

ActiveScrollbar: A Scroll Bar with Direct Scale Ratio Control

Hongzhi Song¹, Yu Qi¹, Lei Xiao¹, Tonglin Zhu¹, and Edwin P. Curran²

¹ HCI Group, College of Informatics, South China Agricultural University, Guangzhou, 510642, China

{hz.song, yuliangqi, lein_xiao, tlzhu}@scau.edu.cn

² School of Computing and Mathematics, Faculty of Engineering, University of Ulster at Jordanstown, Northern Ireland, BT37 0QB, UK
ep.curran@ulster.ac.uk

Abstract. Scroll bar is one of the most frequently used components of graphical user interfaces (GUIs). It is generally considered to provide overview + detail functionality. The scale ratio of a scroll bar refers to the rate of the document dimension to the proportion being displayed. It is usually determined by the length of the document and the height of the display window if considering the case of a vertical scrollbar. The user has no direct control to the scale ratio through the scroll bar, which is inconvenient when dealing with long documents in overview and navigation tasks. This inconvenience is more prominent for small display devices. This paper presents a novel GUI component, ActiveScrollbar, that enhances the standard scroll bar by providing direct scale ratio control without consuming more screen space. This component was expected to be more useful in hand held devices.

Keywords: Information Navigation, Information Overview, GUI Component, ScrollBar, Overview + Detail.

1 Introduction

Scroll bar is one of the most frequently used GUI components. It is generally considered to provide overview + detail functionality. The position of the scroll thumb inside the scroll trough denotes the current position of the detailed view within the overall document. The length of the scroll thumb also portrays the proportion of the document currently shown.

The minimum scale ratio for scroll bars is 1:1, and they are often removed from the display at this ratio because there is no purpose to display an extra widget with the entire document shown in the detail view. Their maximum proportional scale ratio is on the order of 100:1, but it can be set higher if we don't care so much of the proportion. For example, scroll bars can be used to navigate through documents that are several hundreds of pages long. At high scale ratios the length of the scroll thumb does not accurately reflect the proportion of the information space shown in the detail region. To do so truthfully could cause the thumb to shrink to less than one pixel, this will create obvious usability problems in acquiring the thumb. For this reason, a minimum thumb size of approximately ten pixels is used, which reduces the problems

of target acquisition but introduces problems of precise document control because small thumb movements causes large “jerky” document movements [1]. At high scale ratios, users must find alternative techniques to smoothly move their documents, such as the scroll bar’s end arrows, the cursor keys, or rate based scrolling etc.

Traditional scroll bars do not allow direct configuration of the scale ratio, scale ratio control is either hidden from view or performed through a separate widget. For example many software packages provide magnifying glass like icons to control the zooming function. This is not easy to execute, especially for novice users. In standard desktop interfaces, length of the scroll thumb immediately adapts to changes in zoom level within the detail region. There is no reason why the reverse control could not also be implemented, allowing the zoom level to be controlled directly by manipulating the length of the scroll thumb [1]. This paper presents ActiveScrollbar, A scroll bar with direct scale ratio control that enhances the functionality of the traditional scroll bar.

2 Related Work

Traditional scroll bars typically encode only spatial information. However several researchers had experimented with variants that additionally portray semantic information. Value bars [2] embellished the scroll trough with dashed lines, the thickness of which depended on numerical values extracted from columns in formatted data files, to depict the location of interesting or outlier data. Hilland Hollan [3] also used similar techniques to convey the read and edit history of document regions. Masui [4] used a scroll bar variant for browsing and filtering large lists of data. Byrd [5] evaluated a related technique with inconclusive results but positive subjective responses from participants. McCrickard [6] did an evaluation of the traditional scroll bar and its variants. Bederson et al. [7] used visual demarcations in the scroll bar trough to convey the presence of search matches across several months of appointment data in the PDA calendar Date-Lens. Range-slider [8] is similar to ActiveScrollbar, but it was targeted to control the number of records being displayed from a database rather than a generalized GUI component.

