Popup Vernier: a Tool for Sub-pixel-pitch Dragging with Smooth Mode Transition

Yuji Ayatsuka
University of Tokyo
7–3–1 Hongo
Bunkyo-ku
Tokyo, 113, JAPAN
+81-3-3812-2111
`aya@is.s.u-tokyo.ac.jp`

Jun Rekimoto
Sony Computer Science Laboratory Inc.
3–14–13 Higashi-Gotanda
Shinagawa-ku
Tokyo, 141, JAPAN
+81-3-5447-4380
`rekimoto@csl.sony.co.jp`

Satoshi Matsuoka
Tokyo Institute of Technology
2–12–1 Ookayama
Meguro-ku
Tokyo, 152, JAPAN
+81-3-5734-3876
`matsu@is.titech.ac.jp`

ABSTRACT

Dragging is one of the most useful and popular techniques in direct manipulation graphical user interfaces. However, dragging has inherent restrictions caused by pixel resolution of a display. Although in some situations the restriction could be negligible, certain kinds of applications, e.g., real world applications where the range of adjustable parameters vastly exceed the screen resolution, require sub-pixel-pitch dragging. We propose a sub-pixel-pitch dragging tool, *popup vernier*, plus a methodology to transfer smoothly into 'vernier mode' during dragging. A popup vernier consists of locally zoomed grids and vernier scales displayed around them. Verniers provide intuitive manipulation and feedback of fine grain dragging, in that pixel-pitch movements of the grids represent sub-pixel-pitch movements of a dragged object, and the vernier scales show the object's position at a sub-pixel accuracy. The effectiveness of our technique is verified with a proposed evaluation measure that captures the smoothness of transition from standard mode to vernier mode, based on the Fitts' law.

KEYWORDS: Fine Grain Dragging, Vernier, Smoothness, Fitts' law, Multiple Modes

INTRODUCTION

Dragging an object on a computer display directly with some pointing device is one of the most useful and popular graphical user interface techniques. Direct manipulation interface is intuitive in that a user can drag objects for various tasks—for example, the knob of a slider to select an item, windows to rearrange his/her desktop, and file icons to drag-and-drop onto an application icon, etc. A unit of dragging is normally restricted to be at least the pixel of the display, and as such, objects on the display cannot be moved at a half pixel resolution, and also the cursor cannot represent 'sub-pixel-pitch' movement feedback.

Although such a restriction is negligible for many familiar situations, it sometimes causes a problem. For example, remotely-controlled real-world machines often have control parameters whose range is vastly greater than what can be depicted with the pixel resolution of a typical graphic display. A non-linear video editing system typically has a slider-based time scale to move amongst the frames, but it is often quite difficult to manipulate to a specified frame as the number of video frames are enormous. Selecting items with a slider knob from lists consisting of thousands or even millions of items is an another problematic example.

One way to overcome the problem is to change the scale for detailed dragging. Many drawing editors have zooming facilities for precise editing, though coarse and precise editing mode are not 'continuously' connected each other because control of zooming factor interrupts editing. Elastic interface such as the FineSlider[5] can manipulate fine grain movements without zooming with a rubber-band metaphor, albeit indirectly as the user controls the velocity of the control and not its position, thus being less intuitive.

An Alphaslider[6, 1] consists of two or three sub-sliders, each one representing different granularity of movement within the depicted range of the whole slider. A user can select and control the knob of the desired sub-slider depending on the desired granularity of control. The Alphaslider technique is difficult to extend two-dimensional controls, and furthermore, requires more effort on the user's part.

We propose a new fine-grain dragging and feedback technique, called *Popup Verniers*. A popup vernier, invoked by a user with an auxiliary button or a key, consists of a zoomed grid surrounding an object subject to dragging, and verniers around the grid. The object can be dragged with sub-pixel-pitch resolution in an intuitive way. The movement is represented by the counter-movement of

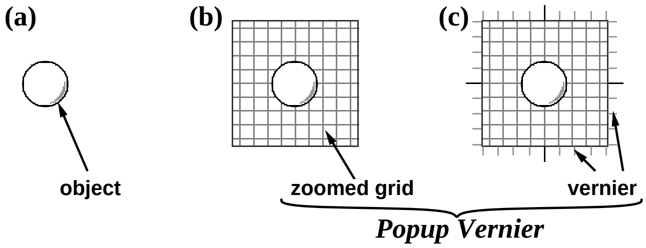

Figure 1: Popup Vernier

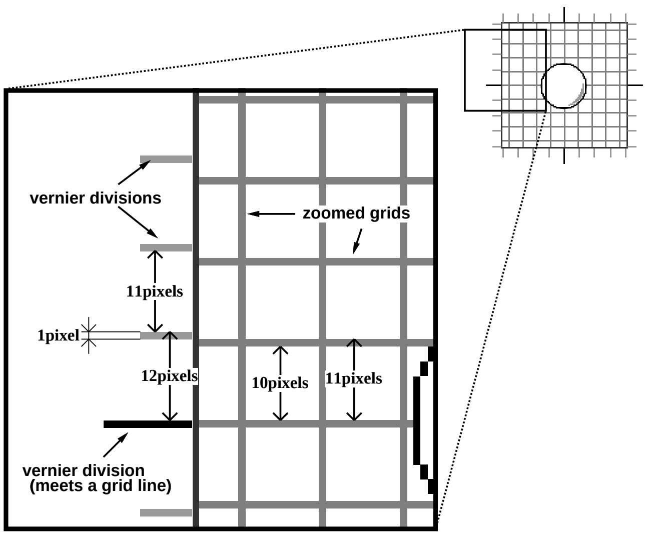

Figure 2: Details of Popup Vernier

the surrounding grid. The precise distance of movement can be recognized by the scales on the vernier, as is with vernier calipers. A user can drag an object coarsely at first, and continuously and smoothly transfer into fine grain dragging mode. We also propose the measure of the smoothness of the transition as a meaningful usability metric based on the Fitts' law[3] over multiple dragging modes. We then perform user studies in "selecting items from lists" situations, comparing a slider with popup verniers to FineSlider and Alphaslider. The results indicate the effectiveness of popup verniers and also the validity of our new metric.

