RAPPORT DE RECHERCHE

LRI

SEMANTIC POINTING : IMPROVING TARGET ACQUISITION WITH CONTROL-DISPLAY RATIO ADAPTATION

BLANCH R / GUIARD Y / BEAUDOUIN-LAFON M

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud-LRI

10/2003

Rapport de Recherche N° 1373

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 650
91405 ORSAY Cedex (France)

Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation

Renaud Blanch
LRI & INRIA Futurs*
Université Paris-Sud
Orsay, France
blanch@lri.fr

Yves Guiard
Mouvement et Perception
CNRS & Université de la Méditerranée
Marseille, France
guiard@laps.univ-mrs.fr

Michel Beaudouin-Lafon
LRI & INRIA Futurs*
Université Paris-Sud
Orsay, France
mbl@lri.fr

ABSTRACT

We introduce semantic pointing, a novel interaction technique that improves target acquisition in graphical user interfaces (GUIs). Semantic pointing uses two independent sizes for each potential target presented to the user: one size in motor space adapted to its importance for the manipulation, and one size in visual space adapted to the amount of information it conveys. This decoupling between visual and motor size is achieved by changing the control-to-display ratio according to cursor distance to nearby targets. We present a controlled experiment supporting our hypothesis that the performance of semantic pointing is given by Fitts' index of difficulty in motor rather than visual space. We apply semantic pointing to the redesign of traditional GUI widgets by taking advantage of the independent manipulation of motor and visual widget sizes.

Author Keywords

Control-display ratio, Fitts' law, graphical user interface, pointing, semantic pointing

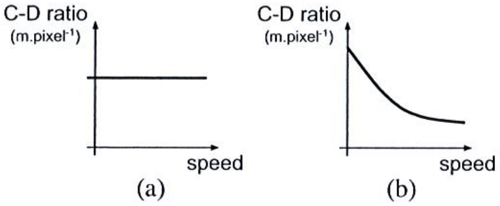
ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces – Graphical user interfaces, Input devices and strategies, Interaction styles, Theory and methods

INTRODUCTION

Pointing is a fundamental task in graphical user interfaces (GUIs). To help manage the growing complexity of software, such as the increasing number of toolbars and menu commands, the HCI literature has introduced new interaction techniques that attempt to reduce pointing time. This paper explores the idea of assigning two separate sizes for objects in the interface: a visual size for display, and a motor size reflecting the importance of the object for interaction. We hypothesize that task difficulty depends on the motor, not visual size of objects, and control the motor size by adapting the control-display (C-D) ratio. We call this technique *semantic pointing*, since motor sizes are used to reflect the local semantics of the screen.

Fitts' law [7] is widely used to design and evaluate interaction techniques and input devices [16]. It links the movement time *(MT)* to acquire a target to the task's index of difficulty *(ID)*. *ID* is the logarithm of the ratio between the target distance *(D)* and its width *(W)*. *MT* is a linear function of *ID* characterizing the system. The implications of Fitts' law have been used in several techniques to facilitate pointing tasks by enlarging the target or by reducing its distance [5, 2, 18, 6, 23].



(a) (b)

Figure 1: C-D ratio as a function of mouse speed
(a) constant C-D ratio (b) mouse acceleration

Control-display ratio adaptation [13, 22, 6] is another approach for facilitating target acquisition. This technique improves pointing performance but has not been analyzed in terms of Fitts' law, and its possible use in real GUIs has not been fully explored. The C-D ratio [17] is a coefficient that maps the pointing device physical displacement to the resulting on-screen cursor movement in a system where there is an indirection between the pointing device and the display (typically with a mouse). The C-D ratio defines the distance the mouse has to cover in the physical world (*dx* in meters) to move the cursor on the screen by a given distance (*dX* in pixels)[1]. The C-D ratio is dx/dX. A typical C-D ratio adaptation is the so-called mouse "acceleration". The cursor moves over a larger distance when the mouse covers a

---

*projet InSitu – Pôle Commun de Recherche en Informatique du plateau de saclay – CNRS, École Polytechnique, INRIA, Université Paris-Sud.

[1] We use the following conventions: capital letters denote quantities (e.g. distances) concerning the screen, and lower case letters, the physical world. For distances, we use two different units (pixels and meters respectively) to help understand ratios that would otherwise be dimensionless.

given amplitude more quickly (Figure 1), capturing an intention: when users move the physical device fast, they typically wish to go further, so the cursor can be displaced even faster to cover the distance more quickly. Other techniques use C-D ratio adaptation to facilitate pointing tasks [13, 22].

After reviewing previous work on facilitating target acquisition, we describe semantic pointing and predict it's effect on pointing performance in terms of Fitt's law. We then describe a controlled experiment that tests our predictions. Finally, we illustrate potential applications of semantic pointing to GUI design.

RELATED WORK

Growing Target and Shrinking Distance

With respect to Fitts' law, there are two simple ways to reduce the difficulty of a pointing task: enlarging the target or moving it closer to the cursor. Both have been explored in several ways. A widely-used direct application of this principle is contextual pop-up menus. Such menus are displayed at the cursor location so that distances to the items are minimal. Pie menus [5] are even more radical: the distance from each menu item to the cursor is constant and very small. The distance can also be reduced by moving the potential targets of a directed movement towards the cursor, as in the drag-and-pop technique [2].

