Analyzing the Kinematics of Bivariate Pointing

Jaime Ruiz* David Tausky† Andrea Bunt‡ Edward Lank§ Richard Mann¶

David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

ABSTRACT

Despite the importance of pointing-device movement to efficiency in interfaces, little is known on how target shape impacts speed, acceleration, and other kinematic properties of motion. In this paper, we examine which kinematic characteristics of motion are impacted by amplitude and directional target constraints in Fitts-style pointing tasks. Our results show that instantaneous speed, acceleration, and jerk are most affected by target constraint. Results also show that the effects of target constraint are concentrated in the first 70% of movement distance. We demonstrate that we can discriminate between the two classes of target constraint using Machine Learning with accuracy greater than chance. Finally, we highlight future work in designing techniques that make use of target constraint to improve pointing efficiency in computer interfaces.

Keywords: bivariate pointing, kinematics, Hidden Markov Models, maching learning, Fitts' law

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction styles

1 INTRODUCTION

In modern graphical user interfaces (GUIs), pointing devices, such as pens, mice, trackpoints and touchpads, are ubiquitous. These devices allow users to invoke commands in the interface by mapping pointing-device manipulation to the movement of an on-screen pointer. Given the ubiquity of pointing devices in GUIs, a significant body of Human-Computer Interaction (HCI) research deals with models of movement [3, 2, 5, 13].

Models of movement have frequently described the movement time, or temporal cost. Temporal-cost models exist for targeted motion [7], constrained motion [3], and drawing motion for single strokes [5]. Temporal models are valuable tools for interface designers, as the designers can rearrange interface elements and redesign interface tasks to maximize the temporal efficiency of the interface. While temporal models allow us to calculate time taken by tasks and modify an interface layout accordingly, they tell us little about the instantaneous kinematic characteristics (position, speed, acceleration, jerk as a function of time or position) of the movement.

Recent work has shown that benefits exist in understanding the kinematic characteristics of motion. For example, techniques such as target expansion [16] and predictive pointing [4] make use of kinematic understanding to speed pointing in interfaces. As well, techniques have been designed that use kinematics to determine sloppiness [14], or to recognize shapes [8]. In some cases [4, 14], the interaction techniques are fully dependent on kinematic understanding. In others [16], kinematics are most useful for some configurations of the interaction technique.

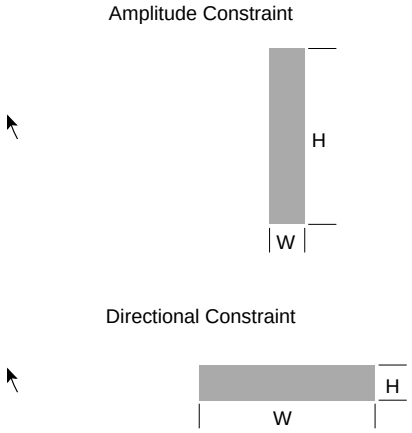


Figure 1: Pointing task constraints. At the top, an amplitude or stopping constraint. At the bottom, a directional or steering constraint.

In this work we examine the kinematics of bivariate pointing tasks, i.e., pointing tasks with either an amplitude or a directional constraint. Figure 1 illustrates these two constraints. An amplitude (or stopping) constraint, pictured at the top, exists when the height of the target is greater than the width of the target. With an amplitude constraint, the distance traveled must be controlled. A directional (or steering) constraint, pictured at the bottom, exists when the width of the target is greater than the height. The user needs to steer into the region of the target, i.e. to control the direction.

Any button widget in a typical graphical user interface places either an amplitude or directional constraint on a pointing task, depending on its position relative to the cursor. As well, amplitude constraints in interfaces include tasks such as targeting a scrollbar or targeting the edge of a window for resizing, and directional constraints include tasks such as targeting menus in computers running the MacOS operating system. Previous work has shown that the two constraints have an unequal impact on movement times for pointing tasks [1, 2]. However, there is not yet an understanding of the impact of amplitude versus directional constraints on the underlying motion characteristics.

The purpose of this research is to examine what effects an amplitude or directional constraint has on the kinematic characteristics of motion in pointing tasks. In particular, we are interested in determining which characteristics of motion are affected and to what degree the effects can be observed. We are also interested in understanding at what point in the gesture such effects become apparent.

We found that bivariate pointing data is complex, and that the two types of constraint result in significant overlap within the parameter space. To address the overlapping nature of the data, we applied machine learning techniques, specifically Hidden Markov Models (HMMs), to determine those parameters most affected by

*e-mail: jgruiz@cs.uwaterloo.ca
†e-mail: datausky@cs.uwaterloo.ca
‡e-mail: abunt@cs.uwaterloo.ca
§e-mail: lank@cs.uwaterloo.ca
¶e-mail: mannr@cs.uwaterloo.ca

Graphics Interface Conference 2008
28-30 May, Windsor, Ontario, Canada

bivariate constraint. We show that instantaneous speed, acceleration, and jerk are affected by the type of constraint. We also observed that orthogonal components (i.e, perpendicular to primary direction of movement) tend to be stronger indicators of target constraint condition than directional components. Finally, we show that the differing effects on the kinematics of amplitude and directional constraints are observed during the first 70% of the gesture's path.

The paper is organized as follows. First, we explore related work on bivariate pointing. We also describe related work on kinematics of pointing tasks and provide an introduction to HMMs. Next, we describe an experiment designed to gather data on pointing tasks representative of those found GUIs. We then present the results of applying HMMs to understand the impacts of bivariate pointing on motion and discuss their implications. We conclude the paper by outlining future work related to the kinematics of bivariate pointing.

