

# Predictive Interaction using the Delphian Desktop

Takeshi Asano   Ehud Sharlin<sup>◦</sup>   Yoshifumi Kitamura   Kazuki Takashima   Fumio Kishino

Human Interface Eng. Lab. Osaka University  
Suita, Osaka 565-0871, Japan  
{asano; kitamura;takashima.kazuki;kishino}  
@ist.osaka-u.ac.jp  
+81.6.6879.7752

<sup>◦</sup>Interactions Lab., Dep. Comp. Sci., University Calgary  
Calgary, Alberta, Canada T2N 1N4  
ehud@cpsc.ucalgary.ca  
+1.403.210.9404

## ABSTRACT

This paper details the design and evaluation of the *Delphian Desktop*, a mechanism for online spatial prediction of cursor movements in a Windows-Icons-Menus-Pointers (WIMP) environment. Interaction with WIMP-based interfaces often becomes a spatially challenging task when the physical interaction mediators are the common mouse and a high resolution, physically large display screen. These spatial challenges are especially evident in overly crowded Windows desktops. The *Delphian Desktop* integrates simple yet effective predictive spatial tracking and selection paradigms into ordinary WIMP environments in order to simplify and ease pointing tasks. Predictions are calculated by tracking cursor movements and estimating spatial intentions using a computationally inexpensive online algorithm based on estimating the movement direction and peak velocity. In testing the *Delphian Desktop* effectively shortened pointing time to faraway icons, and reduced the overall physical distance the mouse (and user hand) had to mechanically traverse.

**ACM Classification:** H.5.2 [User Interfaces]: Graphical User Interfaces (GUI), Interaction Styles; I.3.6 [Methodology and Techniques]: Interaction Techniques.

**General terms:** Design, Human Factors

**Keywords:** Mouse, cursor, prediction, desktop, Graphics User Interfaces (GUI), Windows Icons Menu Pointer (WIMP).

## INTRODUCTION

PC's display size and resolution are constantly on the rise. Recent statistics show that most web surfers are using display resolution of 1024x768 and screen size of 17" [24]. This trend can be expected to continue with larger, higher resolution display systems becoming more accessible and less expensive. With this extended display real-estate, bigger physical size and larger number of pixels, the common Windows-Icons-Menus-Pointers (WIMP) interaction para-

digm can become a rather difficult spatial task to handle, requiring more space (physical mouse movements) and time to fulfill and overall burdening the user.

A common interaction environment which simply exemplifies this problem is the desktop. Desktops often become crowded with many icons spread over the large physical size and resolution they offer. In such setting pointing can become an involved task, especially when dealing with long distances between the current cursor position and the target icon. Users who want to click on a distant icon need to go through a lengthy repositioning phase when the cursor is being tediously relocated from one side of the desktop to the other.

This scenario calls for the introduction of spatial awareness and spatial prediction mechanisms. During cursor repositioning the user is making an intentional spatial action, moving the cursor from one location—the start point, to an unknown goal location—the target icon. Since the interface is fully aware of the cursor's start point, the desktop possible target points and the cursor trajectory why not try and guess where the user is heading?

Several previously published methods exploit cursor position and trajectory to reduce pointing time [2]. Examples range from increasing the size of objects in proximity to the pointer [14,23] or making the cursor jump to a target that is in the direction of its movement [8]. Generally, these methods use static relationships based on the distance or the direction between possible targets and the current position of the cursor. Therefore objects that are simply close to the pointer, or positioned in its direction of movement will be chosen as probable targets, often erroneously and especially so when dealing with long distances to the targets.

The *Delphian Desktop* (named after the Greek Mythology oracle of Apollo at Delphi) enables predictions of user pointing goals over large distances in a robust and precise manner which makes it ideal for large and dense interactive displays. The *Delphian Desktop*'s predictions are based on observations of human spatial perceptual-motor processes in pointing tasks taken from previous psychology and kinesiology research [1,5,6,12,13,18,19]. These observations point to the intuitive finding that users will move the mouse faster to reach faraway objects.

The *Delphian Desktop* predicts the user spatial goal by constantly tracking the cursor location and trajectory at-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST'05, October 23–27, 2005, Seattle, Washington, USA.  
Copyright ACM 1-59593-023-X/05/0010...\$5.00.

tempting to guess whether an overall movement peak velocity value was reached. Once the algorithm determines that the cursor trajectory, and hence the user's hand, reached peak velocity it will use this value along with the movement direction to spatially predict the user's goal. The prediction can then be used as a shortcut to a candidate target icon, allowing the user to automatically or manually jump to the goal location. Alternatively the prediction can allow the system to prepare for the coming action, for example by executing a target application in the background, starting it up even before the expected click.

In the following sections we review some of the common models for the way humans perform positioning and pointing tasks, and the current efforts of introducing prediction to these tasks. We then detail the design and implementation of the *Delphian Desktop*, and describe the findings from a rigorous comparative experimental evaluation of its effectiveness.

### PREDICTION IN POINTING TASKS

Pointing tasks underlie the WIMP interaction paradigm, and are performed frequently and extensively as an integral part of most interactive activities. Consequently, if the time required for pointing tasks can be reduced the usability of WIMP interaction techniques can be enhanced and the overall burden on the user can be reduced; this potential prompted considerable number of research efforts. McGuffin et al. [14] and Zhai et al. [23] suggested a method which reduces the completion time of pointing task by expanding the size of a target as a pointer approaches it. This approach effectiveness was demonstrated both in studies and commercially, but the method falls short of providing any benefit until the cursor is in proximity of the target. Thus, as the distance from pointer to target increases the algorithm's ability to reduce pointing time decreases.

