Towards a Vision Based Cursor Control System. Assessment Using Fitts' Law

Omar Abid
York University
Toronto, Canada
omar.abid4@gmail.com

ABSTRACT
An experiment is described to test how well a vision based cursor system conforms to Fitts' law by measuring the throughput, movement time and error rate. The system is compared to a widely used input device; the mouse using a one dimensional Fitts' task experiment. A user study was tested on three input methods: mouse, camera cursor system with either a spacebar or hand gesture as a selection technique performing a total of 2,880 target selection trials. The mean throughput was significantly higher for the mouse at 3.74 bits/s than the vision system when using either the spacebar or hand gesture (1.39 and 0.79 bits/s respectively) as a selection technique ($F_{2,14} = 228.6, p < 0.001$). The experiment shows that the current implementation of the system does not conform well to Fitts' law ($R^2 = 0.63$) and future improvement is needed for a more robust cursor control system.

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: User Interfaces

Author Keywords
HCI; Fitts' Law; Mouse; Computer Vision.

INTRODUCTION
With the growing popularity in computing comes an explosion in the different ways a human can interact with machines. Traditionally, the mouse has been used to navigate a cursor on the screen and this has proven a better approach than other input devices as has been shown through numerous experiments (e.g.,[3,5,7]).

Despite advances in human computer interaction, novel human computer interaction methods for the purpose of navigating and selecting items on a screen have not been used by the main stream computer user population. The practicality and ease of use are important factors to allow users to transition to alternative input methods. We describe an experiment performed using a vision based cursor system as an alternative input device for the purpose of moving the mouse cursor and performing mouse clicks using hand tracking and gesture recognition. This alternative form of input is compared against a standard USB optical mouse by using a Fitts task experiment described later in this paper. In this section, the details of the system that is used as an input device is described followed by a review of related work as it pertains to this work. We then describe the experiment used to test the robustness of the input method. Finally the results are discussed followed by a conclusion.

Here, the CCS - a computer vision algorithm built using a pipeline consisting of object segmentation followed by tracking with an Adaptive Kalman Filter (AKF) is used as an alternative form to navigate the screen. The CCS performance is compared to the traditional mouse using a Fitts task to evaluate speed and accuracy by calculating the throughput, movement time and error rate. The task of navigation can be broken down into two subtasks; first moving to the element to be selected followed by some form of selection technique ("mouse click"). In the user study that follows the system just described will be tested with two forms of selection: 1) Using the space bar and 2) Using the hand gesture shown in Figure 2b . In addition to performing the Fitts Task, a Task Load Index (TLX) questionnaire on a Likert scale with the following three questions was presented to the users in the study: 1) Ease of Use, 2) Enjoyability and 3) Physical Fatigue.

Our aim is to test the robustness of the CCS by performing a Fitts' task comparing this system against a standard USB optical mouse as an input method. Having alternative forms of input allows increased flexibility of interaction and it is the hope of the author that such modalities can be combined for a more fulfilling human computer interaction experience for the user.

Fitts' Law
Fitts' law can successfully predict the movement time in a pointing task. It has been used in many human computer interaction experiments to measure human performance. In the original experiment, a direct pointing task was carried out and the results were manually recorded. The equations describing this phenomena are quite intuitive. Equation 1 shows the index of difficulty of the pointing task as it depends on both the distance and width of the target to be selected. A larger distance to the target to be selected (A) will result in an increase in difficulty and a reduction in the size of the target to be selected (W) will also result in an increase in the index of difficulty.

$$ID = log_2(\frac{A}{W} + 1) \tag{1}$$

Fitts' Law makes a prediction of the movement time (MT) in seconds required to complete the task given by equations 2 and 3. The constants a and b are determined empirically.

