Fitts' Law and Expanding Targets: An Experimental Study, and Applications to User Interface Design

by

Michael John McGuffin

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

Fitts' Law and Expanding Targets: An Experimental Study,

and Applications to User Interface Design

Michael John M^{C}Guffin

Master of Science

Graduate Department of Computer Science

University of Toronto

2002

There exist several user interface widgets that grow or expand in response to the user's focus of attention. Some of these expand to facilitate their selection, allowing for a reduced initial size in an attempt to optimize screen space use. However, selection performance could plausibly suffer from a decreased initial widget size. We describe an experiment in which users select a single, isolated target button that expands just before it is selected. Our results suggest that users are able to take approximately full advantage of the expanded target size, even if the target only begins expanding after 90 % of the movement towards the target has been completed. For interfaces with *multiple* expanding widgets, however, care must be taken to mitigate the collisions or overlap that may occur between adjacent widgets. We present a number of design strategies that attempt to optimize the performance of multiple, tiled expanding targets.

Acknowledgements

One of the more significant turning points in my life was my first day as a student hire at Alias|wavefront in 1996. (A couple of months earlier, I was utterly convinced that such a cool company would never hire me. Ironically, this made me completely relaxed during the interview for the position, which worked quite to my advantage.) I was assigned to the Input Research Group at Alias|wavefront, under the supervision of Gord Kurtenbach. In the space of a few hours, I was shown Marking Menus, the Rockin' Mouse, and a few other really neat projects, and I thought "Wow ! I've finally found my field of research !"

To me, Human Computer Interaction combines, on one hand, the excitement of active and creative engagement with computers, and on the other, the potential for novel and significant contributions that comes with a young field of research. I am therefore greatly indebted to the researchers I met at Alias|wavefront who introduced me to this field, and in particular to the ones who guided me through my master's:

Gord Kurtenbach — my supervisor, then and now, convinced me to pursue graduate studies at the University of Toronto, and has been a great sounding board and source of advice. He always provided encouraging and thoughtful comments in response to my ideas, no matter how nebulous or tentative.

Ravin Balakrishnan — my 2nd supervisor, whose suggestion to study "expanding targets" ultimately led to the topic of this thesis, has been an energetic teacher and role model for how to do research. Without his knowledge of experimental design and statistical analysis, our study would have never happened.

Both Gord and Ravin played pivotal roles and "fast tracked" me into the world of published research, however there are many others who contributed to this work. Wolfgang Stürzlinger, George Fitzmaurice, Joe Laszlo, Azam Khan, and other members of the Interaction Research Group at the University of Toronto provided valuable ideas, discussions, and reactions to the work as it progressed. Scott MacKenzie gave me some key pointers to previous work on Fitts' law. Michel Beaudouin-Lafon and Yves Guiard offered some important critical insights into the work. Etty Shin helped me to better understand the basis in communication theory for Fitts' law.

I also heartily thank all the students and faculty in the Department of Computer Science who donated their valuable time to participate in our experiment.

Last but not least, thanks are due to Bowen Hui, for providing precious help with LaTeX, which was used to typeset this document.

A number of people contributed indirectly to this work by shaping my development prior to graduate studies. Derrick Moser helped me along the path to mastery of UNIX™, and initiated me to the Dvorak keyboard layout (which enabled me to type this thesis with much less effort !) Brian Wong, Olga Pudelko, and Eugene Kim left indelible marks on my intellectual development, by providing me with abundant supplies of provocative conversations and imaginative speculations. Etty Shin and Martin Blais encouraged me, in small but significant ways, to pursue a career in research. I am of course also very grateful to my parents, siblings, and extended family, for educating me, supporting me, and always encouraging my academic pursuits.

Finally, I am especially thankful to my wife, Alicia, for providing balance in my life, and making this all worthwhile.

Contents

List of Figures

Chapter 1

Introduction

Several interfaces and interaction techniques have been described [18, 44, 62, 51, 7, for example] in which a widget, or portion of a widget, changes size dynamically (e.g. grows or expands) to accommodate the user's focus of attention. A larger widget or viewing region can provide the user with more information and/or a greater area for input. Widgets that dynamically grow (which we will call *expanding widgets*) can now also be found in a popular operating system [4] where the icons in the desktop toolbar expand when the mouse cursor is over them (Figure 1.1). Indeed, as software becomes more complex, with an ever increasing number of commands, buttons, and icons, an effective strategy may be to display widgets at a significantly reduced size, and expand them to a usable size only when needed. This would allow more screen real estate to be used for displaying data or content, and less for displaying user interface elements.

Making buttons or other on-screen targets small, however, may result in reducing the user's ability to select them efficiently, even if they subsequently expand to a larger size. From Fitts' law [16], we know that as a target's size decreases, the time taken to select that target increases. While Fitts' law has been empirically verified and shown to apply to many interaction scenarios [9, 41, 14], these have all been for situations where the target has a constant size. It is unclear what happens if the target changes size *after* the

Figure 1.1: Screenshots of the “dock” in Mac OS X. *(Top)* When the mouse cursor is not over the dock, the icons are in an unmagnified, rest state. *(Middle and Bottom)* If the cursor passes over the dock, the nearest icons expand, and icons further away are pushed to the side.

user has already begun moving towards it, as is the case with expanding widgets. Is the selection time governed by the original size of the target when the user begins moving towards it ? Or is the final size of the target the determining factor ? Or is the answer dependent on when the target begins to expand and how fast it expands ? Further, is it possible to predict a priori what the selection time will be for such expanding targets ?

Without answers to these questions, there is little scientific knowledge to guide the design of interfaces that incorporate expanding widgets. In particular, if selection time is determined by the initial target size, the use of expanding widgets is essentially a tradeoff between saving screen space and the ability of users to select these widgets quickly. On the other hand, if the determining factor is the final target size, then we can take advantage of the benefits of expanding targets without compromising performance. If the answer lies between these two extremes but we can accurately predict the tradeoff, this knowledge will allow designers to make informed decisions about their designs. In addition to the implications for interface design, these questions are also interesting from a human motor control standpoint, since they address the fundamental issue of whether Fitts' law can even be used to model and predict movement times when the target size changes after the onset of movement.

There are also many secondary issues to allay when designing interfaces with multiple expanding targets. Since the currently desired widget or target of the user can change from moment from moment, when should expansion occur, and for which target(s) ? Also, closely spaced targets may overlap or otherwise interfere with each other during expansion. In this case, should we allow occlusion to occur, or should some targets be displaced ?

The following chapters present background information on Fitts' law, a description of an empirical study which investigated the parameters and effects of an expanding target in isolation, applications of the empirical findings to the design of interfaces with multiple expanding targets, conclusions, and proposals for future work.

Much of this thesis (especially the experimental study) is based on work already published by McGuffin and Balakrishnan [46]. In the current work, however, more background material has been added (including two thought experiments (§2.3) devised by the author and designed to help the reader gain an intuitive understanding of Fitts' law), and the chapter on applications includes new design proposals, some of which are based on new, more ambitious mathematical analyses of user interfaces with multiple targets.

Chapter 2

Background

2.1 Introduction to Fitts' Law

Fitts' law describes the time required to acquire (e.g. hit, press, select, click on, ...) a target with a rapid, aimed movement. Given the amplitude A of motion (i.e. the distance to reach the target), and the width W (i.e. the size) of the target measured along the axis of motion, the movement time MT required to reach the target is

$$MT = a + b \log_2 \left(\frac{A}{W} + K \right) \tag{2.1}$$

The constants a and b can be determined empirically, and vary according to the nature of the acquisition task, the kind of motion performed, and the muscles used. They do not, however, vary significantly from person to person. K depends on the specific formulation of Fitts' law that is chosen, and may be 0, 0.5, 1. The logarithmic term is referred to as the index of difficulty I_d or ID, thus Fitts' law can be rewritten as $MT = a + bID$.

Fitts' law has been verified to accurately model many situations, for example: hand and foot movements [25]; movements in air, underwater [36] and under microscopic conditions [38, 37]; reciprocal "back and forth" movements [16], discrete "one-shot" unidirectional movements [17], grasping and pointing [31], dart throwing [35], goal passing [1] and crossing tasks [3]; movements with different input devices, such as the mouse, trackball, joystick, touchpad, helmet-mounted sight, and eye tracker [41]; movements with position control and velocity control devices [30]; linear and rotary movements [69]; movements involving very large ID values [19, 20]; and movements by different populations, such as mentally retarded individuals [73] and pre-school children [74].

When applied to user interfaces for computers, Fitts' law can be thought of as describing the time required to click on a virtual button (or other on-screen target) with a cursor controlled by the mouse or some other pointing device (Figure 2.1).

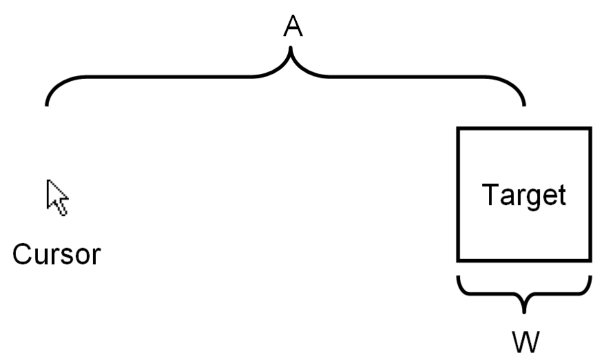

Figure 2.1: A 1-dimensional selection task: the user must move the cursor as quickly as possible onto the target of width W. The performance of the user can be predicted by Fitts' law.

As presented thus far, Fitts' law may seem fairly straight forward. For example, from Equation 2.1, we observe that targets that are farther away or that are smaller require more time to select — this much seems reasonable. However, why is there a logarithmic term in Equation 2.1 ? Even less clear is why it is customary to use bits as the unit for the index of difficulty, or why $1/b$ (measured in bits/second) is an "index of performance" or "bandwidth" that expresses the human rate of information processing.

Section 2.2 will explore these questions by giving the information theoretic foundations behind Fitts' law. Next, section 2.3 will attempt to help the reader develop a deeper appreciation for the mathematical formulation of Fitts' law, paying special attention to the logarithmic term.

2.2 History and Formulation of Fitts' Law

Shannon is well known in the fields of communication engineering and information theory for his Theorem 17 [68, p. 67], now often referred to as the Shannon-Hartley theorem or Shannon's capacity formula, regarding the capacity of an analog communication channel in the presence of white (gaussian) thermal noise:

$$C = B \log_2 \frac{S+N}{N} = B \log_2 \left(\frac{S}{N} + 1 \right) \tag{2.2}$$

C is the channel capacity in bits/second, or the maximum rate at which bits can be transmitted such that the probability of bit error can be made arbitrarily small. B is the bandwidth of the channel in Hertz. N is the power of the noise, and S is the average transmitter power, or the power of the signal.

The Shannon-Hartley theorem provides us with a theoretical upper bound on the usable capacity of a channel. A naive interpretation of Equation 2.2 provides some intuition as to why it is true. We can roughly think of $S/N + 1$ as the number of discrete values or symbols that can be encoded with the continuous signal. The $\log_2$ term then tells us how many bits are carried by each symbol. (For example, if the noise is such that at most 8 different values can be reliably distinguished, then each one carries $\log_2 8 = 3$ bits of information). If B such symbols or values can be transmitted each second, then clearly the product C is the total capacity in bits/second[1].

Fitts extended the notions of signal, noise, and channel capacity to the human motor system. In his seminal work [16] on the topic, he argues that the motor system can be viewed as a transmitter of information, where the transmission of one symbol corresponds to the execution of one motor response. A greater number of possible responses (or symbols) means that each response carries more information (or bits). Furthermore, "Since measurable aspects of motor responses [such as amplitude of movement] are continuous

[1] In fact, the situation is more complicated than this. Coding is required on the bits to achieve capacity, and a rigorous derivation of the Shannon-Hartley theorem is quite complicated. However, the simplified interpretation above is useful in understanding the basis for Fitts' law.

variables, their information capacity is limited only by the amount of statistical variability, or *noise*, that is characteristic of repeated efforts to produce the same response." [16, my emphasis]

Under this view, if a movement is repeated many times, the average time MT required to complete the movement, the amplitude A of the movement, and the variability (or *required accuracy* or *tolerance*) W in the terminal location of the movement are analogous to $1/B$, S, and N, respectively. Fitts went so far as to propose that this analogy holds at a mathematical level, pointing to examples of previous studies where the duration of a movement increases with amplitude [66, 8], or where the error in a movement increases both with amplitude and speed [79]. His hypothesis, then, was that there is a constant channel capacity associated with a given set of muscles and a given motor task, and that this capacity is independent of A and W.

Mathematically, Fitts claimed that this channel capacity, which he termed the Index of Performance I_p, should be computable as

$$I_p = \frac{1}{MT}\log_2\frac{2A}{W} \tag{2.3}$$

This implies that

$$\underbrace{MT}_{duration\ of\ movement} = \underbrace{\frac{1}{I_p}}_{duration\ of\ each\ bit} \underbrace{\log_2\frac{2A}{W}}_{bits/movement} \tag{2.4}$$

Fitts' used Equation 2.3 to compute the channel capacity for different values of A and W within four different tasks (reciprocal tapping (Figure 2.2) with a 1 ounce stylus and with a 1 pound stylus, disc transfer, and pin transfer). The value was found to be approximately constant for each task, which provided the first evidence of the validity of Equation 2.4, which is now known as (the original formulation of) Fitts' law.

We can now make the analogy between Fitts' law and the Shannon-Hartley theorem explicit by rewriting Equation 2.2 as

$$\underbrace{\frac{1}{B}}_{duration\ of\ each\ symbol} = \underbrace{\frac{1}{C}}_{duration\ of\ each\ bit} \underbrace{\log_2\left(\frac{S}{N}+1\right)}_{bits/symbol} \tag{2.5}$$

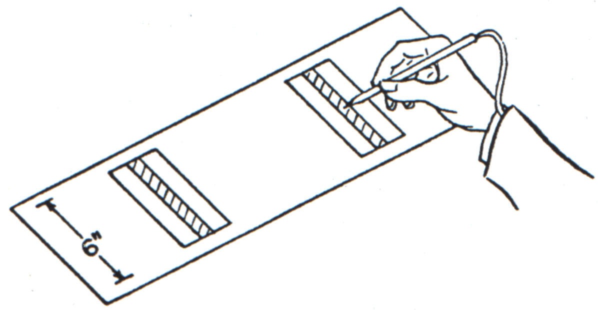

Figure 2.2: Reciprocal tapping apparatus used by Fitts. "The task was to hit the center plate in each group alternately without touching either side (error) plate." [16] (Figure reproduced from Fitts [16].)

where the (simplified) interpretation of each term is indicated.

The only difference between Equations 2.4 and 2.5 lies within the $\log_2$ term, where Fitts replaced the addition of unity with a multiple of 2. Fitts claims that this factor is "arbitrary" [16, p. 388 & p. 390] and was chosen for convenience. Because of this factor of 2, the index of difficulty ID is 0 when $A = W/2$ [40, p. 325] (unfortunately, ID is undefined when $A = 0$). Notice, in addition, that the factor of 2 means we can rewrite the linear Equation 2.4 as an affine equation:

$$MT = \frac{1}{I_p} + \frac{1}{I_p} \log_2 \frac{A}{W} \tag{2.6}$$

Were the factor of 2 changed to some other value, the change would affect each of the computed I_p values differently, which would change the degree to which Fitts' hypothesis is supported. Thus, it seems objectionable to claim that the choice of the factor is truly arbitrary.

Subsequent researchers have tested variations of Fitts' original equation against experimental data, usually adding a degree of freedom that allows the intercept in Equation 2.6 to change and be fitted to the data. For example, Welford [76] [77, p. 147] proposed

$$MT = a + b \log_2 \left(\frac{A}{W} + 1/2 \right) \tag{2.7}$$

where both $b = 1/I_p$ and the intercept a are empirically measured regression parameters. This form can improve fit to data [76], and was even used by Fitts in subsequent work

[17].

Another variation, which is now the most generally accepted form, is the Shannon formulation

$$MT = a + b\log_2\left(\frac{A}{W} + 1\right) \tag{2.8}$$

which is arguably preferable for both theoretical and practical reasons [40, 41]. In particular, the Shannon formulation always yields a non-negative index of difficulty, even when $A = 0$; has been show to provide a better fit with observations; and exactly mimics the Shannon-Hartley theorem.

Card et al.'s 1978 comparative study [9] of different input devices was the first application of Fitts' law to Human Computer Interaction. (Fitts' law provides a standard scale for comparing pointing devices, through measurement of their index of performance.) Over time, a large body of literature on Fitts' law has grown within the fields of Human Computer Interaction and psychomotor studies. MacKenzie maintains an online bibliography [42] of Fitts' law research, which at the time of writing lists 310 papers. Fitts' law has remained one of the few robust predictive tools available to HCI practitioners (notably, it has been joined recently by Accot's steering law [1, 2], which is itself derived from Fitts' law !).