Guiard et al. proposed a method in which the pointer jumps directly from one icon to the adjacent one according to the pointer movement direction [8]. In this method the cursor moves normally over an object but when it is about to leave the object the cursor jumps to the nearest object following the cursor trajectory. The object to which the cursor jumps is determined solely by the movement direction, and the nearest neighbor is always picked. Therefore, with a number of objects situated in the same direction from the cursor it is likely that the algorithm can "jump" to non-target objects.

Blanch et al. proposed a semantic pointing method in which the control-display (C-D) ratio changes according to the distance to the target [3]. The physical movements of the pointing device will translate to faster cursor movements when the pointer is far from a potential target object and slower movements when the pointer is closer to, or over an object. Overall this method will reduce pointing time but in tasks that involve a faraway target with several non-target objects distributed along the pointing path to it, the algorithm will unnecessarily slow down on each of the non-target objects, increasing the overall pointing time.

Worden et al. proposed area cursors and sticky icons in order to assist elderly users [21]; this idea was further improved by Bubble Cursor [7]. Another example is Magic Pointing that uses eye-gaze to predict the target [22]. Not related to pointing tasks in WIMP interfaces, considerable studies have been conducted on predictive user interfaces: Eager [4], Adaptive Action Prompting [10], Split Menu [17], and the concept of efficient traversal of cascading menus [9] are some examples. Several of these ideas are used in off-the-shelf software packages.

### PERCEPTUAL-MOTOR ASPECTS OF POINTING

In the fields of psychology and kinesiology, extensive research on the movement of the hand in the real world has been carried out in order to understand human spatial perceptual-motor processes. Over 100 years ago [20] it was first advocated that the characteristic features of the movement of a hand from an initial position to a target consisted of two processes: initial feedforward movement (or *plan* phase) and error-corrective feedback movement (or *adjustment* phase). The initial feedforward movement indicates an early movement planned according to the distance to a target while the error-corrective feedback movement is a deceleration correction based on real-time control according to information acquired by means of visual feedback. This fundamental observation was verified by many experiments (see for example [16]).

Recently, pointing tasks are often analyzed based on the above characteristic features. Walker et al. [19] examined typical cursor velocity profiles during pointing tasks and showed (Figure 1) that the magnitude of the peak velocity during feedforward movement in mouse-based pointing tasks increases with the distance from a start point to a target.

Based on these characteristics, MacKenzie et al. demonstrated the effects of the width of the target and the distance to a target on the pointing movement in real space [11]. For example, the time of the initial feedforward movement and magnitude of the peak velocity of a hand increases with increasing distance to the target, regardless of the width of the target. Moreover, the time of error-corrective feedback movement increases as the target distance becomes longer or the target width becomes smaller. These characteristic features were also examined in a pointing task [5,6]. In a separate effort we conducted our own investigation (see the following Section) to examine the processes of feedforward movement and error-corrective feedback movement, and suggested the possibility of using the magnitude of the peak velocity to estimate the target position [18].

### DISTANCE TO TARGET AND VELOCITY

We conducted a preliminary experiment to clarify and validate the relationship between the cursor's peak velocity and the distance to the target in pointing tasks [18]. In the experiment we measured and analyzed this relationship in tasks which involved pointing to virtual objects. The experiment was conducted with six male participants. A cubically shaped controller (5 cm/edge) equipped with a six

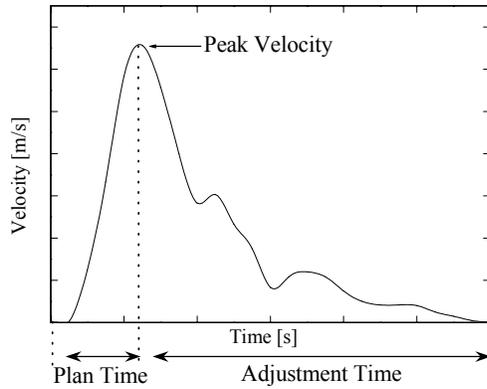


Figure1: Velocity profile during pointing task

degrees of freedom tracker was used to manipulate a square cursor on a virtual two-dimensional (2D) space shown on a display. The controller movement is directly mapped, in 1:1 ratio, to the cursor position. Participants repeatedly attempted to point with the cursor onto the target displayed in the virtual 2D space. Task conditions were as follows:

- The initial position of the cursor relative to the target was preset to create five movement directions to the targets (-90, -45, 0, 45 and 90 degrees in the clockwise direction, where 0 degree means the movement direction was from the bottom of the screen to the top of the screen).
- The initial position of the cursor relative to the target was preset to create three distances to targets (5 cm, 10 cm and 20 cm).

The cursor velocity was calculated from the difference in position data measured by the tracker. Peak velocity was calculated for each pointing task and then compared to the distance to the target and to the movement direction. Figure 2 shows the effect of change in target distance on the average peak velocity. Figure 3 shows the effect of changes in target distance and movement direction on the average peak velocity. Figure 2 demonstrates a linear relationship between the peak velocity and the target distance. Figure 3 shows that the peak velocity is sensitive to the movement direction and that the velocity increases as the movement direction gets closer to horizontal. These preliminary findings confirmed that the peak velocity of the cursor in pointing tasks can be used to try and predict the distance to the target.

