Mousetrap-web validation

Felix Henninger & Pascal J. Kieslich

Validation settings

- Computer: Windows 10, Intel Pentium Dual-Core 3 GHz, 4 GB RAM
- External hardware used to generate predetermined movement patterns
- Cursor position updated every 16 ms
- Browsers
  - Firefox version 72
  - Chrome version 79

Simulations

- For each browser, two simulations with 500 trials each were conducted
- Validation 1: Diagonal path
  - Start click (0,-400) followed by 176 ms pause (11*16 ms)
  - Every 16 ms cursor moves both one px up and left for 800 px, i.e., for 12800 ms in total
  - Cursor pauses at end position (-800,400) for 160 ms and then clicks (10*16 ms)
- Validation 2: Triangular path
  - Start click (0,-400) followed by 176 ms pause (11*16 ms)
  - Every 16 ms cursor moves one px up for the first 800 px
  - … and then one px left for the next 800 px, i.e., for 25600 ms in total
  - Cursor pauses at end position (-800,400) for 160 ms and then clicks (10*16 ms)

General preparation

```r
# Load libraries
library(readbulk)
library(mousetrap)
library(dplyr)
library(ggplot2)
library(tidyr)

# Set custom ggplot2 theme
theme_set(theme_classic()+
  theme(
    axis.line = element_line(colour = "black"),
    axis.ticks = element_line(colour = "black"),
    axis.text = element_text(colour = "black"),
    panel.border = element_rect(colour = "black", fill=NA),
    strip.background = element_rect(colour = NA)
  ))

options(width=95)
```

Read and preprocess raw data

```
raw_data <- read_bulk("validation_data/", subdirectories = TRUE) %>%
  filter(sender=="Decision") %>%
  mutate(
    Browser=Subdirectory,
    Condition = ifelse(
      File=="diagonal.csv",
      "Diagonal","Triangular")
  )
```

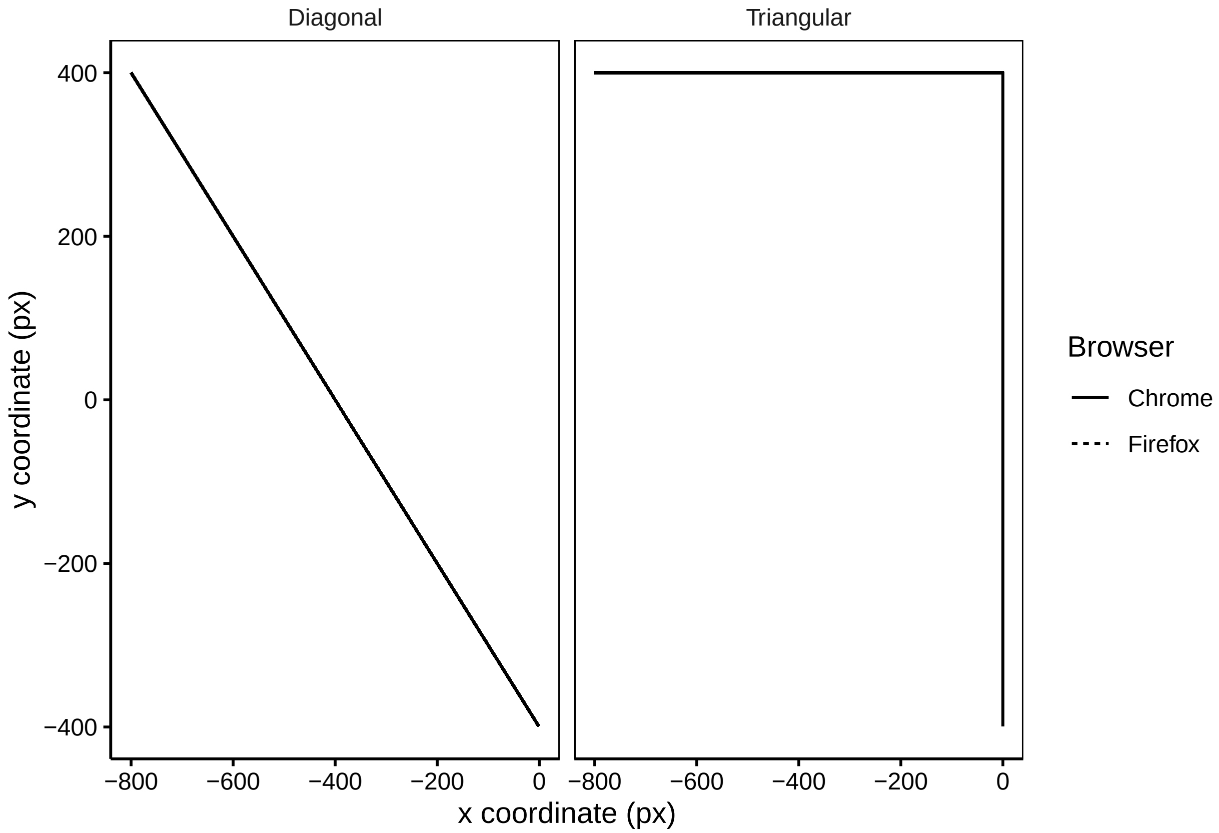
```
## Start merging subdirectory: Chrome

## Reading diagonal.csv

## Reading triangular.csv

## Start merging subdirectory: Firefox

## Reading diagonal.csv

## Reading triangular.csv
```