In summary, three major formulations of Fitts' law have been presented. Fitts' original formulation (Equation 2.4), Welford's formulation (Equation 2.7), and the Shannon formulation (Equation 2.8).

There are other variations on Fitts' law that have been proposed for modelling rapid, aimed motion (for example, see [57], or see [40, p. 325] or [41, pp. 114–116] for lists of additional references). However, the basic logarithmic form already presented has remained the most popular in literature. Indeed, the logarithmic form has been successfully *derived* from various higher level models. For example, Crossman and Goodeve [12] described a first order continuous control system where the instantaneous velocity is proportional to the current error (i.e. the distance left to traverse). The settling time for this system yields Fitts' law. Langolf et al. [37] describe a second order underdamped control system whose settling time also corresponds to Fitts' law. In addition, Langolf et al. [37] describe a discrete response model, originally developed by Crossman and Goodeve [12] and later used by Keele [34], where each corrective movement incrementally improves accuracy by a constant ratio, and requires a constant time. Fitts' law can also be derived from this model. Finally, Meyer et al. [47] showed that Fitts' law is a limiting case of a more general equation sometimes called *Meyer's law* [63, p. 211].

2.3 Interpretation of Fitts' Law

The point of this section is to analyze the mathematical formulation of Fitts' law and develop some intuitive understanding or insight into it, in part by performing thought experiments.

Perhaps the most obvious feature of Fitts' law is the *scale invariance* due to MT being a function only of A/W. This makes clear the speed/accuracy tradeoff that is fundamental to aimed, rapid movements. Further, as stated by Welford, "The essential point of this formulation [of Fitts' law] is that it makes movement time constant for any given ratio between amplitude and target width." [77, p. 145] (Figure 2.3). This means that, for example, a target that is twice as far away and twice as large requires the same time to acquire. Why should this be the case ? Although a larger value of A means there is more distance to traverse, it also means that there is more distance over which to accelerate. Indeed, Hartson [22] claimed that a fixed duration was the basic characteristic of ballistic movements. At the same time, the accuracy of the terminal location of a quick movement decreases with amplitude, and Fitts states that this was known for "many years" [16, p. 383] prior to his 1954 paper.

Thus it seems reasonable that MT should be a monotonically strictly increasing function of A/W. This still does not explain, however, the logarithm in Fitts' law.

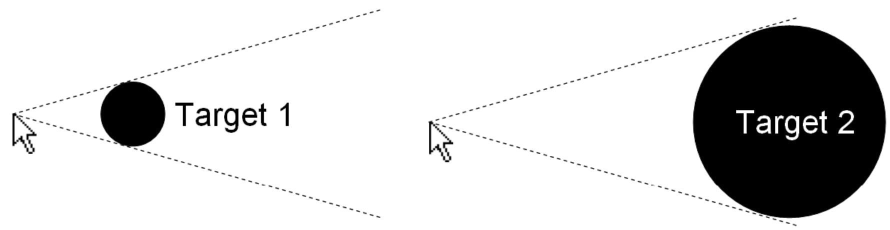

Figure 2.3: Scale invariance of Fitts' law: the two targets illustrated require the same time to select, because the ratio of A/W is the same in both cases.

(Interestingly, a logarithm is also present in the Hick-Hyman Law [77, pp. 61–65] [23] [28], which predates Fitts' law, and is closely related in mathematical form.) Following are some thought experiments meant to provide some intuition for the rationale behind the logarithmic term. These are inventions of the author, and rest on many assumptions — as such, they should not be seen as established or well accepted. They are submitted, however, for consideration by the reader.

2.3.1 Gedankenexperiment 1: An infinite sequence of buttons

This thought experiment is meant to provide some insight as to how Fitts' law describes the motor system's capacity to transmit bits of information.

Imagine an infinite sequence of buttons, each of width W, arranged along an axis, and labelled with the binary integers $0, 1, 10, 11, 100, 101, \ldots$ in ascending order (Figure 2.4). The pointer is located at the origin of the axis, and can be moved to the right to click on a button. Buttons are centred at multiples of W. Furthermore, we assume that whenever the user clicks on a button, the pointer (and the state of the user's motor system) are instantaneously returned to the origin position.

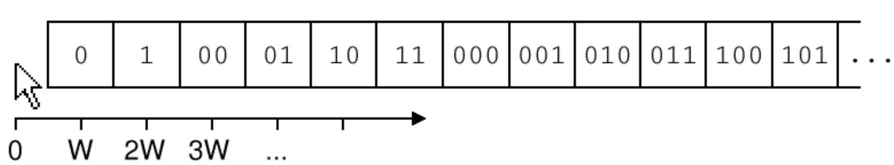

Figure 2.4: An infinite sequence of buttons enumerating all binary integers, and the user's pointer at the origin on the left.

Imagine that the user wishes to enter (or transmit) a string of bits by clicking these buttons. For example, to enter the string "000011", the user could optionally hit the 0-button four times followed by the 1-button twice, or hit the 000-button followed by the 011-button, etc. Buttons labelled with more bits allow the user to enter more information with one click, but the user must travel farther to the right to reach such buttons.

Clicking on an n-bit button requires a movement of amplitude A in the range $[A_{min}, A_{max}] = [(2^n - 1)W, 2(2^n - 1)W]$, so the mean amplitude for n-bit buttons is $A_{average} = \frac{3}{2}(2^n - 1)W$.

Assume the time required to click on an n-bit button is

$$MT = b \log_2 \left(\frac{A}{W} + K \right)$$

where K is, for example, 0.5 or 1. Now, consider that the user must enter an N-bit string by clicking on n-bit buttons. The user will have to click N/n times, yielding an average total time of

$$\begin{aligned}
\frac{N}{n} MT_{average} &\approx \frac{N}{n} b \log_2 \left(\frac{A_{average}}{W} + K \right) \\
&= \frac{N}{n} b \log_2 \left(\frac{3}{2} 2^n - \frac{3}{2} + K \right) \\
&\approx \frac{N}{n} b \log_2 \left(\frac{3}{2} 2^n \right) \\
&= \frac{N}{n} b \left(\log_2 2^n + \log_2 \frac{3}{2} \right) \\
&= \frac{N}{n} b \left(n + \log_2 \frac{3}{2} \right) \\
&= Nb \left(1 + \frac{1}{n} \log_2 \frac{3}{2} \right) \\
&\approx Nb \left(1 + \frac{0.585}{n} \right) \\
&\approx Nb
\end{aligned}$$

The final approximation is valid if n is large, in which case the expression becomes independent of n. Thus, under appropriate simplifying assumptions, the time required to enter an N-bit string is Nb, regardless of whether the user clicks on many nearby buttons that each transmit few bits, or on few distant buttons that each transmit many bits. It seems all the more fitting, then, that the units of b should be seconds/bit, and that the index of difficulty computed by the $\log_2$ term is in bits.

The following modification to the experiment is also informative: rather than requiring the pointer to return instantaneously to the origin after each click, we can allow the user to freely move from one button to the next between clicks. There are 2^n n-bit buttons, and their centres cover a length of $(2^n - 1)\,W$ on the axis. The mean distance between two points randomly selected on a unit segment happens to be 1/3, hence the the average distance to move from one n-bit button to another is $\frac{1}{3}\,(2^n - 1)\,W$. Substituting this as the value for $A_{average}$ above, we obtain the same final result.

2.3.2 Gedankenexperiment 2: A compound selection task

It is well known that the transmittivity of (i.e. the fraction of light that passes through) a material falls off exponentially with the thickness of the material:

$$(transmittivity) \propto e^{-(thickness)} \tag{2.9}$$

One explanation for the exponential falloff is that we require the two situations in Figure 2.5 to be equivalent — that is, the product of the transmittivities of two slabs in sequence should be the same as the transmittivity of a single slab formed by "gluing" the first two together.

Mathematically, this translates into the requirement that $Transmittivity(thickness_1) \times Transmittivity(thickness_2) = Transmittivity(thickness_1 + thickness_2)$, and an exponential function meets this requirement.

It is proposed that an analogous (if only approximate) equivalence exists for selection tasks modelled by Fitts' law, and further that this equivalence explains the necessity of a logarithmic term in Fitts' law. Specifically, we wish to show that a compound selection task, where the user must first select a large target $Target_1$ and then a smaller, nested target $Target_2$, requires the same time as a simple selection task where the user only

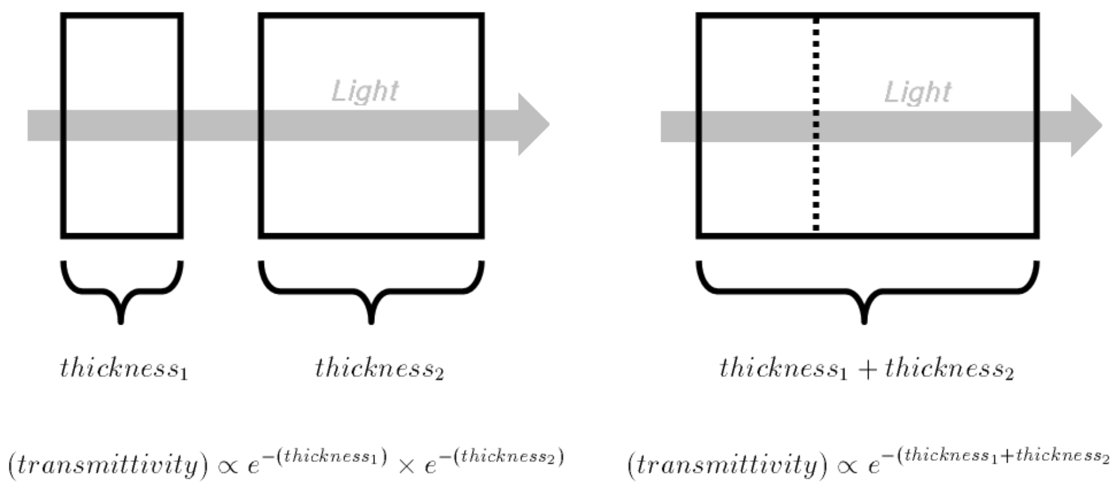

Figure 2.5: On the left, two slabs in sequence have an overall transmittivity equal to the product of their individual transmittivities. On the right, the two slabs have been glued together, and the transmittivity is thus a function of the sum of the original thicknesses.

selects $Target_2$.

Let W_1 and W_2 be the respective widths of the targets, and let A_1 be the distance to the first target (Figure 2.6). Assume the motion of the user is restricted to the horizontal axis, so the heights of the targets are inconsequential.

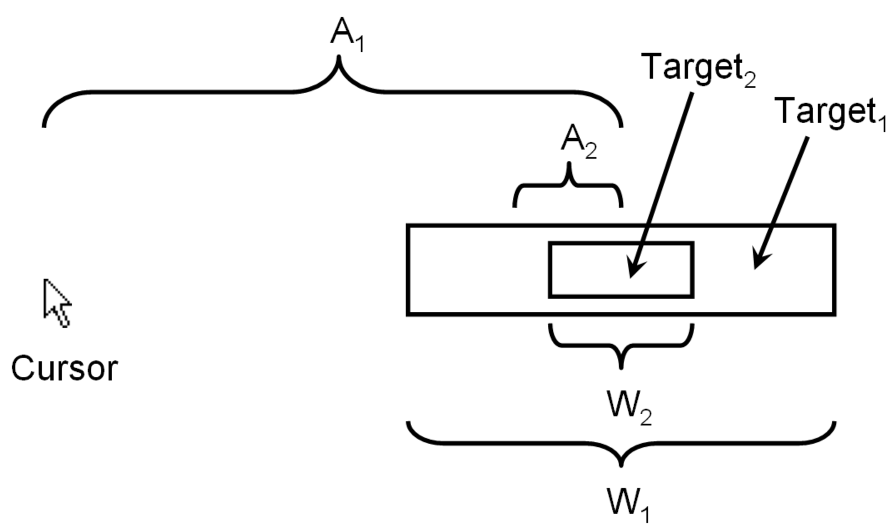

Figure 2.6: A compound selection task, where the user must first acquire $Target_1$, and then acquire $Target_2$. Since this is a one-dimensional selection task, the heights of the targets do not matter.

For the compound selection task, assume that at the end of the first selection, the cursor is at a random, uniformly distributed point on $Target_1$. The mean distance to $Target_2$'s centre will then be $A_2 = \frac{1}{4}W_1$. Next, assuming a version of Fitts' law of the form

$$MT = a + b \log_2 \frac{A}{W}$$

we find the total time for the compound selection task is

$$\begin{aligned} MT_1 + MT_2 &= a + b \log_2 \frac{A_1}{W_1} + a + b \log_2 \frac{A_2}{W_2} \\ &= 2a + b \log_2 \frac{A_1 A_2}{W_1 W_2} \\ &\approx 2a + b \log_2 \frac{A_1 W_1}{4 W_1 W_2} \\ &= 2a + b \log_2 \frac{A_1}{4 W_2} \\ &= 2a - 2b + b \log_2 \frac{A_1}{W_2} \end{aligned}$$

In comparison, the time for a simple selection of only $Target_2$, starting from the same initial point, is $a + b \log_2 \frac{A_1}{W_2}$. If $a = 2b$, these two tasks require exactly the same time. It is doubtful that there is much if any data to support the hypothesis that $a = 2b$, however, this is rather beside the point. The more important result of this thought experiment is that the two logarithmic terms in the compound task time reduce to a single logarithm, *independent* of W_1, and almost identical to the logarithm corresponding to the simple selection task (the only difference is a division by 4, which is negligible for large $\frac{A_1}{W_2}$ ratios.) The reduction occurs because of the property that a sum of logarithms is equal to the logarithm of a product.

An interesting variation on this thought experiment is possible. Rather than thinking of the A in Fitts' law as the distance to the target, we can instead think of it as "the entropy [or noise] of a hypothetical initial distribution of motion amplitudes" [12, p. 252]. Such a point of view seems to have been suggested by Crossman [11] [12, p. 252]. Under this view, we would have $A_2 = W_1$, and the desired equivalence is *exact* — if we ignore the intercept a.

2.4 Two-Dimensional Selection Tasks

As presented so far, Fitts' law concerns aimed motion along a *single dimension*, toward a target whose width W is measured along the axis of motion. Extending Fitts' law to 2D selection tasks must be done with care. First, performance can vary according to the angle of approach. For example, in Jagacinski and Monk's study [29], diagonal motions took slightly longer than horizontal or vertical motions. Second, unless the target is circular, it is not clear what its "width" should be for the purposes of Fitts' law. For example, rectangles that have a very large horizontal width W but a very small vertical height H are more difficult to select when approaching from above than a $W \times W$ square. MacKenzie and Buxton [43] compared the appropriateness of different measures of the size of a 2D rectangular target. These were: the horizontal width W, the area WH, the sum $W + H$ of the width and height, the width $W' = \min\{H/|sin\theta|, W/|\cos\theta|\}$ of the rectangle measured along the axis of motion, and the smaller $\min\{W, H\}$ of the width and height. Of these, the last two were found to be the best measures of target size for the purposes of Fitts' law. Furthermore, the last two were found to not differ significantly in their correlations.

2.5 Selection Tasks with Moving Targets

Jagacinski et al. [30] measured acquisition times for capturing a target moving with constant velocity. Although Fitts' law was able to predict performance when users used a rate control input device, it failed with position control input devices. Jagacinski et al. proposed, without formal derivation, a new index of difficulty, which is a function of the target's speed, to model the measured times.

Subsequently, Hoffmann [24] proposed a different mathematical law for describing Jagacinski's data. Hoffmann gives 3 different derivations of the law, using a first order continuous control system, a second order continuous control system, and a discrete response model. The resulting law fits Jagacinski's data. Interestingly, the law also predicts a critical target speed, beyond which target capture is not possible.

More recently, Port, Lee, et al. [59, 39] studied a different task, where subjects had to *intercept* a moving target within a given "interception zone". Trials where the cursor arrived in the interception zone more than 100 ms earlier than the target, or more than 100 ms later than the target, were classified as *early errors* or *late errors* respectively. Port, Lee, et al. developed models for predicting performance in this task.

Unfortunately, to our knowledge, there have been no studies of tasks where users had to capture a target that begins moving *after* the user has started to move toward the target. An accurate model for this task would be useful for evaluating the designs that are presented in Chapter 4.

2.6 Issues in Motor Control

To help us hypothesize about user performance with expanding targets, it is useful to consider the possible underlying motor control models that may be behind Fitts' law.

One explanation, called the *iterative corrections model* [12, 34], attributes the law entirely to closed-loop feedback control. This model states that the whole movement consists of a series of discrete submovements, each of which takes the user closer to the target and is triggered by feedback indicating the target is not yet attained.