### THE DELPHIAN DESKTOP

We propose a simple method for introducing prediction to pointing tasks in desktop environments. The *Delphian Desktop* is an attempt to help reduce the pointing time to distant icons using real-time spatial prediction of the user actions. We see the introduction of spatial prediction to the WIMP environment as an obvious step: the user past and current actions, manifested through mouse movements, are completely known; so are the user possible intentions, manifested by the spatial location of icons and other desktop entities. The *Delphian Desktop* is an attempt to inte-

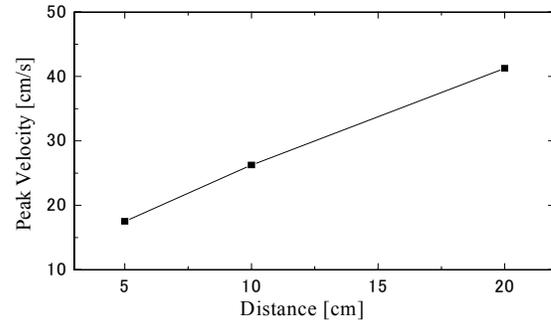


Figure2: Pointing peak velocity vs. target distance

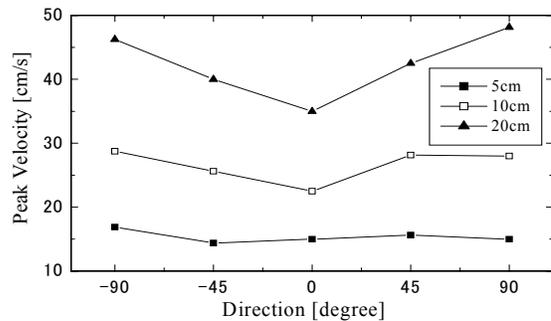


Figure3: Pointing peak velocity vs. target distance and movement direction

grate this existing knowledge in order to create, in real-time, an informed guess on the user pointing intentions.

### Simple Prediction Algorithm

Spatial prediction can be performed in many ways. We chose a simple algorithm for our prediction based on the linear peak velocity-distance to target relationship. Our algorithm for predicting the user's spatial intentions is based on the direction and peak velocity of the mouse trajectories. We hypothesize that when moving long distances between cursor and target icon the following conditions are valid:

1. The user (roughly) follows a direct line to the target.
2. The distance the user is attempting to cover has a linear relationship to the peak velocity of the movement trajectory:

$$D = a \cdot PV + b \quad (1)$$

where  $D$  is the distance between the cursor start point and the target icon,  $PV$  is the peak velocity, and  $a$  and  $b$  are constants.

We expect the constants  $a$  and  $b$  to be dependant on the direction between start point and target and to be different from user to user. We assume the prediction will be valid only in case of a single continuous mouse movement that involves long distances between start point and target.

Our algorithm is arguably too simple; however, we believe that basic pointing prediction in desktop environment can actually be quite straightforward. Also, a simple spatial tracking and prediction algorithm can be easily integrated

into existing desktop environments since it requires very little computing resources for achieving good real-time performance.

### Implementation

Our current implementation of the *Delphian Desktop* was designed to be a proof-of-concept test-bed. We wanted to be able to test and demonstrate that spatial prediction of user actions can be useful in a desktop environment, especially when distances between the current location of the cursor and the target icon are large.

The environment was designed to support three main functions:

1. capture base-line, non-prediction interaction parameters: how long does it take to point to icons on a desktop without prediction? This data can be later used to see if activating our prediction algorithm benefits the overall interaction time or not.
2. calibrate the prediction algorithm for each user. We used linear regression analysis for calibrating the prediction algorithm for each user, and for each spatial direction.
3. predict user spatial intentions following the calibrated prediction algorithm, and measure whether the prediction condition saves time compared to the non-prediction condition.

The basic environment we designed for demonstrating and evaluating the *Delphian Desktop* is presented in Figure 4. The environment includes a cursor, a target icon and a progress bar. The task was always to move the mouse from its current position and to correctly point and click the icon target. In the non-prediction condition the user simply had to move the cursor and click the icon presented on the screen. The calibration phase was similar to the non-prediction phase with the user, again, moving the cursor and clicking on the target icons. The results from the calibration phase were used for the linear regression analysis. The prediction phase used the linear regression results for online prediction. In this phase the trajectory of the mouse was continuously tracked for direction and velocity. The direction was used to choose the corresponding calibration coefficient from the regression analysis results, that is, the direction-related values of  $a$  and  $b$  from equation 1. The velocity was used for determining whether a peak velocity was reached. Once the algorithm determined that the velocity reached a peak value which is higher than a pre-determined threshold, the system performed a “jump”. During a “jump” the cursor was moved along the tracked direction, in the value of equation 1’s  $D$ . When “jumping” action occurred, it was supported graphically by a chime and a fast line animation drawing that briefly connected the original position of the cursor before the “jump” to its position after the “jump”. The system did not take into account the fact that it “knows” where the target icon actually might be and the user still had to manually finish the task of pointing and clicking the target icon. Note that although the

