A Fitts' Law Evaluation of Hands-Free and Hands-On Input on a Laptop Computer

Mehedi Hassan[1(✉)], John Magee[2], and I. Scott MacKenzie[1]

[1] York University, Toronto, ON, Canada
mhassan@eecs.yorku.ca
[2] Clark University, Worcester, MA, USA

Abstract. We used the Fitts' law two-dimensional task in ISO 9241-9 to evaluate hands-free and hands-on point-select tasks on a laptop computer. For the hands-free method, we required a tool that can simulate the functionalities of a mouse to point and select without having to touch the device. We used a face tracking software called *Camera Mouse* in combination with dwell-time selection. This was compared with three hands-on methods, a touchpad with dwell-time selection, a touchpad with tap selection, and face tracking with tap selection. For hands-free input, throughput was 0.65 bps. The other conditions yielded higher throughputs, the highest being 2.30 bps for the touchpad with tap selection. The hands-free condition demonstrated erratic cursor control with frequent target re-entries before selection, particularly for dwell-time selection. Subjective responses were neutral or slightly favourable for hands-free input.

Keywords: Hands-free input · Face tracking · Dwell-time selection · Fitts' law · ISO 9241-9

1 Introduction

Most user interfaces (UIs) for computers and mobile devices depend on physical touch from the user. For instance, a web page on a laptop computer's screen requires a mouse or touchpad for pointing and selecting. Most UIs also require a keyboard to enter text. In this paper, we explore pointing and selecting without using a physical device. Our ultimate goal is to test the hands-free system for accessible computing.

We are particularly interested in methods that do not require specialized hardware, such as eye trackers. Our focus is on methods that use inexpensive built-in cameras, either on a laptop's display or in a smartphone or tablet. Tracking a body position, perhaps on the head or face, is easier than tracking the movement of a user's eyes, which undergo rapid jumps known as saccades [12]. The smoother and more gradual movement of the head or face, combined with the

M. Antona and C. Stephanidis (Eds.): HCII 2019, LNCS 11573, pp. 234–249, 2019.
https://doi.org/10.1007/978-3-030-23563-5_20

ubiquity of front-facing cameras on today's laptops, tablets, and smartphones, presents a special opportunity for users with motor disabilities. Such users desire access to the same wildly popular devices as used by non-disabled users.

Magee et al. [16] did similar research with a 2D Fitts' law task, but we present a modified approach herein. We present and evaluate a hands-free approach, comparing it with hands-on approaches, and provide the results of a comparative evaluation. The hands-free method uses camera input combined with dwell-time selection. The hands-on methods use camera or touchpad input combined with tapping on the touchpad surface for selection. For camera input, we used *Camera Mouse*, described below. We evaluated the participants on a completely hands-on method with pointing and selecting with the touchpad, a partially hands-free method with pointing with *Camera Mouse* and selecting with touchpad, and finally a completely hands-free method with pointing and selecting with *Camera Mouse* only. These methods make our user study relevant for people with partial or complete motor disabilities. Although this experiment only had participants with no motor disabilities, in future we intend to do case studies with disabled participants as well.

We begin with a review of related work, then describe the use of Fitts' law and ISO 9241-9 for evaluating point-select methods. This is followed with a description of our system and the methodology for our user study. Results are then presented and discussed followed by concluding remarks and ideas for future work. Our contribution is to provide the first ISO-conforming evaluation of hands-free input on a laptop computer using a built-in webcam.

2 Related Work

Research on hands-free input methods using camera input is now reviewed. The review is organized in two parts. First, we examine research not using Fitts' law and follow with research where the experimental methodology used Fitts' law testing.

2.1 Research Not Using Fitts' Law

Roig-Maimó et al. [19] present *FaceMe*, a mobile head tracking interface for accessible computing. Participants were positioned in front of a propped-up *iPad Air*. Via the front-facing camera, a set of points in the region of the user's nose was tracked. The points were averaged, generating an overall head position which was mapped to a display coordinate. *FaceMe* is a picture-revealing puzzle game. A picture is covered with a set of tiles, hiding the picture. Tiles are turned over revealing the picture as the user moves her head and the tracked head position passes over tiles. Their user study included 12 non-disabled participants and four participants with multiple sclerosis. All non-disabled participants could fully reveal all pictures with all tile sizes. Two disabled participants had difficulty with the smallest tile size (44 pixels). *FaceMe* received a positive subjective rating overall, even on the issue of neck fatigue.

Roig-Maimó et al. [19] described a second user study using the same participants, interaction method, and device setup. Participants were asked to select icons on the *iPad Air*'s home screen. Icons of different sizes appeared in a grid pattern covering the screen. Selection involved dwelling on an icon for 1000 ms. All non-disabled participants were able to select all icons. One disabled participant had trouble selecting the smallest icons (44 pixels); another disabled participant felt tired and was not able to finish the test with the 44 pixel and 76 pixel icon sizes.

