Modeling Pointing at Targets of Arbitrary Shapes

Tovi Grossman, Nicholas Kong, Ravin Balakrishnan
Department of Computer Science
University of Toronto
www.dgp.toronto.edu
{tovi, kongn, ravin}@dgp.toronto.edu

ABSTRACT

We investigate pointing at graphical targets of arbitrary shapes. We first describe a previously proposed probabilistic Fitts' law model [7] which, unlike previous models that only account for rectangular targets, has the potential to handle arbitrary shapes. Three methods of defining the centers of arbitrarily shaped targets for use within the model are developed. We compare these methods of defining target centers, and validate the model using a pointing experiment in which the targets take on various shapes. Results show that the model can accurately account for the varying target shapes. We discuss the implications of our results to interface design.

Author Keywords
Fitts' law, probabilistic model, arbitrary shapes, pointing.

ACM Classification Keywords
H.5.2. User Interfaces: Graphical user interfaces

INTRODUCTION

In today's user interfaces, pointing to targets such as menus, buttons, and text is a fundamental operation. Fitts' law [4, 11] is commonly used to model movement times for such target acquisition tasks. It predicts the time *MT* taken to select a target of width *W* and distance (or amplitude) *A* from the cursor according to the equation:

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

where *a* and *b* are empirically determined constants. The logarithmic term is the index of difficulty (*ID*) of the task. Numerous studies have validated this model for one-dimensional pointing tasks (see the review by MacKenzie [11]). It has also been used to optimize the layout in new interface designs [21].

A limitation of Fitts' model, from the perspective of an interface designer, is that in its original form it is inherently one-dimensional. In contrast, most targeting tasks in user interfaces are typically two-dimensional. Thus researchers have extended Fitts' law to two dimensions [1, 7, 12]. Unfortunately, these extensions only handle bivariate targets. While this is adequate for modeling interface elements that consist of square, rectangular, or even circular shapes, there are important scenarios in which a designer may wish to model pointing to targets of arbitrary shapes.

First, it is becoming common that interfaces are designed with an importance placed on the visual aesthetics, and include many non-rectangular shapes. The advent of "skins" has made this especially popular (Figure 1a). Second, even if the visual appearance of a target is a square or rectangle, the motor space region, where the cursor can be positioned to select the target, may not be. For example, when using the bubble cursor [6], users can select targets' surrounding voronoi regions, which are arbitrarily shaped convex polygons (Figure 1b).

Thus, it would be useful if selection times for arbitrarily-shaped targets could be modeled, for the purposes of both the prediction and optimization of interface performance. A newly proposed probabilistic model [7] has the potential to be directly applied to such tasks. However, to date, the model has only been validated for rectangular targets.

In this paper, we first discuss the probabilistic model, and some issues that must be considered for applying it to non-rectangular targets. Most importantly we discuss several techniques for defining the center of non-rectangular targets. We then test these techniques and the model in an experiment where users point to various shaped targets. Our study shows that the probabilistic model can indeed be used to predict acquisition times for arbitrarily shaped targets. We conclude with a discussion of the implications of our findings to user interface design.

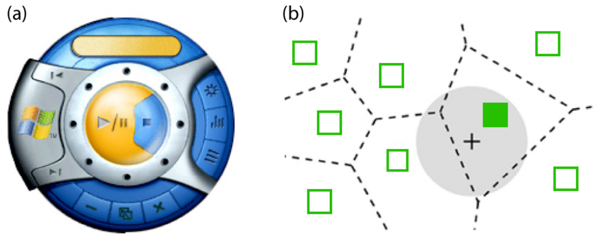

Figure 1. (a) A Windows Media Player skin contains a variety of non-rectangular shaped targets. (b) When selecting targets (shown in green) with the bubble cursor [6], the cursor only needs to be placed within the corresponding non-rectangular voronoi region (shown as dashed lines for illustration only).

RELATED WORK

When selecting rectangular targets on a two-dimensional screen, two factors must be considered beyond the target width and amplitude constraints of the one-dimensional Fitts' model (Equation 1). First, the selection is constrained by both the target's width and height. Second, the cursor must travel along a two-dimensional vector to the target.

MacKenzie and Buxton [12] conducted one of the earliest studies on bivariate pointing. They examined several formulas for the index of difficulty for a rectangular target, and found two which correlated with their experimental data. Their *W′* model (Equation 2) considers the apparent width *(W′*) of the target based on the movement angle. The *min* model (Equation 3) considers the minimum dimension. The relevant *ID*s are computed as follows:

$$ID_{W'} = \log_2\left(\frac{A}{W'}+1\right) \quad (2)$$

$$ID_{\min} = \log_2\left(\frac{A}{\min(W,H)}+1\right) \quad (3)$$

where *W* and *H* are the width and height of the target, respectively. The *min* model had the highest correlation with their experimental data. This model has been used in follow-up work [14, 18, 19], and was also proposed independently by Hoffman and Sheikh [10].

While the Mackenzie and Buxton models accounted for their data reasonably well, Accot and Zhai [1] identified various problems with the formulations. The problem with the *W′* model is that it completely ignores the constraints imposed by the dimension of the target perpendicular to the line of movement, also referred to as the directional constraint [1, 12]. The problem with the *min* model is that it predicts that *H* does not affect the time as soon as it becomes greater than *W*. Similarly, the model is not affected by *W* as soon as it is larger than *H*. Thus, this model does not account for data reported by Sheikh and Hoffman [16] that showed that it is harder to acquire a square than a rectangle with equal height but larger width.

Accot and Zhai [1] thus developed and validated a weighted Euclidean model which alleviates the drawbacks of the Mackenzie and Buxton formulations:

$$ID_{WtEuc} = \log_2\left(\sqrt{\left(\frac{A}{W}\right)^2+\eta\left(\frac{A}{H}\right)^2}+1\right) \quad (4)$$

where η is empirically determined. This model considers (*A/W*, *A/H*) to be the "constraint vector". By taking a weighted norm of this vector they incorporate both variables into an "appropriate distance in a two-dimensional space" [1]. The addition of the parameter η allows the model to weight the effect of the height differently from the effect of the width. The Euclidian model provides a significant improvement over the *min* model as it allows the larger dimension to still affect the movement time.