Another approach consists of modifying target size when the cursor is close enough. This can be achieved by magnifying the target [18], or by adding a "bubble" around it [6]. Evaluations and comparisons with other techniques [22, 6] show that target resizing facilitates pointing even if the expansion is late and unpredictable [18, 23]. The problem in applying such techniques to real GUIs is that in order to expand a target surrounded by other possible targets, its neighborhood must be shrunk and the magnified target then moves when the expansion focus changes [11]. As a consequence, no performance improvement can be observed for systems like the Mac OS X Dock [18, 23]. More generally, techniques that dynamically change the screen layout cause a spatial disorganization that limits their expected benefits.

Adaptive Control-Display Ratio

The C-D ratio is the ratio of the movement of the input device and the movement of the object it controls. The C-D ratio can be a constant (Figure 1a) or, as in mouse "acceleration", a function of mouse speed (Figure 1b). In order to improve target acquisition, the C-D ratio is typically a function of cursor position [22]. Increasing the C-D ratio when the cursor is within a target makes, at constant mouse speed, the cursor slow down: covering the same number of pixels requires moving the mouse by a longer displacement. Figure 2 illustrates this technique in one-dimensional (1D) space. The slope of the function mapping the screen to the physical world is the inverse of the C-D ratio. Within the target (shown as a thick black line), the C-D ratio is increased. Since the cursor stays longer within the target, it is easier for the user to acquire it.

Swaminathan and Sato [21] concluded that in the context of large displays "nonlinear mappings are too counterintuitive to be a general solution for pointer movement". However such non-linear mappings have been successfully applied to 3D rotations [20] and 3D navigation [4], and most studies on pointing with C-D ratio adaptation [13, 22, 6] show a performance improvement. However, the effects of C-D ratio adaptation have always been interpreted in terms of feedback—"sticky" icons [22], pseudo-haptic feedback [14]—and have not been analyzed in terms of Fitt's law.

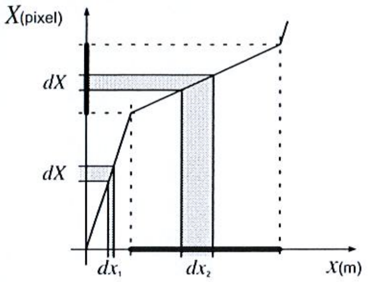


Figure 2: C-D ratio adapted to a target

Fisheye Views and Zoomable Space

As explained below, C-D ratio adaptation can also be interpreted as a dynamic magnification of the physical motor space where the mouse movements take place. This relates to fisheye views and zoomable interfaces that use a local or global magnification of the visual space.

Fisheye views locally distort a visualization by magnifying a particular point—the focus—and contracting its neighborhood—the context—according to a degree of interest function based on *a priori* importance and distance to focus [8]. It has been applied to a variety of contexts and its impact on pointing as been studied [11]. As noted above, such techniques expand target sizes but the movements resulting from a moving focus impair target acquisition [18, 23]. Even fine-tuned versions of fisheye views do not compete with other techniques: hierarchical menus are better than fisheye menus [3], flat representations are better than distorted ones for focus targeting [11].

Igarashi and Hinckley's navigation technique [12] uses speed-dependent automatic zooming to enhance scrolling. It manipulates view magnification to keep a constant optical flow while scrolling at variable speed. Evaluations did not show a quantitative benefit on task completion time. This may be because the magnification level is chosen by the user—even indirectly through the scrolling speed—and so does not automatically adapt to the task.

Furnas and Bederson introduced space-scale diagrams [9] to represent zoomable interfaces. Using a metric on trajectories—the amount of information needed to define them—they predict optimal trajectories and observe that they match those empirically chosen by users, a fact confirmed by Guiard *et al.* [10]. Optimal trajectories have a scale adapted to the distance to the target: as the cursor approaches the target, the visual space is magnified, the cursor thus gains precision while slowing down in target space[2] (Figure 3).

---

[2] The target space is not directly the screen space because magnification introduces a scaling factor.

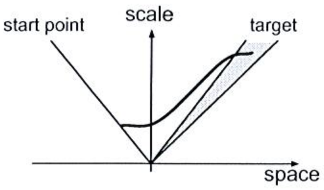


Figure 3: Space-scale diagram
The trajectory shown in black illustrates scale as a decreasing function of target distance.

In summary, many researchers have used, explicitly or implicitly, the effects of C-D ratio adaptation to improve pointing performance, yet there is no unified approach for understanding these effects.

SEMANTIC POINTING

Semantic pointing relies on the following hypothesis: the difficulty of a pointing task is not directly linked to the on-screen representation of the task, but to the actual difficulty of the movement performed in the physical world to accomplish it. We first show that C-D ratio modification can be interpreted as a manipulation of the relative sizes of objects in visual and motor space. We then describe how semantic pointing computes the C-D ratio as a function of a context known to the system, namely, the distances from the cursor to potential targets.

Control-Display Ratio as Motor Space Magnification

The C-D ratio defines the ratio between distances for the physical device and distances on the screen, or how the motor space is projected into the visual space. If the C-D ratio is a constant, for a 1D world this projection is a linear function linking motor space (x in meters) to visual space (X in pixels) as illustrated in Figure 4. The slope of this function—in $pixel.m^{-1}$—is the C-D gain and the inverse of this gain—in $m.pixel^{-1}$—is the C-D ratio.

