Relative Scrollbars

Steven Robbins

September, 1995

Technical Report CS-95-10

Division of Computer Science
The University of Texas at San Antonio
San Antonio, TX 78249

Relative Scrollbars

Steven Robbins
Division of Computer Science
The University of Texas at San Antonio
210-691-5544 srobbins@runner.utsa.edu

ABSTRACT

This paper describes the concept of relative scrollbars. Simple scrollbars have been used for some time to distinguish a particular region in a linear data set. When two users of a cooperative editor are simultaneously working on the same file, each has a distinguished region corresponding to the part of the file which is displayed. Often what is most important is information on the position and amount of overlap of two regions. When the size of the distinguished regions is a small fraction (a few per cent) of the total region, ordinary scrollbars do not give an adequate representation of this information. We describe a new concept, that of relative scrollbars which expand the region of interest.

INTRODUCTION

When several agents are working on the same data, it can be desirable for each agent to know what part of the data each of the other agents is working on. This is critical when several agents are changing parts of the data and it is necessary that they do not modify the same section of data concurrently. An agent wishing to modify a piece of the data will lock that piece, do the modification, and then unlock that piece. Other agents wishing to modify (or read) the locked data will be blocked until the data is unlocked. When the agents are intelligent (in some sense) it may be more efficient if an agent that would otherwise block could perform some useful work. How should the information about which parts of the data are currently being used by other agents be represented? Here we explore a method of representing this information to a human agent in a visual way using *relative scrollbars*. Since the part of the data being modified is often a small portion of the entire file, a fisheye view [1] of the region of interest is used. Relative scrollbars have been implemented in a prototype cooperative editor and have been shown to be effective in representing this information.

At first we will model the linear data as representable on a continuous scale starting at 0 and having a fixed maximum value, and then consider the consequences of representing it on a discrete medium.

TERMINOLOGY

Consider a set of linear data that can be represented by numbers in the range from 0 to some maximum, T. A distinguished set of contiguous data from this set can be represented by an initial value, S, and a length, L. From these an ending value, E can be calculated. If the data is continuous (represented by floating point values), then $E = S + L$, while if it is discrete (represented by integers), $E = S + L - 1$. If there is more than one such set of distinguished data, one set is called primary and the others are secondary. Subscripts are used to distinguish these so that S_p is the start of the primary region and L_s is the length of a secondary region. We refer to the values above as the *world view* representation, as this represents the data as it actually is.

Often it is necessary to modify the view of the data so that it is more easily dealt with. For example, to display continuous data on a computer screen the data would have to be represented in terms of pixels. Panning and zooming could also be used to modify how the data looks. When a transformation is made on the world view we get a new set of numbers which we call the *window view*. We will use upper case letters for the world view parameters and lower case letters for the window view, so that in the window view s_p would represent the start of the primary region and l_s would be the length of a secondary region. A panning window view corresponds to a simple translation of the starting and ending values without changing the length, so panning by an amount x would give $s_p = S_p + x$ and $l_p = L_p$. The transformation used for secondary regions would be the same.

What we call the world view often is not a direct representation of the data, but itself a transformation. For example, a file consists of an array of bits, and the region of interest might be a substring of bits designated by a starting bit location and a count of the number of bits. If the file represents ASCII data, it is more convenient to represent this as a string of bytes and represent the start and length in terms of bytes rather than as bits. In a sense, this is a window view of that data. When stored on disk, the operating system treats a file as a collection of sectors (or blocks) while when represented in memory in a 32-bit computer the actual world view might be closer to a collection of 4-character values. The appropriate world view is determined more by how the data will be used rather than its physical representation. For example, Unix filters often treat data as collections of lines rather than bytes, at least for input and output. The world view of an array of lines is also appropriate for editing applications.

SIMPLE SCROLLBARS

Simple scrollbars have been used for a long time to represent a range of data. They can be used to convey information (an output device) and also to change information (an input device) by clicking or dragging parts of the scrollbar. Scrollbars can convey information efficiently without wasting much screen real estate. Recently, it has been suggested that sound can be used [2] to reduce errors associated with using scrollbars as an input device. We are interested in scrollbars as an output device. The motivating example for this work is a cooperative editor in which several users can simultaneously edit a file. A standard scrollbar can be used to show the displayed region of each user. For a given user, its scrollbar will be the primary one and the scrollbars of the other users will be secondary.