POPUP VERNIERS

In this section, we discuss the issues in fine grain dragging, the possible techniques, how to evaluate their efficiency, and then design a new feedback mechanism for fine-grain movement.

Method for Fine Grain Dragging

Moving or pointing task is generally performed coarsely at first and gradually transfers to finer adjustments, as indicated by the Fitts' law. Thus, a fine grain dragging technique must facilitate a methodology to allow smooth transition from coarse to fine granularity for better usability. Transition from fine to coarse granularity, on

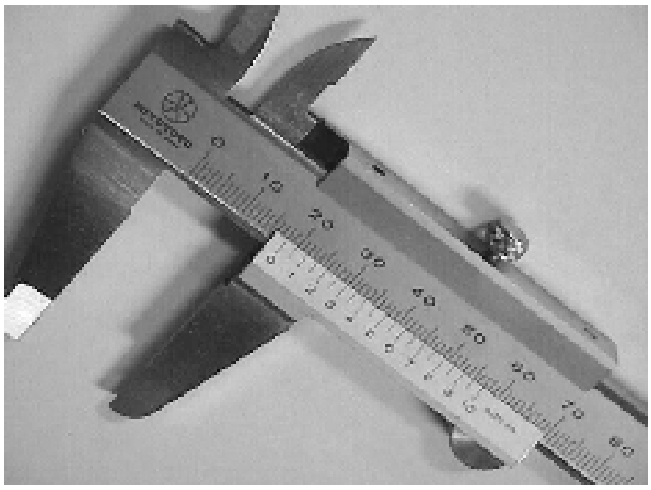

Figure 3: Vernier Calipers

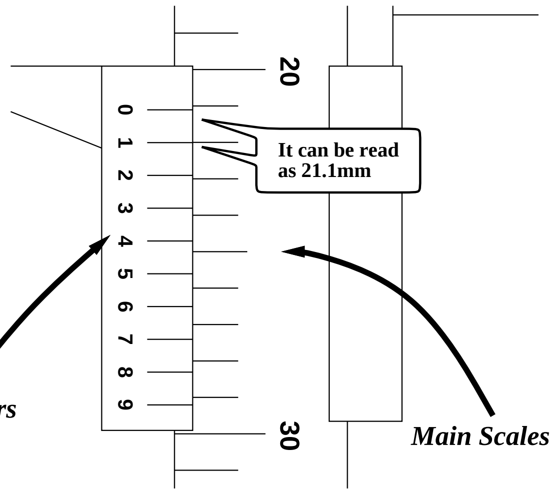

Figure 4: Verniers and Main Scales

the other hand, is of less importance.

In the most graphical user interfaces, a user selects an object by pressing the mouse button, and releases the button to release the object after dragging. For smooth transition from coarse to fine, this sequence should not be interrupted, i.e., the button should not be released until the end of the dragging sequence. We therefore employ an auxiliary button on the mouse or a key as a trigger to invoke the fine grain dragging mode, freeing the user from interruption. The user can go back to coarse-grain by releasing the buttons and restarting to drag the object.

There are alternative ways to make the mode transition without the auxiliary button; for example, triggering on the values of mouse speed or acceleration. However, due to possible frequent mode transitions, it would be difficult for the user to grasp the non-linear relationship