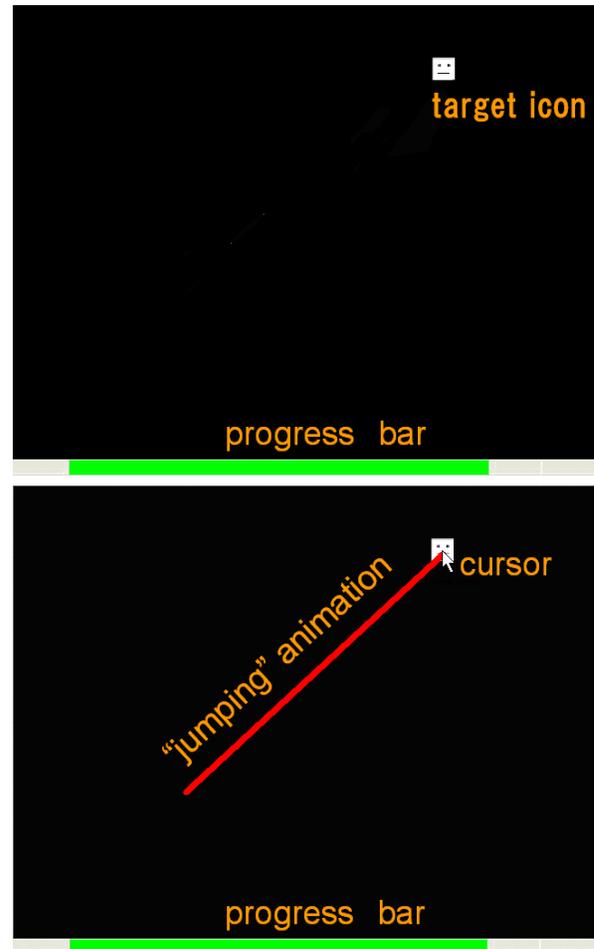


Figure 4: *Delphian Desktop* prediction. A target icon on the desktop (top), the cursor “jumps” to the predicted location with a brief animated motion sequence (bottom).

experimental system described in this paper does not snap the cursor to the closest target, in the non-experimental *Delphian Desktop* prototype the cursor jumps and snaps to the most probable icon. The test system allowed us to flexibly experiment with any desired variations of tasks. For any such mixture the progress bar at the bottom of the screen was used to indicate to the user how many pointing tasks remain before a certain experimental phase is complete.

### EVALUATION

This Section details the *Delphian Desktop* evaluation methodology, a preliminary pilot study we performed, and the final comparative study we conducted to evaluate the *Delphian Desktop* effectiveness.

#### Setup

Figure 5 presents an overview of the experimental setup. We used a DELL Dimension8300, Pentium4 3.2GHz, DDR-SDRAM-1GB RAM, PC as the experimental platform. The graphical environment was generated by OpenGL on Microsoft Windows XP Professional and presented on a 19-inch TFT LCD display with 1280x1024 pixels resolution. A set of stereo speakers were used for sound

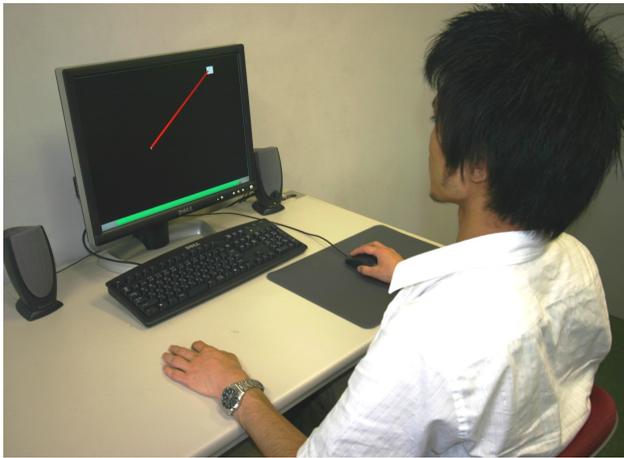


Figure 5: Experimental setup

feedback. The graphical environment was controlled by an optical wheel mouse (Logitech M-BJ58) with an extra large 408x306mm mouse pad (Power Support's Airpad Pro III, AK-07) in order to enable the user to easily perform a set of continuous mouse-movements with less chances of reaching the edge of the mouse pad and of needing to physically lift up and relocate the mouse during pointing tasks.

Participants sat on a chair with adjustable seat height (from 42 to 51 cm), with the equipment placed on a desk with a surface height of 67 cm directly facing them. The chair seat's height was adjusted so that the participant's elbows were at the same height as the desk with the mouse pad situated at a comfortable working position. The experimental environment was implemented using Microsoft Visual C++ 6.0, and the C-D ratio was set to a constant value of 0.5 throughout the experiment.

### Methodology

All the *Delphian Desktop* experiments were conducted following a strict written protocol which was read out loud to the participant by the experiment administrator. The participant was introduced to the system and interviewed quickly answering questions concerning age, education and previous experience in using a WIMP interface.

Each *Delphian Desktop* experiment consisted of two phases: calibration phase and prediction phase. In each phase the participant repeated the task of moving the cursor from a given starting point to a target icon and clicking on it, in different distance and direction settings, with prediction either turned off or on. Each phase was divided to a thorough practice session, which was not used for evaluation, and to an actual session whose results were recorded and analyzed. In both the calibration and prediction phases the participant was told to complete the task by clicking on a target icon as accurately and as fast as possible.

In each task of the calibration phase the cursor velocity was recorded and the pointing task peak velocity was calculated from the recorded data. After the tasks were completed by a participant, a linear regression analysis was done on the

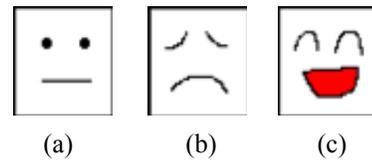


Figure 6: Target Icons, from left: (a) waiting to be clicked, (b) clicked outside (c) clicked inside

peak velocity and target distance data in order to extract the linear regression variables for each direction. The regression variables obtained in the calibration phase were later used to predict the distance to the target in the prediction phase.

