Mousetrap validation

Pascal J. Kieslich & Felix Henninger

Validation settings

- OpenSesame: version 3.1.6 with legacy backend
- Mousetrap-os plugin: version 1.2.1
- Computer: Windows 7 Professional, Intel Pentium Dual-Core 3 GHz, 4 GB RAM
- External hardware (Henninger, 2017) used to generate predetermined movement patterns
- Cursor position updated at the logging resolution (10 ms)
- Two simulations with 1000 trials each

General preparation

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

# 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=90)
```

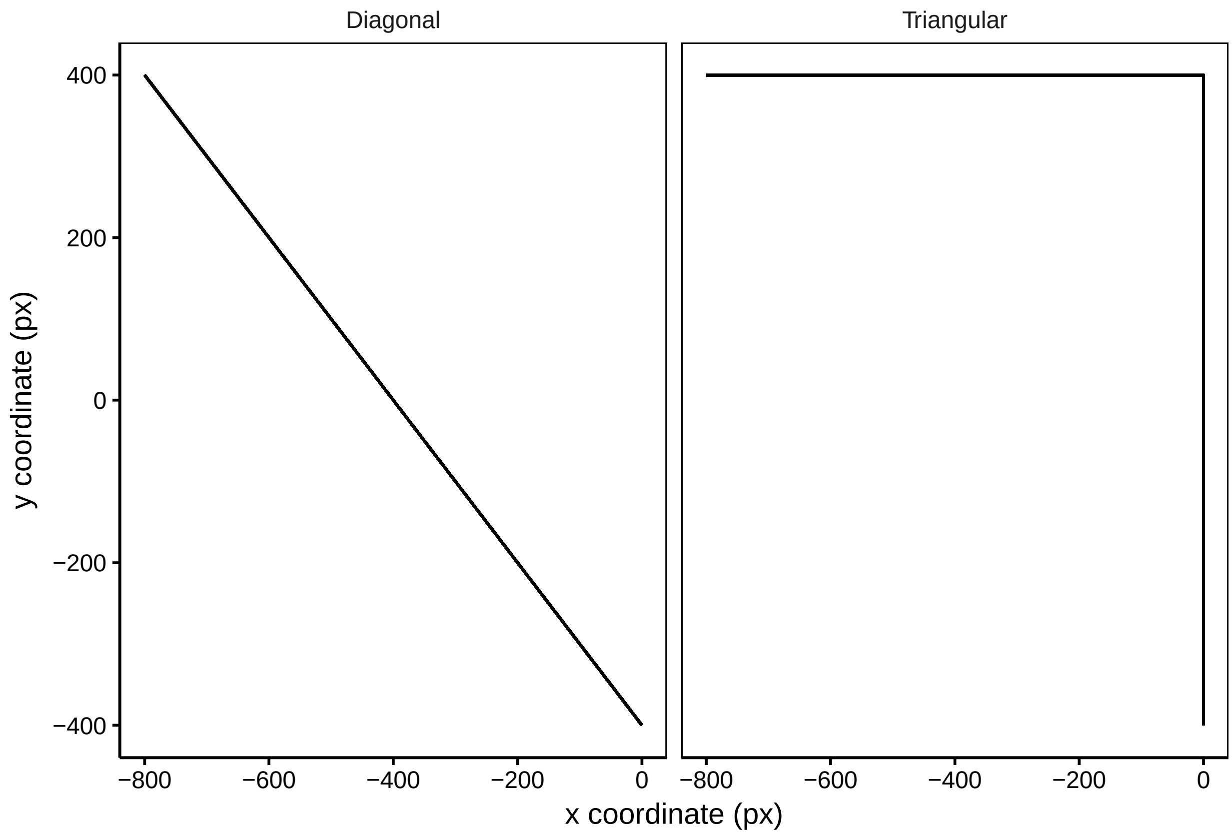
Plot trajectories