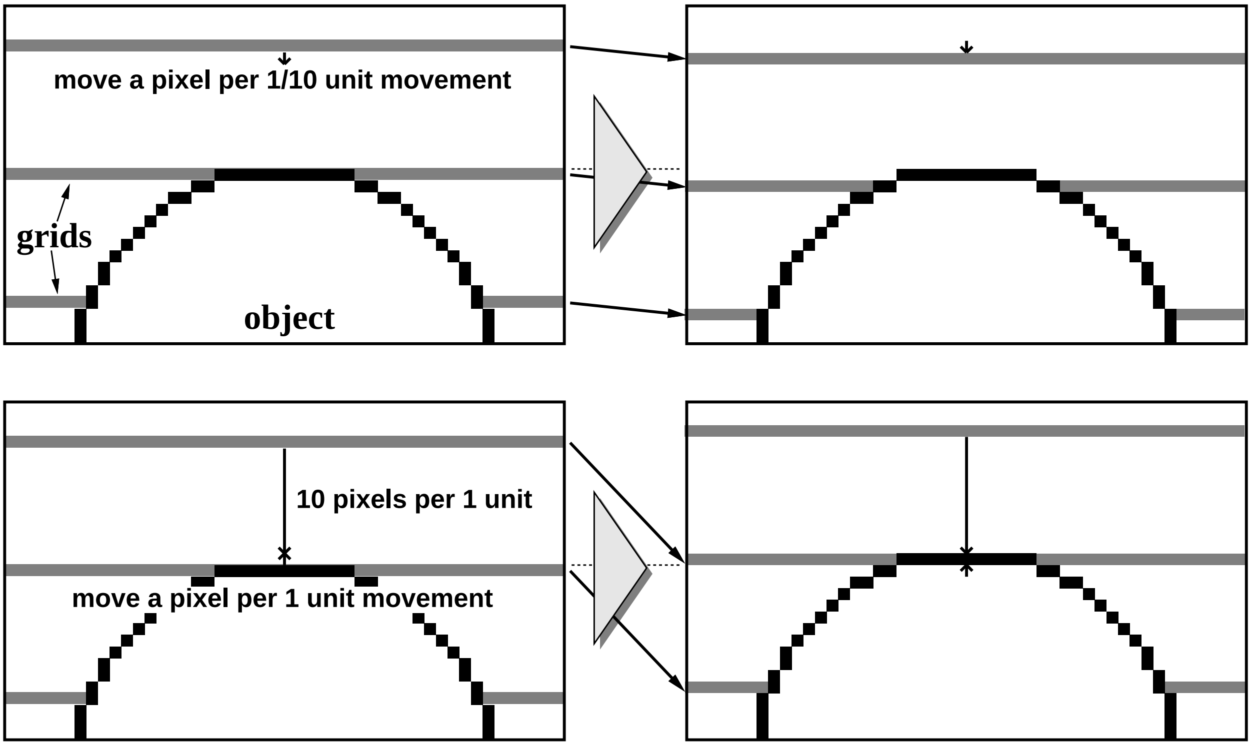

Figure 5: Movements of Grid and an Object

between the speed/acceleration of the mouse and speed of an dragged object. Binding different directions of movement of the mouse to granularity modes, as is with the micrometer interface of Alphaslider, is another possible method; this, however, is only applicable for one-dimensional dragging as it is difficult to extend the technique to two or more dimensions. Furthermore, adding more buttons and other physical controls to a pointing device for alternative interactions is becoming popular, as is with the Microsoft IntelliMouse.

Under our fine grain mode, a unit movement is performed by multiple pixel movement of the mouse cursor, rather than by a single pixel movement as is with the Alphaslider. Such a design has shown to be effective for relieving the user from the pressure of fine adjustments, thus reducing the cost of not only the fine sub-pixel-pitch dragging but also of the coarse, pixel-pitch dragging. We shall verify this with the user studies in a later section.

Evaluating the Smoothness

By all means the most important metric of a dragging method is the rapidness, i.e., the time to complete the task of moving an object precisely onto a target place. However, users might not always feel comfortable with the fastest method, as it may be more stressful. In particular, as interfaces involving multiple granularity will incur transitions between manipulations of different grains, lack of smoothness in the transitions may incur more stress. Thus, we also provide a metric that indicates the smoothness of the transition, based on the Fitts' law.

Time to transfer from one grain to the next —first order continuity:
The most obvious metric is the time to transfer from one grain to the next. The shorter the time interval between the last movement in the coarse grain mode and the first movement in the fine grain mode, smoother the transition is. This metric indicates the first-order continuity of the function which indicates the amount time of user manipulation versus object movement.

Suitability to Fitts' law —higher-order continuity:
As another metric of smoothness, we propose a new metric corresponding to higher order continuity, which can be formalized as suitability of the technique to Fitts' law over multiple modes. Under Fitts' law, dragging in a single granularity mode mostly plots a log-order-curve on the time-to-distance graph, with some errors. In multiple granularity modes, the time-to-distance graph for each mode should also be a log-order-curve, and smooth, higher-order connection of all the curves should ideally form a single, smooth, log-ordered-curve. Under such circumstances, the user should feel more comfort compared to the case when the curve is less continuous. We verify this hypothesis by the user study.

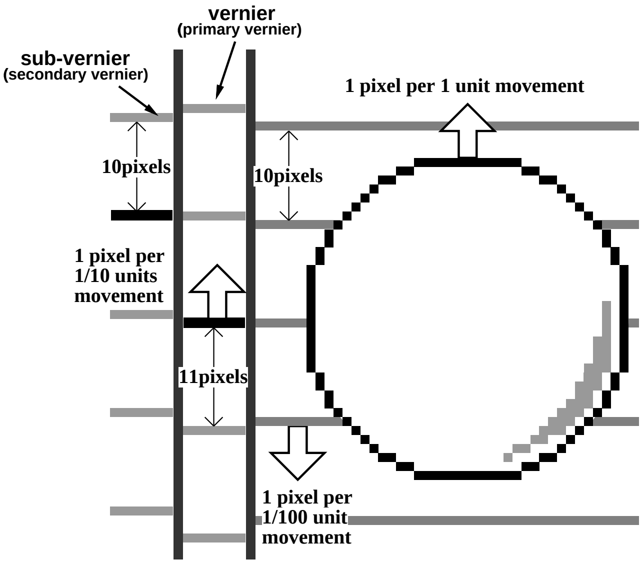

Figure 6: Multiple Level Verniers

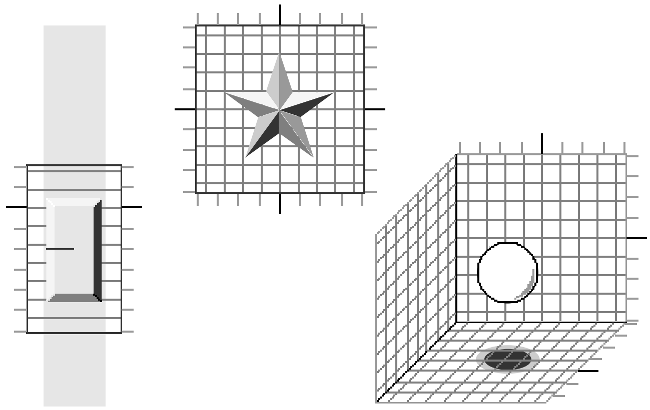

Figure 7: Other Examples of Popup Verniers

Feedback for Fine Grain Movement

The use of anti-aliasing technique improves the visual feedback resolution to one fifth of a pixel[2]. However, for our purpose, we require much finer resolution; in fact, our requirement is not to simulate the showing of sub-pixel positions, but rather, to represent sub-pixel movements during dragging. From this perspective, the seemingly obvious approach of numerically indicating the exact coordinate of the object subject to dragging is not very intuitive for the users, as there is no apparent motion of the dragged object itself, but rather, the numbers change at some different position on the display, with little indication as to the velocity of the movement, etc. A more intuitive feedback is necessary in this regard.