Although each pointing task was simple, the experiment as a whole was extremely repetitive. In order to avoid boredom and keep the participants focused on the task at hand several simple computer game-like techniques were integrated. First, the icons were animated with different shapes and facial expressions. During the experiment, the target icon shows one of the three faces presented in Figure 6. The face in Figure 6 (a) is displayed on the target icon under normal conditions. If a participant clicks on the target icon correctly, the face transforms to that in Figure 6 (c), and a sound plays to signal success. If a participant misses and clicks outside of the target icon the face changes to that in Fig. 6 (b) and a failure sound plays. The progress bar at the bottom of the screen (Figure 4) provides overall indication on the experiment progress and becomes shorter after each completed pointing task. In an attempt to reduce repetitiveness we used a game-like level based experimental environment, with each of the levels containing its own set of three iconic faces, success and failure sounds and progress bar color. After a set of tasks is completed the set of faces on the icon, the set of sounds, and the color of the progress bar all change to a different predefined set in order to keep the participant interested and focused on the repetitive pointing task.

In each phase, there are 40 different task flavors consisting of any combination of eight directions in multiples of 45 degrees x five different distances. The participant might repeat each of these different task setting flavors according to the experimental phase.

The size of a target icon is 50 x 50 pixels. The target icon is selected according to the experimental phase and task status and placed in a predefined location on a 13 x 10 grid of possible target locations, designed by dividing the 1280 x 1024 pixel display to 100-pixel spacing.

On the display, the direction from bottom to top is treated as 0 degree, and positive angles are measured clockwise. For 0-, 90-, 180-, and 270-degree tasks, the target location grid supports distances of 500, 600, 700, 800 and 900 pixels between cursor start point and target. For 45-, 135-, 225- and 315-degree tasks, the target location grid supports distances of 707, 848, 989, 1131 and 1272 pixels between cursor start point and target, in order to allow the tasks to cycle along alternating sets of straight and diagonal lines. A

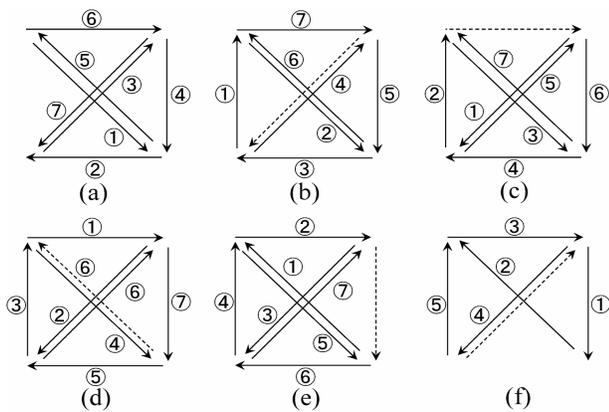


Figure 7: Task presentation preorder

task is completed when a participant single-clicks on a target icon. When a participant clicks on a target, a new target is displayed at a position corresponding to the next task condition. For the next task the previously clicked target position becomes the new start position, allowing easy recognition of the current cursor position.

In order to maintain a cycle of tasks that will cover the different combinations of distance and direction continuously so the participant always start a new task from the target location of the previous one we chose to preorder the tasks spatial layout. The tasks order we used is presented in Figure 7. The presentation order of tasks is shown in six figures from (a) to (f). Tasks are presented in ascending order of circled numbers. Subfigures 7 (b) to (f) present the sequence in the previous subfigure by broken lines for clarity. When the cycle of tasks shown in Figure 7 (a) to (f) is completed, 40 different task conditions are executed exactly once. In the experiment a set of 40 tasks are presented in the order shown in Figure 7 and then the next set of tasks are presented in exactly the reverse order. Due to the shrinking and expanding nature of the task layout and the constant change in directions, all participants, with the exception of one of the pilot study participants, failed to recognize the pattern and reported that the tasks were randomly ordered.

### Pilot Study

A preliminary pilot study was conducted on five male participants (ages ranging from 26-36, average age of 28.2 years) and a 27-year-old female participant. The calibration phase consisted of a single set of practice tasks, 16 sets of actual tasks, and the prediction phase consisted of 3 sets of practice tasks and 8 sets of actual tasks (40, 640, 120 and 320 trials, respectively). There was a five-minute break after each experiential phase.

### Lessons from pilot study

Pilot study participants complained that the experiment phases were too long and tedious to complete consecutively. Following we decided to divide each phase into two parts and ask the participant to rest for three-minute between the two phases. Our initial plan was to execute a separate phase that will not be used for calibration but rather for base-line estimation of the participant performance without predic-

tion. However, following the pilot study results we concluded that the burden on the participant would be too great and decided that the calibration phase would be used both for calibration and as the experiment's non-prediction phase.

This choice has an interesting effect on the way the *Delphian Desktop* results should be interpreted. In the calibration phase, if a participant clicks on a non-target location or stops moving the cursor for 100 ms or longer, the cursor is returned to the starting position for the current task and the task is repeated. This condition is necessary for proper calibration but also implies that the base-line we use for the non-prediction phase would be close to ideal, since tasks where the participant was either slow or imprecise would be excluded and then repeated. This does not apply to the prediction phase where participants can be slow and imprecise, resulting in longer overall time for task completion. Furthermore, because the calibration phase is conducted first, a participant may experience fatigue while conducting the prediction phase. These conditions all balance in favor of the non-prediction phase. Therefore, if the data in the prediction phase demonstrates better user performance than those reflected by the data in the non-prediction phase, these conditions could further support the effectiveness of the *Delphian Desktop* prediction.

