Package 'kollaR'

September 19, 2025

Type Package

Title Event Classification, Visualization and Analysis of Eye Tracking Data

Version 1.1.2

Description Functions for analysing eye tracking data, including event detection, visualizations and area of interest (AOI) based analyses.
The package includes implementations of the IV-T, I-DT, adaptive velocity threshold, and Identification by two means clustering (I2MC) algorithms.
See separate documentation for each function. The principles underlying I-VT and I-DT algorithms are described in Salvucci & Goldberg (2000,\doi{10.1145/355017.355028}).
Two-means clustering is described in Hessels et al. (2017, \doi{10.3758/s13428-016-0822-1}).
The adaptive velocity threshold algorithm is described in Nyström & Holmqvist (2010,\doi{10.3758/BRM.42.1.188}).
See a demonstration in the URL.

URL https://drjohanlk.github.io/kollaR/demo.html

License GPL-3

Encoding UTF-8

LazyData true

Imports dplyr, ggplot2, zoo, ggforce, tidyr, ggpubr, jpeg, patchwork, shiny, plotly, base64enc, magick, scales

RoxygenNote 7.3.1

NeedsCompilation no

Author Johan Lundin Kleberg [aut, cre]

Maintainer Johan Lundin Kleberg <johan.lundin.kleberg@su.se>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2025-09-19 14:10:02 UTC

Contents

`kollaR-package` *Fixation and Saccade Detection, Visualization, and Analysis of Eye Tracking Data*

Description

Functions for analyzing eye-tracking data, including event detection (sometimes referred to as fixation filtering), visualizations, and area of interest (AOI) based analyses. See separate documentation for each function. Make sure it works with your data. Currently included algoritms are I-VT, I-DT, adaptive velocity threshold and two-means clustering). The principles underlying I-VT and I-DT filters are described in Salvucci & Goldberg (2000,doi:10.1145/355017.355028). Two-means clustering is described in Hessels et al. (2017, doi:10.3758/s1342801608221). The adaptive velocity algorithm is described in Nyström and Holmqvist (2010, doi:10.3758/BRM.42.1.188)

Details

Overview of functions: Pre-processing (smoothing, interpolation, downsampling, calculate sample-to-sample RMS):'preprocess_gaze', 'downsample_gaze', 'calculate_rms'

Fixation and Saccade Classification (Fixation Filters)**: 'algorithm_ivt', 'algorithm_idt', 'algorithm_i2mc', 'algorithm_adaptive'

Fixation and Post-Processing)**: 'merge_adjacent_fixations', 'trim_fixations'

Visualization of Output from Fixation and Saccade Detection and Preprocessing Algorithms: 'fixation_plot_ts', 'fixation_plot_temporal', 'fixation_plot_2d', 'plot_velocity_profiles', 'plot_sample_velocity', 'plot_algorithm_results'

Visualization of Gaze Data: 'static_plot', 'animated_fixation_plot'

AOI Based Analyses: 'draw_aoi', 'aoi_test'

Author(s)

Johan Lundin Kleberg <johan.lundin.kleberg@su.se>

`adjust_fixation_timing` *Adjust the onset and offset of fixations to avoid misclassification of saccade samples as belonging to fixations*

Description

Shrink the period classified as a fixation by removing samples close to the onset and offset with excessive differences from the fixation center. This reduces the risk that samples belonging to saccades are misclassified as belonging to a fixation. The function is used internally by other functions and should typically not be called outside of them.

adjust_fixation_timing starts by calculating the median (MD) and MAD of the absolute distances from the fixation center of all included samples. The fixation onset is shifted forwards to the first sample with a distance to the fixation center under t* MAD + MD where t is specified by the input parameter threshold. Analogously, fixation offset is shifted backwards to the last included sample with distance to the fixation center under t* MAD + MD

Usage

```
adjust_fixation_timing(
  fixation.candidate.starts,
  fixation.candidate.stops,
  x,
  y,
  threshold = 3
)
```

Arguments

`fixation.candidate.starts`	First row in the data included in the fixation
`fixation.candidate.stops`	Last row in the data included in the fixation
`x`	X coordinates
`y`	Y coordinates
`threshold`	Threshold for highest accepted distance from fixation center in MADs from the median. Default 3. If NA, just remove NAs at the onset and offest of fixation but ignore deviations from fixation center

Value

data frame with adjusted first and last row of the fixation

---

algorithm_adaptive *Adaptive velocity-based algorithm for saccade and fixation detection*

Description

The function is based on a procedure suggested by Nyström and Holmqvist 2010. Behavior Research Methods, 42, 188-204. Velocity thresholds for saccade onset and offset are adapted to the specific data at hand. Please see Nyström and Holmqvist (2010) for a description of this procedure. STEP 1: The function starts by identifying peak velocities larger than an initial threshold (peak threshold, specified by the parameter peak.threshold.start), and then iteratively adjusts this threshold through the following steps: A) The mean (M) and standard deviation (SD) of the velocities of all samples with velocties below the peak threshold are calculated. B) the updated peak threshold is defined as M +6SD. C) Steps A-B are repeated until the difference between the old and new peak

threshold is < 1 degree. D) The threshold for identification of saccade onsets (saccade onset threshold) is defined as M + onset.threshold *SD. For each segment in the data with velocities above peak threshold, go through steps 2-3:

STEP 2: Saccade onset is defined by searching backwards from the leftmost sample with velocity above peak threshold to the first sample with a velocity above saccade onset threshold and a higher velocity than the previous sample. STEP 3: Define saccade offset threshold as a weighted sum of a) the saccade onset threshold and b) the 'local noise factor' defined as mean + 3SD of sample-to-sample velocity during a period just before saccade onset. This period has the same length as the minimum fixation duration (specified by the parameter min.fixation.duration) Saccade onset is defined by searching forward from the rightmost sample with a velocity above peak threshold to the first sample with a velocity below saccade offset threshold and a lower velocity than the previous sample. STEP 4: Fixations are defined as periods between saccades (as in the `I-VT` algorithm)

If the function can not detect a sample which fulfills both the threshold criterion and the acceleration/deceleration criterion during step 2-3, it will try to find a sample that fulfills the threshold criterion. If this fails, the saccade is discarded.

The input data should be pre-processed (e.g., noise removal and interpolation over gaps)

The output data is a list with three data frames: fixations includes all detected fixations with coordinates, duration and a number of other metrics, saccades includes data for saccades, filt.gaze is a sample-by-sample data frame with time stamps, and gaze coordinates before ("raw") and after fixation detection. The function has a number of parameters for removing potentially invalid fixations and saccades. The parameter min.fixation.duration can be used to remove unlikely short fixations. If the parameter missing.samples threshold is set to a value lower than 1, fixations with a higher proportion of missing raw samples are removed.

Usage

```
algorithm_adaptive(
  gaze,
  min.fixation.duration = 40,
  min.saccade.duration = 10,
  min.saccade.amplitude = 1,
  xcol = "x.raw",
  ycol = "y.raw",
  save.velocity.profiles = FALSE,
  missing.samples.threshold = 0.5,
  merge.ms.threshold = 75,
  distance.threshold = 0.7,
  alpha = 0.7,
  beta = 0.3,
  one_degree = 40,
  velocity.filter.ms = 10,
  peak.threshold.start = 200,
  onset.threshold.sd = 3,
  min.period.ms = 40,
  margin.ms = 3,
  trim.fixations = TRUE,
  trim.dispersion.threshold = NA
)
```

Arguments

gaze	Data frame with gaze data before saccade and fixation data identification. The data frame must include the variable timestamp with timing in milliseconds and columns for x and y coordinates specified by the columns 'xcol' and 'ycol' respectively.
min.fixation.duration	Minimum duration of accepted fixations. This parameter is also used to calculate 'local noise' prior to the onset of a saccade. Default: 40
min.saccade.duration	Minimum duration of accepted saccades in ms. Default: 10
min.saccade.amplitude	Minimum amplitude of accepted saccades in degrees. Default 1
xcol	column in the gaze data frame where x coordinates are found. Default: x.raw
ycol	column in the gaze data frame where y coordinates are found. Default: y.raw
save.velocity.profiles	If TRUE, save velocity profiles of each saccade. Default: FALSE.
missing.samples.threshold	Remove fixations with a higher proportion of missing samples. Range 0 to 1.
merge.ms.threshold	Subsequent fixations occuring within this time window and distance specified by distance.threshold are merged. Set to 0 if you don't want to merge fixations.
distance.threshold	Subsequent fixations occurring withing this distance are merged. Set to 0 if you don't want to merge fixations.
alpha	Weight of the saccade onset threshold when defining threshold for saccade offset
beta	Weight of local noise factor when defining threshold for saccade offset
one_degree	one degree of the visual field in the unit of the x and y coordinates in the data. Typically pixels or degrees.
velocity.filter.ms	If velocity.filter.ms is not NA, the velocity vector is smoothed using a moving median filter corresponding to this value in ms before the propose threshold is identified. Default: 10.
peak.threshold.start	Initial peak threshold value in degrees of the visual field. Default: 200
onset.threshold.sd	Number of standard deviations used to define saccade onset threshold
min.period.ms	Peak velocity threshold will be iteratively updated based on periods in the data in which consecutive values of at least this time period are below the previous threshold estimate.
margin.ms	A margin in ms around periods of identified consecutive values below the previous threshold estimate which will be excluded from the estimates. This makes the algorithm more robust to noise.

