

In : Guiard, Y. & Beaudouin-Lafon, M. (Eds.). Fitts' Law Fifty Years Later: Applications and Contributions from Human-Computer Interaction (pp. 875-905). A special Issue of the *International Journal of Human-Computer Studies*, 2004, Volume 61, Issue 6.

Target Acquisition in Multiscale Electronic Worlds

Yves Guiard ¹ Michel Beaudouin-Lafon ²

¹ Mouvement et Perception
CNRS & Université de la Méditerranée, France
+33 491 172 257
guiard@laps.univ-mrs.fr

² Laboratoire de Recherche en Informatique
CNRS & Université Paris-Sud, France
+33 169 156 910
mbl@lri.fr

Abstract

Since the advent of graphical user interfaces, electronic information has grown exponentially, whereas the size of screen displays has stayed almost the same. Multiscale interfaces were designed to address this mismatch, allowing users to adjust the scale at which they interact with information objects. Although the technology has progressed quickly, the theory has lagged behind. Multiscale interfaces pose a stimulating theoretical challenge, reformulating the classic target-acquisition problem from the physical world into an infinitely rescalable electronic world. We address this challenge by extending Fitts' original pointing paradigm: we introduce the *scale* variable, thus defining a multiscale pointing paradigm. This article reports on our theoretical and empirical results. We show that target-acquisition performance in a zooming interface must obey Fitts' law, and more specifically, that target-acquisition time must be proportional to the index of difficulty. Moreover, we complement Fitts' law by accounting for the effect of view size on pointing performance, showing that performance bandwidth is proportional to view size, up to a ceiling effect. The first empirical study shows that Fitts' law does apply to a zoomable interface for indices of difficulty up to and beyond 30 bits, whereas classical Fitts' law studies have been confined in the 2-10 bit range. The second study demonstrates a strong interaction between view size and task difficulty for multiscale pointing, and shows a surprisingly low ceiling. We conclude with implications of these findings for the design of multiscale user interfaces.

1. INTRODUCTION

According to Fitts' law (Fitts, 1954), the minimum time T needed to reach a target of width W located at a distance D can be predicted with $T = a + b * ID$, where ID , the index of difficulty, is defined as $\log_2(D/W + 1)$. This simple relation, one of the few well-established quantitative laws of experimental Psychology (Kelso, 1992), has been shown to hold in an impressive variety of tasks and situations (Plamondon and Halimi, 1997).

Fitts' law has been quite useful to the field of Human-Computer Interaction (HCI) for about a quarter century. As soon as the first graphical user interfaces (GUIs) became available, HCI researchers acknowledged that Fitts' pointing paradigm could serve to evaluate the design of interaction techniques and input devices in a rigorous and principled fashion (Card, English and Burr, 1978). This pioneering study of Card and colleagues paved the way for the development within the HCI community of an active research stream on Fitts' law (MacKenzie, 1992; Soukoreff & MacKenzie, this issue).

The target acquisition problem that HCI research has been handling since the appearance of graphical interfaces does not differ in essence from Fitts' original problem. Indirect pointing, where a screen cursor stands for the hand and a graphical object stands for the target, was already customary in laboratories of applied and basic Psychology (e.g., Crossman, 1960). In the nineteen nineties, however, a radical change occurred in HCI: researchers introduced multiscale or zoomable interfaces (Perlin & Fox, 1993, Furnas & Bederson, 1995), allowing *multiscale navigation*.¹ Since then, this innovation has been gradually generalized to many of the applications we use today. This change, we believe, involves a real challenge to Fitts' law. With a zoomable user interface, users can move a cursor to some target object (the standard pointing problem) or move their view to some target view in an electronic world (a minor variant of the standard pointing problem, as we will see below), but they can now freely adjust the scale at which they wish to interact with the electronic world. This is an entirely novel category of user action, one that has no counterpart in the real world and could hardly be anticipated in the context of experimental Psychology.

In this article we demonstrate that Fitts' pointing paradigm still applies to multiscale interfaces. This requires enriching Fitts' classic paradigm by introducing a *scale* variable and defining the notion of multiscale pointing. A theoretical analysis leads us to a new formulation of Fitts' law for multiscale pointing, which we test with the results of two experiments. This new formulation, which involves the size of the view, has important implications for the design of user interfaces. We feel that multiscale pointing, a reality from HCI, renews in interesting ways the traditional psychological theme of target-directed movement.

¹ In this article we use the generic term "navigation" to refer to any user action that updates the visualization of a document, whether the change involves a pure spatial move (scrolling or panning), a pure scale variation (e.g., zooming), or a combination of these two categories (e.g., moving a fisheye view).

2. INFORMATION WORLDS

2.1. The Free Rescalability of Pure Information Worlds

Using the language of classical physics, the world that surrounds us involves three fundamental dimensions, mass (M), length (L), and time (T). A physical quantity that involves only length and time is said to be *kinematic*, e.g., velocity = LT^{-1} , or acceleration = LT^{-2} . A quantity that involves all three dimensions is said to be kinetic or, synonymously, *dynamic*, e.g., force = MLT^{-2} or energy = ML^2T^{-2} .

This distinction allows us to draw a sharp contrast between the real, physical world and the so-called electronic, or pure-information worlds that we create with modern computers. Whereas the former is made of atoms of matter, the latter are made of bits of information (Negroponte, 1995). The real world is *dynamic* in essence, whereas electronic worlds, which have no mass component, are worlds of pure *kinematics*.

Galileo Galilei was the first to realize that real world objects can be rescaled only up to some point. Taking the simple example of a wooden beam, he argued that such an object cannot have an arbitrarily large size: as the beam is scaled up, it would gradually lose its solidity because its resistance (a function of its cross section, dimensionally L^2) would grow at a slower rate than its weight (a function of its volume, dimensionally L^3). Therefore, were the beam scaled up, say, a thousand-fold, it would collapse under its own weight.

Galileo was considering the physical world. In an electronic world, in contrast, everything can be freely scaled up and down, without ever experiencing Galileo's problem. Not only can the objects of an electronic world be of any shape and size (e.g., a 20-km tall sky-scraper), but the user can interact at all scales with these objects. For example, the user of a zoomable electronic atlas can perform a complete orbit of the planet in a second or so, which involves a velocity on the order of 50,000 km/s, one sixth of the velocity of light. Since the atlas and the virtual camera used to view it are purely informational, mass-free replicas of our planet and an observation vehicle, the interaction suffers no upper speed limit. In fact, the only limit to navigating such a world is the user's ability to process the inflow of information to control the navigation. As we will see, this has interesting implications with respect to the boundary conditions that heavily constrain the classical version of Fitts' pointing paradigm.

2.2. Fitts' Target Acquisition Task: A Reduction Paradigm for the Study of Electronic Navigation

One major difficulty for the scientific study of human navigation in information worlds is the broad diversity of user tasks, for example, looking for a particular item, going to a particular location, or browsing serendipitously. In this research we focus on one specific task, pointing, or *target acquisition*, which was operationally defined half a century ago by Fitts (1954).

Fitts' target-acquisition paradigm, like any research paradigm, excludes many interesting research problems. We deliberately ignore the cognitive problems related to way-finding, spatial orientation and the like. In this article, the target is supposed to have been defined unequivocally from the outset and the user is assumed not only to know that the target exists, but also to know where it is and how to get there: We are just concerned with the overt act of reaching the target. We also assume that the computer user will work hard to comply with

stringent speed and accuracy instructions—in the context of Fitts' law, which relates target acquisition time T to the geometry of the task, T quantifies the *minimum* time it takes to reach the specified target.

Finally, we leave aside navigation in inherently three-dimensional information worlds, whether designed for work applications (e.g., Card et al. 2004) or games (e.g., Lewis & Jacobson, 2002). We focus on the case of a user navigating a *document*, a flat surface displaying text or (still or animated) graphics. Such documents are pervasive in current computer applications such as text editors, spreadsheets, atlas viewers, photo editors, etc. In fact, as in Fitts' original paradigm, we will often reduce our documents to a single dimension.

2.3. Fitts' Paradigm in Graphical User Interfaces

2.3.1. Mapping the Hand and the Document onto the Screen

In GUIs the target-acquisition problem involves four basic elements and their inter-relations. The elements are the user's hand (typically attached to, and, in this sense, confounded with some input device), the cursor, the view, and the document.

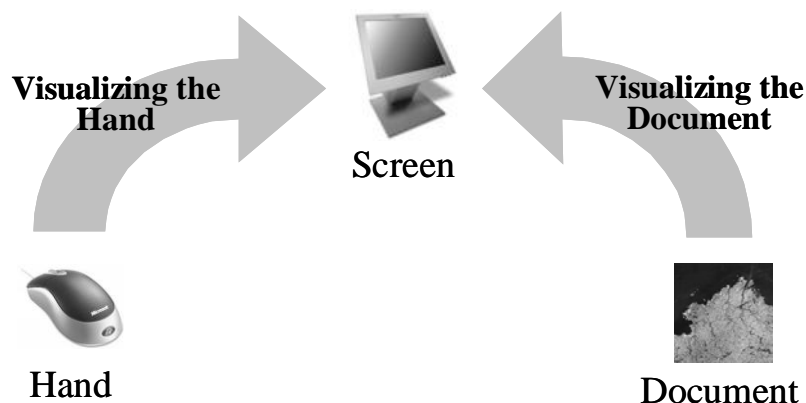


Figure 1. The double mapping scheme of graphical user interfaces.

As illustrated in Figure 1, pointing in a GUI is *indirect*: the hand, which designates, belongs to a different world than the document, which contains the objects to be designated. An encounter can only take place in screen space, which requires a bilateral mapping scheme. First, the user's hand is represented on the screen by a cursor. Hand motion is translated into cursor motion through a display-control mapping function. The slope of this function determines the scale of display space relative to motor space.