2 Related Work

2.1 Bivariate Pointing

Fitts' Law [7] relates pointing time to target size and distance through a logarithmic term referred to as the *Index of Difficulty* or *ID*:

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

In the above equation, A represents the distance to the target, and W represents the size of the target. In Fitts' original studies, the target size was always limited in the collinear direction, the direction parallel to movement toward the target. Therefore, pointing tasks were performed with amplitude, not directional, constraints. Most GUIs, however, contain targets that have bivariate constraints, constraints in the parallel and/or perpendicular directions relative to the direction of motion. To model pointing time for bivariate targets, research by Hoffmann and Sheikh [12] and by MacKenzie and Buxton [15] proposed using the minimum of target width and height as the actual target size. More recently, Accot and Zhai [2] proposed a more refined model of bivariate pointing. By varying width and height parameters, they reformulated Fitts' Law for bivariate pointing tasks as a weighted Euclidean sum of width and height, specifically:

$$T = a + b\ log_2\left[\sqrt{\left(\frac{D}{W}\right)^2 + \eta\left(\frac{D}{H}\right)^2} + 1\right] \quad (2)$$

Accot and Zhai's work on bivariate pointing was subsequently extended by Grossman and Balakrishnan [9]. Grossman and Balakrishnan proposed a probabilistic formulation of ID that could account not only for differences in target width and height, but also the user's angle of approach and target shape. The authors validated their model for differing angles of approach, leaving target shape for future work.

Fitts' Law's primary use in interfaces is as a predictive model of pointing task time. If there is a desire to predict aspects of performance other than time, models that describe kinematic (as opposed to temporal) characteristics of the pointing task are needed. We now examine models that describe the kinematic profile of movement tasks. While these models can be used to describe the basic motion that occurs in targeting tasks, they do not describe the differences that exist between bivariate pointing tasks.

2.2 Kinematic Models of Pointing

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. For Fitts-style pointing tasks, various models exist that seek to explain observed kinematic profiles.

Psychological research on models of aimed movement can be traced to Woodworth [23]. In examining speed profiles of aimed movements, Woodworth hypothesized that movement toward a target consisted of an initial ballistic movement followed by a sensory control phase to steer onto the target. While this basic view of movement is accepted, research in psychology still seeks details on the exact nature of the ballistic and sensory control phases.

The current accepted model of human movement in aimed pointing tasks is the stochastic optimized-submovement model [17]. Similar to Woodworth's model, this model predicts that targeted motion occurs in two stages: (1) an initial, primarily ballistic, motion that brings a subject close to the final target; and, if necessary (2) a secondary corrective movement to acquire the target. In the stochastic-optimized submovement model, 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.

While the stochastic-optimized submovement model corresponds well with Fitts' Law and experimental data, it does not describe bivariate pointing tasks. There are several open research questions on the kinematics of bivariate pointing tasks, including:

- Which parameters of movement are affected by bivariate constraints?
- When is movement first affected by amplitude and directional constraints?
- Given a bivariate pointing task, can we determine which constraint generated a motion profile using parameters of the motion?
- How similar is the effect of bivariate constraints across users?

To answer these questions, we use a machine learning data analysis tool called Hidden Markov Models to analyze motion paths. We provide a basic description of Hidden Markov Models here. Since a full treatment of this topic is beyond the scope of this paper, the interested reader is referred to [20] for further detail.

2.3 Hidden Markov Models

Hidden Markov Models (HMMs) are a type of graphical model, essentially a probabilistic finite state automaton. HMMs are particularly well suited to learning and classifying sequential data. HMMs were first introduced to the machine learning community in the 1990's by Rabiner [20]. HMMs are commonly used in pen and non-pen based gesture recognition [6, 19, 22]. In HCI research, HMMs have been used for purposes such as recognizing user intent in eye-based interfaces (i.e., interfaces controlled by eye movements) [21] and predicting a user's focus of attention in remote collaborative tasks [18].

At a high level, an HMM can be viewed as a functional mapping of a sequence of observations to a probability. The probability represents the likelihood that the automaton generated the observations. A different HMM is created (trained using labeled data) for each of the possible classifications of the observations. To perform recognition, an unlabeled observation is assigned to the class whose HMM has the highest likelihood of having generated that observation. In the remainder of this section we describe the distinction between a Markov model and a Hidden Markov Model, describe how a HMM classifies information, and describe specifically how HMMs are used in this work.

A Markov model is a graph with nodes and arcs. Each node represents an internal state while each arc represents a transition between nodes. A Hidden Markov Model is an extension of a Markov model where the current internal state cannot be directly inferred

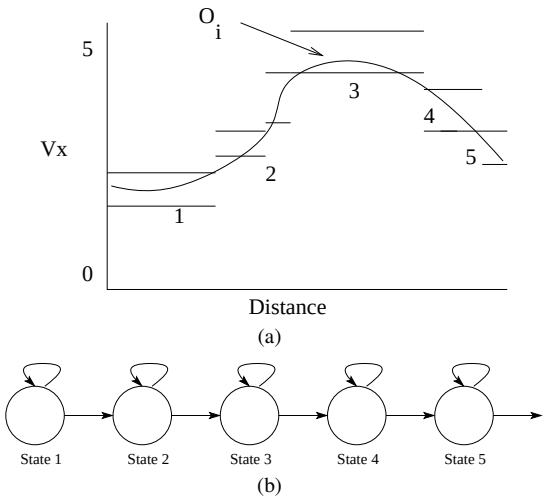


Figure 2: A depiction of a HMM. At the top, the probability distributions for the states within the HMM given two observations, a distance and the x-component of velocity. At the bottom, the topology of the HMM.

from the observation, as multiple states could produce the same observation.