3 Objectives

The aim of this work was to enhance the functionality of the standard scroll bar in dealing with long documents or long components of GUIs. To pursue this aim, four objectives were set up as below:

1. A GUI component was to be designed which should be compatible with standard scroll bars.
2. The component should help the user more in gaining an overview of long documents or long components.
3. The user should have direct control to the scale ratio of the displayed document through this component.
4. The component should be space saving, and it was expected not to take more screen than the standard scroll bar.

4 Methods

Two more arrows were provided on both ends of the scroll thumb of a standard scrollbar, see Figure1. These two arrows control the scale ratio of the displayed document, but they also affect the current position of the detailed view within the overall document because adjusting them will change the length of the scroll thumb. The user performs the function by dragging these two arrows. Dragging the top arrow up increases the height of the scroll thumb, and dragging it down decreases the height. Dragging the bottom arrow up and down performs the reverse function.

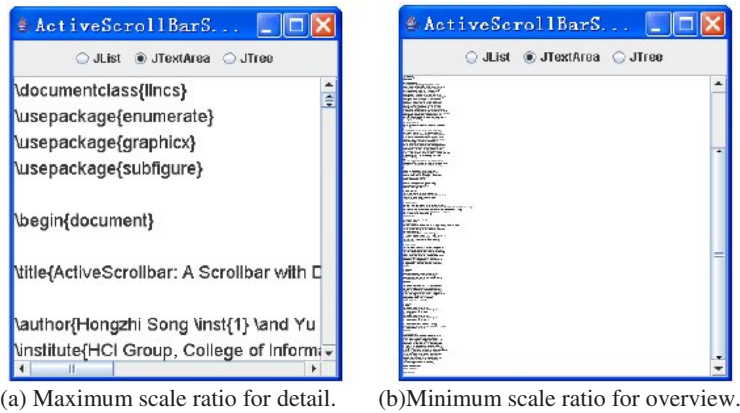


Fig. 1. ActiveScrollbar used to control the scale ratio of a long document

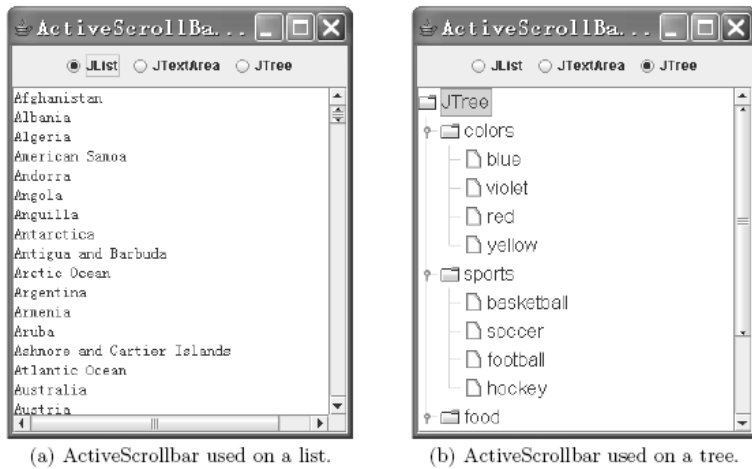


Fig. 2. ActiveScrollbar used to control the scale ratio of long GUI components

The functions of ActiveScrollbar were exemplified by relating the scale ratio to the font size of the displayed document. Increasing the font size means raising the scale ratio, and decreasing the font size means reducing the scale ratio. Three use cases for controlling the scale ratio were illustrated, one was a document in Figure 1, another one was a list component in Figure 2(a), and the third one was a tree component in Figure 2(b).

5 User Test

A couple of users were invited to try ActiveScrollbar, and they were all experienced computer users. The trial was subjective, and the goal was mainly to find user preference of the new widget comparing to its traditional peer. The users showed their interest in ActiveScrollbar. Some of them thought ActiveScrollbar was more direct in manipulation. Most users thought that it was easy to use.