While an improvement over previous models, the Euclidian model still does not completely account for all pertinent factors of a general two-dimensional pointing task. First, it does not take into account the angle of movement towards the target. A two-dimensional pointing model should consider this factor, particularly since previous work [3, 8] indicates that the movement time will depend on the direction of movement. Second, and of particular interest to our current work, the model only considers rectangular targets. There is no clear way to apply the Euclidian model to targets that do not have well defined *W* and *H* values.

A more recent probabilistic model, developed by Grossman and Balakrishnan [7], addresses the two limitations of the Euclidian model. In particular, although not yet confirmed through experimental evaluation, the model can potentially be applied to targets of arbitrary shapes. In their studies, the probabilistic model was found to accurately model movement times when pointing at rectangles for various sizes, distances, and movement angles.

THE PROBABILISTIC MODEL

To help the reader understand the work done in this paper, we first give an overview of the previously proposed probabilistic model, and discuss how we will be applying it for modeling the selection of arbitrarily-shaped targets. We refer the reader to Grossman and Balakrishnan [7] for a full explanation of the probabilistic model.

The central idea behind the model is to map the probability of hitting a target with an open-loop movement to an index of difficulty value for that target. The "open-loop" movement refers to the initial ballistic impulse towards the target, without any feedback-guided final adjustments [11].

A dart board analogy can help explain the concept of a probability of hitting a target [7]. If a dart is thrown at the bull's eye of a dart board, it may land close to the bull's eye, or it may land further away. If a large number of darts are thrown, then a spread of hit locations will result. From this spread of hits, the probability of hitting the bull's eye, or any other target on the dart board, could be predicted. Similarly, if the spread of hits, *S*, when pointing at a graphical target, or region, *R*, using only ballistic and no corrective movement is well understood, then a probability of hitting that target without corrective movements, *P(hit)*, could be assigned. The probabilistic model maps this probability value directly to an index of difficulty value using a function F:

$$ID_{\mathrm{Pr}} = F\left(P_{R,S}(hit)\right) \quad (5)$$

The probabilistic index of difficulty is thus completely determined by calculating the probability that the target will be hit by an open-loop movement. This is an intuitive idea, since a target which is less likely to be hit by an open-loop movement should be harder to select. To determine the exact nature of the function *F,* which maps probability values to index of difficulty values, Grossman and

Balakrishnan examined the one-dimensional case. In this scenario, the spread of hits has been found to be normally distributed [20]. With a mean value of zero, corresponding to the center of the target, the index of difficulty becomes:

$$ID_{\mathrm{Pr}} = F\left(P\left(-\frac{W}{2} \le X_{N(0,\sigma)} \le \frac{W}{2}\right)\right) \tag{6}$$

where $X_{N(0,\sigma)}$ is a random variable in the normal distribution with mean 0 and standard deviation σ (Figure 2).

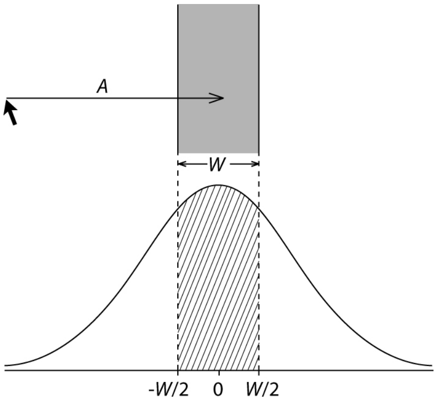

Figure 2. Target in relation to Normal distribution.

Further, the spread of hits, σ, increased linearly with the target amplitude A. This again is intuitive, since the spread of hits should diverge as the target distance increases. Reverting to the dart board analogy, the points that the darts hit will become more spread out if they are thrown from further away. The index of difficulty thus becomes:

$$ID_{\mathrm{Pr}} = F\left(P\left(-\frac{W}{2} \le X_{N(0,cA)} \le \frac{W}{2}\right)\right) \tag{7}$$

where c is an empirically determined constant, found to be 0.07 by Balakrishnan and Grossman [7].

To generate the function F, the well accepted values for the 1D Fitts' law index of difficulty are substituted in the left side of Equation 7, and the right side is stated in terms of the cumulative standard normal distribution Φ:

$$\log_2\left(\frac{A}{W}+1\right) = F\left(\phi\left(\frac{W}{2cA}\right) - \phi\left(-\frac{W}{2cA}\right)\right) \tag{8}$$

Unfortunately, there is no closed formula for the cumulative normal distribution function, so F is computed numerically, by simply substituting different values of A and W. Figure 3 shows the F function generated by substituting different values of A and W into both sides of Equation 8.

Using this function, the index of difficulty can be calculated for *any* pointing task, by only calculating the probability of hitting that target with an open loop movement. For two-dimensional pointing, the spread of hits is defined as the points $P = (X', Y')$, where X' is the error parallel to the line of movement and Y' is the error perpendicular to the line of movement (Figure 4). The spread is modeled by the bivariate normal distribution, $N(\mu_{X'},\mu_{Y'},\sigma_{X'},\sigma_{Y'},\rho_{X'Y'})$. The means $(\mu_{X'},\mu_{Y'})$, again, are zero, corresponding to the target center. The standard deviation $\sigma_{X'} = cA$, for some constant c, is measured from the center of the target, collinear with the direction of movement. The standard deviation $\sigma_{Y'} = dA$, for some constant d, is measured from the center of the target, perpendicular to the direction of movement. For simplicity, X' and Y' are assumed to be independent. The resulting equation for the bivariate normal density function is:

$$bndf(X',Y') = \frac{1}{cA\sqrt{2\pi}} e^{-\left(\frac{X'^2}{2(cA^2)}\right)} \frac{1}{dA\sqrt{2\pi}} e^{-\left(\frac{Y'^2}{2(dA^2)}\right)} \tag{9}$$

So, the probabilistic index of difficulty for two-dimensional pointing is:

$$ID_{\mathrm{Pr}} = F\left(\iint_R bndf(X',Y')\,dY'\,dX'\right) \tag{10}$$

where R is the target's region. The term inside the brackets is just the integral with respect to X' and Y' over the 2D region R. A desirable property of this model is that it does not require the target to have a well defined height and width. The equation can provides IDs for targets of arbitrary shape by just integrating over a different region R.