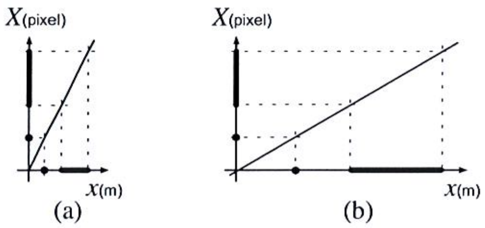


Figure 4: C-D ratio as motor space scale
(a) low C-D ratio (b) high C-D ratio

When the C-D ratio is low (Figure 4a) the motor space is contracted compared to a higher C-D ratio (Figure 4b). In fact, the C-D ratio can be seen as the *motor space scale* relative to the visual space (called simply "scale" in the rest of the paper). At low scale, movement to acquire a target is short but target size is also small. On the other hand, at high scale, the target distance is longer but the accuracy needed to acquire it is reduced. In any case the task difficulty remains the same since it is characterized by the non-dimensional ratio D/W which is insensitive to a uniform scaling. In other words, uniform scaling does not affect pointing task ID. This illustrates the trade-off between target distance and target size [17].

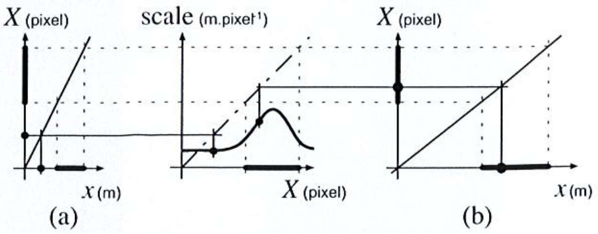


Figure 5: Context-dependent motor space scale

The principle of semantic pointing is to dynamically adapt the motor space scale to reduce both target distance and the accuracy needed to acquire it. The idea is to choose a low scale, adapted to the task extension D, when the cursor is far from any goal (Figure 5a) and to choose a high scale, adapted to the task precision W, when the cursor is close enough to the target (Figure 5b). So, without changing the visual layout, the target is both closer and bigger in motor space.

Motor Space Deformation and Local Information Density

Instead of interpreting the contextual C-D ratio adaptation as the dynamic change of a linear function slope, we can interpret it as the local slope of a certain non-linear function. This is equivalent because the scale is only a function of position. As in a fisheye view, scale becomes a local property: some areas are expanded while others shrink. The scale function can be chosen so that the resulting distorted motor space has the following property: important areas for interaction, such as pointing targets, are bigger while non-important areas, such as empty space, are shrunken (Figure 6). In empty space, accuracy is less necessary than speed, while near a possible target, accuracy becomes more important than speed. The distortion is then consistent with the objective of aiding target acquisition.

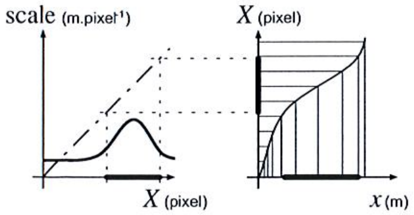


Figure 6: Non-uniform motor space scale

This "need for accuracy" is not uniform across the screen and depends, for each pixel, on whether it is part of a potential target, e.g. a button or icon, or part of empty space, e.g. a window background. As noticed in a companion paper [1], there is a mismatch between the abstract selection task and a pointing task on screen. Selecting an icon on a typical desktop consists of pointing a 48×48 pixels target on a 1600×1200 pixels screen, *i.e.*, providing $log_2(\frac{1600\times1200}{48\times48}) \sim 10$ bits of information to the system. For the user as well as for the system however, the real information is only the choice of one icon within those present on the desktop. Choosing one icon among a set of 64 only requires $log_2(64) = 6$ bits of information[3].

---

[3] Those bits can be understood by thinking of a binary search: the target is (or not) in the first half of the set. This boolean is one bit of information and now the target is in a known set of 32 icons, *etc*.

By making scale dependent on pixel semantics, semantic pointing makes important pixels bigger in motor space, and thus helps to reduce the mismatch between the abstract task of selection and its execution. By using information that is normally known to but ignored by the system, namely the potential targets, the amount of information the user need provide to the system is reduced. The local "need for accuracy" is in fact a local information density expressing how much each pixel is relevant to the manipulation.

Scale Formulation

The simplest scale function that magnifies the pixels within a target pixels is a step function (Figure 7). It can be defined using the rectangle function:

$$\Pi(u) = \begin{cases} 1 & for\ |u| \le \frac{1}{2} \\ 0 & otherwise \end{cases}$$

For a target of size W at coordinate D the scale is then[4]:

$$\text{scale}(X) = (1 - \Pi(\tfrac{X-D}{W})) + S \times \Pi(\tfrac{X-D}{W})$$

The first term of the sum takes the value 1 in empty space and 0 within the target whereas the second term takes the value S within the target and 0 outside. A generalized version for multiple targets is:

$$\text{scale}(X) = (1 - \sum_i \Pi(\tfrac{X-D_i}{W_i})) + \sum_i S_i \times \Pi(\tfrac{X-D_i}{W_i}),$$

where D_n, W_n and S_n are the position, size and scale of the nth target (Figure 7).

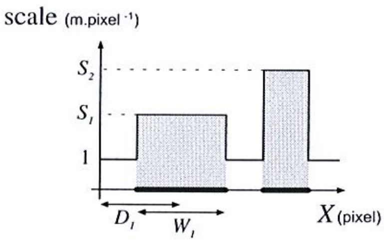


Figure 7: Scale as step function

Index of Difficulty in Motor Space

Since in a 1D world there is only one possible path linking two points, the target size in motor space can be computed by integrating the scale function over the target:

$$\begin{aligned} w &= \int_{D-\frac{W}{2}}^{D+\frac{W}{2}} \text{scale}(X)\, dX \\ &= S \times W \end{aligned}$$

This size is the area of the grayed rectangles of Figure 7. If there is only one target[5] we can similarly compute the target distance in motor space:

$$\begin{aligned} d &= \int_0^D \text{scale}(X)\, dX \\ &= (D - \tfrac{W}{2}) + S \times \tfrac{W}{2} \end{aligned}$$

