Endpoint Prediction Using Motion Kinematics

Edward Lank
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
lank@uwaterloo.ca

Yi-Chun Nikko Cheng
Computer Science Department
San Francisco State University
San Francisco, CA, USA
nikko@sfsu.edu

Jaime Ruiz
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
jgruiz@cs.uwaterloo.ca

ABSTRACT
Recently proposed novel interaction techniques such as cursor jumping [1] and target expansion for tiled arrangements [13] are predicated on an ability to effectively estimate the endpoint of an input gesture prior to its completion. However, current endpoint estimation techniques lack the precision to make these interaction techniques possible. To address a recognized lack of effective endpoint prediction mechanisms, we propose a new technique for endpoint prediction that applies established laws of motion kinematics in a novel way to the identification of motion endpoint. The technique derives a model of speed over distance that permits extrapolation. We verify our model experimentally using stylus targeting tasks, and demonstrate that our endpoint prediction is almost twice as accurate as the previously tested technique [13] at points more than twice as distant from motion endpoint.

Author Keywords
Cursor prediction, minimum jerk, Fitts' Law, motion, kinematics.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Novel interaction techniques such as target expansion [13] and cursor jumping [1] seek to overcome the limitations of Fitts' Law by modifying the Index of Difficulty in typical Fitts' pointing tasks. In the case of target expansion, the width of a target is increased, and in cursor jumping the distance of traversal is reduced. While both these techniques may speed pointing precision in arbitrary pointing tasks, both pre-suppose an existing mechanism for the prediction of motion endpoint at some intermediate point along the gesture, particularly for dense target arrangements. Despite a recognized need for endpoint prediction techniques (see, for example, [13], p. 416), little existing research exists on endpoint prediction. What little research there is relies either on probabilistic analysis of direction [17] [16], or on extrapolation algorithms [1] [13] [12]. Researchers typically use these techniques to assign probabilities to candidate widgets in the user interface, and the probabilities may be modified by an analysis of user behaviour. However, as noted by McGuffin and Balakrishnan [13], assuming all targeting is aimed toward identifiable widgets limits container space in the expansion region around widgets to output only. While endpoint prediction could be a valuable tool in user interfaces, techniques that determine endpoint location are not currently grounded in an understanding of the typical kinematics of human motion.

In our work, we seek a predictor of motion endpoint independent of underlying target likelihood. While target likelihood can easily be incorporated into endpoint estimation techniques, the ability to predict endpoint independent of underlying target priors, and an understanding of the accuracy with which target can be predicted, should allow more effective application of underlying probabilistic models of widget activation. As well, at a more academic level, a better understanding of how the kinematic profiles, i.e. the speed and distance over time profiles, can be used in predictive interfaces is of interest as our understanding of user behaviour is improved.

In this paper, we present a target prediction technique that extends established laws of motion kinematics, currently academic models of motion, into the realm of predictive tools. We first develop a theoretical model of endpoint prediction using principles from the minimum jerk law [19] [10] and the stochastic optimized-submovement model [15], and then validate our predictive model through subject testing. Our endpoint estimation technique doubles the accuracy of a previously analyzed technique for endpoint prediction [13]. In a tiled, collinear target arrangement, where no intervening white space separates targets aligned with direction of motion, our algorithm can identify specific target over 42% of the time, and predict adjacent to the correct target an additional 39% of the time (off-by-one), even on targets as small as 15 pixels in diameter. As well, our prediction occurs at more than twice the distance from motion endpoint of the previously analyzed technique.

RELATED WORK

A significant body of research on improving pointing speed in interfaces exists. Techniques exist that: manipulate the control-display ratio (e.g. [4]); skip empty inter-widget space (e.g. [9] [1]); alter the cursor activation area (e.g. [8] [11]) or the target size [13]; or move the target closer to the cursor (e.g. [2]) or the cursor onto the nearest target (e.g. [3]). McGuffin and Balakrishnan [13] note that, as long as target widgets are sparsely placed, these techniques can aid the selection of a single target. However, as targets become more densely arranged on the display, or in the extreme case when targets are tiled, these pointing facilitation techniques require some ability to identify gesture endpoint.

Relatively little Human-Computer Interaction work has been performed on endpoint prediction. The current techniques exist in two forms, both of which involve linear extrapolation. The first technique uses peak movement velocity as a basis point for linear extrapolation, while the second involves linear extrapolation near gesture endpoint (during the last 10% of gesture motion). Some work has examined the linear nature of paths on computers [17], and the temporal characteristics of aimed motion [14].

Endpoint prediction using peak velocity is a two stage process. First, an interface must be tuned to aid the system in identifying peak velocity, and the typical location of peak velocity for individual users. By identifying peak velocity, looking at the gesture length prior to reaching peak velocity, and multiplying the distance to peak velocity by a scale factor, Asano et al. predict target location [1]. They use their algorithm to speed pointing by jumping the cursor from a position just beyond peak speed to its predicted motion endpoint. However, they judge their technique effective only for distances over 800 pixels on a 1024 X 768 resolution display. At shorter distances, for example, 500, 600, and 700 pixels, their technique does not speed pointing. They do not report on the accuracy of their prediction algorithm, but instead only elaborate on the temporal reduction in pointing tasks as a result of cursor jumping.

In their work on expanding targets, McGuffin and Balakrishnan [13] develop a simple target predictor based on 3-point linear extrapolation to a future point where velocity is 0, and analyze the accuracy of their estimation technique for tiled target arrangements. If, during extrapolation, a 0-velocity point exists, and the distance remaining to that point is less than 10% of the distance traversed to the current point in the gesture, the algorithm predicts endpoint based on linear decelleration from the current point. Using this predictor, with only 9% of gesture remaining, they predict final target on a tiled button bar 21% of the time, are off by one button 26% of the time, and are more distantly incorrect 53% of the time.

To improve on target prediction, we introduce a new extrapolation technique derived from established laws of motion kinematics. In the following sections, we describe our technique, validate it experimentally, and describe the theory that underlies its effectiveness.

CALCULATING GESTURE ENDPOINT

Psychology, neurophysiology, and psychophysics have analyzed human motion with the goal of describing the laws that guide the speed and distance profiles of this motion. The laws, developed in over 20 years of research in this domain, can serve as an effective starting point for a more theoretical approach to motion analysis. Beyond Fitts' Law [5], these laws include the stochastic optimized-submovement model and the minimum jerk law.