`trim.fixations` If TRUE, the onset of each fixation will be shifted forwards to the first non-missing (non-NA) sample during the period. The offset will be shifted backwards to the last non-missing sample. If TRUE, and the parameter trim.dispersion.threshold is a positive number, samples at the margins with large distances from the centroid will also be excluded

`trim.dispersion.threshold`
If not NA and trim.fixations is TRUE, fixation onsets and offests will also be shrinked to exclude any samples at the margins with a larger distance from the fixation centroid than trim.dispersion.threshold * MAD.

Value

list including separate data frames for fixations, saccades, and sample-by-sample data

---

`algorithm_i2mc` *Fixation detection by two-means clustering*

Description

Identify fixations in a gaze matrix using identification by two-means clustering. The algorithm is based on Hessels et al 2017. Behavior research methods, 49, 1802-1823. Data from the left and right eye are not processed separately. Adjust your analysis scripts to include this steps if you want the algorithm to include this step, as in Hessels et al 2017. Input data must be a data frame with the variables timestamp, x.raw and y.raw as variables. Other variables can be included but will be ignored. This function does not perform pre-processing in the form of interpolation or smoothing. Use the function process.gaze for this. Timestamps are assumed to be in milliseconds. X and y coordinates can be in pixels or proportion of the screen. Make sure that the parameter one_degree is consistent with the format of your data. The output data is a list with two data frames: fixations includes all detected fixations with coordinates, duration and a number of other metrics, filt.gaze is a sample-by-sample data frame with time stamps, raw gaze coordinates (e.g., before fixation detection) and fixation coordinates. If the input downsampling.factors is not empty, transition weights will be calculated based on the data in the original sampling rate and data at all sampling rate specified in this variable. According to Hessels et al 2017, this step makes the analysis less vulnerable to noise in the data. If the parameter threshold.on.off is not NA, the onsets and offsets of fixations will be shifted to exclude samples at the margins with a larger absolute distance from the fixation centroid than a threshold value. The threshold value is defined as the median of all absolute distances from the centroid plus threshold.on.off * MAD of the absolute distances. Default threshold is 3. Samples with large distances from the fixation centroid right at the onset and offset of a fixation may belong to a saccade.

Usage

```
algorithm_i2mc(
  gaze_raw,
  windowlength.ms = 200,
  distance.threshold = 0.7,
  one_degree = 40,
```

```
  window.step.size = 6,
  min.fixation.duration = 40,
  weight.threshold = 2,
  xcol = "x.raw",
  ycol = "y.raw",
  merge.ms.threshold = 40,
  downsampling.factors = NA,
  missing.samples.threshold = 0.5,
  threshold.on.off = 3
)
```

Arguments

`gaze_raw`	Data frame with gaze data prior to fixation detection. Include the variable timestamp with timing in ms and columns with x and y data as specified by the parameters xcol and ycol or their default values
`windowlength.ms`	Length of the moving analysis windows
`distance.threshold`	Subsequent fixations occurring withing this distance are merged. Set to 0 if you do not want to merge fixations.
`one_degree`	One degree of the visual field in the unit of the raw x and y coordinates, typically pixels or proportion of the screen. Make sure that the setting matches the format of your data
`window.step.size`	Distance between starting points of subsequent analysis windows in samples
`min.fixation.duration`	Minimum duration of accepted fixations. Shorter fixations are discarded
`weight.threshold`	Samples with a transition weight exceeding it are candidates for fixation detection.
`xcol`	Name of the column where raw x values are stored. Default: x.raw
`ycol`	Name of the column where raw y values are stored. Default: y.raw
`merge.ms.threshold`	Only fixations occurring within this time window in milliseconds are merged
`downsampling.factors`	Factors to downsample the data by in calculating fixation weights. If downsampling.factors has the values `c(10, 2)`, transition weights will be calculated base on data in the original sampling rate as well as the two downsampled data sets.
`missing.samples.threshold`	Remove fixations with a higher proportion of missing samples. Range 0 to 1.
`threshold.on.off`	if not NA, shift fixation onset and offset to exclude samples at the margin with absolute distances from the fixation center > threshold.on.off * MAD (distance) + median (distance)

Value

list including separate data frames for fixations and sample-by-sample data including gaze coordinates before ("raw") and after fixation detection. The "fixations" data frame gives onset, offset, x, y, sample-to-sample root-mean-square deviations (RMSD, precision), RMSD from fixation centroid, and missing samples of each fixation.

Examples

```
gaze <- algorithm_i2mc(sample.data.processed)
```

---

algorithm_idt *Dispersion-based fixation detection algorithm* (I-DT)

Description

Apply a dispersion-based fixation (I-DT) filter to the eye tracking data. The algorithm identifies fixations as samples clustering within a spatial area. The procedure is described in Blignaut 2009 Input data must be a data frame with the variables timestamp, x.raw and y.raw as variables. Other variables can be included but will be ignored. This function does not perform pre-processing in the form of interpolation or smoothing. Use the function preprocess_gaze for this. Timestamps are assumed to be in milliseconds. The output data is a list with two data frames: fixations includes all detected fixations with coordinates, duration and a number of other metrics, filt.gaze is a sample-by-sample data frame with time stamps, raw gaze coordinates (e.g., before event detection) and fixation coordinates. The function can be very slow for long recordings and/or data recorded at high sampling rates.

Usage

```
algorithm_idt(
  gaze_raw,
  one_degree = 40,
  dispersion.threshold = 1,
  min.duration = 50,
  xcol = "x.raw",
  ycol = "y.raw",
  distance.threshold = 0.7,
  merge.ms.threshold = 75,
  missing.samples.threshold = 0.5
)
```

Arguments

gaze_raw	Data frame with gaze data before fixation detection. Include the variable timestamp with timing in ms and columns with raw x and y data as specified by the parameters xcol and ycol or their default values

one_degree — One degree of the visual field in the unit of the raw x and y coordinates. The unit is typically pixels or proportion of the screen. Make sure that the setting matches your data

dispersion.threshold — Maximum radius of fixation candidates. Samples clustering within a circle of this limit will be classified as a fixation if the duration is long enough.

min.duration — Minimum duration of fixations in milliseconds

xcol — Name of the column where raw x values are stored. Default: x.raw

ycol — Name of the column where raw y values are stored. Default: y.raw

distance.threshold — Subsequent fixations occurring withing this distance are merged. Set to 0 if you don't want to merge fixations.

merge.ms.threshold — Only subsequent fixations occurring within this time window are merged

missing.samples.threshold — Remove fixations with a higher proportion of missing samples. Range 0-1

Value

list including separate data frames for fixations and sample-by-sample data including gaze coordinates before and after fixation detection. The fixations data frame includes onset, offset, x, y, sample-to-sample root-mean-square deviations (RMSD, precision), RMSD from fixation centroid, and missing samples of each fixation.

---

algorithm_ivt *I-VT algorithm for fixation and saccade detection*

Description

Apply an I-VT event detection algorithm to the eye tracking data. The algorithm identifies saccades as periods with sample-to-sample velocity above a threshold and fixations as periods between saccades. See Salvucci and Goldberg 2000. Identifying fixations and saccades in eye tracking protocols. Proc. 2000 symposium on Eye tracking research and applications for a description.

Input data must be a data frame with the variables timestamp, x.raw and y.raw as variables. Other variables can be included but will be ignored. This function does not perform pre-processing in the form of interpolation or smoothing. Use the function preprocess_gaze for this. Timestamps are assumed to be in milliseconds. X and y coordinates can be in pixels or proportion of the screen depending on the format of your data. Make sure that the parameter one_degree matches the unit of your data The output data is a list with three data frames: fixations includes all detected fixations with coordinates, duration and a number of other metrics, saccades includes data for saccades, filt.gaze is a sample-by-sample data frame with time stamps, and gaze coordinates before ("raw") and after fixation detection. The function has a number of parameters for removing potentially invalid fixations and saccades. The parameter min.fixation.duration can be used to remove unlikely short fixations. If the parameter missing.samples threshold is set to a value lower than 1, fixations with a higher proportion of missing raw samples are removed.