```r
raw_data <- read_opensesame("validation_data")
```

```
## Reading diagonal.csv.gz

## Reading triangular.csv.gz
```

```r
mt_data <- mt_import_mousetrap(raw_data)
mt_data <- mt_remap_symmetric(mt_data)
mt_data$data$Condition <- factor(mt_data$data$subject_nr,levels=c(1,2),
                                 labels=c("Diagonal","Triangular"))
mt_plot(mt_data,facet_col = "Condition")+
  xlab("x coordinate (px)") + ylab("y coordinate (px)")
```




```r
# ggsave("FigureA1.pdf", width = 16, height=9, unit="cm")
# ggsave("FigureA1.png", width = 16, height=9, unit="cm", dpi=600)
```

Validation 1: Diagonal path

- Start click (0,-400) followed by 110 ms pause
- Every 10 ms cursor moves both one px up and left for 800 px, i.e., for 8000 ms in total
- Cursor pauses at end position (-800,400) for 100 ms and then clicks

Read and preprocess data

```r
raw_data <- read_opensesame("validation_data",extension = "diagonal.csv.gz")
```

```
## Reading diagonal.csv.gz
```

```r
mt_data <- mt_import_mousetrap(raw_data)
mt_data <- mt_remap_symmetric(mt_data,remap_xpos = "no")
mt_data <- mt_measures(mt_data)
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_no_reset <- mt_import_mousetrap(raw_data,reset_timestamps = FALSE)
```

Temporal analyses

```r
mt_data$data <- mt_data$data %>%
  mutate(
    time_start_click =
      time_get_start_click+response_time_get_start_click,
    delta_click_stimulus =
      time_present_stimulus-time_start_click,
    delta_stimulus_tracking =
      mt_data_no_reset$trajectories[,1,"timestamps"]-time_present_stimulus,
    delta_click_tracking =
      mt_data_no_reset$trajectories[,1,"timestamps"]-time_start_click
  )

summary(select(mt_data$data,starts_with("delta"),response_time),digits = 8)
```

```
##  delta_click_stimulus delta_stimulus_tracking delta_click_tracking response_time
##  Min.   :6.000        Min.   :0.000           Min.   :7.000        Min.   :8202.000
##  1st Qu.:6.000        1st Qu.:0.000           1st Qu.:7.000        1st Qu.:8202.000
##  Median :7.000        Median :1.000           Median :8.000        Median :8203.000
##  Mean   :6.936        Mean   :0.656           Mean   :7.592        Mean   :8202.939
##  3rd Qu.:7.000        3rd Qu.:1.000           3rd Qu.:8.000        3rd Qu.:8203.000
##  Max.   :9.000        Max.   :1.000           Max.   :9.000        Max.   :8206.000
```

```r
mt_data$data %>%
  select(starts_with("delta"),response_time) %>% summarise_all(c("sd"))
```

```
##   delta_click_stimulus delta_stimulus_tracking delta_click_tracking response_time
## 1            0.6902031               0.4752787            0.6402702     0.8763829
```

```r
# mousetrap-os response_time matches mt_measures RT
table(mt_data$data$response_time==mt_data$measures$RT)
```

```
##
## TRUE
## 1000
```

Logging resolution

```r
mt_check_resolution(mt_data,desired = 10)
```

```
## $summary
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000  10.000  10.000   9.991  10.000  12.000
##
## $sd
## [1] 0.2533348
##
## $frequencies
## log_diffs
##      1      2      3      4      5      6     10     11     12
```

```
##      5    399    453     87     50      5 819852    147      1
##
## $relative_frequencies
## log_diffs
##      1      2      3      4      5      6     10     11     12
## 0.0000 0.0005 0.0006 0.0001 0.0001 0.0000 0.9986 0.0002 0.0000
##
## $frequencies_desired
## log_diffs_class
## smaller desired greater
##     999  819852     148
##
## $relative_frequencies_desired
## log_diffs_class
## smaller desired greater
##  0.0012  0.9986  0.0002
```