Second, the document is represented on the screen through a view. The view displays a part of the document called the selection, using a viewing function. Figure 2 illustrates this mapping between the view and the document in the one-dimensional case. Both the documents (on the x axis) and the screen (on the y axis), appear as intervals. The slope of the viewing function defines the visualization scale, while its intercept defines the offset. View pointing (see below) consists in controlling these two parameters: varying the slope changes the viewing scale and is called *zooming*, while varying the intercept changes the offset and is called *scrolling* or *panning*.²

² The term *scrolling* was introduced in HCI along with the classic WIMP interface. The term *panning* appeared more recently, coupled with the *zooming* facility in multiscale interfaces. Although their connotations differ, we need not distinguish the two terms here.

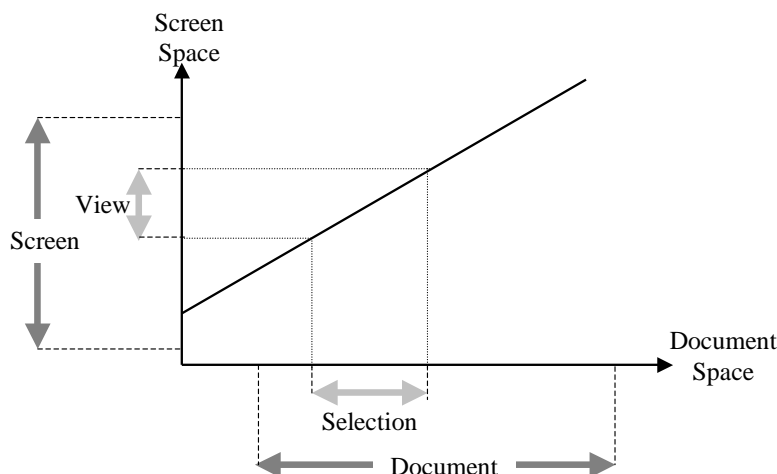


Figure 2. One-dimensional representation of document mapping onto the screen.

Owing to the exponential progression of the storage and processing capacity of computers over the past 20 years, the size of the electronic documents that users commonly handle has increased by more than a thousand-fold, whereas the size of our screens has hardly doubled (Beaudouin-Lafon, 2004). As a consequence, selection size, in percentage of document size, has become smaller and smaller (Guiard et al., 2004). It has gradually become impossible to explore a document by just scrolling—hence the crucial importance of scale manipulation tools such as zooming (Perlin & Fox, 1993, Furnas & Bederson, 1995).

2.3.2. *Cursor Pointing vs. View Pointing*

However reductive, Fitts' target-acquisition paradigm captures something quite general. In particular the concept of a target is sufficiently generic to subsume the main two categories of target-directed movements that can be observed in HCI, namely, *cursor pointing* and *view pointing* (Guiard et al., 2004, Gonzalez et al., in press). In traditional cursor pointing, the target, such as an icon, a menu item, a hypertext link, or a button is visible in the current view, so the user can acquire it by bringing the screen cursor to it. If the target is located out of sight however, the user first needs to adjust the view so that the target object becomes visible. We call this *view pointing*, since it consists of moving the current view to a target view that includes the target object. Once the target object is visible, the user can proceed to cursor pointing. In many cases, however, view pointing is not followed by cursor pointing. For example, when reading a document on-line, one might use view pointing to navigate to the references section, without selecting a target in the destination view. Such a view pointing task, which requires scrolling, seems very different from a cursor pointing task, such as reaching a menu item with one's cursor. In terms of Fitts' paradigm, however, they are identical: in both cases the user must select a subset of the document.

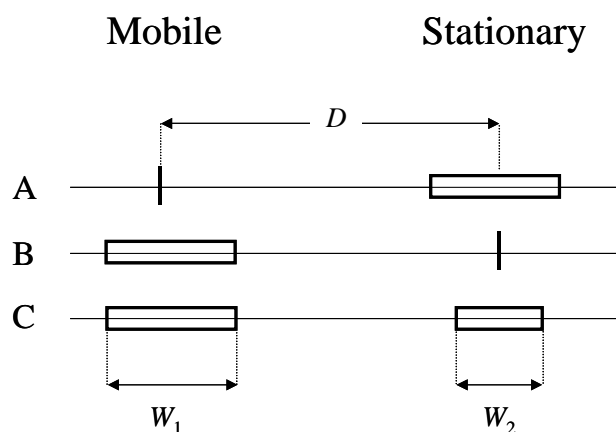


Figure 3. Cursor pointing (A), ‘Prince’ pointing (B), and view pointing (C).

Figure 3 illustrates, in 1D space, the three possible cases for a pointing task. In case A, a point cursor must be moved to reach a target, specified by an interval (an area in 2D space). In case B, symmetrical to A, an interval cursor (an area cursor in 2D space) must be moved so as to eventually include a point target (Kabbash and Buxton, 1995). Cases A and B differ only in their arbitrary frames of reference (Mottet et al., 2001). W is a measure of the target in one case and of the area cursor in the other, but the computation of the ID is identical. Case C, which seems not to have been explicitly considered in the literature, generalizes cases A and B: a mobile interval (or area cursor in 2D) must be moved so as to overlap the target interval (or target area in 2D). View pointing falls into case C, where the mobile interval is the view that must be moved so that the target enters it and becomes visible.

The quantification of task difficulty is just as straightforward in case C as it is in the other two cases. Target distance D is the distance separating the centers of the two intervals. Target width W can be quantified in one of two ways, depending on the goal pursued in pointing. If the user only wants to select an object, then an overlap will suffice, in which case W must be computed as the sum of the width of the mobile interval W_1 and the width of the stationary interval W_2 :

$$ID = \log_2 \left(\frac{D}{W_1 + W_2} + 1 \right) \quad (1)$$

If, however, the user wants to reach some target region to see it, as is often the case with view pointing, then the larger interval must include the smaller interval, and W becomes the absolute-value difference between W_1 and W_2 :

$$ID = \log_2 \left(\frac{D}{|W_1 - W_2|} + 1 \right) \quad (2)$$

Either way, Fitts’ law can be assessed in view pointing just as easily as it has been in ordinary cursor pointing. The concept of view pointing, associated with Equations 1 and 2, shows that Fitts’ law can serve not only to model classic pointing with a cursor, but also document navigation—obviously a broad class of user activities in HCI.

3. MODELING POINTING IN MULTISCALE ELECTRONIC WORLDS

Let us now consider the task of pointing to a target in a multiscale information space. First we introduce Furnas and Bederson’s (1995) space-scale diagrams, which help us to understand multiscale documents and multiscale navigation. We then present several multiscale pointing techniques.

3.1. Space-Scale Diagrams

Let us consider the 2D document in Figure 4 (left), with its origin at the intersection of the x and y axes, and let us represent it at several levels of magnification. The representation at scale 1 corresponds to the original document. The representation at scale s , noted R_s , corresponds to the original document magnified by a factor s , that is, a point at position (x, y) in the original document is at position $(s.x, s.y)$ in the R_s .

Now let us stack these multiple representations along a vertical axis, as in Figure 4 (right), so that each scaled view R_s intersects the vertical axis at s , and the vertical axis goes through the origin of each scaled representation. A *space-scale diagram* (Furnas and Bederson, 1995) is a conceptual extension of this stacking where every possible scaled representation of the document is included (in actual practice, of course, just a few will do). A space-scale diagram is therefore a 3D representation of a 2D document, with two dimensions of space and one dimension of scale. The main property of a space-scale diagram is that a point (x, y) of the original document is represented by a great ray—i.e., a semi-line going through the origin and the point $(x, y, 1)$. All the points on the semi-line are of the form $(s.x, s.y, s)$. This property can be captured by the following *scaling invariant*: distance d at scale s is identical to distance d' at scale s' if and only if

$$d / s = d' / s' \quad (3)$$

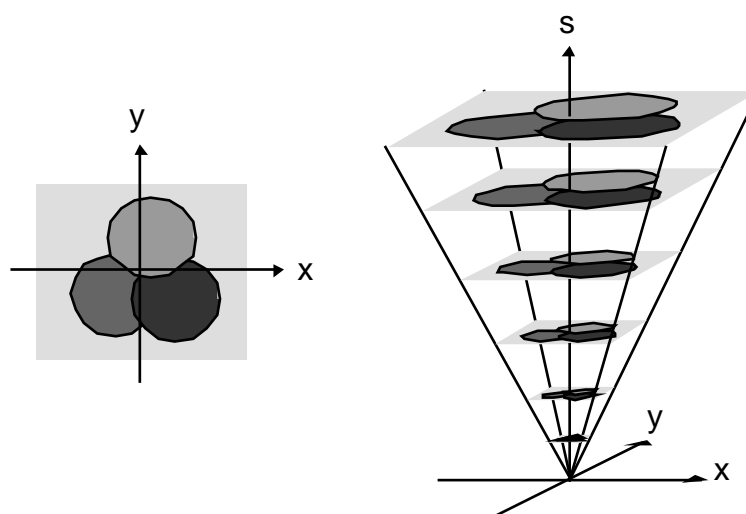


Figure 4. Furnas et Bederson's (1995) space-scale diagram. A document (left) is represented at several levels of magnification (right). Axes x and y define space, axis s defines scale.

To describe and understand navigation in a multiscale document, a space-scale diagram is more efficient than the purely spatial representation of Figure 2 because it contains many possible scaled representations in a single chart. A view of a multiscale document is specified by a (usually) rectangular window (Figure 5, left), together with the scale at which the document is viewed. This scale corresponds to the plane that contains the view in the diagram (Figure 5, middle). *Panning* consists of moving the view within its plane, i.e. at constant scale. *Zooming* consists of moving the view up and down along the scale dimension (Figure 5, right): zooming-in magnifies the view by moving up the scale dimension, thus revealing details, while zooming-out minifies the view by moving down the scale dimension, thus revealing context. Note that the size of the view does not change while zooming: at a larger scale, the view will show more detail, but this will be at the cost of a reduced selection of the document.