To model a pointing motion, one state will model a specific region of the motion. Together, all the states of a HMM will contribute to the complete pointing motion. This is illustrated in Figure 2. The upper plot shows the pointing motion superimposing the region that each state represents. The lower figure illustrates the path through the HMM that corresponds to the pointing motion. In this situation a particular observation o_i is most probably generated by state 3 as it has a value of $V_x = 5$. Using this rational we can determine the probability that the entire observed sequence was generated by the HMM.

In the context of this paper, continuous Gaussian HMMs are used as follows. First, we learn the parameters of a HMM so that a single HMM, $\lambda_{amplitude}$, models amplitude-constrained motion and a second HMM, $\lambda_{directional}$ models directionally-constrained motion. Second, given an unlabeled motion, O, we can compute the probability that the motion was generated by either $\lambda_{amplitude}$ or $\lambda_{directional}$. We classify the motion as amplitude constrained if $P(O|\lambda_{amplitude})$ is greater than $P(O|\lambda_{directional})$.

3 Experiment

To determine the kinematic effects of bivariate pointing, we designed a data-collection task that captured bivariate pointing data. In this section, we describe the participants, display task, experiment design, apparatus, and the procedure by which data was collected.

3.1 Participants

Eight people, two female and six male, all right-handed, participated in the experiment. All participants were university students.

3.2 Task

The task (displayed in Fig. 3) was a discrete, one-dimensional pointing task. Initially a green starting rectangle was displayed on the screen (shown on the left in Fig. 3). The task began when the participant used the cursor to click within the starting location. At that time, a red target would appear on the opposite side of the display (shown on the right in Fig. 3). Participants were required to move the cursor to the red target and use the mouse button to click on the target. A successful target acquisition (i.e., clicking within the target region) was indicated with the target changing color. Users were told to acquire the target as quickly and accurately as possible, similar to other Fitts' Law tasks.

Figure 3: The experimental task.

3.3 Design

The experiment consisted of a within-subjects design with repeated measures. The independent variables were target ID and the bivariate constraint of the target (amplitude or directional). The IDs ranged between 3.17 and 7.01, and were the result of the 15 distance/width and 15 distance/height combinations (in pixels): 512/4, 512/8, 512/16, 512/32, 1024/8, 1024/16, 1024/32, 1024/64, 1536/12, 1536/24, 1536/48, 1536/96, and 1536/192.

3.4 Apparatus

The experiment was conducted on a generic desktop computer (P4, 2.0GHz) with a 23-inch 1920x1200 LCD display running custom software written in C#. Input was collected using a Wacom Intuos3 five button mouse on a 12x19 inch tablet set to a 1:1 control display ratio. The 1:1 control display ratio ensured that motor space and visual space coincided throughout the pointing task. The tablet was used because of its high sampling rate.

3.5 Procedure

The experiment consisted of eight blocks: one practice block and seven experimental blocks. Each block consisted of 15 D/W combinations presented twice for each constraint, resulting in 60 tasks per block. The order of presentation of the D/W combinations and constraints was randomized. To minimize fatigue, participants were required to take a five minute break between blocks. The experiment took approximately 60 minutes to complete.

3.6 Measures

The custom software captured mouse movements at 200Hz. Movement time, X position, and Y position were captured for each registered mouse movement. Movement time was calculated from when the user clicked the start target to when the user acquired the intended target. To examine the kinematics of motion, we used the position and time information to calculate velocity, acceleration, jerk and curvature for each data point in both the X and Y directions.

4 Preliminary Data Analysis

In this section, we present some initial analysis of the data. The purpose of this initial analysis is to verify that our data agrees with observations of bivariate pointing times by Accot and Zhai [2], and to demonstrate the overlap of the two constraint conditions within individual parameters of motion.

Figure 4 plots movement time against Index of Difficulty for both amplitude and directional constraint. Similar to results reported by Accot and Zhai [2], we see that amplitude constraint (stopping)

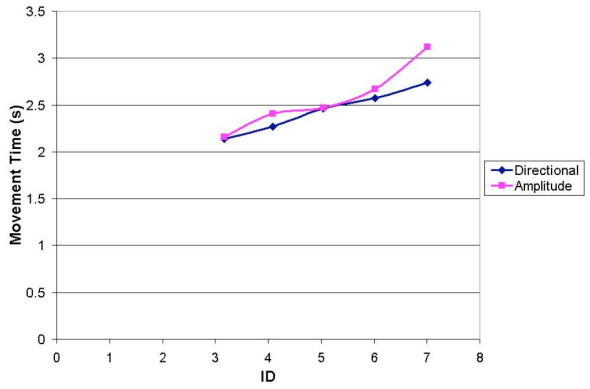


Figure 4: Movement times by target constraint.

typically takes longer than directional constraint. Analysis using repeated-measures ANOVA shows a significant effect for constraint ($F_{1,7} = 23.30, p = .002$) and for ID ($F_{4,28} = 44.230, p < .001$). To examine the interaction between constraint and ID, we used univariate ANOVA.[1] The results indicate that there is a significant effect for target constraint*ID ($F_{4,2891} = 3.13, p < .01$).