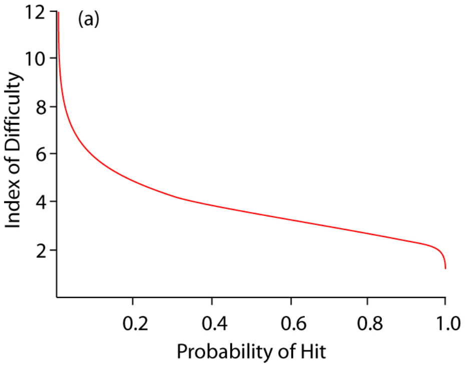

Figure 3. *F* function for c = 0.07.

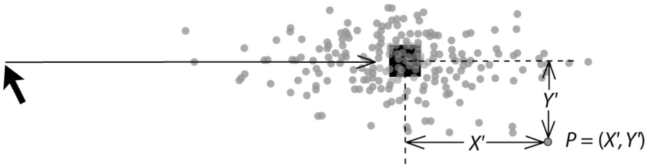

Figure 4. Example spread of hits in two-dimensional pointing. X′ is error parallel to, and Y′ is error perpendicular to, the line of movement.

The only issue remaining in applying the probabilistic model to arbitrary shapes is to determine what the "center" of the target is, or in other words, the target's area for which the bivariate normal distribution should be centered over. We discuss target center calculation in the next section.

CALCULATING THE TARGET CENTER

In the model, we calculate the probability of hitting a target using an open-loop ballistic movement. As such, we consider the target center to be the point on the target where we would expect the user to initially aim. The center of the distribution function which is to be integrated over is aligned with this target center (Figure 5a). Determining where the target center is will thus have an effect on the probabilistic model, as it will affect the probability, and in turn the index of difficulty, in two ways. First, it will determine how the distribution of hits overlaps the target. Second, it will affect the standard deviations of the distribution. This is because the standard deviations are a function of *A,* the distance to the target center. If the center is placed closer to the starting point of the cursor, then the distribution will become more compact (Figure 5b).

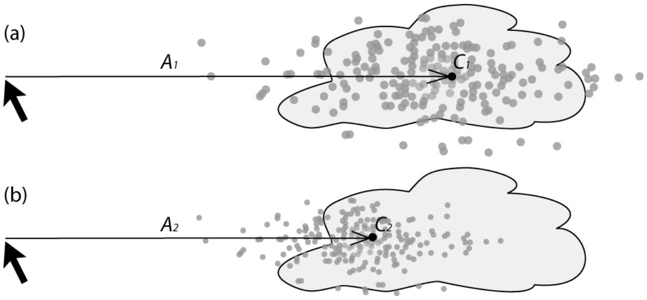

Figure 5. (a) The center of the distribution function is aligned with the center of the target, *C*$_1$. (b) Moving the center, *C*$_2$, closer to the starting point of the cursor reduces *A*, and shifts and compacts the distribution, affecting the probability.

Previously, the probabilistic model has only been used to model movement times to rectangular shapes, so the target center was easily defined as the center of the rectangle. This may in fact be a simplification, as there is no existing evidence showing that users initially aim for the center of rectangles. Regardless, if we are to use the probabilistic model for arbitrary shapes, we need a more formal method for defining the target center. In the following sections we present three methods for defining the target center, and provide an example of how the center is calculated for an example "crescent" shape which is used in our experiment.

Polygon Centroid

One possible method would be to use the mathematically defined polygon centroid, or "center of mass" of the 2D target. This method was suggested for triangular targets by Grossman and Balakrishnan [7].

The centroid of a polygon is calculated by taking an average of the N vertices in the polygon (x_i, y_i), i = 0, ... N-1, where the last vertex (x_N,y_N) is the same as the first, as the polygon is closed. Using a standard calculation of the polygon centroid, we first define the area of the polygon A:

$$A = \frac{1}{2}\sum_{i=0}^{N-1}\left(x_i y_{i+1} - x_{i+1} y_i\right) \tag{11}$$

We can then define the centroid point (*cx*, *cy*):

$$cx = \frac{1}{6A}\sum_{i=0}^{N-1}\left(x_i + x_{i+1}\right)\left(x_i y_{i+1} - x_{i+1} y_i\right) \tag{12}$$

$$cy = \frac{1}{6A}\sum_{i=0}^{N-1}\left(y_i + y_{i+1}\right)\left(x_i y_{i+1} - x_{i+1} y_i\right) \tag{13}$$

The advantage of this strategy is that the centroid is very simple to calculate. The disadvantage, however, is that applying this to arbitrary shapes may lead to undesirable results, as the defined centroid may be completely different from where a user would be expected to aim. For example, with a wide target, it may be expected that users would click towards the closest side of the target.

In Figure 6, we illustrate the vertices of our example crescent shape, which are used in the calculation for the target centroid. As can be seen, the resulting polygon centroid is horizontally centered. This will be the case for any target which is symmetrical about the vertical axis. Similarly, if the target were symmetrical about the horizontal axis, then the polygon center would be vertically centered. As for the vertical alignment on the centroid in this example, it appears to be much higher than where we would actually expect a user to aim. This demonstrates a possible disadvantage of using a strict mathematical definition of the target center.

