Improving the Camera Mouse

Emily Yu
Williams College
ey1@williams.edu

Margrit Betke
Boston University
betke@cs.bu.edu

Abstract

Many people all over the world suffer from disabilities that impair their ability to move their arms and hands. Losing the control of these appendages may cause the loss of the ability to use a mouse as a pointing device on a computer. The Camera Mouse software was designed with these people in mind. By using a USB web camera, a facial feature can be tracked to translate their facial movement to pointer movement. Developments to make the Camera Mouse easier to use are always being made. One way to make the Camera Mouse more usable is to study the movement ability of current users in order to determine which directions of movement, if any, are easier to make than others. Another was to enhance the Camera Mouse software for its users is a feature that would add blink detection to substitute mouse pointer clicks instead of the current dwell-time feature.

Introduction

The Camera Mouse was developed as a non-intrusive, relatively inexpensive mouse alternative [1]. While a working version of the Camera Mouse is currently online and free to download, improvements are always being worked on to make navigating a computer screen easier for disabled Camera Mouse users.

One way to improve the Camera Mouse is to figure out how disabled users move. The importance of finding out which directions are easier for disabled user to move in is important for applications designed to be used in conjunction with the Camera Mouse. Basic movements can be classified as diagonal or straight, left, right, up or down, and short or long. By asking disabled users to make these movements and recording their path trajectories, these trajectories can be analyzed to determine which movements were easier to make. This can be done by comparing the path taken by the user to the shortest, straight-line path between each target. By finding the mean distance and the difference in path length of the actual path from the shortest, straight-line path, the accuracy of the actual path can be determined.

Another way to expand the capabilities of the Camera Mouse, making it easier for some users, is the addition of blink detection. This way, users can voluntarily control mouse clicks by blinking. Two ways of implementing this are discussed. One was is to use difference images to find the specific image made by blinks. The other is to use template images of the eyes, both open and closed, and search through the video images for these images.

Related Work

Akram et al. discussed motion analysis of disabled Camera Mouse users. The test described in this paper is modeled roughly after the test in [2]. It is designed to test all the major directions and lengths of movement. While the test described in [2] is designed to test the user's ability to dwell in a target for 3 seconds, this test was not designed to test dwell-time abilities. Also unlike the test by Akram et al, this test was not designed to test the user's ability to guide the cursor to targets of differing sizes.

Blink detection has been discussed in [3]. The method used to locate the eyes is by using difference images. Then using a normalized correlation coefficient a template search is used to detect blinks. This successful detection of blinks was never incorporated into the Camera Mouse. In [3] the blink detection is designed to follow a sequence of steps: it first uses difference images to locate the eye, then tracks the eye and compares it to a templates of the open and closed eye to determine the extent of closure. A blink's duration is then classified by the number of frames in a row where the eye is considered closed. This is not the way that the blink detection in this paper is attempted, but many of the same ideas are used.

The Test

This 15-target motion analysis test was designed specifically to determine which directions of movement, if any, were easier for disabled users to travel in. All the targets were kept at the same size, and were activated in an order such that horizontal, vertical, and diagonal movements in differing lengths were required. Since only the movements and not the dwell-time ability were being tested, the dwell time setting on the Camera Mouse for mouse clicks was set to the fastest setting (less than .1 second dwell-time equals a mouse click). Also, to truly test only the motion, all the targets were set to be the same size. The test design can be viewed in Figure 1.

We had a total of three testing sessions with disabled Camera Mouse users. We also had two non-disabled users to compare results with. In each test run, a few added lines in the Camera Mouse code were used to record the screen coordinates of the mouse in one text document and the screen coordinates of each mouse click in a separate text document. With these two data sets, the coordinates could be plotted using Microsoft Excel to find the motion trajectory that each subject took.

We had six subjects in total, four with disabilities and two without. The subjects with disabilities spanned from 14 years old to mid-forties. Three of these subjects were students at the Campus School at Boston College who suffered from cerebral palsy. The fourth subject was a man, who has been disabled since birth and has also been an integral person in the development of the Camera Mouse.

The data trajectories in Figures 2-29 are from two testing sessions with one of our test subjects. After the first session we ran at the Campus School, the data, seen in Figures 30 and 31, could not be used for analysis. This was evident during the testing session. The subjects at the Campus School were unresponsive and did not seem to comprehend the instructions given to them to complete the test. This became one of the major setbacks in this project.

During the last two weeks of my stay at Boston University, another test subject agreed to come in and help us with our testing. He successfully ran the test twice, and from these runs we were able to analyze his movement and compare it to the movement of the non-disabled users.

Analysis

We used both quantitative and qualitative methods to analyze the motion trajectories of the disabled and non-disabled users. The quantitative methods included measuring how long it took each subject to complete the test, finding the length difference between the approximate shortest path and the actual path taken and finding the mean distance the actual path was from the shortest path. The shortest path was determined by finding the coordinates of the mouse clicks on each target and finding the straight-line paths between each of these coordinates. To find the length difference, the shortest path length was found by adding up the distance between each target coordinate, the total path length was found by adding up the distance between each screen coordinate recorded, and then these the shortest path length was subtracted from the total path length. A diagram of how the mean distance was found can be seen in Figure 32.

Using the quantitative methods of analyzing the trajectories of disabled subjects and the non-disabled subjects, it was clear that it was much easier for the non-disabled subjects to complete the test. On average, it took the disabled subjects 58.706 seconds longer to complete the test. The paths that the disabled subjects took were, on average, 21858.777 screen pixels longer than the shortest path while the paths the non-disabled subjects took were, on average, only 1352.278 screen pixels longer than the shortest path. The trajectories were also separated into their individual paths between two targets to see the specific paths taken between each target. For each of these trajectories, the mean distance from the shortest, straight-line path was calculated. This showed that not only were the non-disabled subjects able to complete the test faster, but they were also able to complete the test with greater accuracy. For all the trajectories, the average mean distance from the straight-line path for the non-disabled users was 13.019 screen pixels, while the average mean distance for the disabled users was 100.998 screen pixels.

The qualitative ways of analyzing the data were used to determine whether certain movements were easier for the disabled users to make. By looking at the path trajectories, judgments can be made based on how closely the actual path resembled the straight-line path. From these observations, we determined that certain directions of movement were easier to make than others.

The major directions of movement are right, left, up, down, or a diagonal combination of two. Additionally, these movements can be either long or short. By looking at Figures 4, 7, 15, 20, 21, and 29 it can be seen that these trajectories are more accurate than the others. These trajectories are, incidentally, all in the same direction: diagonal and either up and right, or down and left. From the tests with the disabled users, we concluded that this diagonal movement was the most natural and easiest direction of movement.