To simplify the discussion we assume that the data is continuous and that the world view is represented by values between zero and 1. A region of interest can be represented by a simple interval, $[a, b]$, where a is the start and b is the end. In the terminology above, a primary region in the world view would be represented by $[S_p, E_p]$.

The range of data manipulated by multiple users can be represented by several adjacent scrollbars. Figure 1 shows three scrollbars for users examining data in the intervals $[.1, .3]$, $[.05, .2]$, and $[.2, .8]$, respectively.

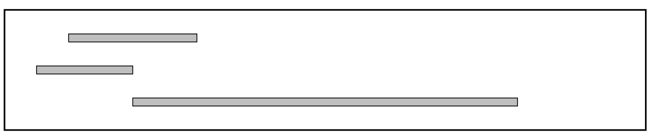

Figure 1: Three scrollbars representing the ranges $[.1, .3]$, $[.05, .2]$, and $[.2, .8]$.

RELATIVE SCROLLBARS

The three scrollbars in Figure 1 are treated symmetrically. If one of the scrollbars is distinguished, say because it is directly controlled by the user, the region represented by that scrollbar should be the primary region. Color can be used, but if the primary region is a small percentage of the total, this might be hard to see. For example, when editing a 1000-line file, the displayed region is often represented by a scrollbar. If only 20 lines are displayed, this represents two per cent of the entire region. When displayed on a screen using 500 pixels for the entire scrollbar, the region of interest is represented by 10 pixels. If two such regions overlap by one line, the region of overlap is represented by only half a pixel.

The primary region can be expanded to increase the resolution. The topmost scrollbar in Figure 1 covers 20 percent of the length of the figure. To expand the area of interest, we expand this to a fixed fraction of the displayed region, say 50 percent. This creates a window view in which the primary region covers half of the total and is placed in the appropriate relative position so that the ratio of the space on the left to the space on the right remains unchanged. The expansion is shown in Figure 2. The top scrollbar is the same as in Figure 1. This primary scrollbar divides the figure into three regions. Region A is to the left of the primary region. Region B is the primary region, and region C is to the right of the primary region. Each of these is linearly mapped into the corresponding region a, b, or c. Region b is of fixed length, 50 percent of the total in our example. Its position is determined by $A/a = C/c$. This mapping is determined solely by the primary region and is used for mapping both the primary and secondary scrollbars. The result of applying the same window view to all three scrollbars of Figure 1 is shown in Figure 3. The regions in which the secondary scrollbars overlap the primary region are shown in black.

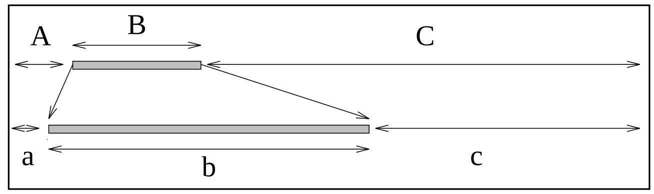

Figure 2: A primary scrollbar divides the figure into three regions. Each region is linearly mapped to a new region.

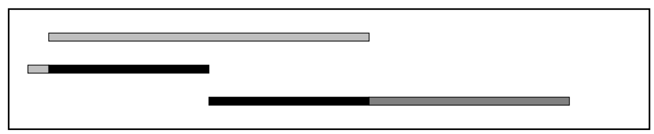

Figure 3: The three scrollbars of Figure 1 representing the ranges $[.1, .3]$, $[.05, .2]$, and $[.2, .8]$ are shown in the window view. The primary scrollbar is on the top and fills exactly half of the entire region. The regions of the secondary scrollbars which overlap the primary scrollbar are shown in black.

Figure 4: Three scrollbars representing the displayed lines 300 to 319, 282 to 301, and 320 to 339 of a 1000-line file.

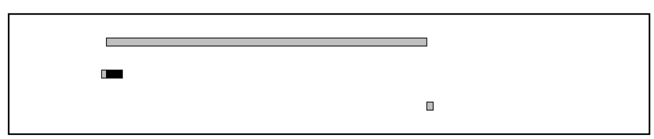

Figure 5: Three relative scrollbars representing the displayed lines 300 to 319, 282 to 301, and 320 to 339 of a 1000-line file. The top scrollbar is distinguished and expanded to be one half of the entire region.