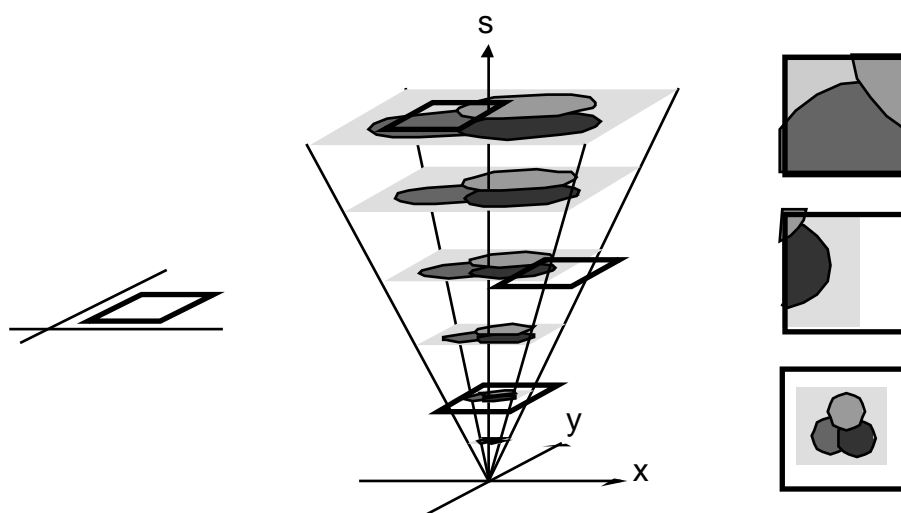


Figure 5. The thick rectangle (left) represents the current view. View size is constant across all magnification levels (middle). The same view at increasing scales shows more detail at the cost of less context (right).

So far we have assumed that the content of the document is scale-independent. This is unrealistic if we consider large ranges of scale. Consider geographical maps: depending on their scale, they display varying levels of information. For example, a city will appear as a dot and city name on a map of the whole country while the streets will be visible on a map of the city. Objects in a multiscale document can be associated with a scale-range, so as to be displayed only when the scale of the viewing window is within that range (Figure 6, left). Some zoomable user interfaces, such as Pad (Perlin & Fox, 1993), Pad++ (Bederson & Hollan, 1994) or Jazz (Bederson et al., 2000) automatically define the scale-range of each object according to its size: when the view size of the object is smaller than a few pixels or bigger than the view, it is faded out.

Scale ranges can also be used to implement *semantic zooming* (Perlin & Fox, 1993): according to the viewing scale, a different object is shown (Figure 6, middle), e.g. a dot and city name at small scale, and a street-map at higher scale. We use semantic zooming below to represent the target of a pointing task as a fixed-size *beacon* at low scales, i.e. when the target is smaller than a few pixels. The beacon is an object whose representation is independent of the viewing scale (Figure 6, right).

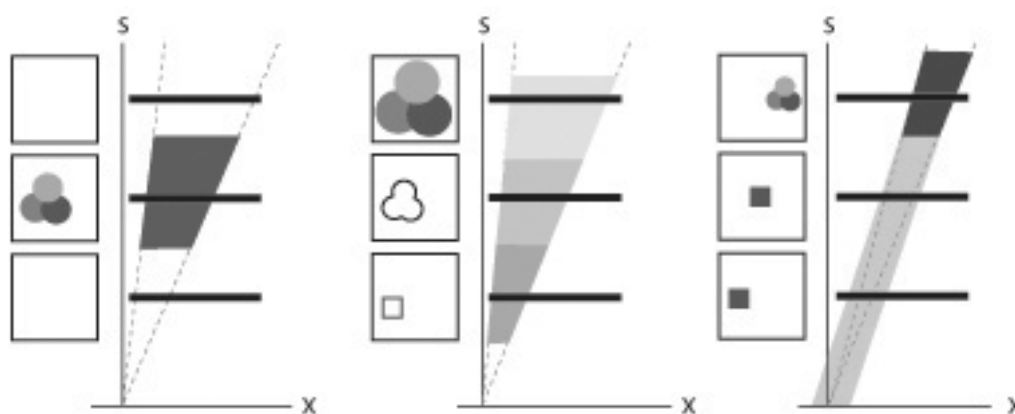


Figure 6. Object with scale-range (left): the object is only visible when the view is within its scale-range; Semantic zooming (middle): the object has different representations at different scales (simple square at low scale, simplified outline at medium scale, complete detail at high scale) ; Scale-independent beacon (right): below a certain scale, the object is always visible and has a fixed display size.

3.2. Modeling Multiscale Pointing

Pointing at a target in a multiscale document can be modeled with the same variables as with a regular document: target distance D and target size W . As with traditional Fitts' pointing tasks, we can reduce the problem to a single dimension, and therefore use space-scale diagrams with one dimension of space (horizontal) and one dimension of scale (vertical). Figure 7 shows the three points that define a pointing task in a space-scale diagram. The starting point (left) and the edges of the target (right) are represented by three great rays. If we consider the pointing task at different scales (scales a , b and c in Figure 7), we observe that the index of difficulty is scale independent since the ratio D/W is invariant. Therefore one can vary the scale of the task for a fixed ID and study the effect of scale on pointing performance (Guiard, 2001; Accot & Zhai, 2001).

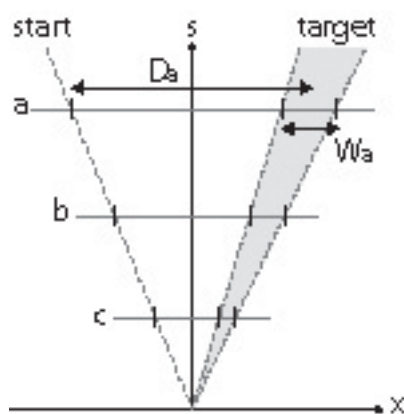


Figure 7. A Fitts task shown at three different levels of scale (a , b and c), using the 1D version of Furnas and Bederson's space-scale diagram. Whatever the scale, the ratio of target distance to target width (D_a/W_a , D_b/W_b , D_c/W_c) is constant. D_b , W_b , D_c and W_c are not shown for readability.

At very low scale, the displayed amplitude becomes so small, say less than a pixel, so as to render the task impossible. At a very large scale, the viewing window cannot include both the starting point and the target, and therefore the user has to use view navigation in order to acquire the target. Finally, even if the view includes both the starting point and the target, it may be impossible to select the target if it is too small relative to the view, e.g. smaller than a pixel. This occurs for high values of the D/W ratio, i.e. for large ID s. In summary, pointing a target in a multiscale document generally requires view navigation and is therefore qualitatively different from traditional pointing. For this reason, we call it *multiscale pointing*.

The question then is whether or not multiscale pointing is subject to Fitts' law. As we have seen above, multiscale pointing can be defined with the same independent variables as traditional pointing: target distance D and target width W . Space-scale diagrams suggest a related set of independent variables to study multiscale pointing: scale and difficulty. The scale of the task corresponds to its *absolute* amplitude D , while its difficulty corresponds to its *relative* amplitude D/W . As shown in Figure 7 above, we can vary the scale of a task without varying its difficulty. Figure 8 below shows that, conversely, we can vary the task difficulty

without varying its scale. If multiscale pointing follows Fitts' law, we expect movement time to vary with task difficulty but not with task scale.

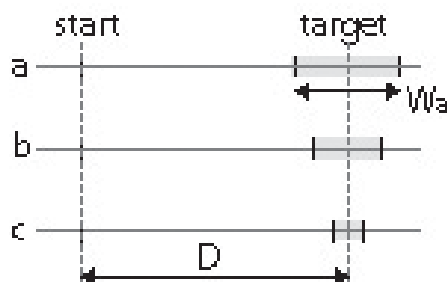


Figure 8. Variation of task difficulty at a constant level of scale. Movement amplitude (D) is constant. Relative movement amplitude (D/W_a , D/W_b , D/W_c) and hence the ID increases from a to b to c . W_b , and W_c are not shown for readability. This is not a space-scale diagram; if it were, the three cases would appear on a single horizontal line).

3.3. Extant Techniques for Multiscale Pointing

Several techniques have been introduced to help users reach their targets in large multiscale documents (see Leung and Apperley, 1994, for a review). We describe three, the bi-focal, fish-eye and pan-and-zoom techniques, using space-scale diagrams, and briefly discuss their advantages and drawbacks in relation to the target-acquisition problem. Note that, in principle, the various techniques can be combined.

3.3.1. The Bi-Focal (Context-plus-Detail) Technique

A bi-focal display (Apperly and Spence, 1982) is a display that simultaneously offers two views: a small-scale view, which provides context by showing all or most of the document, with little detail, and a large-scale view, which shows the details of a small focus area.

The two views can be either tiled or overlaid. Tiling may make it difficult to relate a detailed view to its general context. With overlaying, however, the detailed view typically hides part of the context view: DragMag (Ware & Lewis, 1995), for example, addresses this problem by drawing a link with thin outlines and allowing the user to freely move either view relative to the other (see Figure 9).



Figure 9. Bi-focal display of a text document.

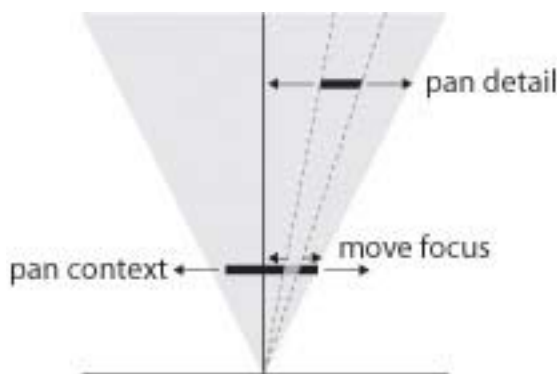


Figure 10. Space-scale representation of the bi-focal visualization technique with view navigation operations: panning the context view, moving the focus area, and panning the detail view. The representation being reduced to a single dimension, the document appears as a gray angular sector.

Figure 10 shows a bi-focal display, represented in a space-scale diagram with two view windows. Each view is constrained to move within its own plane. Three basic operations are possible: panning the context view, moving the focus area and panning the detail view. Panning the context view is used to navigate quickly towards the target area. In general, the focus area stays fixed relative to the context view, so the content of the detail view changes as well. Moving the focus area is used to position the detail view coarsely around the target. It changes the content of the detail view alone and is achieved by moving the outline of the detail view within the context view. Finally, panning the detail view is used to fine-tune the position of the focus area. Moving the focus area and panning the detail view have the same effect (changing the detail view) but are controlled at different scales.

