displays a grid of thumbnail snapshots of the frames under the hovering cursor as seen in Figure 5. This is an extension of the well-known thumbnail display of frame on timeline to better suit larger videos. Larger videos have more frames per timeline interval, and so frames skip or flash too quickly on cursor movement. Their work presented a group of frames available at the certain time in a grid instead of a single frame. The authors conducted a total of three thorough studies in their paper, as well as addressed some of the problems regarding occlusion and content finding. Their paper was a resource and reference in our experimental design.

3. METHODOLOGY

In this section we will discuss the methodology followed in our user study. Details about the participants, scrollbar implementation, experiment design and procedure are portrayed.

3.1 Participants

A total of twelve unpaid participants were recruited to evaluate ActiWare, six male and six female. On average, our participants were 23.6 years of age. Six participants identified to use social platforms mainly for browsing user interactions, the other six identified using it for chatting. Most users navigated social platforms on a desktop – instead of a mobile version.
3.2 Implementation of the Scrollbar

Two versions of scrollbars were implemented: screening version and clickable version. The clickable version allows for the immediate view-teleportation on-click, whereas the screening version only shows the position; users have to scroll manually to it. The scrollbars were implemented in JavaScript as a Google Chrome extension. The source code of an existing Chrome extension [9] was modified to display the indications of dynamic user activities on the webpage, and to record performance measures. This was done by listening to the presence of a specific CSS class in the webpage’s DOM and dynamically binding the boxes relative to the view. Red-indications pop out of the scrollbar’s trough for both versions for the user to scroll to. The clickable version is shown in Figure 6.

![Figure 6: The implemented clickable scrollbar; (a) shows the red indication on presence of unseen activity on page; (b) shows how it interacts on hovering and clicking.](image)

3.3 Procedure

Participants were first asked to complete a demographic questionnaire to provide background information. Participants were then familiarized with the interface and the task to be completed. Finally, participants were asked to complete the task, during which they were assessed using two objective measures. A subjective questionnaire was presented at the end for qualitative analysis.

3.4 Task

The designed task involved the identification of a comment activity on a social website (Facebook). The participants were asked to: 1) Find the post with the most recent comment interaction – highlighted in blue; 2) Click on it. Figure 7 shows a visual explanation of the task. The following performance metrics were then measured automatically by our widget extension:

- Time (in seconds) taken to find the posting.
- Mileage of scrolling; the total distance (in page-height) scrolled to complete the task.

![Figure 7: Visual explanation of the designed task.](image)

We therefore, have two dependent variables. To facilitate such a task, we created a dummy profile with a fixed number of posts on its homepage. The user starts in a location that is equidistant from two potential activity locations – 5 pages from top location and 5 pages from bottom location. A comment activity was added on this homepage manually by the experimenter on one of the two potential locations – unknown to the user. Once the activity was added, the user was asked to click on the page, find the activity by scrolling or navigating, and click on the blue box - once found. This concludes the task. The first click on the page initiates our performance-recording routine.

3.5 Experiment Design

Participants navigated a social website (Facebook) on a desktop-based Google Chrome browser that is equipped with three variations of scrollbars. The experiment was a single-factor within-subject design. The independent variable in our experiment is the type of scrollbar used, thus we have one independent variable with three levels: activity-aware clickable scrollbar, activity-aware screening scrollbar and the native scrollbar. All participants were asked to complete the designed task using each of the scrollbars. Counterbalancing was carried out to eliminate any order effects.

4. RESULTS AND DISCUSSION

The activity-aware scrollbars significantly outperformed the native scrollbar. More specifically, users used 95% less scrolling mileage and 59.3% less time using the clickable scrollbar over the native scrollbar to complete the task. The screening scrollbar also outperformed the native scrollbar with 40.5% decrease in completion time and 45.4% decrease in scrolling mileage.

4.1 Completion Time

The grand mean for task completion time was 5.24 seconds for the three scrollbar types.
As seen in Figure 8, the clickable scrollbar was the fastest at 3.20 seconds, while the native scrollbar was slowest at 7.86 seconds. The main effect of scrollbar type on task completion time was statistically significant ($F_{2,9} = 14.9, p < .0005$), which was due solely to the difference between the native scrollbar and the clickable scrollbar, as determined by a Scheffé post hoc analysis. This means that the upgrade from native to screening scrollbar does not yield statistical difference in completion time. Also, the upgrade from a screening scrollbar to clickable scrollbar does not yield statistical difference in completion time.

The group effect on completion time was not statistically significant ($F_{2,9} = 1.84, p > .05$). This means that any learning effect that might have occurred from the ordering of one group was effectively offset by a similar and opposing learning effect for the other group; i.e. counterbalancing worked. The Scrollbar Type (F1) x Group interaction was however of statistical significance ($F_{4, 18} = 8.546, p < .001$). This means that there was an asymmetric transfer of skill.

### 4.2 Scroll Mileage

As for the scroll mileage, the grand mean across all scrollbars was 5.27 pages. As seen from Figure 9, the native scrollbar yielded the highest mileage at 9.92 pages, screening scrollbar yielded 5.42 pages, and finally the clickable scrollbar was best with almost no page scrolls at 0.50 pages. The main effect of scrollbar type on scroll mileage was statistically significant ($F_{2,9} = 35.8, p < .0001$), and was between all pairs of scrollbars as determined by the Scheffé post hoc analysis. This means that an upgrade from native to screening scrollbar has a significant effect on the scroll mileage in completing the task.

Additionally, an upgrade from screening to clickable scrollbar also had a significant effect on the mileage in completing the task. The group effect was not statistically significant ($F_{2,9} = 3.12, p > .05$). However, there was again, a statistically significant Scrollbar type (F1) x Group interaction effect ($F_{4, 18} = 3.47, p < .05$).

### 4.3 Ease of Use

Finally, users were asked to rate each technique on a scale between 1: most difficult to use, to 5: easiest to use, based on their overall assessment. As seen from Figure 10, the clickable version ranked highest at 4.42, followed by the screening version at 4.00. The native scrollbar ranked last at a 2.33 rating in ease of use.

### 5. CONCLUSION

We introduced ActiWare, an activity-aware scrollbar for displaying dynamic interactions on social platforms. It allowed for a new interaction technique for navigating dynamic pages in high activity scenarios. Two versions of scrollbars were implemented - clickable version and screening version - to visualize activity outside the user’s current view. The results of our study showed that clickable scrollbar was significantly better than the existing native scrollbar in locating new content, by a factor of 59.3% decrease in time and 95% decrease in scroll mileage to complete the task. The screening scrollbar also showed an apparent improvement over the native scrollbar with 40.5% decrease in time and 45.4% decrease in scroll mileage. Given the recent increased popularity of social platforms and dynamic content, we believe our work represents a timely and useful contribution.
6. REFERENCES