$$MT = a + bID \tag{2}$$

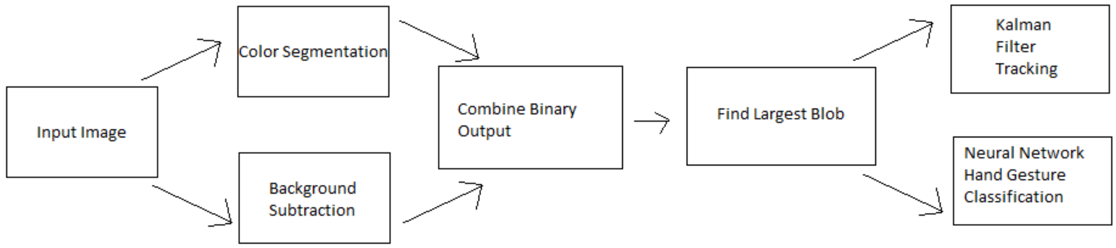


Figure 1: Pipeline for image processing.

$$MT = a + b\log_2(\frac{D}{W} + 1) \tag{3}$$

The movement time (MT) is given as the sum of the time it takes to move to the target (PT) and the time it takes for selection (ST).

$$MT = PT + ST \tag{4}$$

Taking the ratio of the index of difficulty to the movement time gives the *throughput (TP)* in bits per second as a measure for human performance by combining speed and accuracy in a single measure [7].

$$TP = \frac{ID}{MT} \tag{5}$$

Camera Cursor System

The camera cursor system (CCS) is implemented in Matlab using an image processing pipeline as shown in Figure 1. This constitutes two input methods; using either the spacebar or hand gesture as a method of selection while using the position of the hand to approximate the mouse cursor position.
Processing begins by reading an image from a webcam and performing color segmentation and background subtraction on this image in parallel. The result of these two computations are combined using a logical AND operation to obtain a binary representation of the hand location. Cluster analysis is then performed on this output to find the largest blob in order to prevent false detection of other objects of similar color to the glove used for color segmentation. The output image representation after cluster analysis is then fed through an Adaptive Kalman Filter (AKF) to approximate the location of the hand at time *t* based on where it was found at a previous time (*t-1*). The result is smooth tracking of the position of the hand given in pixels.

Mouse Movement Simulation

The control of the mouse using the CCS was done by using the output coordinates (xCoordinate, yCoordinate) given by the AKF and performing a *mouse.mousePress(xCoordinate, yCoordinate)* function.

Mouse Click Simulation

The region of the hand was extracted using the largest blob analysis (see Figure 2) , and a neural network was pre-trained on a set of images. Figure 2 shows a sample image output used for classification after running the input image through the image processing pipeline. Figure 2a and 2b show the hand gestures when the participant is pointing and selecting a target respectively.

RELATED WORK

Numerous work on using computer vision algorithms to enhance human computer interaction has been proposed but none that uses a finger based tracker as an alternative form of navigating the display as presented in this paper. Gaze input tracking [10] uses eye fixations on a specific portion of the display to maneuver followed by a manual input (space bar) as a means of selection with a mean completion time of 1.32 seconds. Another paper has attempted to use algorithms which combine gaze and frowning [8] as another form of an interaction technique with good results. The system tested against Fitts law had a throughput of 12.7 bits/s for the gaze and frowning technique and 5.2 bits/s when using a standard mouse. Tests to see how well the data fits to Fitts law also showed a high correlation coefficient of $R = 0.988$.

Flow Mouse

The FlowMouse [9] system uses optical flow techniques to model the motion of the hand followed by a capacitive sensor as a means of interaction. Using an additional sensor is disadvantageous as it requires an extra piece of hardware to be present. The system presented does not included this complexity by simply using a hand gesture as a method of selection or using a keyboard key to make a selection once the cursor is in the target area. Even so, results show that their model conforms well to Fitts law with a throughput of 2.15 bits/s and an average movement time of 1.70 s on their FlowMouse system.

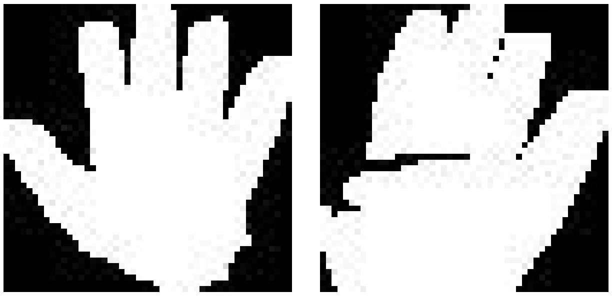
METHODS

Participants

8 right handed York University students (4 male, 4 female) with an age range of 18-28 were recruited for the user study. All participants were experienced mouse users, using the computer 5-10 hours a day.