Gips et al. [7] developed the *Camera Mouse* input method that we included in our evaluation. *Camera Mouse* uses a camera to visually track a selected feature of the body. The feature could be the nose or, for example, a point between the eyebrows. During setup, the user adjusts the camera until their face is centered in the image. Upon clicking on a face feature, *Camera Mouse* begins tracking and draws a 15×15 pixel square centered at the clicked location. This location is output as the "mouse position". Camera images are processed at 30 frames per second. The tracked location moves as the user moves their head. No user evaluation was presented in this initial paper on *Camera Mouse*.

Cloud et al. [3] conducted an experiment with *Camera Mouse* that tested 11 participants, one with severe physical disabilities. The participants were tested on two applications, *EaglePaint* and *SpeechStaggered*. *EaglePaint* is a simple painting application that uses a mouse pointer. *SpeechStaggered* allows users to spell words and phrases by accessing five boxes that contain the English alphabet. Measurements for entry speed or accuracy were not reported; however, a group of participants wearing glasses showed better performance than a group not wearing glasses.

Betke et al. [1] describe further advancements with *Camera Mouse*. They compared different body features for robustness and user convenience. Twenty participants without physical disabilities were tested along with 12 participants with physical disabilities. Performance was tested on two applications, *Aliens Game*, which is an alien catching game requiring movement of the mouse pointer, and *SpellingBoard*, a typing application where entry involved selecting characters with the mouse pointer. The non-disabled participants showed better performance with a normal mouse than *Camera Mouse*. Nine of the 12 disabled participants showed eagerness in continuing to use the *Camera Mouse* system.

Magee et al. [17] present *EyeKeys*, a gaze detection interface which exploits the symmetry between the left and right eyes to determine whether the user's gaze direction is center, left, or right. They developed a game named *BlockEscape* for a quantitative evaluation. *BlockEscape* presents horizontal black bars with gaps in them. The bars move upward on the display. The user controls a white block which is moved left and right, and aligned to fall through a gap in the black wall to the wall below, and so on. If the block reaches the bottom of the display, the user wins. If the block is pushed to the top of the screen, the game ends. Three input methods were compared: *EyeKeys* (eyes), camera mouse (face tracking), and the keyboard (left/right arrow key). The win percentages were 100% (keyboard), 83% (*EyeKeys*), and 83% (*Camera Mouse*).

2.2 Research Using Fitts' Law

Magee et al. [16] did a user study using an interactive evaluation tool called *FittsTaskTwo* [13, p. 291]. *FittsTaskTwo* runs on a laptop computer and implements the two-dimensional (2D) Fitts' law test in ISO 9241-9 (described in the following section). The primary dependent variable is *throughput* in bits per second, or bps. They also used *Camera Mouse* configured with two selection methods: 1000 ms dwell-time and *ClickerAID*. *ClickerAID* generates button events by sensing an intentional muscle contraction from a piezoelectric sensor contacting the user's skin. The sensor was positioned under a headband, making contact with the user's brow muscle. A third baseline condition used a conventional laptop computer touchpad. In a user study with ten participants, throughputs were 2.10 bps (touchpad), 1.28 bps (*Camera Mouse* with dwell-time selection), and 1.43 bps (*Camera Mouse* with *ClickerAID*). For the *Camera Mouse* conditions, participants indicated a subjective preference for *ClickerAID* over dwell-time selection.

Magee et al. [16] included a follow-on case study with a patient affected by the neuromuscular disease Friedreich's Ataxia. Throughputs were quite low at 0.49 bps (*Camera Mouse* with dwell-time selection) and 0.45 bps (*Camera Mouse* with *ClickerAID*). To accommodate the patient's motor disability, the dwell time was increased to 1500 ms.

Cuaresma and MacKenzie [4] designed an experimental application named *FittsFace*, which is similar to *FittsTaskTwo*, except it runs on Android devices and uses facial sensing and tracking for input (instead of touch). A user study with 12 participants evaluated two navigation methods (positional, rotational) in combination with three selection methods (smile, blink, dwell). Positional navigation with smile selection was best in terms of throughput (0.60 bps) and movement time (4383 ms). Positional navigation with smile selection and positional navigation with blink selection had similar error rates, about 11%. Ten of the 12 participants preferred positional navigation over rotational navigation. Seven out of the 12 participants preferred dwell-time selection.

Roig-Maimó et al. [18] conducted a target selection experiment using a variation of the *FaceMe* software described above. As their motivation was to test target selection over an entire display surface by head-tracking, they used a non-standard Fitts' law task: The targets were positioned randomly during the trials. The mean throughput was 0.74 bps. They also presented design recommendations for non-ISO tasks which include keeping amplitude and target width constant within each sequence of trials and using strategies to avoid reaction time.

Hansen et al. [8] described a Fitts' law experiment using a head-mounted display. They compared three pointing methods (gaze, head, mouse) in combination with two selection methods (dwell, click). The hands-free conditions are therefore gaze or head pointing combined with dwell-time selection. Dwell time was 300 ms. In a user study with 41 participants, throughputs were 3.24 bps (mouse), 2.47 bps (head-pointing), and 2.13 bps (gaze pointing). Gaze pointing was also less accurate than head pointing and the mouse.