The main limitation of bifocal displays is that the relative scale of the two views cannot be too large, otherwise view navigation becomes cumbersome. They are thus appropriate only for moderately large documents unless used in combination with another multi-scale technique.

3.3.2. The Fish-eye Technique

Fish-eye views (Furnas, 1986) use optical distortion to include both detail and context within a single view. Unlike the bi-focal display, there is a continuous transition between context and detail. Several distortion modes are possible, including the spherical (Figure 11, left) and the linear distortion (Figure 11, right) (see Carpendale and Montagnese, 2001, for a larger list). With a fish-eye view, no parts of the document are hidden by the overlay of multiple views. However, the distortion tends to make it difficult to interpret the content of the view, especially when text is displayed. In practice, the distortion limits the scale of the focus area relative to the scale of the overall view, and so this technique suffers, probably to a greater extent, the same kind of limitation as the bi-focal technique.

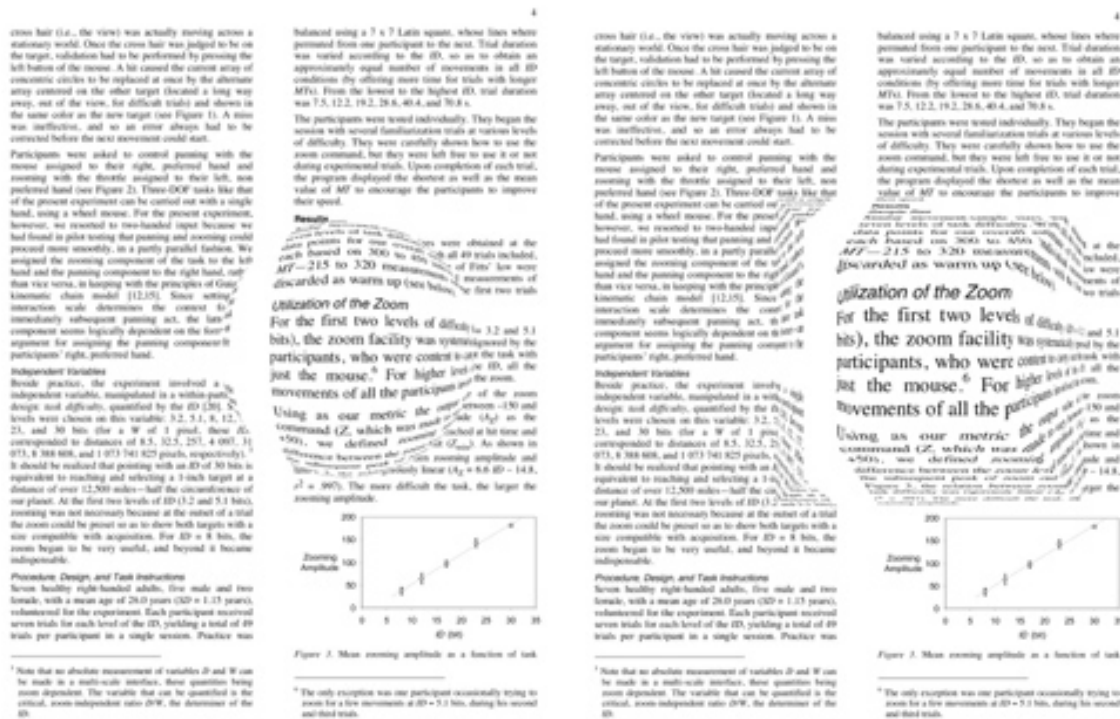


Figure 11. The fish-eye technique with spherical (left) and linear (right) distortion.

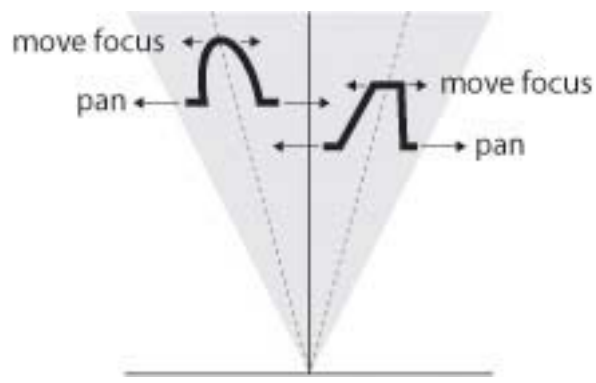


Figure 12. Space-scale representation of the fish-eye visualization technique with navigation operations. Left: spherical; right: linear.

Since the view contains representations of different parts of the document at different scales, its representation in a space-scale diagram is no longer a horizontal line segment (Figure 12). Reaching targets with a fish-eye view involves both panning the view and moving the focus point of the distorted area. As shown by Gutwin (2002), the latter hinders performance because objects move as they are focus-approached. Some prototypes, however, fix the focus point at the center of the view, so the user can pan only the view.

3.3.3. The Pan-and-Zoom Technique

Pan-and-zoom uses a single view which shows the document at a uniform scale (Perlin and Fox, 1993). Panning is used to move the view at a fixed level of scale, while zooming is used to change the current scale of the view. Zooming occurs around a focus point which is left invariant by the scale change. The focus point is generally coincident with the center of the view, more rarely with the cursor position. In a space-scale diagram, zooming amounts to moving the view along the great ray that goes through the focus point (Figure 13).

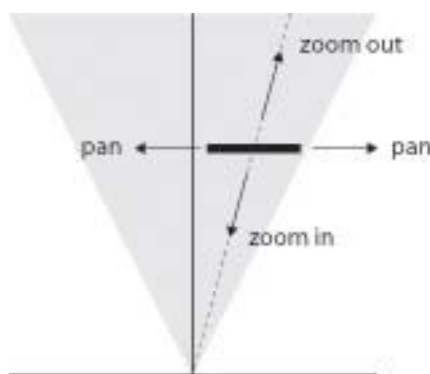


Figure 13. Space-scale representation of the pan-and-zoom visualization technique with navigation operations.

Zooming appears continuous on the screen, but is actually achieved by multiplying the current scale by a fixed zoom factor, f , when zooming-in, or, inversely, dividing by f when zooming-out. The scale values that are navigated by zooming thus follow a geometric, rather than arithmetic, series, i.e. a power law.

In order to measure distances on the scale dimension, we introduce the *zoom index*, $z = \log s$, where s is the scale. Navigating the scale dimension increases or decreases the zoom index by a constant factor $z_f = \log f$:

$$\text{zoom-in: } z' = \log s' = \log s \cdot f = \log s + \log f = z + z_f$$

$$\text{zoom-out: } z' = \log s' = \log s / f = \log s - \log f = z - z_f$$

We normalize the zoom-index dimension by using the logarithm in base f and therefore $z_f = 1$. Controlling the zoom index rather than the scale is the key to efficiently navigating a large range of scales. Typically, a zoom factor of 1.1 allows continuous zooming. With this zoom factor, the zoom-index range $[-50, 50]$ corresponds to a scale range of approximately $[1/100, 100]$, i.e. a ratio of 10,000 between the smallest and largest scale, while the zoom-index range $[-100, 100]$ corresponds to a scale range of approximately $[1/15,000, 15,000]$, that is, a ratio of 225 million.

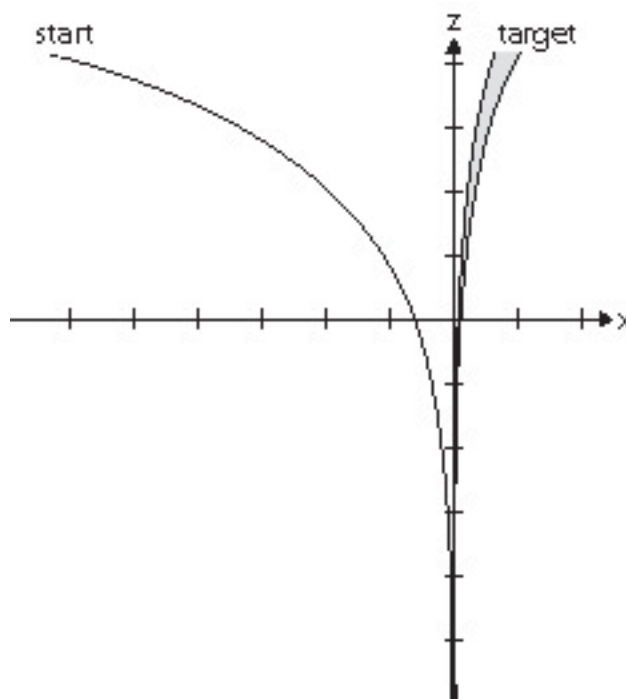


Figure 14. Space-scale diagram in zoom-index space, showing the starting position and target of a pointing task, as in Figure 7. Zoom-index value 0 (horizontal axis), corresponds to scale 1.

If we replace the vertical scale dimension by a zoom-index dimension in the space-scale diagram, the great rays become exponential curves (Figure 14). Such a diagram is more true to the user experience of navigating a multiscale world with zooming and panning than the traditional one: constant-speed panning corresponds to constant-speed horizontal motion of the view (as in a traditional space-scale diagram), and constant-speed zooming corresponds to constant-speed vertical motion of the view (unlike a traditional space-scale diagram). Notice that, since the space dimension has not been changed, the viewing window still has a constant size across the diagram.

Target acquisition with a pan-and-zoom interface typically involves three consecutive phases of action: (1) zooming-out, (2) zooming-in-and-panning, and (3) pointing (Figure 15). In the first phase, the user zooms-out until the target area is visible in the view window; in the second phase, the user combines zooming-in and panning until the target is large enough within the view to be selected; in the third phase, the user selects the target with a traditional pointing action. Either or both of the first two phases may be unnecessary: phase one is unnecessary if the target is already visible, while phase two is unnecessary if the target is big enough.