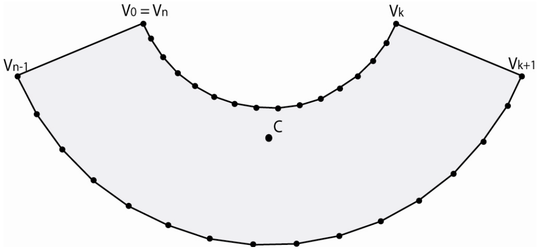

Figure 6. The vertices of the shape are used to calculate the centroid (C) of the polygon.

Selection Center

A potentially more accurate method for defining the target center would be to look at where users actually click when selecting the target. The distribution of these points could be averaged to determine where the center of the target should be set. The advantage of this technique is that it could account for effects such as the user tending to aim for portions of the target closer to the starting point of the cursor. The main disadvantage of this method is that the center of a target could only be computed after acquisition data for that particular target has been obtained. Another disadvantage is that it still may not be completely accurate as it is only estimating the final selection point, and not the initial point for which the user is aiming.

To illustrate this method, we provide example data of where a user's final selection points may occur when acquiring the "crescent" shape from the left (Figure 7). The data is from the experiment we present in the next section. Figure 7 also shows the resulting calculated selection center, obtained by averaging the position of the selection points. As can be seen, this method seems to provide a better estimate of where the user is aiming than the polygon centroid method.

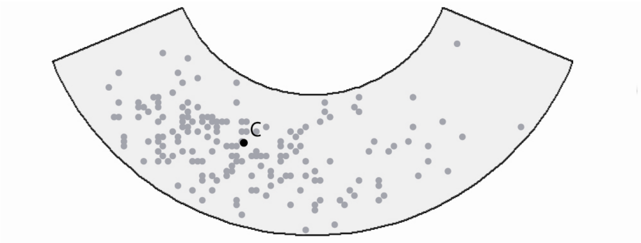

Figure 7. The selection center, C, of a shape is calculated by averaging the pre-captured selection points for that shape.

Optimized Center

While the selection center seems more suitable than the polygon centroid, it requires empirically measured data for the particular target of interest. It would be desirable to find a method which is as accurate, but does not require existing data for the calculation.

The final method we explore is to choose the center such that the probability of hitting the target is maximized. Humans are arguably quite good at optimizing their performance, so it may be reasonable to assume that users will initially aim for a point on the target which maximizes their chances of hitting anywhere on the target, with a single ballistic movement. To determine the center with this method, we set the probability of hitting the target with a ballistic movement as a function of the coordinates of the center point on the target, *cx* and *cy*. The center point is calculated by maximizing this function:

$$f(cx, cy) = P_{(cx,cy)}(hit) \qquad (14)$$

The main disadvantage of this method is that this function, along with its maximum value and corresponding values for *cx* and *cy*, can only be determined using computational methods. However our hope is that this method will provide results at least as accurate as the selection center, with the advantage that previously collected data is not required.

Figure 8 illustrates the calculation of the optimized center for our example crescent shape. The calculation assumes that the cursor is approaching from the left. In Figure 8a, we see a 3D plot of the function from Equation 14. The maximum point of this function can be determined using computational software such as Maple. The resulting value for the optimized center is illustrated in Figure 8b.

Even though the shape is symmetrical about the vertical axis, the function is slightly skewed towards the left, from where the cursor approaches. This is better illustrated in Figure 8b, where the function is mapped onto the actual shape. The gradient at each point represents the probability for that point, with lighter values representing higher probabilities. The optimized center is the lightest point on this shape, as it has the highest probability.

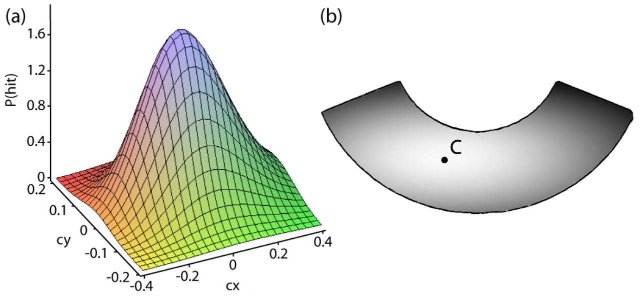

Figure 8. a) A 3D graph of the probability of hitting the target, as a function of the center position (cx, cy), which represents were the user is aiming. b) The optimized center, C, is defined as the maximum point on the function in a).

It is fairly intuitive that the function is skewed towards the left of the shape, as placing the center closer to the cursor will decrease the deviations of the distribution, increasing the probability of a hit, while placing it further away will increase the deviation of the distribution, reducing the probability of a hit (Figure 5).

EXPERIMENT

We conducted an experiment to test if the probabilistic model could accurately predict movement times when pointing at arbitrarily shaped targets. We also wished to evaluate our three methods for defining the target center when using the probabilistic model.

Apparatus

The experiment was conducted on a 1.4 Ghz PC running Windows XP with OpenGL for graphics, and a 19-inch monitor at 1280 by 1024 pixels. A puck on a WACOM tablet was used for input. The puck and tablet were used with an absolute mapping, to prevent clutching actions that might occur if a relative mapping or regular mouse was used. While such clutching actions do occur in practical settings, for the purposes of validating the pointing model, it is preferable to avoid potential confounds that might be introduced by clutching. Only a portion of the tablet was mapped to the screen space, to create a control-gain ratio of 2. This is the same hardware setup used by Grossman and Balakrishnan [7], allowing us to use their determined *c* and *d* values for the probabilistic model (Equation 9).

Participants

Ten unpaid volunteers (1 female, 9 male) participated in the experiment. Participants ranged in ages from 18 to 26, were all right-handed, and controlled the input device and consequently the cursor, with their right hands.

Procedure

A reciprocal 2D pointing task required participants to select two fixed-sized targets back and forth in succession. The target centroids were positioned equidistant from the centre of the display in opposite directions along the horizontal axis. This controlled the movement angle, simplifying our experimental design. The target to be selected was colored green, and the other red. When participants correctly selected a target, the targets would swap colors, as an indication that the participant had to now move to and select the other target. Participants had to successfully select the green target before the colors would swap, even if it required multiple clicks. This removes the possibility that participants may try to "race through the experiment by clicking anywhere". The total error rate was presented to participants after each set of reciprocal trials, and they were told to balance speed and accuracy such that their error rate remained at approximately 4% throughout the experiment.

