Microsoft Scrolling Strip Prototype: Technical Description

Primary features implemented in prototype

Ken Hinckley 7/24/00

We have done at least some preliminary usability testing on all of the features described in this document.

Physical Manifestation of the Scrolling Strip

Several exemplar embodiments of a touch-sensitive strip are shown in Figure 1. In general, the scrolling strip has a *main scrolling area*, an *up scrolling region*, and a *down scrolling region*. In some embodiments, the up and down scrolling areas are physically separate from the main scrolling area.

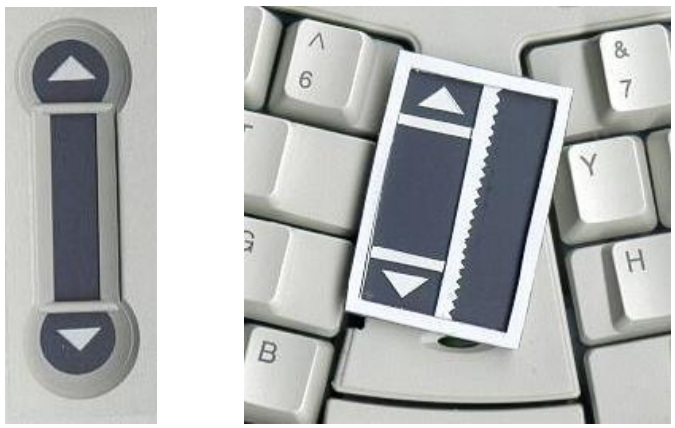


Figure 1: Some alternative physical embodiments of the touch-sensitive strip.
In the prototype pictured on the left, the separation between the arrows and the strip is a mild tactile break on the surface of the pad itself. This seemed to work best in user tests. In the same picture, the region to the right (with the "sawtooth" edge giving a gentle tactile cue) provides the *absolute scrolling function* described below.

Basic scrolling functionality

As done in the prior art, touching and sliding one's finger on the touch-sensitive strip scrolls the document relative to its current position. This is known as the "Touch-and-slide" gesture. The amount of scrolling is multiplied by a user-selectable gain factor. A number of improvements to the signal handling, as well as techniques to handle certain difficulties, are also possible for an optimal embodiment of a touch-sensitive scrolling strip.

- *Speed-Sensitive Acceleration:* We apply an acceleration component to the amount of scrolling, which is dependent on how fast the user moves his or her finger. The amount of scrolling, dy, is calculated as an exponential transformation of $(T_i - T_{i-1})$, the vertical distance between the current and the previous touch strip samples:

  $$dy = K_1(T_i - T_{i-1})(e^{1 + K_2(T_i - T_{i-1})} - e + 1)$$

  This transformation allows precise, slow scrolling at low speeds, but rapid movement across longer distances when moving quickly. We do not use a threshold for any particular speed, but rather apply a continuous variable gain depending on how quickly the user is moving. When using acceleration on the touch-sensitive scrolling strip, one drawback is that movements are not exactly reversible: if the user moves a certain distance along the strip, and then returns to his or her previous position, the document may not scroll back to the same spot. However, this is usually a minor problem relative to the performance gains that are realized.

- *Rounding to whole lines:* For some or all applications, the amount of scrolling can be rounded to a whole integer number of lines or pages. To implement this concept correctly, two different scrolling positions need to be computed. The first, which we call the *actual scroll position*, is the amount that the visible document on the screen has been scrolled; the actual scroll position always reflects a whole integer number of lines or pages of scrolling. The second, which we call the *virtual scroll position*, keeps track of a higher-resolution scroll position within an imaginary document which does *not* need to be rounded to whole integer numbers of lines or pages. The user's gesture is received as an input, and this is converted into a virtual scrolling position with as much precision as possible. After the current virtual scrolling position has been computed, only then is the *actual scroll position* computed, by rounding the virtual scrolling position to the desired increment. This concept is important, as it allows subtle changes of the user's input to add up over time (accumulating in the virtual scroll position), which eventually affect the (more coarse) actual scrolling position in terms of whole lines or pages of scrolling visible on the screen. Finally, checks need to be made to ensure that the virtual scroll position does not become less than the beginning scroll position of the document, nor greater than the end scroll position of the document. Typically, the rounding of lines is applied to all methods of scrolling (relative motion, jumping, auto-scrolling, etc.).