The stochastic optimized-submovement model of Meyer et al. predicts that targeted motion occurs in two stages [15]. A large initial impulse is aimed at the centre of the motion's target. This initial impulse, lasting time T_1, consists of primarily ballistic motion that brings a subject close to the final target. As the subject nears the final target, feedback mechanisms in the neurophysiological system correct the movement, if necessary, with secondary movements lasting time T_2. Goal directed movement is a stochastic optimization problem, where the increased error rate of higher initial motion amplitudes (with higher probability of secondary impulses) trades off against the shorter time to traverse the distance to the final target. The model corresponds well with Fitts' Law and experimental data, converging on a logarithmic Index of Difficulty term as the number of secondary impulses increases.

While this model of aimed motion exists, the model has not been used to predict target endpoint by user interface or psychology researchers. One reason for this is that the model predicts temporal, not spatial characteristic of movements. As well, while duration of movements are predicted, instantaneous velocities are not. Finally, the model has been derived from rhythmic, not single-gesture, motion. A partial understanding of how the characteristics of this law might be adapted can be informed by recent HCI research.

In work on expanding targets, both Zhai et al. [20] and McGuffin and Balakrishnan [13] note that time taken for targeting depends on final target size, not initial size, even if expansion occurs as late as at 90% of the total distance traversed by a gesture. When coupled with Meyer's work on submovement in gesture, it follows that, based on two-phases of motion, corrective feedback for targeting is concentrated during the last 10% of any gesture's motion.

In Figure 1, we see that the final 10% of a ballistic (unaimed) gesture's displacement consumes 25% of the total time. Based on the velocity profiles in work of MacKenzie et al. [14] and Graham and MacKenzie [7], the final 10% of aimed gesture displacement may consume as much as 50% of gesture time. However, if we aim to enable interaction techniques such as expanding widgets, we must predict gesture endpoint prior to the final 10% of gesture motion, i.e. prior to dominant aiming effects. As a result, we focus our modeling on the primary submovement in a gesture, the primarily ballistic component. In essence, our work serves two purposes: it develops a technique for endpoint prediction; and mathematically characterizes the possible accuracy of endpoint prediction prior to 90% of gesture displacement.

Unconstrained Motion

The kinematics of unconstrained, ballistic motion obey the minimum jerk law [19]. Jerk, the time derivative of acceleration, is minimized by velocity signatures that vary smoothly over time (i.e. with no rapid changes in acceleration). The minimum jerk law was initially formulated by Hogan [10] using the calculus of variation. While the full details are beyond the scope of this paper, the minimum jerk path has pop (the sixth derivative of position) equal to 0, or, more specifically, constant crackle (the fifth derivative of position in time), yielding an equation for distance traveled, x, of the form:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \qquad (1)$$

Assuming a subject starts at rest (i.e. speed and acceleration are equal to 0), and setting the initial displacement to 0, the equation becomes:

$$x(t) = a_3 t^3 + a_4 t^4 + a_5 t^5 \qquad (2)$$

More generally, Hogan noted that, for paths that seek to travel a total displacement D in T seconds, the equation can be written as:

$$x(t) = D\left[10\left(\frac{t}{T}\right)^3 - 15\left(\frac{t}{T}\right)^4 + 6\left(\frac{t}{T}\right)^5\right] \qquad (3)$$

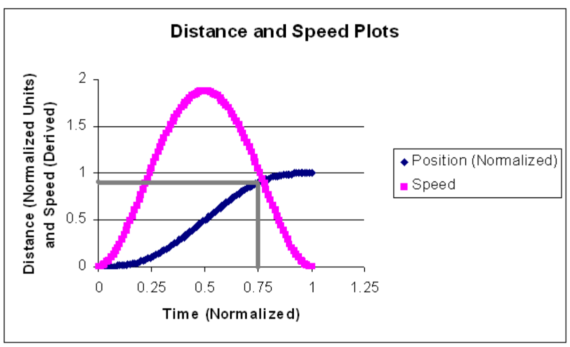

Figure 1. Theoretical distance and speed versus time profiles predicted by the Minimum Jerk Law.

Our goal is to predict endpoint in real time based on kinematics during the unconstrained initial submovement of an aimed gesture. As a result, we focus in this section on the overall form of the equations, so units of distance and time are, at this point, arbitrary. Normalizing such that $D = 1$ and $T = 1$ for simplicity (i.e. assuming arbitrary unit distance over arbitrary unit time), we can rewrite Equation 3 as:

$$x(t) = 10t^3 - 15t^4 + 6t^5 \qquad (4)$$

and produce an equation for speed by taking the derivative of Equation 4 with respect to t, yielding, after some simple algebra:

$$v(t) = 30t^2 (t-1)^2 \qquad (5)$$

The equations for distance and speed are plotted in Figure 1. The minimum jerk model is a well-established model of unconstrained motion, and has been validated by the analysis

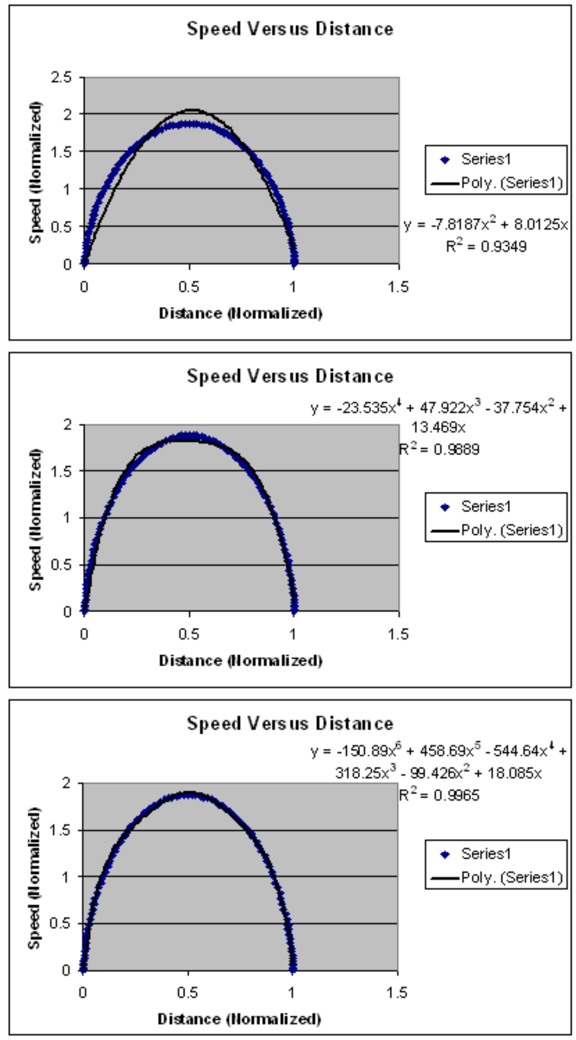

Figure 2. Theoretical speed versus distance profile predicted by the Minimum Jerk Law.

of the speed signatures of subjects engaged in rhythmic motion using, for example, pen input devices.