Blink Detection

Another project of the summer was implementing a blink detection feature for the Camera Mouse as a substitution for mouse clicks. In some ways this is more accurate than the current dwell-time mouse click feature. The act of blinking is much more deliberate than dwelling in a certain radius for a certain amount of time. Also, many of the current Camera Mouse users cannot steadily hold the cursor in one spot for long. The settings of 1 second, .5 second, and even .25 second are too long. The most sensitive setting is less than .1 second to detect a mouse click. The problem with this is that any split-second pause will cause a click. This setting was used during the motion analysis testing, and many false clicks were produced, making the trajectories harder to find.

There are many ways to implement blink detection. The first way is based purely on difference images. Difference images are produced when two adjacent images from the video frames are taken and the pixels are subtracted. When a person blinks their eyes while keeping the rest of the head relatively still, the two images are almost identical, except in the area with the eyes. When the images are then subtracted, the areas that are almost identical have pixel values of almost 0, which translates to black. Where the eyes are, however, there is a distinct difference, making two grayish-white ovals. An example of this can be seen in Figure 33.

Taking this difference image, projection images can be made in the x and y directions. The projection in the x direction takes each column and adds up all the pixels in the column. The projection in the y direction takes each row and does the same. In a regular difference image where the whole head is moving, a projection in the x direction might look like Figure 34. In a difference image like Figure 33, the x and y projections will look like Figures 35 and 36.

The first blink detection method took the x and y projections of each difference image and looked for the distinct pattern in Figures 35 and 36. If the pattern was found, it would be counted as a blink and the mouse would be clicked. This method of blink detection worked pretty successfully.

There were some significant problems to this method. One problem was that by moving the head vertically a specific distance, the pattern in the projection images was identical to the blink pattern, causing false detections. Another problem was that any significant movement in the background would effectively break the blink detector by causing noise in the projection images.

Another way of detecting blinks was then attempted as an effort to avoid the above problems. It was based on the method described in [3]. Using the difference images to detect deliberate blinks, the Camera Mouse prompted the user to blink in the "eye boxes" until templates of the open and closed eye were found. This can be seen in Figure 37.

Once the templates were found, the images from the video frames were processed to look for these templates. Using a search area, the Normalized Correlation Coefficient was used to compare a region of the image to the template. Unfortunately, I ran out of time and reached the end of my summer at Boston University before I could finish this second method of blink detection.

If I had had time to finish this, here is how it would have worked. After using the Normalized Correlation Coefficient (NCC) to find the best match to the template of the

closed eye, the value of the NCC would be compared to a threshold. The NCC value is one between -1 and 1, 1 being a perfect match, and -1 being an "opposite match" (i.e. a negative image). A certain threshold would have to be set to determine what values would actually qualify as a match. If the NCC value were greater than the set threshold, a counter would be incremented. If not, the counter would be reset to 0. Then the same analysis would be done with the next image. If the counter reached a certain value, say five frames in a row, a blink would be detected and the mouse would click.

Obstacles

Throughout the ten weeks I was at Boston University, I came across many obstacles. The biggest problem was that after our first unsuccessful testing session at the Campus School, we did not get another chance to use our movement analysis tests with other subjects until the last week and a half of my summer. Due to this delay, I could not proceed on my research on the movement abilities of disabled Camera Mouse users and instead switched to working on implementing a blink detector for the software.

In doing this, I ran into a whole separate set of problems. I started out using an OpenCV function called matchTemplate to find matches to the eye templates. The documentation for OpenCV, however, is not very detailed, so, even with the help of several other people in the lab, I was unable to get this function to successfully find the blinks. After struggling with this for several weeks, I decided to write my own code to find the NCC of a template in a larger image. Before I could get this working, we got the opportunity to work with another test subject, and I went back to working on my original project.

Conclusions and Future Work

The motion analysis testing showed that strictly horizontal or vertical movements are not very natural motions for the disabled test subjects. Also, longer diagonal trajectories that are either up and right or down and left seemed to be the easier and most accurate directions for the subjects.

One way this can be seen is qualitative analysis just by looking at the trajectory plots. In Figures 15 and 29, for example, which are both horizontal movements up and to the right, it is clear from the plots that the paths taken by the subject in both are very close to the shortest, straight-line path. Another way this can be seen is through quantitative analysis. In the trajectories in Figures 4 and 15, the mean distances of the subject's paths from the straight-line paths are 16.871 and 26.591, respectively. In all of the subject's paths, these two are by far the closest to the straight-line paths. Also, just by looking at the trajectories, one can see that the path is the most accurate in Figure 29. This is backed up by the quantitative analysis, too. The mean distance for the path in Figure 29 is 20.307, which is the most accurate out of all the paths for that subject. In Figures 4, 15, and 29, the paths are all diagonal and either up and right or down and left.

In the future, it would be greatly helpful to be able to run the movement analysis test with a wider range of disabled subjects. Having a bigger sample of users would help us to generalize better the way Camera Mouse users move. Time was also a crucial

setback in the blink detection. With more time, hopefully the blink detection will be fully implemented into the Camera Mouse software.

References

[1] W. Akram, L. Tiberii and M. Betke. "A customizable camera-based human computer interaction system allowing people with disabilities autonomous hands free navigation of multiple computing tasks." C. Stephanidis, M. Pieper (Eds.), Universal Access in Ambient Intelligence Environments -- 9th International ERCIM Workshop "User Interfaces For All" UI4ALL 2006, Königswinter, Germany, September 2006, Revised Papers. LNCS 4397, pages 28-42. Springer-Verlag. Technical report BU-CS-2006-006.

[2] W. Akram, L. Tiberii, and M. Betke, 2008. "Designing and Evaluating Video-based Interfaces for Users with Motion Impairments," Universal Access in the Information Society. In review.

[3] K. Grauman, M. Betke, J. Gips, G. R. Bradski, "Communication via Eye Blinks - Detection and Duration Analysis in Real Time," Proceedings of the IEEE Computer Vision and Pattern Recognition Conference CVPR 2001, Vol. 2, pp. 1010-1017, Kauai, Hawaii, December 2001.

Figures

Figure 1: The test design. The subject starts by clicking on the start button, then the next target (represented by the arrow) changes from a blue box to the image seen below. The subject then moves the cursor to the target and clicks on it. Once clicked on, the image returns to a blank, blue box, and the next target is activated.