- *Moving / Not Moving Detection:* When the user's finger is in contact with the surface, our software infers whether or not the user is intentionally moving his or her finger. This decision is used as a building block for several other features and recognized gestures, such as *Scroll Freeze*, and the onset of Auto-Scrolling modes in some situations (which we will describe subsequently). When the user first touches the scrolling strip, the user's finger is considered to be *Not Moving* on the strip. When the user's motion exceeds a certain threshold, the finger is then considered to be *Moving*. The finger continues to be considered *Moving* until a timeout , currently set at 500 ms, expires during which total movement of the finger is less than a small percentage of the length of the strip. If this condition is met, the finger once gain is considered to be *Not Moving*. Subsequent motion can revert to the *Moving* state, as before.

- *Scroll Freeze:* When scrolling, we ignore very small finger motions in some circumstances. This prevents the document from "swimming" slightly due to small variations in the sensor readings. User contact with the strip does not cause any scrolling until the *Moving* state, as described above, is detected. From then on, all motions of the user's finger cause changes to the scrolling amount, until the *Not Moving* state is detected, at which point scrolling is locked to its current position, until the *Moving* state resumes.

- *Lift-off problems:* Sometimes when the user removes his or her finger from the touch-sensitive scrolling strip, the user's finger may unintentionally perturb the current scroll position. This can be prevented by maintaining a FIFO (first-in, first-out) queue of recent finger positions. When the user breaks contact with the pad, we search back through this queue to look for a sample where the scrolling motion came to a definite stop. If such a sample is found, the scroll position is adjusted to occur at that stop point. If no such stop point is found, or if the amount (distance) of the correction is too large, then the scroll position is *not* adjusted; this probably means the user was moving his or her finger very quickly when they let go of the strip, so in this case a refinement of the scroll position might not be helpful. A related queuing technique has been proposed for cursor positioning using touch-sensitive tablets (Buxton, Hill et al. 1985).

Long-Distance Navigation

Our touch-sensitive strip may include separate up/down regions as shown above in Fig. 1 (Center), or the strip may have an absolute position sensing capability which allows it to know the position of the user's finger on the pad (rather than just motion of the finger on the pad). The absolute position of the finger on the pad may be used to jump to a corresponding absolute position in the current document.

:

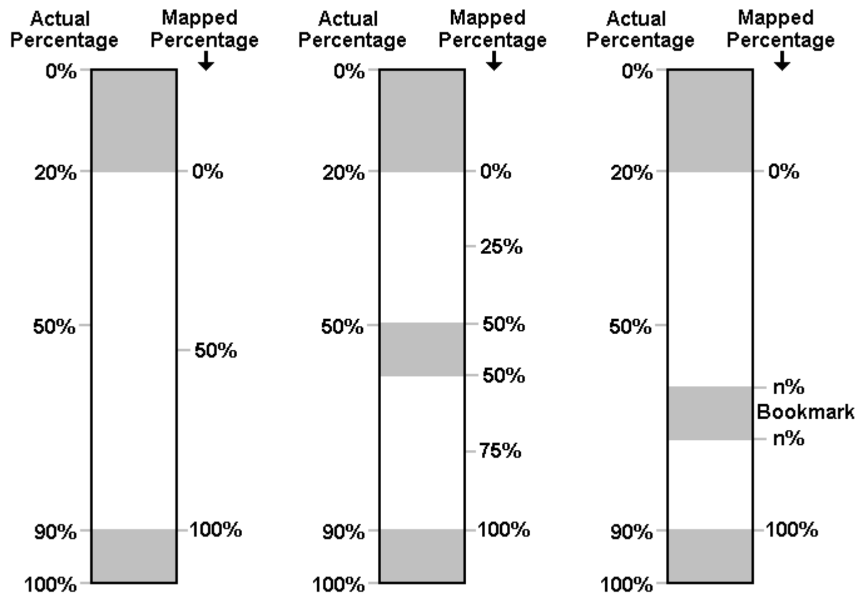


Figure 2: The input mapping of the touch strip may be distorted to make it easier for the user to jump to certain preferred regions. *Left*: Distortion of Home and End regions with unequal size (Home region larger). This distortion should be applied for both *jumping* and *absolute scrolling mode*.

- *Absolute Scrolling Mode*: The user can hold down a control key, or touch a second part of the strip separate from the main area, to trigger an absolute scrolling mode. The absolute scrolling mode establishes a one-to-one relationship between the main area of the touch sensitive strip (or some portion thereof, per the *Distortion of Input Regions* section above) and the position of the scrolling handle within the traditional graphical scroll bar. While in the absolute scrolling mode, the document behaves much like it would if the user were to click and drag the handle of the scrollbar up or down: it allows absolute movement across the entire length of the document. The user can exit the absolute scrolling mode by breaking finger contact with the surface. One particularly effective way to provide an absolute scrolling mode is to divide the input surface is divided vertically into two halves; one half of the surface is used only for relative scrolling, while the other half of the surface is used for absolute jumping (Buxton and Myers 1986).