Usage

```
algorithm_ivt(
  gaze_raw,
  velocity.filter.ms = 20,
  velocity.threshold = 35,
  min.saccade.duration = 10,
  min.saccade.amplitude = 1,
  min.fixation.duration = 40,
  one_degree = 40,
  save.velocity.profiles = FALSE,
  xcol = "x.raw",
  ycol = "y.raw",
  distance.threshold = 0.7,
  merge.ms.threshold = 75,
  missing.samples.threshold = 0.5,
  trim.fixations = FALSE,
  trim.dispersion.threshold = NA
)
```

Arguments

`gaze_raw`	Data frame with gaze data before fixation and saccade detection. Include the variable timestamp with timing in ms and columns with raw x and y data as specified by the parameters xcol and ycol or their default values
`velocity.filter.ms`	Window in milliseconds for moving average window used for smoothing the sample to sample velocity vector.
`velocity.threshold`	Velocity threshold for saccade detection in degrees/second
`min.saccade.duration`	Minimum duration of saccades in milliseconds
`min.saccade.amplitude`	Minimum amplitude of saccades in degrees
`min.fixation.duration`	Minimum duration of fixations in milliseconds
`one_degree`	One degree of the visual field in the unit of the raw x and y coordinates, typically pixels or proportion of the screen. Make sure that it is consistent with the format of your data
`save.velocity.profiles`	If TRUE, return velocity profiles of each detected saccade as a variable in the saccades data frame
`xcol`	Name of the column where raw x values are stored. Default: x.raw
`ycol`	Name of the column where raw y values are stored. Default: y.raw
`distance.threshold`	Subsequent fixations occurring withing this distance are merged. Set to 0 if you don't want to merge fixations.

`merge.ms.threshold`
: Subsequent fixations occuring within this time window and distance specified by distance.threshold are merged. Set to 0 if you don't want to merge fixations.

`missing.samples.threshold`
: Remove fixations with a higher proportion of missing samples. Range 0 to 1.

`trim.fixations`
: If TRUE, the onset of each fixation will be shifted forwards to the first non-missing (non-NA) sample during the period. The offset will be shifted backwards to the last non-missing sample. If TRUE, and the parameter trim.dispersion.threshold is a positive number, samples at the margins with large distances from the centroid will also be excluded

`trim.dispersion.threshold`
: If not NA and trim.fixations is TRUE, fixation onsets and offsets will also be shrinked to exclude any samples at the margins with a larger distance from the centroid than trim.dispersion.threshold*MAD.

Value

list including separate data frames for 1) fixations, 2)sample-by-sample data including x and y coordinates before and after fixation detection, and 3) sample-to-sample velocity The fixations data frame gives onset, offset, x, y, sample-to-sample RMSD (precision),root-mean-square deviations from fixation centroid, and missing samples of each fixation.

Examples

```
ivt_data <- algorithm_ivt(sample.data.processed, velocity.threshold = 30,
min.fixation.duration = 40)
```

---

animated_fixation_plot

Create GIF animation of fixations on a stimulus images

Description

This function plots and returns a .gif showing fixations on a background with one or multiple images, typically the stimuli. The interval to plot is defined by sample numbers. Fixations must have the variables x, y, and onset. The function works with .jpg images. If paths to multiple images are given, all will be displayed. Fixations are shown on a white background if no background images are defined. .gif images can be saved to a file. Gaze data are plotted on a reversed y-axis where x and y are 0 is the upper left corner, corresponding to the structure of data from Tobii eye trackers. If there are multiple participants specified in the variable id, each participant will get a unique color. You may get an error message if some participants lack data during single frames. This is usually no cause for concern.

Usage

```
animated_fixation_plot(
  gazedata,
  xres = 1920,
  yres = 1080,
  plot.onset,
  plot.offset,
  background.images = NA,
  filename = "scanpath.gif",
  save.gif = FALSE,
  gif.dpi = 300,
  gazepoint.size = 2,
  n.loops = 1,
  show.legend = TRUE,
  id_color_map = NA,
  resolution.scaling = 0.5,
  framerate = 10,
  show.progress = TRUE
)
```

Arguments

gazedata	Data frame with fixation data which must include columns for x and y coordinates as well as the variable onset which indicates the onset of the fixation. Make sure the onset variables match the timing the plot.onset and plot.offset input. If the categorical or factor variable id is included, separate colors will represent each participant. Make sure the onset variables match the timing the plot.onset and plot.offset input.
xres	horizontal resolution of the screen or area to plot on. Default 1920
yres	vertical resolution of the screen or area to plot on. Default 1080
plot.onset	Onset of the interval in the gaze_data$onset variable to plot in the same unit, typically milliseconds
plot.offset	Offset of the interval in the corresponding to the variable onset in the input data frame gazedata to plot in the same unit, typically milliseconds
background.images	data frame with information about background images to use as background. The data frame must include the variables min.x, min.y, max.x, and max.y variables representing where the images should be placed on the background, the variable path specifying a full file path, and the onset and offset of each image in units corresponding to the time stamps of the gazedata matrix. Background images should be in JPEG format. This is an example: `background.images <- data.frame( min.x = c(100, 800), min.y = c(100, 100), max.x = c(300, 100), max.y = c(600, 600), path = c("~/path_to_image1/image1.jpg", "~/path_to_image1/image2.j onset = c(1, 4000), offset = c(4000, 6000)`
filename	Name of path where the .gif is saved
save.gif	If TRUE, save the created .gif file under the name specified in the filename parameter

gif.dpi	Resolution in dpi if .gif is saved. Lower values give smaller files but the resoulution will be poorer.
gazepoint.size	Size of marker representing fixation coordinates.
n.loops	Specify the number of times to play the plotted sequence. Default is 1. If n.loops is 0, the .gif will play in an eternal loop
show.legend	If TRUE, show values of the variable id in legend
id_color_map	A character vector with HEX color codes for each id. If NA, a color map with unique colors for each id is created by the function. You can create a specific color map for your data using the following code: new_color_map <- c("#FB61D7", "#00C094") names(new_color_map) <- c("Id1", "Id2")
resolution.scaling	Scaling of the original images and gaze data. Default is 0.5. Decreasing the size of the images can make the function quicker. This can be useful if you want to assign specific colors for different groups
framerate	Frames per seconds of the returned animation. Default 10
show.progress	If TRUE, show progression of the function in the prompt

Value

a magick animation of raw and fixated values plotted on the y axis and sample number on the x axis

---

aoi_test *Test whether a gaze coordinates are within or outside a rectangular or elliptical AOI. The aois df must contain the variables x0, x1, y0 and y1. x0 is the minimum x value, y0 the minimum y value. x1 the maximum x value. y1 the maximum y value and type where rect means that the AOI is a rectangle and circle that the AOI is a circle or ellipse If a column called name is present, the output for each AOI will be labelled accordingly. Otherwise, the output will be labelled according to the order of the AOI in the data frame. The df 'gaze' must contain the variables onset, duration, x, and y. Latency will be defined as the value in onset of the first detected gaze coordinate in the AOI Make sure that the timestamps are correct! The function can be used with gaze data either fixations, saccades, or single samples. Note that the output variables are not equally relevant for all types of gaze data. For example, both total duration and latency are relevant in many analyses focusing on fixations, but total duration may be less relevant in analyses of saccades.*

Description

Test whether a gaze coordinates are within or outside a rectangular or elliptical AOI. The aois df must contain the variables x0, x1, y0 and y1. x0 is the minimum x value, y0 the minimum y value. x1 the maximum x value. y1 the maximum y value and type where rect means that the AOI is a rectangle and circle that the AOI is a circle or ellipse If a column called name is present, the output

for each AOI will be labelled accordingly. Otherwise, the output will be labelled according to the order of the AOI in the data frame. The df 'gaze' must contain the variables onset, duration, x, and y. Latency will be defined as the value in onset of the first detected gaze coordinate in the AOI Make sure that the timestamps are correct! The function can be used with gaze data either fixations, saccades, or single samples. Note that the output variables are not equally relevant for all types of gaze data. For example, both total duration and latency are relevant in many analyses focusing on fixations, but total duration may be less relevant in analyses of saccades.

Usage

```
aoi_test(gaze, aois, outside = FALSE)
```

Arguments

gaze	data frame with each row representing a gaze coordinate from a fixation, saccade, or sample. Must include the variables x, y, duration, and onset. Onset zero should typically be trial onset
aois	data frame with AOIs.
outside	If FALSE, summarize data within AOIs. If TRUE, summarize data outside AOIs.

Value

data frame with total duration, number of occurrences and latency to first detected gaze coordinates for each AOI. Data are in long format.

---

calculate_rms *Calculate sample-to-sample root mean square distance (RMS) of a data segment*

Description

Calculate the RMS of a gaze segment with X and Y coordinates. RMS is inversely related to precision, so that lower RMS indicates higher precision calculate_rms calculates the root mean square of the Euclidean distance between subsequent samples Adjust the parameters 'xcol' and 'ycol' to specify the data you want to work with. For example, it is possible to calculate the RMS of the left and right eye separately by specifying appropriate columns.