Another explanation, called the *impulse variability model* [64], attributes the law almost entirely to an initial impulse delivered by the muscles, flinging the limb towards the target. The last part of the movement time consists of the limb merely coasting towards the target.

It has been pointed out [78, 63], however, that neither of these two explanations adequately accounts for all the effects shown in the large body of experimental data in the literature.

The most successful and complete explanation to date [63], called the *optimized initial impulse model* [47], is a hybrid of the iterative corrections model and the impulse variability model. This suggests that the process modeled by Fitts' law is as follows (Figure 2.7): An initial movement is made towards the target. If this movement hits the target, then the task is complete. If, however, it lands outside the target, another movement is necessary. This process continues until the target is reached. Since the goal is to reach the target as quickly as possible, in an ideal case the subject should make a single high-velocity movement towards the target. In reality, however, the spatial accuracy of such movements is highly inaccurate. It can be shown [47, 63] that the standard deviation S of the endpoint of any movement increases with the distance D covered by that movement, and decreases with its duration T:

$$S = k\frac{D}{T} \tag{2.10}$$

where k is a constant. Thus, a movement with a long distance and short duration could be executed, but would result in a high standard deviation and therefore a low probability of actually hitting the target. Conversely, a series of long duration and short distance movements could be executed, hitting the target with certainty, but the total movement time would be extremely long. The solution, therefore, is to find the optimal balance of D and T that minimizes the total movement time [63, p. 211]. In essence, this means that most aimed movements consist of an initial large and fast movement that gets the subject reasonably close to the target, followed by one or more shorter, and slower, corrective movements that are under closed-loop feedback control.

Based on this explanation, in the situation where the target's width expands at some point during the movement, it can be expected that the first large and fast movement towards the target is planned and executed with the initial, unexpanded, target width as the input parameter to the subject's motor control system. However, subsequent corrective submovements should, according to this model, be able to respond to changes in the target's size since these submovements are under closed-loop feedback control.

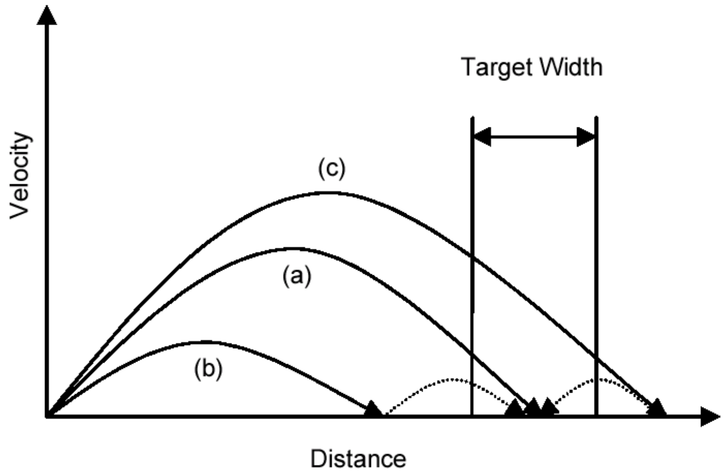

Figure 2.7: Possible sequence(s) of submovements toward a target as described by the *optimized initial impulse model* [63]. (a) is the case where a single movement reaches the target. (b) and (c) are the more likely cases where the initial movement under or over shoots the target, requiring subsequent corrective movements.

This is the key part of our main hypothesis, to be tested experimentally (see Chapter 3), that users will benefit from expanding targets.

2.7 Cursor Trajectory Prediction

Algorithms for predicting the desired trajectory (or target) of a cursor could be useful for aiding the user in performing selections. The earliest work we are aware of is a 1989 article by Miyasato [48] which evaluated 5 different prediction schemes. Subsequent work by Murata [52, 53, 54] examined issues such as prediction accuracy as a function of number of targets, target positioning, and sampling rate.

Baldwin et al. [5] developed and evaluated 3 different predictive Kalman filters to anticipate cursor motion and reduce visual latency in a telepresence application. A physical model of the mouse as a point mass under the influence of a constant external force and friction corresponds to the first filter. Neglecting friction yields a simpler, constant acceleration model, which corresponds to the 2nd filter. Neglecting the external force yields an even simpler, constant velocity model, corresponding to the 3rd filter. The 3rd filter was found to be the most accurate for predicting cursor motion. In subsequent work

[6], Baldwin et al. improved the external force + friction model by better determining the error covariance matrices for the filter.

Cursor trajectory prediction has potential applications in the design of haptic interfaces. Haptic pointing devices can be made to "stick" or be attracted to a widget, making the widget easier to select. However, in situations with multiple widgets, the forces experienced when incidentally passing over widgets can be a hindrance to the user [49, 56, 55]. Accurate prediction of the user's *desired* target would allow the interface to only activate haptic forces when the user is near this widget. Münch et al. [50, 49] have proposed a target prediction system which gradually learns from the behaviour of the user. The system uses both trajectory information and the pattern of dialog of the user (i.e. recording the most frequently used sequences of targets) [49]. Two drawbacks of this system are: (i) time is required for the learning phase before prediction becomes accurate, and (ii) it is not clear that the prediction would ever be accurate enough to deal with closely-spaced widgets such as toolbar buttons or menu items [56]. Dennerlein and Yang [13] have considered the practicality of a partially successful target prediction system.

Oirschot and Houtsma [71, 72] studied the accuracy of prediction based on trajectory. Their findings indicate that the parameters of a good prediction algorithm would have to vary greatly across devices and users.

In unpublished work, Mensvoort and Oirschot [70] have proposed using genetic algorithms to determine a good cursor trajectory prediction algorithm.

All of the preceding work could be relevant to the design of expanding targets. In a situation with multiple targets on a screen, the best strategy may be to use the current trajectory of the mouse pointer to predict which target the user is aiming for, and then expand that target.

2.8 Optimization of Selection Tasks

Fitts' law has been used to guide the arrangement of widgets in order to optimize (or, at least, reduce) average selection time. For example, Sears and Shneiderman described *Split menus* [67], in which the most frequently accessed menu items are moved to the top of the menu to reduce the distance to them. Hoffmann [26, 27] studied physical arrays of controls (such as knobs) and modelled the task of adjusting one control as a two part task. The first part requires the user to reach the general location of the control (this is made easier if the control is larger), and the second part requires the user to insert their fingers into the space between adjacent controls (this is made easier if there is a large space between controls). Given a required density of controls, Hoffmann shows how to compute the optimal control size.

More recently, Schmitt and Oel [65] used simulated annealing to find the optimal arrangement and sizes for static, square buttons on a 2D plane, given the pairwise probabilities $w(i, j)$ that the user will travel from button i to button j.

In Chapter 4, we will apply optimization strategies to finding optimal sizes for a linear strip of buttons. Our work differs from that of Schmitt and Oel [65] in that (i) we limit attention to a 1D arrangement of buttons, (ii) we require that the ordering of buttons never change, (iii) we have no *a priori* knowledge of any probabilities associated with the buttons, and (iv) in our work, the optimal arrangement changes over time, adjusting to the user's current behaviour and changing the current expansion of widgets.

2.9 Non-linear Magnification

Since we're concerned with targets that expand, it is informative to examine how expansion has been used in other user interface schemes.

A large body of literature [33] exists on *non-linear magnification* schemes. Within Human Computer Interaction, examples include fisheye lenses [18, 51], the perspective wall [44], the document lens [62], and fisheye menus [7]. Recently, Carpendale introduced a framework [10] within which many fisheye schemes are unified. A theme common to most of these examples is an attempt to optimize screen space use by packing a dense data set into the area of the screen, and then magnifying the currently relevant portion of the data while maintaining a sense of surrounding context (hence the term "focus+context displays"). These schemes emphasize the *display* of information rather than the selection of targets.

Recently, issues of selection within such displays have been given more attention. For example, Gutwin [21] describes a problem with fisheye displays where approaching a target with the pointer causes the target to shift in the opposite direction of the pointer's motion. As a remedy, Gutwin suggests reducing the magnification of the fisheye display as a function of pointer speed. As we will see, the same problem exists in the Mac OS X dock (which can, in fact, be thought of as a 1D fisheye lens). Our work differs from Gutwin's in that, rather than trying to fix an existing problem with expanding interfaces, we try to use expansion to improve selection performance beyond that in normal (unexpanding) interfaces.

2.10 Summary

An overview of Fitts' law has been given, with attention paid to forming an intuitive understanding for its formulation. The underlying motor control aspects behind Fitts' law were also discussed, allowing us to hypothesize that users should benefit from expanding targets. However, despite previous studies involving moving targets, human performance with expanding targets is an open question. The next chapter describes experiments that explore this question.

In this chapter, we have also discussed two-dimensional selection tasks, cursor trajectory prediction, and optimization of selection tasks. All of these will be useful in

Chapter 4, where user interface designs are proposed that incorporate multiple expanding targets.

Chapter 3

Experiment with Expanding Targets

In this chapter, an empirical study is presented which investigates if human performance when selecting expanding targets can be accurately modeled and predicted and what, if any, are the factors that influence that performance. We explore the effect of varying the time at which the target begins to expand. We also explore two different expansion strategies. We determine if performance in such tasks is governed by the initial or final target size, or a combination of both. In the following chapter, we discuss how this work applies to the design of expanding widgets, and present some initial design ideas.

As explained in Section 2.6, the optimized initial impulse model suggests that corrective movements toward the end of a motion are performed under closed-loop feedback control, and therefore should be able to take advantage of an enlarged target size. Our main hypothesis, therefore, is that in most cases, target acquisition time should be dependent largely on the *final* target size and not the *initial* one at the onset of movement. In the following experiment, we empirically verify this hypothesis.

There remains the question as to when the target should begin expanding. A safe option would be to expand the target sometime during the execution of the initial movement, and have it completely expanded before the subject plans and executes the corrective submovement(s). From an interface design standpoint, however, it would be advantageous to be able to delay expansion of the target to the last possible moment. This would allow for the interface widgets to remain small and not obscure other more important elements of the display until absolutely needed. At the same time we want to gain whatever advantage the expanded target size will have on target acquisition time. Thus, it is critical to determine this crossover point at which the target *must* expand in order to realize the significant advantages of such expansion.

3.1 Goals

Our experiment is designed to answer the following questions for a typical discrete target selection task where the target's width expands dynamically after the onset of movement towards that target:

1. Can such a task be modeled by Fitts' law ?
2. If it can be modeled by Fitts' law, is it possible to predict performance in such tasks from a base set of data where no expansion takes place ? In other words, if we obtain a Fitts' law equation for the base case, can movement time for the expansion case be determined simply by substituting new values for target width W ?
3. Is it true, as suggested by our analysis in the previous section, that movement time is dependent on the final target width and not the initial one at onset of movement ?
4. At what point should the target begin expanding ?
5. Do different target expansion strategies affect performance ?

3.2 Apparatus

The experiment was conducted on a graphics accelerated workstation running Linux, with a 21-inch, 1280×1024 resolution, colour display. A puck on a Wacom Intuos 12×18 inch digitizing tablet was used as the input device. The puck was used to drive the system cursor, and worked in absolute mode on the tablet with a constant linear 1-1 control-display ratio.

3.3 Task and Stimuli

A discrete target selection task was studied. As shown in Figure 3.1, a small box appeared on the left of the screen. Subjects were asked to move their cursor into this box. Once the cursor had dwelled in the box for one second, a rectangular target appeared on the right of the screen. Subjects were instructed to move the cursor as quickly and accurately as possible into the target, and to indicate completion by clicking the puck button. Timing began when the target appeared, and ended when the target was successfully selected. We collected all movement data so that we could later identify reaction time, and the start of actual movement. Also, while there were no "error" trials per se, the data allowed us to subsequently identify when subjects made mistakes and clicked outside the target.

3.4 Pilot Study

We first conducted a pilot study with three subjects in order to get a sense if all the experimental conditions we were considering would actually have significant effects on performance. This would not only tell us if we were on the right track, but would possibly allow us to eliminate any extraneous conditions which would lengthen and complicate the final experiment without corresponding benefits.

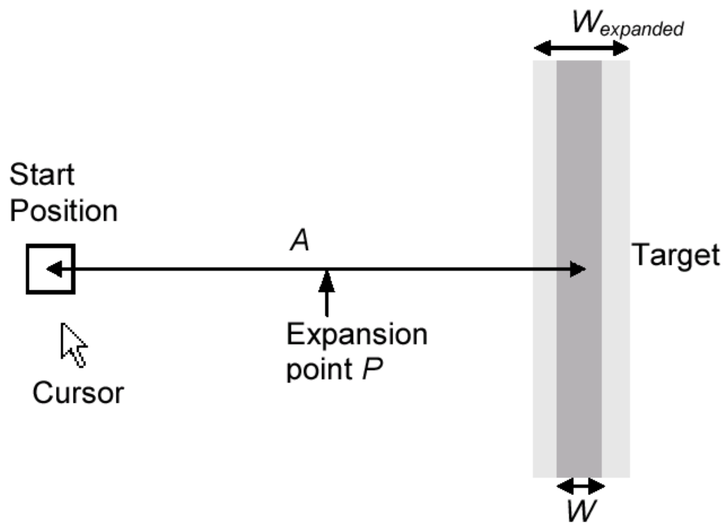

Figure 3.1: Stimuli. In the base case, the target had a width of W. In the expanded cases, the target began with a width W but expanded to $W_{expanded}$ when the cursor moved past a specified expansion point P. The amplitude A was measured from centre of start position to centre of target.

3.4.1 Design

There were three conditions which manipulated the target expansion parameter:

- *Static.* This is a base case of a standard Fitts' law style aiming task which serves as a basis for comparison.

- *Spatial expansion.* The target width grows from W to $W_{expanded}$ over a given expansion time period T. This is likely to be the preferred expansion strategy in real interface design. Gradual expansion is chosen to avoid the visual jarring that might occur if the target changed size instantly. (An instant visual change might cause "loss of context" and require the user to visually reacquire the target.)

- *Fading-in expansion.* The target width is expanded instantly at a given time, but, on the screen, the enlarged size of the target is faded-in (at full size) gradually over time T. Here, the benefit of the larger target is available to the user instantly in the *motor domain* (the set of all possible mouse positions) while the gradual visual fade-in again prevents any visually jarring effects in the *visual domain* (what the user sees on the screen).

For both expansion conditions, target expansion time T was set at 200 milliseconds which gave the impression of a smooth visual transition between target sizes. For both expansion conditions, we also had three different values for the point P at which the target began to expand: 1/4, 1/2, 3/4 of A measured from the starting point.

Thus, in summary, we had a total of seven conditions: base case, spatial expansion with $P = 1/4$, 1/2, and 3/4 respectively, and fading-in expansion with $P = 1/4$, 1/2, and 3/4 respectively.

For all the conditions, in units of 16 pixels, we used four target widths ($W = 0.5$, 1, 2, and 4 units), fully crossed with four target amplitudes ($A = 8$, 16, 32, and 64 units) resulting in sixteen A-W combinations with seven levels of task difficulty (ID) ranging from 1.58 to 7.01 bits.

In all cases, the expanded target width $W_{expanded}$ was set to twice the initial target width W. While we conceivably could have varied this parameter as well, we felt that a $2\times$ magnification was representative of what would be used in real interface widget design and was sufficient to address the main goals of the present study.

A repeated measures design was used for each of these conditions — subjects were presented with five blocks, each consisting of all sixteen A-W combinations appearing five times each in random order within the block. Subjects were allowed to rest between blocks.

3.4.2 Pilot Results and Discussion

Regression analyses showed that the data for all conditions fit the Fitts' law equation with r^2 values above 0.97. This is good news in that the selection of expanding targets can be modeled using Fitts' law.

A repeated measures analysis of variance showed a significant main effect for the seven main conditions ($F_{2,6} = 61$, $p < .0001$). Pairwise means comparison tests showed that the base condition significantly differed from the others indicating that expanding targets resulted in better performance than the non-expanding ones. This indicates that performance in the expanding target conditions is governed more by the final target width rather than its initial width.

There was no significant difference between the two different expansion strategies ($p > .05$).

Varying the value of expansion point P also had no significant effect ($p > .05$). This is excellent news for interface widget design in that target expansion can occur as late as 3/4 of the way to the target and still result in performance that is as good as if the target had expanded much earlier. In order to determine how far we could push the value of P, we performed a second pilot study with a single subject using a P value of 0.9. At this value of P, performance was not significantly different from when P was 1/4, 1/2, or 3/4. From a motor control standpoint, this indicates that the corrective submovements performed under closed-loop feedback control towards the end of movement can react quickly, accurately, and take advantage of last minute changes in target size.

3.5 Full Study

3.5.1 Subjects

Twelve volunteers (9 male, 3 female) participated as subjects in the experiment. All were right-handed and had experience with computer pointing devices.

3.5.2 Design

Given that the results of the pilot study showed no difference in performance between the two expansion strategies, we decided to only use the *spatial* expansion strategy for our full scale experiment. This was chosen as the preferred technique since, if used in real interfaces, it would avoid the visual interference of alpha blending two images as with the *fading-in* technique.