Visualize recorded trajectories

```
mt_data <- mt_import_mousetrap(raw_data)
```

```
## Warning in mt_import_mousetrap(raw_data): Trajectory encountered where timestamps are not
## monotonically increasing.

## Warning in mt_import_mousetrap(raw_data): Trajectory encountered where timestamps are not
## monotonically increasing.

## Warning in mt_import_mousetrap(raw_data): Trajectory encountered where timestamps are not
## monotonically increasing.

## Warning in mt_import_mousetrap(raw_data): Trajectory encountered where timestamps are not
## monotonically increasing.
```

```
mt_data$trajectories[,,"xpos"] <- mt_data$trajectories[,,"xpos"]-840
mt_data$trajectories[,,"ypos"] <- 525-mt_data$trajectories[,,"ypos"]

mt_plot(
  mt_data, facet_col = "Condition", linetype = "Browser")+
  xlab("x coordinate (px)") + ylab("y coordinate (px)")
```



```r
# # For vectorized plots, only print one trajectory (looks identical)
# # as otherwise rendering takes too much time
# mt_plot(
#   mt_data,facet_col = "Condition", linetype = "Browser",
#   subset=mt_id%in%c("id0001","id0501","id1001","id1501"))+
#   xlab("x coordinate (px)") + ylab("y coordinate (px)")

# ggsave("Figure_Val.pdf", width = 20, height=9, unit="cm")
# ggsave("Figure_Val.eps", width = 20, height=9, unit="cm")
# ggsave("Figure_Val.png", width = 20, height=9, unit="cm", dpi=600)
```

Mouse-tracking preprocessing

```r
mt_data <- mt_import_mousetrap(raw_data)
```

```
## Warning in mt_import_mousetrap(raw_data): Trajectory encountered where timestamps are not
## monotonically increasing.

## Warning in mt_import_mousetrap(raw_data): Trajectory encountered where timestamps are not
## monotonically increasing.

## Warning in mt_import_mousetrap(raw_data): Trajectory encountered where timestamps are not
## monotonically increasing.

## Warning in mt_import_mousetrap(raw_data): Trajectory encountered where timestamps are not
## monotonically increasing.
```

```
mt_data <- mt_derivatives(
  mt_data, return_delta_time = TRUE,
  dimensions = "xpos", prefix = "xpos_")

mt_data <- mt_derivatives(
  mt_data, return_delta_time = TRUE,
  dimensions = "ypos", prefix = "ypos_")

mt_data_long <- mt_export_long(
  mt_data,use2_variables = c("Browser","Condition"))
```

Sampling frequency

```
bin_counts <- mt_data_long %>%
  mutate(
    Bin = cut(xpos_delta_time,breaks=c(seq(-8,88,16),173),labels=c(0:5,">5"))
    ) %>%
  count(Browser,Condition,Bin) %>%
  group_by(Browser,Condition) %>%
  mutate(
    percent = n/sum(n),
    percent_round = round(percent,3),
    percent_round = ifelse(percent>0 & percent_round==0, "<.001",percent_round)
    ) %>%
  select(Browser,Condition,Bin,percent_round) %>%
  pivot_wider(names_from=Bin,values_from = percent_round) %>%
  replace(is.na(.), 0) %>%
  as.data.frame()

bin_counts
```

```
##   Browser  Condition     0     1     2     3     4     5    >5
## 1  Chrome   Diagonal 0.002 0.954 0.044 <.001 <.001     0     0
## 2  Chrome Triangular 0.001 0.956 0.043 <.001 <.001 <.001     0
## 3 Firefox   Diagonal 0.001  0.93 0.066 0.003 <.001 <.001 <.001
## 4 Firefox Triangular 0.001 0.931 0.065 0.003 <.001     0     0
```

```
# xtable::xtable(bin_counts)
# knitr::kable(bin_counts)
```