Another possibility is to have a gesture on the main area of the strip that performs a single discrete jump. Early user testing suggests that this is more of an advanced feature that should be turned off by default. Detection of double-tapping in our current software is unreliable, but even if it were reliable, a significant usability problem is inability to jump to a particular location with precision, as well as the possibility of triggering a jump by mistake.

- *Double tap to Jump*: When the jumping feature is enabled, we require two consecutive taps at the same location on the surface to initiate a jump. Sometimes, double tap is used to jump, but single tap is still used to initiate some other scrolling feature which is less harmful if activated accidentally, such as placing the text insertion point.

- *Animated Transitions*: The jump to a distant target takes place as an animated transition. The animation occurs as a smooth "slow-in, slow-out" animation of the scroll position between the start position and the target position of the document, lasting approximately 0.33 seconds. This helps the user to visually track the change to the document view, without becoming disoriented by the sudden change in position.

- *Distortion of Input Regions*: The actual sensed positions on the pad are mapped to corresponding positions in the document with a distortion such that jumping to Home (at 0%, the beginning of the document) or jumping to End (at 100%) are distinguished positions. This is accomplished by artificially enlarging the 0%

and 100% regions to take up more space, while the remaining percentages are scaled to fit in the remaining real estate of the input surface. The exact percentages of the strip depend on its physical dimensions and the exact geometry of the strip's bezel.

Auto-scrolling

Our touch-sensitive strip can provide auto-scrolling behaviors if desired. Three different types of auto-scrolling are supported:

- *Auto-Scroll Regions*: If the user initiates contact with the scrolling strip within the up-scrolling or down-scrolling regions (see Figure 1 for illustrations of some embodiments of these regions), then the document begins auto-scrolling up or down, following a short time-out (possibly as short as 0 ms). When the auto-scroll regions are part of the main area of the touch-sensitive strip, auto-scrolling only occurs if the user holds still (is not moving). This prevents auto-scrolling from engaging if the user mistakenly starts a stroke on the strip in the auto-scrolling regions.

- *Hold-and-Scroll*: If the users touches and holds their finger anywhere on the strip without moving, then after a short time-out, auto-scrolling is enabled, and subsequent motions of the finger cause the document to scroll (or optionally pan in two dimensions, if the strip can sense horizontal motion as well) at a rate proportional to the distance between the initial contact point and the user's current position. While this feature can be useful, some users will activate it by mistake.

Depending on user preference, some, all, or none of the above auto-scrolling behaviors can be enabled in the software.

Auto-scrolling rate calculation

Many different functions for mapping the rate of scrolling to the user's input are possible. The simplest is to use a fixed rate of scrolling (which may be user-selectable). Many touchpads currently on the market can sense the *contact area* of a finger on the pad. In some prior art, the contact area is used to estimate degree of pressure by using fixed thresholds for the amount of contact area, or simply by using the raw contact area measurement itself. This has problems because contact area is only an indication of pressure, not a direct measurement of it. Furthermore, the actual amount of contact area varies depending on the size of the user's finger(s), the angle at which the user holds his or her finger, and the amount of pressure that the user naturally exerts.

We calculate the scrolling rate based on finger contact area using an algorithm which (1) normalizes for the amount of finger contact on the current scrolling action, and (2) performs an exponential transformation of the finger contact area to provide a controllable range of scrolling speeds, from fast to slow.

The scrolling rate is calculated in two steps. First we calculate:

$$P = K_1 \left((p / p_0) - 1\right)$$

Where P is the normalized pressure estimate based on contact area, K_1 is a gain factor, p is the current pressure reading, and p_0 is the pressure chosen for the minimum-rate point (described below).

Next, we use this to calculate the rate of scrolling:

$$dy = K_2 \left(e^{(P+1)} - e + 1\right)$$

where P is calculated as defined above, K_2 is a gain factor, and dy is the resulting calculated rate of scrolling. If dy is less than zero, then it is ignored (set to 0). When the scrolling rate is applied to the document, a positive value of dy is used for scrolling down, or dy is multiplied by -1 for scrolling up.