Our goal is the prediction of the endpoint of single gesture movement. While path motion tends to move in a straight line directly toward the target [6], the length of the path, peak speed, and time varies. If we can predict the final length of a gesture at the beginning, midpoint, or some other sufficiently early intermediate point on the gesture, endpoint location in screen coordinates can be inferred by projecting along the direction of motion to the appropriate length.

When predicting endpoint, our goal is to extrapolate using a function fit to a partial gesture. Various approximation techniques can be used to determine values for gesture endpoint based on the above equations [18]. The most convenient representation would be an instantaneous speed versus distance graph, as we could fit a $(distance, speed)$ function and then calculate total distance directly from the function, thus eliminating any noise contribution from gesture time.

To understand how instantaneous speed varies over distance, the minimum jerk model must be transposed from a speed signature over time to a speed signature over distance. Figure 2 depicts this relationship by plotting $(distance, speed)$ points calculated using Equations 4 and 5 over time interval $[0, 1]$. Also shown are three polynomial fitting functions: at the top, a quadratic polynomial (x^2), in the middle a quartic polynomial (x^4), and at the bottom a degree six polynomial, each with a least-squares fit polynomial and the polynomial's correlation.

Predicting Gesture Length

Endpoint prediction involves extrapolation of a best-fit polynomial to determine gesture length. This best-fit polynomial will describe the variation in speed over distance traveled. Neurophysiologists have also analyzed the path taken during motion, and note that the end-effector, in this case the pen or mouse, will typically follow a straight line [6] [17]. We can predict endpoint by extrapolating a distance, as we know that aimed motion follows a straight line path from starting point to ending point.

Extrapolation, particularly extrapolation of distant points, is a numerically unstable process [18]. As a result, it is desirable to use the lowest degree polynomial possible to extrapolate. While a polynomial of degree six has best fit, shown in Figure 2, there is a risk of over-fitting the data, which can affect extrapolation. Overfitting effects are present in both the degree six and quartic polynomial. Figure 3 demonstrates the use of a quartic polynomial to extrapolate. The quartic function oscillates until reaching the last data point, and, rather than continuing smoothly, instead bends abruptly toward the x-axis.

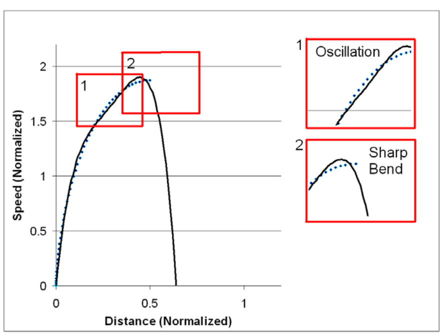

Figure 3. Fitting issues with a quartic polynomial include undesirable oscillation (1) and sharp bends rather than smooth continuity (2).

Using least squares fitting on data points, we can calculate a quadratic polynomial that behaves regularly and use this polynomial to extrapolate. However, one challenge with degree two polynomial fits is that the velocity versus distance profile is not a perfect parabolic function. When we examine the fit of a degree two polynomial for the theoretical velocity versus distance plots in Figure 4 taken at 30%, 50%, 80%, and 90% of stroke length, the polynomial underestimates prior to 80%, is accurate at 80%, and then overestimates.

Two sub-optimal solutions present themselves. The first, attempting to fit a higher order polynomial, results in an inability to effectively extrapolate due to the need to predict a data point $(x, v(x))$ that is distant from our sampled data points during a partial gesture [18]. The second, fitting a lower-order, well-behaved polynomial is the established approach for performing distant extrapolation on data, but results in measurable inaccuracies in our predicted endpoint, even on theoretical data. While the standard rule of thumb for extrapolation is to use the lowest degree polynomial possible, what is needed in this case is some technique to correct measurable theoretical errors in extrapolated values. In this section, we describe an extrapolate-then-correct process for endpoint estimation. Later in the paper, we describe the motivation for this extrapolate-then-correct approach, and the characteristics that allow it to be effective in endpoint estimation.

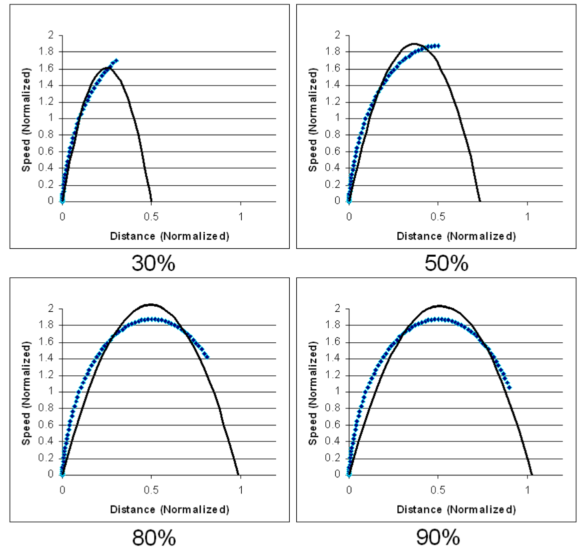

Figure 4. Fitting inaccuracies at 30%, 50%, 80% and 90% of gesture. At 80% of gesture, polynomial x-intercept and actual endpoint correspond perfectly.

Our solution to correctly estimate endpoint uses a set of coefficients calculated from the theoretical data produced by Equations 4 and 5 to correct the prediction. The coefficients are determined by comparing the extrapolated value produced by a polynomial fit to theoretical data to the known endpoint. Table 1 depicts the coefficient obtained by dividing actual endpoint of our theoretical curves ($x = 1$) by the x-intercept of a quadratic polynomial fit to the first s_i fraction of data points in the stroke. The assumption is that real human motion will be represented sufficiently accurately by the laws that were used to create this data that it, too, will produce polynomial fits that exhibit similar inaccuracies, and that the same coefficients will apply. We can generate coefficients at an arbitrary density along the gesture extents by simply calculating endpoint, fitting, and tabulating the reciprocal of the value. In our implemented prediction algorithm, we currently tabulate coefficients at 1000 equally spaced points along the theoretical gesture. The coefficients, calculated only using Equations 4 and 5, are independent of actual subject motion.