Usage

```
calculate_rms(gaze_in, xcol = "x.raw", ycol = "y.raw", one_degree = 40)
```

Arguments

gaze_in	data frame with gaze to process. Must contain the variables specified in the parameters 'xcol' and 'ycol'
xcol	Name of column containing x coordinates
ycol	Name of column containing y coordinates
one_degree	One degree of the visual field in the units of the X and Y coordinates as specified in 'xcol' and 'ycol'

Value

sample-to-sample RMS

Examples

```
rms <- calculate_rms(sample.data.unprocessed)
```

---

cluster2m *Fixation detection by two-means clustering*

Description

Identify fixations in a gaze matrix using identification by two-means clustering. The algorithm is based on Hessels et al 2017. Behavior research methods, 49, 1802-1823. Data from the left and right eye are not processed separately. Adjust your analysis scripts to include this steps if you want the algorithm to include this step, as in Hessels et al 2017. Input data must be a data frame with the variables timestamp, x.raw and y.raw as variables. Other variables can be included but will be ignored. This function does not perform pre-processing in the form of interpolation or smoothing. Use the function process.gaze for this. Timestamps are assumed to be in milliseconds. X and y coordinates can be in pixels or proportion of the screen. Make sure that the parameter one_degree is consistent with the format of your data! The output data is a list with two data frames: fixations includes all detected fixations with coordinates, duration and a number of other metrics, filt.gaze is a sample-by-sample data frame with time stamps, gaze coordinates before fixation detection ("raw") for and fixation coordinates. If the input downsampling.factors is not empty, transition weights will be calculated based on the data in the original sampling rate and data at all sampling rate specified in this variable. According to Hessels et al 2017, this step makes the analysis less vulnerable to noise in the data.

Usage

```
cluster2m(
  gaze_raw,
  windowlength.ms = 200,
  distance.threshold = 0.7,
  one_degree = 40,
  window.step.size = 6,
  min.fixation.duration = 40,
```

```
  weight.threshold = 2,
  xcol = "x.raw",
  ycol = "y.raw",
  merge.ms.threshold = 40,
  downsampling.factors = NA,
  missing.samples.threshold = 0.5
)
```

Arguments

gaze_raw	Data frame with gaze data before fixation detection. Include the variable timestamp with timing in ms and columns with x and y data as specified by the parameters xcol and ycol or their default values
windowlength.ms	Length of the moving analysis windows
distance.threshold	Subsequent fixations occurring withing this distance are merged. Set to 0 if you do not want to merge fixations.
one_degree	One degree of the visual field in the unit of the raw x and y coordinates, typically pixels or proportion of the screen. Make sure that the setting matches the format of your data
window.step.size	Distance between starting points of subsequent analysis windows in samples
min.fixation.duration	Minimum duration of accepted fixations. Shorter fixations are discarded
weight.threshold	Samples with a transition weight exceeding it are candidates for fixation detection.
xcol	Name of the column where raw x values are stored. Default: x.raw
ycol	Name of the column where raw y values are stored. Default: y.raw
merge.ms.threshold	Only fixations occurring within this time window in milliseconds are merged
downsampling.factors	Factors to downsample the data by in calculating fixation weights. If downsampling.factors has the values `c(10, 2)`, transition weights will be calculated base on data in the original sampling rate as well as the two downsampled data sets.
missing.samples.threshold	Remove fixations with a higher proportion of missing samples. Range 0 to 1.

Value

list including separate data frames for fixations and sample-by-sample data including gaze coordinates before fixation classification ("raw") and fixation coordinates. The "fixations" data frame gives onset, offset, x, y, sample-to-sample root-mean-square deviations (RMSD, precision), RMSD from fixation centroid, and missing samples of each fixation.

Examples

```
gaze <- cluster2m(sample.data.processed)
```

`downsample_gaze` *Downsample gaze*

Description

This function downs-samples gaze by a specified factor. Data are down-sampled by splitting the data in bins and calculating the mean of each bin.

Usage

```
downsample_gaze(data_in, ds.factor, xcol = "x", ycol = "y")
```

Arguments

`data_in`	Data frame which must contain the variables specified by the parameters xcol and ycol.
`ds.factor`	The factor to down-sample by. For example, setting ds.factor to 10 down-samples data recorded at 1000 HZ to 100 HZ.
`xcol`	Name of the column where raw x values are stored. Default: x
`ycol`	Name of the column where raw y values are stored. Default: y

Value

Data frame with downsampled gaze data. The output variables are x, y, and the numbers of the first and last samples of the original data frame included in the bin.

`draw_aois` *Draw one or more areas of interest, AOIs, on a stimulus image and save to the R prompt. The input is the path to a 2D image. Supported file formats: JPEG, BMP, PNG. The function returns a data frame with all saved AOIs. By default, AOIs are drawn in a coordinate system where y is 0 in the lower extreme of the image, e.g., an ascending y axis. Tobii eye trackers use a coordinate system with a descending y-axis, e.g., x and y are 0 in the upper left corner of the image. Make sure that your AOIS match the coordinate system of your eye tracker output. By setting the parameter reverse.y.axis to TRUE, the saved AOIs will be reformatted to fit a coordinate system with a descending y-axis. All AOIS have the variables x0, x1, y0 and y1. x0 is the minimum x value, y0 the minimum y value. x1 the maximum x value. y1 the maximum y value*

Description

Draw one or more areas of interest, AOIs, on a stimulus image and save to the R prompt. The input is the path to a 2D image. Supported file formats: JPEG, BMP, PNG. The function returns a data frame with all saved AOIs. By default, AOIs are drawn in a coordinate system where y is 0 in the lower extreme of the image, e.g., an ascending y axis. Tobii eye trackers use a coordinate system with a descending y-axis, e.g., x and y are 0 in the upper left corner of the image. Make sure that your AOIS match the coordinate system of your eye tracker output. By setting the parameter reverse.y.axis to TRUE, the saved AOIs will be reformatted to fit a coordinate system with a descending y-axis. All AOIS have the variables x0, x1, y0 and y1. x0 is the minimum x value, y0 the minimum y value. x1 the maximum x value. y1 the maximum y value

Usage

```
draw_aois(image.path, reverse.y.axis = FALSE)
```

Arguments

`image.path`	path to a valid image file with the suffix .jpeg, .jpg, .png or .bmp
`reverse.y.axis`	If TRUE, save AOIs positioned on a reverse Y-axis where y is 0 in the upper end of the image. Note that AOIs will be converted to fit a reversed Y axis before printed in the R prompt and saved, but will be shown in the original coordinate system when plotted inside the function.

Value

data frame with type and coordinates of drawn AOIs

`filt_plot_2d` *Plot fixations vs. individual sample coordinates in 2D space. In the current release, filt_plot_2d is a wrapper around fixation_plot_2d which accepts the same arguments.*

Description

This function plots and returns a ggplot2 figure showing fixations and individual gaze coordinates plotted against time. The interval to plot can be defined as a proportion of the data frame or by sample numbers. This function uses one data.frame with fixations and one with sample-by-sample raw data

Usage

```
filt_plot_2d(
  raw.data,
  fixation.data,
  plot.window = c(NA, NA),
  raw.columns = c("x.raw", "y.raw"),
  fixation.columns = c("x", "y"),
```

```
  fixation.radius = 40,
  xres = 1920,
  yres = 1080,
  order.vertical = FALSE,
  verbose = TRUE
)
```

Arguments

raw.data	gaze matrix which must include columns for x and y coordinates in the and raw data (single samples) as specified in the raw.columns parameter
fixation.data	Data frame with fixation data which must include columns for fixation x and y coordinates as specified in the fixation.columns parameter as well as the variable onset which indicates the onset of the fixation. Make sure the onset varables match the timing in the raw.data df
plot.window	vector defining the time window to plot. If left empty, the 50-65 <0, they are assumed to be proportions, e.g., `plot.window = c(0.3,0.35)` plots the 30-35 percent of max.length interval of the data. Numbers >1 are assumed to refer to sample order in the data
raw.columns	Names of variable containing raw data. Default x.raw and y.raw
fixation.columns	Names of variable containing filtered data. Default x and y
fixation.radius	Radius of circles showing fixations.
xres	horizontal resolution of the screen or area to plot on. Default 1920
yres	vertical resolution of the screen or area to plot on. Default 1080
order.vertical	If TRUE, stack subplots on top of each other in a single column
verbose	if TRUE, print the resulting plot

Value

a ggplot of raw and fixated values plotted on the y axis and sample number on the x axis

---

`filt_plot_temporal` *Plot fixation filtered vs. raw gaze coordinates. This function will be replaced by fixation_plot_temporal in future releases. It is currently a wrapper around fixation_plot_temporal accepting the same arguments.*

Description

This function plots and returns a ggplot2 figure showing two time series of gaze coordinates plotted against time. The interval to plot can be defined as a proportion of the data frame, by timestamps, or by sample numbers. Use this function to plot data before and after processing or filtering to examine their effects. For example, unprocessed x or y coordinates can be plotted against x and y coordinates following pre-processing and/or event detection with a fixation detection algorithm Either the x or the y vector is plotted

Usage

```
filt_plot_temporal(
  data_in,
  plot.window = c(NA, NA),
  var1 = "x.raw",
  var2 = "x",
  x.index.var = "sample.index",
  verbose = TRUE
)
```

Arguments

data_in	gaze matrix which must include columns for filtered and unfiltered data as specified in the var1 and var2 paramters
plot.window	vector defining the time window to plot. If left empty, the 50-65 <0, they are assumed to be proportions, e.g., plot.window = c(0.3,0.35) plots the 30-35 in the data found in the variable 'timestamp'
var1	Name of the first variable to plot. Default "x.raw"
var2	Name of the second variable to plot (overlayed on var1) Default: "x"
x.index.var	Name of the variable to plot on the X axis, for example timestamp or sample index. If the default setting is used, gaze coordinates are plotted against sample number in the selected data interval.
verbose	If TRUE, print the resulting plot

Value

a ggplot with gaze coordinates plotted on the y axis and sample number on the x axis

Examples

```
new.plot <- filt_plot_temporal(sample.data.classified, plot.window = c(1000, 2000))
```

---

find.transition.weights

Find transition weights for each sample in a gaze matrix.