Different values of the gain factors are used for the *auto-scrolling up* and the *auto-scrolling down* regions. This is because depending on the exact shape of the strip, the shape of the bezel, and whether or not auto-scrolling occurs in physically separate Up and Down, inherently more or less of the user's finger may contact the pad in the *auto-scrolling up* region than in the *auto-scrolling down* region.

The choice of an appropriate value for the p_0 minimum-rate point is critical. Several methods are possible, but we find that the best ones use the dynamics of the *current* user gesture itself to determine a value of p_0 in real time as the user is attempting to specify a rate of scrolling.

- Pressure for the minimum rate point is sampled following a 200ms time-out after initial contact. This time-out is long enough for the user to establish firm initial contact with the pad. During this 200ms period, the current value of the pressure is continuously used as a preliminary estimate for p_0, so that the user may start scrolling without any perceptible delay.

- Alternatively, a maximum threshold on the rate of scrolling is imposed. If the computation of dy as shown above results in a rate larger than the maximum threshold, then the value for p_0 is recomputed using a sliding window such that the current pressure value p would result in the maximum rate of scrolling. Also, optionally, if the minimum pressure exerted falls below the bottom of the sliding window, the value of p_0 may be recomputed by sliding the window downward. Similarly, the known range of pressure values which can be sensed by the input device may be used as a basis for choosing initial minimum and maximum points of the sliding window. The *Rate limits* and/or *Pressure maximum* criteria described below may be used in such a sliding-window strategy.

The following criteria may be added to these algorithms to improve the selection of the p0 minimum-rate point:

- *Rate limits*: A maximum or minimum rate of movement can be imposed on the above equations, such that if the user exceeds one of these limits, the p0 minimum-rate point is recalculated to satisfy the limit. This has the effect of adapting the sensed pressures to the user's gesture and typical reading speeds. *(equations)*
- *Pressure maximum*: Because most touch-sensitive pads or strips can sense only contact area and not a true measure of pressure in Newtons, there is often a practical limit on how much "pressure" can be applied. Knowledge of typical such values can be used to help choose the p0 minimum-rate point. For example, if the user starts by pressing hard, then the document will immediately scroll quickly since it is known that the user will not be able to further increase the contact area. *(equations)*

The pressure maximum or the parameters in the above equations may differ for the Up and Down arrows since the typical contact area of the user's finger with different physical regions may vary. Such adjustments to the parameters can make the pressure response seem more consistent in the different areas.

These features allow the user to continuously adjust the scrolling rate as he or she scans through the document. The continuous adjustment of the scrolling rate afforded by these algorithms is more controllable and predictable than methods that use thresholds of the contact area.

Audio Feedback for Scrolling Interactions

Audio feedback may be used to enhance usability of our scrolling techniques. Much of the audio feedback is fully interactive in that it responds based on real-time qualities of the user's physical gesture, including speed, position, and timing. Audio feedback enhances the quality of user interaction in several ways:

- It provides distinct feedback for user actions, confirming that the computer has understood and responded to the user's gesture. If the user makes a mistake, it will be immediately clear since the audio feedback will not sound as expected.

- It helps to enhance eyes-free use of the touch-sensitive strip, since the user can hear the response without looking at the strip itself, or even the graphical display of the document or scroll bar.

- The sound is pleasant and makes the interface seem more lively.

Some users like audio feedback, but testing so far indicates that it is not an essential feature. However, we have not done side-by-side tests to determine if it helps users to learn and distinguish the various features of the strip. Sounds associated with these events seem the most useful:

Interaction Event	Description
Jump	When user activates the jump-to-target feature, the user hears what sounds like and object quickly whooshing by and stopping quickly at a new spot.

EndOfDoc / Right Edge	A distinct thud occurs when the user reaches the end of the document (by any means).
BeginOfDoc / Left Edge	A distinct snap occurs when the user reaches the beginning of the document (by any means).
Touch Up Arrow	Any initial contact with the Up Arrow.
Touch Down Arrow	Any initial contact with the Down Arrow.
Release Down Arrow	Any loss of contact with the Down Arrow. May sound distinct from normal releases.