3 Evaluation Using Fitts' Law and ISO 9241-9

Fitts' law – first introduced in 1954 [6] – is a well-established protocol for evaluating target selection operations on computing systems [2,11]. This is particularly true since the mid-1990s with the inclusion of Fitts' law testing in the ISO 9241-9 standard for evaluating non-keyboard input devices [9,10,20]. The most common ISO evaluation procedure uses a two-dimensional task with targets of width W arranged in a circle. Selections proceed in a sequence moving across and around the circle (see Fig. 1). Each movement covers an amplitude A, the diameter of the layout circle. The movement time (MT, in seconds) is recorded for each trial and averaged over the sequence of trials.

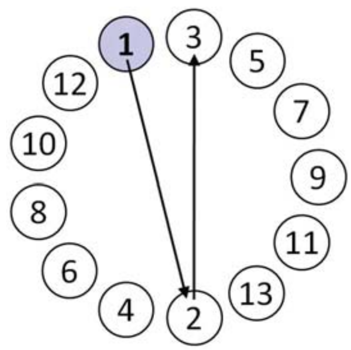

Fig. 1. Two-dimensional Fitts' law task in ISO 9241-9.

The difficulty of each trial is quantified using an index of difficulty (ID, in bits) and is calculated from A and W as

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

The main performance measure in ISO 9241-9 is throughput (TP, in bits/second or bps) which is calculated over a sequence of trials as the ID-MT ratio:

$$TP = \frac{ID_e}{MT}. \quad (2)$$

The standard specifies calculating throughput using the effective index of difficulty (ID_e). The calculation includes an adjustment for accuracy to reflect the spatial variability in responses:

$$ID_e = \log_2\left(\frac{A_e}{W_e} + 1\right) \quad (3)$$

with

$$W_e = 4.133 \times SD_x. \quad (4)$$

The term SD_x is the standard deviation in the selection coordinates computed over a sequence of trials. For the two-dimensional task, selections are projected onto the task axis, yielding a single normalized x-coordinate of selection for each trial. For $x = 0$, the selection was on a line orthogonal to the task axis that intersects the center of the target. x is negative for selections on the near side of the target center and positive for selections on the far side. The factor 4.133 adjusts the target width for a nominal error rate of 4% under the assumption that the selection coordinates are normally distributed. The effective amplitude (A_e) is the actual distance traveled along the task axis. The use of A_e instead of A is only necessary if there is an overall tendency for selections to overshoot or undershoot the target (see [14] for additional details).

Throughput is a potentially valuable measure of human performance because it embeds both the speed and accuracy of participant responses. Comparisons between studies are therefore possible, with the proviso that the studies use the same method in calculating throughput. Figure 2 is an expanded formula for throughput, illustrating the presence of speed and accuracy in the calculation.

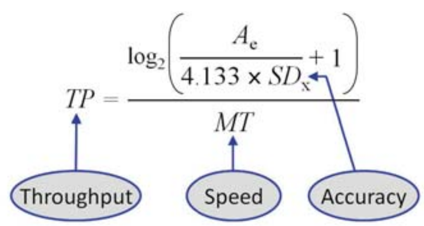

Fig. 2. The calculation of throughput includes speed and accuracy.

Our testing used *GoFitts*[1], a Java application which incorporates *FittsTask-Two* and implements the 2D Fitts' law task described above. *GoFitts* includes additional utilities such as *FittsTrace* which plots the cursor trace data captured during trials.

4 Method

The goal of our user study was to empirically evaluate and compare two pointing methods (touchpad, *Camera Mouse*) in combination with two selection methods (tap, dwell). The hands-free method combines *Camera Mouse* with dwell-time selection. A 2D Fitts' law task was used with three movement amplitudes combined with three target widths.

We recruited 12 participants. Nine were male aged 23–33 and three were female aged 23–29. All participants were from the local university community. None had prior experience using *Camera Mouse*.

[1] http://www.yorku.ca/mack/GoFitts.

4.1 Apparatus

An Asus *X541U* laptop was used as hardware. Both the built-in touchpad and the webcam provided input, depending on the pointing method. The touchpad was configured with the medium speed setting ("5") and with single-tap selection enabled.

The laptop's webcam provided images to *Camera Mouse*, as described under Related Work. Both horizontal and vertical sensitivity were set to medium. Although *Camera Mouse* can generate click events upon hovering the mouse cursor for a certain dwell time, this feature was not used since *GoFitts* provides dwell-time selection (see below).

Camera Mouse was setup to track the participant's nose. An example of the initialization screen is shown in Fig. 3. The experiment tasks were presented using *GoFitts*, described earlier. The 2D task was used with 11 targets per sequence. Three amplitudes (100, 200, 400 pixels) were combined with three target widths (20, 40, 80 pixels) for a total of nine sequences per condition.