While most pilot study participants found the *Delphian Desktop* to be very helpful, one complained about the entire concept of a predictive "jumping" cursor, and another complained that the prediction reduced performance when dealing with arcs and other non-direct movements. However, we were encouraged by the overall analysis of the pilot study preliminary results which indicated that the *Delphian Desktop* was shortening the time required for task completion relative to the non-prediction condition.

### Full Study

A full study of the *Delphian Desktop* was conducted on eight male participants (average age: 23.0 years) and eight female participants (average age: 25.8 years). All participants were recruited from the Osaka University Suita campus and were all experienced in WIMP interaction. The experiment started with calibration phase consisting of three sets of practice tasks, a short break, 8 sets of actual tasks, a short break and another 8 sets of actual tasks (120, 320 and 320 trials, respectively). The participant was then asked to take a five-minute break and to start the prediction phases which consisted of three sets of practice tasks, a short break, 8 sets of actual tasks, a short break and another 8 sets of actual tasks (120, 320 and 320 trials, respectively). An entire experiment session took around an hour to complete.

### RESULTS

A two-way ANOVA by with/without prediction and by target distance was conducted on the full study data. We found that the effects of with/without prediction ( $F(1, 300) = 12.593, p < .001$ ), target distance ( $F(9, 300) = 23.893, p < .001$ ) and the interaction between with/without prediction

Table 1: ID of each task

$D(\text{pixel})$	$W, H(\text{pixel})$	$ID(\text{bit})$
500	50	3.92
600	50	4.16
700	50	4.38
707	50	4.39
800	50	4.56
848	50	4.64
900	50	4.72
989	50	4.85
1131	50	5.04
1272	50	5.21

and target distance ( $F(9, 300) = 5.626, p < .001$ ) were significant.

Figure 8 shows the movement time and standard deviation relative to target distance in the calibration phase, used as the non-prediction phase (“non-prediction phase” hereafter), and in the prediction phase. Figure 8 shows that for short target distances, such as 500, 600 and 707 pixels, the movement time in the prediction phase is longer than that in the non-prediction phase.

In order to analyze the distance threshold value where the prediction was effective, we conducted a two-way ANOVA by with/without prediction and by target distance, changing the range of the target distances. Figure 9 shows ANOVA p-values of interaction between with/without prediction and target distance relative to the range of the target distances. For a target distance range of 800 pixels or longer there are no significant differences in the interaction of with/without prediction and the target distance ( $p > .30$ ), and thus only the effect of with/without prediction shows a significant difference.

Table 1 details the index of difficulty (ID) of each task condition and Figure 10 shows the relation between ID and movement time. Note that ID is computed by using Equation (2) [1], a model proposed for the 2-D pointing task.  $W$ ,  $H$  and  $D$  stand for target width, target height and target distance, respectively; the value of  $\eta$  depends on the pointing device, for our analysis we used  $\eta=1$ ; a larger ID value indicates a more difficult task:

$$ID = \log_2 \left( \sqrt{\left(\frac{D}{W}\right)^2 + \eta \left(\frac{D}{H}\right)^2} + 1 \right) \quad (2)$$

Table 1 and figure 10 show that for a task with a 4.4-bit or higher ID, pointing with prediction is more effective than pointing with non-prediction. In the prediction phase, the actual distance a cursor moves, excluding the “jump” can be calculated as the sum of the distances from the task start position to the cursor “jump” origin and from the cursor “jump” destination to the target. This value corresponds to the actual physical distance the mouse and thus the partici-

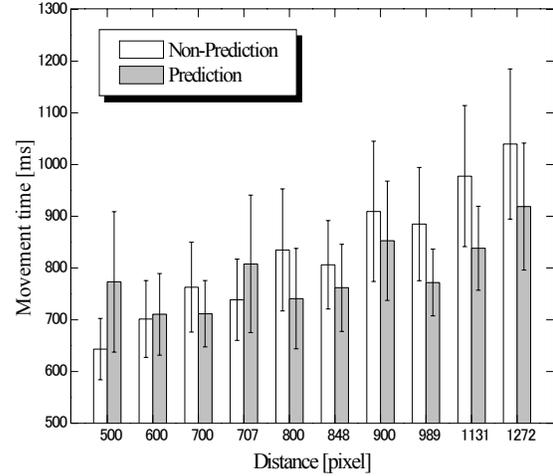


Figure 8: Movement time relative to target distance

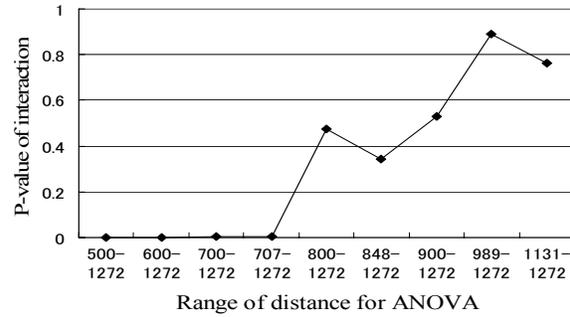


Figure 9: P-value of interaction relative to with/without Prediction and Target Distance Movement time relative to target distance

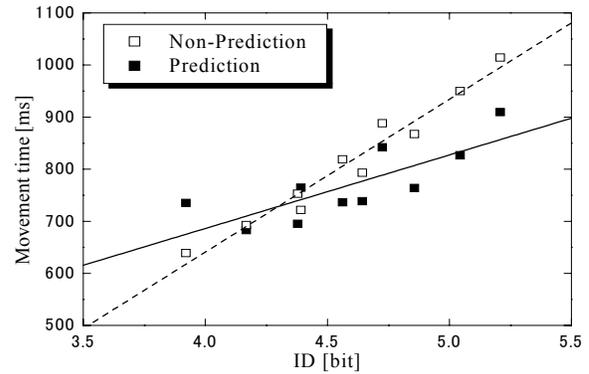


Figure 10: Movement time vs. ID

part’s hand moves. Figure 11 shows a gain value calculated as the ratio of this distance and the distance between origin and target. The mean of the gain value was 1.21.