Our initial hope was that apparent differences would exist in motion profiles for the different constraint conditions. However, our examination of the kinematic characteristics of motion did not clearly display differences by the two types of constraints. Consider, for example, Figures 5(a) and 5(b), which plot velocity in both the Y and X directions for the two constraints as a function of the percentage of stroke completion. These figures illustrate the degree to which the characteristics of motion overlap regardless of the type of constraint. Plots of other motions characteristics were similar in their lack of observable differences. Likewise, plots with motion characteristics separated by ID showed obvious differences according to ID, however, differences owing to constaint were again difficult to assess based on visual inspection alone.

Despite the lack of readily observable divergences between profiles, the significant effect that constraint has on movement time requires that some variations in motion must be present. To extract from the data those parameters of motion that result in the differences in time for amplitude and directional constraint, we trained HMMs on various combinations of features and use changes in recognition accuracy to determine those parameters affected by changes in target constraint.

5 Results

In this section we present the results of using HMMs to recognize and characterize the differences between pointing motions in bivariate pointing tasks.

5.1 HMM Training

We trained the HMMs using two strategies. The first strategy resulted in what we refer to as *user-specific HMMs*. This strategy, which provided the classifier with access to data from all users during training, employed an 10-fold cross-validation technique. For each fold, the HMM was trained using 90% of the data from all users and then tested on the remaining 10%. This process was then repeated nine additional times, each time using different training/test sets. The second training strategy resulted in *generic HMMs*. With this strategy, data was withheld from one user during training to test the classifier's ability to generalize to new users. In particular, we used 8-fold cross validation, where, for each fold, the HMM was trained on data from seven users and tested on the remaining user.

For user-specific HMMs to be useful for on-line prediction, the system would have to observe the user perform a number of pointing gestures (and have knowledge on target dimensions) prior to reaching the levels of accuracy described in here. The generic HMMs are useful in understanding potential predictive accuracy when the classifier has no knowledge of a given user.

All HMMs had five states, with one mixture per state. The HMMs were also fully connected, meaning that transitions were possible between all states. Before settling on five states, we experimented with other configurations. We found that for zero to five states, accuracy increased as the number of states increased. After five states, there was there was a very slight increase in accuracy. For the purpose of our experiments, however, this increase was not enough to warrant the additional training time.

5.2 HMM Inputs

The inputs to the HMMs are all derived from the position and timing information of the pointing motion. We identify three categories of parameters: instantaneous parameters, path-based parameters, and cumulative parameters. The inputs for each these three parameter categories are summarized in table 1.

Input	Description
	Instantaneous Components (Vectors)
V_x	Velocity w.r.t. X-axis
V_y	Velocity w.r.t. Y-axis
A_x	Acceleration w.r.t. X-axis
A_y	Acceleration w.r.t. Y-axis
J_x	Jerk w.r.t. X-axis
J_y	Jerk w.r.t. Y-axis
	Path-Based Components
κ	Local curvature of trajectory
	Cumulative Components (Scalars)
V_{c_x}	Cumulative speed w.r.t X-axis
V_{c_y}	Cumulative speed w.r.t Y-axis
A_{c_x}	Cumulative acceleration w.r.t X-axis
A_{c_y}	Cumulative acceleration w.r.t. Y-axis
J_{c_x}	Cumulative jerk w.r.t. X-axis
J_{c_y}	Cumulative jerk w.r.t. Y-axis

Table 1: Inputs

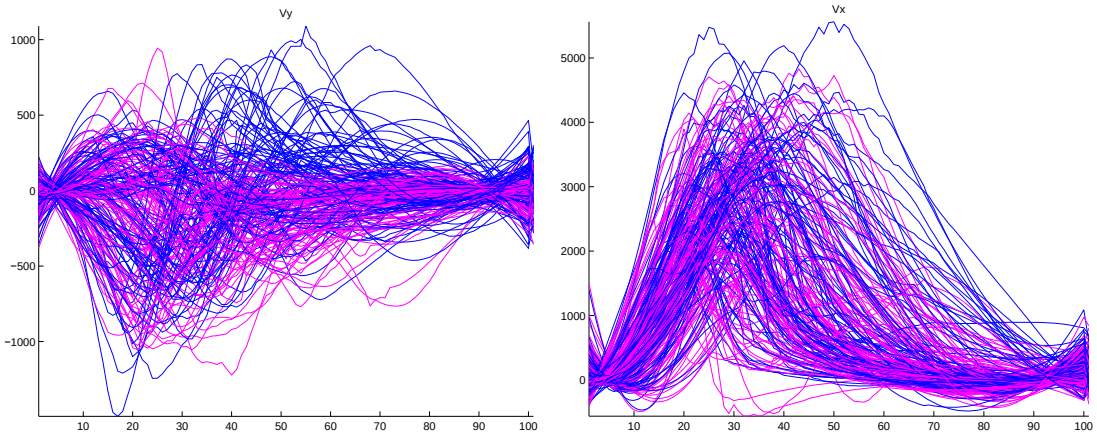
Our rationale for these three categories of parameters is an observation that target constraint can affect motion in three ways. First, the act of either steering or stopping can cause changes in instantaneous components of movement during motion, a result of trying to control one or more instantaneous parameters of movement. As an example, the y-component of jerk might vary more abruptly during motion to keep the trajectory aligned with the target.