The first term of this sum corresponds to the target distance in empty space and the second term to the supplemental distance added by the magnification of the target as the end of the movement runs across half of it. When D is small, the first term becomes negligible because the section of D that overlaps empty space tends toward zero whereas when D is much larger than $W/2$, the second term becomes negligible. We can now predict the index of difficulty of the task[6] in motor space *(id)* as a function of the usual index of difficulty in visual space *(ID)* when the scale of the target is *S:*

$$\begin{aligned} id &= log_2(\tfrac{d}{w/2}) \\ &\xrightarrow{D \to W/2} log_2(\tfrac{S \times D}{S \times W/2}) = ID \\ &\xrightarrow{D \gg W/2} log_2(\tfrac{D}{S \times W/2}) = ID - log_2(S) \end{aligned}$$

For difficult tasks (when D is much larger than $W/2$), the task difficulty is reduced by the number of bits of the motor scale. Figure 8 shows the resulting gain for a motor scale of 2 and 4. It shows that, for large *ID*s, the difficulty in motor space gains one bit each time the scale doubles.

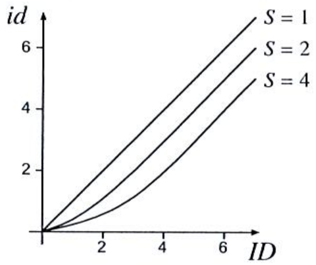


Figure 8: Index of difficulty in motor space

Bell-Shaped Mixing Function

Instead of using a simple rectangle function (Π), we use a bell-shaped function (Ω) as a mixing function to avoid discontinuities in the scale function. Figure 9 shows the differences between Π and Ω.

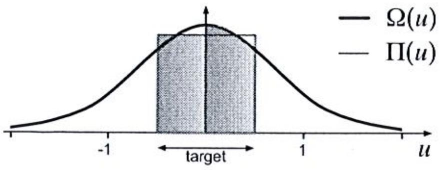


Figure 9: Mixing functions
The bell function avoids discontinuities in the scale.
The grayed surfaces have the same area.

Like Π, Ω has been chosen to be compliant with the following fundamental requirements: correctly scaling the targets, *i. e.*, $\int_{-1/2}^{1/2} \Omega(u)\, du = 1$, and rapidly decreasing towards zero when outside a target, *i. e.*, $\int_{1/2}^{\infty} \Omega(u)\, du \le 1$ for example:

$$\Omega(u) = \frac{\ln(3)}{\cosh^2(\ln(3) \times u)}$$

---

[4] $X - D$ is the distance from the cursor *(X)* to the target *(D)* in a 1D world. In higher dimensions, $X - D$ must be replaced by the euclidian distance to the target.

[5] The influence of distractors will be discussed later.

[6] We chose Fitts' formulation of the *ID*, rather than Shannon's [15], for the sake of convenience (analytical calculations are thus possible). It should be mentioned that this option has no effect on the bottom line of our argument.

This particular function was chosen because its integral can be computed analytically. The scale function then becomes:

$$\mathrm{scale}(X) = (1 - \sum_i \Omega(\tfrac{X-D_i}{W_i})) + \sum_i S_i \times \Omega(\tfrac{X-D_i}{W_i}),$$

and the relationship between *id* and *ID* has the same characteristics as with the rectangle version (Figure 8).

EXPERIMENT

We conducted a controlled experiment to evaluate the benefits of semantic pointing as predicted by our analysis.

Task

Participants had to perform successive 1D discrete pointing tasks. They had to move the cursor, represented by a one pixel thick vertical black line, to the start position marked by a gray rectangle on the left of the screen, rest there for 0.5 s, start moving to the target—a blue rectangle—as soon as it appeared on the right, and click it (Figure 10).

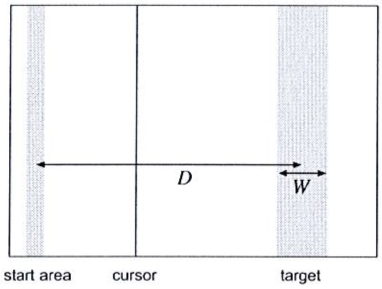


Figure 10: Screen layout

After each click, a signal indicated whether or not they missed the target. After each block, their error rates were displayed and they were encouraged to conform to a nominal 4% error rate by speeding up or slowing down.

Conditions & Procedure

The first *(control)* condition held the scale constant, so the motor task was exactly the same as displayed. In the second *(double)* and third *(quadruple)* conditions, the scale was adapted accordingly to our model so that the target size was either doubled and quadrupled in motor space. We used the bell-shaped mixing function.

Five *ID*s (4, 5, 6, 7 and 8) and two sizes ($D = 512$ or 1024 pixels) were used, giving ten possible tasks. Each condition repeated each task 10 times, resulting in a total of 100 trials presented in pseudo-random order. This series consisted of every possible successive pair of tasks counter-balanced to account for order effects. This series was split into two blocks of 50 trials each (*A* and *B*).

Each participant performed in this order six blocks of tasks (two for each of the three conditions): *A*0 (block *A*, condition 0), *B*1, *A*2, *B*0, *A*1, *B*2. The six permutations of the conditions order were repeated for pairs of subjects. Each block was preceded by 10 randomly-chosen tasks using the same condition to train the participants. This was chosen after a pilot study suggested that, after ten trials, the movement times were stable.

Subjects

Twelve unpaid adult volunteers, 11 male and one female, aged 27.2 years on average ($SD = 6.10$ years), served in the experiment.