Instead, we propose to pop-up a zoomed grid for feedback purposes (Figure 1 (b)). The grid is superimposed onto the background scene, and placed behind the target object. Each grid line serves as the pixel coordinate indicator of the object, and the interval between the grid lines represent the sub-pixel resolution. For example, when the sub-pixel resolution is 1/10th of a pixel, the interval between grid lines is 11 pixels (Figure 2). Suppose that one drags an object, whose uppermost pixel is juxtaposed with a grid line, upward under fine-resolution dragging mode. The object itself does not initially move, but instead, the entire grid moves one pixel downward, indicating the physical movement of the object 1/10th of a pixel upward. When the grid moves 10 pixels downward, then the object moves one pixel upward, again being juxtaposed with the next grid line above the original one at its top. Such continues counter-directional movement of the grid coupled with the object moving itself when 'overflow' occurs in the grid movement effectively presents the user with an illusion of continuous sub-pixel-pitch movement.

Here, the issue is how to indicate exactly where the object is located, without using unintuitive numerical coordinates. In a real-world setting, physical gauges for fine grain measures such as calipers or micrometers utilize *verniers* to obtain the readout of fractional measurement values smaller than the resolution of the main scales (Figure 3, 4), but (rather surprisingly) have not been well-exploited in human-computer interaction. For our purpose, we place the vernier lines surrounding the grid, whose interval is 1 pixel larger than the grid line interval, allowing easy readout of the current sub-pixel grid position (Figure 2). We call the combined grid and the surrounding vernier as shown in Figure 1 (c), *Popup Vernier*, as an interaction technique that allows unobtrusive and continuous interactions for fine grain object dragging. Another merit of popup vernier is that it is applicable to any object, irrespective of their shape or position within the display. As a result, popup verniers are easily applicable in 1D, 2D, 3D situations (Figure 7), in contrast to other techniques.

In order to obtain even finer resolution dragging, one must increase the interval between the grid lines. Thus, for example, if 1/100th pixel resolution is required, the grid line interval must be 101 pixels—too sparse to be usable in practice. Instead, we can extend the vernier technique to allow even finer-grained object manipulation. We situate a secondary (sub-)vernier alongside the first (primary) vernier, resulting in multiple-level verniers (Figure 6). In the figure, the sub-vernier represents 1/10th pixel resolution, while the primary vernier 1/100th pixel resolution. The grid lines moves a pixel for each 1/100th unit movement of an object, counter-directional to dragging. The primary vernier, in turn, moves a pixel for each 1/10th unit movement, towards the same direction as dragging.

USER STUDIES

In order to study the advantages of our approach, we performed comparative studies in a one-dimensional dragging, or more specifically, selection tasks using sliders.

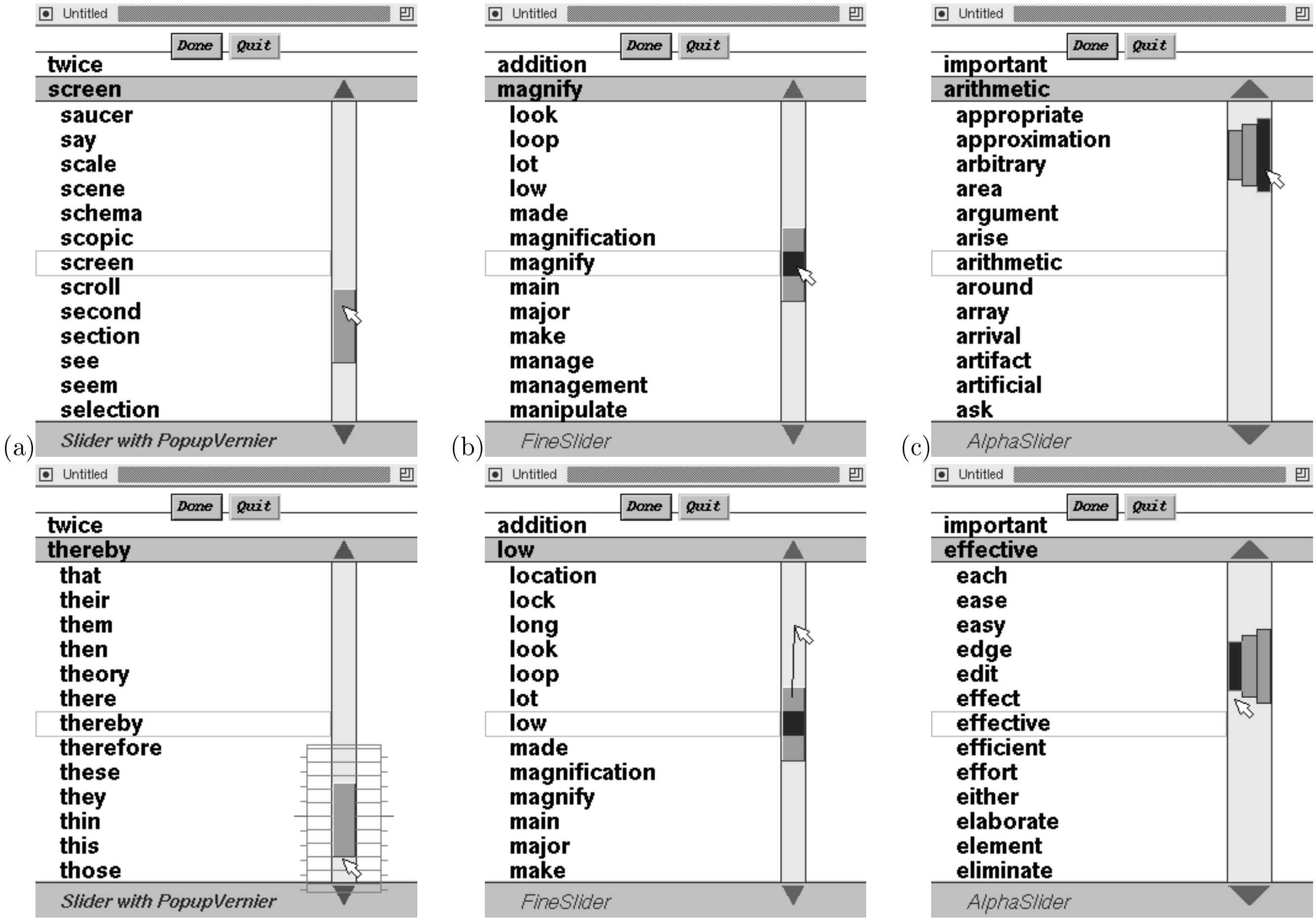

Figure 8: Compared Sliders: (a) Slider with Popup Vernier (PVSlider), (b) FineSlider, (c) Alphaslider.
(The upper row shows the coarse-grain mode and the lower row shows fine-grain mode)