Independent Variables

Target shapes

To test the probabilistic model when pointing at targets with arbitrary shapes, it is desirable to use a variety of different shapes, each with their own unique visual properties. To do so, we referred to computer vision literature in object classification. In that field, classification systems are created by giving the system a set of example shapes. Each example shape represents a unique class, and the classes are created such that given an arbitrary shape, a human would be able to distinguish which class it belongs to. We have chosen a set of 8 object classes, which have been commonly used as the example shapes for such object classification algorithms [2, 9]. Figure 9 illustrates the ten shapes which we included in our study based on these object classes. The shapes were mirrored on opposite sides of the display, with the target on the left side shown in Figure 9. Shapes 9 and 10 were included to mirror shapes 6 and 7, which were not symmetrical along the vertical axis.

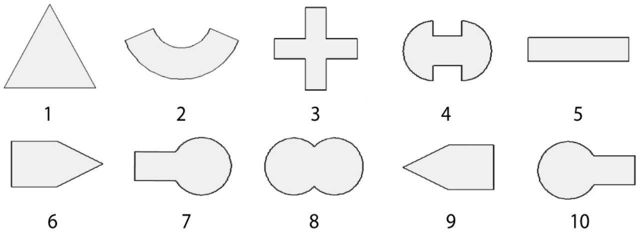

Figure 9. The ten shapes used in the experiment.

Target Sizes

Each target shape was presented at one of three sizes. At the smallest size (*Size* = 1), each target would fit snugly into a 37 (width) by 28 (height) pixel bounding box. Targets were also presented at two times (*Size* = 2), and three times (*Size* = 3) their smallest size. It is important to note that the purpose of this experiment is *not* to compare movement times across the ten shapes, nor to worry too much about the sizes of the targets as measured by exact area; it is only important to determine if the probabilistic model works across a spectrum of shapes. If it is shown that the model does work for arbitrary shapes, then the model could, in turn, be used to predict if a target of particular size and shape will be easier to select than another.

Target Distances

The final independent variable of the experiment is target distance, or amplitude. We used distances of 256, 512, and 1024 pixels, defined as the distance between the polygon centroids of the two reciprocal targets.

Design

A repeated measures within-participant design was used. The independent variables were *shape* (1-10), *size* (1, 2, 3), and *distance* (256, 512, 1024 pixels). A fully crossed design resulted in 90 combinations of *shape, size,* and *distance*.

Each participant performed the experiment in one session lasting approximately one hour. The session was broken up into 4 blocks, within which each of the 90 conditions occurred exactly once in random order. Each condition consisted of five reciprocal movements between the two targets, resulting in 1800 trials per subject.

To familiarize participants with the task and with the absolute mapping of the input device, participants were given a warm-up block before the actual experiment began.

Results

Movement Time

In our analysis of movement time we first removed trials in which errors occurred, 3.9% of the data. From this we then removed outliers more than 3 standard deviations from the group mean movement time, 0.8% of the data.

Repeated measures analysis of variance showed a main effect for target *shape* ($F_{9,81}$ = 214, p < .0001), *size* ($F_{2,18}$ = 6092, p < .0001), and *distance* ($F_{2,18}$ = 10934, p <.0001). The fact that distance and size had such strong effects is not surprising, and could have been predicted from the original formulation of Fitts' law [4] (Equation 1). What is more interesting is the effect of shape on movement time. This also could have been somewhat expected, as each unique shape should provide its own index of difficulty. This further validates our need for a predictive model which takes the shape of the target into account. Figure 10 shows the movement times for each of our ten shapes.

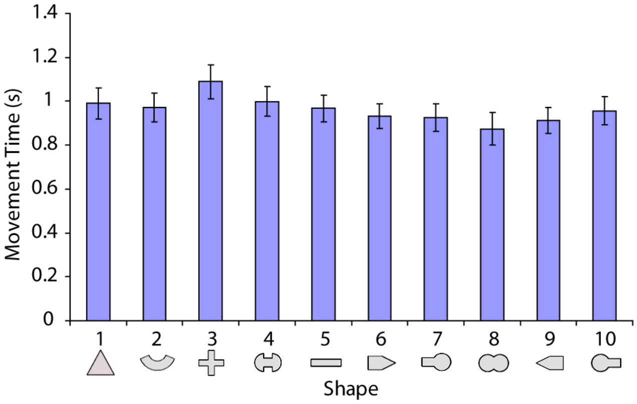

Figure 10. Movement times by shape, with 95% CI marked.

Although the goal of this experiment was not to compare movement times for the different shapes per se, there are some interesting observations which can be made.

Post hoc analysis shows that shape 3, the plus sign, had significantly higher movement times than any of the other shapes ($p < .0001$). This is despite the fact that its total surface area, or screen real estate which it covers, is comparable to many of the other shapes. This provides evidence that highly concave shapes which are segmented may be inefficient shapes for target acquisition.

It is also interesting to compare the results of shapes 6 and 9, and shapes 7 and 10, as they are both mirror images of each other. In both cases, movement times did not differ significantly. This demonstrates that for non-symmetrical targets, designers may not need to worry about which direction the cursor will be approaching from, as movement times will be similar. However, it should be noted that, although the differences were not significant, the targets with the larger portion closer to the starting point (Shapes 7 and 9) were slightly faster in both cases. This effect could be significant for targets whose asymmetry is more extreme.

While it is useful to make such observations, and hypothesize as to their possible causes, the probabilistic model should provide a theoretical reasoning for why they were observed. In particular, we will provide further discussion on why the plus sign had slower movement times when we discuss the modeling of the observed movement times.