Thus, we have two main conditions, *static* and *expanding*.

Similarly, since our pilot results showed no effect on performance when expansion point P was changed, we only used a single value for P of 0.9. With such a high P, we decided to reduce the expansion time T to 100 milliseconds. This still results in smooth transition between target sizes but has the advantage of giving the user more time to react to, and advantageously utilize, the expanded target.

As in the pilot study, the expanded target width $W_{expanded}$ was set to twice the initial target width W.

Since $P = 0.9$, having conditions where the target width is initially already more than 10 % of the amplitude would mean that the user would already be in the unexpanded target before it begins to expand, thus gaining no advantage from the expansion. Accordingly, for both expansion conditions, we eliminated the three easiest *A*-W conditions (*A*-W = 8-2, 8-4, 16-4) from the original sixteen used in the pilot study. We thus have thirteen *A*-W combinations (8-0.5, 8-1, 16-0.5, 16-1, 16-2, 32-0.5, 32-1, 32-2, 32-4, 64-0.5, 64-1, 64-2, 64-4 in units of 16 pixels) with five levels of task difficulty (ID) ranging from 3.17 to 7.01 bits.

The two conditions were counter balanced between the subjects: one group of six subjects did the *static* condition first followed by the *expanding* condition, while the other group of six subjects did the *expanding* condition followed by *static* condition. The thirteen *A*-W conditions within each expansion condition were within-subjects. A repeated measures within-subjects design was used for each condition — subjects were presented with five blocks, each consisting of all thirteen *A*-W combinations appearing in random five times each within the block. Thus, the experiment consisted of 7800 trials in total, computed as follows:

12 subjects $\times$

2 conditions ×

13 *A*-*W* combinations ×

5 trials per *A*-*W* combination ×

5 blocks of trials

= 7800 trials in total

At the start of the experiment, for each of the two conditions, subjects were given a warmup block of trials consisting of a a single trial for each *A*-*W* condition, just to familiarize them with the task and conditions. Data from these warmup trials was not used in our analysis. The experiment was conducted in one sitting and lasted about 50 minutes per subject. Subjects were allowed breaks between blocks of trials.

3.5.3 Hypotheses

We expect to find the following effects in our experimental data:

H1. The *expanding* condition will result in faster movement times than the *static* condition.

H2. Performance in both conditions can be accounted for by Fitts' law.

H3. Performance in the *expanding* condition is dependent largely on the target's final size, not its initial one.

H4. Performance in the *expanding* condition can be predicted based on the Fitts' law equation generated in the base *static* condition.

3.5.4 Results and Discussion

Repeated measures analysis of variance showed a significant main effect for condition ($F_{1,11} = 1345$, $p < .0001$). The overall mean movement times were 1.335 seconds for the *static* condition and 1.178 seconds for the *expanding* condition. These results clearly indicate that expanding targets can result in improved performance, thus confirming hypothesis H1. Figure 3.2 illustrates.

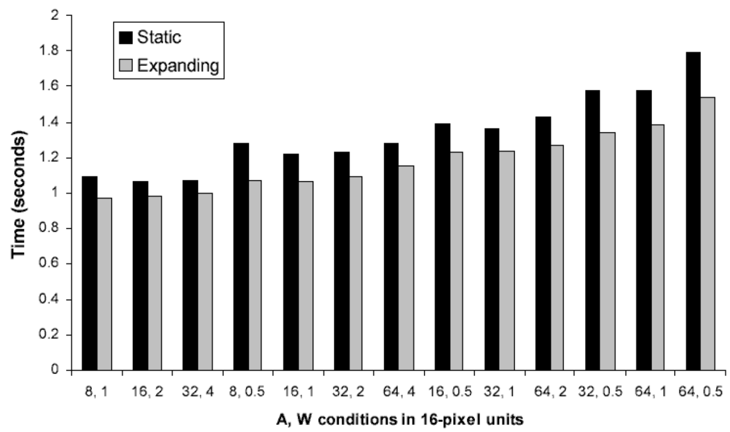

Figure 3.2: Comparison of movement times for *static* and *expanding* conditions for each A and W condition studied, for all twelve subjects.

Linear regression analysis showed that the data for each of the two conditions fit a Fitts' law equation with r^2 values above 0.97 (Figure 3.3). Thus, hypothesis H2 is confirmed.

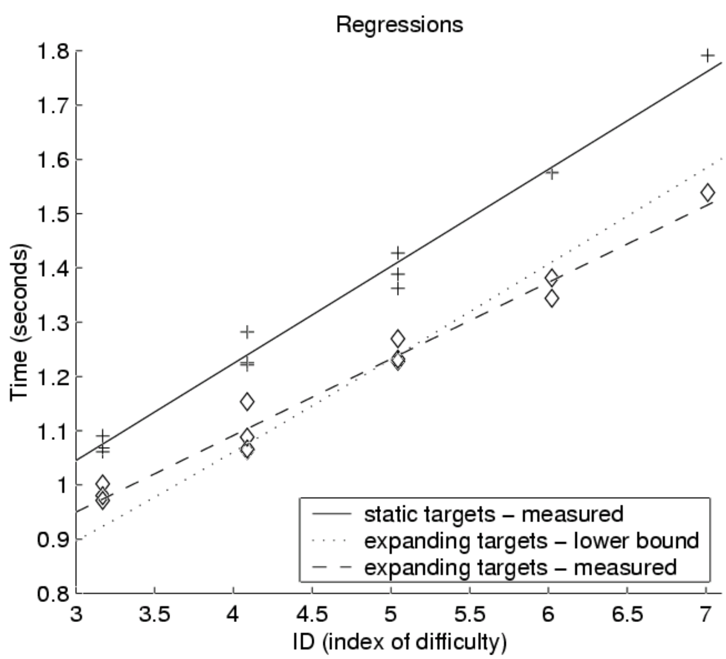

Figure 3.3: Regressions of the measured data for both conditions (solid and dashed lines), and a theoretical lower bound for the expanding case (dotted line).

Given the a and b constants used to fit the data in the static condition, we can estimate a lower bound on movement time in the expanding condition. To acquire an expanding target, the user should take at least as much time as they would to acquire a target that is always expanded:

$$MT \geq a + bID_{expanded} \tag{3.1}$$

where

$$ID_{expanded} = \log_2\left(\frac{A}{W_{expanded}} + 1\right) = \log_2\left(\frac{A}{2W} + 1\right) \tag{3.2}$$

and the initial ID of the target is

$$ID = \log_2\left(\frac{A}{W} + 1\right) \tag{3.3}$$

Solving the last two equations, we can find $ID_{expanded}$ in terms of ID and substitute into the first equation, yielding

$$MT \geq a + b\left(\log_2\left(2^{ID} + 1\right) - 1\right) \tag{3.4}$$

This bound is plotted in Figure 3.3, and as can be seen by visual inspection, is close to the data measured for the expanding condition. Although one might reasonably expect this for small values of P (the point of expansion) in which case the user would have more time to take advantage of the expanded target, our data was collected with $P = 0.9$, suggesting that the user can gain the full advantage of a large target even if the target is small for most of the acquisition task. Thus performance depends largely on the final target size, confirming hypothesis H3.

There was a significant $ID \times$ condition interaction ($F_{4,11} = 30$, $p < .0001$), indicating that the performance gains due to target expansion varied depending on the value of ID. Closer inspection of Figure 3.3 indicates that the targets with easier ID's do not benefit from target expansion as much as targets with harder ID's. It is plausible that for lower ID's, where the initial impulse movement dominates, the user is less able to react to and take advantage of an expanded target size. If this is true, we should expect the performance for expanding targets to approach that of static targets at low ID's. This possibility is sketched in Figure 3.4. However, for the ID range examined in our study (particularly at the higher end), performance with expanding targets approaches the theoretical bound, and therefore it is not surprising that the measured data can be fit to a straight line with $r^2 > 0.97$.

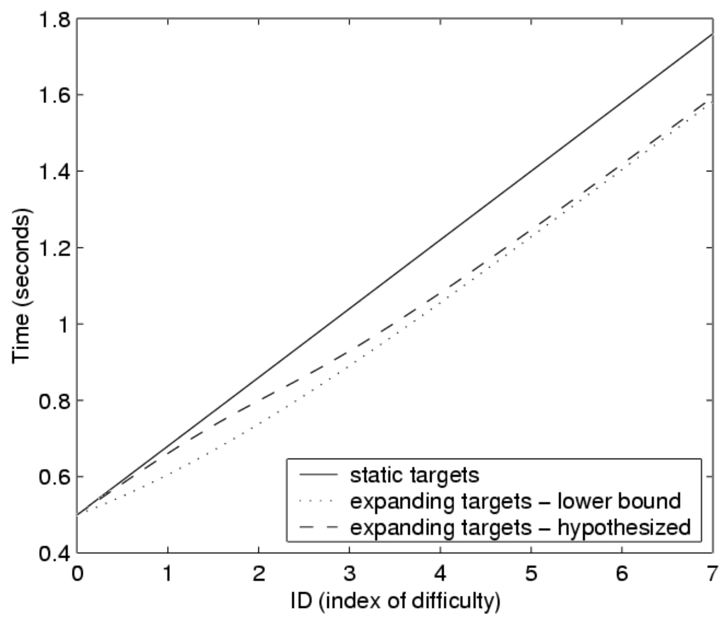

Figure 3.4: A theoretical sketch. The time MT to acquire a static target is $MT = a+bID$ (solid line). For targets that expand to twice their size, we can establish a lower bound of $MT = a + b(\log_2(2^{ID} + 1) - 1)$ (dotted line). For small ID's, where the initial impulse movement dominates, the actual movement time for expanding targets (dashed line) and static targets should be close. However, for higher ID's, closed-loop feedback control dominates, allowing the user to take advantage of the expanded target size and approach the lower bound.

Furthermore, given that the range of ID's in our study are representative of those encountered in common selection tasks, we believe therefore that the lower bound serves as a useful (if not precise) estimate of performance with expanding targets. Thus, Fitts' law can be used to roughly predict performance in the expanding case, confirming hypothesis H4.

The only other significant effect was a learning effect across the blocks of trials ($F_{4,11} = 16$, $p < .0001$), which is typical in these sorts of experimental tasks.

3.6 Summary of Findings

Our results indicate that the task of acquiring an isolated expanding target can be accurately modelled by Fitts' law. Furthermore, the degree to which performance is aided by expanding targets is governed by the target's *final* size, not its initial size. Finally, users are able to take approximately full advantage of the target's expanded size, even when expansion occurs after 90 % of the distance towards the target has been traversed.

Chapter 4

Applications to Multiple Targets

Our experimental results have significant implications for interface design, in particular for the design of buttons, menus, or other selectable widgets. Clearly, an isolated widget that expands to a larger size will be easier for the user to click on. However, when there are many such widgets on the screen, they may collide or overlap during expansion, and this leads to many subtle problems. (Interestingly, a parallel situation has been encountered in work on haptic interfaces [49, 56, 55], where single target interactions are easily enhanced, but multi-target interactions are more challenging to design.) In this chapter, a number of different designs are considered for interfaces with multiple expanding targets. The first section describes a simple design for the trivial case of "untiled targets", where we have plenty of screen space. Sections 4.2 and 4.3 explore more challenging problems. Section 4.2 treats schemes for tiled targets where the expansion depends solely on the current mouse pointer position, and includes the two prototypes previously described by McGuffin and Balakrishnan [46]. Section 4.3 describes more ambitious schemes that do not depend solely on the current mouse pointer position — it is probably here that the most potential (and work left to be done) lies.

4.1 Untiled Targets

As shown in our experiment, even if expansion occurs after 90 % of the distance toward the target has been traversed, the user still gains the full benefit of the expanded target's size. Thus an interface with multiple expanding targets need not predict the pointer's trajectory to anticipate which widget(s) to expand. Rather, simply expanding widgets that are *near* the pointer suffices to significantly facilitate selection. This also means that the user is less likely to be distracted by multiple expanding targets on screen, since expansion need only occur in proximity to the cursor (ostensibly when it is convenient for the user).

This works best if there is space between the widgets — i.e., the widgets do not tile the screen or any region of the screen. The space between widgets allows expansion to occur without interfering with or occluding any other targets on the screen.

Figure 4.1 shows an interface for visualizing a 3D mesh, with a button at each corner of the screen for selecting an alternate camera view. In this case, there is more than enough space for the buttons to expand without any mutual interference. This is also an example of how expanding targets not only make selection easier, but can use their expanded size to show the user more data (in this case an enlarged preview of the camera view, just prior to selection) when appropriate.

An important distinction can be made at this point, between the *visual domain* (what the user sees on the screen) and the *motor domain* or *motor space* (the set of all possible mouse pointer positions). Although the buttons in Figure 4.1 appear to expand, the expansion only really occurs in the visual domain. The mapping from mouse positions to buttons is fixed, hence in the motor domain there is no expansion — in fact, the buttons have a fixed (but still large) size in motor space (see the right frame of Figure 4.1).

Thus, we can think of the expanding buttons in this case as simply multiplexing what is displayed on the screen. Moving toward a corner causes an enlarged button to be displayed, and moving back to the centre causes the central view to take up most of the

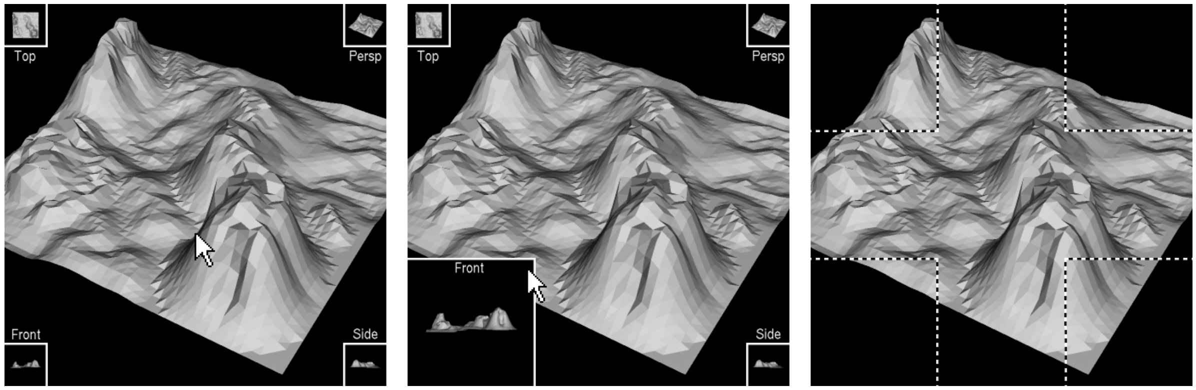

Figure 4.1: In this interface, a button for switching to a different view of the mesh is located at each corner of the screen. *Left:* the cursor is near the centre of the screen, and the buttons are in their "rest" state, allowing the mesh being viewed to occupy more screen space. *Middle:* the cursor approaches a button, and the button expands, making itself easier to acquire and also showing the user an enlarged preview of the view that would be selected. *Right:* dotted lines show that, in the motor domain, the four buttons are actually fixed in size.

screen. However, the mapping from pointer locations to widgets never changes. As a result, the space within the dotted regions in the right frame of Figure 4.1 *cannot* be used to click on the 3D mesh; it is only used to display the 3D mesh when the pointer is near the centre of the screen.

In summary, the advantages of these expanding buttons should be clear: they do not take up the screen space of large buttons, but at the same time should be as easy to select as large buttons.

4.2 Tiled Targets without Motor Domain Expansion

Typically, widgets such as buttons or menu items are grouped into arrays (e.g. toolbars) and arranged adjacently to save screen space. Widgets that are "tiled" like this have no space between them, hence simply expanding one widget will occlude neighbouring widgets, making them harder to select.

Rather than expanding an individual widget, we might try to expand the entire group of widgets around the group's centre, avoiding occlusion. For small groups of widgets, such as floating panels of a few tools, this might work well. However, if the group is large, widgets on the group's periphery will be moved far from their original position during expansion. Such widgets would thus be moving targets for the user.

4.2.1 Imitating the Mac OS X dock

An alternative is to expand the nearest widgets, and to move adjacent widgets out the way. This strategy is used in the Mac OS X dock [4], although not to facilitate selection: icons in the dock are expanded only after the pointer has already moved over them. We have built a prototype [45] that uses this strategy to aid selection. Figures 4.2 A and B show the prototype's button strip before and after the pointer moves over a button. Acquisition of targets is eased when the pointer approaches from above or below. However, when approaching a target from the side, the expansion and contraction of neighboring icons creates a significant sideways motion, shifting the target's position and making it more difficult to acquire (Figure 4.2 C). This problem is also present in the Mac OS X dock.