Apparatus

The pointing experiment was conducted on a 22-inch 1600 × 1200 resolution color monitor, using a Wacom Intuos 12×18 inch digitizing tablet with a puck. The baseline C-D ratio was set at the screen resolution (1*cm* on screen for 1*cm* in motor space) and the system had no mouse acceleration.

RESULTS

The effects of semantic pointing are explored by analyzing three dependent variables: reaction time *(RT)*, movement time *(MT)*, and error rate *(ER)*. Repeated measures analyses of variance were performed on these three variables. We analyzed the effects of the three factors (3 conditions, 5 index of difficulty, and 2 sizes) within participants.

Non-Significant Effects

Effect of the Task Size

No effect of the task size *(D)* on the three dependent variables was found to be statistically significant. This is consistent with Fitts' law and our model: both state that the performance of target acquisition is a function of the non-dimensional ratio *D/W*. The size effect is thus neglected for the rest of the analysis, and the following plots merge the two task sizes for each *ID*.

Effect on Reaction Time

The reaction time *(RT)* was about 253 ms on average with small variations ($SD = 75.76$ ms). *RT* grew slightly with the task index of difficulty but this effect was not statistically significant. No significant difference was found among the three conditions.

Semantic Pointing Effect on Task Completion Time

The movement time *(MT)* as a function of the index of difficulty *(ID)* is plotted for the three conditions in Figure 11. There was a significant effect of condition ($F_{2,33} = 5.35$, $p = .0097$) and *ID* ($F_{4,55} = 30.04$, $p < .0001$) on *MT* but no significant interaction between the two factors was found.

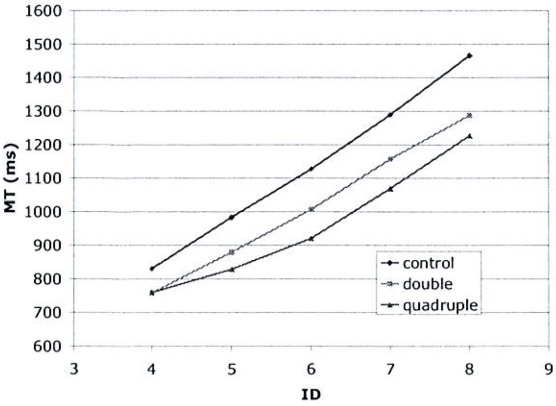


Figure 11: Movement time vs. index of difficulty

As predicted, the benefit of semantic pointing first grows with *ID* before remaining almost constant for difficult tasks. The maximum relative gain on task completion time is obtained for $ID = 6$ but for $ID \geq 5$ the *MT* reduction is at least 10% (10.9% on average) for the *double* condition and at least 15% (16.9% on average) for the *quadruple* condition.

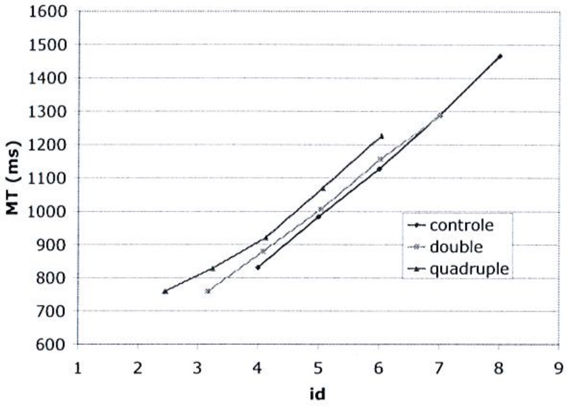


Figure 12: Movement time vs. motor index of difficulty

Figure 12 shows movement time as a function of index of difficulty in motor space *(id)*. If our hypothesis that the performance of semantic pointing is given by Fitts' index of difficulty in motor rather than visual space is correct, the three conditions should be superimposed, which is nearly the case. However, we can note that for the *quadruple* condition, the benefit of semantic pointing is less than that predicted by the model. The study of the error rate will provide us with an explanation of those slight differences.

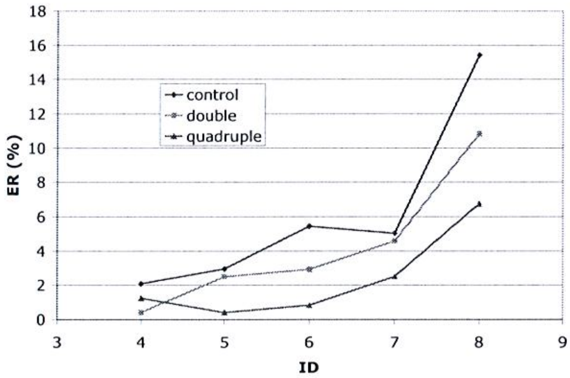


Figure 13: Error rate vs. index of difficulty

Effect on Error Rate

The participants were told to conform to a nominal 4% error rate *(ER)* on each block. The mean *ER* is in fact 4.26% but the differences between the three conditions are significant (Figure 13). On average, the *ER* was 6.2% in the *control* condition, 4.25% in the *double* one, and 2.35% in the *quadruple* one. The *double* and *quadruple* conditions were more accurate than the *control* one for every *ID*, and, except for $ID = 4$, *quadruple* had a better *ER* than the other two conditions. So the reason why semantic pointing is not fully exploited to reduce target acquisition time is because it also serves to reinforce selection accuracy.

Between Subject Variations

This argument is confirmed if we take a closer look at individual performances. The means reported in the foregoing are representative of most subjects, but there were individual strategies. Some participants took advantage of semantic pointing essentially by reducing their error rate; others conformed to the constant error rate requirement. Importantly, it is in the latter category of participants that performance was rigorously governed by the motor component of the task. This result confirms that semantic pointing unquestionably facilitates pointing, with this facilitation effect benefiting to various extents to target acquisition time and/or pointing accuracy. Furthermore, we conducted informal testing of a 2D desktop prototype. We observed that users did not realize when semantic pointing was on or off and yet took advantage of it to improve their performance.