Stroke Index (s_i)	Coefficient
30%	2.01
40%	1.58
50%	1.36
60%	1.20
70%	1.09
80%	1.02
90%	0.97

Table 1. Coefficients to correct for predicted endpoint, as calculated on theoretical data. We use the values to correct estimation in actual gestures.

Given these coefficients, predicting gesture length for real user motion is a two-step process. Given a partial gesture drawn by the user, we fit a quadratic polynomial to a partial gesture's $(x, v(x))$ data points. One x-intercept occurs at $(0, 0)$; the other occurs at some location along the x-axis, x_{calc}. We seek a prediction for endpoint, x_{actual}. To determine x_{actual}, we must determine the coefficient by which x_{calc} must be multiplied. However, to determine coefficient, we must also determine portion of gesture we have completed, s_i, which is also unknown. We determine coefficient and fraction of gesture numerically as follows.

Let us assume that a user has begun a gesture and has traversed distance d of the total intended gesture length, L. We wish to determine an estimate of L which we call x_{actual}. Given the user's partial gesture of length $d < L$, we can fit a quadratic polynomial to the $(x, v(x))$ data points of the partial gesture, giving us two pieces of data: the distance drawn from the starting point to the current point along the gesture, d, calculated based on Euclidean geometry (we know where the user started and their current location); and the x-intercept calculated from a quadratic polynomial fit to the data, x_{calc}, using least squares fitting of a quadratic polynomial to the entire set of points from beginning of the gesture to current location. Unknown are the coefficient, c_r, and the actual endpoint predicted by our formula, x_{actual}. Note that the coefficient, c_r, is a function of the fraction of the gesture that has been completed, s_i, tabulated above in Table 1. Two equations present themselves:

$$x_{actual} = c_r x_{calc} \tag{6}$$

$$d = s_i x_{actual} \tag{7}$$

Equation 6 is the mechanism we use for calculating our predicted endpoint, x_{actual}, multiplying the x-intercept by a specific coefficient, c_r. Equation 7 describes d, the distance traversed, as a function of fractional distance s_i from estimated endpoint x_{actual}. Substituting x_{actual} in Equation 7 using Equation 6, we find that:

$$d = s_i c_r x_{calc} \tag{8}$$

Because c_r is a function of s_i based on tabulated values, we can numerically determine the values for s_i and c_r based on Table 1 that satisfy the equality in Equation 8. Once c_r has been determined, we can predict endpoint location x_{actual} using Equation 6. We determine c_r via exhaustive search, a process that takes about 1ms.

To summarize, we use the following real-time process to predict endpoint of a partially completed gesture:

1. Given a partial gesture of length $d < L$, L = total gesture length, we fit a quadratic equation to the data points $(x, v(x))$ along the partial gesture.
2. One x-intercept occurs at point $(0, 0)$, the other at a more distant point, x_{calc} along the x-axis. We determine x_{calc} by solving the quadratic polynomial for its roots.
3. Given x_{calc}, we use Equation 8 and Table 1 to determine a value for c_r.
4. We multiply c_r by x_{calc} to determine x_{actual}, an estimate of actual gesture length L.

In the following section, we analyze the accuracy of x_{actual} as a predictor of L, the actual gesture length. Following an analysis of the predictive ability of our model, we more fully analyze the theoretical underpinnings of our prediction process.

VALIDATING THE MODEL

As noted earlier, extrapolation is a numerically unstable process. An ability to predict motion endpoint is based on a strong convergence of actual subject motion to the theoretical model of motion that underlies our prediction technique. The theoretical model has typically been used in rhythmic motion, rather than single gesture motion. An open question exists as to the accuracy with which the theoretical model predicts single gesture behaviour.

Method

To validate our model, we asked ten subjects to draw 100 stylus targeting gestures on a 14 inch tablet computer with 1024 X 768 screen resolution. The length of the targeting gestures varied from 200 to 600 pixels by 100-pixel increments. Using circular targets, we varied target diameter between 15 and 75 pixels by 15-pixel increments. Subjects saw five different gesture-length/target-size pairs, counterbalanced using a 5X5 Latin square. Subjects drew 20 gestures for each of the five length/size combinations they were assigned. The order of presentation of individual length/size combination was randomized, as was the direction of the gesture. During a single gesture, subjects were presented with a start location. They depressed the stylus inside the start location. After a 1.5 second time-out, a target was presented. Subjects drew a gesture to the target and lifted their pen inside the target region. Similar to directives in typical Fitts' Law experiments, subjects were asked to draw "as quickly as possible and as accurately as possible." The software captured location information in tablet ink coordinates and time in ticks to maximize data resolution for analysis. 35 of the gestures drawn were target misses, a 3.5% error rate which agrees well with ideal performance in Fitts' Law pointing tasks. We eliminated these gestures when doing analysis.

To analyze our data, we calculated speed and position for points along the gesture. Speed data was smoothed using