Target Centers

Before evaluating the probabilistic model on our obtained data we first need to define the target centers, using the three methods discussed previously. Obviously, these centers must be calculated for each of our 10 shapes. Moreover, we must also calculate the target centers for each distance and size combination. While the *polygon centroid* for each shape will not change with the distance and size, the *selection center* and *optimized center* methods of calculation will likely result in different values depending on size and distance. We therefore calculated the target centers for each of the 90 conditions in our experiment design, using each of the three described methods. As an example, Figure 11 shows the target centers of each shape for distance = 512, size = 2.

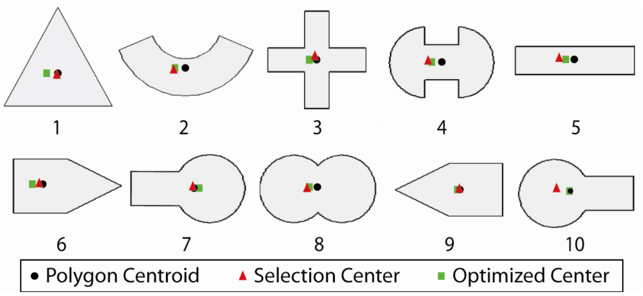

Figure 11. The calculated centers for each target, under the condition distance = 512, size = 2.

It is interesting to note that in all cases, the three methods for defining the center provide very similar results. In most cases, the *optimized center* is to the left of the *polygon centroid*, and very close to the *selection center*. The similarity between the selection and optimized centers demonstrates that users are good at optimizing their performance by aiming at an optimal location.

Modeling

Now that we have the target centers defined, we can use the probabilistic model to calculate the index of difficulty for each experimental condition. This calculation is done using Equation 10, where the region of integration is set to the target shape, which was offset by the appropriate center point. We used the c and d constant values ($c = 0.0717$, $d = 0.0284$) determined by Grossman and Balakrishnan [7].

Table 1 shows the results using a least-squares fit method for the three target center methods. The table shows parameter estimates and standard errors. The last column provides R^2 values for the regression. We see each of the models provide a high fit to the data, with R^2 values over 0.9. The *selection center* performed slightly worse than the other two models. The *optimized center* and *polygon centroid* performed best, both with R^2 values of 0.94. Figure 12 shows the regression analysis for the optimized center, for each of the shapes.

	A	*Std. Err.*	B	*Std. Err.*	R^2
Polygon Centroid	278.3	20.27	143.7	4.03	0.94
Selection Center	261.0	21.60	145.9	4.27	0.93
Optimized Center	301.3	18.60	140.7	3.75	0.94

Table 1. Fit of ID_{prob} to the experiment data, for the three methods of defining the target center.

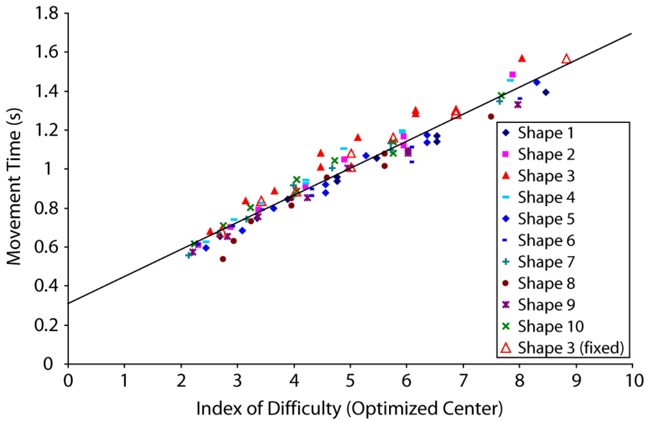

Figure 12. Movement time by the probabilistic index of difficulty for the optimized center method.

An important observation that can be made from this graph is that shape 3, the "plus sign", stands out, and is above the regression line in all conditions. This corresponds to the results reported earlier, where the "plus sign" was found to require significantly higher movement times. These results indicate that times for this shape are slighter higher than what the model would predict. This is in contrast to the rest of the shapes, for which the model seemed to work quite well. Indeed, when we remove the data for shape 3, the R^2 value for the optimized center goes up to 0.959. In light of this, we further analyzed the data for shape 3, to try to understand why the model was underestimating its movement times.

Our hypothesis was that because shape 3 appears to have two distinct sections, a horizontal and vertical rectangle, users would bias their movements towards the easier of these two sections, the horizontal rectangle [1]. To support this hypothesis, we looked at the selection points for all of the trials with shape 3. As seen in Figure 13, a large proportion of the selection points are within the bounds of the horizontal rectangle. In fact, more than 90% of the time, users were selecting within the region defined by this section of the target. In essence, in the majority of the trials, users were acting as if the rest of the target didn't even exist. This would explain why the model underestimated movement times for this shape, as the model integrates the distribution over the entire target region.

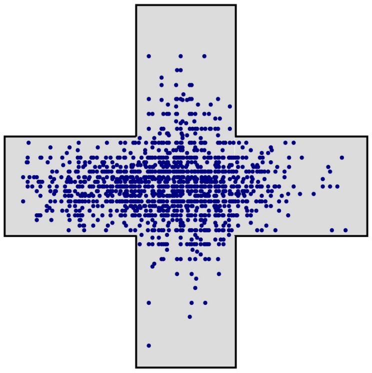

Figure 13. Selection data for the "plus" shape. Over 90% of selection points fall within the horizontal region of the target.

To further validate this belief, we repeated the calculation of the index of difficulty for shape 3, using only the horizontal rectangle as the region of integration. This resulted in a much tighter fit with the rest of the data, providing an overall R^2 value of .961. The result of this analysis is important, as it shows that users may not take full advantage of the screen real estate which a target with distinct segments takes up, making such a target inefficient.

Errors

The overall error rate across all trials was 3.9%, which is very close to the optimal 4% error rate value for target acquisition studies [11]. The shape had a significant effect on error rate ($F_{9,81} = 5.76$, $p < .0001$). Post hoc analysis showed that this effect was caused by shape 5, the rectangular shape, which had a significantly higher error rate (7.3%) than the rest of the shapes ($p < .05$).