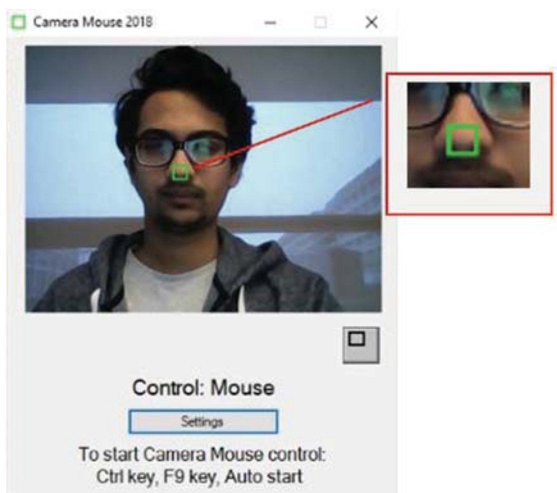

Fig. 3. *Camera Mouse* initialization screen.

Selection was performed by the *GoFitts* software (not *Camera Mouse*). For dwell-time selection, a setting of 2000 ms was used. This somewhat long value was chosen after considerable pilot testing as it provided a balance between good selection and avoiding inadvertent selections.

Selection occurred after the cursor entered and remained in the target for 2000 ms. Errors were not possible. Visual feedback on the progress of the dwell timer was provided as a rotating arc inside the target. See Fig. 4.

During dwell-time selection, if the cursor exited the target before the timeout, the timer was reset. When the cursor next entered the target, the software logged a "target re-entry" event.

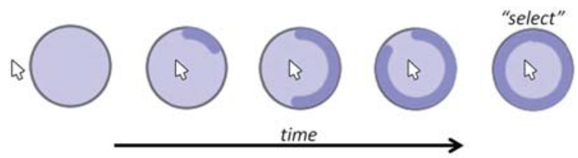

Fig. 4. Visual feedback indicating the progress of the dwell timer.

For tap selection, participants were instructed to perform a single-tap with their finger on the touchpad surface.

4.2 Procedure

Participants were welcomed into the experiment. We explained the experiment to each participant and made them aware of the purpose of it. To make participants comfortable with the setup of the experiment and *Camera Mouse*, practice trials were allowed until they felt comfortable with the interaction.

Participants were instructed to select targets as quickly and accurately as possible, but at a comfortable pace. For each sequence, they were to proceed from the first to last target without hesitation. Between sequences, they could pause at their discretion. Figure 5 shows a participant doing the experiment task (a) using the touchpad with tap selection and (b) using *Camera Mouse* with dwell-time selection. At the end of the experiment, participants provided feedback on a set of questions. They were asked about their preferred combination of pointing method and selection method. They also provided feedback on two 5-point Likert scale questions for physical fatigue and the overall rating of the hands-free phase.

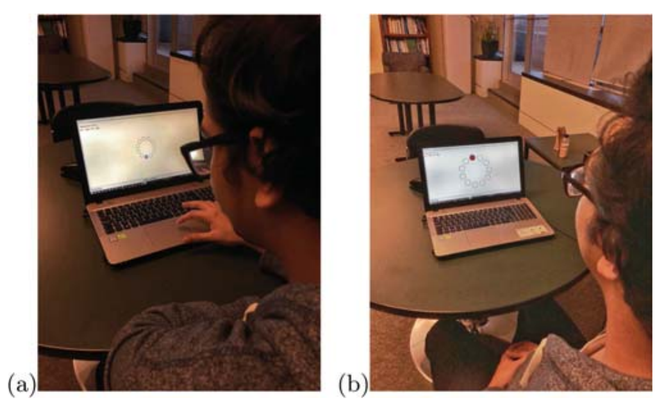

Fig. 5. Participant doing the experiment task (a) touchpad + tap selection (b) *Camera Mouse* + dwell-time selection.

4.3 Design

The experiment was a $2 \times 2 \times 3 \times 3$ within-subjects design. The independent variables and levels were as follows:

- Pointing method (touchpad, *Camera Mouse*)
- Selection method (tap, dwell)
- Amplitude (100, 200, 400 pixels)
- Width (20, 40, 80 pixels)

The primary independent variables were pointing method and selection method. Amplitude and width were included to ensure the conditions covered a range of task difficulties. The result is nine sequences for each test condition with *IDs* ranging from $\log_2(\frac{100}{80} + 1) = 1.17$ bits to $\log_2(\frac{400}{20} + 1) = 4.39$ bits. For each sequence, 11 targets appeared.

The dependent variables were throughput (bps), movement time (ms), error rate (%), and target re-entries (TRE, count/trial). There were two groups for counterbalancing, one starting with the touchpad and the other starting with *Camera Mouse*.

The total number of trials was 4752 ($= 2 \times 2 \times 3 \times 3 \times 11 \times 12$).

5 Results and Discussion

Results are presented below organized by dependent variables. For all dependent variables, the group effect was not statistically significant ($p > .05$). This indicates that counterbalancing was effective in offsetting learning effects.

Cursor trace examples, Fitts' law regression models, and a distribution analysis of the selection coordinates are also presented. Statistical analyses were done using the *GoStats* application.[2]

5.1 Throughput

Pointing with the touchpad and *Camera Mouse* had mean throughputs of 1.70 bps and 0.75 bps, respectively. The effect of pointing method on throughput was statistically significant ($F_{1,10} = 117.8, p < .0001$). Clearly, doing the experiment task with *Camera Mouse* was more difficult than with the touchpad. Of course, there is no expectation that hands-free point-select interaction would compete with hands-on point-select interaction.