Distance travelled

```
x_or_y_counts <- mt_data_long %>%
  mutate(
    x_or_y = abs(ifelse(xpos_dist==0,ypos_dist,xpos_dist)),
    x_or_y = ifelse(x_or_y>5,">5",x_or_y),
    x_or_y = factor(x_or_y,levels=c("0","1","2","3","4","5",">5"))
  ) %>%
  count(Browser,Condition,x_or_y) %>%
  group_by(Browser,Condition) %>%
  mutate(
    percent=n/sum(n),
    percent_round=round(percent,3),
    percent_round = ifelse(percent>0 & percent_round==0, "<.001",percent_round)
    ) %>%
  select(Browser,Condition,x_or_y,percent_round) %>%
  pivot_wider(names_from=x_or_y,values_from = percent_round) %>%
  replace(is.na(.), 0) %>%
  as.data.frame()

x_or_y_counts
```

```
##   Browser  Condition     0     1     2     3     4     5    >5
## 1  Chrome   Diagonal 0.001 0.955 0.043 <.001 <.001     0     0
## 2  Chrome Triangular 0.001 0.956 0.043 <.001 <.001     0     0
## 3 Firefox   Diagonal 0.001 0.952 0.047 <.001 <.001 <.001 <.001
## 4 Firefox Triangular 0.001 0.953 0.046 <.001 <.001     0     0
```

```
# xtable::xtable(x_or_y_counts)
# knitr::kable(x_or_y_counts)
```

Mouse-tracking indices

```
mt_data <- mt_measures(mt_data)
results <- merge(mt_data$data,mt_data$measures,by="mt_id")

mean_measures <-
  results %>%
  group_by(Browser,Condition) %>%
  select(MAD,AUC,AD)%>%
  summarize_all(.funs=mean) %>%
  ungroup() %>%
  pivot_wider(names_from=Condition,values_from = c(MAD,AUC,AD)) %>%
  add_row(
    .before = 1, Browser="Expected",
    MAD_Diagonal=0,
    AUC_Diagonal=0,
    AD_Diagonal=0,
    MAD_Triangular=.5*sqrt(799^2+800^2),
    AUC_Triangular=.5*800*799,
    AD_Triangular=mean(c(seq(0,799,1),seq(798,1,-1))/sqrt(2))
    ) %>%
  select(Browser,ends_with("_Diagonal"), ends_with("_Triangular")) %>%
  as.data.frame()
```

```
## Adding missing grouping variables: `Browser`, `Condition`
```

```
mean_measures[,-1] <- round(mean_measures[,-1],2)
mean_measures
```

```
##    Browser MAD_Diagonal AUC_Diagonal AD_Diagonal MAD_Triangular AUC_Triangular AD_Triangular
## 1 Expected            0            0           0         565.33         319600        282.49
## 2   Chrome            0            0           0         565.30         319600        282.33
## 3  Firefox            0            0           0         565.27         319564        282.46
```

```r
# xtable::xtable(mean_measures)
# knitr::kable(mean_measures)

sd_measures <-
  results %>%
  group_by(Browser,Condition) %>%
  select(MAD,AUC,AD)%>%
  summarize_all(.funs=function(x) round(sd(x),3)) %>%
  ungroup() %>%
  pivot_wider(names_from=Condition,values_from = c(MAD,AUC,AD)) %>%
  select(Browser,ends_with("_Diagonal"), ends_with("_Triangular")) %>%
  as.data.frame()
```

```
## Adding missing grouping variables: `Browser`, `Condition`
```

```r
sd_measures
```

```
##   Browser MAD_Diagonal AUC_Diagonal AD_Diagonal MAD_Triangular AUC_Triangular AD_Triangular
## 1  Chrome            0            0           0          0.145          0.110         0.102
## 2 Firefox            0            0           0          0.177        114.584         0.230
```