DISCUSSION

The most important contribution of the presented work is that we have validated the probabilistic model for arbitrarily shaped targets. We selected a range of shapes, each with their own unique visual properties, and found the probabilistic model to accurately predict movement times, with R^2 values above 0.9 across all shapes.

Furthermore, we have proposed and tested three methods for defining the center of arbitrarily shaped targets. The *polygon centroid* is simple to compute, and did better than expected, with an R^2 value of 0.94. The *optimized center* method was found to be just as good for predicting movement times, and has the advantage over the *selection center* method of requiring no previous user data for the target of interest. If computational power is not an issue, we believe this method could be used with the probabilistic model, whereas the polygon centroid provides a simpler alternative with comparable results.

Aside from discussing the theoretical benefits of our model, it is also worth mentioning how our model compares to the Mackenzie and Buxton models from a practical standpoint (Equations 2 and 3) [12]. For some non-rectangular targets these models could potentially provide designers with simple and suitable estimates of movement times. However, the very definitions of these models tell us that they cannot accurately model pointing to *arbitrarily* shaped targets. For example, if modeling the plus sign shape, both the *MIN* and *W'* models would give this shape the same *ID* values as a 1D pointing task with the same horizontal width regardless of the thicknesses of the horizontal and vertical regions which the target consists of. Without even conducting an experiment, we can conclude than in many situations these models will predict movement times that have little or no correlation to actual movement times. As such, when considering non-rectangular targets, our model can provide sufficient benefit to be worthwhile for adoption in practice.

Given our results, it is now possible for interface designers to assess interface layouts which contain non-rectangular shapes. For example, we can apply the probabilistic model to the interface illustrated in Figure 1, to calculate index of difficulty values for each of the targets. To do so, we created a simple program which allows a user to sketch over regions of a bitmap image to define a target of interest. The program triangulates this region, and the resulting triangles are used as the region of integration for determining the index of difficulty. We used the *optimized center* method to calculate the indices of difficulty, and assumed the cursor approached each target from the left, from a distance of 640 pixels from the center of the interface (Figure 14a). In Figure 14b we see the resulting calculated index of difficulty for each of the targets.

Similarly, the model can be used to analyze the performance of selection techniques such as the bubble cursor, which increases the motor size of selection space to non-rectangular Voronoi regions [6]. In the initial study of this technique, the layout of targets was controlled, such that the resulting voronoi regions were square, as the authors wished to model the selection times. With the probabilistic model, such selection techniques can be studied with general non-rectangular layouts. Such analysis could provide insight on how selection techniques could be modified or enhanced.

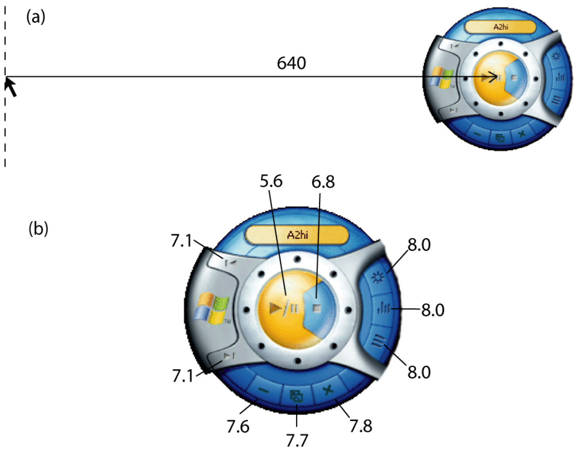

Figure 14. Calculating the indices of difficulty for an interface containing non-rectangular targets. a) We assume the cursor approaches each target from the left, from a distance of 640 pixels measured from the center of the interface. b) Each target is labeled with its corresponding index of difficulty, calculated using the probabilistic model.

Along with predicting user performance for interface layouts, designers can also optimize their layouts for user performance using the probabilistic model. For example, the Metropolis keyboard [21] could be further enhanced such that virtual keys are positioned *and* shaped in an optimal manner for user performance.

Another important finding from our results is that the model was not as accurate for the "plus sign" shape, as it underestimated movement times. Further analysis revealed that this was likely due to the clearly distinguishable segments in the plus sign. Users did not take full advantage of the entire surface area of the target, as they over-biased their movements towards the easier of these two segments. This observation tells us that highly concave, or segmented shapes, are probably not the most appropriate for interface targets. They will waste screen space as there will be parts of the target which will not be used for their selection.

NON RECTANGULAR USER INTERFACES

Our contributions will become increasingly important as the use of arbitrarily shaped targets in everyday interfaces increases. There is already a transition towards aesthetically pleasing interfaces, although this is at an early stage. In today's user interfaces, the majority of arbitrarily shaped targets are seen in skins and HTML image maps. However, they are also starting to occur in general program user interfaces. For example, along with designing skins, The Skins Factory (www.theskinsfactory.com) is also developing aesthetically pleasing application user interfaces which use arbitrary target shapes. We believe this trend will continue, as more importance is placed on the aesthetics of a user interface, and as the proper tools are available to interface designers for creating such interfaces. One justification for this belief is from an observation of this transition in the realm of user interface design of physical products. For example, if we examine the early generation of mobile phones, almost all of the buttons were rectangular (Figure 15a). In contrast, it is most certainly challenging to find a single rectangular button in the current generation of mobile phones (Figure 15b). This transition has also been seen in the automotive industry. In the early 1990's there was shift away from "boxy" shaped dashboards or control panels in cars. Today's car dashboards and their buttons and display panels tend to be composed of smooth flowing shapes, containing few sharp corners, if any.