Relative scrollbars can be considered an example of a generalized fisheye view as described by Furnas [1]. Furnas uses

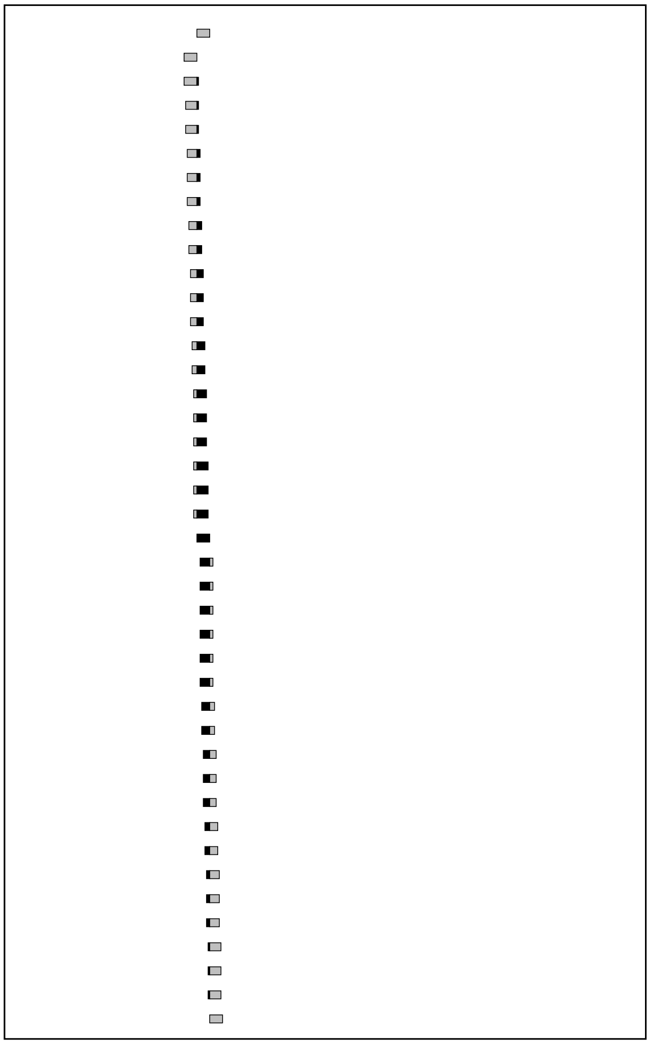

Figure 6: A primary scrollbar and several secondary scrollbars each representing a region displayed by one line from the previous one.

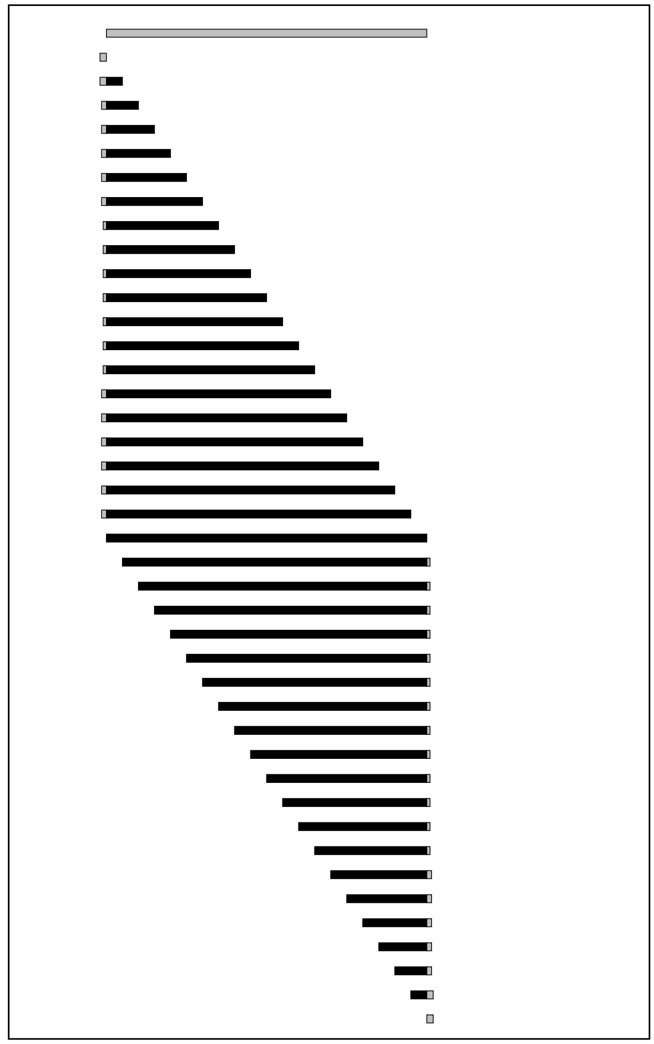

Figure 7: Relative scrollbars representing the same regions as in Figure 6. The region of overlap is more easily distinguished.

a degree of interest function and throws away those points whose degree of interest is smaller than a cutoff. With relative scrollbars nothing is discarded, but the points of interest are enlarged while the points which are not of interest are shrunk.