Second, a steering constraint could cause the path to bend more than a stopping constraint, a result of trying to steer back to a target. We used local curvature to measure variations in the "straightness" of the motion path. Curvature was calculated using the standard curvature formula:

$$\kappa = \frac{|v_x a_y - v_y a_x|}{(v_x^2 + v_y^2)^{3/2}} \quad (3)$$

Finally, different constraints could result in variations in the overall components of motion. For example, a steering constraint might result in a higher peak speed than a stopping constraint for

[1] With RM-ANOVA, only one within-subjects factor can be included in the analysis at a time.



(a) Velocity in the Y direction

(b) Velocity in the X direction

Figure 5: Velocity in the X and Y directions for the target constraints by percentage of stroke completion. Magenta represents amplitude constraints and blue represents directional constraints. (Requires colour viewing)

a given distance. Cumulative parameters encapsulate information on the entire trajectory. We computed cumulative parameters as the sum of the absolute values of velocity, acceleration and jerk components, allowing us to analyze whether changes in overall motion occurred as a result of amplitude and directional constraints.

5.3 Analysis

HMMs were trained using different combinations of the inputs in Table 1 to determine which inputs contained the most differentiating information. We considered eighteen different combinations of features covering both parameters within the three categories and mixtures of parameters from across the different categories. We selected the eighteen feature sets based on what we felt would be the most informative and also provide a good breadth of coverage of the parameter space. While it would be possible to experiment with additional feature combinations, we did not feel that they would provide the same insight, and both the time necessary to train the HMMs (using the 10-fold cross validation) and the tractability of the analysis were limiting factors.

Our first goal was to determine where, along motion path, observable differences caused by target constraint become apparent. Figure 6 depicts a typical graph of recognition accuracy given partial results. This figure is representative of the results for both the user-specific and the generic HMMs. After having observed only 10 % of the task, as expected, the recognition accuracy is about 52%, or near chance. However, once 70% of the task is observed, accuracy rises to about 73% and remains stable for the remainder of the sequence. This implies that once 70% of the task is observed we can make as accurate a guess as if the task was completed. Consequently, in the remainder of our analysis we consider only the first 70% of the observations. During the final 30% of the motion, no new constraint-specific information can be gleaned from the motion.

Of the three categories of features, we found that instantaneous components of motion were most affected by variations in target constraint. In contrast, target constraint had much less of an impact on the overall components of movement and the path-based parameters. As instantaneous parameters were most affected by bivariate constraint, we present six different combinations of these features. Table 2 indicates the specific cases that will be presented for analysis in the following section. For comparison purposes, we also present representative results from the other two categories (Case 7 and Case 8) and an additional parameter combination that incorporates combinations of instantaneous, path-based, and cumulative

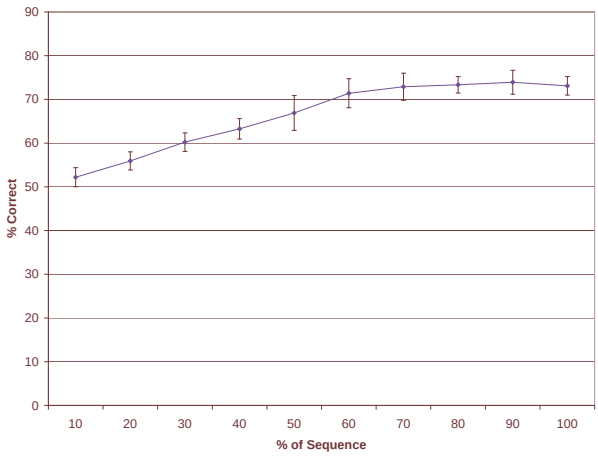


Figure 6: Accuracy given partial observations for a user-specific HMM using instantaneous x- and y-components of motion.

features (Case 9). In the eighteen cases that we examined, a number of cases combined features from the different categories, however, such combinations resulted in no improvement in recognition. In Case 9, we use all available parameters from all categories.

For the combinations listed in Table 2, we now explore classifier accuracy. As noted in Section 5.1, we used two different strategies for training HMMs. The first was to create user-specific HMMs, which are trained on data from a specific user. The second was to create generic HMMs, which analyze information given no prior data on a specific user. Figure 7 presents the results for user-specific HMMs, and Figure 8 presents recognition accuracy for generic HMMs. Errors bars in the graphs denote the standard deviations from the cross-validation. As noted in section 5.1, user-specific HMMs (Figure 7) were evaluated using 10-fold cross-validation, and generic HMMs (Figure 8) were evaluated using a leave-one-out strategy.

In our results, we use HMMs to distinguish those features of motion that are most affected by target constraint. We do this by providing selected features from an unlabeled instance of a constrained pointing task to a trained HMM and asking it to classify whether the

Case	Inputs
	Instantaneous Parameters
1	$V_x, V_y, A_x, A_y, J_x, J_y$
2	V_x, V_y
3	V_x, A_x, J_x
4	V_y, A_y, J_y
5	V_x
6	V_y
	Path-Based Parameters
7	κ
	Cumulative Parameters
8	V_{c_x}, V_{c_y}
	Multi-Category Parameters
9	$V_x, V_y, A_x, A_y, J_x, J_y, \kappa, V_{c_x}, V_{c_y}, A_{c_x}, A_{c_y}, J_{c_x}, J_{c_y}$