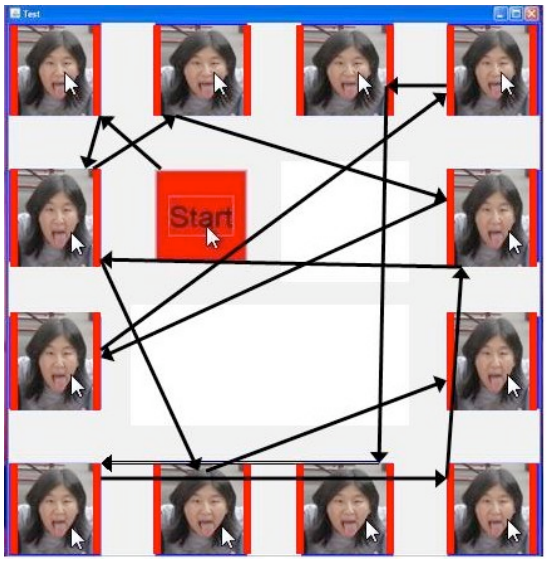


Figure 2-15: The individual trajectories of one of the disabled subjects

Figure 2:

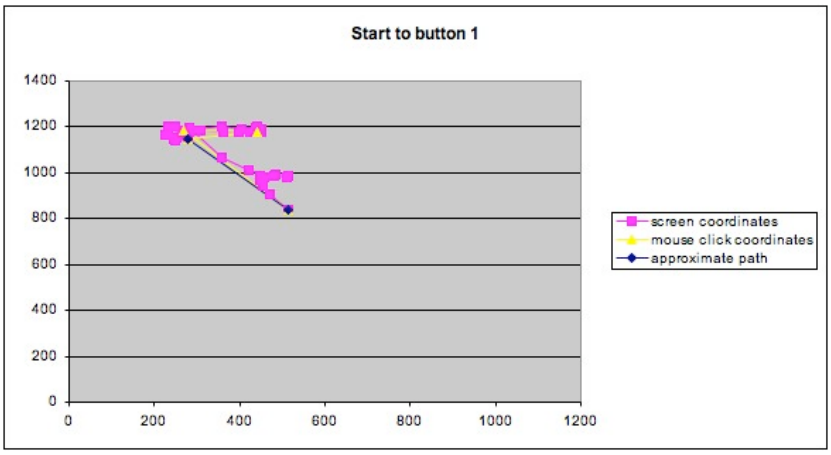


Figure 3:

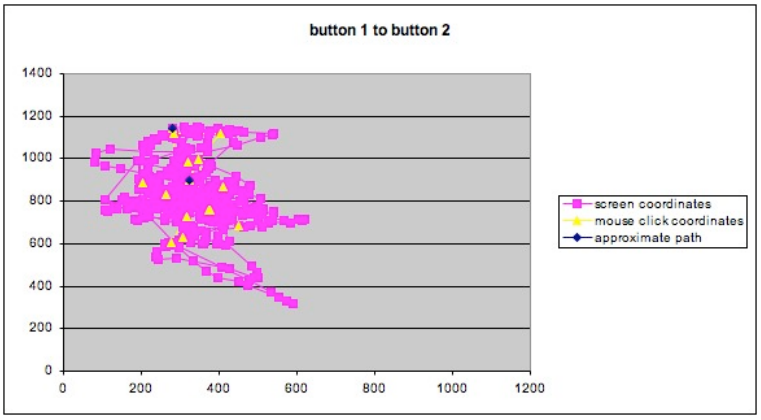


Figure 4:

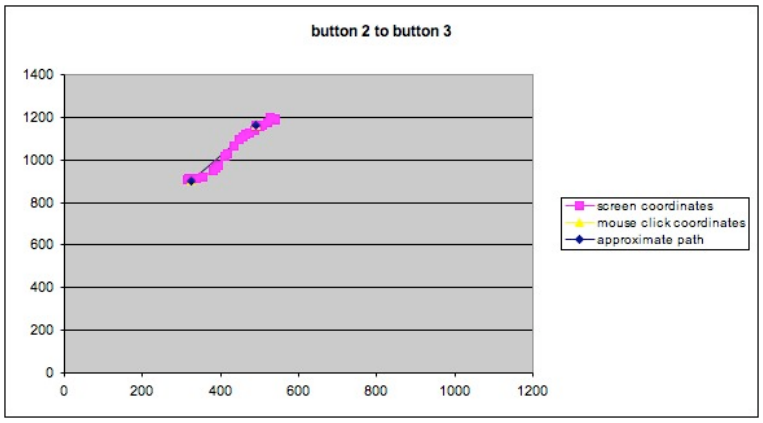


Figure 5:

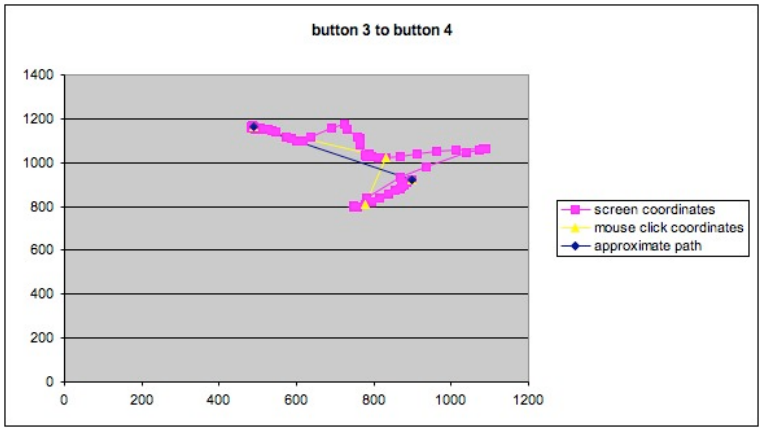


Figure 6:

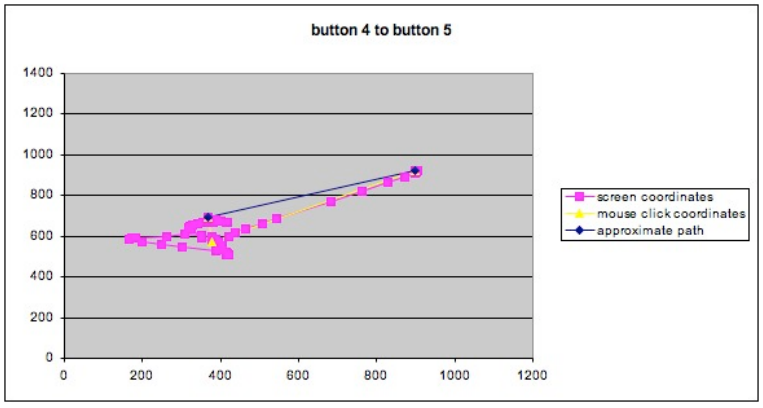


Figure 7:

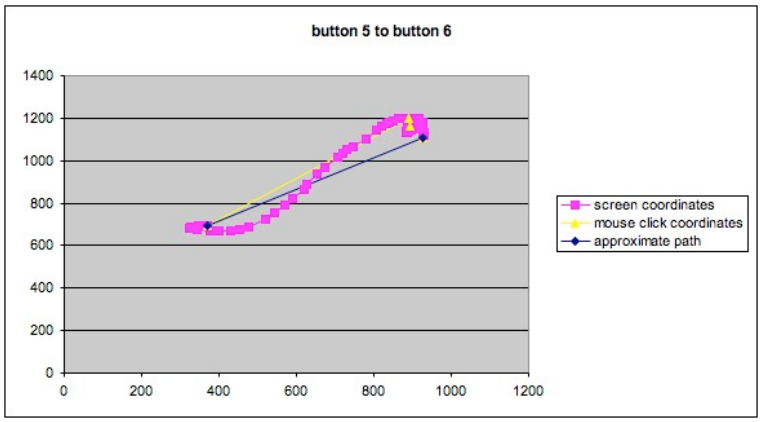


Figure 8:

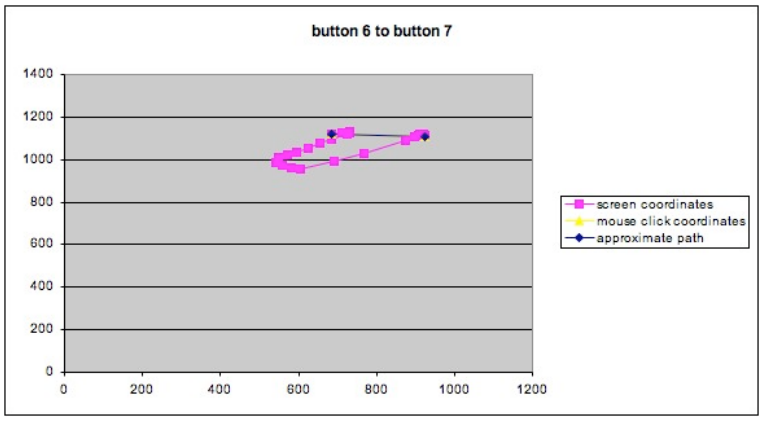


Figure 9:

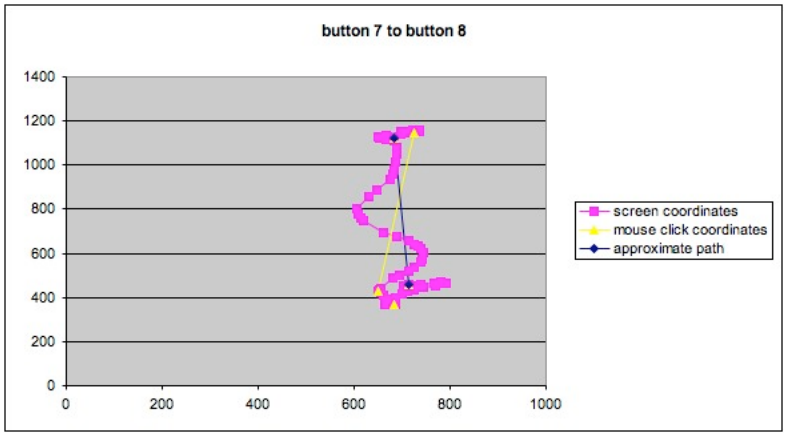


Figure 10:

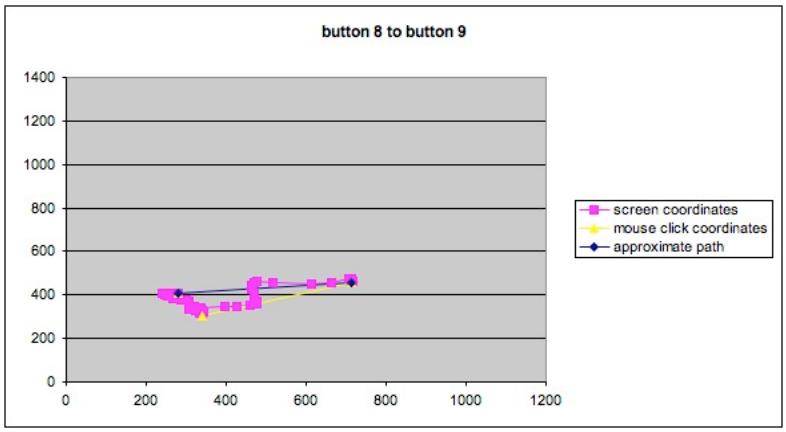


Figure 11:

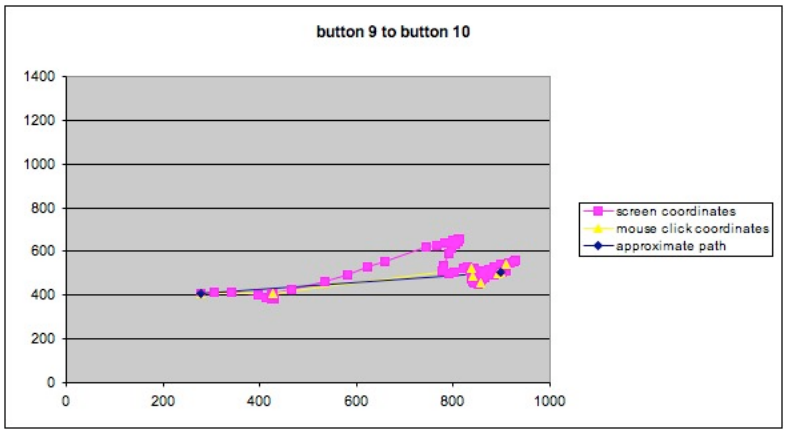


Figure 12:

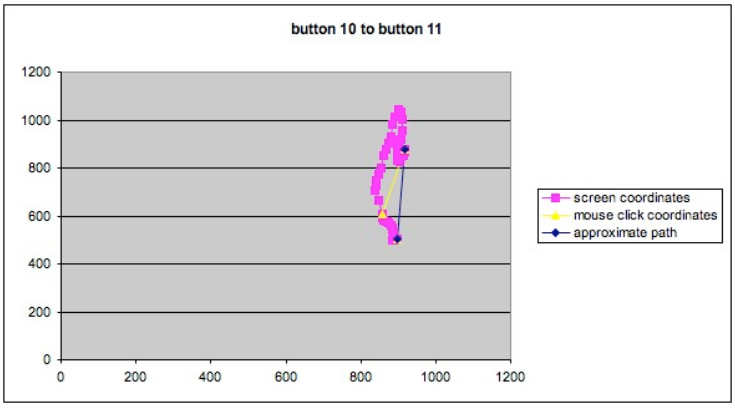


Figure 13:

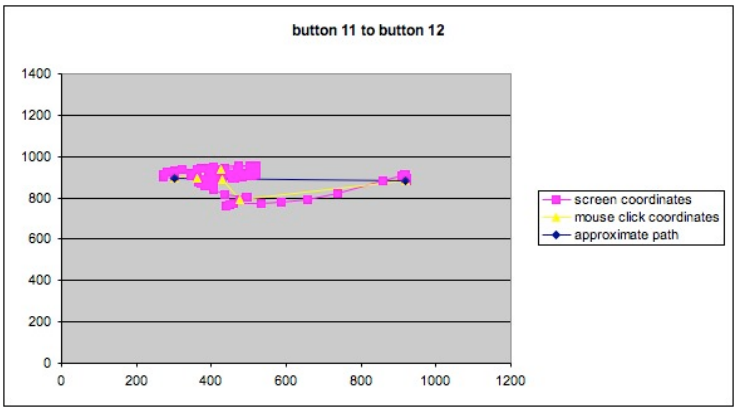


Figure 14:

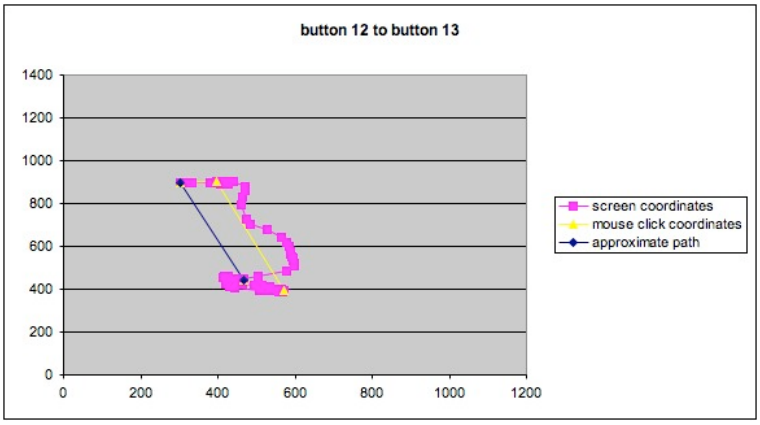
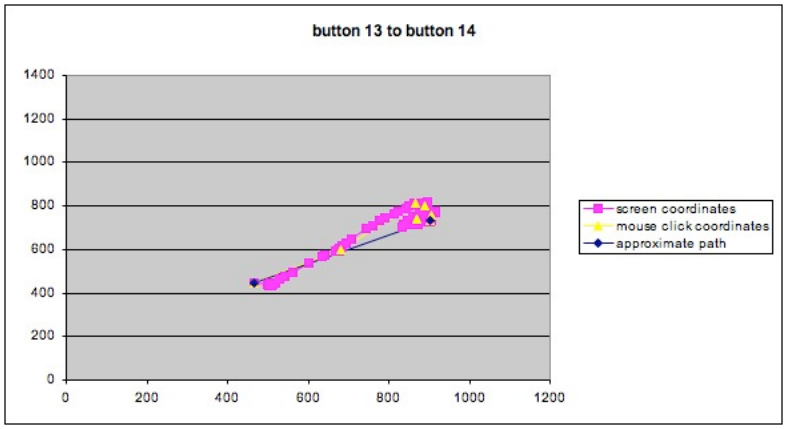


Figure 15:



Figures 16-30: The trajectories from another disabled subject's test run

Figure 16:

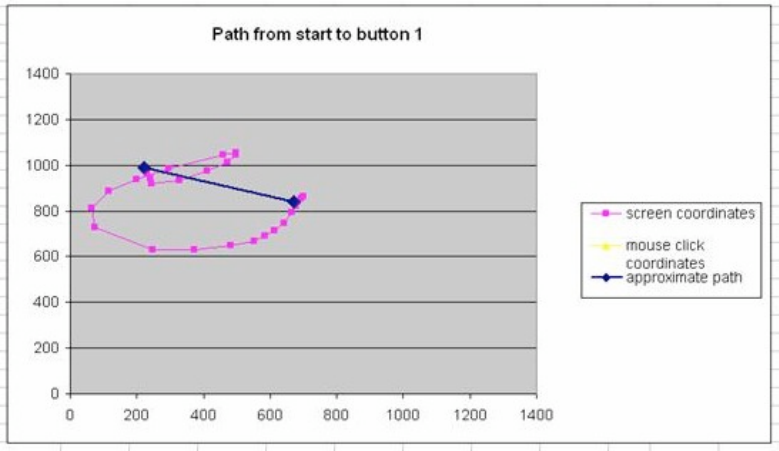


Figure 17:

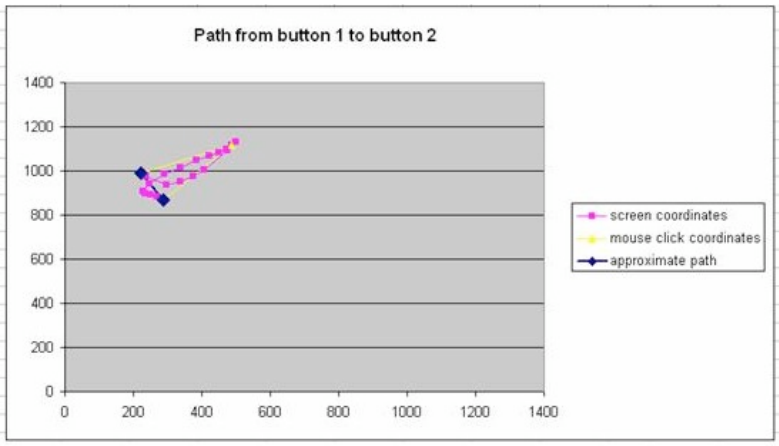


Figure 18:

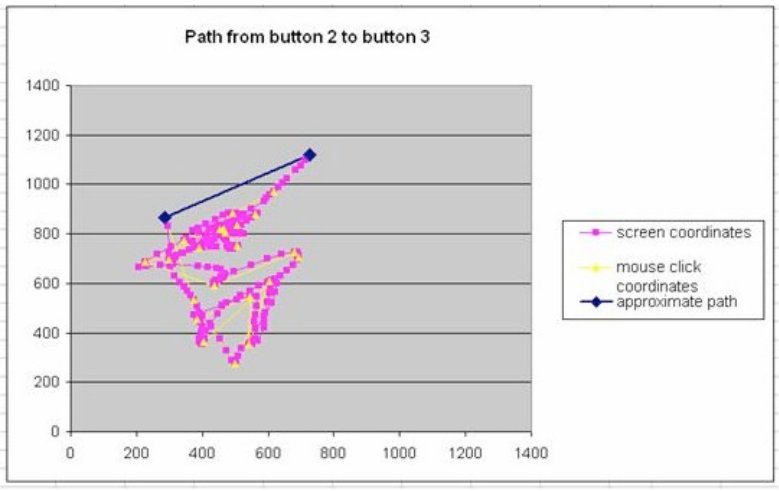


Figure 19:

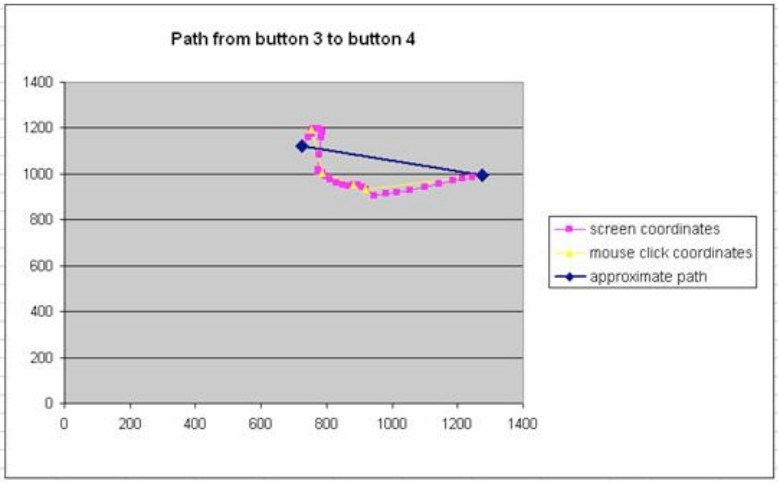


Figure 20:

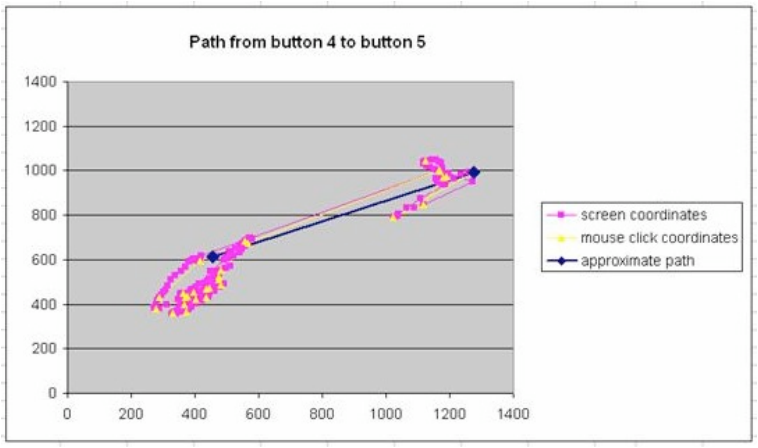


Figure 21:

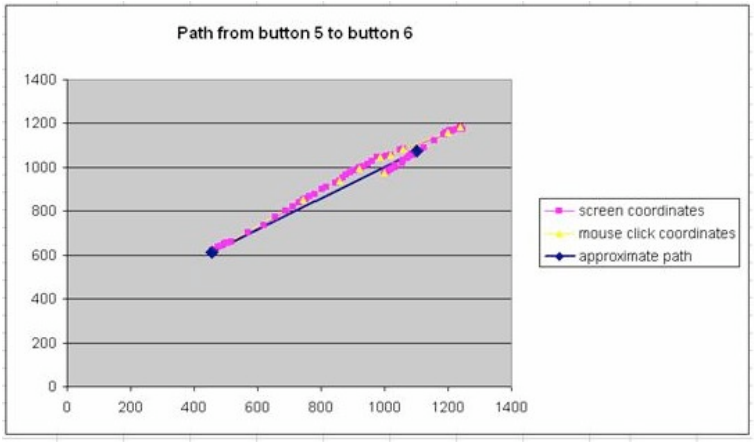


Figure 22:

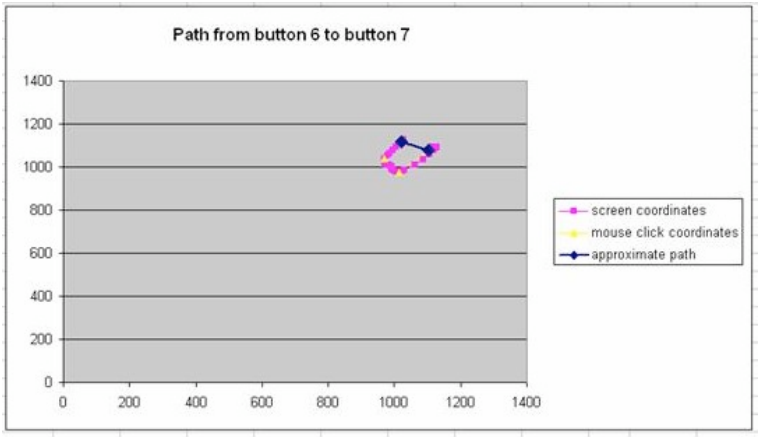


Figure 23:

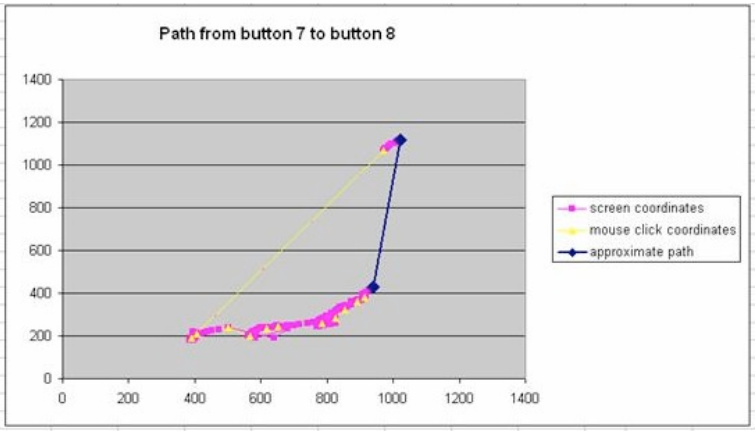


Figure 24:

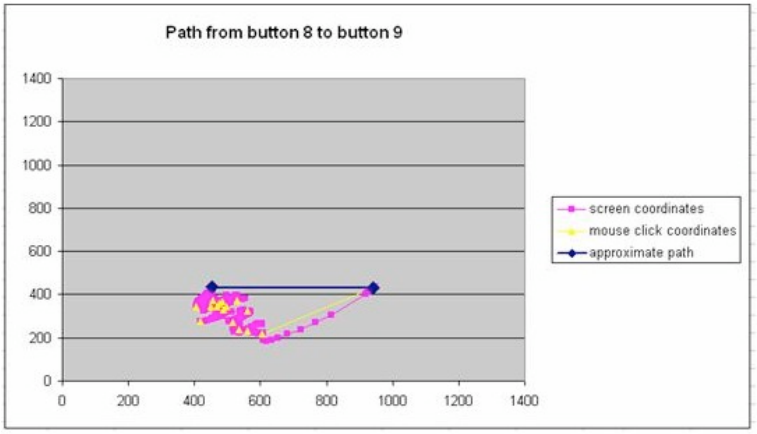


Figure 25:

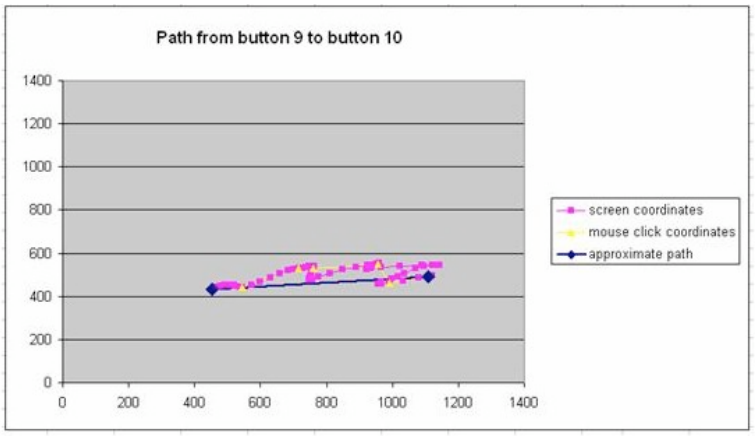


Figure 26:

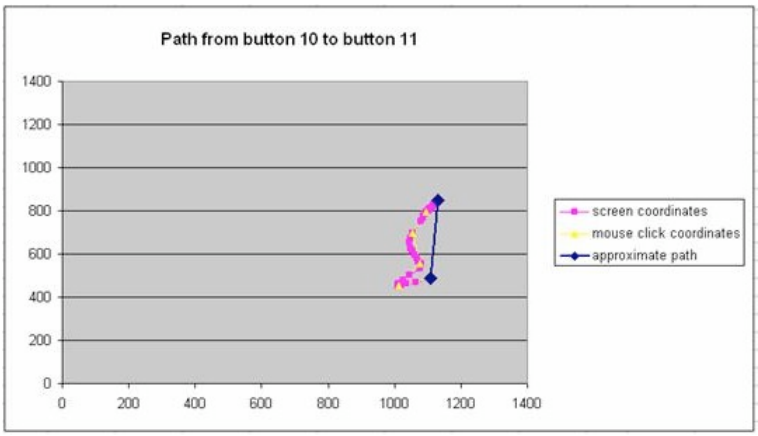


Figure 27:

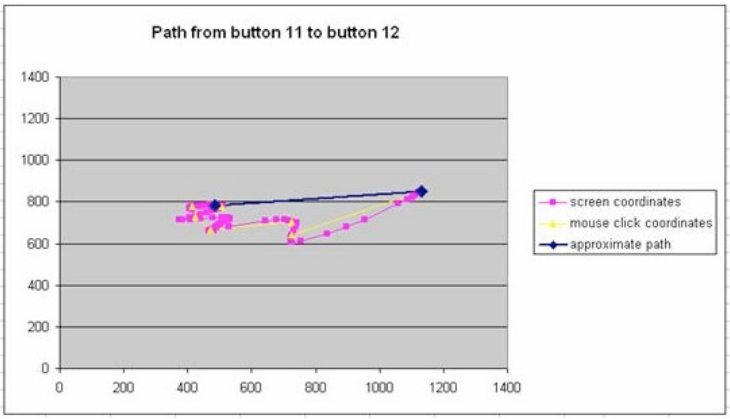


Figure 28:

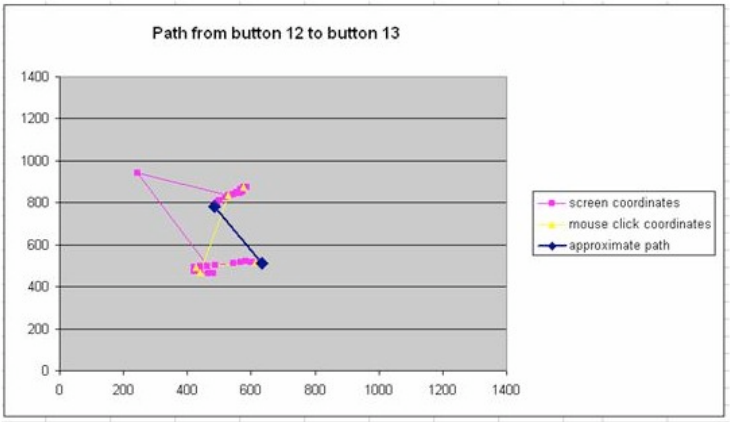


Figure 29:

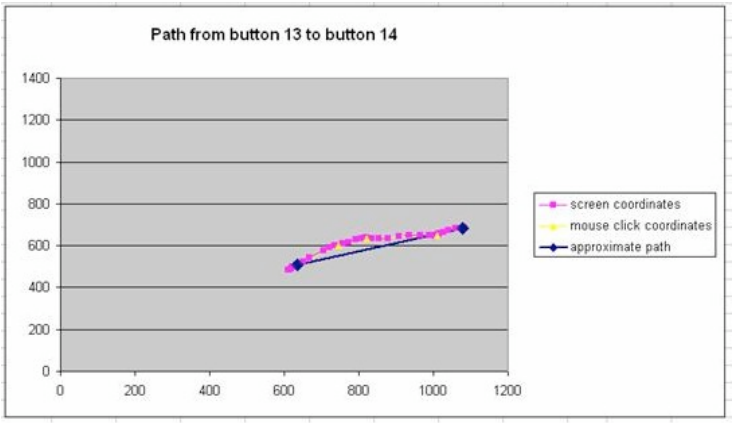


Figure 30: Subject #1 from the Campus School

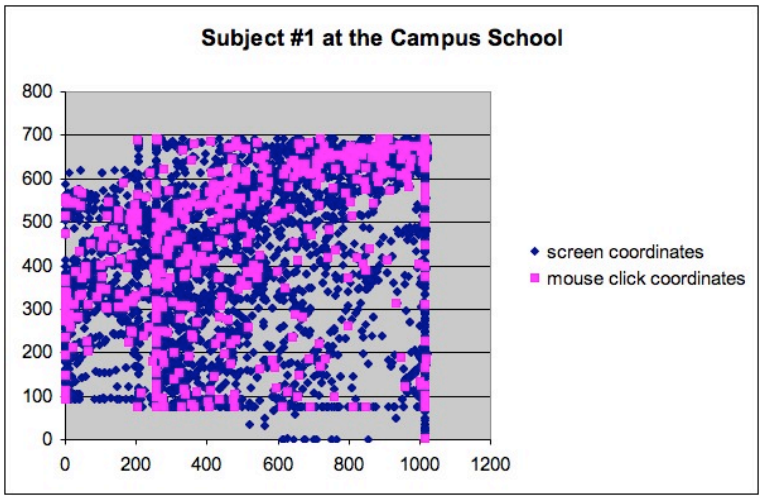


Figure 31: Subject #2 from the Campus School

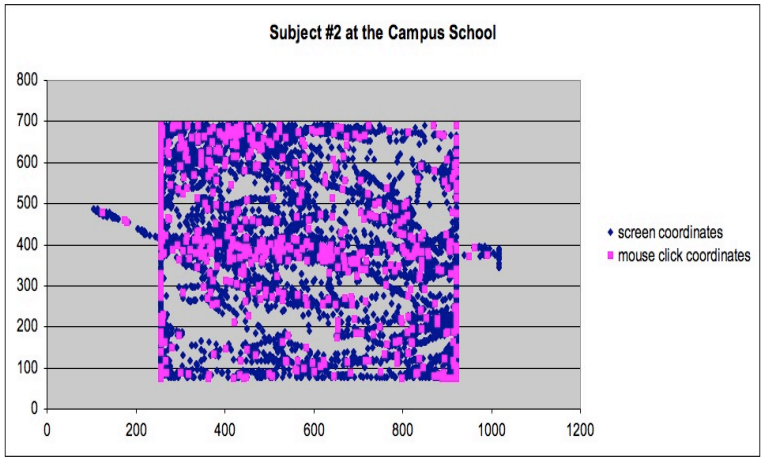


Figure 32: How the mean distance from the shortest path is calculated

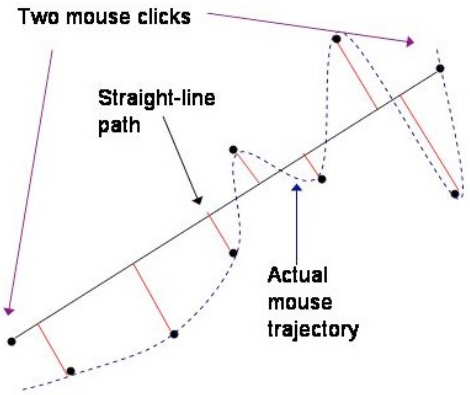


Figure 33: Difference image of a distinct blink

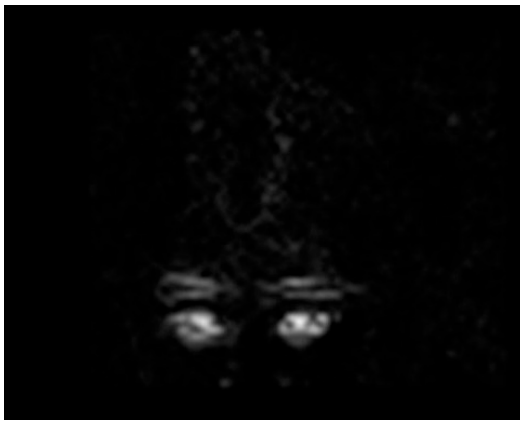


Figure 34: Projection image of entire head movement

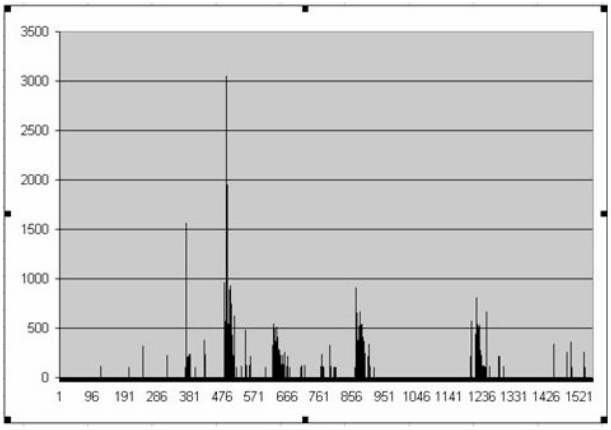


Figure 35: Horizontal projection of a blink

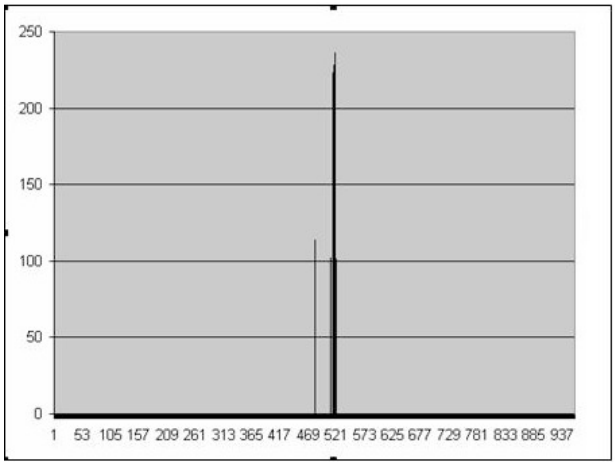


Figure 36: Vertical projection of a blink

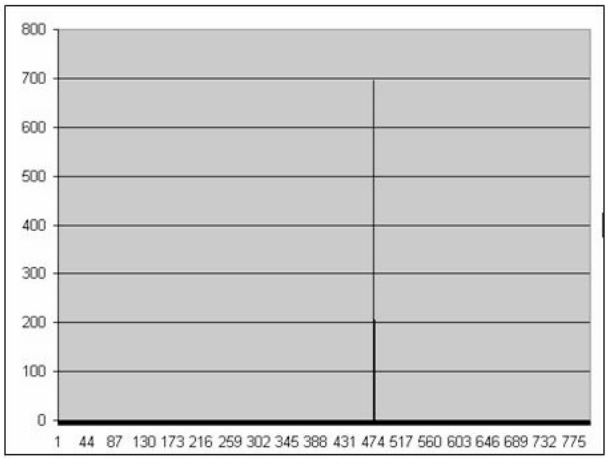


Figure 37: A sample template of eyes