```r
# Percent of lags that lag by 1 ms only
147/148
```

```
## [1] 0.9932432
```

Distances between subsequently recorded positions

```r
# Frequency of distances converge across both x and y coordinates
table(as.numeric(mt_data$trajectories[,-1,"xpos_dist"]),
      as.numeric(mt_data$trajectories[,-1,"ypos_dist"]))
```

```
##
##              0      1      2
##   -2         0      0      4
##   -1         0 799992      0
##   0      21003      0      0
```

```r
# Frequency of distances
table(as.numeric(mt_data$trajectories[,-1,"xpos_dist"]))
```

```
##
##     -2     -1      0
##      4 799992  21003
```

```r
# Percent of distances
round(table(as.numeric(mt_data$trajectories[,-1,"xpos_dist"]))/
        sum(!is.na(mt_data$trajectories[,-1,"xpos_dist"])),6)
```

```
##
##       -2       -1        0
## 0.000005 0.974413 0.025582
```

Comparison of expected and observed position

```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"))
```

```r
# 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=10),]
expected$ypos <- (-expected$ypos)

# Set constant for delay between start click and tracking onset
delta_tracking_onset <- 7

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

# Correlation between observed and expected position
cor_xpos <- cor(as.vector(mt_data$trajectories[,,"xpos"]),
                as.vector(mt_data$trajectories[,,"xpos_expected"]),
                use="complete.obs")
cor_ypos <- cor(as.vector(mt_data$trajectories[,,"ypos"]),
                as.vector(mt_data$trajectories[,,"ypos_expected"]),
                use="complete.obs")
print(cor_xpos,digits = 15)
```

```
## [1] 0.999999999956697
```

```r
print(cor_ypos,digits = 15)
```

```
## [1] 0.999999999956697
```

```r
# Compute difference between expected and observed position
mt_data <- mt_add_variables(mt_data,use="trajectories",variables=list(
  xpos_diff = mt_data$trajectories[,,"xpos_expected"]-mt_data$trajectories[,,"xpos"],
  ypos_diff = mt_data$trajectories[,,"ypos_expected"]-mt_data$trajectories[,,"ypos"]
))

# Frequency of differences between observed and expected positions across both dimensions
# --> differences converge across both dimentions
table(mt_data$trajectories[,,"xpos_diff"],mt_data$trajectories[,,"ypos_diff"])
```

```
##
##            0      1
##   -1       0      4
##   0   821995      0
```

```r
# Percent of differences
round(table(mt_data$trajectories[,,"xpos_diff"])/
        sum(!is.na(mt_data$trajectories[,,"xpos_diff"])),6)
```

```
##
##       -1         0
```

```
## 0.000005 0.999995
```

Mouse-tracking indices

```
summary(select(mt_data$measures,MAD,AUC,AD),digits = 8)
```

```
##       MAD                    AUC          AD
##  Min.   :8.038873e-14   Min.   :0   Min.   :5.270548e-16
##  1st Qu.:8.038873e-14   1st Qu.:0   1st Qu.:5.270548e-16
##  Median :8.038873e-14   Median :0   Median :5.270548e-16
##  Mean   :8.038873e-14   Mean   :0   Mean   :5.270555e-16
##  3rd Qu.:8.038873e-14   3rd Qu.:0   3rd Qu.:5.270548e-16
##  Max.   :8.038873e-14   Max.   :0   Max.   :5.276968e-16
```

```
mt_data$measures %>% select(MAD,AUC,AD) %>% summarise_all(c("sd"))
```

```
##   MAD AUC           AD
## 1   0   0 2.030078e-20
```

Validation 2: Triangular path

- Start click (0,-400) followed by 110 ms pause
- Every 10 ms cursor moves one px up for the first 800 px
- … and then one px left for the next 800 px, i.e., for 16000 ms in total
- Cursor pauses at end position (-800,400) for 100 ms and then clicks

Read and preprocess data

```
raw_data <- read_opensesame("validation_data",extension = "triangular.csv.gz")
```