During pointing with the touchpad, selecting with tap and dwell had mean throughputs of 2.30 bps and 1.10 bps, respectively. While pointing with *Camera Mouse*, selecting with tap and dwell had mean throughputs of 0.85 bps and 0.65 bps, respectively. See Fig. 6. The effect of selection method on throughput was statistically significant ($F_{1,10} = 93.0, p < .0001$). The lowest throughput of 0.65 bps was for the *Camera Mouse* with dwell-time selection – hands-free

[2] http://www.yorku.ca/mack/GoStats.

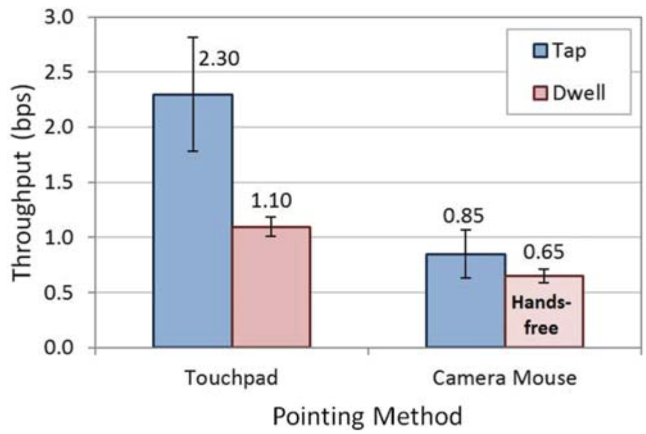

Fig. 6. Throughput (bps) by selection method and pointing method. Error bars show ±1 *SD*.

interaction. This value is low, but is expected given the pointing and selection methods employed. Throughput values in the literature are generally about 4–5 bps for the mouse [20, Table 4]. Other devices generally fair poorer with values of about 1–3 bps for the touchpad or joystick. Throughputs <1 bps sometimes occur when testing unusual cursor control schemes or when engaging participants with motor disabilities [4,5,15,18].

The closest point of comparison is the work of Magee et al. [16] who also used *Camera Mouse* with dwell-time selection. They obtained a throughput of 1.28 bps, about 2× higher than the value reported above. The biggest contributor to the difference is probably their use of a 1000-ms dwell-time, compared to 2000 ms herein. All else being equal, an increase in dwell-time yields an increase in movement time which, in turn, decreases throughput (see Eq. 2). Other points of distinction are their use of an external webcam (our apparatus used the laptop's built-in web cam) and having dwell-time selection provided by *Camera Mouse* (vs. *GoFitts* in our study). It is not clear how these differences might impact the value of throughput, however.

5.2 Movement Time and Error Rate

Since throughput is a composite measure combining speed and accuracy, the individual results for movement time and error rate are less important. They are briefly summarized below. See Fig. 7.

The effects on movement time were statistically significant both for pointing method ($F_{1,10} = 395.1, p < .0001$) and for selection method ($F_{1,10} = 93.0, p < .0001$). Note in Fig. 7a the long movement time of 5012 ms for *Camera Mouse* with dwell-time selection. As errors were not possible with dwell-time selection, the long movement time is likely caused by participants having difficulty

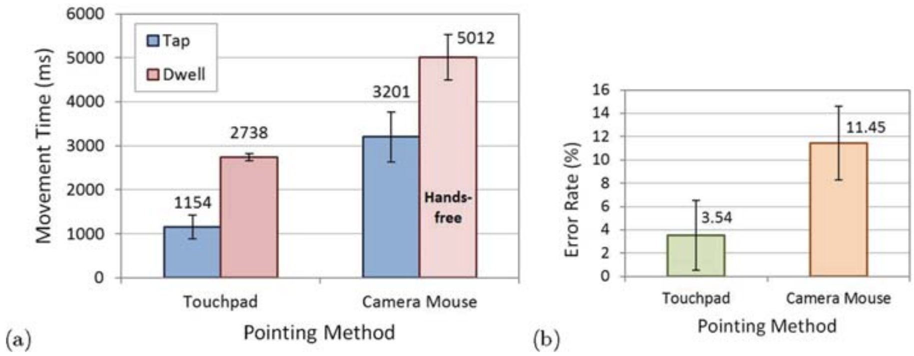

Fig. 7. Results for speed and accuracy (a) movement time by pointing method and selection method (b) error rate by pointing method with tap selection.

maintaining the cursor inside the target for the required dwell-time (2000 ms). This point is examined in further detail below in the analyses for target re-entries.

Figure 7b only shows the results by pointing method using tap selection, since errors were not possible for dwell-time selection. The effect of pointing method on error rate was statistically significant ($F_{1,10} = 67.3, p < .0001$).

5.3 Target Re-entries (TRE)

The grand mean for target re-entries (TRE) was 0.21 re-entries per trial. The implication is that for approximately one in every five trials the cursor entered the target, then left and re-entered the target. Sometimes this occurred more than once per trial.