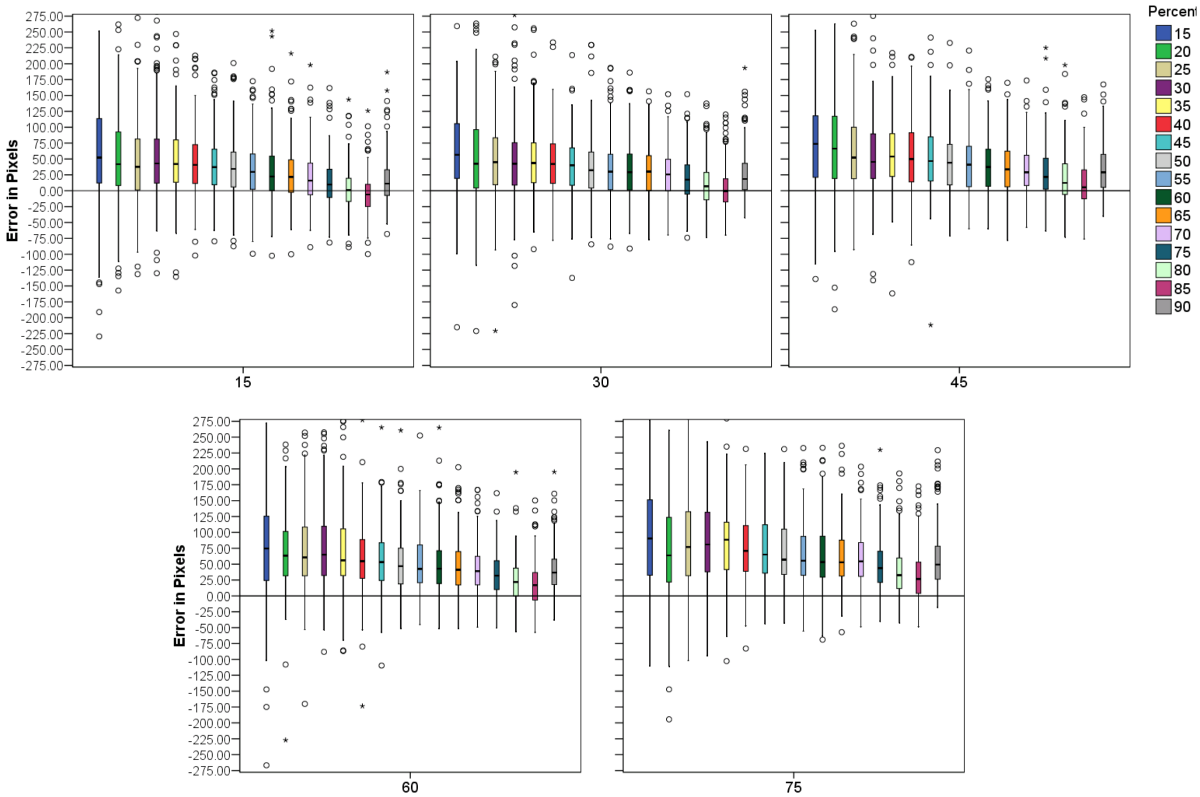

Figure 5. Predictive accuracy of our algorithm at locations along gesture path. Gesture path percentage is estimated based on $\frac{distance\_traveled}{predicted\_endpoint}$.

an interpolating degree 2 polynomial. We fit portions of the gesture, specifically the first 15% 20%, 25%, 30%, etc. of (distance, speed) points in 5% increments of gesture length to generate endpoint predictions at locations along the gesture.

We performed two analyses of our results. First, we calculated percentage of stroke distance based on known expected gesture endpoint and used that coefficient to estimate endpoint (i.e. we calculated percentage based on ground truth values of percentage of gesture completed). We also calculated our endpoint based on our numerical solution using tabulated values to search for equality as described above and stored predictions along the gesture. We report our results using the numerical calculation of gesture percentage (the real-time technique) rather than percentages calculated from known gesture length, as, in a real world predictive technique, this represents expected behaviour. This broadens our distributions slightly. When gestures are estimated longer than actual (i.e. x_{calc} is larger than it should be), the percentage of gesture completed is smaller than it should be. This results in a larger coefficient, meaning that x_{actual} is pushed even longer than if we used absolute percentages. Similarly, when we underestimate, the smaller corresponding coefficient biases to a shorter prediction.

At each portion of the gesture, we compared our predicted endpoint with ground truth for the current gesture. Two candidates present themselves as possible ground truth values: the centre of the target and the observed endpoint of the gesture we are analyzing. Both produce similar error measurements. We chose target centre for two reasons. First, based on established laws of motion, a subject should aim toward the centre of the target, and the actual endpoint of their gesture should be normally distributed around that centre. The actual endpoint of any gesture is a result of an initial submovement and, potentially, secondary, corrective movements that occur after initial submovement. Those gestures requiring unpredictable secondary submovements would increase our error rate, while those without would reduce the error rate. Depending on the frequency of secondary submovements, prediction error might be biased either for or against our algorithm. Second, if we use gesture endpoint for the same target presented to the same subject twice, then each gesture produces its own endpoint and ground-truth is a gesture-specific measure. Repeatability of measurements does not exist, and analyses of the distribution of predictions are gesture-specific rather than condition-specific. Measuring accuracy by condition allows us to determine whether our prediction will be useful as an enabling technology, or whether it simply constitutes an intellectual exercise which, while still of value, has little practical relevance.

Results

Figure 5 shows the accuracy distributions for our endpoint predictions using a box and whisker plot. For each target size, endpoint prediction is plotted from 15% to 90% of gesture and is reported as distance from ground truth. Boxes contain 50% of the values; whiskers contain all non-outlier data values. Our best predictive power seems to occur at approximately 80% of gesture length. This corresponds to 67% of initial submovement time, based on the equations for distance and speed in time, Equations 4 and 5, plotted in Figure 1. Based on work on expanding targets, 80% seems a convenient percentage of gesture length from which to predict endpoint location. At this point, 42.4% of target predictions fall within $\pm 0.5W$ of target centre, i.e. within the target, and an additional 39% of target values fall within $\pm 1.5W$, i.e. within the adjacent target.

Examining fitting accuracy at 80% of the gesture, we would expect target size to affect accuracy of prediction, as larger target size allows more tolerance for endpoint. Target effects can be observed in Figure 5. ANOVA of prediction accuracy (pixel error) for target size and distance shows a significant effect for target size, $F_{4,961} = 29.167$, distance, $F_{4,961} = 7.230$, and target*distance interaction, $F_{16,949} = 6.082$, $p < 0.01$ in all cases. Post-hoc pairwise tests (Tukey's) indicate significant ($p < 0.05$) differences between all targets except 15 and 30 and 45 and 60. Only distance of 600 pixels differs significantly, in its case from all other distances.

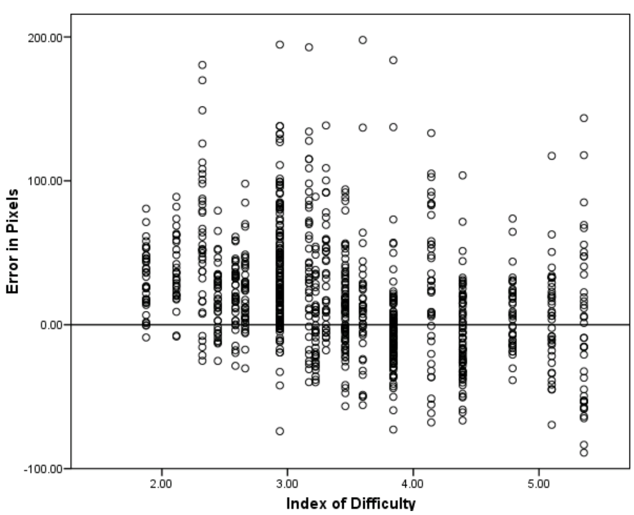

Figure 6. Scatterplot of prediction accuracy versus Index of Difficulty.