Apparatus

The experiment was conducted on a 17' Macbook Pro Retina display laptop using the builtin webcam for image processing. The resolution of the screen was set at 1920x1080 with an aspect ratio of 16:9. A standard USB optical mouse with two buttons and a scroll wheel was used for one of the input



(a) (b)

Figure 2: Hand Gesture for: (a) Pointing Gesture (b) Selection Gesture

devices. Due to the large amount of noise present when processing the data, a blue glove was used for tracking and gesture recognition in order to make colour segmentation easier.

Two pieces of software were used for this experiment. The first is the hand tracking and gesture recognition software implemented in Matlab [1]. This program allows for mouse movement and click events as described in the introduction. The second is the FittsTaskOne [6] software which was modified to register spacebar events as mouse clicks. This software is used to carry out the experiment as shown in Figure 3.

Task

The experimental design of the task is shown in Figure 3. A standard one dimensional Fitts' law task was performed on each of the test conditions. The distance (A) and width (W) of the bars was set at either [300,500] and [30,60,90] pixels respectively. A sequence is defined as a set of 10 trials with a chosen distance and width of the bars. Hence, there were a total of 6 unique sequences for each input method tested. The user was asked to complete the task by selecting the bar highlighted as red, and after each mouse click, the other bar was filled with red indicating as the current bar to be selected. In the CCS system, the mouse movement and click were simulated as described in the introduction. When the task was first performed no error was allowed on the first bar selection, on subsequent selections errors were allowed. See Figure 3 for a visual representation of the task.

Procedure

Each participant was brought to the lab under constant lighting conditions and received instructions on completing the task. A demonstration of the task was then given with each input method being tested. The participants were then given practice trials to become comfortable doing the task with each input device. After the completion of each sequence, the participant was given the freedom to take a break before continuing. After the completion of the task with an input method, each participant was asked to complete a short questionnaire as shown in Table 1. Instructions were given to perform the task quickly yet at a comfortable pace while putting an emphasis on accuracy of target selection.

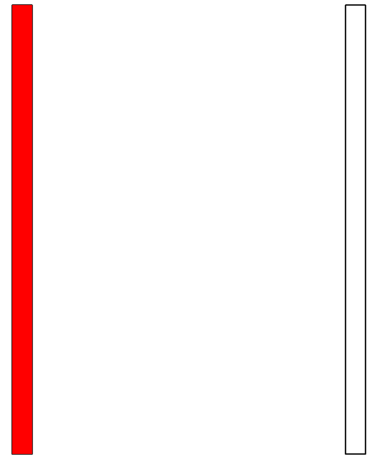


Figure 3: Experimental design for the Fitts Task used to test the different input methods. The filled red bar is the current target that the user must select. After selection, the other bar is filled with the red color. Each task with a given amplitude and width consists of 10 trials.

The use of a camera cursor system was compared against a standard mouse using FittsTaskOne [6]. This task has been used in previous experiments [9,10] and is a standard for comparing interaction techniques. Each participant performed the task on each input device two times with each block consisting of six sequences (A=[300,500] and W = [30,60,90]) with 10 trials each.

Design

A 2x3 within subject user study was carried out with the independent and dependent variables given below.

1. Independent Variable
   - Input Device
     - Mouse
     - Camera Cursor + Spacebar (*CCSpacebar*)
     - Camera Cursor + Hand Gesture (*CCGesture*)
   - Block (number of repeats of the same task condition)
2. Dependent Variables
   - Movement Time (seconds)
   - Error Rate (% errors)
   - Throughput (bits/s)
   - TLX (See Table 1)
     - Ease of use
     - Enjoyability
     - Physical Fatigue

Note that from herein the abbreviations *CCSpacebar* and *CCGesture* will be used for the camera cursor + spacebar and camera cursor + hand hesture levels respectively.

Questions	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The task was easy to complete with the given input method.	1	2	3	4	5
The task was enjoyable with the given input method.	1	2	3	4	5
The task caused a lot of physical strain with the given input method.	1	2	3	4	5

Table 1: Likert Scale Questionnaire on a 5-point scale. Each participant was asked to answer these questions after each method was completed.