```
## Reading triangular.csv.gz
```

```
mt_data <- mt_import_mousetrap(raw_data)
mt_data <- mt_remap_symmetric(mt_data,remap_xpos = "no")
mt_data <- mt_measures(mt_data)
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_no_reset <- mt_import_mousetrap(raw_data,reset_timestamps = FALSE)
```

Temporal analyses

```
mt_data$data <- mt_data$data %>%
  mutate(
    time_start_click =
      time_get_start_click+response_time_get_start_click,
    delta_click_stimulus =
      time_present_stimulus-time_start_click,
```

```r
    delta_stimulus_tracking =
      mt_data_no_reset$trajectories[,1,"timestamps"]-time_present_stimulus,
    delta_click_tracking =
      mt_data_no_reset$trajectories[,1,"timestamps"]-time_start_click
  )

summary(select(mt_data$data,starts_with("delta"),response_time),digits = 8)
```

```
##  delta_click_stimulus delta_stimulus_tracking delta_click_tracking response_time
##  Min.   :6.000        Min.   :0.000           Min.   : 7.000       Min.   :16201.000
##  1st Qu.:6.000        1st Qu.:0.000           1st Qu.: 7.000       1st Qu.:16203.000
##  Median :7.000        Median :1.000           Median : 7.000       Median :16203.000
##  Mean   :6.918        Mean   :0.656           Mean   : 7.574       Mean   :16203.134
##  3rd Qu.:7.000        3rd Qu.:1.000           3rd Qu.: 8.000       3rd Qu.:16203.000
##  Max.   :9.000        Max.   :1.000           Max.   :10.000       Max.   :16205.000
```

```r
mt_data$data %>%
  select(starts_with("delta"),response_time) %>% summarise_all(c("sd"))
```

```
##   delta_click_stimulus delta_stimulus_tracking delta_click_tracking response_time
## 1            0.6750791               0.4752787            0.6457096     0.9396419
```

```r
# mousetrap-os response_time matches mt_measures RT
table(mt_data$data$response_time==mt_data$measures$RT)
```

```
##
## TRUE
## 1000
```

Logging resolution

```r
mt_check_resolution(mt_data,desired = 10)
```

```
## $summary
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000  10.000  10.000   9.996  10.000  25.000
##
## $sd
## [1] 0.1786731
##
## $frequencies
## log_diffs
##       1       2       3       4       5       8       9      10      11      12      13
##      57     310     439      65     114       2       3 1619692     284      12       2
##      14      25
##       2       1
##
## $relative_frequencies
## log_diffs
##      1      2      3      4      5      8      9     10     11     12     13     14
## 0.0000 0.0002 0.0003 0.0000 0.0001 0.0000 0.0000 0.9992 0.0002 0.0000 0.0000 0.0000
##     25
## 0.0000
##
```

```
## $frequencies_desired
## log_diffs_class
## smaller desired greater
##     990 1619692     301
##
## $relative_frequencies_desired
## log_diffs_class
## smaller desired greater
##  0.0006  0.9992  0.0002
```

```r
# Percent of lags that lag by 1 ms only
284/301
```

```
## [1] 0.9435216
```

Distances between subsequently recorded positions

```r
# Frequency of distances across both x and y coordinates
table(as.numeric(mt_data$trajectories[,-1,"xpos_dist"]),
      as.numeric(mt_data$trajectories[,-1,"ypos_dist"]))
```

```
##
##            0      1      2
##   -3       1      0      0
##   -2      82      0      0
##   -1  799833      0      0
##   0    21143 799848     76
```

```r
# Frequency of distances for x coordinates
table(as.numeric(mt_data$trajectories[,-1,"xpos_dist"]))
```

```
##
##     -3     -2     -1      0
##      1     82 799833 821067
```

```r
# Percent of distances for x coordinates
round(table(as.numeric(mt_data$trajectories[,-1,"xpos_dist"]))/
        sum(!is.na(mt_data$trajectories[,-1,"ypos_dist"])),6)
```