Description

This function is used internally by the function algorithm_i2mc

Usage

```
find.transition.weights(data_in, window.step.size = 6, windowsize)
```

Arguments

data_in	Input data
window.step.size	Step size
windowsize	Window size

Value

transition weights.

---

find.valid.periods *Find subsequent periods in a vector with values below a threshold. Used internally by the function suggest_threshold*

Description

This function is used internally by suggest_threshold.

Usage

```
find.valid.periods(data_in, threshold, min_samples, margin = 0)
```

Arguments

data_in	Data to process
threshold	Search for values under this threshold
min_samples	Minimum length of consecutive run in samples
margin	Shrink the period of consecutive runs at both ends with this margin

---

fixation_plot_2d *Plot fixations vs. individual sample coordinates in 2D space.*

Description

This function plots and returns a ggplot2 figure showing fixations and individual gaze coordinates plotted against time. The interval to plot can be defined as a proportion of the data frame or by sample numbers. This function uses one data.frame with fixations and one with sample-by-sample raw data

Usage

```
fixation_plot_2d(
  raw.data,
  fixation.data,
  plot.window = c(NA, NA),
  raw.columns = c("x.raw", "y.raw"),
  fixation.columns = c("x", "y"),
  fixation.radius = 40,
  xres = 1920,
  yres = 1080,
  xmin = 1,
  ymin = 1,
  order.vertical = FALSE,
  font.size = 15,
  verbose = TRUE
)
```

Arguments

raw.data	gaze matrix which must include columns for x and y coordinates in the and raw data (single samples) as specified in the raw.columns parameter
fixation.data	Data frame with fixation data which must include columns for fixation x and y coordinates as specified in the fixation.columns parameter as well as the variable onset which indicates the onset of the fixation. Make sure the onset varables match the timing in the raw.data df
plot.window	vector defining the time window to plot. If left empty, the 50-65 <0, they are assumed to be proportions, e.g., `plot.window = c(0.3,0.35)` plots the 30-35 percent of max.length interval of the data. Numbers >1 are assumed to refer to sample order in the data
raw.columns	Names of variable containing raw data. Default x.raw and y.raw
fixation.columns	Names of variable containing filtered data. Default x and y
fixation.radius	Radius of circles showing fixations.
xres	Maximum of the X axis (horizontal resolution of the screen or area to plot on). Default 1920
yres	Maximum of the Y axis (vertical resolution of the screen or area to plot on). Default 1080
xmin	Minimum of the X axis (default 1).
ymin	Minimum of the Y axis (default 1)
order.vertical	If TRUE, stack subplots on top of each other in a single column
font.size	Text font size
verbose	if TRUE, print the resulting plot

Value

a ggplot of raw and fixated values plotted on the y axis and sample number on the x axis

---

`fixation_plot_temporal` *Plot fixation classified vs. raw gaze coordinates*

Description

This function plots and returns a ggplot2 figure showing two time series of gaze coordinates plotted against time. The interval to plot can be defined as a proportion of the data frame, by timestamps, or by sample numbers. Use this function to plot data before and after processing or filtering to examine their effects. For example, unprocessed x or y coordinates can be plotted against x and y coordinates following pre-processing and/or event detection with a fixation detection algorithm Either the x or the y vector is plotted

Usage

```
fixation_plot_temporal(
  data_in,
  plot.window = c(NA, NA),
  var1 = "x.raw",
  var2 = "x",
  x.index.var = "sample.index",
  verbose = TRUE
)
```

Arguments

`data_in`	gaze matrix which must include columns for filtered and unfiltered data as specified in the var1 and var2 paramters
`plot.window`	vector defining the time window to plot. If left empty, the 50-65 <0, they are assumed to be proportions, e.g., `plot.window = c(0.3,0.35)` plots the 30-35 in the data found in the variable 'timestamp'
`var1`	Name of the first variable to plot. Default "x.raw"
`var2`	Name of the second variable to plot (overlayed on var1) Default: "x"
`x.index.var`	Name of the variable to plot on the X axis, for example timestamp or sample index. If the default setting is used, gaze coordinates are plotted against sample number in the selected data interval.
`verbose`	If TRUE, print the resulting plot

Value

a ggplot with gaze coordinates plotted on the y axis and sample number on the x axis

Examples

```
new.plot <- fixation_plot_temporal(sample.data.classified, plot.window = c(1000, 2000))
```

`fixation_plot_ts` *Plot fixation classified vs. raw gaze coordinate time series*

Description

This function plots and returns a ggplot2 figure showing time series of gaze coordinates plotted against time. The interval to plot must be specified in the same unit as the vaiable specified in the input variable x.index.var. By default, this variable is called timestamp. Use this function to compare the raw gaze coordinates to the output from one or more fixation classification algorithms. Either the X or the Y coordinates are plotted. If the variable specified in the parameter algorithm.name is present in the input data AND contains more than one unique category, you will get subplots for each category. For example, if the input data contains fixation and raw coordinates for data from the I-VT and I2MC algorithms, each will be plotted in a separate subplot. Note that the X index variable (time stamps or sample number) must be the same.

Usage

```
fixation_plot_ts(
  data_in,
  plot.window = c(NA, NA),
  x.index.var = "timestamp",
  var1 = "x.raw",
  var2 = "x",
  xres = 1920,
  yres = 1080,
  verbose = TRUE,
  algorithm.name = "fixation.algorithm",
  ylim = NA,
  font.size = 15
)
```

Arguments

`data_in`	gaze matrix which must include columns for filtered and unfiltered data as specified in the var1 and var2 paramters
`plot.window`	vector defining the time window to plot. Correxponding values must be found in the variable specified in the parameter x.index.var (default: timestamp) in the data found in the variable 'timestamp'
`x.index.var`	Name of the variable to plot on the X axis, for example timestamp or sample index. If the default setting is used, gaze coordinates are plotted against timestamp (stored in a variable with that name). It can be replaced with any other numerical variable present in the data.
`var1`	Name of the first variable to plot. Default "x.raw"

var2	Name of the second variable to plot (overlayed on var1) Default: "x"
xres	Maximum of the X axis (horizontal resolution of the screen or area to plot on). Default 1920
yres	Maximum of the Y axis (vertical resolution of the screen or area to plot on). Default 1080
verbose	If TRUE, print the resulting plot
algorithm.name	Name of the fixation algorithm/or preprocessing procedure used for this data. The default is "fixation.algorithm" which is automatically stored in the output of kollaR fixation classification algorithm. It can be replaced by another variable name. If different categories exist, each will be plotted in a separate subplot.
ylim	Limits of y axis. If NA, the Y axis covers the range in the input data, otherwise it's restricted to the values in ylim. ylim = c(0, 1920) sets the minimum and maximum values on the Y acus to 0 and 1920 respectively
font.size	Font size

Value

a ggplot with gaze coordinates plotted on the y axis and sample number on the x axis

Examples

```
new.plot <- fixation_plot_ts (sample.data.classified, plot.window = c(1000, 2000))
```

---

idt_filter *Dispersion-based fixation detection algorithm* (I-DT)

Description

This function will be replaced by the function algorithm_idt in subsequent versions. The two functions take the same input arguments.idt_filter is a wrapper around idt_algorithm.

Apply a dispersion-based fixation (I-DT) detection algorithm to the eye tracking data. The algorithm identifies fixations as samples clustering within a spatial area. The procedure is described in Blignaut 2009 Input data must be a data frame with the variables timestamp, x.raw and y.raw as variables. Other variables can be included but will be ignored. This function does not perform pre-processing in the form of interpolation or smoothing. Use the function process.gaze for this. Timestamps are assumed to be in milliseconds. The output data is a list with two data frames: fixations includes all detected fixations with coordinates, duration and a number of other metrics, filt.gaze is a sample-by-sample data frame with time stamps, raw gaze coordinated and fixation coordinates. The function can be slow for long recordings and/or data recorded at high sampling rates.