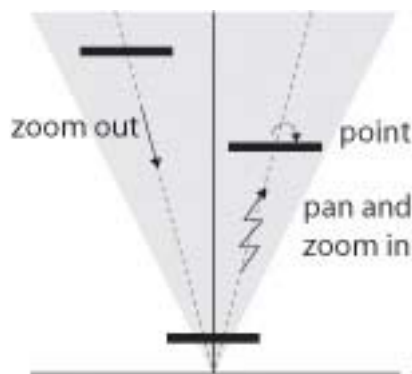


Figure 15. A typical sequence of action with the pan-and-zoom technique: zooming-out to visualize the target beacon, panning and zooming-in to reach the target proper, finally cursor-pointing to it.

4. FITTS' LAW IN MULTISCALE POINTING: A THEORETICAL ANALYSIS

The navigation techniques introduced in the previous section have been mostly evaluated in terms of their usability for high-level tasks. For example, Hornbæk and Frøkjær (2001) have compared the usability of linear, fish-eye and bi-focal displays for reading documents and Baudisch et al. (2002) have compared focus-plus-context, fisheye and pan-and-zoom interfaces for tasks such as verifying connections on a circuit board. Gutwin (2002) conducted an experiment on target-reaching with a fish-eye display using the ISO 9241-9 point-select task (Soukoreff and Mackenzie, this issue), although they did not systematically control target distance and target size. Guiard et al. (1999) conducted an experiment with a bi-focal display with Fitts' (1954) reciprocal pointing protocol, but the user controlled the cursor at a single scale. Assessing Fitts' law in multiscale electronic worlds is still an open research question.

This section presents the theoretical analysis that structured our experimental work on Fitts' law in multiscale pointing. As described above, multiscale pointing consists of three phases and therefore movement time (MT) corresponds to the sum of the durations of these phases. Phase 2, zooming-in-and-panning, is not only the phase whose completion takes most time, it is also the most complex of the three. While phase 1 involves just scale and phase 3 just space, phase 2 demands an intricate coordination of action in both space and scale. This is why we focus on phase 2 below.

4.1. Multiscale View Pointing

Consider a view-pointing task defined by a view size V , a distance D to the target and a target size W (D and W are specified in document space, e.g. at scale 1; recall that V is scale independent). We are interested in the case where target distance is very large compared with target size—that is, in tasks with a high index of difficulty $ID = \log_2(D/W + 1)$.

To start with (Phase 1), one must zoom-out until the target (in fact, the beacon that represents it) enters the view. Let us call s_{in} the scale at which this happens (Figure 16). Once the target is within the view, the user can start panning and zooming-in (Phase 2) towards the target. Navigation ends when the target is big enough within the view so as to be pointed at. Since we are interested in view pointing rather than the final act of cursor pointing, let us consider the limiting case where navigation ends when the target completely fills the view, calling s_{out} the scale at which this happens (Figure 16). On the scale dimension, the user has navigated from s_{in} to s_{out} (again, we ignore the zoom-out phase to concentrate on zoom-in-and-pan). In

zoom-index space, which is the only space where distances in scale can be computed, the zoom-index distance (ZD) is $\log s_{out} - \log s_{in}$.

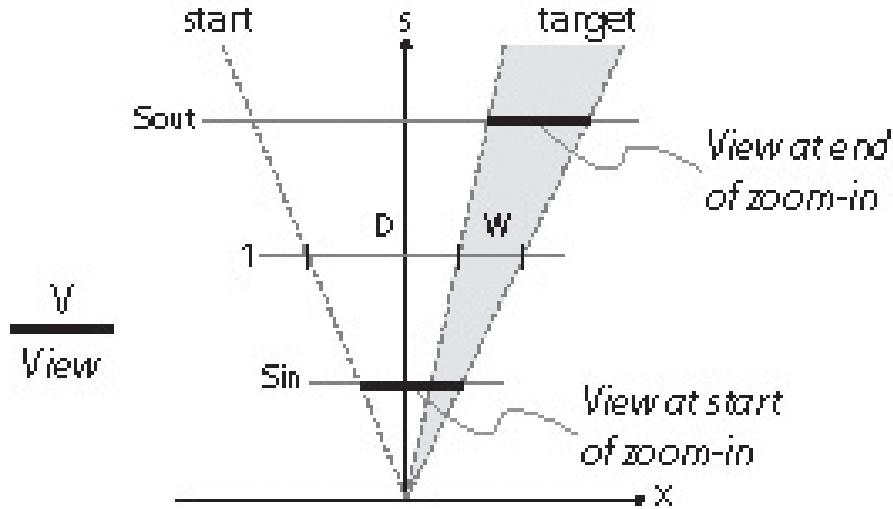


Figure 16. Multi-scale pointing towards a target. D and W are measured at scale 1. S_{in} and S_{out} correspond to the start and end of the zoom-in phase.

ZD is computed using the scaling invariant (equation 3). We first compute s_{in} and s_{out} . At scale s_{in} , the view (size V) covers the distance D to the target, and the target itself (size W), so we have:

$$(D + W) / 1 = V / s_{in} \quad \text{or: } s_{in} = V / (D + W)$$

At scale s_{out} , the view (size V) covers the target (size W) so we have:

$$W / 1 = V / s_{out} \quad \text{or: } s_{out} = V / W$$

We now compute ZD :

$$ZD = \log s_{out} - \log s_{in} = \log V/W - \log V/(D+W) = \log (D+W)/W = \log (1 + D/W) = ID$$

Therefore the distance to navigate in zoom-index space is exactly the index of difficulty (ID) of the task. Note that this is independent of view size V : indeed, a smaller view would require starting at a lower scale, but the task would be completed at a lower scale as well.

4.2. From the Steering law to Fitts' law

The view pointing task described above is reminiscent of a steering task (Accot & Zhai, 1997): from the moment the target is in the view, zooming-in is similar to driving a vehicle through a tunnel. The driver uses the accelerator to proceed through the tunnel and turns the wheel to correct any current aiming error. Here, the user zooms-in toward the target and pans to correct any aiming error. If the target moves out of view (the user is off the road), the user must zoom-out (back up) until the target gets back into view and then zoom-in again. For successful navigation, the target must stay within the view while zooming-in proceeds, just as the car must stay inside the tunnel while traversing it. Therefore we can use Accot and Zhai's (1997) steering law for tunnels to predict navigation time:

$$MT = k d / w$$

where d is the length of the tunnel and w its width. In our case, d is the distance navigated along the scale dimension, which we have shown to be equal to the index of difficulty ID , and w is the target size of the view, which we call V . Therefore we have:

$$MT = (k / V) ID \quad (4)$$

Since V is fixed for a given navigation, this shows that multiscale view pointing follows Fitts' law. As the motor control involved in multiscale navigation is quite different from that of a traditional pointing task, this is a nontrivial finding. Fitts' law is known to hold for movements in which speed goes up, then down, without a plateau. However in our case, zooming-in requires continuous control where panning movements correct the error revealed by the increased scale. When starting the task, the user does not know the size of the target, just its approximate position, as marked by the scale-invariant beacon that represents the target at small scales. The user 'cruises' through the tunnel at a fairly constant speed determined by her/his ability to process information, until the real target replaces the beacon and fills the view.

The above derivation offers a theoretical basis for predicting that multiscale pointing follows Fitts' law—in fact a simplified version of the law with a zero intercept. In addition it delivers a new prediction, namely that navigation time must be inversely proportional to view size. Inverse proportionality is a highly non-linear relationship. Equation 4 says that gradually decreasing view size should lengthen MT in an accelerated way, thus suggesting that V is a critical factor that must be taken into account in modeling multiscale navigation. Equation 4 has a problem, however: it implies that as V rises toward infinity, MT should tend to 0, obviously an implausible prediction. In the next section we handle this difficulty by enriching the model with a supplementary assumption.

4.3. Limited Human Capacity for Handling Information Flows

Equation 4 can be recast as

$$ID/MT \text{ (bits/s)} = k V, \quad (5)$$

simply noting $1/k$ as k , since the inverse of a constant is a constant. This equation states that the bandwidth of navigation in a zoomable interface is proportional to view size. This proportionality makes sense for small views. However, as view size is scaled up, we must obviously assume some ceiling constraint to reflect the limited capacity of humans for exploiting the information outflow from a computer—for example, in 2D space one cannot reasonably predict that the navigation bandwidth will be doubled if the view is increased from 1m^2 to 2m^2 .

This human-factor limitation can be construed of in terms of a simple fluid-dynamics model where some fluid must flow through two pipes mounted in series, with the first pipe modeling the view offered by computer (V_c) and the second pipe the maximal view a human can exploit (V_h), assumed here to be constant. In this model the flow of liquid that successfully traverses the system is proportional to the cross section of the *smaller* pipe. That is,

$$\text{for } V_c < V_h, \quad ID/MT \text{ (bit/s)} = k V \quad (6)$$

$$\text{for } V_c \geq V_h, \quad ID/MT \text{ (bit/s)} = \text{constant} \quad (7)$$

5. EXPERIMENTAL RESULTS

We conducted two experimental studies on target acquisition in a pan-and-zoom interface to test the predictions of the above model. Experiment 1 (Guiard et al., 2001) tested the prediction that multiscale pointing should obey Fitts' law with a zero intercept (Equation 4). Experiment 2 (Guiard et al., 2004) tested the predictions that for small view sizes, pointing time should be inversely proportional to view size (Equation 6) and that for larger view sizes,

pointing time should become independent of view size (Equation 7). All three experiments used the same basic experimental protocol, which we first describe.

5.1. Methods

5.1.1. The Document

We had to circumvent a common problem of zooming interfaces, known as *desert fog*: one gets lost if one zooms-in an empty region of a multiscale document (Jul & Furnas, 1998). Since we were interested in the basic sensory-motor processes involved in target reaching, rather than the higher cognitive processes of orientation, we used a document designed to preclude disorientation. We used an electronic map containing nothing but an infinite set of equidistant concentric circles centered around the target. Zooming-in caused a radial expansion, and zooming-out a radial shrinking of the current document selection around the screen center, but the number of visible arcs displayed in the document was held roughly constant due to a self-similar fractal generation mechanism. When the target was out of sight (as in Figure 17), our participants always knew in what direction and how far they had to move their view, since target direction was specified by the perpendiculars to the tangents to the visible arcs and target distance by arc curvature, the inverse of the radius D .