Table 2: Description of cases

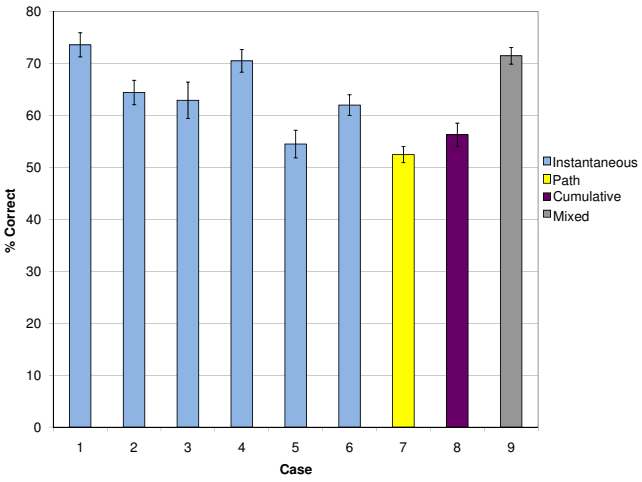


Figure 7: User-Specific HMM Results

sample was generated by an amplitude or directional constraint. As we add additional features, we expect to see the HMM's accuracy increase if the features it uses are affected by target constraint. Accuracy should remain constant (or decrease slightly[2]) if the features are not affected.

Our first result is that the best possible HMM we observed was a user-specific HMM that had access to all instantaneous kinematic components of motion. Under this condition, the HMM can classify target constraint with 73.6% accuracy. This is represented as Case 1 in Figures 7. These results indicate that there are detectable differences between the pointing motions with amplitude constraints and with directional constraints. Furthermore, it is possible to detect this difference on an individual observation (given training data) in real-time with much better-than-chance accuracy.

Figure 8 illustrates the results of training the HMMs under the second scenario, where each HMM is trained on seven users data, then tested on the remaining user. This represents the more difficult recognition task of creating a user-independent model. As expected the recognition rates decreased, with the best case averaging 67.1%, and the standard deviations tending to be larger. Although these recognition rates are lower, they are again much better than chance. The generic results also correlate well with the results from the user-specific HMMs. Furthermore, this result indicates that the difference between amplitude-constrained pointing motions and directionally constrained pointing motions are consistent across multiple users.

[2]Using spurious features frequently causes Machine Learning (ML) techniques to perform slightly worse unless a sufficiently large training corpus is available. With enough training, ML techniques can be trained to ignore spurious features. However, for reasonable training sets, the observation of poorer performance with additional features is sufficient to conclude that the new features contain no useful information.

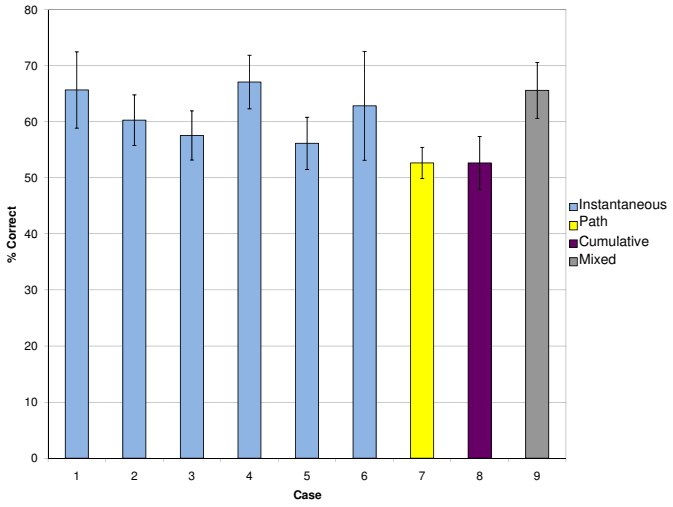


Figure 8: Generic HMM Results

As mentioned above, Case 1 represents the best-case scenario in our results. To isolate which inputs provide the most useful information, we experimented with several different combinations of the features in Case 1, shown in Figures 7 and 8 as Cases 2 - 6. As indicated in Table 2, Case 2 is x and y components of speed, Case 3 is instantaneous x-components, Case 4 is instantaneous y components, Case 5 is the x-component of velocity and Case 6 is the y-component of velocity. The single most important input was velocity with respect to the y-axis (Case 6). Adding acceleration and jerk with respect to the y-axis (Case 4) produced nearly as accurate recognition, 70.5%, as Case 1, indicating that most discriminating information is contained in motion perpendicular to the direction of the target.

Other categories of features were less affected by target constraint. Little discriminating information exists in curvature (52% accuracy, only slightly above chance, shown in Case 7). Cumulative components do perform better than chance (Case 8). However, cumulative components include no new information, as shown in Case 9, where when all components are added to the HMM we observe a slight drop in recognition. If any of the additional components added useful information, we would expect to see an increase in performance for the mixed HMM.

In summary, we have shown that the difference between amplitude- and directionally constrained pointing motions is detectable. We have shown the most discriminating information is encoded in the motion perpendicular to the direction of the target. We have presented evidence that the differences between pointing motions are detectable once only 70% of the motion has been observed, and that these observations are user independent.

6 Discussion

Section 2.2 introduced four questions on the kinematic properties of bivariate pointing:

- When is movement first affected by amplitude and directional

constraints?

- Which parameters of movement are affected by bivariate constraints?
- Given a bivariate pointing task, can we determine which constraint generated a motion profile using parameters of the motion?
- How similar is the effect of bivariate constraints across users?

Our results provide answers to each of these questions.