We see somewhat poor performance for predictions of 15-pixel targets at 600 pixels distance. A scatterplot of target error versus Index of Difficulty in Figure 6 demonstrates the increased challenge of the most distant 15-pixel target (the highest ID line on the graph, to the right). The distribution seems uniform, rather than normalized. However, in Figure 7 we show a histogram of endpoint distribution for each of the target sizes over all distances. Targets of other sizes allow consistent prediction accuracy at all distances. We have no good explanation for the 15-pixel target at 600-pixel distance being an outlier. It may be that, with such a distant small target, subjects relied more on steering. More analysis is necessary before we conclude either that there is an Index of Difficulty limit to predictive power, or that the two users who had this target-distance combination were outliers.

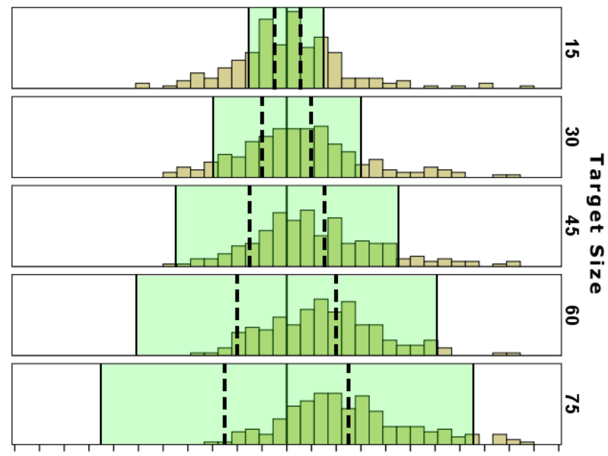

Figure 7. Histograms of endpoint predictions. Dashed lines representing ± 0.5 target size and shaded regions representing ± 1.5 target size are superimposed on the image. 42.4% of predictions fall within the dashed regions, i.e. within the target, while 81.4% of predictions fall within the shaded regions, i.e. $\pm$ one target, assuming tiled, collinear widgets of identical size.

DISCUSSION

Compared to McGuffin and Balakrishnan's predictive accuracy of 21% and 26% respectively, we are approximately twice as accurate at twice the distance from endpoint. Conveniently, our peek predictive ability occurs at 80% of gesture length, the point where our correction coefficient $c_r = 1$. Also, if we assume, as in McGuffin and Balakrishnan's work, that we will only expand our predicted target, and would occlude candidate targets (to avoid collinear motion of adjacent targets), then our prediction rates for target p and adjacent target q would allow an optimal expansion of target by 1.56 times and an Index of Difficulty benefit, despite partial adjacent target occlusion. As well, our predictive algorithm does not consider underlying candidate probabilities based on user behaviour. Incorporating accurate priors on underlying target likelihood would allow a further refinement of prediction and a correspondingly larger reduction in Index of Difficulty through increased target expansion. As well, studying collinear target motion that results from the expansion of adjacent targets might allow the expansion of three candidate targets, resulting in over 80% accuracy and adjacent target likelihood of less than 20%.

Analytically, it seems we could improve our results slightly if, instead of using coefficients from theoretical data, we calculated coefficients based on observed data. We would have to test these ad hoc coefficients on new subjects, as the apparent benefit might be a subject specific result that would not generalize beyond the ten subjects in this trial. New coefficients would not affect the spread of endpoint values, but they might centre endpoint prediction on the origin, boosting our accuracy. An open question exists as to whether the error we are observing is in our calculation (i.e. measurable inaccuracies in the minimum jerk model), or if user motion typically overshoots and decelerates. More work is needed before arbitrarily adjusting coefficients.

One shortcoming of previous prediction techniques is the use of a single location, either at peak speed or at 90% of gesture length, and performing prediction based on one to three points of localized data. Human motion exhibits naturally occurring noise (see, for example [5]), and PC-based measurements also exhibit noise resulting from idiosyncratic event delivery and discrete pixel-based measurement of continuous data. Using only a very small portion of the data magnifies the effect of noise. Using the entire data set in extrapolation allows us to generate more stable results, as we maximize our signal to noise ratio. Examining Figure 5, before a sufficiently strong signal exists our extrapolation distribution has much more variance.

The one remaining issue in our endpoint prediction technique is a lingering concern regarding the extrapolate and correct approach to endpoint prediction. It may seem informal to some. Although we note that our optimal predictive accuracy occurs at 80% of gesture, meaning that without a corrective coefficient we can still obtain an accurate result, it is academically of interest to consider predictive ability at other locations along gestures. This allows comparison to other work. For example, Asano et al. [1] propose a cursor jumping technique using peak velocity and a grid-like placement of targets (with grid size of 2W). Analyzing our extrapolation technique at 50% of gesture (peak velocity of initial submovement), we can accurately identify the correct target grid approximately 48.3%, and are off by one an additional 35.7% of the time comparing our target size of 45 pixels to their 50-pixel target.

The typical way that extrapolation techniques are evaluated is by how accurately they predict the real-world phenomena they seek to model, and occasionally by some independent estimate of error in the extrapolation technique. We cannot independently measure error in our extrapolation technique, as we base our technique on theoretical data that corresponds to a law defined by rigid equations, not experimental data. Our error measurements are based on rate of accurate prediction, as above. The specific concern that remains may be that some more accurate polynomial describing speed as a function of distance could be created, eliminating the need for partial polynomial correction and allowing a direct calculation of endpoint. We address this concern in the next section, justifying the need for an approximating polynomial through an application of the fundamental theorem of algebra and Galois theory.

THEORETICAL UNDERPINNINGS

In this section, we describe the theory that underlies our endpoint approximation technique, particularly the extrapolate-then-correct approach described just prior to our study. Earlier, we described equations for position and speed based on the minimum jerk law, Equations 4 and 5, reproduced here as Equations 9 and 10:

$$x(t) = 10t^3 - 15t^4 + 6t^5 \tag{9}$$

$$v(t) = 30t^2 (t-1)^2 \tag{10}$$

Note that these equations are normalized; time and position exist on an arbitrary $[0, 1]$ scale in our range of interest.