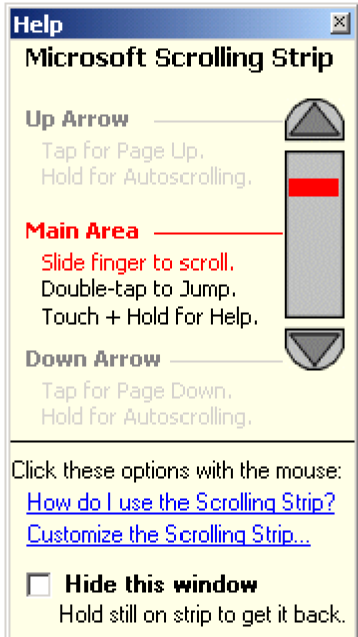
Up and Down Arrows
Physically distinct arrows, or "virtual" arrows at the extreme ends of the touch strip, may be used to trigger multiple functions that may not be available on the main area of the scrolling strip. A single tap in these regions executes a Page Up or Page Down command, while pressing and holding initiates pressure-sensitive auto-scrolling (as described below).

- *Tap arrows to Page Up / Page Down:* A single tap, or series of taps in quick succession, issues one Page Up or Page Down command for each tap on the up arrow or down arrow, respectively.

- *Tap and hold to auto-scroll:* The view will auto-scroll (at a fixed rate or with a pressure sensitive response to the amount of contact, as detailed above) while contact is maintained with the up or down arrow. The auto-scrolling starts after a short time-out of approximately 500-700ms so that if the user just taps or double-taps on the arrow for paging or other behavior, a small amount of unintended auto-scrolling will not be incurred. When the up & down regions are a virtual extension of the main scrolling strip, and not physically distinct regions, they may also support auto-scrolling by sliding the finger into the regions and then holding still.

Help and Configuration Options Panel
When the user first installs and starts using the scrolling strip, any initial contact with the strip will bring up a help screen which displays Help and Configuration Options. The screen is similar to a tool-tip except that it is activated by touching the scrolling strip. When the Help and Configuration Options Panel appears, it may be positioned along an edge of the current window with keyboard focus (if there is room for it), or along an edge of the screen.

The essential "help" information is displayed in this tooltip. For more detailed help or advanced options, the user can then access options in this panel by clicking on the desired item using the mouse. The window does *not* grab keyboard focus when it appears. When the user stops touching the scrolling strip, the panel remains visible for 10 seconds before automatically disappearing. The user can also explicitly dismiss or move it (like a regular window), or via the "Hide this window" option shown in the figure.



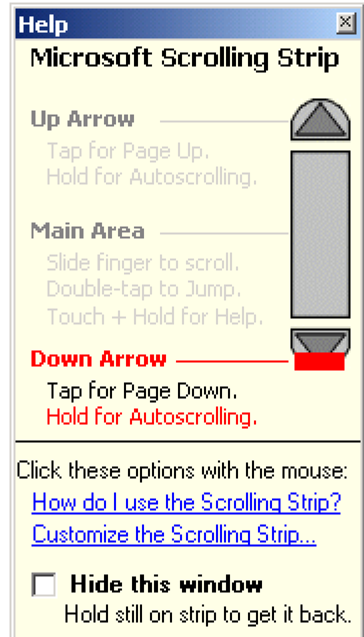


Figure 3: Example implementation of a Help & Configuration Options panel. The help is a full-fledged window that can be moved or closed. As the user touches different areas of the scrolling strip, the available functions are shown, and the currently activated function (if any) is highlighted. These examples show the user touching the main area of the scrolling strip (center) and the down arrow (far right). The red rectangle provides feedback of exactly where the user is touching.

Other Features of the Scrolling Strip

Placing the Insertion Point: In word processing and data entry applications, the user may scroll to a new location to begin text entry. When scrolling an application using the scroll bar, typically the insertion point ("IP"), where new text will appear, does not move. When scrolling using the touch-sensitive scrolling strip, the user may wish to place the insertion point without having to grab the mouse and click in the window. Several policies for placing the insertion point with scrolling are possible, such as keeping the IP always visible on the current page, having it follow the scrolling line-by-line, or having a specific gesture (such as a single tap on the main area of the strip) that places it on the current page, if it is not already visible there. **We currently do not have a solution for this problem implemented, and if Synaptics has a way to do this, that would be of interest.**

Modifier Keys: In some embodiments, the use of a keyboard modifier key may be used in combination with any of the gestures recognized on the touch-sensitive scrolling strip to support additional functionality. For example, holding down the Control key (Ctrl) while sliding one's finger along the strip can control the current zoom factor of the document (which may be in discrete zoom levels, or continuous if supported by the application). Another example is holding down the ALT or SHIFT key to trigger the absolute scrolling mode.
Horizontal Scrolling: Horizontal scrolling is also desirable. Left and Right arrows (with short movement in blocks on a single tap, and pressure-sensitive auto-scrolling when held) are one solution. A strip that is wide enough to allow horizontal movement is another.