First, we analyzed various parameters of movement at different positions along the trajectory. We note that after 10% of the trajectory, amplitude and directional constraints have little effect on motion. However, by 20% of movement, we are able to predict constraint with an accuracy that exceeds chance, and this accuracy increases until 70% of trajectory is observed. This indicates that corrective components of motion can occur early in motion, and that they become more pronounced through to 70% of movement. During the last 30% of the trajectory, corrective movement is undoubtedly occurring, but there exists little qualitative difference between instantaneous, path-based, or cumulative characteristics of movement.

Second, based on Figures 7 and 8 we can conclude that most effects of target constraint occur in the instantaneous y-component of movement. Furthermore, we found that path-based and cumulative components of motion are not affected by target constraint.

Third, we note that we can determine the target constraint from parameters of motion 73.6% of the time with user-specific Gaussian HMMs – a result that is much better than chance. Since our focus in this paper is on understanding the underlying kinematics, we have not yet explored ways to improve classifier recognition. Therefore, it is likely that with further attention paid to improving recognition that this accuracy rate will rise. Whether or not such a classifier could achieve an accuracy rate that acceptable from the user's perspective, given an interaction technique designed to leverage these predictions, remains an open question.

Finally, based on the accuracy of the user-specific and generic HMMs, the y-component effects of target constraint generalize across our subjects. However, Case 1 in Figure 7 shows that a small improvement in recognition occurs when x-components of motion are considered in user-specific HMMs. This improvement is not apparent in Figure 8, indicating that the variability in the x-component does not generalize across users.

6.1 Study Limitations

There are two aspects of our study design that might limit generalizability to all interface pointing tasks. First, when exploring bivariate pointing, we focused on instances of bivariate constraint where the target constraint exists in only one dimension. This decision allowed us to isolate the impact of amplitude versus directional constraint. We do not consider the effect that interactions of these constraints have on motion, although a future study exploring this would be valuable. Similarly, in our study, we examined only horizontal motion on the display, mirroring related work in this area [1, 2, 7].

An additional limitation of our study is the manner in which data was supplied to the HMMs. In particular, stroke data was always supplied in chronological order, starting from the beginning of the stroke. With this training method, we found that no new information was gained after 70% of the stroke. However, it might not be the first 70% of the *movement* that is the most discriminating, but rather the first 70% of the *data*. Further experimentation with the manner in which the HMMs are trained would be necessary to resolve this issue.

7 Future Work

A number of recent research results have examined different aspects of modeling in interfaces. Beyond bivariate pointing, examples of this style of work include character modeling [5], goal crossing [1], endpoint modeling with complex targets [10], endpoint prediction [13], and others. This work seeks understanding of the phenomena that underlie such diverse tasks as drawing characters, hitting targets, or moving through regions.

In our work, we focus specifically on understanding which kinematic properties of motion result in the temporal difference observed between amplitude and directional constraints [2]. Given some understanding of the kinematic properties that are affected by bivariate target constraints, our next goal is to understand how we might make use of this understanding in interaction. In this section, we focus on three areas of future work: improving recognition, designing interaction techniques, and studying the effect of error-prone intelligent interaction techniques on users.

7.1 Improving Recognition

In this paper, our overall goal was to understand which parameters of motion were most affected by bivariate target constraint. Given the observation of temporal differences between amplitude and directional constraints by Accot and Zhai [2], some parameters of movement must necessarily have been affected by the constraints. HMMs allowed us to identify those parameters of motion that are most altered by varying target constraint.

While our accuracy results with simple HMMs were greater than chance, several options exist for further improving recognition. These include improving learning or exploring other recognition techniques. Strategies such as "competitive learning", where several competing HMMs are trained on each target constraint might allow an individual HMM to adapt to a particular distance, ID, or other task-specific factor. Other recognition techniques also exist for sequential data, including hybrid HMMs, conditional random fields, or graph transformer networks. Any of these might provide better recognition results.

7.2 Designing Interaction Techniques

A variety of existing interaction techniques can be modified to take into account the primary constraint under which the user is generating motion. Examples of these interaction techniques include oriented target expansion and oriented cursor acceleration.

In their work on expanding targets, McGuffin and Balakrishnan [16] assumed that target expansion occurred uniformly in two dimensions. However, as they note, when interface widgets are densely arranged on the screen, this expansion may cause a large disruption in the display. As well, in the case of tiled display targets, expanding targets provides no benefit to the interaction. To accommodate these special cases, the system might use recognition of user constraint to assign priors to candidate targets based on target profiles. For certain observed parameters, a subset of targets could be expanded, while others could be left unexpanded, and the expanded targets could be expanded only in the dimension of the constraint. This could serve two benefits. First, it would minimize disruption to the display by limiting the number of targets expanded, and limiting the expansion to a single axis. Second, any predictive technique might allow better identification of the specific candidate target, thus improving performance in dense widget arrangements.

Even without manipulating display widgets, other options exist for improving pointing performance. Various researchers (see, for example, [11]) have manipulated the control-display ratio to improve pointing performance while on widgets. As well, the ballistic variation of mouse-pointer movement allows users to traverse longer distances on the computer screen at high speeds than an equivalent displacement would at low speeds (cursor acceleration).

Given an understanding of the orientation of the target on the display, the control display ratios could be selectively altered to enlarge targets in the constrained dimension or to stabilize movement along the constrained axis.