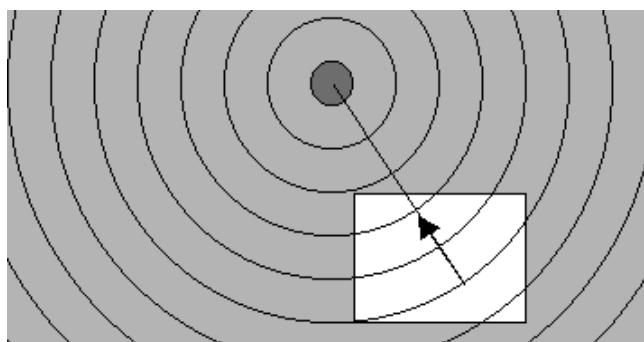


Figure 17. A schematic illustration of the array of concentric circles with the target located out of sight. The view, shown as an empty rectangle, intercepts four arcs which specify target direction and distance. The radius and the arrow, not shown in the experimental display, appear here for explanatory reasons.

We had recourse to semantic zooming (Perlin & Fox, 1993; Furnas & Bederson, 1995). As long as the magnification was insufficient to allow target representation, a constant-size (i.e., zoom-insensitive) beacon represented the target. As soon as, thanks to zooming-in, the target could be displayed with more than one screen pixel, it replaced the beacon, the document then revealing a colored disc that was zoomable and could be clicked.

5.1.2. Input device

Zooming was controlled with the left, non-preferred hand by rotating a game-controller throttle. Panning was controlled with the right, preferred hand by operating a mouse (in Experiment 1) or a Wacom stylus set to the relative mode (in Experiment 2). For both zooming and panning, the control was in zero-order mode (i.e., angles were directly converted into scale levels, and positions into positions, respectively).

There was no cursor. A stationary crosshair permanently shown at the center of the view served for aiming, as in shooting games. Moving the panning device with the right hand

resulted in the screen image moving in the opposite direction, giving the impression that the view was actually moving over a stationary document.

5.1.3. Task

We adapted Fitts' (1954) reciprocal tapping task for the purpose of this study. Participants had to record as many hits as possible for a variable, pre-determined period of time (a trial) by clicking alternatively on two colored discs. Once the crosshair was judged to be on one target, validation had to be performed by clicking. A hit caused the current array of concentric circles to be replaced at once by the alternate array centered on the other target (typically located a long way away, out of sight). A miss was ineffective, and so an error always had to be corrected before the next movement could start— therefore error rate was always 0%.

5.2. Experiment 1: Fitts' Law Beyond the 10-Bit Barrier

In order to test the validity of Fitts' law in multiscale, pan-and-zoom pointing, we focused in this experiment on tasks of very high difficulty. Fitts' law experiments usually have *ID*s in the 2-10 bit range (Soukoreff & MacKenzie, this issue). We presented our participants with a selection of *ID*s ranging from 3.2 bits ($D/W = 8$) up to 30 bits ($D/W = 10^9$).

5.2.1. Procedure

The critical independent variable, manipulated in a within-participant design, was *task difficulty*, quantified as $ID = \log_2(D/W + 1)$ (MacKenzie, 1992). We chose seven levels: 3.2, 5.1, 8, 12, 17, 23, and 30 bits (for a W of 1 pixel, these *ID*s corresponded to distances of 8, 32, 257, 4 097, 31 073, 8 388 608, and 1 073 741 825 pixels, respectively). Note that pointing with an *ID* of 30 bits is equivalent to reaching and selecting a 1-inch target at a distance of about half the circumference of our planet. At the first two levels of *ID*, zooming was not necessary because at the outset of a trial the zoom could be preset so as to show both targets on the screen with a size compatible with acquisition. For $ID = 8$ bits, the zoom began to be very useful, and beyond this level of difficulty it became indispensable.

5.2.2. Results and Discussion

Zoom Utilization

For the first two levels of difficulty ($ID = 3.2$ and 5.1 bits), the zoom facility was systematically ignored by the participants, who were content to carry out the task with just panning. For higher levels of the *ID*, all the movements of all participants involved a zooming component.

Using as our metric the output value of the zoom index (Z , which was made to vary between -150 and $+50$), we defined zooming amplitude (A_Z) as the difference between the zoom index reached at hit time and the zoom index reached at the time of peak zoom-out (Z_{\min}). As visible in Figure 18, which takes into account only the five levels of *ID* where the zoom was used, the relation between zooming amplitude and task difficulty was rigorously linear, with $A_Z = 6.6 ID - 14.8$ ($r^2 = .997$). The more difficult the task, the larger the zooming action.

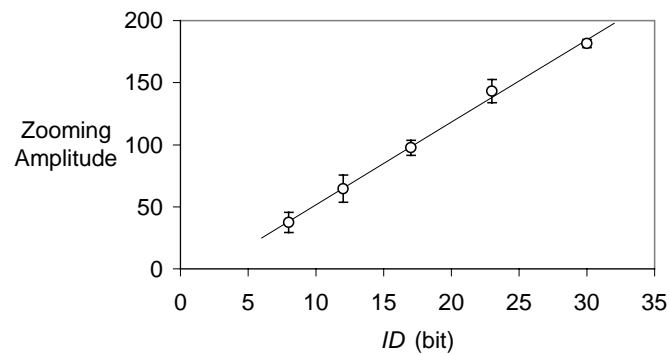


Figure 18. Mean zooming amplitude as a function of task difficulty for $ID = 8$ bits, shown with 95% confidence intervals.

Movement Time

We defined MT as the amount of time elapsed between two consecutive hits. As shown in Figure 19, the mean MT vs. ID curve was almost perfectly modeled by the usual version of Fitts' law, namely, $ID = \log_2(D/W + 1)$ (see Soukoreff & Mackenzie, this issue). A linear regression yielded $MT = 0.32 ID - 0.04$, $r^2 = .995$, with all seven participants displaying a very good linear fit ($.981 < r^2 < .999$). Keeping in mind that we used data points spread over a very wide-ranging selection of ID s, such a fit is remarkable.

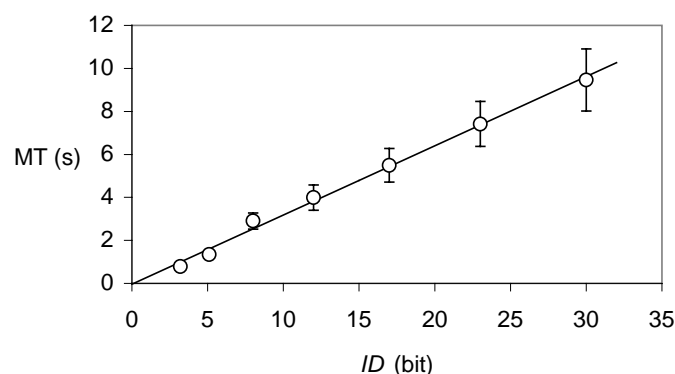


Figure 19. Fitts' law assessed over a selection of ID s ranging far beyond 10 bits, due to the zooming function.

Fitts' law intercept was close to zero (0.04 s on average), consistent with our prediction of the simplest possible version of the law, of the form $MT = k * ID$.

Zooming-Out and Zooming-In Time

We parsed MT into *zooming-out time* (Z_oT), the time separating a successful click from the subsequent peak of zoom-out (Z_{min}) and *zooming-in time* (Z_iT), the time separating a peak of zoom-out from the subsequent hit. Recall that while Z_oT corresponded to a virtually pure zooming time (with little or no panning), Z_iT represented the duration of a composite process made up of a series of intricately alternating pans and zooms, which started almost systematically with a large-scale pan.

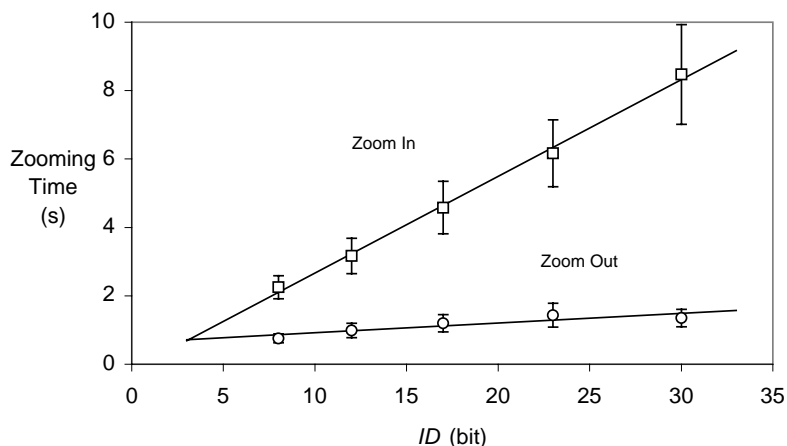


Figure 20. Mean zooming-out and zooming-in time as functions of task difficulty.

As shown in Figure 20, most of the time elapsed between two consecutive hits consisted of Z_iT , with the initial zoom-out phase accounting for a vanishing proportion of MT as task difficulty increased. Both components of MT were linearly dependent on the ID , but with markedly different slopes ($Z_iT = 0.28 ID - 0.16$, $r^2 = .997$, and $Z_oT = 0.03 ID + 0.63$, $r^2 = .809$).

Index of Performance

Fitts (1954) proposed using the ratio ID/MT , referred to as the index of performance (IP), to quantify the rate of information transmission, or channel bandwidth, in the spirit of Shannon's information theory. Fitts' first hypothesis was that the IP (dimensionally, in bit/s) should be invariant along the continuum of task difficulty, but his own data showed the IP to vary with task difficulty, as has been usually found in Fitts' law research ever since (Soukoreff & MacKenzie, this volume; Zhai, this issue).