Experiment 1: Comparison to Other Techniques

We compared words selection task using a slider constructed out of popup vernier (PVSlider), FineSlider and Alphaslider. Each subject was given a list of 1000 English words and a target word. Only several words are shown in the list; the subject must navigate through the list using the given slider, to select the target word by placing it into the center of the display area (indicated by a colored frame). Then, the subject presses the space key when the target word is selected, upon which a bell rings, and the next target is shown. Each task consists of 10 such word selections repeated consecutively.

At the start of the experiment, we introduced three sliders to the subject, without informing him/her of which is the original method and which were the ones for comparison (although some subjects knew). We rotated through the sliders in random order. For each given slider, one practice task was performed with random word targets, and subsequently, we measured the time for the real task, whose targets had been prepared beforehand and the same for all subjects. When the experiment was complete, we also interviewed the subject as to which one they preferred or felt easy to use, etc.

For FineSlider (Figure 8(b)), we employ $f(x) = \kappa \cdot x + 1$ as the *adjustment unit function*, which had been proposed and resulted in the best result in [4]. Here, κ is a constant, and x is an intermediate variable computed from the distance d, which in turn represents the interval between the end of the elastic line at the mouse cursor location and the control knob of the slider. x is 0 when d is less than a constant b, otherwise x is $d - b$. The constant b is 45(pixel) in our implementation, as was in [4]. Alphaslider in our experiment (Figure 8(c)) has both the positional interface and arrow buttons for fine-grain movement, as was introduced in [1] as the 'redesigned' Alphaslider. The resolution of the finest sub-slider has been set so that the user can select the next item by a single pixel movement of the mouse cursor. For our PVSlider, (Figure 8(a)), we defined the fine grain mode so that the user can select the next item by a six-pixel movement[1]. The popup vernier, i.e., the fine grain mode, can be invoked by the right button of

[1] An applet version of a slider with popup vernier is available on `http://www.is.s.u-tokyo.ac.jp/~aya/PVSlider.html`.

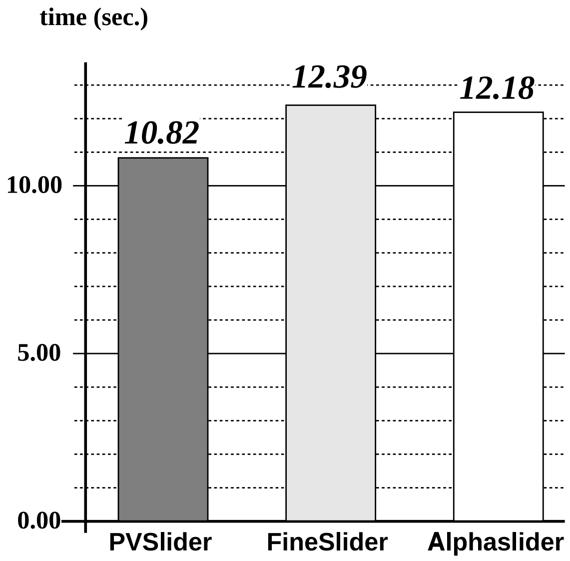

Figure 9: Average Time to Select a Target

the mouse or 'v' key. We also provided arrow buttons for FineSlider and PVSlider, as is with the Alphaslider. Subjects were told to use the arrow buttons freely when the sliders themselves seems obtrusive for fine grain adjustment.

Experiment 1: Results

32 subjects participated in the experiment. The subjects consisted of different expertise in GUI, anywhere from novice users to expert users. Figure 9 shows the average time to select a target word for each slider. As we can see, PVSlider is about 1.4 seconds or 12% faster than either FineSlider or Alphaslider ($p < 0.01$). The difference between FineSlider and Alphaslider, on the other hand, is not significant.

Figure 10 shows the transition time into fine-grain mode, which is defined as the time interval from the last movement in the coarse grain mode to the first movement in the fine grain mode. PVSlider is about 1.0 seconds or 42% faster than other sliders ($p < 0.001$). No subject used the 'v' key to pop-up the grid/vernier, instead of the right mouse button. The remaining distance from the user's current selection to the target to be selected at the time when a user makes a transition into the fine mode (in case of Alphaslider, into the finest mode) shows little difference for respective sliders (Table 1). It is supposed that Alphaslider has the middle grain mode so that the time is shorter.

Table 2 presents how many times subjects transfered into finer grain mode (including the use of the arrow buttons), and back into coarse grained mode from fine-grained mode. The result shows that the frequency of transition into fine-grain mode is over 9 times that of transition back into coarse-grain more for all the sliders, indicating the importance of concentrating on the coarse-to-fine transition in GUI design.