DESIGN PERSPECTIVES

In this section we present applications of semantic pointing to GUI design.

Semantic Importance as a New Attribute

In traditional GUIs, the size of an object is determined by visualization and manipulation constraints: the object must be big enough for the relevant information to be accessible to the user, and for the user to be able to manipulate it. When an object conveys little information, such as a button or scroll-bar, the size is determined by the manipulation constraint, wasting screen real-estate. Conversely, when a lot of information must be displayed, such as in a web page, the parts that can be manipulated, such as the links, may end up very small and difficult to interact with.

Semantic pointing resolves such conflicts by allowing two sizes to be manipulated independently: the size in visual space, constrained by the information to be displayed, and the size in motor space, constrained by the importance of the object for manipulation. These sizes are manipulated through a new attribute, semantic importance (*si*), which acts as the scale of motor-space size relative to visual-space size. When $0 < si < 1$, the object is smaller in motor space than in visual space, which is appropriate for objects that are not manipulated; when $si > 1$, the object is bigger in motor-space than in visual space, making it easier to manipulate; $si = 1$ corresponds to traditional GUIs.

Traditional GUI Widgets Redesign

In order to redesign traditional GUI widgets such as scroll-bars, menus and buttons, we considered two aspects:

- How much information does it provide to the user?
- How important is it for the manipulation?

We show that semantic pointing can either reduce the screen footprint of widgets without affecting the interaction, or facilitate interaction without affecting the screen layout.

Scroll-bars

The information provided by a traditional scroll-bar is rather poor: it specifies a position in the document and sometimes the proportion of the document that is currently displayed in the view. A typical scroll-bar uses a 15 pixels wide strip along the whole window (Figure 14a). However the same information can be conveyed by a much thinner strip, e.g. 3 pixels (Figure 14b). In order to still be able to manipulate the thumb and arrow buttons, these are given a semantic importance of 5 so as to be as big in motor space as in the original design (Figure 14c[7]).

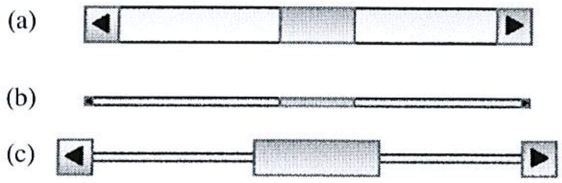
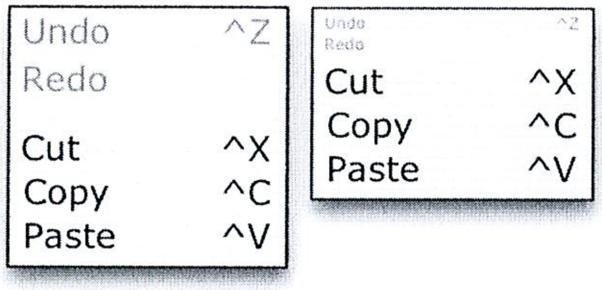


Figure 14: Scroll-bar redesign
(a) original version. (b) new version: visual space (what it looks like) and (c) motor space (what it feels like when interacting with it).

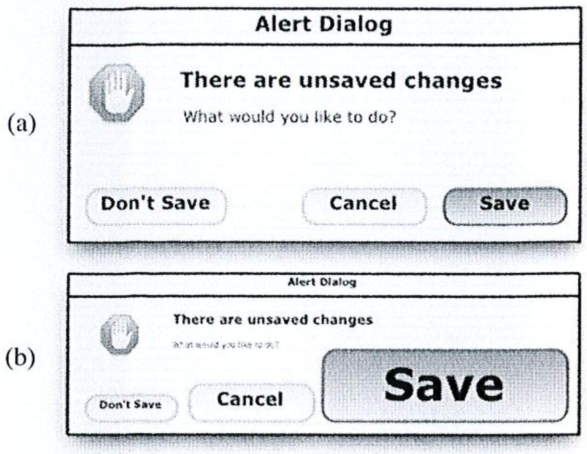
Menus
The main real-estate constraint for menus is that labels must be readable, so the visual size of menu items cannot be reduced significantly (Figure 15a). However, the importance of menu items with respect to manipulation is variable. Disabled items and separators cannot be selected, so they can be given a small semantic importance, reducing the distance in motor space from the top of the menu to the items below them (Figure 15b).



(a) (b)

Figure 15: Menu redesign
(a) unchanged visual version (b) motor space version

Buttons & Hyperlinks
As for menu items, the buttons and messages of a dialog box must be readable (Figure 16a). However, for the manipulation, only the buttons are relevant, so the rest of the box can be shrunken. Furthermore, the importance of the various buttons need not be equal. The default button, assumed to be the most likely choice, can be given a higher importance. More generally, the importance can be proportional to the probability of being selected (Figure 16b). "Dangerous" buttons can also be given a smaller importance to make them harder to select.



There are unsaved changes
What would you like to do?

There are unsaved changes

Figure 16: Button redesign
(a) unchanged visual version (b) motor space version

In a similar way, the visual layout of rich documents such as web pages is often designed with aesthetics and visual communication in mind. But as far as interaction with such hyperdocuments is concerned, only the hyperlinks matter. Therefore, magnifying the hyperlinks in motor space should help users acquire them and improve navigation.