4.2.2 Overlapping Buttons

To avoid sideways shifting of the buttons, an alternative strategy is to allow *limited* overlap between neighbouring buttons. We have built a second prototype [45] that implements this idea (Figure 4.3). The occlusion created by overlap can interfere with inspection and selection of some targets, however, use of transparency and appropriate icon design could both reduce this problem. In addition, we adopted two additional techniques to minimize the interference caused by overlap. First, our design guarantees that no button is occluded more than a given percentage, the *Max Occlusion* factor, that can be tuned to adjust behavior. Second, buttons that are occluded are always expanded at least enough so that their visible area is equal to their original unoccluded area. This ensures a rough lower bound on how difficult they are to see or acquire at any given time.

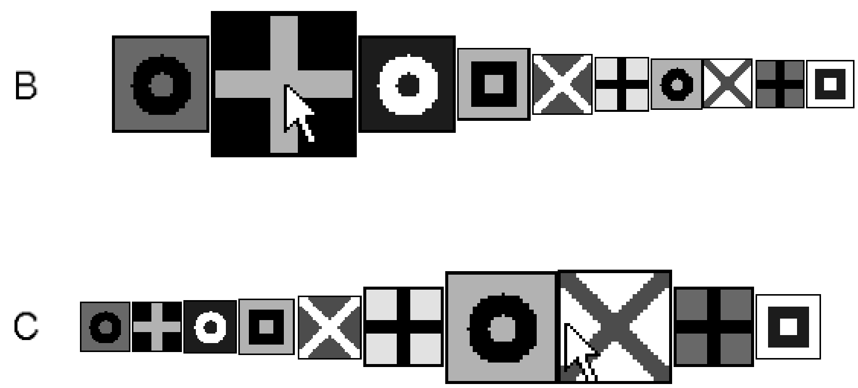

Figure 4.2: A design that roughly imitates the dock in Mac OS X. (A) The buttons are un-expanded when the pointer is far away. (B) A button is fully expanded when the cursor is over it, and neighboring buttons are partially expanded and pushed sideways. (C) A user starting in the state shown in (B) may try to move to the right to select the button with the light X on the dark background. By the time the cursor reaches the desired button's location, the button has moved to the left and the user is now over a different button (one with a *dark* X on a *light* background).

One consequence of this design is that, even with a *Max Occlusion* factor of 0 % (i.e. no occlusion allowed) which forces buttons to move sideways significantly, our design remains well-behaved in the sense that a fully expanded target will cover all the possible positions that its unexpanded self could appear in, thus reducing the possibility of incorrect selections.

Initial trials with the overlapping buttons design indicate that, with reasonable expansion factors (200-400 %), good values for the *Max Occlusion* factor fall between 20 and 50 %. We believe that this design is promising for one–dimensional arrays of widgets in that it allows for an adjustable trade-off between excessive sideways motion and mutual occlusion between targets.

Figure 4.3: In this design, limited overlap is allowed between adjacent buttons, which alleviates the problems caused by sideways motion in Figure 4.2. The *Max Occlusion* factor controls the amount of overlap between neighbouring buttons.

4.2.3 An Optimization Strategy: Shrinking Targets

Yet another approach is to recast the expansion problem as an optimization problem to be solved with calculus. One may reasonably suppose that, given any pointer location, we wish to configure the buttons such that the total index of difficulty for the buttons is minimized.

Let N be the number of buttons, each centred at c_i and with width W_i, where $1 \le i \le N$. Assume the buttons must tile the $[0, 1]$ interval (so $0 < c_1 < \ldots < c_N < 1$, and $W_1 + \ldots + W_N = 1$). If the pointer is located at $x \in [0, 1]$, then the index of difficulty ID_i for the ith button is $\log_2(|x - c_i|/W_i + 1)$. Thus, we seek the set of c_i and W_i that minimize

$$\sum_{i=1}^{N} ID_i = \sum_{i=1}^{N} \log_2(\frac{|x - c_i|}{W_i} + 1) \tag{4.1}$$

(Note that the Shannon formulation of the index of difficulty is used here to avoid a singularity when $x = c_i$ for some i).

Oddly, this optimization actually causes the nearest targets to *shrink*. An approximation of the resulting interaction is sketched in Figure 4.4. It is interesting that the visual behaviour of this design appears to be opposite of that in Figure 4.2. Although it is based on a strict (but naive) notion of mathematical optimization, it is probably the least viable design in this chapter, because it makes it difficult for the user to see a target just before they click on it.

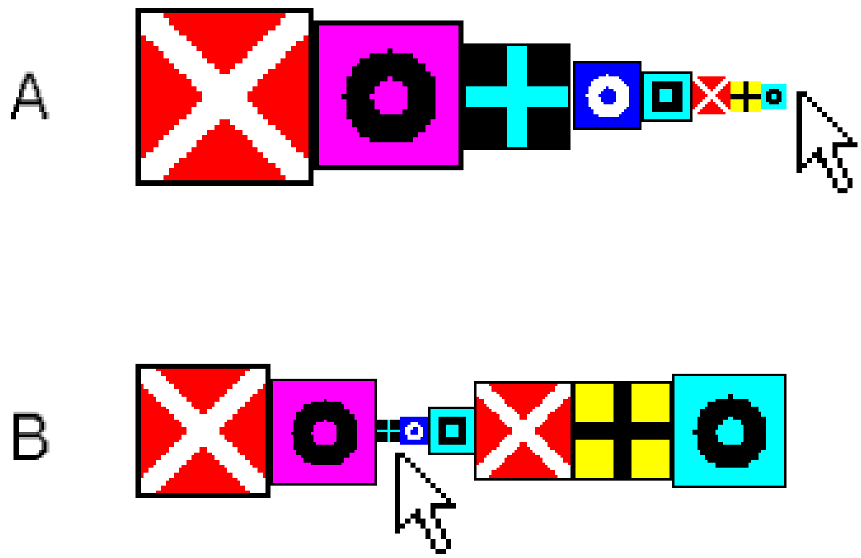

Figure 4.4: In this scheme, at any given moment, the sum of the indices of difficulty is a minimum. Interestingly, this causes nearby targets to shrink.

4.2.4 The Bad News

A problem common to the last three designs is that, in each case, expansion depends solely on the current position of the mouse cursor. As with the untiled design of Section 4.1, this means that, in the motor domain, the buttons have a fixed size. In Section 4.1, the buttons in the motor domain are considerable larger than their visual, unexpanded versions, affording the user a corresponding advantage for selection. However, in the last three designs, such a large advantage may not be possible.

Take the overlapping buttons design for example. In Figure 4.3, the cursor is over a button, and this button *looks* expanded. However, the full width of the expanded button is not available to the user: as soon as the user moves off the centre of the button, it starts to contract.

Figure 4.5 shows the rectangular region that the pointer must be within to acquire the same button. This rectangle is in fact what the button looks like in motor space. So, vertically, the button is larger than it looks when un-expanded. However, horizontally, there is no expansion in the motor domain. In fact, the buttons in all of the last three designs are rectangles of the same size in motor space. So, with respect to motor space, the shrinking targets design is just as good as the overlapping targets design.

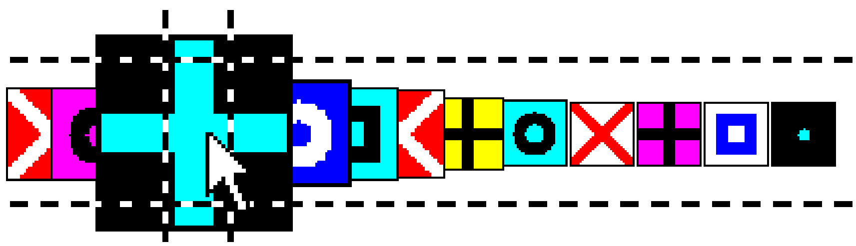

Figure 4.5: Dashed lines delimit a rectangle which the pointer must be within to acquire the button. Although the button looks larger than this rectangle, its full size is not available to the user: as soon as the pointer moves off the button's centre, the button begins to contract.

This presents some problems. First, since these rectangles are no wider than the original buttons, we should expect them to be no easier to select when approaching from the side. (Fortunately, since the rectangles are taller than the original buttons, given the results our experiment we expect them to be easier to select when approaching from above or below. As we know from MacKenzie [43], for the purposes of Fitts' law, the "size" of a rectangular target can be computed as the size measured along the axis of motion, i.e. the height of a rectangle when approaching from above or below.) Second, because there is no motor domain expansion *along the axis of tiling*, the schemes presented so far would suffer if extended to two-dimensional arrays of widgets. In such an array, where the tiling is along both axes, there would be no expansion at all in the motor domain.

On the bright side, two avenues still show promise. Although none of the designs presented achieve horizontal expansion in the motor domain, the visual feedback of having buttons expand may in fact make them easier to acquire (perhaps by making it easier for the user to see when they're over a desired target). Our current knowledge of Fitts' law cannot tell us if "apparent expansion" or other forms of visual pop-out make targets easier to acquire — only future experimental work can address this question.

Second, it is possible to design expansion schemes that do not depend solely on the pointer's current location, and thus possibly achieve true expansion in the motor domain. This thread is explored in the next section.

4.3 Tiled Targets with Motor Domain Expansion

The designs in this section attempt to achieve true (horizontal) expansion in the motor domain. The expansion in these schemes is a function not only of the pointer's current location, but also of the pointer's history. First, a "drifting buttons" design is considered, in which the expansion at any given moment is a function of the previous moment's expansion and the current pointer location. Next, schemes are presented that involve prediction of the pointer's future position. The notion of optimization with calculus is also revisited.

4.3.1 Drifting Buttons

As we have seen, a major problem with the Mac OS X dock and the design in Figure 4.2 is the the buttons shift horizontally when approached from the side. Why not simply ensure that this shifting never occurs ? Given the current state of the buttons, and knowledge of which button the user is aiming for, we can expand the desired button around its *current* centre (Figure 4.6).

Note that this introduces a kind of *hysteresis* into the interface, since the current pointer position alone does not fully determine the state of the buttons. Unfortunately, there is potential for the whole array of buttons to drift to the side as successive buttons are targeted (compare Figure 4.6 A and D). The drifting motion vaguely resembles the

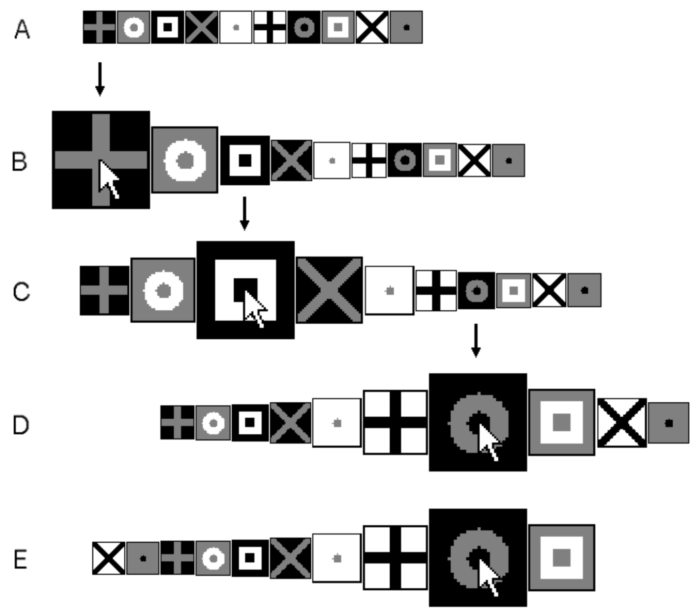

Figure 4.6: A shows the buttons at rest. B, C and D show successive expansions of buttons, where the expansion (indicated by black downward pointing arrows) occurs around the target button's *current* centre. Unfortunately, in this case the sequence of chosen targets causes the entire array of buttons to gradually drift to the right. E shows a possible remedy for the drifting: the 2 right-most buttons have been "wrapped around" to keep the array from moving too far to the right.

successive contracting and stretching of an earth worm or a caterpillar.

In addition, we require an accurate method for anticipating which target the user is aiming for. Ideally, some kind of trajectory prediction would be performed. A simpler method is to use the proximity of the cursor to guess which button the user is aiming for; however, this would increase the number of false predictions, and aggravate drifting when the user moves sideways through the array.

One possible solution for eliminating drift is to "wrap" the buttons around if they go too far sideways (Figure 4.6 E). Unfortunately, there is the possibility that the user could be chasing a target only to have it jump to the opposite end of the array. It might also be disturbing to the user to have the ordering of buttons constantly changing, since this would work against habituation and memory of where buttons are located.

4.3.2 Expansion with a Fixed Edge

A basic tradeoff we have encountered has been that the advantage of expanding one target is somehow offset by having to move, contract, or occlude other targets. One might reasonably wonder whether this tradeoff will always hinder our efforts: it may be the case that, when averaged over many selections, the net gain in performance of tiled, expanding targets is always zero. After all, we are in some sense trying to "beat" Fitts' law, or more precisely, trying to exceed the index of performance $1/b$ of static targets. Given the theoretical and experimental robustness of Fitts' law, this may be too much to ask. On the other hand, a net performance gain may turn out to be possible, in which case there is likely a "sweet spot" within the tradeoff where this gain is maximized.

Before these possibilities can be explored, we must first quantify the tradeoff. In what follows, such a quantification will be given (for a simplified model of tiled, expanding targets), and this will give evidence that a net gain is possible, and also motivate another design scheme.

Consider a tiled set of buttons, each of width W. Assume that, during each selection by the user, trajectory prediction (based on the pointer's history) is used to estimate which button the user is aiming for. The predicted button, which may or may not be the one the user is really aiming for, is expanded by a magnification factor M, and neighbouring buttons are occluded without being moved. Assume further that the expansion occurs toward the end of the user's motion, precisely when the user has "homed in" on the desired target to within a diameter of $2W$ (Figure 4.7 A), and that the predicted target remains expanded until completion of the selection (this ensures true expansion in the motor domain), even if the prediction was wrong.

The moment the user is within a diameter of $2W$ of the desired target, there is only 1 bit left to "transmit" to complete the selection, since (assuming the targets don't expand) the user need only reduce the "noise" in the cursor's position by a factor of 2. If the prediction system were perfect and always estimated correctly, and the desired target

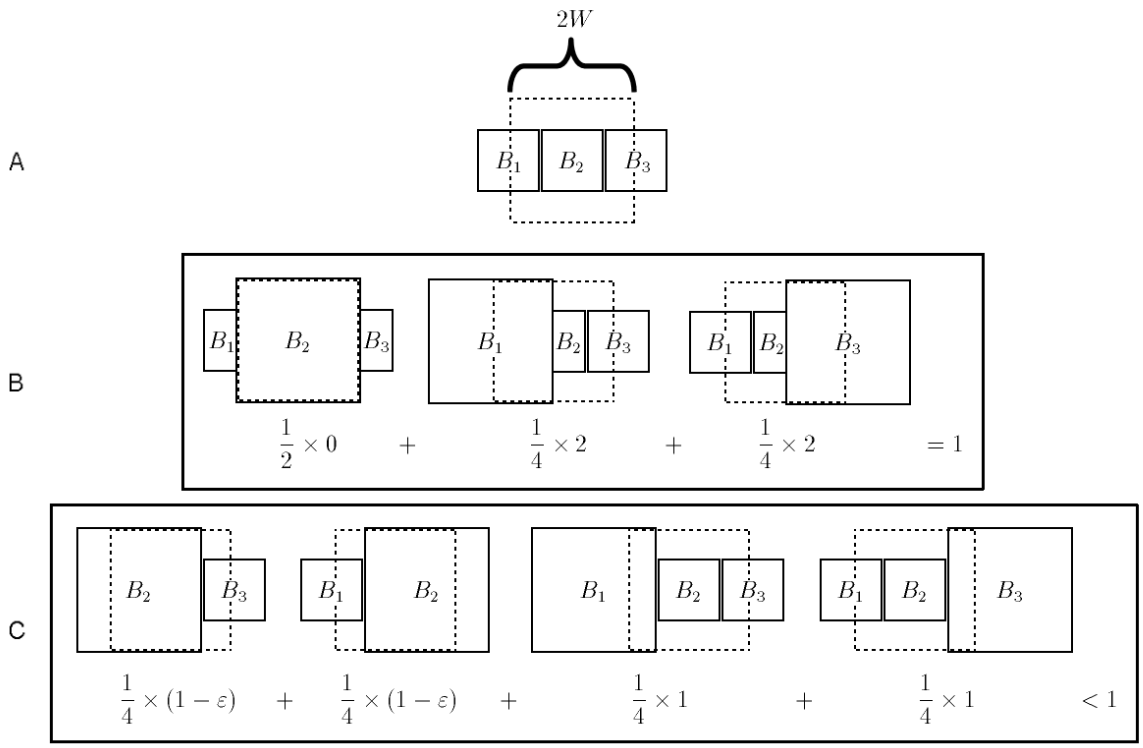

Figure 4.7: Of the three buttons B_1, B_2, B_3, each with width W, assume the user wishes to acquire B_2. A: Towards the end of the selection, the user comes within a diameter of $2W$ of B_2. At this point, there should only be 1 bit of information left to transmit. In B and C, we compute the bits necessary to complete the selection, given two different expansion schemes where targets expand by a factor of 2. B: The predicted target expands around its centre. Half the time we expect the system to predict correctly, in which case the user will already be over B_2 (so zero bits are left to transmit). The rest of the time, the system is wrong, requiring the user to transmit 2 more bits to acquire B_2 which is now half its normal size. On average, the user must transmit 1 bit. C: The predicted target expands around the edge closest to the predicted final pointer location. When the system is correct, we expect slightly less than 1 bit to be required to complete the selection, since B_2 is now twice its original size (though its centre is also displaced). When the system is wrong, only 1 bit is needed to acquire B_2, since it is undisturbed. On average, then, the user needs to transmit less than 1 bit to complete the selection.

were expanded by a factor of $M \geq 2$, then the pointer would already be over the desired target without any further motion, and the user would not have to bother transmitting the last bit.