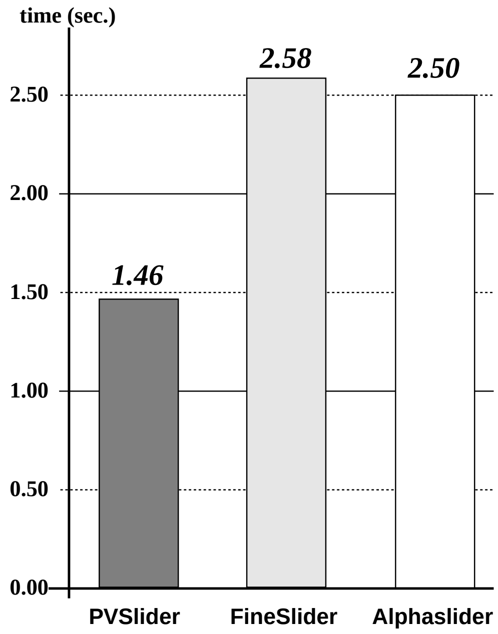

Figure 10: Time to Transfer into Fine Grain Mode

type	PVSlider	FineSlider	Alphaslider
remaining dist.	18.6	18.2	16.5

Table 1: Distance from Target upon Transition into Fine Grain Mode

type	PVSlider	FineSlider	Alphaslider
to fine	347	395	389
to coarse	37	37	21

Table 2: Number of Transition between Modes

Table 3 shows how many targets the users employed the arrow buttons during the course of their navigation, an indirect indication of whether the particular slider is easy to use. Obviously, lower the number, the more effective the technique is. With our PVSlider, the arrow buttons were used almost one third as frequent as Alphaslider, and one fourth as frequent as FineSlider.

Table 4 and Figure 11 show the results of the interviews after the experiments. The subjects were asked which slider they preferred or felt easy to use, and the reason for their selection. About 78% of the subjects answered that PVSlider is better than the other sliders. Some who chose PVSlider answered that they can make the transition into the fine-grained mode quickly; some said that they liked the fact that they do not have to release the mouse button during dragging. On the other hand,

type	PVSlider	FineSlider	Alphaslider
use arrows	37/320	167/320	113/320
percentage	11.6%	52.2%	35.3%

Table 3: Number of Targets for which Subjects Used Arrows

one subject indicated that transition into the coarse-grained mode was difficult, and another one said that pressing two buttons was troublesome.

Figure 12 shows the distance vs. the time graph: the horizontal axis (in log-scale) indicates the remaining distance to the given target, and the vertical axis shows the average time interval from time the slider being at the distance to the target (and not exceeding the distance), up to the time of the selection. For example, for the PVSlider, the average time interval from when the target is less than 100 items away until when the target is selected is 7.00 seconds. For distances farther than 40, the graph of all the sliders are nearly parallel, likely because coarse-grained mode of all the sliders are almost identical. When the sliders transfer to fine-grained mode, the graph of PVSlider remains almost linear, while the other sliders exhibit 'higher-order' discontinuity around the distance value of 40 (note that the average distance when mode transition occurs is approximately 18).

Now, let us compute the regression line in the form of $y = \alpha x + \beta,\ err = \epsilon$, where x is natural logarithm of the distance and y is the time in seconds. Smaller coefficient α indicates that tasks are completed faster, and smaller average error and smaller absolute value of y-intersect β indicate that the graph is more favorable under Fitts' law. The computed regression lines for each slider are as follows:

PVSlider: $y = 1.49x + 0.02,\ err = 0.57$

FineSlider: $y = 1.78x + 0.36,\ err = 3.09$

Alphaslider: $y = 1.67x + 0.51,\ err = 3.11$

Here, let us refer to these regression lines as *approach lines*. Furthermore, if one restricts the range of the graph to distance values of less than 30, then the graphs of FineSlider and Alphaslider are linear for the range. There, the regression lines are $y = 2.14x - 0.15$, $err = 0.16$ for FineSlider, and $y = 2.08x - 0.06$, $err = 0.04$ for Alphaslider, respectively. The graphs indicate that PVSlider is advantageous over other sliders according to Fitts' law, over multiple resolution modes.

Experiment 2: Verification of the Metric

The results of the experiment 1 strongly supports the effectiveness of popup verniers, in that the selection slider with popup vernier was shown to be superior in many aspects according to the experiment. Still, it is unclear whether advantage of the technique according to

type	PVSlider	FineSlider	Alphaslider
prefer	25	3	4
ratio	78%	9%	13%

Table 4: Subject Preference

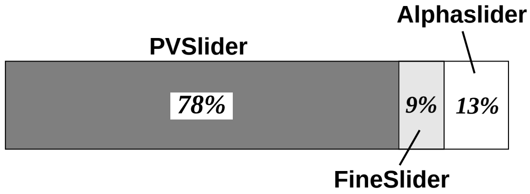

Figure 11: Subject Preference Ratio

Fitts' law contributes to true subject preference of the technique, especially due to transition between multiple resolution modes was not considered in the original Fitts' law. In particular, our claim that, preservation of 'higher-order' continuity so that the combined graph will remain linear for smooth and continuous transition, must be validated as actually effective for the user.

To certify our claims, we performed another experiment. We provided PVSliders of various speed for the fine-grain mode. A subject can make a unit movement of the grid by n pixels movement of the mouse cursor, where n is 2, 4, 6, 8, 10, or 12. Obviously, the grid moves faster when n is smaller and vice versa. Each task was same as the experiment 1: a subject first practices a task at mid-speed ($n = 6$, the same as used in the experiment 1), and then performed six tasks from slower to faster, or faster to slower. All the target selections were made randomly. After the all tasks were completed, we interviewed each subject as to which speed they preferred with respect to ease of use (we allowed multiple answers).