Pointing with the touchpad and *Camera Mouse* had mean TREs of 0.09 and 0.31, respectively. So, TRE was about 3× higher for *Camera Mouse*. The effect of pointing method on TRE was statistically significant ($F_{1,10} = 10.2, p < .01$).

During pointing with the touchpad, selecting with tap and dwell had mean TREs of 0.08 and 0.11, respectively. While pointing with *Camera Mouse*, selecting with tap and dwell had mean TRE of 0.18 and 0.46, respectively. See Fig. 8. The effect of selection method on TRE was statistically significant ($F_{1,10} = 27.7, p < .0005$). Further discussion on target re-entries continues below in an examination of the trace paths for the cursor during pointing.

5.4 Cursor Trace Examples

The high value for TRE with *Camera Mouse* warrants further investigation. This was done by examining the cursor trace files generated by *GoFitts*. Cursor movements were relatively clean for the touchpad pointing method. It was a different story for *Camera Mouse*, however, where some erratic cursor movement patterns were observed. For comparison, Fig. 9 provides two examples. Both are

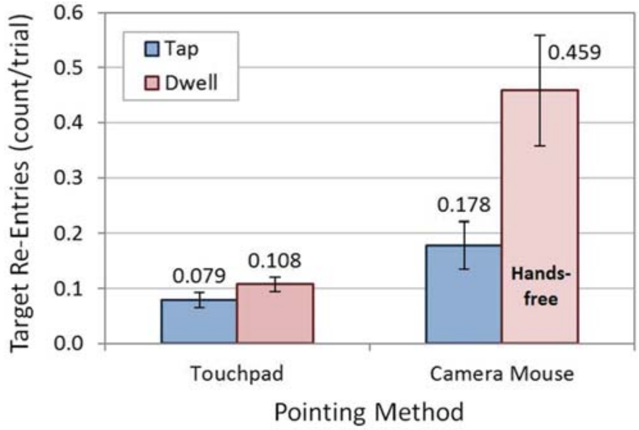

Fig. 8. Target re-entries (count/trial) by selection method and pointing method. Error bars show ±1 *SE*.

for dwell-time selection with $A = 200$ pixels and $W = 20$ pixels. Figure 9a is for pointing with the touchpad, while Fig. 9b is for pointing with *Camera Mouse*.

It is evident that the cursor movement paths were more direct for the touchpad (Fig. 9a) than for *Camera Mouse* (Fig. 9b). In fact, the difference is dramatic, as seen in the call-out in Fig. 9b. This particular trial had $MT = 14199$ ms with six target re-entries. Clearly, the participant had considerable difficulty keeping the cursor inside the target for the 2000-ms interval required for dwell-time selection.

5.5 Fitts' Law Models

To test for conformance to Fitts' law, we built least-squares prediction equations for each test condition. The general form is

$$MT = a + b \times ID \tag{5}$$

with intercept a and slope b. See Table 1. The most notable observation in the table is the very high intercepts for the dwell models. Ideally, intercepts are 0 (or $\approx$ 0) indicating zero time to complete a task of zero difficulty, which has intuitive appeal. However, large intercepts occasionally occur in the literature. A notable case is the intercept of 1030 ms in Card et al.'s Fitts' law model for the mouse [2, p. 611].

Scatter plot and regression line examples are seen in Fig. 10 for pointing using *Camera Mouse*. Although the tap model provides a good fit ($r = .9622$, Fig. 10a), the dwell-time model is a much weaker fit ($r = .8627$, Fig. 10b). Behaviour was clearly more erratic in the *Camera Mouse* + dwell condition.