Instead of a perfect prediction system, we will assume that the predicted target point is distributed uniformly within the diameter of $2W$ around the desired button. In this case, we should expect the system to be wrong half of the time (expanding the left or right neighbour of the desired button, one quarter of the time each), and correct the other half of the time.

Figure 4.7 B illustrates this for a system where the predicted button is expanded around its centre by a factor of $M = 2$. When the system is correct, the "cost" (i.e. the number of bits the user must still transmit) is zero. However, when the system is wrong, the cost is 2 bits, because the desired button is now half occluded (meaning the user must aim for a target half its original size). The net or average cost, then, is 1 bit, which is no better than the cost involved in static targets.

In this particular case, however, it turns out that the cost changes if we allow M to vary. When the prediction is correct, the cost to the user is $\log_2 2/M$ (i.e. 1 bit for each factor of 2 by which the user must "home in"). On the other hand, when the system is wrong, the cost is $\log_2 2/(1-(M-1)/2)$ (this approaches infinity as M approaches 3, which makes sense since a button expanded by a factor of 3 will completely occlude its two neighbours). The sum of these, weighted by their probabilities, yields the net cost of

$$\frac{1}{2}\log_2\frac{2}{M} + \frac{1}{2}\log_2\frac{2}{1-\frac{1}{2}(M-1)} = \frac{3}{2} - \log_2\sqrt{M(3-M)} \tag{4.2}$$

which is 1 when $M = 1$ (i.e. no expansion) and when $M = 2$, but is minimized when $M = 3/2$, where its value is approximately 0.915 bits, a net gain over static targets !

At this point, there are many criticisms that could be made of our assumptions. For example, it is unlikely that the distribution of predicted target points should be uniform over the diameter of $2W$ around the desired button. Also, the notion that the expansion should occur exactly when the user is within a diameter of $2W$ of the *desired* button may seem implausible, since the system cannot know with certainty which is the desired button ! We have also ignored any reaction time that might be required by the user to assess which target has expanded and adjust their motion correspondingly (In the experiment described in the previous chapter, users were able to take full advantage of the expanded size of the target, but this was for an isolated target that expanded in a predictable manner. In the present scenario, the user cannot be sure which target will expand as they home in.) Nonetheless, we feel that a set of simplifying assumptions is necessary to render tiled, expanding targets amenable to quantified analysis. The proposed set of assumptions may not be the best, but they do at least provide evidence that a net gain over static targets is possible.

Furthermore, although our analysis suggests that a net gain is possible, it is not a very large gain (a difference of less than 0.1 bits). A second expansion scheme is thus proposed, which attempts to reduce the cost associated with incorrect predictions.

In this new scheme, the predicted button is not expanded around its centre, but rather around its edge *closest to the predicted target point*. For example, if the system predicts that the pointer will land within the right half of button B at the end of the motion, then B is expanded around its right edge, meaning that the neighbour B^r to the right of B is not occluded at all. If it turns out that the prediction is incorrect, then the user was most likely really aiming for B^r, which will then only cost 1 bit to acquire.

The key notion here is that the edge between the two most likely buttons remains *fixed* during expansion, so that there is no penalty (beyond the normal 1 bit) to acquire the second most likely button.

Figure 4.7 C breaks down the net cost of this scheme given our previous assumptions, with $M = 2$. When the prediction by the system is correct (which should happen half of the time), it is not clear what the cost to the user is, since the desired target will grow to twice its size but also have its centre slightly displaced. We suspect, however, that the cost should be less than 1 bit by at least a small amount ε. As shown in the figure, the net cost is therefore less than 1 bit.

Since the value of ε is unknown, we cannot find the optimal value for M as we did for the previous scheme. It would appear, however, that the "fixed edge" expansion scheme reduces the penalty for incorrect predictions, and outperforms the previous scheme at least when $M = 2$. We can also suggest further refinements to the fixed edge expansion scheme.

First, since a real prediction system would probably not be as good as the one assumed here, we suggest disallowing any occlusion. For example, if button B is expanded around its right edge, then its left neighbour B^l (and all subsequent neighbours to the left) should be pushed to the left, so that no buttons are occluded. Second, it may be advantageous to also expand the right neighbour B^r, since it is the second most likely button desired by the user. If B^r is expanded, then *its* right neighbour B^{rr} (and all subsequent neighbours to the right) should also be moved to the right rather than occluded (Figure 4.8). Of course, if it turns out that the user was really aiming for some button other than B or B^r, there will be a penalty associated with having moved the desired button sideways. However, the cost of acquiring the two most likely buttons B and B^r should be less than 1 bit. Although we do not have enough information to compute optimal expansion factors, this scheme seems to show more promise than expansion around a button centre, since it reduces the cost of the *two* most likely target buttons rather than just the one.

4.3.3 Prediction and Optimization

Although it does not always lead to elegant solutions, an aggressive, brute force approach to problem solving can at least be informative. For the next and final design scheme of this chapter, we will try to "throw everything we've got" at the problem. An extension of the standard index of difficulty will be given, which can be minimized to (hopefully) optimize the sizes and arrangement of tiled buttons at any given moment.

Earlier, in the shrinking targets design, we proposed minimizing the sum of indices of

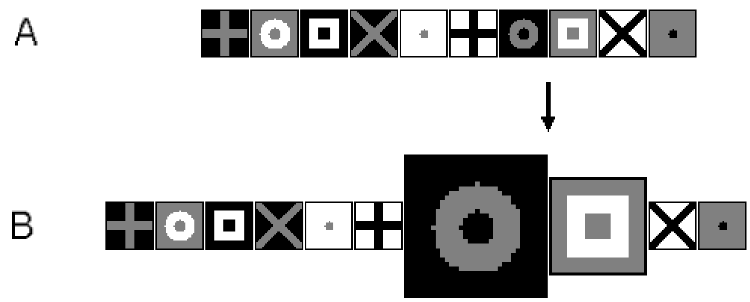

Figure 4.8: In this expansion scheme, the edge closest to the predicted target point is held fixed, and the two most likely target buttons are expanded around this edge. All other buttons are moved sideways. If the expansion factors involved are not too extreme, the advantage afforded by the expanded buttons should, on average, outweigh the penalty of an incorrect prediction. A: The buttons at rest. B: Expansion resulting from a prediction that the pointer's trajectory is heading for the right half of the 4th button from the right.

difficulty (Equation 4.1). As with the other schemes of Section 4.2, the shrinking targets scheme depends only on the current pointer position, and thus there is no expansion in the motor domain. This motivates two questions: (i) are there ways of expressing the difficulty of acquiring a button that are *independent* of the current pointer location, and (ii) if we are unable to achieve expansion in the motor domain, what is the optimal size and arrangement of fixed buttons in the motor domain (e.g. should the buttons all be of the same size, or otherwise) ?

Both of these questions can be answered to some degree by introducing a new metric of target difficulty. We propose *integrating* the index of difficulty over all possible pointer positions, yielding what we term an *integrated index of difficulty* or IID. A sum of such IIDs provides us with an overall measure of the difficulty of all targets involved, and the targets can then be adjusted so as to minimize this quantity.

The notion of integrating Fitts' index of difficulty has been explored before. Accot's steering law [1, 2] was derived by integration. Conceptually, Accot and Zhai decomposed the task of steering through a tunnel into an infinite sequence of infinitesimal goal passing tasks [1]. The integral thus gives a measure of the total difficulty of a *single* task: that of navigating the tunnel. What we propose, however, is to integrate the index of difficulty over different mouse positions, to measure the overall difficulty of *many* selection tasks.

Let N be the number of buttons in a linear strip that tile the $[0, 1]$ interval, with edges at $x_0, x_1, \ldots, x_N$, where $0 = x_0 < x_1 < \ldots < x_N = 1$. The ith button thus has width $W_i = x_i - x_{i-1}$ and centre $c_i = (x_{i-1} + x_i)/2$, where $1 \leq i \leq N$. If the pointer is located at $x \in [0, 1]$, then the index of difficulty ID_i for the ith button is $\log_2(|x - c_i|/W_i + 1)$. If we limit our attention to pointer locations within the $[0, 1]$ interval, then the integrated index of difficulty for the ith button is

$$IID_i = \int_0^1 ID_i dx = \int_0^1 \log_2(\frac{|x - c_i|}{W_i} + 1)dx \tag{4.3}$$

(Once again, the Shannon formulation of the index of difficulty is used to avoid a singularity when $x = c_i$ for some i). We can thus define an overall measure of the difficulty of the button strip as

$$\sum_{i=1}^{N} IID_i = \sum_{i=1}^{N} \int_0^1 \log_2(\frac{|x - c_i|}{W_i} + 1)dx \tag{4.4}$$

As luck would have it, there is an analytic form for the integral on the right hand side of Equation 4.4. The resulting expression can be written in many different ways:

$$\begin{aligned}
&\sum_{i=1}^{N} IID_i \\
&= \sum_{i=1}^{N} \frac{1}{\ln 2} \left[(c_i + W_i)\ln(c_i + W_i) + (1 - c_i + W_i)\ln(1 - c_i + W_i) - (2W_i + 1)\ln(W_i) - 1\right] \\
&= \sum_{i=1}^{N} \frac{1}{\ln 2} \left[(c_i + W_i)\ln\left(\frac{c_i}{W_i} + 1\right) + (1 - c_i + W_i)\ln\left(\frac{1 - c_i}{W_i} + 1\right) - 1\right] &\quad (4.5) \\
&= \sum_{i=1}^{N} W_i \left[\left(\frac{c_i}{W_i} + 1\right)\log_2\left(\frac{c_i}{W_i} + 1\right) + \left(\frac{1 - c_i}{W_i} + 1\right)\log_2\left(\frac{1 - c_i}{W_i} + 1\right) - \frac{1}{W_i \ln 2}\right] &\quad (4.6)
\end{aligned}$$

Equation 4.6 has an intriguing right hand side, as it contains clear remnants of the index of difficulty, and the common terms in it hint at how to compute the sum of IIDs efficiently. However, if we wish to reduce the number of free variables by rewriting W_i and c_i in terms of x_i and x_{i-1}, then the simpler Equation 4.5 becomes

$$\begin{aligned}
\sum_{i=1}^{N} IID_i &= \sum_{i=1}^{N} \frac{1}{2\ln 2} \left[(3x_i - x_{i-1})\ln\left(\frac{3x_i - x_{i-1}}{2(x_i - x_{i-1})}\right)\right. \\
&\qquad \left. + (x_i - 3x_{i-1} + 2)\ln\left(\frac{x_i - 3x_{i-1} + 2}{2(x_i - x_{i-1})}\right) - 2\right]
\end{aligned} \tag{4.7}$$

Or goal, then, is to find values of $x_0, x_1, \ldots, x_N$ that minimize 4.7, subject to the constraint that $0 = x_0 < x_1 < \ldots < x_N = 1$. This is therefore an $(N-1)$-variate minimization problem.

A natural question to ask is: if the buttons all have equal width $1/N$ (i.e. $x_i = i/N$), does this minimize the sum of IIDs ? A simple experiment was performed using the symbolic algebra package Maple [75] with $N = 3$. Equation 4.7 was found to be minimized when the edges

$$(x_0, x_1, x_2, x_3) \approx (0, 0.3623122390, 0.6376877610, 1)$$

In other words, with three buttons, the buttons at the ends should be approximately 32 % wider than the middle button to minimize the sum of IIDs. A plausible explanation for this is that, over most of the $[0, 1]$ interval, the pointer is farther from the end buttons than from the middle button. Hence, the end buttons must be made slightly larger to compensate.

The same phenomenon is seen for larger N. Figure 4.9 shows the approximately optimal button arrangement for $N = 16$.

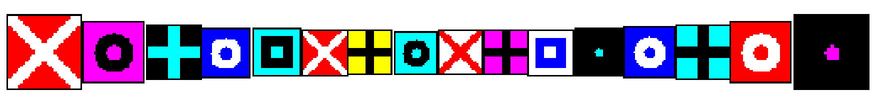

Figure 4.9: The widths and centres of these buttons minimize the sum of integrated indices of difficulty in Equation 4.4.

Although these results are interesting, it is not clear how relevant they are for real UI design. In our computation of the IIDs, we constrained the pointer to the $[0, 1]$ interval, but this is not reflective of many interfaces where the pointer can travel far away from a button strip. Also, we do not seem to be any closer to achieving expansion of buttons in the motor domain.

A critical point to observe is that the sum of IIDs in Equation 4.4 expresses overall difficulty, but only in so far as all of the buttons are equally likely targets. Given information on the pointer's motion, there will usually be some buttons that are more likely targets than others. If we can compute, or at least estimate, the probability p_i that the ith button is desired by the user, we can weight the sum of IIDs by these probabilities thus:

$$\sum_{i=1}^{N} p_i IID_i = \sum_{i=1}^{N} \frac{p_i}{\ln 2} \left[(c_i + W_i) \ln\left(\frac{c_i}{W_i} + 1\right) + (1 - c_i + W_i) \ln\left(\frac{1-c_i}{W_i} + 1\right) - 1 \right] \quad (4.8)$$

and then find the $x_1, \ldots, x_{N-1}$ that minimize this new quantity. Over time, as the pointer is moved and the estimated probabilities $p_1, \ldots, p_N$ change, we can iteratively recompute the values of $x_1, \ldots, x_{N-1}$ to keep Equation 4.8 minimized.

A prototype interface that implements this idea was written in Java. Two steps are iterated: a predictive step, where the program estimates the probability that each button is the desired target, and an optimization step, where the program adjusts the buttons to minimize the weighted sum of IIDs.

During the predictive step, the last 3 known pointer positions are quadratically extrapolated to predict (assuming constant acceleration) the target point, i.e. the location the pointer will come to rest at. A Gaussian distribution is defined over the $[0, 1]$ interval that is centred at the predicted target point. The area under this curve delimited by each button is computed, and these areas are normalized to yield the probabilities $p_1, \ldots, p_N$ which sum to 1. Thus, buttons far from the predicted target point will have a smaller p_i associated with them. The spread, or standard deviation, of the Gaussian distribution can be adjusted interactively through a *Prediction Confidence* slider. A smaller confidence value means the Gaussian distribution is more flat, and the probabilities will be more uniform.

During the optimization step, the program makes an initial guess of $x_i = \sum_{j=1}^{i} p_j$ (i.e. the button's widths are proportional to their probabilities) and then uses a variant of gradient descent to minimize the sum of weighted IIDs. A fast, "conjugate gradient" minimization algorithm is described in Press et al.'s excellent reference [60, chapter 10]. For our purposes, however, a more modest algorithm is sufficient, since we only need to determine $x_1, \ldots, x_{N-1}$ to a precision of 1 pixel. Initially, naive gradient descent was used for minimization, which proved to be rather slow to converge. However, adding "momentum" to the algorithm (i.e. always stepping in the direction of an average of the gradient and the previously used direction) increased the speed of convergence by a factor of approximately 10.

Unfortunately, as the prediction changes from moment to moment, the optimization step can cause targets to shift sideways significantly. To reduce the effects of this, a third step was added after optimization. Rather than displaying the optimized buttons directly, a weighted averaging is done between the optimal $x_1, \ldots, x_{N-1}$ values and equally spaced edges at $1/N, 2/N, \ldots, (N-1)/N$. The button edges $x'_1, \ldots, x'_{N-1}$ used for display are thus defined as

$$x'_i = wx_i + (1-w)i/N$$

for $1 \leq i < N$, where w is the weight for the averaging, and is specified by the user through an *Effect of Optimization* slider.

Figure 4.10 shows some resulting arrangements of buttons for different values of the *Prediction Confidence* and *Effect of Optimization* sliders.