Semantic Importance as a Dynamic Degree of Interest

So far we have mostly considered semantic importance as a static attribute of interface objects. An exception was menu items, whose importance vary according to their state: a disabled item has a low importance, which becomes high when the item is enabled. The same applies to disabled buttons in a dialog box. Another example where semantic importance can reflect the state of an object is the application icons in current desktops. When an application requires user attention, it blinks in the Microsoft Windows task-bar or its icon is animated in the Mac OS X Dock. Since the user is likely to click such an icon to activate the application, it would help to magnify it in motor space.

More elaborate strategies can be used to compute the semantic importance according to the state and history of the interaction. For example, applications in the Microsoft Office Suite have adaptive menus that reconfigure themselves so that the most often used items are at the beginning of the menu. The instability of menus is known to be a source of confusion for users [19]. With semantic pointing, the importance of menu items can match the frequency of their use. This has a positive effect similar to adaptive menus, i.e. often-used commands are easier to reach, without disturbing the spatial layout of the menu items.

CONCLUSION & FUTURE WORK

In this paper we introduced semantic pointing, a technique that decouples motor space from visual space to improve pointing performance. We showed how to use C-D ratio adaptation to control the mapping between motor and visual space, interpreting it as a motor-space scale. We also showed that the index of difficulty of a pointing task is defined by the

---

[7] The motor space distortion caused by semantic pointing is not accurately representable in euclidian geometry. Thus the representations in motor space cannot be exact and are given for illustration purpose.

size of the target in motor rather than visual space, validating the hypothesis behind semantic pointing. In addition, we observed that users did not notice the distortion introduced by semantic pointing, making the technique effective and yet transparent.

We presented several applications of semantic pointing to improve the design of traditional GUIs, by specifying the two sizes of each object with a new attribute: semantic importance. In some cases, the visual footprint of objects is reduced without changing their motor size, saving screen real-estate, while in other cases the visual layout is left untouched but the motor space is enlarged in order to facilitate interaction. We have also shown how the semantic importance of an object can change over time to adapt to the user needs.

Our future work will concentrate on the problem of distractors. The presence of a potential target on the path of a pointing movement increases the distance of the real target, thus reducing the benefit of semantic pointing[8]. We intend to study these effects systematically and experiment with techniques to minimize them. We also intend to explore more applications of semantic pointing and evaluate it in real settings.

ACKNOWLEDGMENTS

The authors would like to thank the whole InSitu team, and especially Wendy Mackay, Caroline Appert and Stéphane Conversy, for their useful comments and suggestions about this work.

REFERENCES

[1] A. Anonymized. Vector pointing: Object vs. pixel selection in graphical user interfaces. Paper submitted to CHI 2004.

[2] P. Baudish, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger. Drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems. In *Proc. Interact 2003*, 2003. In press.

[3] B. B. Bederson. Fisheye menus. In *Proc. UIST 2000*, pages 217–225. ACM Press, 2000.

[4] N. Burtnyk, A. Khan, G. Fitzmaurice, R. Balakrishnan, and G. Kurtenbach. Stylecam: interactive stylized 3d navigation using integrated spatial & temporal controls. In *Proc. UIST 2002*, pages 101–110. ACM Press, 2002.

[5] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. An empirical comparison of pie vs. linear menus. In *Proc. CHI 1988*, pages 95–100. ACM Press, 1988.

[6] A. Cockburn and A. Firth. Improving the acquisition of small targets. In *Proc. HCI 2003*, 2003. In press.

[7] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psycology*, 47:381–391, 1954.

[8] G. W. Furnas. Generalized fisheye views. In *Proc. CHI 1986*, pages 16–23. ACM Press, 1986.

[9] G. W. Furnas and B. B. Bederson. Space-scale diagrams: understanding multiscale interfaces. In *Proc. CHI 1995*, pages 234–241. ACM Press/Addison-Wesley Publishing Co., 1995.

[10] Y. Guiard, F. Bourgeois, D. Mottet, and M. Beaudouin-Lafon. Beyond the 10-bit barrier: Fitts'law in multiscale electronic worlds. In *Joint proc. HCI 2001 and IHM 2001*, pages 573–587. Springer Verlag, 2001.

[11] C. Gutwin. Improving focus targeting in interactive fisheye views. In *Proc. CHI 2002*, pages 267–274. ACM Press, 2002.

[12] T. Igarashi and K. Hinckley. Speed-dependent automatic zooming for browsing large documents. In *Proc. UIST 2000*, pages 139–148. ACM Press, 2000.

[13] D. V. Keyson. Dynamic cursor gain and tactual feedback in the capture of cursor movements. *Ergonomics*, 12:1287–1298, 1997.

[14] A. Lécuyer, S. Coquillart, A. Kheddar, P. Richard, and P. Coiffet. Pseudo-haptic feedback: Can isometric input devices simulate force feedback? In *Proc. VR 2000*, pages 83–90, 2000.

[15] I. S. MacKenzie. A note on the information-theoretic basis for Fitts'law. *Journal of Motor Behavior*, 21:323–330, 1989.

[16] I. S. MacKenzie. Fitts'law as a research and desgin tool in human-computer interaction. *Human-Computer Interaction*, 7:91–139, 1992.

[17] I. S. MacKenzie and S. Riddersma. Effects of output display and control-display gain on human performance in interactive systems. *Behaviour & Information Technology*, 13:328–337, 1994.

[18] M. McGuffin and R. Balakrishnan. Acquisition of expanding targets. In *Proc. CHI 2002*, pages 57–64. ACM Press, 2002.

[19] J. Mitchell and B. Shneiderman. Dynamic versus static menus: an exploratory comparison. *ACM SIGCHI Bulletin*, 20(4):33–37, 1989.