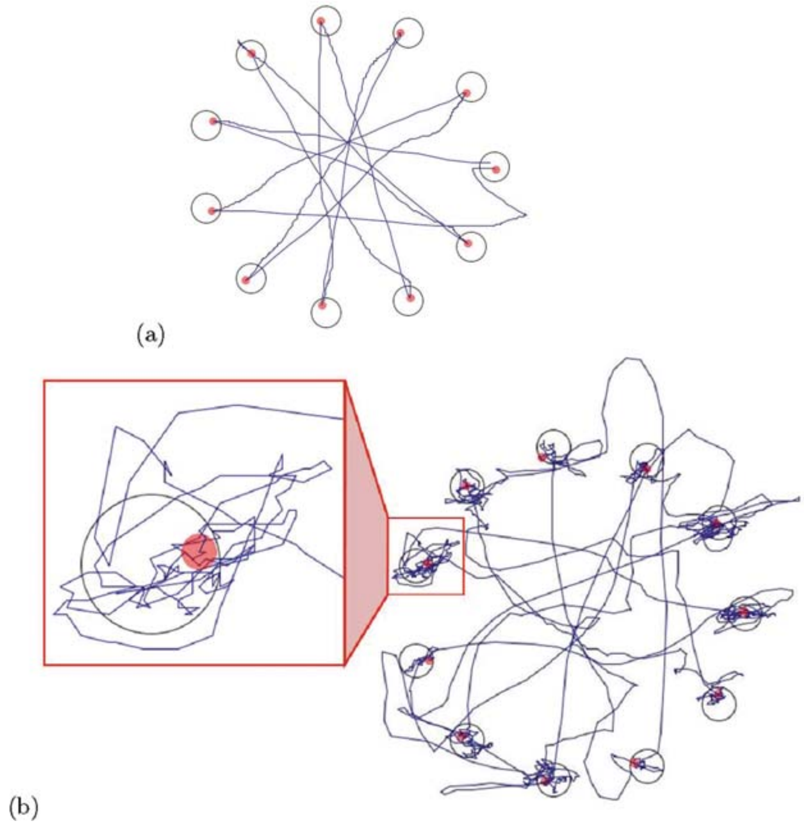

Fig. 9. Cursor trace examples for dwell-time selection with $A = 200$ pixels and $W = 20$ pixels. The pointing methods are (a) touchpad and (b) *Camera Mouse*. See text for discussion.

Table 1. Fitts' law models

Condition	Intercept, a (ms)	Slope, b (ms/bit)	Correlation (r)
Touchpad + tap	699.2	171.4	.9124
Touchpad + dwell	2270.5	176.6	.9653
Camera Mouse + tap	404.5	1055.1	.9622
Camera Mouse + dwell	1135.9	1462.7	.8627

5.6 Distribution of Selection Coordinates

The calculation of throughput uses the effective target width (W_e) which is computed from the standard deviation in the selection coordinates for a sequence of trials (see Eq. 4). There is an assumption that the selection coordinates are normally distributed. To test the assumption, we ran normality tests on the

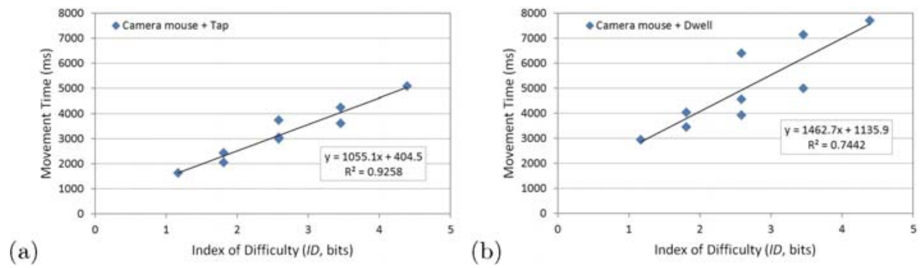

Fig. 10. Example Fitts' law models for *Camera Mouse*. Selection using (a) tap or (b) dwell.

Table 2. Lilliefors normality test on selection coordinates by trial sequence

Condition	Sequences	Normality hypothesis	
		Rejected	Not-rejected
Touchpad + tap	108	25	83
Touchpad + dwell	108	3	105
Camera Mouse + tap	108	15	93
Camera Mouse + dwell	108	7	101
Total	432	50	382

x-selection values, as transformed onto the task axis. A test was done for each sequence of trials. We used the Lilliefors test available in *GoStats*. The results are seen in Table 2.

As seen in Table 2, the user study included 432 sequences of trials (12 participants × 2 pointing methods × 2 selection methods × 3 amplitudes × 3 widths). Of these, 382, or 88.4%, had selection coordinates deemed normally distributed. Thus, the assumption of normality is generally held. The best results in Table 2 are for dwell-time selection; however, this is expected since all the selection coordinates were inside the targets. For some reason, the touchpad with tap selection had 25 of 108 sequences (23.1%) with selection coordinates considered not normally distributed.

5.7 Participant Feedback

Participants were asked to provide feedback on the experiment and indicate their preferred test condition. Eight of 12 participants chose the touchpad with tap selection as their preferred test condition. *Camera Mouse* with tap selection was preferred by two of the 12 participants. *Camera Mouse* with dwell-time selection and touchpad with dwell-time selection were preferred by one participant each.

Participants also provided responses to two 5-point Likert scale questions. One question was on the participant's level of fatigue with *Camera Mouse* (1 = *very low*, 5 = *very high*). The mean response was 2.4, closest to the *low* score. The second question was on the participant's rating of the hands-free phase of the experiment (1 = *very poor*, 5 = *very good*). The mean response was 3.4, just slightly above the *normal* score. So, interaction with *Camera Mouse* fared reasonably well, but there is clearly room for improvement.

6 Conclusion

We compared four input methods using the 2D Fitts' law task in ISO 9241-9. The methods combined two pointing methods (touchpad, *Camera Mouse*) with two selection methods (tap, dwell). Using *Camera Mouse* with dwell-time selection is a hands-free input method and yielded a throughput of 0.65 bps. The other methods yielded throughputs of 0.85 bps (*Camera Mouse* + tap), 1.10 bps (touchpad + dwell), and 2.30 bps (touchpad + tap).

Cursor movement was erratic with *Camera Mouse*, particularly with dwell-time selection. This was in part due to the long 2000 ms dwell-time employed. Participants gave the hands-free condition a neutral, or slightly better than neutral, subjective rating.

For future work, we plan to extend our testing to different platforms. Effort to port *Camera Mouse* to mobile devices is on-going. We are also planning to test with disabled participants and with different age groups.