Figure 12 shows the movement time relative to the error between the target direction and the direction predicted when detecting the peak velocity. The figure shows that as the error increases, the movement time also increases.

The *Delphian Desktop* errors were analyzed by examining error rates or the average number of times the participant

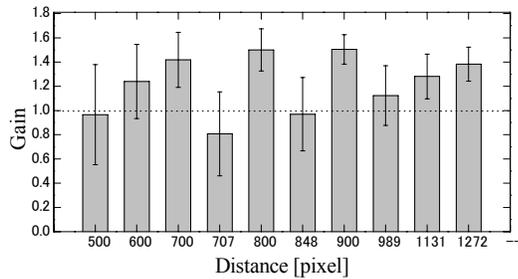


Figure 11: Gain relative to target distance

missed and clicked outside a target; distance error ratios, calculated by the average absolute value of the distance error in pixels divided by the distance to the target; and the average absolute angular error. All errors were calculated over all tests and again for all tests with long distance to target (>800 pixels). We found the overall average error rate to be 6.59% overall and 6.37% for long distances (note that this error rate exceeds only slightly the 4% typical Fitts's Law error [7], suggesting that the influence of the "jump" on the error rate was not severe). The overall average absolute values distance error ratio was 15.4% overall and 17.9% for long distances. The general distance error tendency was for undershoot error. Average angular errors were 3.89 degrees overall and 3.47 degrees for long distances.

## DISCUSSION

The full study confirmed that based on pointing time the *Delphian Desktop* provides effective predictions of users spatial intentions when trying to point to faraway targets. Figures 8 and 9 show that the *Delphian Desktop* predictions are effective when distances to target are 800 pixels or more. The predictions were not effective for distances shorter than 800 pixels. When the cursor "jumps", it is necessary to ascertain the actual time to "jump" and the time the participant recognizes the cursor "jump". We believe that these additional times are significant for short distances. Figure 10 shows that the slope of the movement time relative to ID is smaller for prediction than for non-prediction. Thus, our prediction method will be more effective for more distant targets.

In Figure 11, the directions of 500-, 600-, 700-, 800- and 900-pixel tasks are vertical or horizontal, while the directions of 707-, 848-, 989-, 1131- and 1272-pixel tasks are diagonal. The gains for the diagonal tasks are smaller than those for the vertical or horizontal tasks. This is presumably due to the fact that the error between the target direction and the "jump" direction is larger for the eight diagonal tasks than for the vertical or horizontal tasks; therefore, the total cursor movement distance for the diagonal tasks is longer than that for the vertical or horizontal tasks.

For both vertical/horizontal tasks and diagonal tasks, the gain increases as the target distance increases. The reason is that the longer the target distance, the longer the "jump", which decreases the effect of the increased movement dis-

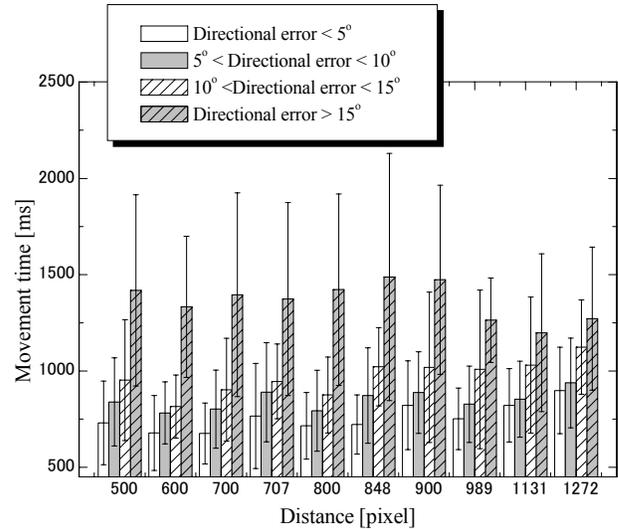


Figure 12: Movement time relative to error between target direction and predicted direction

tance due to the error in the target direction and the "jump" direction. Furthermore, since the mean gain is 1.21, exceeding 1, the *Delphian Desktop* prediction technique is expected to be effective in reducing the mechanical mouse and hand movement and overall in reducing the burden of mouse operation.

Did our experimental design compromise accuracy for the sake of speed? We believe not. Participants couldn't move to the next task unless they correctly pointed to and clicked on the target icon. Misusing the *Delphian Desktop* predictions for quick imprecise jumps should have resulted in overall slow performance in testing.

The *Delphian Desktop* prediction method for reducing pointing time over long distances does not conflict with the expanding target techniques [14, 23] or the C-D ratio adaptation technique [3]. Therefore, the total pointing time may be further reduced by first applying the *Delphian Desktop* prediction in cases of high peak velocity and when reaching the proximity of the target applying more local techniques such as expanding targets or adaptive C-D ratio.