A full graph of Equation 10, with t taking all negative and positive values, produces the graph pictured in Figure 8. In real world phenomena, negative time and time values above 1 ($t < 0$ and $t > 1$ in the graph) are not observed as they occur before and after the gesture, respectively. Given equation 10, however, we can rewrite the equation as:

$$t^2 - t \pm \sqrt{\frac{v(t)}{30}} = 0 \tag{11}$$

solvable using the quadratic formula such that:

$$t = \frac{1 \pm \sqrt{1 \pm 4\sqrt{\frac{v(t)}{30}}}}{2} \tag{12}$$

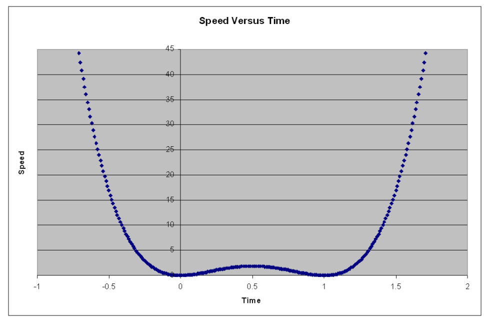

Figure 8. A plot of t over all negative and positive values.

Consider the term $\pm 4\sqrt{\frac{v(t)}{30}}$ in Equation 12. If the $\pm$ operation is addition, then the term inside the square root is greater than 1, the numerator is greater than 2 or less than 0, and t values lie outside our range of interest. We can restrict the operation preceding the nested square root term as subtraction. As well, with subtraction as the operation, we know that $4\sqrt{\frac{v(t)}{30}}$ must be less than 1 to produce real (as opposed to complex) values. It can be easily demonstrated that $v(t) = \left[0, \frac{30}{16}\right]$ produces real values in our range of interest. In Hogan's work, he noted the existence of a ratio of peak speed to average speed of $1.875 : 1$, similar to the value calculated above [10]. Plotting t-values over the range $\left[0, \frac{30}{16}\right]$ in equation 9 produces the graph pictured in Figure 9.

We now extend Hogan's work with some additional mathematical manipulation. In Equation 12, we can substitute $V_t = \sqrt{1 - 4\sqrt{\frac{v(t)}{30}}}$ to simplify mathematical manipulation of the square root term. Equation 12 can be rewritten as:

$$t = \frac{1 \pm V_t}{2} \tag{13}$$

Substituting this value back into our distance equation, Equation 9, multiplying out the t^3, t^4, and t^5 terms, and grouping like terms yields an equation for $x(V_t)$ of the form:

$$\pm \left[\frac{3{V_t}^5 - 10{V_t}^3 + 15V_t}{16}\right] + \frac{1}{2} - x = 0 \tag{14}$$

We seek an equation for speed, v, as a function of distance traveled, x. To do this, we must solve the above fifth degree polynomial for its V_t roots and substitute our V_t expression back into the root equation to determine speed as a function of distance. We cannot simplify any of the V_t terms, as neither x nor $\frac{1}{2}$ contain terms in V_t. From Galois Theory, we know that all polynomials of degree > 4 cannot be solved exactly for roots. One must resort to Newton's method to numerically approximate roots. Numerical root finding will require an approximation of the overall function, and, as we aim to extrapolate from the function, we need some mathematical expression to describe as yet unobserved regions of the $(x, v(x))$ graph. We obtain this mathematical expression by least-squares fitting a polynomial to the speed versus distance plot in Figure 9.

In Figure 2, we see that quartic and sixth degree polynomials fit the data well. However, as we note earlier, higher degree polynomials are inappropriate for extrapolation (see Figure 3), and quadratic polynomials do not accurately fit the data, undershooting and overshooting to degrees that seem predictable (see Figure 4). Our goal is to overcome the shortcomings of both quadratic and higher degree polynomials through some mathematical analysis of characteristics of the underlying functions. We will perform an implicit higher degree fit using a lower degree polynomial.

Based on the fundamental theorem of algebra, we know that any polynomial with real coefficients of degree > 2 can be written as the product of a series of polynomials in x^1 and x^2. The real roots of the polynomials, if they exist, correspond to the x^1 term(s), while the complex roots correspond to the x^2 term(s). It is a natural corollary of this theorem that any odd degree polynomial has at least one real root.

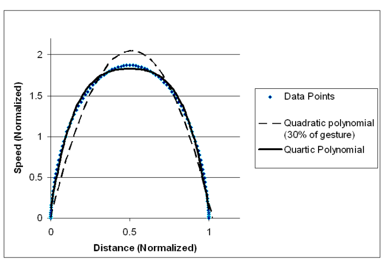

Figure 9. Quadratic and quartic polynomials fit to theoretical data.

Let us assume that we are performing a fit using the quartic polynomial, as shown in Figure 9, solid line. We know the quartic polynomial has two real valued roots, approximated by 0 and 1, and two imaginary roots, approximated by Matlab as $0.51 \pm 0.55i$. We also know that the polynomial is symmetric about a vertical axis. Furthermore, because this polynomial has two real roots and two complex roots, we know that it can be rewritten as:

$$p(x) = (a_1x - a_0)(b_1x - b_0)(c_2x^2 + c_1x + c_0) \quad (15)$$

Given a portion of the data points from the start point to some point d along the polynomial, $p(x), x = [0, d]$, we approximate the polynomial with a degree 2 polynomial and determine two roots. However, we are approximating a degree four polynomial. The differences between a quadratic polynomial and a quartic polynomial symmetric about a vertical axis are in horizontal scale. A quartic polynomial is slightly shorter and slightly wider. Moreover, if one attempts to approximate this fourth degree polynomial using a quadratic polynomial on a contiguous subset of points starting from position 0 and extending distance d into the polynomial, $d < L$, L = the length of the gesture, the distant root of the quadratic polynomial will be too close to the origin. If one plots the scaling factors that transform a quadratic least squares fit polynomial into the appropriate quartic polynomial, they lie along a quadratic curve, as any degree four polynomial can be represented as the product of two quadratic equations. The quadratic curve for the more distance $v(x) = 0$ root is pictured in Figure 10, and is comprised of the calculated coefficients we identified when describing our prediction algorithm.

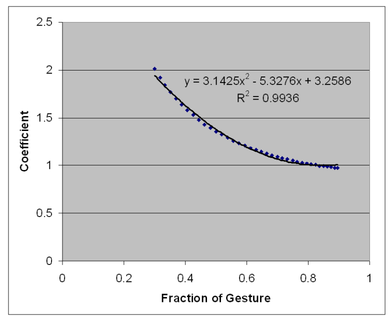

Figure 10. Quadratic curve fit to coefficients that are used to correct our endpoint location.