Users were split into three groups each performing the task using the input methods in different orders for counterbalancing. The order in which they performed the task with the given input device was determined using a Latin square. The total number of trials over all participants and all three input devices tested is 2 blocks x 6 sequences x 10 trials/block x 3 input methods x 8 participants = 2,880 trials.

RESULTS AND DISCUSSION

Preliminary results on a two dimensional version of FittsTaskOne; FittsTaskTwo [6] was too difficult for the participants to complete due to the limitations of the system. Hence FittsTaskOne was carried out. A two-way ANOVA was performed on each of the dependent variables tested followed by a post hoc Scheffé comparison when appropriate.

Throughput

The mean throughput was 3.74 bits/s for the mouse condition, dropping down by 63.9% and 78.7% to 1.39 and 0.79 bits/s for the *CCSpacebar* and *CCGesture* system respectively. There was a statistically significant effect of the input device on the throughput ($F_{2,14} = 228.6, p < 0.001$). A Scheffé post hoc analysis determined that the difference in throughput was statistically significant between all three input methods tested. This is shown in Figure 4. Given the difficulty and the inherent noise present in the system, the lower throughput seen in the vision based cursor system is not unexpected. The *CCSpacebar* condition was easier for the participant to perform since hand tracking was an easier task than target selection using the *CCGesture* system.

Movement Time

The average movement time for the mouse was 788 ± 154 ms, while the *CCSpacebar* and *CCGesture* system took 2312 ± 667 ms and 3149 ± 857 ms respectively. There was a significant effect of the input device on the movement time ($F_{2,14} = 55.942, p < 0.001$). In addition, there was no significant effect within blocks across participants ($F_{2,14} = 0.363, ns$) indicating that learning was not a confounding factor. A Scheffé post hoc analysis shows that the difference in movement times was statistically significant between all three input methods tested. The results in Figure 5 show that the participants did not have lower movement times across blocks indicating that they had reached a maximum performance within the trials performed.

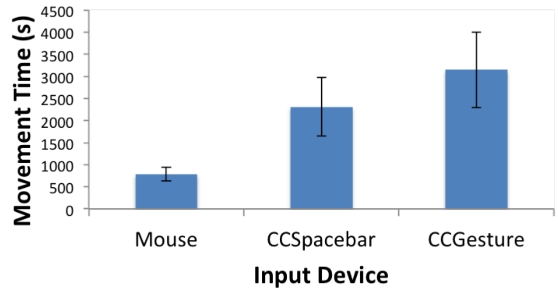


Figure 5: Average movement time (in seconds) by input method

This indicates that further practice may not have a significant effect on movement time and that the limitation most likely exists in the vision system implementation rather than poor user performance. The camera cursor system was more sensitive to movement than the mouse and had a fixed rate of movement to allow for smooth tracking. However this tradeoff between accuracy and speed is potentially disadvantageous and future work would look into an optimal control display gain [2] for this input method. As a result, participants had to perform the task at a slower rate to reach the target while avoiding overshooting by performing controlled slow movements.

Error Rate

The error rate for the mouse condition was low at 4.31%. The *CCSpacebar* condition also had a low error rate of 5.2%, while the *CCGesture* condition increased substantially to 18.3%. An ANOVA indicates that there was a significant effect of input method on accuracy ($F_{2,14} = 190.2, p < 0.001$) and a post