There remain many weaknesses with the current implementation of this scheme. The most significant is the poor performance of the prediction algorithm, which is due in large part to the low mouse sampling rate available through standard programming interfaces such as that provided by Java. Even with a higher sampling rate, however, it is not clear that the quadratic extrapolation algorithm currently used would be sufficient for good prediction. Ideally, if the user is travelling along a "steady" trajectory, the predicted final point should not change much, especially toward the end of the motion. This is the only way that the arrangement of buttons can stay steady toward the end of the motion, thereby achieving expansion in the motor domain. A quadratic extrapolation of the last 3 pointer positions may be too sensitive to small perturbations in the samples for our purposes. Other schemes for mouse trajectory prediction or target prediction (see §2.7) may be better.

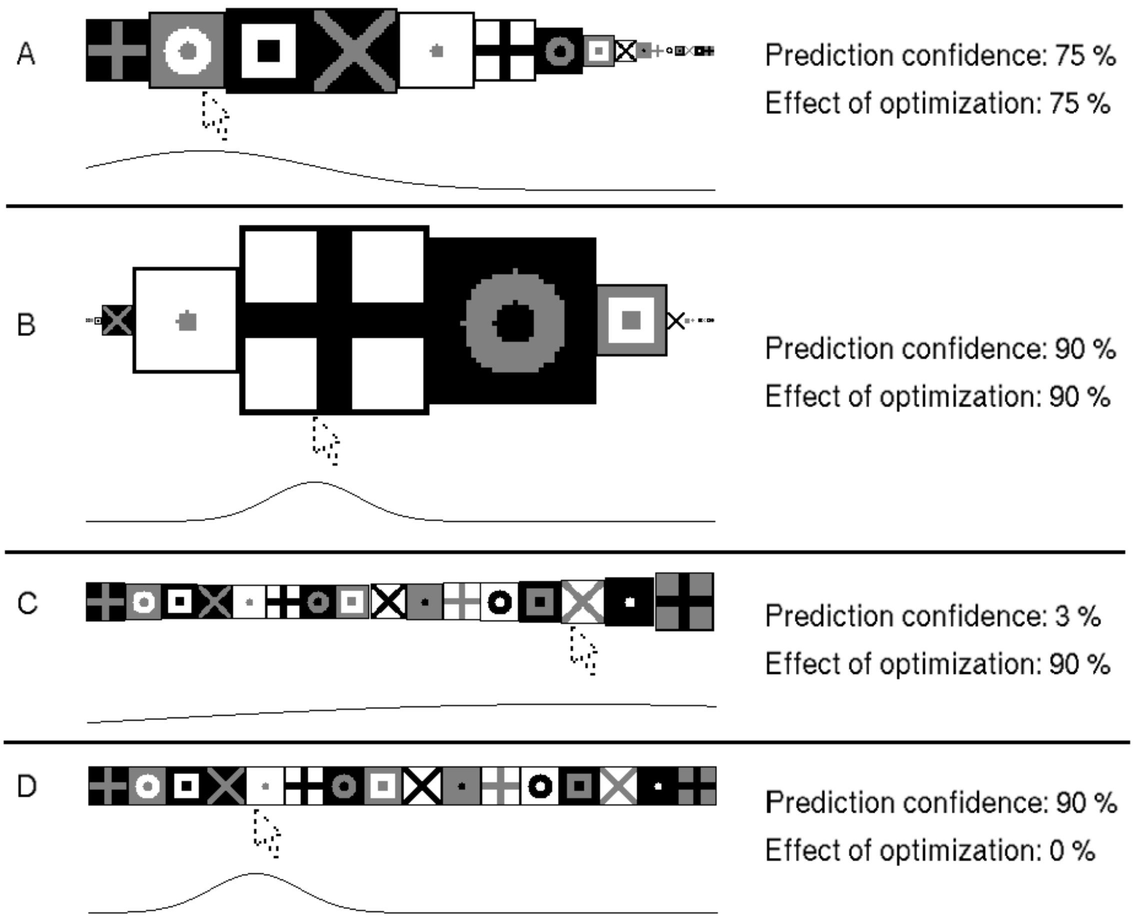

Figure 4.10: Results of the prediction and optimization scheme. The dashed outlines of cursors show the predicted target point (i.e. the predicted location the cursor will come to rest at). Under each button strip is a plot of the Gaussian distribution generated by the prediction step. The *Prediction Confidence* parameter controls the spread of the distribution; a small value (C) makes the distribution almost flat, causing the arrangement of buttons to approach that in Figure 4.9. The *Effect of Optimization* parameter controls interpolation between optimal edge values and equi-spaced edges; setting it to zero (D) gives all buttons the same width.

Second, as already pointed out, as the predicted final point changes, the optimization step can cause significant sideways motion of the buttons. Although this problem can be reduced by adjusting the *Effect of Optimization* slider, it seems wasteful and inelegant to perform all the work of a careful optimization, only to then "throw away" much of the information by averaging with equi-spaced edges. Perhaps, ultimately, the phrasing of the optimization problem should be changed, to take into account the cost of sideways shifting of button centres.

4.4 Ultimate Goals

As already stated, our goal with tiled, expanding targets is to, in some sense, "beat" Fitts' law, or at least beat the index of performance $1/b$. One of the foundations on which Fitts' law rests is the Shannon-Hartley theorem regarding maximum channel capacity. Section 2.2 demonstrated the close analogy between the Shannon-Hartley theorem and Fitts' law. In exceeding the index of performance $1/b$, are we not, by analogy, somehow violating the Shannon-Hartley theorem ?

In our (simplified) interpretation of the Shannon-Hartley theorem, recall that the time required to transmit one symbol is determined by the *bandwidth* of the channel, a physical limitation of the medium of communication. However, with Fitts' law, the time required to transmit one symbol is the time required to complete a physical motion. Normally, only the endpoint of the motion is considered useful information; however, the motion can be sampled throughout time, and the information present in the motion allows a predictor to anticipate, at least to some degree, where the endpoint will likely be. For example, if a pointer is located in the centre of a strip of buttons, and the user moves the pointer to the right, we know that the user is most likely *not* aiming for a button on the left end of the button strip.

Thus, by making use of the information available in the user's motion, it may be possible to exceed the index of performance $1/b$ without violating any known fundamental principles. The preceding design schemes have suggested ways of doing this, and of them the optimization schemes seem most promising, but there remain problems with them. Ultimately, to find the best design, a rephrasing of the optimization problem may be necessary. To this end, new visual representations of the problem may be instrumental in developing the right intuition and deeper understanding of the problem to solve.

Since Fitts' law involves space, time, and motion across space, it may be informative to look to tools used in physics for describing related problems. According to Einstein's Special Theory of Relativity [15, 61], the speed c of light is constant with respect to inertial frames of reference, and is an upper bound on the speed of all massive objects or particles. As a consequence, there is a large portion of spacetime that each of us can never hope to visit, since we would have to travel faster than light to get there. Light cones [32, 58] are a graphical representation of this: a person or other massive object located at the apex of the cone may only hope to visit the volume of spacetime within the cone (Figure 4.11). Furthermore, as a massive object travels through spacetime, the apex of the cone follows the object, forever further limiting what spacetime can be visited.

Since Fitts' law implies an upper bound on the speed at which aimed motions can be performed, there may be a surface in spacetime, analogous to the light cone, which bounds the volume that can be visited. However, whereas the spatial location of particles in Figure 4.11 are *exact points*, with Fitts' law we are concerned with the location *and precision* (i.e. width) of the final target. The current location of a pointer, then, can be thought of as a particle that is *smeared out* over space, having a central position and an associated width (or noise or distribution). As already mentioned in §2.3.2, Crossman seems to have already suggested a similar point of view [11] [12, p. 252]. Moving this particle across space and reducing the width of the particle each have a cost associated with them, as given by Fitts' law. Hence, it is proposed that in our diagram, one axis represent the current spatial position A, and a second axis represent the precision W of

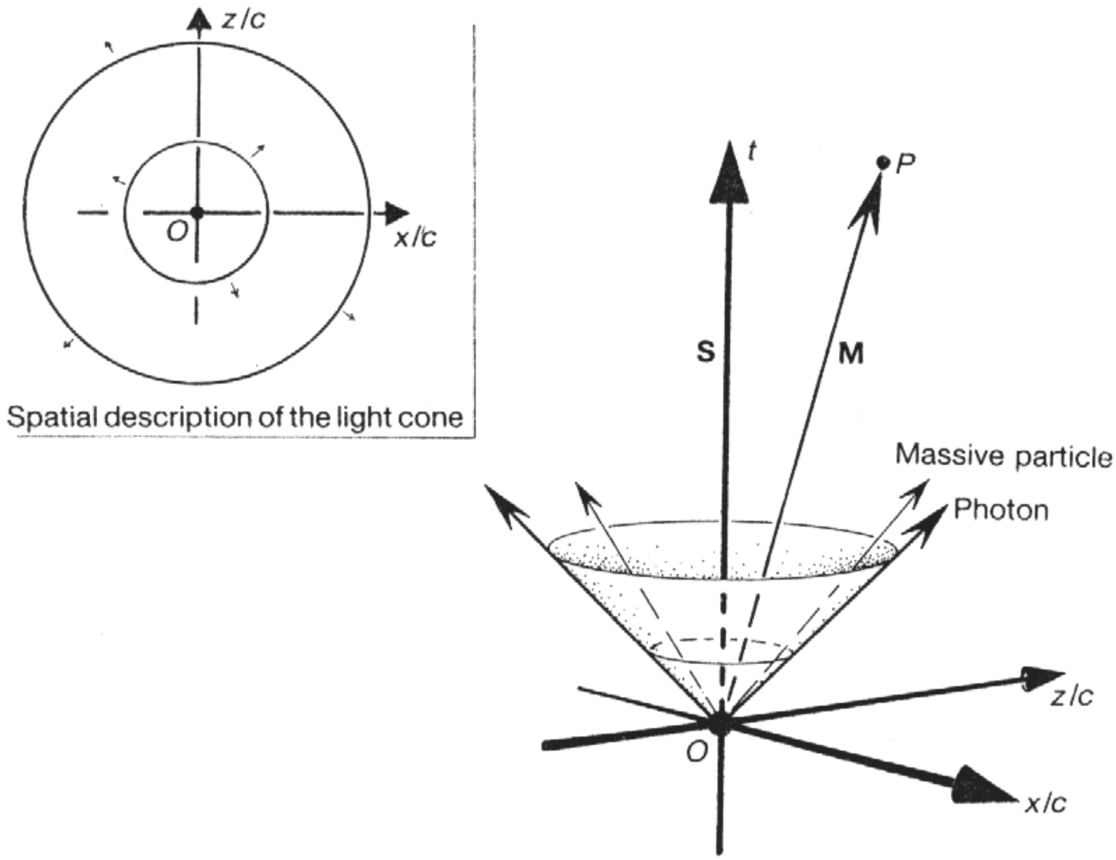

Figure 4.11: In this diagram, space is reduced to two dimensions (the axes on the horizontal plane) and time is the vertical axis. The light cone describes the history of a flash of light originating at the origin O. The spatial axes are normalized so that the cone has a slope of 1. A massive particle at the origin may only travel within the volume bounded by the cone, since travelling outside would require it to exceed the speed of light. (Figure reproduced from Penrose [58].)

this position. Finally, a third axis denotes time T. The surface

$$T = \log_2\left(\frac{A}{W} + 1\right)$$

is plotted in Figure 4.12, and looks like a kind of “wedge” in spacetime. It is suggested that, like the light cone, this wedge shows us what portions of spacetime can be possibly visited, and even follows the user as the user travels through spacetime. A target, such as a button to click on, corresponds to a vertical line with a given location and precision (width). The time required to acquire such a target is given by the intersection of the vertical line with the wedge surface. Expanding and/or translating this target in the motor domain corresponds to a horizontal translation of the line in the spacetime diagram.

It may be useful to think about tiled, expanding targets as multiple, vertical lines in spacetime, and to think of the optimization problem as a minimization of the time coordinates of intersections between these lines and the wedge surface. It is unclear at present how much extra intuition this may afford us, however the potential remains.

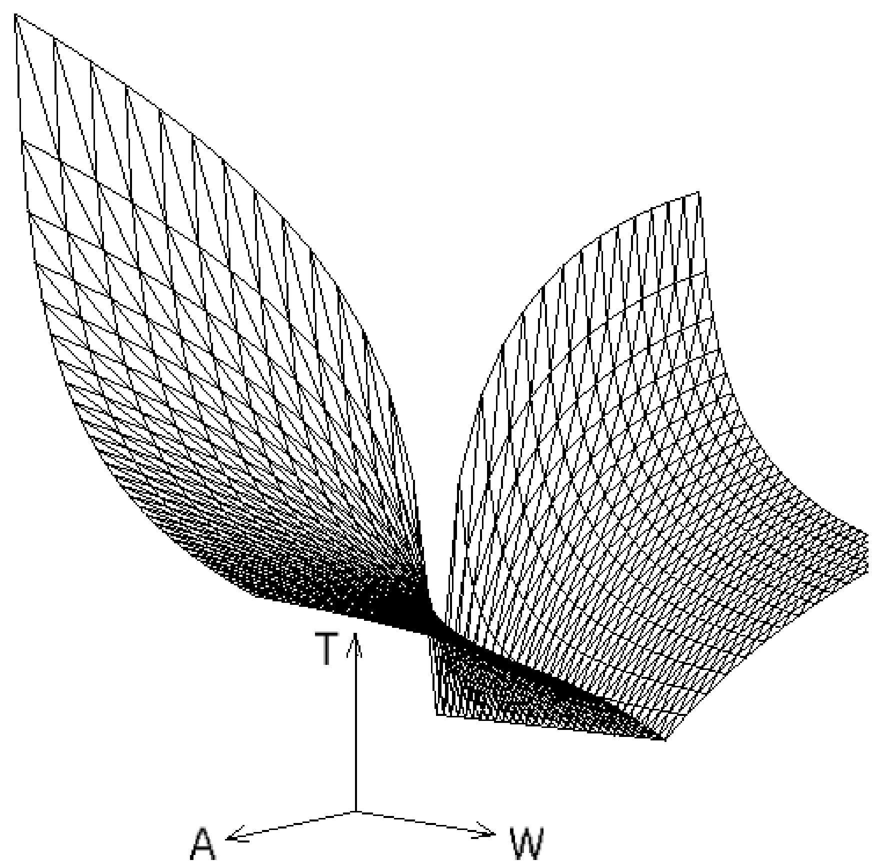

Figure 4.12: In this diagram, space is reduced to 1 dimension but is described by two axes: one for location A, and one for the precision W of the location. The vertical axis is time. The wedge surface is described by Fitts' law, and bounds the region of spacetime that can be visited by a user starting at the origin.

Chapter 5

Conclusions and Future Directions

5.1 Conclusions

We have presented experimental work that investigates parameters and performance of expanding targets. Our results indicate that the acquisition of a single expanding target can be accurately modeled with Fitts' law, and that the acquisition time is dependent on the target's *final* size, not its initial size. Furthermore, the gain in performance is approximately as much as one could expect, given Fitts' law: a target that expands just as the user is about to reach it (even after 90 % of the distance toward it has been traversed) can be acquired approximately as fast as a target that is always in an expanded state. This means that we can roughly predict performance with an expanding target using Fitts' law and a base set of data where no expansion takes place. It also indicates that we can take advantage of the benefits of expanding targets without any loss in performance.

We have also discussed the implications of these results for the design of user interface widgets that dynamically change in size to aid selection. Unfortunately, interfaces that incorporate multiple expanding targets in close proximity present various problems. Care must be taken to mitigate the potential collisions or overlapping that can occur between widgets during expansion. It is not clear which design scheme is best, however we have identified some subtle pitfalls and offered promising solutions (§4.2.2, §4.3.2, §4.3.3).

5.2 Contributions

From a motor control perspective, our experimental results corroborate the *optimized initial impulse model* [47] of rapid, aimed motion, and also indicate that the corrective movements toward the end of the motion can be made to take advantage of last moment changes in target size.

With regard to Fitts' law, our work can be added to the long list of previous studies that have tested Fitts' law under different conditions or increased our knowledge of its applicability. Our contribution has been to study targets that change size. There is also a potential that the thought experiments in Section 2.3 and the spacetime diagram in Section 4.4 may eventually help researchers to develop a better intuition and understanding of Fitts' law.

In the domain of user interface design, our work offers mathematical evidence (§4.3.2) that an appropriately designed strip of tiled, expanding buttons can exceed the performance of standard, static targets. Loosely speaking, it may be possible to beat Fitts' law and to exceed the index of performance normally associated with a set of static buttons.

Within the field of human computer interaction, there are few quantitative tools available to researchers (this is partly a consequence of studying humans, but also because the design aspect of human computer interaction is open ended). Despite efforts to "harden" the field, Fitts' law remains one of the few robust models available. Our work has demonstrated how, within a narrow problem domain, design problems can be recast as optimization problems in calculus and solved mathematically. Although the assumptions and approximations used in such work can at times yield misleading results, these can be subsequently tested with experiments. We therefore hope that mathematical optimization may play a role in future models of interaction for theoretical design work. More generally, we also hope that more mathematical tools may serve as "tools of thought", as demonstrated in this thesis.