From Figure 10, we see that the coefficient, c_r, is a function of the fraction of the gesture s_i. Examining Equation 8, we now see that we could rewrite the equation as an equation in one unknown. Given the equation for coefficients by least-squares fitting of data points in Figure 10, i.e.:

$$c_r = 3.14{s_i}^2 - 5.33s_i + 3.26 \quad (16)$$

and substituting into Equation 8, we obtain a cubic equation in s_i:

$$(3.14{s_i}^3 - 5.33{s_i}^2 + 3.26s_i)x_{calc} - d = 0 \quad (17)$$

The fundamental theorem of algebra guarantees at least one real root in the above equation. Picking arbitrary values for x_{calc} and $d < x_{calc}$ in the range $[0, 1]$ shows that the cubic equation has only one x-intercept.

Our ability to correct the quadratic extrapolating function is surprising. In general, extrapolation is sufficiently numerically unstable that an attempt to correct an arbitrary under-fit value through numerical expansion would be almost impossible, unless some law strongly constraining the shape of the polynomial were available. Fortunately, human motion follows the minimum jerk model with enough precision that the overall shape of the polynomial is constrained sufficiently to permit correction of a under-estimate of actual endpoint.

This discussion assumes normalized time, t, to the range $[0, 1]$ and also distance, x, to the range $[0, 1]$, with average velocity equal to 1 unit distance per unit time. Any real gesture's $(distance, speed)$ data points follow the overall shape of the equation, scaled appropriately in speed and distance. An overall scaling to the $(x, v(x))$ data points occurs during the initial quadratic polynomial fit, and the expansion of the endpoint is determined using the coefficient table, which scales the calculated endpoint to the corrected position. Coefficients are included in Equations 16 and 17; these are numerical approximations, a result of the fact that we could not solve Equation 14 exactly.

In the algorithm implemented in predictive software, we typically do not solve Equation 17. We instead keep the tabulated values of coefficients, a subset of which are shown in Table 1. The full set of values are easily calculated from Equations 4 and 5, and these values can be generated dynamically at run-time. We keep the tabulated values for two reasons. First, an exhaustive search of even 1000 values is a trivial computational process, and can be expedited with binary search if necessary. Second, the tabulated values much more accurately represent the curve in Figure 10 than the least squares quadratic equation shown in Equation 16.

CONCLUSION

Based on a hypothesis that the initial impulse in Meyer's stochastic optimized-submovement model obeys the minimum jerk law, we derive a method for predicting gesture endpoint. Testing this model, we observe accuracies more than double those previously reported at more than twice the distance from estimated endpoint. With over one-third of gesture time and one fifth of gesture motion remaining, we predict target with over 42% accuracy and are adjacent to the target with an additional 39% accuracy. The accuracy is particularly impressive in the absence of intervening whitespace or underlying target probabilities to massage recognition rate.

ACKNOWLEDGMENTS

We thank subjects who participated in our studies. The research in this grant was funded by the Natural Science and Engineering Research Council of Canada (NSERC) and by the National Science Foundation, Award IIS-0448540.

REFERENCES

1. T. Asano, E. Sharlin, Y. Kitamura, K. Takashima, F. Kishino, "Predictive interaction using the delphian desktop", Proc. UIST 2005, pp. 133 - 141.

2. P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, A. Zierlinger, "Drag-and-pop and drag-and-pick", Proc. of INTERACT 2003, pp. 57 - 64.

3. E. Bier and M. Stone, "Snap-dragging", Proc. of ACM SIGGRAPH 1986, pp. 233 - 240.

4. R. Blanch , Y. Guiard , M. Beaudouin-Lafon, 11Semantic pointing: improving target acquisition with control-display ratio adaptation", Proc. CHI 2004, p.519-526.

5. P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement", J. of Exp. Psychology, 47 (1954), pp. 381-391.

6. T. Flash and N. Hogan. "The coordination of arm movements: an experimentally confirmed mathematical model", Journal of Neuroscience, Vol. 5, pp. 1688–1703.

7. E. Graham, C. L. MacKenzie. "Physical versus virtual pointing," Proc. CHI 1996, pp. 292-299.

8. T. Grossman, R. Balakrishnan, "The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area", Proc. CHI 2005, pp. 281 - 290.

9. Yves Guiard , Renaud Blanch , Michel Beaudouin-Lafon, Object pointing: a complement to bitmap pointing in GUIs, Proc. of Graphics interface 2004, p.9-16.

10. N. Hogan, "An organizing principle for a class of voluntary movements" Journal of Neuroscience, Vol 4, pp. 2745-2754.

11. P. Kabbash , W. Buxton, "The *prince* technique: Fitts' law and selection using area cursors", Proc. CHI 1995, pp. 273-279.

12. C. Lane, S. C. Peres, A. Sandor, and H. A. Napier, "A Process for Anticipating and Executing Icon Selection in Graphical User Interfaces", Int. J. Human-Computer Interaction, 19:2 (2005), Pages 241-252.

13. M. McGuffin, R. Balakrishnan, "Fitts' Law and Expanding Targets: Experimental Studies and Designs for User Interfaces" TOCHI 12:4, pp. 388 - 422.

14. C. L. MacKenzie, R. Marteniuk, C. Dugas and B. Eickmeier, "Three-Dimensional Movement Trajectories in Fitts' Task", Quarterly J. of Exp. Psychology, 39A (1987), pp. 629 - 647.

15. D. Meyer, J. Smith, S. Kornblum, R. Abrams, and C. Wright, "Speedaccuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action", in *Attention and Performance XIII*, M. Jeannerod, Ed., Lawrence Erlbaum, Hillsdale, NJ. 173 - 226.

16. A. Murata, "Improvement of Pointing Time by Predicting Targets in Pointing", Int. J. of Human-Comp Interaction, 10:1, pp. 23 - 32.

17. H. Oirschot and A. Houtsma, "Cursor Trajectory Analysis", Proc. of the Haptic Human Computer Interaction Workshop, 2000, pp. 127-134.

18. W. Press, S. Teukolsky, W. Vetterling and B. Flannery, "Numerical Recipes in C++", Cambridge Press, 2002.

19. J. Richardson and T. Flash, "Comparing Smooth Arm Movements with the Two-Thirds Power Law and the Related Segmented-Control Hypothesis", *M. Neuroscience*, 22:18 (Sept. 2002), pp. 8201 - 8211.

20. S. Zhai, S. Conversy, M. Beaudouin-Lafon, and Y. Guiard, "Human on-line response to target expansion.", Proc. CHI 2003, CHI Letters 5:1, pp. 177 - 184.