A problem was also found. Although the function was thought to be good, but it is not intuitive enough, the users did not know what it is for until it was tried, or they were told to try dragging the small arrows beforehand. Perhaps a more intuitive icon than the two arrows needs to be designed.

6 Implementation

ActiveScrollbar was developed in Java. It subclasses the Java *JScrollBar* component in the Swing GUI toolkit, so it can be used as a replacement for *JScrollBar*. The example used to generate Figure 1 is the content of this paper prepared in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ format, Figure 2(a) is the list of countries, and Figure 2(b) is the default example of the Java *JTree* component. The widget responds to user interaction immediately, and the scale ratio changes smoothly when dragging the mouse on the two small arrows.

7 Discussion

This work produced a scrolling control component, ActiveScrollbar, that is fully compatible with the standard scroll bar, but having direct control to the scale ratio of the displayed document. It can be generally used as a substitute to the standard scroll bar. The scale ratio is programmable by the practitioner. For example the controlled variable can be mapped to the page size or the font size of the displayed document. If not mapped to any property the component will degrade to a standard scroll bar.

8 Conclusion

Traditional scroll bars do not allow direct manipulation of the scale ratio, but the length of the scroll thumb immediately adapts to changes in zoom level within the detail region. There is no reason why the reverse control could not also be implemented, allowing the zoom level to be controlled directly by manipulating the length of the scroll thumb. Although such a widget is expected, it has not been

provided yet. Some related work produced useful widgets for this purpose, but they were not designed for general use in everyday user interfaces. ActiveScrollbar implemented the user control of the scale ratio on scroll bars so that zooming control components may be removed from GUIs, and more screen space would be saved. It was expected to be more useful in dealing with long documents or long components such as GUI lists, trees or tables etc. It was also expected to be more useful in hand held devices.

Acknowledgements

This work was jointly sponsored by the National Science Foundation of Guangdong Province of China under grant number 06300433, and Talent Introducing Fund from South China Agricultural University under grant number 2005K099. The feed back and suggestions to improve Active Scrollbar from voluntary users are appreciated.

References

1. Cockburn, A., Karlson, A., Bederson, B.B.: A review offocus and context interfaces. HCIL Tech Report 2006-09, Department of Computer Science, University of Maryland, College Park, MD 20742, USA (2006)
2. Chimera, R.: Valuebars: An information visualization and navigation tool. In: Proc. CHI'92: Human Factorsin Computing Systems, Monterey, California, USA, pp. 293–294. ACM Press, New York (1992)
3. Hill, W.C., D., H.J., Wroblewski, D., McCandless, T.: Editwear and readwear. In: Proc.CHI'92: Human Factorsin Computing Systems, Monterey, California, USA, pp. 3–9. ACM Press, New York (1992)
4. Masui, T.: LensBar: Visualization for browsing and filtering large lists of data. In: Proc. IEEE Symposiumon Information Visualization '98, North Carolina, USA, pp. 113–120. IEEE Computer Society Press, Los Alamitos (1998)
5. Byrd, D.: A scrollbar-based visualization for document navigation. In: Proc. ACM Fourth International Conferenceon Digital Libraries, Berkeley, CA, USA, pp. 122–129. ACM Press, New York (1999)
6. McCrickard, D., Catrambone, R.: Beyond the scrollbar: An evolution and evaluation of alter nativen avigation techniques. In: Proc. IEEE Symposiumon Visual Languages '99, Tokyo, Japan, pp. 270–277. IEEE Computer Society Press, Los Alamitos (1999)
7. Bederson, B.B., Clamage, A., Czerwinski, M.P., Robertson, G.G.: DateLens: A fish eye calendar interface for PDAs. ACM Transactionson Computer-Human Interaction 11(1), 90–119 (2004)
8. Ahlberg, C., Shneiderman, B.: The Alphaslider: Acompact and rapid selector. In: Proc. ACM Conferenceon Human Factorsin Software, CHI'94, Boston, MA, USA, pp. 365–371. ACM Press, New York (1994)