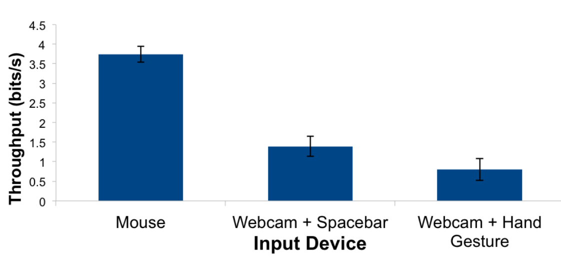


Figure 4: Average throughput (bits/s) by input method

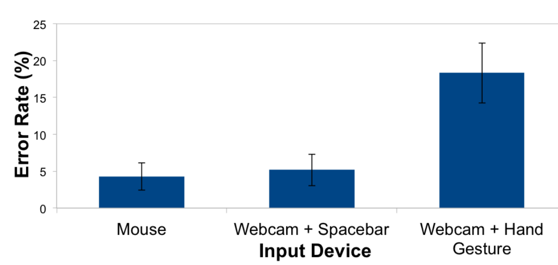


Figure 7: Average error rate (%) by the input method

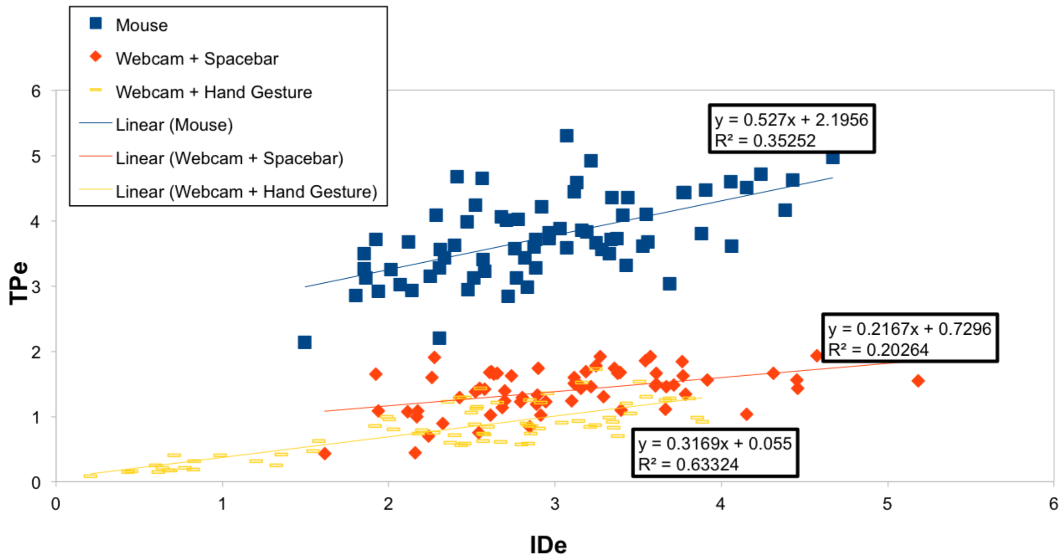


Figure 6: The effective index of difficulty ID_e as a function of the effective throughput TP_e by the input method

hoc Scheffé analysis indicates that the effect of input method on error rate was significant between all input methods tested. The error rates shown in Figure 7 are to be expected. All participants were regular computer users and performed well on the mouse condition, however the camera cursor system required fine hand movements and hence resulted in a higher error rate. The use of a gesture in the vision system presented further difficulty due to unwanted movement of the cursor position when a gesture was performed. This is evident with the substantially higher error rate in the *CCGesture* system. The hand tracking performed well overall with most errors occurring in the sequences with a small width (W).

Fitts' Law Model

Figure 6 shows a plot of the ID_e versus the TP_e for all three test conditions. The regression coefficient for the mouse model, the *CCSpacebar* and *CCGesture* were $R^2 = 0.353$, $R^2 = 0.203$ and $R^2 = 0.633$ respectively. The regression coefficients gives a measure of how much variability can be explained by the model. Here, the coefficients indicate that the experimental data collected did not conform to Fitts' Law well. Due to the data being collected from only 8 participants each performing 120 trials for each condition and using a limited set of points (*A=[300,500], W=[30,60,90]*) the results are expected. Collecting data from more participants and increasing the number of sequences as was done in Fitts' original experiment [4] would result in a better fit to the model as shown by equation 5.

Questionnaires

Subjects were asked to complete a questionnaire after the completion of the experiment with each input method. The mouse was the easiest to use with a score of 4.75 ± 0.46 while the *CCSpacebar* and *CCGesture* had 39% and 63% lower scores of 2.88 ± 0.83 and 1.75 ± 0.89 respectively. The effect of input method on the ease of use is statistically significant ($F_{2,14} = 65.681, p < 0.001$). The effect of the input method on the physical fatigue was also statistically significant ($F_{2,14} = 64.867, p < 0.001$) and a Scheffé post hoc comparison test showed that the effect was significant for the mouse and both vision system input methods but not between the vision systems. Enjoyability was not statistically significant ($F_{2,14} = 1.727, ns$). The high amount of physical fatigue and difficulty of use of the vision system discourages users from using such an input method. The results in Figure 8 show that significant improvement of the system is necessary in order for such a system to be usable.