```
##
##       -3       -2       -1        0
## 0.000001 0.000051 0.493425 0.506524
```

```r
# Frequency of distances for y coordinates
table(as.numeric(mt_data$trajectories[,-1,"ypos_dist"]))
```

```
##
##      0      1      2
## 821059 799848     76
```

```r
# Percent of distances for y coordinates
round(table(as.numeric(mt_data$trajectories[,-1,"ypos_dist"]))/
        sum(!is.na(mt_data$trajectories[,-1,"ypos_dist"])),6)
```

```
##
##        0        1        2
## 0.506519 0.493434 0.000047
```

Comparison of expected and observed position

```r
# 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=10),]
expected$ypos <- (-expected$ypos)

# Set constant for delay between start click and tracking onset
delta_tracking_onset <- 7

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

# Correlation between observed and expected position
cor_xpos <- cor(as.vector(mt_data$trajectories[,,"xpos"]),
                as.vector(mt_data$trajectories[,,"xpos_expected"]),
                use="complete.obs")
cor_ypos <- cor(as.vector(mt_data$trajectories[,,"ypos"]),
                as.vector(mt_data$trajectories[,,"ypos_expected"]),
                use="complete.obs")
print(cor_xpos,digits = 15)
```

```
## [1] 0.999999992661407
```

```r
print(cor_ypos,digits = 15)
```

```
## [1] 0.999999994872842
```

```r
# Compute difference between expected and observed position
mt_data <- mt_add_variables(mt_data,use="trajectories",variables=list(
  xpos_diff = mt_data$trajectories[,,"xpos_expected"]-mt_data$trajectories[,,"xpos"],
  ypos_diff = mt_data$trajectories[,,"ypos_expected"]-mt_data$trajectories[,,"ypos"]
))


# Frequency of differences between observed and expected positions across both dimensions
table(mt_data$trajectories[,,"xpos_diff"],mt_data$trajectories[,,"ypos_diff"])
```

```
##
##          -1     0     1
##   -2      0     1     0
##   -1      0     4     0
```

```
##   0     1136 1619213       2
##   1        0    1627       0
```

```r
# Percent of differences for xpos
round(table(mt_data$trajectories[,,"xpos_diff"])/
        sum(!is.na(mt_data$trajectories[,,"xpos_diff"])),6)
```

```
##
##       -2       -1        0        1
## 0.000001 0.000002 0.998994 0.001003
```

```r
# Percent of differences for ypos
round(table(mt_data$trajectories[,,"ypos_diff"])/
        sum(!is.na(mt_data$trajectories[,,"ypos_diff"])),6)
```

```
##
##       -1        0        1
## 0.000700 0.999298 0.000001
```

Mouse-tracking measures

```r
# Descriptives
summary(select(mt_data$measures,MAD,AUC,AD),digits = 8)
```

```
##       MAD                  AUC                AD
##  Min.   :565.6854   Min.   :320000   Min.   :279.00550
##  1st Qu.:565.6854   1st Qu.:320000   1st Qu.:279.00637
##  Median :565.6854   Median :320000   Median :279.00637
##  Mean   :565.6854   Mean   :320000   Mean   :279.00871
##  3rd Qu.:565.6854   3rd Qu.:320000   3rd Qu.:279.00637
##  Max.   :565.6854   Max.   :320000   Max.   :279.33641
```

```r
mt_data$measures %>% select(MAD,AUC,AD) %>% summarise_all(c("sd"))
```

```
##   MAD AUC         AD
## 1   0   0 0.02159266
```

```r
# Expected MAD
.5*sqrt(800^2+800^2)
```

```
## [1] 565.6854
```

```r
# Expected AUC
.5*800^2
```

```
## [1] 320000
```

```r
# Expected AD
mean(c(seq(0,800,1),seq(799,0,-1))/sqrt(2))*1601/1622
```

```
## [1] 279.0064
```