References

1. Betke, M., Gips, J., Fleming, P.: The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities. IEEE Trans. Neural Syst. Rehabil. Eng. **10**(1), 1–10 (2002). https://doi.org/10.1109/TNSRE.2002.1021581
2. Card, S.K., English, W.K., Burr, B.J.: Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. Ergonomics **21**, 601–613 (1978). https://doi.org/10.1080/00140137808931762
3. Cloud, R.L. Betke, M., Gips, J.: Experiments with a camera-based human-computer interface system. In: Proceedings of the 7th ERCIM Workshop on User Interfaces for All, UI4ALL, pp. 103–110. European Research Consortium for Informatics and Mathematics, Valbonne, France (2002)
4. Cuaresma, J., MacKenzie, I.S.: FittsFace: exploring navigation and selection methods for facial tracking. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2017. LNCS, vol. 10278, pp. 403–416. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58703-5_30
5. Felzer, T., MacKenzie, I.S., Magee, J.: Comparison of two methods to control the mouse using a keypad. In: Miesenberger, K., Bühler, C., Penaz, P. (eds.) ICCHP 2016. LNCS, vol. 9759, pp. 511–518. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41267-2_72
6. Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. J. Exp. Psychol. **47**, 381–391 (1954). https://doi.org/10.1037/h0055392

7. Gips, J., Betke, M., Fleming, P.: The camera mouse: preliminary investigation of automated visual tracking for computer access. In: Proceedings of RESNA 2000, pp. 98–100. Rehabilitation Engineering and Assistive Technology Society of North America, Arlington (2000)
8. Hansen, J.P., Rajanna, V., MacKenzie, I.S., Bækgaard, P.: A Fitts' law study of click and dwell interaction by gaze, head and mouse with a head-mounted display. In: Proceedings of the Workshop on Communication by Gaze Interaction - Article no. 7. ACM, New York (2018). https://doi.org/10.1145/3206343.3206344
9. ISO: Ergonomic requirements for office work with visual display terminals (VDTs) - part 9: Requirements for non-keyboard input devices (ISO 9241–9). Technical report, Report Number ISO/TC 159/SC4/WG3 N147, International Organisation for Standardisation (2000)
10. ISO: Evaluation methods for the design of physical input devices - ISO/TC 9241–411: 2012(e). Technical report, Report Number ISO/TS 9241–411:2102(E), International Organisation for Standardisation (2012)
11. MacKenzie, I.S.: Fitts' law as a research and design tool in human-computer interaction. Hum.-Comput. Interact. **7**, 91–139 (1992). https://doi.org/10.1207/s15327051hci0701_3
12. MacKenzie, I.S.: An eye on input: research challenges in using the eye for computer input control. In: Proceedings of the ACM Symposium on Eye Tracking Research and Applications - ETRA 2010, pp. 11–12. ACM, New York (2010). https://doi.org/10.1145/1743666.1743668
13. MacKenzie, I.S.: Human-Computer Interaction: An Empirical Research Perspective. Morgan Kaufmann, Waltham (2013)
14. MacKenzie, I.S.: Fitts' law. In: Norman, K.L., Kirakowski, J. (eds.) Handbook of Human-Computer Interaction, pp. 349–370. Wiley, Hoboken (2018). https://doi.org/10.1002/9781118976005
15. MacKenzie, I.S., Teather, R.J.: FittsTilt: the application of Fitts' law to tilt-based interaction. In: Proceedings of the 7th Nordic Conference on Human-Computer Interaction - NordiCHI 2012, pp. 568–577. ACM, New York (2012). https://doi.org/10.1145/2399016.2399103
16. Magee, J., Felzer, T., MacKenzie, I.S.: Camera mouse + ClickerAID: dwell vs. single-muscle click actuation in mouse-replacement interfaces. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2015. LNCS, vol. 9175, pp. 74–84. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20678-3_8
17. Magee, J.J., Scott, M.R., Waber, B.N., Betke, M.: EyeKeys: a real-time vision interface based on gaze detection from a low-grade video camera. In: Computer Vision and Pattern Recognition Workshop at CVPRW 2004, pp. 159–159. IEEE, New York (2004). https://doi.org/10.1109/CVPR.2004.340
18. Roig-Maimó, M.F., MacKenzie, I.S., Manresa, C., Varona, J.: Evaluating Fitts' law performance with a non-ISO task. In: Proceedings of the 18th International Conference of the Spanish Human-Computer Interaction Association, pp. 51–58. ACM, New York (2017). https://doi.org/10.1016/j.ijhcs.2017.12.003
19. Roig-Maimó, M.F., Manresa-Yee, C., Varona, J., MacKenzie, I.S.: Evaluation of a mobile head-tracker interface for accessibility. In: Miesenberger, K., Bühler, C., Penaz, P. (eds.) ICCHP 2016. LNCS, vol. 9759, pp. 449–456. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41267-2_63
20. Soukoreff, R.W., MacKenzie, I.S.: Towards a standard for pointing device evaluation: perspectives on 27 years of Fitts' law research in HCI. Int. J. Hum.-Comput. Stud. **61**, 751–789 (2004). https://doi.org/10.1016/j.ijhcs.2004.09.001