Returning to our motivating example, Figure 4 shows three ordinary scrollbars representing 20-line regions of a 1000-line file. The top scrollbar represents the the primary region of the 20 lines 300 through 319 and the others represent secondary regions consisting of the lines 282 to 301 and 320 to 339. It is difficult to distinguish the region of overlap even though it has been darkened. The expanded view using relative scrollbars is in Figure 5.

Perhaps the most important event to recognize is the starting of the overlap of the primary region with a secondary one. Consider again the case of the 1000-line file with a 20-line displayable region. The smallest overlap of two displayable regions is 1 line. If the total scrollbar length is 500 pixels, this overlap would be represented by about half a pixel. However, with relative scrollbars, the primary 20-line region will be 250 pixels long and a one-line overlap would be represented by over 12 pixels. This can easily be noticed even on a monochrome monitor and stands out even more when a distinctive color is used for the overlapping region.

Figure 6 shows a primary scrollbar and successive secondary scrollbars in which each secondary region is displaced by one line from the previous one. The primary region is fixed at lines 300 to 319 of a 1000-line file. The first secondary scrollbar represents the region from line 280 to 299 which does not overlap the primary region. The next secondary region overlaps the primary by just one line. It looks almost identical to the previous scrollbar which does not overlap. In the middle of the diagram the secondary region coincides with the primary one. The last secondary region shown no longer overlaps the primary region. Figure 7 shows the same regions using relative scrollbars. There is a clear distinction between the overlapping and non-overlapping cases.

RELATIVE SCROLLBARS TRANSFORMATION

This section describes the transformation from a world view to a relative scrollbar window view in the case that the world view is continuous and is represented by the interval $[0, 1]$. This is the case $T = 1$ with the primary region $[S_p, E_p]$ having length $L_p = E_p - S_p$. The primary region is mapped into an interval of size F. The window view transformation should be linear in the three regions determined by the endpoints of the primary scrollbar. There is only one piecewise linear mapping of $[0, 1]$ onto itself which is linear in these three regions and maps the primary region into an interval of size F such that the ratio of the space to the right and left of the interval remains unchanged. A point x satisfying $0 \leq x \leq 1$ is mapped into a point y given by

$$y = \begin{cases} \frac{x(1-F)}{1-L_p} & \text{if } x \leq S_p \\ \frac{xF - S_p F}{L_p} + \frac{S_p(1-F)}{1-L_p} & \text{if } S_p < x < E_p \\ \frac{x(1-F)+F-L_p}{1-L_p} & \text{if } x \geq E_p \end{cases}$$

DISCRETE SCROLLBARS

Since most output devices are pixel-based, actual scrollbars are discrete, not continuous. It is necessary to translate the ideas above into discrete calculations. We start by mentioning some of the properties that the translation should have. We assume that the world view is also discrete and consists of the integers in the interval $[0, T-1]$. The window view will also be represented by integers in an interval $[0, t-1]$ which can be thought of as the pixel values used in the display. The primary region will be expanded (or contracted) so that it takes up a total fraction of about F. To understand the drawbacks of directly using the formulas from the continuous case, consider again the example of a 1000-line file in which the displayable region is 20 lines. In this case $T = 1000$. We will use $F = .5$ and $t = 500$ corresponding to a scrollbar of total length 500 pixels.

There are several special values in the display that should be distinguished. These include

- The first point which represents the value 0 in the world and window views.
- The last point which represents the value $T - 1$ in the world view and $t - 1$ in the window view.
- The starting point of the primary region which is represented by S_p in the world view and s_p in the window view.
- The ending point of the primary region which is represented by E_p in the world view and e_p in the window view.

The transformation must map each of these world viewpoints into the corresponding window view point. Since these points are special, we make the further requirement that no other points in the world view are mapped into these points in the window view. So, for example, the only point in the world view that gets mapped into s_p is S_p.