## CONCLUSION

The *Delphian Desktop* spatially predicts user intentions in a desktop environment by analyzing cursor trajectories. We presented our current implementation based on a simple peak velocity-based predictive algorithm, the evaluation test-bed we designed and the results of a user study we conducted. In testing the *Delphian Desktop* was effective in reducing pointing time for 800-pixel or longer target distances, and in reducing the distance of physical mouse movement the user is required to perform during pointing tasks. We believe that that the *Delphian Desktop* can be instrumental in easing mouse interaction burden when using high-resolution, large display systems.

We currently have a working demo that uses the *Delphian Desktop* on a regular desktop interface and we are trying to

find an elegant method for subtle self-calibration of the prediction algorithm.

In the near future we are planning to improve the *Delphian Desktop* direction prediction and to compare our simple prediction method to other methods such as the Kalman Filter. We would also try to extend the *Delphian Desktop* functionality to different environments that might benefit from spatial prediction, like tabletop and large projection-based displays.

#### ACKNOWLEDGEMENTS

This research was partially supported by the 21st Century Centers of Excellence, Ministry of Education, Culture, Sports, Technology and Science (MEXT) Japan, by Grants-in-Aid for Scientific Research Category S, Japan, by Strategic Information and Communications R&D Promotion Programme (SCOPE), by the Canadian Natural Sciences and Engineering Research Council (NSERC) and a University of Calgary startup grant, Canada.

#### REFERENCES

1. J. Accot and S. Zhai. "Refining Fitts' law models for bivariate pointing," In Proc. of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 193-200, 2003.
2. R. Balakrishnan. "Beating Fitts' law: virtual enhancements for pointing facilitation," International Journal of Human-Computer Studies, Vol. 61, Issue 6, pp. 747-922, 2004.
3. R. Blanch, Y. Guiard and M. Beaudouin-Lafon. "Semantic pointing: improving target acquisition with control-display ratio adaptation," In Proc. of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 519-526, 2004.
4. A. Cypher. "Eager: programming repetitive tasks by example," In Proc. of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 33-39, 1991.
5. E. Graham, C. L. MacKenzie. "Pointing on a computer display," in Proc. of Conference Companion on Human Factors in Computing Systems (ACM SIGCHI), pp. 314-315, 1995.
6. E. Graham, C. L. MacKenzie. "Physical versus virtual pointing," In Proc. of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 292-299, 1996.
7. T. Grossman and R. Balakrishnan. "The Bubble Cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area," In Proc. of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp.281-290, 2005.
8. Y. Guiard, R. Blanch and M. Beaudouin-Lafon. "Object pointing: a complement to bitmap pointing in GUIs," In Proc. of Graphics Interface, 2004.
9. M. Kobayashi and T. Igarashi. "Considering the direction of cursor movement for efficient traversal of cascading menus. In Proc. of Symposium on User Interface Software and Technology (ACM UIST), pp. 91-94, 2003.
10. T. Kuhme. "Adaptive action prompting -- a complementary aid to support task-oriented interaction in explorative user interfaces," Technical Report #GIT-GVU-93-19, Georgia Institute of Technology, Department of Computer Science, Graphics, Visualization, and Usability Center, 1993.
11. C. L. MacKenzie, R. G. Marteniuk, C. Dugas, and B. Eickmeier. "Three-dimensional movement trajectories in Fitts' task: implications for motor control," The Quarterly Journal of Experimental Psychology, 39A, pp. 629-647, 1987.
12. S. MacKenzie and W. Buxton. Extending fitts' law to two-dimensional tasks. In Proc. of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 216-226, 1992.
13. I. S. MacKenzie and W. Buxton. "The prediction of pointing and dragging times in graphical user interfaces," Interacting with Computers, 6, 213-227, 1994.
14. M. McGuffin and R. Balakrishnan. "Acquisition of expanding targets," In Proc. of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 57-64, 2002.
15. A. Murata. "Improvement of performance by method for predicting targets in pointing by mouse," ICEC Transactions Fundamentals, vol. E78-A, no. 11, pp. 1537-1541, 1995.
16. R. Plamondon and A. Alimi. "Speed/accuracy trade-offs in target-directed movements," Behavioral and Brain Sciences, 20, pp 279-349, 1997.
17. A. Sears and B. Shneiderman. "Split menus: effectively using selection frequency to organize menus," ACM Transactions on Computer-Human Interaction, Vol. 1, No. 1, pp. 27-51, 1994.
18. R. Takagi, Y. Kitamura, S. Naito and F. Kishino. "A fundamental study on error-corrective feedback movement in a positioning task," In Proc. of Asian Pacific Computer Human Interaction, pp. 160-172, 2002.
19. N. Walker, D. E. Meyer, and J. B. Smelcer. "Spatial and temporal characteristics of rapid cursor-positioning movements with electromechanical mice in human-computer interaction," Human Factors, 35-3, pp. 431-458, 1993
20. R. Woodworth. "The accuracy of voluntary movement," Psychological Review Monograph Supplement, 3, 1899.
21. A. Worden, N. Walker, K. Bharat and S. Hudson. "Making computers easier for older adults to use: area cursors and sticky icons," In Proc. of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 266-271, 1997.
22. S. Zhai, C. Morimoto and S. Ihde. "Manual and gaze input cascaded (MAGIC) pointing," In Proc. of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 246-253, 1999.
23. S. Zhai, S. Conversy, M. Beaudouin-Lafon and Y. Guiard. "Human on-line response to target expansion," In Proc. of Conference on Human Factors in Computing Systems (ACM SIGCHI), 177-184, 2003.
24. <http://www.prismo.ch/surveys/evaluation.php>