Dashboard / My courses / COSC368-20S2 / Sections / Labs / Lab 4: Conducting a Fitts' Law Experiment

Started on	Thursday, 6 August 2020, 12:14 AM
State	Finished
Completed on	Thursday, 6 August 2020, 1:39 AM
Time taken	1 hour 25 mins
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Complete

Mark 1.00 out of 1.00

Aims and Fitts' Law

Fitts' law is used to assess human performance in pointing tasks. The methodology is relatively standardised, and experiments allow rapid comparison of the efficiency of different pointing devices and techniques. In this lab you will measure and analyse your own performance with a mouse pointing device. This will allow you to empirically derive your Fitts' Law *a* and *b* values for the experimental interface you created in Lab 3.

At the end of this lab you should understand the basics of the following:

- designing and conducting Fitts' law experiments; and
- analysing and interpreting data from Fitts' law experiments.

Fitts' law

Recall from lectures that Fitts' law states that human pointing performance is a linear function of the target 'index of difficulty' (*ID*):

```
MT = a + b * ID
```

where *ID* is the logarithm of the ratio of target distance (or 'amplitude' *A*) to target width (*W*).

$$ID = \log_2(A/W + 1)$$

The constants *a* and *b* are empirically derived intercept and slope constants that roughly represent targeting preparation time and the rate of information processing in cursor control. The reciprocal of the slope is also called the 'throughput' or 'bandwidth' of control, measured in bits per second (where ID is measured in bits due to its base 2 logarithm). Low throughput means that pointing times increase rapidly with *ID;* high throughput means they increases slowly with *ID.* Typically, mouse throughput is ~6-8 bits/second, and preparation time is ~200ms (although both values are sensitive to exact experimental set up).

In Lab 3 you created a Fitts' law experimental system built using the TkInter canvas widget. It should control exposure to a series of one-dimensional pointing tasks at specific amplitude and width settings, and it should record task times in a log file. (If you failed to write this program, ask the tutor, who will supply one to you.)

Generate the data

Set your experimental interface (or the one supplied to you) to measure your pointing performance with the following settings:

- Amplitude ∈ {64, 128, 256, 512}
- Width ∈ {8, 16, 32}
- 8 repetitions per combination

Generate your log file by completing the experimental selections focusing on performing as quickly and accurately as possible.

Write a Python script to extract the data

The log file contains useful source data, but it is not in a particularly convenient format for processing.

Write a *separate* python program that inputs each trial's data from the log file into a dictionary of key-value pairs, where each key is a tuple of the form (amplitude, width, selection number) and the value is the associated time.

Modify this program so that it calculates the average time across all selections for each combination of amplitude and width. The results should then be written to a file "summary.csv" in the following format:

```
A       W       ID      mean time
64      8       3.17    2.054
64      16      2.32    1.786
64      32      1.58    1.842
128     4       5.04    2.094
. . .
```

Where the first column shows the amplitude, the second width, the third is the index of difficulty (use math.log2 to calculate the logarithm value), and the fourth column is the mean time for all of the repeated trials.

Note: Times here are in *seconds* not *milliseconds*. If you're using the demo programs provided (or made your own program match the behaviour of those), you'll need to divide the times by 1000 (as the times there were written in milliseconds).

Next, modify your script so that it ignores the first two selections in each block (combination of amplitude and width). The first trial, in particular, is likely to be an outlier because the user must identify the new target location while the clock is running. Eliminating the first two trials should reduce the noise in your sample.

Finally, modify your program so that it calculates the mean time for each ID and writes the results to the summary.csv file, with ID in column 1 and mean time in column 2, as shown below:

```
ID      mean time
1.585   0.578
2.322   0.634
3.170   0.787
4.087   0.932
5.044   1.102
6.022   1.18
```

Analyse your data in Excel

Open the parsed data file in Excel, and create an *x*-*y* scatter plot with *ID* on the *x*-axis and mean selection time on the *y*-axis. Label the axes. You should find that the data-points follow a near straight line, showing that the Fitts' law prediction of a linear relationship between *ID* and selection time holds.

To inspect the strength and nature of the relationship, add a linear trendline to the data by right-clicking on any of the data-points and select "Add Trendline...". In the "Format Trendline" dialog box make sure that the "Trend/Regression Type" is set to "Linear", and select the checkboxes for "Display Equation on chart" and "Display R-square value on chart". (This can also be done in LibreOffice by double clicking on the chart or data points and picking insert trend line)

The trendline is calculated by a linear regression (which will be covered in more detail later in the course). The *R*-squared value describes the "goodness" of the relationship - the more values deviate from the line, the lower the value; if all values are perfectly on the line, then the value is 1. For Fitts' law experiments, you should expect the R^2 value to be over 0.9.

From the equation, examine your intercept and slope values (the *a* and *b* values in your Fitts' law model). Calculate your throughput or bandwidth value. Is it in the anticipated range (5-10 bits/second)?

Record your own intercept and slope parameters as you will need them for next week's lab.

Submit your XLS or ODS file here.

result.xlsx

Comment:

◄ Lab 3: Python TkInter Canvas Widget (and Fitts' Law Experimental UI)

Jump to...

fitts_src_lab5.zip ►