This restriction might be difficult to satisfy if $t < T$. To understand this difficulty, suppose that the primary region consists of lines 4 through 23. Remember that lines are numbered starting at 0, not 1, so the first four lines are not in the primary region. There are 4 lines above the primary region and 976 lines below, giving a ratio of 1:244. Since only 250 pixels (half of the 500 total when $F = .5$) are available for displaying the region outside the primary region, only one pixel is used for the region above the primary region, and so $s_p = 1$. If this is the case, what does line 1 get mapped into? Only 0 can be mapped into 0 and only $S_p = 4$ can be mapped into $s_p = 1$. The conclusion is that S_p (line 4 in the world view) must be mapped into a pixel of at least 2 in the window view if a secondary region starts at line 1. The display of the primary scrollbar in the window view should be independent of the secondary scrollbars. That is, moving a secondary region should not affect the primary region.

Figure 8 shows a primary region and the six ways a secondary region can overlap it. In order to display a secondary scrollbar, its position relative to the primary scrollbar must be determined. Its endpoints then need to be mapped.

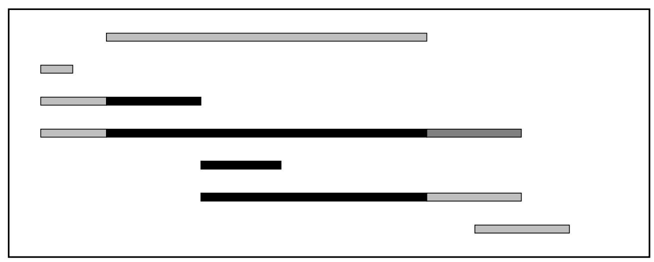

Figure 8: A primary region and the six ways that a secondary region can overlap with it.

While it might be difficult to write down a formula for the mapping as we did in the continuous case, an algorithm for computing the transformation is not too difficult. Given S_p, E_p, $L_p = E_P - S_p + 1$, F, T, and t, we need to calculate s_p and e_p so that the primary region contains (about) Ft points and the ratio of the number of points before the start and after the end of the primary region is (almost) the same in the world and window views. In each case we do not count the endpoints. With a little calculation we get:

- $s_p = \frac{(S_p-1)(t-Ft-1)+T-E_p-2}{T-L_p-2}$
  if $S_p \neq 0$ and $s_p < 2$, set $s_p = 2$.
- $e_p = s_p + Ft - 1$
  if $E_p = T - 1$, $e_p = t - 1$,
  else if $e_p > t - 3$ set $e_p = t - 3$.

In each case the floating point value obtained in the calculation can be converted to an integer in any standard way, such as by truncation or rounding. This completes the calculations needed to display the primary scrollbar. Now to compute the transformation of any other point,

- map 0 to 0.
- if $S_p > 1$, map the interval $[1, S_p - 1]$ to $[1, s_p - 1]$,
- map S_p to s_p,
- if $L_p > 2$, map the interval $[S_p + 1, S_e - 1]$ to $[s_p + 1, s_e - 1]$,
- map E_p to e_p,
- if $S_e < T - 2$, map the interval $[S_e + 1, T - 2]$ to $[s_e + 1, t - 2]$,
- map $T - 1$ to $t - 1$.

The interval mapping can be done in any reasonable way. For example to map $[A, B]$ into $[a, b]$, use

$$x \to \frac{b-a}{B-A}(x - A) + a$$

It is not necessary for the endpoints to be mapped into distinct points.

The above calculations fail if the primary region in the world view is so large that it covers all but possibly one point of the entire region. It is recommended that when the primary region is more than half of the entire region that ordinary scrollbars be used as relative scrollbars would produce a contraction of the region of interest in this case.

CONCLUSIONS

There are many interface issues that need to be solved before users will be willing to switch from standard editing tools to cooperative editors. Scrollbars are familiar to most users and the generalization of relative scrollbars is intuitive and easy to master. The feedback generated by relative scrollbars should add to the ease of transition to the next generation of cooperative tools.

REFERENCES

[1] Furnas, G. W. Generalized Fisheye Views. ACM CHI 1986 Proceedings, 16-23.

[2] Brewster, S. A, Write, P. C, and Edwards, A. D. N. The Design and Evaluation of an Auditory-Enhanced ScrollBar. ACM CHI 1994 Proceedings, 173-179.