7.3 User Perception and Action

Even with improvement to the classifiers, no on-line prediction mechanism will ever reach perfect accuracy. Any time interaction techniques make use of error-prone recognition technology, there is the potential to harm user performance. However, the computer mouse is already error-prone. Both mechanical and optical mice suffer from occasional tracking problems due to friction or reflective characteristics of the surface on which they operate. Users seem to adapt to these idiosyncrasies during pointer motion.

Given the already error-prone nature of the computer mouse, one open question is whether we can manipulate the interaction through error-prone recognition, distinguish when the recognition operated incorrectly based on user compensation, and then adjust our recognition hypothesis and attendant interaction technique to our new recognition result. This feedback loop might result in similar behaviour to the behaviour exhibited by users dealing with tracking issues in computer mice.

8 Summary

In this paper, we explored the kinematic characteristics of bivariate pointing. While past research by Accot and Zhai [2] observed temporal differences between amplitude and directional constraints, we studied why, in terms of the kinematics of motions, such temporal differences occur.

We performed an experiment to collect bivariate pointing data and found that visual inspection of the data along various dimensions of movement revealed few discriminating characteristics. As a result, we employed machine learning techniques and Hidden Markov Models. Using HMMs, we found that the constraints do create differences in the underlying motion characteristics and that the differences lie in the instantaneous components of motion. In particular, we found the primary effect to be concentrated in motion along the axis orthogonal to the primary direction of motion. Other parameters examined, pertaining to path-based parameters and cumulative components, were found to have little discriminating power. Additional results of interests include the fact that constraint effects can be detected by 70% of the gesture and that the HMMs can predict target constraint at a level significantly better than chance even with no prior knowledge of the current user. Promising avenues for future work include improving classifier accuracy, implementing new interaction techniques that leverage target-constraint prediction, and studying the effects of inaccurate predictions on user behaviour.

Acknowledgements

The authors wish to thank Michael Terry for his comments on earlier versions of this work, our study participants, and the anonymous reviewers for their helpful comments.

References

[1] J. Accot and S. Zhai. More than dotting the i's — foundations for crossing-based interfaces. In *CHI '02: Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 73–80, 2002.

[2] J. Accot and S. Zhai. Refining Fitts' law models for bivariate pointing. In *CHI '03: Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 193–200, 2003.

[3] J. Accott and S. Zhai. Beyond Fitts' law: Models for trajectory-based HCI tasks. In *CHI'97: Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 295–302, 1997.

[4] T. Asano, E. Sharlin, Y. Kitamura, K. Takashima, and F. Kishino. Predictive interaction using the delphian desktop. In *UIST '05: Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 133–141, 2005.

[5] X. Cao and S. Zhai. Modeling human performance of pen stroke gestures. In *CHI '07: Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 1495–1504, 2007.

[6] G. Fang, W. Gao, X. Chen, C. Wang, and J. Ma. Signer-independent continuous sign language recognition based on SRN/HMM. In *Proceedings of the IEEE ICCV Workshop on Recognition*, pages 90–95, 2001.

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

[8] L. Gennaria, L. B. Karaa, T. F. Stahovich, and K. Shimad. Combining geometry and domain knowledge to interpret hand-drawn diagrams. *Computers and Graphics*, 29(4):547–562, 2005.

[9] T. Grossman and R. Balakrishnan. A probabilistic approach to modeling two-dimensional pointing. *ACM Transactions on Computer-Human Interaction*, 12(3):435–459, 2005.

[10] T. Grossman, N. Kong, and R. Balakrishnan. Modeling pointing at targets of arbitrary shapes. In *CHI '07: Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 463–472, 2007.

[11] Y. Guiard, R. Blanch, and M. Beaudouin-Lafon. Object pointing: a complement to bitmap pointing in guis. In *GI '04: Proceedings of Graphics Interface*, pages 9–16, 2004.

[12] E. Hoffmann and I. Sheikh. Effects of varying target height in fitts movement task. *Ergonomics*, 37(6):91–139, 1994.

[13] E. Lank, Y. Cheng, and J. Ruiz. Endpoint prediction using motion kinematics. In *CHI '07: Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 637–646, 2007.

[14] E. Lank and E. Saund. Sloppy selection: Providing an accurate interpretation of imprecise selection gestures. *Computers & Graphics*, 29(4):490–500, 2005.

[15] I. S. MacKenzie and W. Buxton. Extending fitts' law to two-dimensional tasks. In *CHI '92: Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 219–226, 1992.

[16] M. McGuffin and R. Balakrishnan. Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *ACM Transactions on Computer-Human Interaction*, 12(4):388–422, 2005.

[17] D. Meyer, J. Smith, S. Kornblum, R. Abrams, and C. Wright. Speedaccuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action. In M. Jeannerod, editor, *Attention and Performance XIII*, pages 173 – 226. Lawrence Erlbaum, Hillsdale, NJ, 1990.

[18] J. Ou, L. M. Oh, S. Fussell, T. Blum, and J. Yang. Analyzing and predicting focus of attention in remote collaborative tasks. In *ICMI'05: Proceedings of the International Conference on Multimodal Interfaces*, pages 116–123, 2005.

[19] R. Plamondon and S. N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.

[20] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February 1989.

[21] D. Salvucci. Inferrring intent in eye-based interfaces: Tracing eye movements with process models. In *CHI'99: Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 254–261, 1999.

[22] T.-S. Wang, H.-Y. Shum, Y.-Q. Xu, and N.-N. Zheng. Unsupervised analysis of human gestures. *Lecture Notes in Computer Science*, 2195:174–181, 2001.

[23] R. Woodworth. The accuracy of voluntary movement. *Psychology Review*, 3(13), 1899.