[20] I. Poupyrev, S. Weghorst, and S. Fels. Non-isomorphic 3d rotational techniques. In *Proc. CHI 2000*, pages 540–547. ACM Press, 2000.

[21] K. Swaminathan and S. Sato. Interaction design for large displays. *interactions*, 4(1):15–24, 1997.

[22] A. Worden, N. Walker, K. Bharat, and S. Hudson. Making computers easier for older adults to use: area cursors and sticky icons. In *Proc. CHI 1997*, pages 266–271. ACM Press, 1997.

[23] S. Zhai, S. Conversy, M. Beaudouin-Lafon, and Y. Guiard. Human on-line response to target expansion. In *Proc. CHI 2003*, pages 177–184. ACM Press, 2003.

---

[8] The presence of a distractor cannot make semantic pointing worse than traditional pointing in which all pixels between the cursor and the target are distractors.

RAPPORTS INTERNES AU LRI - ANNEE 2003

N°	Nom	Titre	Nbre de pages	Date parution
1345	FLANDRIN E LI H WEI B	A SUFFICIENT CONDITION FOR PANCYCLABILITY OF GRAPHS	16 PAGES	01/2003
1346	BARTH D BERTHOME P LAFOREST C VIAL S	SOME EULERIAN PARAMETERS ABOUT PERFORMANCES OF A CONVERGENCE ROUTING IN A 2D-MESH NETWORK	30 PAGES	01/2003
1347	FLANDRIN E LI H MARCZYK A WOZNIAK M	A CHVATAL-ERDOS TYPE CONDITION FOR PANCYCLABILITY	12 PAGES	01/2003
1348	AMAR D FLANDRIN E GANCARZEWICZ G WOJDA A P	BIPARTITE GRAPHS WITH EVERY MATCHING IN A CYCLE	26 PAGES	01/2003
1349	FRAIGNIAUD P GAURON P	THE CONTENT-ADDRESSABLE NETWORK D2B	26 PAGES	01/2003
1350	FAIK T SACLE J F	SOME b-CONTINUOUS CLASSES OF GRAPH	14 PAGES	01/2003
1351	FAVARON O HENNING M A	TOTAL DOMINATION IN CLAW-FREE GRAPHS WITH MINIMUM DEGREE TWO	14 PAGES	01/2003
1352	HU Z LI H	WEAK CYCLE PARTITION INVOLVING DEGREE SUM CONDITIONS	14 PAGES	02/2003
1353	JOHNEN C TIXEUIL S	ROUTE PRESERVING STABILIZATION	28 PAGES	03/2003
1354	PETITJEAN E	DESIGNING TIMED TEST CASES FROM REGION GRAPHS	14 PAGES	03/2003
1355	BERTHOME P DIALLO M FERREIRA A	GENERALIZED PARAMETRIC MULTI-TERMINAL FLOW PROBLEM	18 PAGES	03/2003
1356	FAVARON O HENNING M A	PAIRED DOMINATION IN CLAW-FREE CUBIC GRAPHS	16 PAGES	03/2003
1357	JOHNEN C PETIT F TIXEUIL S	AUTO-STABILISATION ET PROTOCOLES RESEAU	26 PAGES	03/2003
1358	FRANOVA M	LA "FOLIE" DE BRUNELLESCHI ET LA CONCEPTION DES SYSTEMES COMPLEXES	26 PAGES	04/2003
1359	HERAULT T LASSAIGNE R MAGNIETTE F PEYRONNET S	APPROXIMATE PROBABILISTIC MODEL CHECKING	18 PAGES	01/2003
1360	HU Z LI H	A NOTE ON ORE CONDITION AND CYCLE STRUCTURE	10 PAGES	04/2003
1361	DELAET S DUCOURTHIAL B TIXEUIL S	SELF-STABILIZATION WITH r-OPERATORS IN UNRELIABLE DIRECTED NETWORKS	24 PAGES	04/2003
1362	YAO J Y	RAPPORT SCIENTIFIQUE PRESENTE POUR L'OBTENTION D'UNE HABILITATION A DIRIGER DES RECHERCHES	72 PAGES	07/2003
1363	ROUSSEL N EVANS H HANSEN H	MIRRORSPACE : USING PROXIMITY AS AN INTERFACE TO VIDEO-MEDIATED COMMUNICATION	10 PAGES	07/2003

RAPPORTS INTERNES AU LRI - ANNEE 2003

N°	Nom	Titre	Nbre de pages	Date parution
1364	GOURAUD S D	GENERATION DE TESTS A L'AIDE D'OUTILS COMBINATOIRES : PREMIERS RESULTATS EXPERIMENTAUX	24 PAGES	07/2003
1365	BADIS H AL AGHA K	DISTRIBUTED ALGORITHMS FOR SINGLE AND MULTIPLE-METRIC LINK STATE QoS ROUTING	22 PAGES	07/2003
1366	FILLIATRE J C	WHY : A MULTI-LANGUAGE MULTI-PROVER VERIFICATION TOOL	20 PAGES	09/2003
1367	FILLIATRE J C	A THEORY OF MONADS PARAMETERIZED BY EFFECTS	18 PAGES	09/2003
1368	FILLIATRE J C	HASH CONSING IN AN ML FRAMEWORK	14 PAGES	09/2003
1369	FILLIATRE J C	DESIGN OF A PROOF ASSISTANT : COQ VERSION 7	16 PAGES	09/2003
1370	HERMAN T TIXEUIL S	A DISTRIBUTED TDMA SLOT ASSIGNMENT ALGORITHM FOR WIRELESS SENSOR NETWORKS	32 PAGES	09/2003