5.3 Future Directions

There are many aspects of an expanding target operating in isolation that could yet be explored.

First, the data collected in our experiment could be re-examined to compute the effective target width [41, pp. 106–109] used by users, to see if they truly took advantage of the spatially enlarged targets. If they did not, it may be that the visual pop-out effect of having a target expand simply made it easier for the user to see when they were over the target.

Second, more data could be collected to help develop a more precise mathematical model of performance with an expanding target. As suggested by Figure 3.4, the ID may have an effect on the user's ability to react to an expanded size.

Third, data could be collected with different $W_{expanded}/W$ ratios. (In our experiment, the $W_{expanded}/W$ ratio was always 2.) In particular, it would be interesting to test the condition $W_{expanded}/W < 1$, to see if the effects of *shrinking* targets mirror the effects of expanding targets.

Fourth, the overlapping buttons design scheme (§4.2.2) shows some promise, but does not achieve horizontal expansion in the motor domain. Nevertheless, it may be that the visual feedback of having buttons appear to expand helps the user to select them more rapidly when approaching from the side. (The potential benefit of such visual popping is also suggested by Figure 3.3, where we see that, for large ID values, the performance of expanding targets appears to exceed the theoretical bound.) Thus, an experiment could be performed with a single target where expansion occurs in the visual domain but not in the motor domain. This would enable a more definitive evaluation of the overlapping buttons design scheme.

Fifth, we are unaware of any work on moving targets where the target starts moving *after* the user has started to move toward the target. Single target experiments could be performed to develop a model of the penalty incurred from such target movement, and would help inform better mathematical analyses of the fixed edge expansion (§4.3.2) and prediction + optimization (§4.3.3) schemes.

With multiple expanding targets, there are also many open avenues.

First, it is plausible that having multiple expanding targets on a screen may cause distraction and reduce performance. Experiments with multiple targets could be designed to test for this, and (ideally) determine conditions where the distraction is eliminated, or at least minimized.

Second, the mathematical evidence that tiled expanding targets can exceed the performance of static targets must be verified experimentally. A simple experiment to test the evidence of §4.3.2 could be performed that would not require any prediction algorithm. It would involve 3 buttons (such as those in Figure 4.7 B), where the user must repeatedly select the middle button, and for each selection a randomly chosen button expands by a factor of either 1, 1.5, or 2. The evidence presented in §4.3.2 indicates that, if the middle button is expanded half of the time, then the expansion factors of 1 and 2 should result in the same average selection time, whereas the expansion factor of 1.5 should result in a net reduction of average selection time. Since this experiment would involve no actual prediction, it would be easy to implement.

Third, the most promising expanding target schemes are probably the fixed edge expansion (§4.3.2) and prediction + optimization (§4.3.3) schemes. The former, being simpler, may be easier to implement in a robust fashion; however it is not clear which may ultimately achieve better performance. These schemes should be implemented and tested experimentally to see if they can indeed be operated faster than static targets.

Fourth, there is still potential to develop a better mathematical framework within which to think about tiled, expanding targets, and to find a design that is optimal in terms of average movement time. Our prediction + optimization scheme optimizes in terms of integrated indices of difficulty, but does not take into account the cost of sideways motion of targets. It may be that better metrics of overall difficulty (e.g. metrics that take into account the cost of changing the current configuration of buttons) can yet be developed, and then minimized.

Bibliography

[1] Johnny Accot and Shumin Zhai. Beyond Fitts' law: Models for trajectory-based hci tasks. In *Proceedings of ACM CHI 1997 Conference on Human Factors in Computing Systems*, pages 295–302, 1997.

[2] Johnny Accot and Shumin Zhai. Performance evaluation of input devices in trajectory-based tasks: An application of the steering law. In *Proceedings of ACM CHI 1999 Conference on Human Factors in Computing Systems*, pages 466–472, 1999.

[3] Johnny Accot and Shumin Zhai. More than dotting the i's — Foundations for crossing-based interfaces. In *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, pages 73–80, 2002.

[4] Apple Computer, Inc. The "Dock", a feature of the "Mac OS X" operating system. http://www.apple.com/macosx/theater/dock.html.

[5] Jonathan Baldwin, Anup Basu, and Hong Zhang. Predictive windows for delay compensation in telepresence applications. In *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, pages 2884–2889, 1998.

[6] Jonathan Baldwin, Anup Basu, and Hong Zhang. Panoramic video with predictive windows for telepresence applications. In *Proceedings of the 1999 IEEE International Conference on Robotics & Automation*, pages 1922–1927, 1999.

[7] Ben Bederson. Fisheye menus. In *Proceedings of ACM UIST 2000 Symposium on User Interface Software and Technology*, pages 217–225, 2000.

[8] J. S. Brown and A. T. Slater-Hammel. Discrete movements in the horizontal plane as a function of their length and direction. *Journal of Experimental Psychology*, 39:84–95, 1949.

[9] Stuart K. Card, William K. English, and Betty J. Burr. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics*, 21(8):601–613, 1978.

[10] M. Sheelagh T. Carpendale and Catherine Montagnese. A framework for unifying presentation space. In *Proceedings of ACM UIST 2001 Symposium on User Interface Software and Technology*, pages 61–70, 2001.

[11] E. R. F. W. Crossman. *The Measurement of Perceptual Load in Manual Operations*. PhD thesis, Birmingham University, 1956.

[12] E. R. F. W. Crossman and P. J. Goodeve. Feedback control of hand-movement and Fitts' law. *Quarterly Journal of Experimental Psychology*, 35A:251–278, 1983. (Original work presented at the meeting of the Experimental Psychology Society, Oxford, England, July 1963).

[13] Jack Tigh Dennerlein and M.C. Yang. Haptic force-feedback devices for the office computer: Performance and musculoskeletal loading issues. *Human Factors*, 43(2):278–86, 2001.

[14] Sarah A. Douglas and Anant Kartik Mithal. *The ergonomics of computer pointing devices*. Springer Verlag, 1997.

[15] Albert Einstein. Elektrodynamik bewegter körper. *Annalen der Physik*, 17:891–921, 1905. (Translated into English and reprinted in H. A. Lorentz, A. Einstein, H.

Minkowski, H. Weyl, The Principle of Relativity: A Collection of Original Memoirs on the Special and General Theory of Relativity. Notes by A. Sommerfeld, 1st ed. 1923. New York: Dover Publications, Inc.).

[16] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, June 1954. (Reprinted in Journal of Experimental Psychology: General, 121(3):262–269, 1992).

[17] Paul M. Fitts and James R. Peterson. Information capacity of discrete motor responses. *Journal of Experimental Psychology*, 67(2):103–112, February 1964.

[18] George W. Furnas. Generalized fisheye views. In *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, pages 16–23, 1986.

[19] Yves Guiard, Michel Beaudouin-Lafon, and Denis Mottet. Navigation as multiscale pointing: Extending Fitts' model to very high precision tasks. In *Proceedings of ACM CHI'99 Conference on Human Factors in Computing Systems*, pages 450–457, 1999.

[20] Yves Guiard, Frederic Gourgeouis, Denis Mottet, and Michel Beaudouin-Lafon. Beyond the 10-bit barrier: Fitts' law in multi-scale electronic worlds. In *People and Computers XV Interaction without Frontiers: Joint proceedings of IHM 2001 and HCI 2001*, pages 573–587, 2001.

[21] Carl Gutwin. Improving focus targeting in interactive fisheye views. In *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, pages 267–274, 2002.

[22] L. D. Hartson. Contrasting approaches to the analysis of skilled movements. *J. gen. Psychol.*, 20:263–293, 1939.

[23] W. E. Hick. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4:11–26, 1952.

[24] Errol R. Hoffmann. Capture of moving targets: A modification of Fitts' law. *Ergonomics*, 34(2):211–220, 1991.

[25] Errol R. Hoffmann. A comparison of hand and foot movement times. *Ergonomics*, 34:397–406, 1991.

[26] Errol R. Hoffmann. Movement time to an array of controls. In *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting v.1*, pages 625–629, 1993.

[27] Errol R. Hoffmann. Optimum layout of an array of controls. *International Journal of Industrial Ergonomics*, 14(3):251–261, October 1994.

[28] R. Hyman. Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, 45:188–196, 1953.

[29] Richard J. Jagacinski and Donald L. Monk. Fitts' law in two dimensions with hand and head movements. *Journal of Motor Behavior*, 17(1):77–95, 1985.

[30] Richard J. Jagacinski, Daniel W. Repperger, Sharon L. Ward, and Martin S. Moran. A test of Fitts' law with moving targets. *Human Factors*, 22(2):225–233, 1980.

[31] Marc Jeannerod. *The Neural and Behavioural Organization of Goal-Directed Movements*. Oxford: Clarendon Press, 1988.

[32] William J. Kaufmann, III. *Black Holes and Warped Spacetime*. W. H. Freeman and Company, 1979. Chapter 6: Exotic Properties of Black Holes.

[33] T. Alan Keahey. Nonlinear magnification home page, 2000. http://www.cs.indiana.edu/hyplan/tkeahey/research/nlm/nlm.html.

[34] Steven W. Keele. Movement control in skilled motor performance. *Psychological Bulletin*, 70(6):387–403, December 1968.

[35] B. A. Kerr and Gary D. Langolf. Speed of aiming movements. *Quarterly Journal of Experimental Psychology*, 29:475–481, 1977.

[36] R. Kerr. Movement time in an underwater environment. *Journal of Motor Behavior*, 5:175–178, 1973.

[37] Gary D. Langolf, Don B. Chaffin, and James A. Foulke. An investigation of Fitts' Law using a wide range of movement amplitudes. *Journal of Motor Behavior*, 8:113–128, 1976.

[38] Gary D. Langolf and W. M. Hancock. Human performance times in microscope work. *AIIE Transactions*, 7(2):110–117, 1975.

[39] Daeyeol Lee, Nicholas Lindman Port, and Apostolos P. Georgopoulos. Manual interception of moving targets: II. Online control of overlapping submovements. *Experimental Brain Research*, 116(3):421–433, 1997.

[40] I. Scott MacKenzie. A note on the information-theoretic basis for Fitts' law. *Journal of Motor Behavior*, 21(3):323–330, 1989.

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

[42] I. Scott MacKenzie. Bibliography of Fitts' law research, 2002. http://www.yorku.ca/mack/RN-Fitts_bib.htm.

[43] I. Scott MacKenzie and William Arthur Stewart Buxton. Extending Fitts' law to two-dimensional tasks. In *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, pages 219–226, 1992.

[44] Jock D. Mackinlay, George G. Robertson, and Stuart K. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, pages 56–63, 1991.

[45] Michael McGuffin. Expanding target prototype designs, 2002. http://www.dgp.toronto.edu/m̃jmcguff/research/expandingTargets/.

[46] Michael McGuffin and Ravin Balakrishnan. Acquisition of expanding targets. In *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, pages 57–64, 2002.

[47] David E. Meyer, Richard A. Abrams, Sylvan Kornblum, Charles E. Wright, and J. E. Keith Smith. Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review*, 95(3):340–370, 1988.

[48] Tsutomu MIYASATO. Target prediction for pointing operation. *Information Processing Society of Japan (IPSJ) SIGNotes Computer Graphics and CAD (GC)*, (No.039–009), 1989. http://www.ipsj.or.jp/members/SIGNotes/Eng/10/1989/039/article009.html.

[49] Stefan Münch and Rüdiger Dillmann. Haptic output in multimodal user interfaces. In *Proceedings of ACM IUI'97 International Conference on Intelligent User Interfaces*, pages 105–112, 1997.

[50] Stefan Münch and M. Stangenberg. Intelligent control for haptic displays. *Computer Graphics forum*, 15(3):217–226, 1996.

[51] Tamara Munzner. H3: Laying out large directed graphs in 3D hyperbolic space. pages 2–10, 1997.

[52] Atsuo MURATA. Basic studies on human-computer interaction on target prediction in pointing by mouse and utilization of voice input. *Information Processing Society*

of Japan (IPSJ) SIGNotes Spoken Language Processing (SLP), (No.005–001), 1994. http://www.ipsj.or.jp/members/SIGNotes/Eng/21/1994/005/article001.html.

[53] Atsuo MURATA. Discussion on method for predicting targets in pointing by mouse. In *Proceedings of the Sixth International Conference on Human-Computer Interaction*, pages 719–724, 1995.

[54] Atsuo MURATA. Improvement of pointing time by predicting targets in pointing with a PC mouse. *International Journal of Human-Computer Interaction*, 10(1):23–32, 1998.

[55] Ian Oakley, Alison Adams, Stephen A. Brewster, and Philip D. Gray. Guidelines for the design of haptic widgets. In *Proceedings of British HCI 2002*, 2002.

[56] Ian Oakley, Stephen A. Brewster, and Philip D. Gray. Solving multi-target haptic problems in menu interaction. In *Extended Abstracts of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 2001.

[57] Peter Oel, Paul Schmidt, and Alfred Schmitt. Time prediction of mouse-based cursor movements. In *Proceedings of Joint AFIHM-BCS Conference on Human-Computer Interaction IHM-HCI 2001, Volume II*, pages 37–40, 2001.

[58] Sir Roger Penrose. *The Emperor's New Mind*. Oxford University Press and Penguin Books, 1989. Chapter 5: The classical world.

[59] Nicholas Lindman Port, Daeyeol Lee, Paul Dassonville, and Apostolos P. Georgopoulos. Manual interception of moving targets: I. Performance and movement initiation. *Experimental Brain Research*, 116(3):406–420, 1997.

[60] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.

[61] Robert Resnick and David Halliday. *Basic Concepts in Relativity and Early Quantum Theory*. Macmillan, 2nd edition, 1992.

[62] George G. Robertson and Jock D. Mackinlay. The document lens. In *Proceedings of ACM UIST'93 Symposium on User Interface Software and Technology*, pages 101–108, 1993.

[63] David A. Rosenbaum. *Human motor control*. Academic Press, San Diego, California, 1991.

[64] Richard A. Schmidt, Howard N. Zelaznik, Brian Hawkins, James S. Frank, and John T. Quinn, Jr. Motor-output variability: A theory for the accuracy of rapid motor acts. *Psychological Review*, 86(5):415–451, September 1979.

[65] Alfred Schmitt and Peter Oel. Calculation of totally optimized button configurations using Fitts' law. In *Proceedings of the HCI International '99 Conference on Human-Computer Interaction*, pages 392–396, 1999.

[66] L. V. Searle and F. V. Taylor. Studies in tracking behavior. I. Rate and time characteristics of simple corrective movements. *Journal of Experimental Psychology*, 38:615–631, 1948.

[67] Andrew Sears and Ben Shneiderman. Split menus: Effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 1(1):27–51, March 1994.

[68] Claude Elwood Shannon and Warren Weaver. *The Mathematical Theory of Communication*, chapter IV, page 67. The University of Illinois Press, Urbana, Illinois, 1949. Theorem 17.

[69] T. B. Sheridan and W. R. Ferrell. Remote manipulative control with transmission delay. *IEEE Transactions on Human Factors in Electronics*, 4:25–29, 1963.

[70] Koert M. van Mensvoort and Hilde Keuning van Oirschot. Cursor trajectory prediction using a genetic algorithm, 2001. http://www.koert.com/work/cursorpredictor/.

[71] Hilde Keuning van Oirschot and Adrian J. M. Houtsma. Cursor trajectory analysis. In *Haptic Human Computer Interaction Workshop*, pages 127–134, 2000.

[72] Hilde Keuning van Oirschot and Adrian J. M. Houtsma. Cursor displacement and velocity profiles for targets in various locations. In *Proceedings of Eurohaptics 2001*, pages 108–112, 2001.

[73] M. G. Wade, Karl Maxim Newell, and Stephen A. Wallace. Decision time and movement time as a function of response complexity in retarded persons. *American Journal of Mental Deficiency*, 83:135–144, 1978.

[74] Stephen A. Wallace, Karl Maxim Newell, and M. G. Wade. Decision and response times as a function of movement difficulty in preschool children. *Child Development*, 49:509–512, 1978.

[75] Waterloo Maple Inc. http://www.maplesoft.com/.

[76] A. T. Welford. The measurement of sensorimotor performance: Survey and reappraisal of twelve years progress. *Ergonomics*, 3:189–230, 1960.

[77] A. T. Welford. *Fundamentals of Skill*. Methuen, 1968.

[78] Alan M. Wing. Crossman and Goodeve (1963): Twenty years on. *Quarterly Journal of Experimental Psychology*, 35A:245–249, 1983.

[79] Robert S. Woodworth. The accuracy of voluntary movement. *Psychol. Monogr.*, 3, 1899.