Correlation between observed and expected positions

Prepare recorded data

```r
mt_data <- mt_import_mousetrap(
  raw_data,
  unordered="remove",
  digits=1)
```

```
## Warning in mt_import_mousetrap(raw_data, unordered = "remove", digits = 1): Trajectory
## encountered where timestamps are not monotonically increasing. The corresponding timestamps
## were removed.

## Warning in mt_import_mousetrap(raw_data, unordered = "remove", digits = 1): Trajectory
## encountered where timestamps are not monotonically increasing. The corresponding timestamps
## were removed.

## Warning in mt_import_mousetrap(raw_data, unordered = "remove", digits = 1): Trajectory
## encountered where timestamps are not monotonically increasing. The corresponding timestamps
## were removed.

## Warning in mt_import_mousetrap(raw_data, unordered = "remove", digits = 1): Trajectory
## encountered where timestamps are not monotonically increasing. The corresponding timestamps
## were removed.
```

```r
mt_data$trajectories[,,"xpos"] <- mt_data$trajectories[,,"xpos"]-840
mt_data$trajectories[,,"ypos"] <- 525-mt_data$trajectories[,,"ypos"]
```

Diagonal

```r
# Read in raw data from hardware that generated mouse movements
mouse_coordinates <- read.csv(
  "mouse_diagonal.csv",sep=",",
  col.names = c("xpos","ypos","click"))

# Create data frame with expected position for each timestamp
expected <- mouse_coordinates[rep(seq(which(mouse_coordinates$click==1)[1],
                                      which(mouse_coordinates$click==1)[2]),
                                  each=16),]
expected$ypos <- (-expected$ypos)

# Set constant for delay between start click and tracking onset
delta_tracking_onset <- 174+16

# Subset recorded data
current_mt_data <- mt_subset(mt_data,  Condition == "Diagonal")

# Determine expected position
# (taking delay between start click and tracking onset into account)
current_mt_data <- mt_add_variables(current_mt_data,use="trajectories",
  variables = c("xpos_expected","ypos_expected"))
for (i in rownames(current_mt_data$trajectories)){
  current_mt_data$trajectories[i,,"xpos_expected"] <-
    expected[current_mt_data$trajectories[i,,"timestamps"]+delta_tracking_onset,"xpos"]
  current_mt_data$trajectories[i,,"ypos_expected"] <-
    expected[current_mt_data$trajectories[i,,"timestamps"]+delta_tracking_onset,"ypos"]
}

# Compute correlations
mt_export_long(
  current_mt_data,
  use_variables = c("xpos","xpos_expected","ypos","ypos_expected"),
  use2_variables = "Browser"
  ) %>%
  group_by(Browser) %>%
  summarize(
    xpos_cor = cor(xpos,xpos_expected),
    ypos_cor = cor(ypos,ypos_expected)
  ) %>%
  as.data.frame() %>%
  print(digits=7)
```

```
##   Browser  xpos_cor  ypos_cor
## 1  Chrome 0.9999999 0.9999999
## 2 Firefox 0.9999969 0.9999969
```

Triangular

```
# Read in raw data from hardware that generated mouse movements
mouse_coordinates <- read.csv(
  "mouse_triangular.csv",sep=",",
  col.names = c("xpos","ypos","click"))

# Create data frame with expected position for each timestamp
expected <- mouse_coordinates[rep(seq(which(mouse_coordinates$click==1)[1],
                                      which(mouse_coordinates$click==1)[2]),
                                  each=16),]
expected$ypos <- (-expected$ypos)

# Set constant for delay between start click and tracking onset
delta_tracking_onset <- 174+16

# Subset recorded data
current_mt_data <- mt_subset(mt_data,  Condition == "Triangular")

# Determine expected position
# (taking delay between start click and tracking onset into account)
current_mt_data <- mt_add_variables(current_mt_data,use="trajectories",
  variables = c("xpos_expected","ypos_expected"))
for (i in rownames(current_mt_data$trajectories)){
  current_mt_data$trajectories[i,,"xpos_expected"] <-
    expected[current_mt_data$trajectories[i,,"timestamps"]+delta_tracking_onset,"xpos"]
  current_mt_data$trajectories[i,,"ypos_expected"] <-
    expected[current_mt_data$trajectories[i,,"timestamps"]+delta_tracking_onset,"ypos"]
}

# Compute correlations
mt_export_long(
  current_mt_data,
  use_variables = c("xpos","xpos_expected","ypos","ypos_expected"),
  use2_variables = "Browser"
  ) %>%
  group_by(Browser) %>%
  summarize(
    xpos_cor = cor(xpos,xpos_expected),
    ypos_cor = cor(ypos,ypos_expected)
  ) %>%
  as.data.frame() %>%
  print(digits=7)
```

```
##   Browser  xpos_cor  ypos_cor
## 1  Chrome 0.9999999 1.0000000
## 2 Firefox 0.9999986 0.9999987
```