Consistent with previous observations, the IP declined from the first to the second level of ID , in the lower region of the difficulty continuum where the participants were better off without the zoom (see Figure 21). This effect, evident in each individual participant, was highly significant. For the higher five levels of difficulty, which required zooming, no more decrease in the IP was obtained. On the contrary, we observed a slight but significant improvement of the IP as pointing became more difficult ($F(4, 24) = 10.38$, $p = .00005$). However, this effect was essentially due to the IP value at 8 bits being located lower than the other four—an effect that presumably reflects the sub-optimal size of the screen we used. So, focusing on the 12- to 30-bit range, it seems fair to conclude that in this experiment the processing rate, or throughput for multiscale pointing was essentially *insensitive* to task difficulty, in keeping with Fitts' (1954) initial expectation and our model.

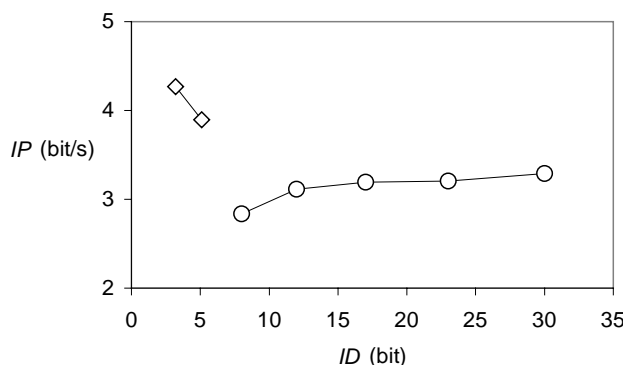


Figure 21. Index of performance as a function of task difficulty.

5.2.3. Summary

To sum up, the present exploratory study confirmed our prediction that Fitts' law can be investigated, with the help of zooming, far beyond the classic 10-bit barrier, without altering the essence of Fitts' aimed-movement paradigm. Zooming started to be useful at 8 bits; beyond that threshold zooming amplitude was linearly dependent on task difficulty. Our MT measurements for multiscale pointing were consistent with the simplest possible version of Fitts' law, $MT = k * ID$, a form which implies a constant information outflow, in keeping with Fitts' (1954) expectation and as predicted by our own model of multiscale target acquisition (see Section 4).

5.3. Experiment 2: Effect of View Size on Bandwidth

We now turn to our experimental tests of Equations 6 and 7 on the effect of view size on movement bandwidth. Whereas in Experiment 1 the view appeared in full-screen mode, in Experiment 2 the view appeared in the middle of a black screen as a rectangular aperture showing the current selection of the document. View center, coincident with screen center, was permanently marked with a colored cross-hair. The task, alternatively reaching and clicking on two targets located a long distance away from each other, was the same as in Experiment 1.

5.3.1. Experiment 2a: Wide Range of View Size

Independent variables and procedure

The experiment involved two independent variables, task difficulty ID and view size V . We used two levels of difficulty: $ID = 14.6$ and 17.9 bits. A 17.9-bit pointing task is roughly equivalent to a 1 cm target at a 2.5 km distance. However, since the only effect of the (initial) ID in our data-reduction strategy is to provide a longer time series for the evaluation of the bandwidth, as can be understood from Figure 22, we ignore this factor.

View shape per se might have influenced navigation performance, so to avoid confounding we resolved to vary view size in 2D, keeping a constant 1.33 aspect ratio. We explored six levels of V , defined as the view's half-diagonal, spread over a large range: 20, 40, 80, 160, 320, and 640 pixels (full screen).

Dependent variable

Since Equation 5 links a variable that has an information flow dimension (bits/s) to view size, we resolved to directly measure that flow, rather than MT , as our central dependent variable. To this end, we resorted to a simple linear regression analysis (Figure 22). The first step was to assess, for each sample of each cursor-position time series, the current ID_t level, noted ID_t and defined as $\log_2(D_t/W+1)$ where D_t , in document space, stands for the remaining distance to the target at time t . Since D , but not W , gradually drops during task progression, so does the ratio D/W , and hence the current value of the ID . Note that, as a result of this data-reduction technique, the ID no longer appears in the experimental design as an independent variable, but this does not mean the ID is kept aside—rather than manipulating difficulty from condition to condition, here we consider the way in which this variable is handled by participants over time.

Figure 22 shows the evolution of ID_t over one representative instance of a target-reaching movement from one participant. The reduction rate of ID_t over time oscillates around a fairly stable value. Therefore a linear regression analysis over the whole movement—save the short initial zoom-out phase, during which ID_t remains constant—suffices to obtain an estimate of the mean slope. We took this slope as our measure of the characteristic information flow, in bit/s, for each individual movement. Note that we always obtained excellent linear fits, with few r squares below .9.

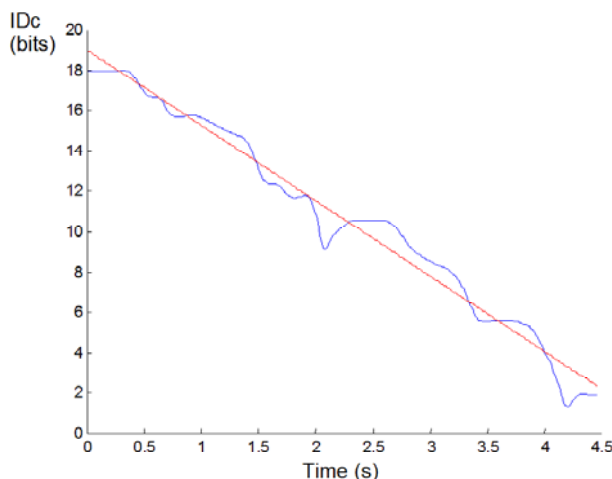


Figure 22. Evolution of the current level of ID over a representative target-reaching movement. The straight line shows the linear equation of best fit (here, $r^2=.96$).

Results and Discussion

Figure 23 shows that the information flow was essentially constant over the broad range of view sizes selected for Experiment 2a, save its extreme lower end. Although the effect of V was globally significant ($F(5,35) = 17.32, p < .05$), only the leftmost data point significantly differed from others. The form of the mean curve shown in Figure 23 displayed surprisingly little between-participant variability. For all nine participants, the bandwidth systematically increased as V was raised from 20 pixel to 40 pixels, whereas none showed any notable variation of bandwidth over the higher range of view sizes. The critical values of V appear to be situated in a lower range of view sizes than expected, certainly below 40 pixels.

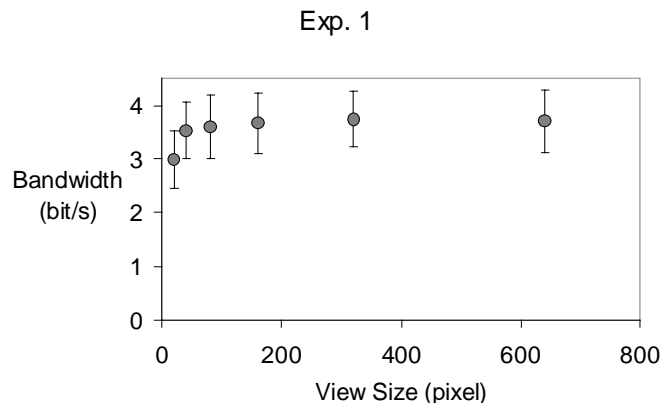


Figure 23. Bandwidth for a broad range of view sizes. Error bars represent $\alpha=0.05$ confidence limits based on between-participant variability. Note that the sign of the y variable (the slope of the ID_t vs. time function) has been inverted.

5.3.2. Experiment 2b: Lower Range of View Size

Using a new sample of volunteers, we replicated Experiment 2a with a lower range of view sizes to check whether the variations of V below 40 pixels does indeed influence bandwidth and, if yes, to evaluate whether or not the proportionality relation stated by Equation 6 holds. The task, the levels of ID (14 and 17 bits), and the design were the same. The only change was that the levels of V were now 5, 7, 10, 15, 20, and 40 pixels.

Results and Discussion

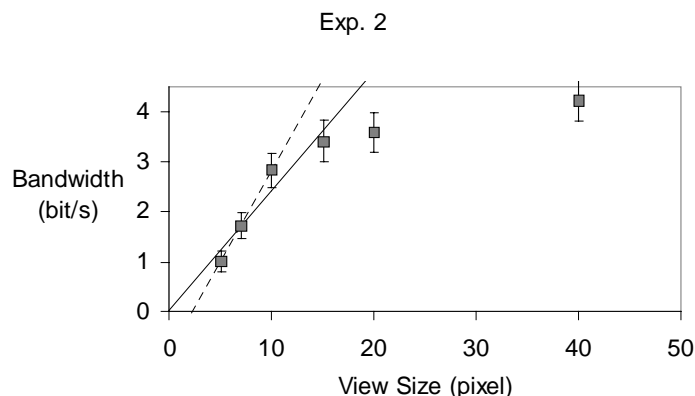


Figure 24. Bandwidth for a lower selection of view sizes.

The results, illustrated in Figure 24, confirmed our suspicions. Not only did the bandwidth decrease dramatically as V was reduced below 20-40 pixels but the shape of the curve was consistent with our hypothesis that the relationship starts as a proportionality. Evaluated over the leftmost three and four points of the plot, the r^2 for fitting a straight line was fairly high in all six participants (mean \pm sd = .988 \pm .009 and .913 \pm .046, respectively) and the intercept small enough to give credit to the proportionality hypothesis (-0.82 \pm 0.32 and 0.03 \pm 0.34, respectively).

Summary

Experiment 2 supports the hypothesis that in multiscale view pointing (1) the bandwidth of the interaction varies proportionally with view size up to a certain critical point and (2)

beyond this critical point, the effect of view size vanishes. This ceiling appeared unexpectedly early in the curve, i.e. when the view's half-diagonal had hardly reached 50 pixels, which corresponds to a view of 80x60 pixels. This finding presumably reflects the non-linear impact of the current panning error on target approach while zooming-in. Since each subsequent magnification step multiplies all visualized distances by a constant zoom factor, the current error increases exponentially, making it risky for the user to tolerate excessive deviations from the beacon. Thus it seems that only the few pixels that surround the target beacon count when controlling the zooming-in phase.

6. CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK

This article extends Fitts' (1954) classic pointing paradigm to include target acquisition in multiscale electronic worlds. This requires introducing the *scale* variable and defining the notion of multiscale pointing. We show that, in the context of zoomable interfaces, multiscale pointing required view navigation. Our theoretical analysis shows that Fitts' law still applies to multiscale pointing, even though the types of movement involved differs markedly from classic pointing. In addition, the derivation explicitly takes into account the size of the view, which had previously been considered constant. Our two studies provide empirical support for the new theoretical model.

The interaction between view size and task difficulty is of particular interest. For small views, the effect of task difficulty, i.e., the slope of Fitts' law, depends on view size: the smaller the view, the steeper the slope, that is, bandwidth is proportional to view size. This has important consequences for the design of user interfaces with small displays, such as PDAs, cell phones and wrist-watch computers. The smaller the display, the greater the need for multiscale navigation but at the same time the lower the navigation bandwidth. Therefore reducing display size risks decreasing the usability of multiscale interfaces. The appearance in our data of a ceiling effect at a rather low level of view size may mitigate this problem, but further tests with additional navigation tasks are necessary to confirm it. This ceiling effect also suggests that display magnification far above the standard size, despite its merits in other respects (see Hascoët, 2003, for an example), should have no noticeable effect on the bandwidth of multiscale view pointing.

The fact that current multiscale pointing techniques have dramatically increased the range of *IDs* that computer users can handle has allowed us to inquire about the validity of Fitts' law over a range of task difficulty far wider than the classic 2-10 bit range: our experiments involved *IDs* up to 30 bits. However, multiscale navigation takes time. In Experiment 1, our participants took up to 10 seconds to reach an extremely difficult target using pan-and-zoom, currently the best multiscale technique. This suggests that further facilitating target acquisition for very difficult cases will require the design of novel interaction techniques. Over the last few years a number of Fitts' law based techniques aimed at artificially facilitating pointing have been described in the HCI literature (see Balakrishnan, this issue). Many of these solutions are effective, but apply only to fixed-scale GUIs. The present theoretical framework should help design new pointing-facilitation techniques for multiscale interfaces and improve our understanding of multiscale navigation.

7. ACKNOWLEDGMENTS

We gratefully acknowledge the generous financial support of the Programme Cognitique of the French Ministry of National Education and Research (Grant # ACI-COG101 to the first author). We warmly thank Shumin Zhai, Dennis Pasveer, and Julien Bastin for stimulating discussions, and Wendy Mackay for carefully reading and helping us improve this report.

8. REFERENCES

- Accot, J., Zhai, S., 1997. Beyond Fitts' law: Models for trajectory-based HCI tasks. Proceedings of ACM Human Factors in Computing Systems (CHI 97, Atlanta, Georgia). ACM Press, New York, pp. 295-302.
- Accot, J., Zhai, S., 2001. Scale effects in steering law tasks. Proceedings of ACM Human Factors in Computing Systems (CHI 2001, Seattle, Washington). ACM Press, New York, pp. 1-8.
- Apperley, M., Spence, I. T. R., 1982. A bifocal display technique for data presentation. In Proceedings of Eurographics, pp. 27-43.
- Balakrishnan, R., this issue. "Beating" Fitts' law: Virtual enhancements for pointing facilitation.
- Baudisch, P., Good, N., Bellotti, V., Schraedley, P., 2002. Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. Proceedings of ACM Human Factors in Computing Systems (CHI 2002, Minneapolis, Minnesota). ACM Press, New York, pp. 259-266
- Beaudouin-Lafon, M., 2004. Designing Interaction, not Interfaces. Proceedings of Advanced Visual Interfaces (AVI'04, Gallipoli, Italy). ACM Press, New York, pp. 15-22.
- Bederson, B.B., Hollan, J.D., 1994. PAD++: A Zooming graphical interface for exploring alternate interface physics. Proceedings of ACM Symposium on User Interface Software and Technology (UIST 94, Marina Del Ray, California). ACM Press, New York, pp. 17-26.
- Bederson, B.B., Meyer, J., Good, L., 2000. Jazz: an extensible zoomable user interface graphics toolkit in Java. Proceedings of ACM Symposium on User Interface Software and Technology (UIST 2000, San Diego, California). ACM Press, New York, pp. 171-180.
- Card, S. K., English, W. K., Burr, B. J., 1978. Evaluation of mouse, rate controlled isometric joystick, step keys and text keys for text selection on a CRT. Ergonomics 21, 601-613.
- Card, S. K., Hong, L., Mackinlay, J. D., Chi, E. H., 2004. 3Book: a 3D electronic smart book. Proceedings of Advanced Visual Interfaces (AVI'04, Gallipoli, Italy). ACM Press, New York, pp. 303-307.
- Carpendale M.S.T., Montagnese, C., 2001. A framework for unifying presentation space. Proceedings of ACM Symposium on User Interface Software and Technology (UIST 2001, Orlando, Florida). ACM Press, New York, pp. 61-70.
- Crossman, E. R. F. W. (1960). The information-capacity of the human motor-system in pursuit tracking. Quarterly Journal of Experimental Psychology 12, 1-16.

- Fitts, P. M., 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 381-391 (reprinted in *J. Exp. Psychol.: General*, September 1992, 121(3), 262-269).
- Furnas, G.W., Bederson, B.B., 1995. Space-scale diagrams: Understanding multiscale interfaces. *Proceedings of ACM Human Factors in Computing Systems (CHI 95, Denver, Colorado)*. ACM Press, New York, pp. 234-241.
- Furnas, G., W, 1986. Generalized fisheye views. *Proceedings of ACM Human Factors in Computing Systems (CHI 86, Boston, Massachusetts)*. ACM Press, New York, pp. 16-23.
- Gonzalez, D., Casalta, D., Mottet, D., Guiard, Y., in press. Le Pointage bi-échelle : Evaluation de quatre variantes d'interfaces graphiques utilisateur. *Le Travail Humain*.
- Guiard, Y., 2001. Disentangling relative from absolute movement amplitude in Fitts' law experiments. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI 2001, Seattle, Washington)*. ACM Press, New York, pp. 315-316.
- Guiard, Y., Beaudouin-Lafon, M., Bastin, J., Pasveer, D., Zhai, S., 2004. View size and pointing difficulty in multiscale navigation. *Proceedings of Advanced Visual Interfaces (AVI'04, Gallipoli, Italy)*. ACM Press, New York, pp. 117-124.
- Guiard, Y., Bourgeois, F., Mottet, D., Beaudouin-Lafon, M., 2001. Beyond the 10-bit barrier : Fitts' law in multiscale electronic worlds. *Proceedings of IHM-HCI 2001*. In: Blandford, A., Vanderdonckt, J., Gray, P. (Eds.), *People and Computers XV - Interactions without frontiers*. Springer, London, pp. 573-587.
- Guiard, Y., Beaudouin-Lafon, M., Mottet, D. (1999). Navigation as multiscale pointing : extending Fitts' model to very high precision tasks. *Proceedings of ACM Human Factors in Computing Systems (CHI 99, Pittsburgh, Pennsylvania)*. ACM Press, New York, pp. 450-457.
- Gutwin, C., 2002. Focus Targeting in Interactive Fisheye Views. *Proceedings of ACM Human Factors in Computing Systems (CHI 2000, Minneapolis, Minnesota)*. ACM Press, New York, pp. 267-274.
- Hascoët, M. (2003). Throwing models for large displays. *Proceedings of 17th Annual Human-Computer Interaction Conference (HCI'2003, Bath, UK), Designing for Society, Vol. 2*, pp. 73-77.
- Hornbæk, K., Frøkjær, E., 2001. Reading of electronic documents: the usability of linear, fisheye and overview+details interfaces. *Proceedings of ACM Human Factors in Computing Systems (CHI 2001, Seattle, Washington)*. ACM Press, New York, pp. 293-300.
- Jul, S., Furnas, G. W., 1998. Critical zones in desert fog: Aids to multiscale navigation. *Proceedings of ACM Symposium on User Interface Software and Technology (UIST 98, San Francisco, California)*. ACM Press, New York, pp. 97-106.
- Kabbash, P., Buxton, W., 1995. The "Prince" technique: Fitts' law and selection using area cursors. *Proceedings of ACM Human Factors in Computing Systems (CHI 95, Denver, Colorado)*. ACM Press, New York, pp. 273-279.
- Kelso, J. A., 1992. Theoretical concepts and strategies for understanding perceptual-motor skill: From information capacity in closed systems to self-organization in open, nonequilibrium systems. *Journal of Experimental Psychology: General*, 121, 260-261.

- Leung, Y.K., Apperley, M.D., 1994. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction* 1(2),126-160.
- Lewis, M., Jacobson, J., 2002. Game engines in scientific research: Introduction. *Communications of the ACM* 45 (1), Jan. 2002, 27-31.
- MacKenzie, I. S., 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction* 7, 91-139.
- Mottet, D., Guiard, Y., Bootsma, R. J., Ferrand, T., 2001. Two-handed performance of a rhythmical Fitts task by individuals and dyads. *Journal of Experimental Psychology: Human Perception and Performance*, 26, 1275-1286.
- Negroponte, N., 1995. *Being digital*. Alfred A. Knopf Inc., New York.
- Perlin, K., Fox, D., 1993. Pad: An alternative approach to the computer interface. *Proceedings of ACM SIGGRAPH (SIGGRAPH 93, Anaheim, California)*. ACM Press, New York, pp. 57-64.
- Plamondon, R., & Alimi, A. M. (1997). Speed/accuracy trade-offs in target-directed movements. *Behavioural and Brain Sciences* 20, 279-349.
- Soukoreff, R.W., MacKenzie, I.S., this issue. Towards a standard for pointing device evaluation – Perspectives on 27 years of Fitts' law research in HCI.
- Ware, C., Lewis M. S., 1995. The DragMag image magnifier. *Proceedings of ACM Human Factors in Computing Systems (CHI 95, Denver, Colorado)*. ACM Press, New York, pp. 407-408.
- Zhai, S., this issue. Characterizing computer input with Fitts' law parameters — The information and non-information aspects of pointing