Usage

```
idt_filter(
  gaze_raw,
  one_degree = 40,
  dispersion.threshold = 1,
  min.duration = 50,
  xcol = "x.raw",
  ycol = "y.raw",
  distance.threshold = 0.7,
  merge.ms.threshold = 75,
  missing.samples.threshold = 0.5
)
```

Arguments

gaze_raw	Data frame with gaze data before event detection. Include the variable timestamp with timing in ms and columns with raw x and y data as specified by the parameters xcol and ycol or their default values
one_degree	One degree of the visual field in the unit of the raw x and y coordinates. The unit is typically pixels or proportion of the screen. Make sure that the setting matches your data
dispersion.threshold	Maximum radius of fixation candidates. Samples clustering within a circle of this limit will be classified as a fixation if the duration is long enough.
min.duration	Minimum duration of fixations in milliseconds
xcol	Name of the column where raw x values are stored. Default: x.raw
ycol	Name of the column where raw y values are stored. Default: y.raw
distance.threshold	Subsequent fixations occurring withing this distance are merged. Set to 0 if you don't want to merge fixations.
merge.ms.threshold	Only subsequent fixations occurring within this time window are merged
missing.samples.threshold	Remove fixations with a higher proportion of missing samples. Range 0-1

Value

list including separate data frames for fixations and sample-by-sample data including gaze coordinates with and without fixation detection. The fixations data frame includes onset, offset, x, y, sample-to-sample root-mean-square deviations (RMSD, precision), RMSD from fixation centroid, and missing samples of each fixation.

Examples

```
idt_data <- idt_filter(sample.data.processed)
```

interpolate_with_margin *Interpolate over gaps (subsequent NAs) in vector.*

Description

Interpolate over gaps (subsequent NAs) in vector.

Usage

```
interpolate_with_margin(data_in, marg, max_gap)
```

Arguments

data_in	Vector to interpolate in
marg	Margin in samples before and after gap to use for interpolation
max_gap	Maximum length of gaps in sample

Value

vector with interpolated gaps

ivt_filter *I-VT algorithm for fixation and saccade detection*

Description

Apply an I-VT filter to the eye tracking data. This function is a wrapper around the function ivt_algorithm. It will be replaced by algorithm_ivt in future versions. The algorithm identifies saccades as periods with sample-to-sample velocity above a threshold and fixations as periods between saccades. See Salvucci and Goldberg 2000. Identifying fixations and saccades in eye tracking protocols. Proc. 2000 symposium on Eye tracking research and applications for a description.

Input data must be a data frame with the variables timestamp, x.raw and y.raw as variables. Other variables can be included but will be ignored. This function does not perform pre-processing in the form of interpolation or smoothing. Use the function process.gaze for this. Timestamps are assumed to be in milliseconds. The output data is a list with three data frames: fixations includes all detected fixations with coordinates, duration and a number of other metrics, saccades includes data for saccades, filt.gaze is a sample-by-sample data frame with time stamps, and gaze coordinates before ("raw") and after fixation detection. The function has a number of parameters for removing potentially invalid fixations and saccades. The parameter min.fixation.duration can be used to remove unlikely short fixations. If the parameter missing.samples threshold is set to a value lower than 1, fixations with a higher proportion of missing raw samples are removed.

Usage

```
ivt_filter(
  gaze_raw,
  velocity.filter.ms = 20,
  velocity.threshold = 35,
  min.saccade.duration = 10,
  min.saccade.amplitude = 1,
  min.fixation.duration = 40,
  one_degree = 40,
  save.velocity.profiles = FALSE,
  xcol = "x.raw",
  ycol = "y.raw",
  distance.threshold = 0.7,
  merge.ms.threshold = 75,
  missing.samples.threshold = 0.5,
  trim.fixations = FALSE,
  trim.dispersion.threshold = NA
)
```

Arguments

gaze_raw	Data frame with gaze data before fixation and saccade detection. Include the variable timestamp with timing in ms and columns with raw x and y data as specified by the parameters xcol and ycol or their default values
velocity.filter.ms	Window in milliseconds for moving average window used for smoothing the sample to sample velocity vector.
velocity.threshold	Velocity threshold for saccade detection in degrees/second
min.saccade.duration	Minimum duration of saccades in milliseconds
min.saccade.amplitude	Minimum amplitude of saccades in degrees
min.fixation.duration	Minimum duration of fixations in milliseconds
one_degree	One degree of the visual field in the unit of the raw x and y coordinates, typically pixels or proportion of the screen. Make sure that it is consistent with the format of your data
save.velocity.profiles	If TRUE, return velocity profiles of each detected saccade as a variable in the saccades data frame
xcol	Name of the column where raw x values are stored. Default: x.raw
ycol	Name of the column where raw y values are stored. Default: y.raw
distance.threshold	Subsequent fixations occurring withing this distance are merged. Set to 0 if you don't want to merge fixations.

`merge.ms.threshold`
: Subsequent fixations occuring within this time window and distance specified by distance.threshold are merged. Set to 0 if you don't want to merge fixations.

`missing.samples.threshold`
: Remove fixations with a higher proportion of missing samples. Range 0 to 1.

`trim.fixations`
: If TRUE, the onset of each fixation will be shifted forwards to the first non-missing (non-NA) sample during the period. The offset will be shifted backwards to the last non-missing sample. If TRUE, and the parameter trim.dispersion.threshold is a positive number, samples at the margins with large distances from the centroid will also be excluded

`trim.dispersion.threshold`
: If not NA and trim.fixations is TRUE, fixation onsets and offests will also be shrinked to exclude any samples at the margins with a larger

Value

list including separate data frames for fixations and sample-by-sample data including gaze coordinates before and after fixation detection. The fixations data frame gives onset, offset, x, y, sample-to-sample root-mean-square deviations (RMSD, precision), RMSD from fixation centroid, and missing samples of each fixation.

Examples

```
ivt_data <- ivt_filter(sample.data.processed, velocity.threshold = 30, min.fixation.duration = 40)
```

---

merge_adjacent_fixations

Merge adjacent fixations

Description

Merge fixations which appear close in space and time. This function is called by other functions and typically not used outside them

Usage

```
merge_adjacent_fixations(
  fixations,
  gaze_raw,
  distance.threshold = 0.5,
  ms.threshold = 75,
  one_degree = 40,
  xcol = "x.raw",
  ycol = "y.raw"
)
```

Arguments

fixations	Data frame with fixations
gaze_raw	Data matrix with raw data. See description of the algorithm_ivt function
distance.threshold	Subsequent fixations occurring withing this distance are merged. Set to 0 if you don't want to merge fixations.
ms.threshold	Maximum time elapsed between fixations to be merged.
one_degree	One degree of the visual field in the scale of the x and y coordinates. Typically pixels or proportion of the screen. Make sure the setting matches your data.
xcol	X coordinates in the raw gaze data matrix (gaze_raw)
ycol	Y coordinates in the raw gaze data matrix (gaze_raw)

Value

A new data frame with fixations

---

movmean.filter *Calculate the moving mean of a vector*

Description

Calculate the moving mean of a vector while ignoring NAs. Used internally in pre-processing functions

Usage

```
movmean.filter(x, k = 3)
```

Arguments

x	data to process
k	window length

Value

filtered data

plot_algorithm_results

Plot vdescriptives one or more fixation detection algorithms

Description

This function visualizes validity measures of fixations detected with one or more fixation detection algorithms. The function is tested for fixation data frames generated with kollaR event detection algorithms. By default, the function can plot Root Mean Square Deviations of subsequent samples within the detected fixations (precision), the RMSD from the fixation centroid, fixation duration and the proportion of missing raw samples. The output data is a ggplot which can be modified further outside the function. If you want to use this function to compare more than one fixation detection algorithms, combine them using the function rbind in base R. For example, `rbind(my_data1[["fixations"]], my_data2[["fixations"]])` would generate a combined data frame with the fixations detected by two event classification procedures.

Usage

```
plot_algorithm_results(data_in, plot.variable = "rmsd")
```

Arguments

`data_in`	Data frame with fixations to plot
`plot.variable`	Variable to plot. If left empty, RMSD of fixations are plotted. Alternatives are "rmsd", "duration", "missing.samples", "rms.from.center"

Value

A ggplot with visualizations of the selected validity measure

Examples

```
plot_algorithm_results(data_in = sample.data.fixations, plot.variable = "rmsd")
```

plot_filter_results *Plot descriptives from one or more fixation detection algorithms*

Description

This function visualizes validity measures of fixations detected with one or more fixation detection algorithms. The function is tested for fixation data frames generated with kollaR event detection algorithms. By default, the function can plot Root Mean Square Deviations of detected fixations, fixation duration and the proportion of missing raw samples. The output data is a ggplot which can be modified further outside the function. If you want to use this function to compare more

than one fixation detection algorithms, combine them using the function rbind in base R. For example, `rbind(my_data1[["fixations"]], my_data2[["fixations"]])` would generate a combined data frame with the fixations detected by two event classification procedures. This function will be replaced by 'plot_algorithm_results' in future versions.