CONCLUSION

In this paper a camera cursor system (CCS) was introduced and tested using either a spacebar or a hand gesture as a selection method while using the position of the hand to move the cursor on the screen. Results show that the throughput and movement time was statistically lower on the camera cursor + spacebar

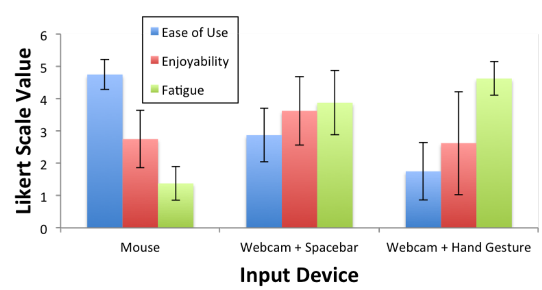


Figure 8: Likert scale questions asked after task was completed for each input condition. See Table 1 for the questions asked. 1 = Strongly Disagree, 5 = Strongly Agree

and camera cursor + hand gesture system compared to the mouse as an input device ($F_{2,14} = 228.6, p < 0.001$, $F_{2,14} = 55.942, p < 0.00$)). Moreover the input device also had a significant effect on the error rate ($F_{2,14} = 190.2, p < 0.001$). Due to the inherent noise present in the system and prior experience of using a mouse daily among the participants, the results are expected. There was also an additional factor of increased motor demand to point to the target due to the sensitivity of the camera cursor system as well as learning how to perform the correct gesture in order to register as a click event. Participants also complained about arm and shoulder pain. All of these factors need to be considered in future studies to improve the system. Current results indicate that Fitts' law does not apply well to the data and if alternative forms of human computer interaction are to be considered, better performance and a more seamless user experience is necessary. Future work would look into optimizing the control display gain [2] for improving throughput and movement time while maintaining a low error rate.

REFERENCES

1. Abid, O., *Camera Computer Vision System*. https://github.com/omarabid59/CameraCursorSystemMatlab, (accessed April 15, 2016).

2. Casiez, G., Vogel, D., Balakrishnan, R., and Cockburn, A., The impact of control-display gain on user performance in pointing tasks, *Human-Computer Interaction*, *23(3)*, 2008, 215-250.

3. English, W. K., Engelbart, D. C., and Berman, M. L., Display-selection techniques for text manipulation. *IEEE Transactions on Human factors in electronics*, *HFE-8*, 1967, 5-15.

4. Fitts, P. M., The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, *47(6)*, 1954, 381.

5. MacKenzie, I. S., Input devices and interaction techniques for advanced computing, in *Virtual environments and advanced interface design*,(W. Barfield and T. A. Furness.). Toronto,ON: Oxford, 1995 437-470.

6. MacKenzie, I. S., and Isokoski, P.,Fitts' throughput and the speed-accuracy tradeoff. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems -CHI 2009*, (New York: ACM, 2008), 1633-1636.

7. MacKenzie, I. S., *Fitts' Law Software Download*. http://www.yorku.ca/mack/FittsLawSoftware/ (accessed April 15, 2016)

8. MacKenzie, I. S., Sellen, A., and Buxton, W. A., A comparison of input devices in element pointing and dragging tasks. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems -CHI 1991*, (New York: ACM, 1991), 161-166.

9. Surakka, V., Illi, M., and Isokoski, P., Gazing and frowning as a new human–computer interaction technique. *ACM Transactions on Applied Perception*, *1(1)*, 2004, 40-56.

10. Wilson, A. D., and Cutrell, E., FlowMouse: A computer vision-based pointing and gesture input device, in *Human-Computer Interaction-INTERACT 2005*, (M. F. Costabile and F. Paternò). Rome,Italy:Springer, 2005, 565-578.

11. Zhai, S., Morimoto, C., and Ihde, S., Manual and gaze input cascaded (MAGIC) pointing, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems -CHI 1999*, (Pittsburgh: ACM, 1999), 246-253.