Experiment 2: Results

21 subjects participated this experiment. 17 subjects (referred to as group A) answered $n = 4$, 6 and/or 8 to be better than other speeds, and 6 of them (group A$'$) answered $n = 4, 6$ to be the best. Approach lines of group A are:

n=2: $y = 1.35x + 0.116,\ err = 0.208$
n=4: $y = 1.29x - 0.017,\ err = 0.308$
n=6: $y = 1.27x - 0.104,\ err = 0.300$
n=8: $y = 1.43x - 0.149,\ err = 0.318$
n=10: $y = 1.47x - 0.080,\ err = 0.319$
n=12: $y = 1.63x - 0.205,\ err = 1.005$

Note that the line for $n = 2$ has smaller coefficient than that of the line for $n = 8$. This means that the subjects in group A completed the tasks faster when $n = 2$ compared to $n = 8$, while they preferred $n = 8$ over $n = 2$ (moreover, most subjects indicated that $n = 2$ is difficult as being too fast). It is surprising that the average

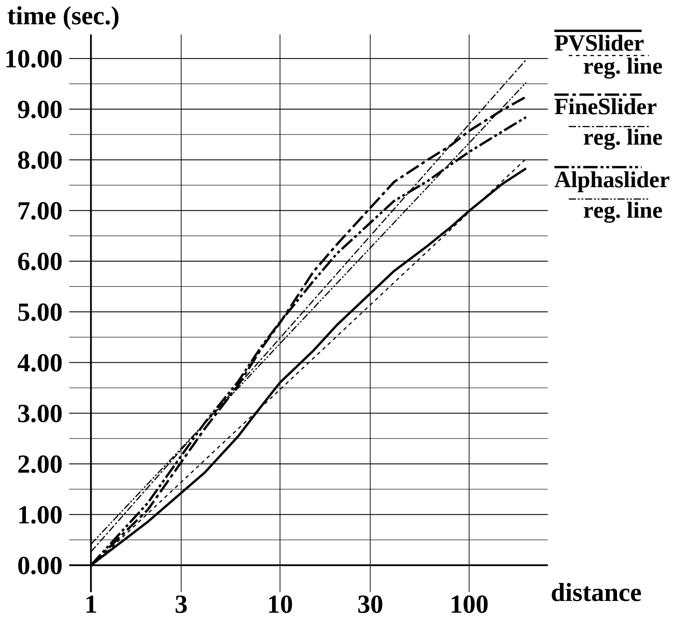

Figure 12: Remaining Distance to Time Graph

error is 0.208 and the y-intersect is 0.116 for $n = 2$, while they are 0.318 and -0.149 for $n = 8$, respectively; the average errors are not significantly different, but the y-intersects are positive for $n = 2$ and *negative* for $n = 8$. Lines for $n = 8$ and $n = 10$ are similar in average errors and y-intersects, and few answered that $n = 10$ is difficult. The line for $n = 12$ has much larger average error than other lines, and some subjects indicated that it is too slow to use.

Approach lines of group A′ are:

n=2: $y = 1.24x + 0.066,\ err = 0.238$
n=4: $y = 1.15x - 0.055,\ err = 0.200$
n=6: $y = 1.30x - 0.169,\ err = 0.364$
n=8: $y = 1.48x - 0.238,\ err = 0.421$
n=10: $y = 1.40x - 0.109,\ err = 0.343$
n=12: $y = 1.59x - 0.226,\ err = 0.552$

The line for $n = 2$ has smaller α than that of the line for $n = 6$, while y-intersect of $n = 2$ remains positive. The y-intersect of $n = 8$ is negative, but its absolute value is much greater than $n = 4$ or $n = 6$. The absolute value of y-intersect of $n = 8$ in group A′ is much larger than the absolute value of y-intersect of $n = 8$ in group A. It can be said that the absolute value of y-intersects distinguishes group A′ from A. We can also observe the approach lines of group A′ and A, in that smaller α and y-intersect within the range from -0.2 to 0.0 seems to make users feel comfortable. Most average errors remain within similar ranges; alternatively, when the average error is larger, the y-intersect is also out of the 'comfort' range (e.g., the lines for $n = 12$ for both group A and A′), and thus their effect on the user comfort is still somewhat unclear.

There are 2 subjects (group B) who answered that the best one is $n = 12$. Approach lines of group B are:

n=2: $y = 1.65x - 0.245,\ err = 0.446$
n=4: $y = 1.62x + 0.269,\ err = 1.837$
n=6: $y = 1.74x + 0.059,\ err = 0.833$
n=8: $y = 1.67x - 0.266,\ err = 1.732$
n=10: $y = 1.63x - 0.234,\ err = 1.387$
n=12: $y = 1.52x - 0.108,\ err = 0.422$

The approach line for $n = 12$ does not have significantly smaller α compared to other lines, but its y-intersect is in the 'comfort' range mentioned above and also has smaller average error value. Although the sample is small, we can say that, despite the difference in speed, user seem to prefer the cases where we judged as being advantageous.

The other 2 subjects do not belong to neither group A nor B. One answered that $n = 2$ is the best and the other said $n = 8$ and $n = 10$ are better than the others. More samples are needed to analyze such cases.

DISCUSSIONS

These results reveal that user preference manifests in not only the speed (coefficient) of the task, but also in other parameters such as y-intersect and average error values. In particular, except for most preferences by the subjects, y-intersect seemed to be a better metric over speed. Specific range of y-intersect (in this case, -0.2 to 0.0) reflects user's preference rather accurately. The average error value do not quite reflect user preference as y-intersects or coefficients, but larger errors certainly do seem to indicate that the parameters are not preferable.

As a whole, the results seems to support the validity of our new metric, in that Fitts' law extended to multiple granularity modes seems to reflect the user's preference and comfort. More precisely, the best parameter is slightly different from what would have been under single-mode dragging. Further experiments are required to explore whether the observed difference is general or not.

One of the interesting aspect is that, since with popup vernier mouse movements no longer directly reflect object movement on the display, mouse movements are no longer the single suitable metric under our extension. In fact, despite that the speed of the mouse increases considerably at the time of mode transition, no one of the subjects expressed uncomfort despite the sudden change in correlation of mouse speed versus the speed of the object being dragged. Further experiments and analyses of this fact may reveal new features of the human perception and motion control mechanism.

The metric may be applicable to automatic adjustment of parameters. Observation of the parameters of approach lines according to the changes of system parameters could provide the best set of the system parameters for each user. If a user's preference changes, it may be detected by the system and the system parameters can be gradually re-tuned.