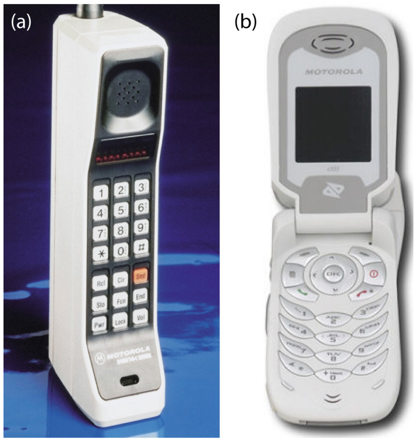

Figure 15. a) The Motorola International 3200, the first digital mobile phone, was introduced in 1992. It was composed entirely of rectangular buttons. b) The Motorola i450, released in 2005, does not contain a single rectangular button.

There are two possible reasons for this transition. First, when producing new technology such as cell phones, the majority of the attention in its production goes towards its technical implementation. Once this has been perfected, the attention will shift towards the aesthetics.

Second, the increasing sophistication of 2D and 3D graphic design and modeling tools has given industrial designers the ability to easily create technical drawings and 3D models composed of smooth curves and surfaces. Indeed, the prevalence of such tools has coincided in time with the shift towards more aesthetically pleasing designs.

We believe the majority of computer user interfaces are still in the early phase of this transition for the second of these two reasons. Designers and programmers at software companies generally use standard toolkits. These toolkits simplify the creation of interface elements, such as buttons and icons. However toolkits generally limit these elements to, square or rectangular shapes. When toolkits evolve to allow the use of arbitrarily shaped targets, it will be easier to create more aesthetically interesting user interfaces.

Of course, there is also an added cost of a more complex "hit test" when the targets are non-rectangular. In fact, in some examples of today's non rectangular interface targets, the system still assumes a rectangular bounding box for performing the hit test. However, with the speed of today's CPUs, a simple within-polygon algorithm would not cause any noticeable slow-down, and could be integrated into the toolkits which allow for the addition of arbitrary shapes.

There are also those who believe that interface designers who attend to the visual aesthetics of an interface may neglect its usability [15], or that the aesthetic elements themselves may degrade usability [5, 13]. Contrary to this, however, it has been shown that there is a correlation between perceived aesthetics and perceived ease of use of an interface [17]. Using the probabilistic model, designers now have the ability to predict movement times for their aesthetically designed non-rectangular shapes, ensuring that they are not degrading the interface's usability.

CONCLUSION

We have presented a formal experiment which validates the use of a probabilistic Fitts' law model for arbitrarily shaped targets. The model was found to accurately predict movement times across a range of non-rectangular shapes.

Furthermore, we discussed three methods for defining the center of arbitrarily shaped targets for the purposes of modeling, and found the *polygon centroid* and *optimized center* to be the most suitable techniques.

We also presented a discussion of the implications of our work towards user interface design. Interface designers can use the model to both predict and optimize performance of their user interfaces, and HCI researchers can use the model to analyze and improve interaction techniques.

Finally, although not the goal of our paper, we have provided some rationale as to why we believe arbitrarily shaped targets will become more prevalent in user interfaces. The work presented here will likely become all the more valuable when such a transition occurs.

REFERENCES

1. Accot, J. and Zhai, S. (2003). Refining Fitts' law models for bivariate pointing. *ACM CHI Conference on Human Factors in Computing Systems*. p. 193-200.
2. Bicego, M. and Murino, V. (2004). Investigating hidden Markov models' capabilities in 2D shape classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(2). p. 281-286.
3. Boritz, J., Booth, K. and Cowan, W. (1991). Fitts' law studies of directional mouse movement. *Graphics Interface*. p. 216-223.
4. Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *J. of Exp. Psychology*. 47. p. 381-391.
5. Foley, J., van Dam, A., Feiner, S. and Hughes, J. (1994). *Computer graphics: principles and practice (2nd ed.)*. Addison-Wesley. p. 1174.
6. Grossman, T. and Balakrishnan, R. (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *ACM CHI Conference on Human Factors in Computing Systems*. p. 281-290.
7. Grossman, T. and Balakrishnan, R. (2005). A probabilistic approach to modeling 2D pointing. *ACM Transactions on Computer-Human Interaction*. 12(3): p. 435-459.
8. Hancock, M. and Booth, K. (2004). Improving menu placement strategies for pen input. *Graphics Interface*. p. 221-230.
9. He, Y. and Kundu, A. (1991). 2-D shape classification using hidden Markov model. *IEEE Trans. Pattern Anal. Mach. Intell.* 13(11). p. 1172-1184.
10. Hoffmann, E. and Sheikh, I. (1994). Effect of varying target height in a Fitts movement task. *Ergonomics*. 37(6). p. 1071-1088.
11. MacKenzie, S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*. 7. p. 91-139.
12. MacKenzie, S. and Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. *ACM CHI Conference on Human Factors in Computing Systems*. p. 219-226.
13. Marcus, A. (1992). *Graphic design for electronic documents and user interfaces*. ACM Press. p. 288.
14. Murata, A. (1999). Extending effective target width in Fitts' law to a two-dimensional pointing task. *International Journal of Human-Computer Interaction*. 11(2). p. 137-152.
15. Nielsen, J. (1993). *Usability Engineering*. AP Professional.
16. Sheikh, I. and Hoffmann, E. (1994). Effect of target shape on movement time in a Fitts task. *Ergonomics*. 37(9). p. 1533-1548.
17. Tractinsky, N. (1997). Aesthetics and apparent usability: empirically assessing cultural and methodological issues. *ACM CHI Conference on Human factors in Computing Systems*. p. 115-122.
18. Ware, C. and Balakrishnan, R. (1994). Reaching for objects in VR displays: Lag and frame rate. *ACM Transactions on Computer-Human Interaction*. 1(4). p. 331-356.
19. Ware, C. and Lowther, K. (1997). Selection using a one-eyed cursor in a fish-tank VR environment. *ACM Transactions on Computer-Human Interaction*. 4(4). p. 309-322.
20. Welford, A. (1968). *The fundamentals of skill*. London: Methuen.
21. Zhai, S., Hunter, M. and Smith, B. (2000). The Metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. *ACM UIST Symposium on User Interface Software and Technology*. p. 119-128.