Usage

```
plot_filter_results(data_in, plot.variable = "rmsd")
```

Arguments

`data_in`	Data frame with fixations to plot
`plot.variable`	Variable to plot. If left empty, RMSD of subsent samples (precision) are plotted. Alternatives are "rmsd", "rms.from.center", "duration", "missing.samples"

Value

A ggplot with visualizations of the selected validity measure

Examples

```
plot_filter_results(data_in = sample.data.fixations, plot.variable = "rmsd")
```

---

plot_sample_velocity *Plot the sample-to-sample velocity of eye tracking data.*

Description

This function visualizes the sample-to-sample velocity in a period of eye tracking data. This can be helpful when determining a suitable velocity threshold for saccade detection Input data must be a data frame with the variables timestamp, x.raw and y.raw as variables. Other variables can be included but will be ignored. This function does not perform pre-processing in the form of interpolation or smoothing. Use the function process.gaze for this. Timestamps are assumed to be in milliseconds. Default settings assume that x and y coordinates are in pixels. The output data is a plot of sample-to-sample velocity in the selected interval.

Usage

```
plot_sample_velocity(
  data_in,
  velocity.filter.ms = 20,
  plot.window = c(NA, NA),
  xcol = "x.raw",
  ycol = "y.raw",
  threshold.line = NA,
  one_degree = 40,
  font.size = 16,
  verbose = TRUE
)
```

Arguments

data_in	Data frame with gaze data to plot. Include the variable timestamp with timing in ms and columns with raw x and y data as specified by the paramerers xcol and ycol or their default values
velocity.filter.ms	Window in milliseconds for moving average window used for smoothing the sample-to-sample velocity vector.
plot.window	vector defining the time window to plot. If left empty, the 50-65 <0, they are assumed to be proportions, e.g., plot.window = c(0.3,0.35) plots the 30-35 percent of max.length interval of the data. Numbers >1 are assumed to refer to sample order in the data
xcol	Name of the column where raw x values are stored. Default: "x.raw"
ycol	Name of the column where raw y values are stored. Default: "y.raw"
threshold.line	Can be specified to add a line showing a potential velocity threshold for saccade detection. No threshold is shown if this parameter is NA
one_degree	One degree of the visual field in the unit of the raw x and y coordinates, typically pixels
font.size	Font size
verbose	If TRUE, print the resulting plot

Value

A ggplot showing the sample-to-sample velocity of the selected data interval

Examples

```
plot_sample_velocity(data_in = sample.data.processed, threshold.line = 35)
```

---

plot_velocity_profiles

Create ggplot of saccade velocity profiles

Description

This function plots and returns a ggplot showing velocity profiles of saccades plotted against time. Saccades should be generated with the ivt.filter functions with the save.velosity.profiles parameter set to TRUE. The interval to plot is defined by saccade number as they appear in the sacccades data frame.

Usage

```
plot_velocity_profiles(saccades, onset = NA, offset = NA, verbose = TRUE)
```

Arguments

saccades	data frame including saccades. Each saccade must have a list with a vector of the veloctity profiles
onset	first saccade to plot. The value must correspond to a number in the variable "number" in the saccades data frame. If left empty, all saccades are plotted
offset	last saccade to plot. The value must correspond to a number in the variable "number" in the saccades data frame.
verbose	If TRUE, print the resulting plot

Value

ggplot with velocity profiles

Examples

```
new.plot <- plot_velocity_profiles(sample.data.saccades, onset = 10, offset = 20)
```

---

preprocess_gaze *Interpolation and smoothing of gaze-vector*

Description

Pre-processing gaze Interpolate over gaps in data and smooth the x and y vectors using a moving average filter. The gaze vector must contain the variables timestamp, and variables containing unfiltered x and y coordinates. Default names: x.raw and y.raw. Timestamps are assumed to be in milliseconds. The unprocessed x and y variables are kept under the names x.unprocessed and y.unprocessed for comparison. The function will add the variable timestamp.t to the data frame before returning. This is a theoretical timestamp based on the detected median sample-to-sample timestamp difference as compared to the actual registered time stamps in the data. This can be useful in some validation analyses.

Usage

```
preprocess_gaze(
  gaze_raw,
  max_gap_ms = 75,
  marg_ms = 5,
  filter_ms = 15,
  xcol = "x.raw",
  ycol = "y.raw",
  na.ignore = TRUE
)
```

Arguments

gaze_raw	Data frame containing unfiltered timestamp, x.raw and y.raw vectors.
max_gap_ms	The maximum gaps defined as subsequent NAs in the data to interpolate over in milliseconds. Default 75 ms
marg_ms	The margin in milliseconds before and after the gap to use as basis for interpolation.
filter_ms	The size of the moving average window to use in smoothing. Default 15 ms
xcol	Name of column containing unprocessed x coordinates
ycol	Name of column containing unprocessed y coordinates
na.ignore	If TRUE, ignore NAs when filtering. If FALSE, the zoo function rollmean with the argument na.pad = TRUE will be used.

Value

data frame with gaze data after interpolation and filtering

Examples

```
processed_gaze <- preprocess_gaze(sample.data.unprocessed)
```

---

process_gaze *Interpolation and smoothing of gaze-vector. This function will be replaced by preprocess_gaze in future versions. process_gaze is a wrapper around preprocess gaze (the two functions produce the same result)*

Description

Pre-processing of gaze. Interpolate over gaps in data and smooth the x and y vectors using a moving average filter. The gaze vector must contain the variables timestamp, and variables containing unfiltered x and y coordinates. Default names: x.raw and y.raw. Timestamps are assumed to be in milliseconds. The unprocessed x and y variables are kept under the names x.unprocessed and y.unprocessed for comparison. The function will add the variable timestamp.t to the data frame before returning. This is a theoretical timestamp based on the detected median sample-to-sample timestamp difference as compared to the actual registered time stamps in the data. This can be useful in some validation analyses.

Usage

```
process_gaze(
  gaze_raw,
  max_gap_ms = 75,
  marg_ms = 5,
  filter_ms = 15,
  xcol = "x.raw",
  ycol = "y.raw"
)
```

Arguments

gaze_raw	Data frame containing unfiltered timestamp, x.raw and y.raw vectors.
max_gap_ms	The maximum gaps defined as subsequent NAs in the data to interpolate over in milliseconds. Default 75 ms
marg_ms	The margin in milliseconds before and after the gap to use as basis for interpolation.
filter_ms	The size of the moving average window to use in smoothing. Default 15 ms
xcol	Name of column containing unprocessed x coordinates
ycol	Name of column containing unprocessed y coordinates

Value

data frame with gaze data after interpolation and filtering

Examples

```
processed_gaze <- process_gaze(sample.data.unprocessed)
```

---

sample.data.classified

Sample-to-sample raw and fixation classified data from 1 individual

---

Description

This dataset contains sample-to-sample data from 1 individuals during a free viewing tasks. Data were recorded at 1200 Hz using a Tobii Pro Spectrum eye tracker. Fixations were classified with the I-VT algorithm with a velocity threshold set to 30 degrees/seconds and default settings in the function algorithm_ivt

Usage

```
sample.data.classified
```

Format

A data frame

x fixation position x

y fixation position y

x.raw fixation position x

y.raw fixation position y

timestamp timestamp in milliseconds

Source

The dataset was stored in the package at 'data/example_data.RData'

`sample.data.fixation1` *Fixations from 1 individual*

Description

This dataset contains fixation data from 1 individuals during a free viewing tasks. Data were recorded at 1200 Hz using a Tobii Pro Spectrum eye tracker

Usage

```
sample.data.fixation1
```

Format

A data frame

x fixation position x

y fixation position y

duration duration of fixation in milliseconds

onset onset of fixation in milliseconds

offset offset of fixation in milliseconds

rmsd Sample-to-sample root mean square deviation of all samples

rms.from.center Root means square deviation of all included samples from the centroid of the fixation

missing.samples proportion of missing samples

fixation.algorithm Name of the fixation filter algorithm

threshold Threshold setting for the fixation classification algorithm

id Participant id

Source

The dataset was stored in the package at 'data/example_data.RData'

`sample.data.fixations` *Fixations from 7 individuals*

Description

This dataset contains fixation data from 7 individuals during a free viewing tasks. Data were recorded at 1200 Hz using a Tobii Pro Spectrum eye tracker

Usage

```
sample.data.fixations
```

Format

A data frame

x fixation position x

y fixation position y

duration duration of fixation in milliseconds

onset onset of fixation in milliseconds

offset offset of fixation in milliseconds

rmsd Sample-to-sample root mean square deviation of all samples

rms.from.center Root means square deviation of all included samples from the centroid of the fixation

missing.samples proportion of missing samples

fixation.algorithm Name of the fixation filter algorithm

threshold Threshold setting for the fixation classification algorithm

id Participant id

Source

The dataset was stored in the package at 'data/example_data.RData'

sample.data.processed *Pre-processed sample-by-sample example data*

Description

This dataset contains data from 1 individuals during a free viewing tasks after pre-processing. Data were recorded at 1200 Hz using a Tobii Pro Spectrum eye tracker

Usage

```
sample.data.processed
```

Format

A data frame

id participant number

timestamp timestamp in ms recorded by the eye tracker

x.raw gaze position x

y.raw gaze position y

x.unprocessed copy of gaze position x before preprocessing

y.unprocessed copy of gaze position y before preprocessing

timestamp.t "'Theoretical timestamp' for comparison."