Tuning by our extended Fitts' law can bring other advantages. In a tuned system, user's manipulation is more predictable than in an untuned system, because their actions can be approximated with a simple line. For example, a sequence of mouse movements could contain sufficient information to recognize a target position that the user desires. Such predictions enable automatic snapping to the predicted target, or automatic mode transitions as the user approaches the target.

The metric could also be applicable not only to GUIs or human-computer interfaces, but also for wide range human-machine interfaces. For example, it could be applied to Jog edit dials typically seen in high-end video players. The dial are used to select an appropriate frame from a lengthy video sequence, allowing them to control the motion of the tape. The dial does not exhibit linear properties, but rather, accommodates various changes in speed vs. dial angles. Some cell phones and PDAs have such dials to quickly select the phone number of the desired contact person. Our metric could be used to evaluate such interfaces.

APPLICATION AREAS

We have already implemented and employed popup verniers in a remote controlled camera system employed in a chat-augmented conference system reported at CHI'98[7]. The remote controller is a laptop PC which has a VGA-size LCD display. The width of the display is 640 pixels, while the camera we employed (Sony EVI-D30) has a pan resolution of about 3500. We created a 2-D camera direction control knob, which could be dragged at 0.1 pixel resolution with a popup vernier. This allowed smoothly controlled panning of the camera remotely from a different room during the entire conference which lasted for three days. Fine-grained control provided with popup verniers proved essential for situations where extensive zooming was required. For example, during Q&A period, the camera could be quickly panned to the person asking the question, then his face be zoomed, and the camera direction finely adjusted using the popup vernier.

We are planning to apply popup verniers to other applications. As is with the PVSlider, slider bars of many applications are also suitable for popup verniers. For example, in a non-linear video editing system, x-axis normally represents the time scale. If a pixel represents one frame, the display width is about 1200 pixels, and a frame rate is 30 frames per second, the display can only embody 40 seconds of video. Applying popup verniers will make the display contain minutes of video and will allow seamless transition from coarse specification to frame-by-frame specification, allowing smooth video editing.

Popup verniers are useful for higher resolution displays. Even if the display resolution becomes several hundred pixels per inch, at that point users will not be able to distinguish individual display pixels, much less being able to do precision pointing to a specified pixel. In fact, higher-resolution displays will allow more novel vernier-based interfaces based on rotational movements. We could project polar grid coordinates and verniers around the grid just as the case for rectilinear grids.

CONCLUSIONS

We have proposed a sub-pixel-pitch dragging and feedback technique called popup vernier. A user can invoke the popup vernier while dragging an object with a trigger button (typically the right button of a mouse), which allows him/her to make transition to fine-grained mode for precision dragging. The grid and the vernier serves as the feedback indicator for sub-pixel resolution object positioning. The grid moves counter-directional to the object being dragged, and the precise coordinate can be intuitively obtained by the vernier scales. This allows smooth transition from coarse-grained to fine-grained object dragging, compared to previous proposals.

We have also proposed a new metric for smoothness and user comfort over multiple modes of resolution in dragging, by observing the continuity of the distance-time graph of the combined resolutions by extending the Fitts' law. The 'higher-order continuity' observed by the approach line reflects the smoothness of interaction, and we have also observed the appropriate parameters that could reflect the user's preference and comfort.

The effectiveness of popup verniers and the validity of the new metric have been confirmed with user studies. Comparisons with FineSlider and Alphaslider had been made by applying the popup vernier technique to sliders, and performing selection task from a large data set. The experiment showed that not only the popup vernier resulted in faster search times, but the approach line it generated is almost linear and is thus more compliant to Fitts' law. The second experiment involved the same dragging task, but tested the popup vernier slider with variant parameters. We confirmed that the metric under the extended Fitts' law coincided well with user's preference and comfort.

As a future work, we are going to implement popup verniers into various applications that require sub-pixel resolution dragging, due to the vast range that has to be specified with direct manipulation. We also plan to perform further experiments for other GUIs augmented with popup vernier techniques. Quantitative evaluation of the feedback method with verniers is another issue. One useful application of our new metric is automatic interface tuning and customization, which is another promising direction.

ACKNOWLEDGMENTS

We thank all the members of the chat-augmented conference project for WISS'97. We also thank the members of Masuda Laboratory of University of Tokyo, Sony Computer Science Laboratory, the TRIP user interface group, and all the participants of our experiments.

REFERENCES

1. C. Ahlberg and B. Shneiderman. The Alphaslider: A compact and rapid selector. In *Proceedings of ACM Conference on Computer-Human Interaction*, pages 365–371. ACM, April 1994.

2. D. B. Diner. Sub-pixel resolution in 3-d television for teleoperation. In *International Conference on Systems, Man, and Cybernetics*, 1991.

3. P. M. Fitts and J. R. Peterson. Information capacity of discrete motor responses. *Journal of Experimental Psychology*, (67):103–112, 1964.

4. K. Kashiwagi, G. R. Borden IV, and T. Masui. "FineSlider": a high precision slider (in Japanese). In *Tenth Human Interface Symposium of SICE*, pages 297–300, October 1994.

5. T. Masui, K. Kashiwagi, and G. R. Borden IV. Elastic graphical interfaces for precise data manipulation. In *CHI'95 Conference Companion*, pages 143–144. ACM, 1995.

6. M. Osada, H.Liao, and B. Shneiderman. Alphaslider: Seaching textual lists with sliders. In *Proceedings of Ninth Annual Japanese Conference on Human Interface*, October 1993.

7. Jun Rekimoto, Yuji Ayatsuka, Hirotaka Uoi, and Toshifumi Arai. Adding another communication channel to reality: An experience with a chat-augmented conference. In *CHI'98 Summary (late breaking results)*, pages pp.271–272. ACM, April 1998.