sample sample nr in recording

Source

The dataset was stored in the package at 'data/example_data.RData'

sample.data.saccades *Saccades from 3 individuals*

Description

This dataset contains saccade data from 3 individuals during a free viewing tasks. Data were recorded at 1200 Hz using a Tobii Pro Spectrum eye tracker

Usage

```
sample.data.saccades
```

Format

A data frame

onset onset of the saccade in ms

x.onset gaze position x at onset

y.onset gaze position y at onset

offset offset of the saccade in ms

x.offset gaze position x at offset

y.offset gaze position y at offset

duration duration of saccade in ms

amplitude amplitude of saccade in degrees

peak.velocity peak velocity of saccade

velocity.profile velocity profile

missing.samples proportion of missing samples

id participant number

Source

The dataset was stored in the package at 'data/example_data.RData'

---

sample.data.unprocessed

Unprocessed sample-by-sample example data

Description

This dataset contains data from 1 individual during a free viewing tasks before pre-processing. Data were recorded at 1200 Hz using a Tobii Pro Spectrum eye tracker

Usage

```
sample.data.unprocessed
```

Format

A data frame

id participant number

timestamp timestamp in ms recorded by the eye tracker

x.raw gaze position x

y.raw gaze position y

Source

The dataset was stored in the package at 'data/example_data.RData'

`static_plot` *Plot fixations in 2D space overlaied on a stimulus image*

Description

This function plots and returns a ggplot2 figure showing fixations on a background with one or multiple images, typically the stimuli. Data can represent one or multiple participants The interval to plot is defined by sample numbers. Fixations must have the variables x, y, and onset. The function is tested with .jpg-images. If paths to multiple images are given, all will be displayed. Fixations are shown on a white background if no background images are defined

Usage

```
static_plot(
  gazedata,
  xres = 1920,
  yres = 1080,
  plot.onset,
  plot.offset,
  background.images = NA,
  show.legend = TRUE,
  group.by = NA,
  gazepoint.size = 4,
  id_color_map = NA,
  connect.lines = TRUE,
  verbose = TRUE
)
```

Arguments

`gazedata`	Data frame with fixation data which must include columns for x and y coordinates as well as the variable onset which indicates the onset of the fixation. If the categorical or factor variable id is included, separate colors will represent each participant. Make sure the onset variables match the timing the plot.onset and plot.offset input.
`xres`	horizontal resolution of the screen or area to plot on. Default 1920
`yres`	vertical resolution of the screen or area to plot on. Default 1080
`plot.onset`	Onset of the interval in the gaze_data$onset variable to plot in the same unit, typically milliseconds
`plot.offset`	Offset of the interval in the gaze_data$onset variable to plot in the same unit, typically milliseconds
`background.images`	data frame with background images to use as background. The data frame must include the variables min.x, min.y, max.x, and max.y variables representing where the images should be placed on the background and the variable path

	specifying a full file path. #Example: `background.images <- data.frame( path = "my_image.jpg", min.x = 1, min.y = 1, max.x = 200. max.y = 200)`
`show.legend`	If TRUE, show values in "id" in legend
`group.by`	If not NA, plot each level in the variable in a separate panel. For example group.by = "group" returns a separate panel for each group and group.by ="id" returns a separate panel for each id.
`gazepoint.size`	Size of the circle illustrating the point of gaze
`id_color_map`	A ggplot color map specifying a color to plot for each id. ids should match the variable id'in the gazedata matrix. Set to NA to assign values automatically.
`connect.lines`	If TRUE, gaze coordinates are connected with lines
`verbose`	If TRUE, the resulting figure is displayed automatically

Value

a ggplot of raw and fixated values plotted on the y axis and sample number on the x axis

---

`suggest_threshold` *Data-driven identification of threshold parameters for adaptive velocity-based saccade detection.*

Description

The function is based on a procedure suggested by Nyström and Holmqvist 2010. Behavior Research Methods, 42, 188-204. The function can be used to identify specific thresholds for saccade onset for individuals and/or segments of the data, as an alternative to using the same thresholds for each participants. It is used in kollaR by the function 'algorithm_adaptive'

Peak velocity and saccade amplitude are typically highly postively correlated. It is therefore important to consider that differences in gaze behavior between individuals and/or data segment may lead to differences in proposed saccade onset velocity threshold. #' The input data should be preprocessed (e.g., noise removal and interpolation over gaps) The output is a list with three cells: "peak.threshold" and "onset.threshold" are parameters used by the function algorithm_adaptive (see Nyström and Holmqvist 2010 for details). "velocity" is a data frame with sample-to-sample velocity in the unit specified by the parameter one_degree

Usage

```
suggest_threshold(
  gaze,
  velocity.filter.ms = 10,
  one_degree = 40,
  ycol = "y.raw",
  xcol = "x.raw",
  peak.threshold.start = 130,
  onset.threshold.sd = 3,
  min.period.ms = 40,
  margin.ms = 3
)
```

Arguments

gaze	Data frame with gaze data before saccade and fixation data identification. The data frame must include the variable timestamp with timing in milliseconds and columns for x and y coordinates specified by the columns 'xcol' and 'ycol' respectively.
velocity.filter.ms	If velocity.filter.ms is not NA, the velocity vector is smoothed using a moving median filter corresponding to this value in ms before the propose threshold is identified. Default: 10.
one_degree	one degree of the visual field in the unit of the x and y coordinates in the data. Typically pixels or degrees.
ycol	column in the gaze data frame where y coordinates are found. Default: y.raw
xcol	column in the gaze data frame where x coordinates are found. Default: x.raw
peak.threshold.start	initial peak threshold value in degrees of the visual field. Default: 200
onset.threshold.sd	sd of sample-by-sample velocities used to select the proposed velocity threshold (proposed.velocity.threshold)
min.period.ms	Update the peak velocity thresholds iteratively based on data within consecuitive runs of samples below the previous thresholds. Should be approximately minimum fixation duration.
margin.ms	A margin around min.period.ms. This reduces the risk that samples included in the threshold estimation belong o a saccade

Value

list including separate data frames for proposed saccade onset threshold, peak threshold, and sample-to-sample velocity

---

summarize_fixation_metrics

Summarize fixation statistics

Description

Summarize descriptives for a fixation defined by onset and offset rows in the data. Used internally by event classification functions.

Usage

```
summarize_fixation_metrics(
  fixation.candidate.starts,
  fixation.candidate.stops,
  x,
```

```
  y,
  timestamp,
  one_degree = 40
)
```

Arguments

fixation.candidate.starts	First row in the data included in the fixation
fixation.candidate.stops	Last row in the data included in the fixation
x	X coordinates
y	Y coordinates
timestamp	Timestamps in milliseconds
one_degree	one degree of the visual field in the unit of the x and y coordinates in the data. Typically pixels or degrees.

Value

data frame with fixation descriptives

---

trim_fixations *Adjust the onset and offset of fixations to avoid misclassification of saccade samples as belonging to fixations*

Description

Adjust the onset and offset of all fixations in a data frame (The function adjust_fixation_timing does this for a single fixation).

Shrink the period classified as a fixation by removing samples at the onset and offset with excessive differences from the fixation center or which are missing (X or Y are NA). This reduces the risk that samples belonging to saccades are misclassified as belonging to a fixation. Please note that this procedure is included by default in the event classification algorithm 'alogorithm_i2mc' (see documentation for this function for details)

The procedure starts by calculating the median (MD) and MAD of the absolute distances from the fixation center of all included samples. The fixation onset is shifted forwards to the first sample with a distance to the fixation center under t* MAD + MD where t is specified by the input parameter threshold. Analogously, fixation offset is shifted backwards to the last included sample with distance to the fixation center under t* MAD + MD

trim_fixations will look for variables called 'fixation.algorithm' and 'threshold' in the data frame 'fixations'. These columns are produced by kollaR event classification algorithms. If they are found, they will be transferred to the output data frame.

Usage

```
trim_fixations(
  fixations,
  gaze,
  xcol = "x.raw",
  ycol = "y.raw",
  threshold = 3,
  one_degree = 40
)
```

Arguments

fixations	Data frame with fixations to trim. The data frame must include the variables 'firstline' (index of first row in the sample-by-sample data belonging to each fixation), 'lastline' (index of last row in the sample-by-sample data belonging to each fixation). The function works with the fixation output from kollaR event classification algorithms the fixation)
gaze	Data frame with sample-to-sample data. Must include timestamps in milliseconds specified by the variable timestamp, and X and Y coordinates specified by the parameters 'xcol' and 'ycol'.
xcol	Variable in the sample-to-sample data frame where X coordinates (before event classification) are found
ycol	Variable in the sample-to-sample data frame where X coordinates (before event classification) are found
threshold	Threshold for highest accepted distance from fixation center in MADs from the median. Default 3. If NA, just remove NAs at the onset and offest of fixation but ignore deviations from fixation center
one_degree	One degree of the visual field in the units of the X and Y coordinates (which is typically pixels or degrees of the visual field)

Value

data frame with fixations after adjustment of onset and offset

Index