Package: glasses3 (via r-universe)

November 17, 2026

Type Package

Title Read and Plot 'Tobii Pro Glasses 3' Recordings

Version 0.1.0

Description Reads recordings made with the 'Tobii Pro Glasses 3' wearable eye tracker directly from the folder on the SD card, so 'Tobii Pro Lab' is not needed. Gaze, pupil diameter, inertial measurement unit and sync port data are read from the compressed JSON files into tibbles, along with the recording metadata. Samples can be aligned with sync port pulses and cut into epochs. Further functions fill gaps in the time series, average samples into time bins and convert gaze directions to angles. Pupil diameter and gaze position can be plotted.

License MIT + file LICENSE

URL https://github.com/mcanzi/glasses3, https://mcanzi.de/glasses3/

BugReports https://github.com/mcanzi/glasses3/issues

Encoding UTF-8

Language en-GB

Depends R (>= 4.1.0)

Imports dplyr (>= 1.1.0), ggplot2 (>= 3.4.0), jsonlite, purrr (>= 1.0.0), stats, stringr (>= 1.5.0), tibble, tidyr (>= 1.3.0), utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Config/pak/sysreqs libicu-dev

Repository https://mcanzi.r-universe.dev

Date/Publication 2026-11-17 08:12:50 UTC

RemoteUrl https://github.com/mcanzi/glasses3

RemoteRef HEAD

RemoteSha ac6caeff5feb040d74d2227dbf5e71ba9d5b9372

Contents

g3_add_syncport . . . 2
g3_downsample . . . 3
g3_epochs . . . 4
g3_example . . . 5
g3_fill_gaps . . . 5
g3_gaze_angles . . . 6
g3_pupil_long . . . 7
g3_syncport . . . 8
list_g3_recordings . . . 8
plot_g3_gaze . . . 9
plot_g3_pupil . . . 10
read_g3 . . . 11
read_g3_events . . . 12
read_g3_gaze . . . 13
read_g3_imu . . . 14
read_g3_info . . . 15
theme_g3 . . . 16

Index **17**

`g3_add_syncport` *Add sync port information to samples*

Description

Adds the sync port state and the most recent pulse to each sample of gaze or IMU data. With several recordings, samples are matched to the pulses of their own recording through the `recording` column.

Usage

```
g3_add_syncport(x, events, direction = c("out", "in"))
```

Arguments

`x`	A data frame with a `time` column, such as the output of `read_g3_gaze()` or `read_g3_imu()`.
`events`	Event data from `read_g3_events()`.
`direction`	`"out"` for the signal sent by the glasses, `"in"` for the signal received from external equipment.

Value

x with three added columns: syncport (1 while the signal is high, 0 otherwise), pulse (number of the most recent pulse onset, NA before the first one) and pulse_time (seconds since that onset). If x is a grouped tibble, the result has the same groups.

Examples

```
rec <- read_g3(g3_example())
gaze <- g3_add_syncport(rec$gaze, rec$events)
gaze |>
  dplyr::filter(syncport == 1) |>
  dplyr::select(time, pulse, pulse_time)
```

---

g3_downsample *Downsample to fixed time bins*

Description

Averages samples within bins of fixed width, for example to reduce 50 Hz data to 100 ms bins before fitting a generalised additive mixed model. Fewer samples make models faster to fit and reduce the autocorrelation the model has to account for.

Usage

```
g3_downsample(x, bin = 0.1, min_valid = 0.5, by = NULL)
```

Arguments

x	Gaze data from read_g3_gaze(), epochs from g3_epochs() or long pupil data from g3_pupil_long().
bin	Bin width in seconds.
min_valid	Proportion of samples in a bin that must have a value. A numeric column is NA in bins where fewer of its samples are available, so that bins mostly lost to a blink do not get a value from the few samples left.
by	Names of further columns to group by, such as a participant or condition column. They are kept in the result.

Details

Bins are counted from time 0, or from the pulse onset for epochs from g3_epochs(), so that the onset falls on a bin boundary. Each bin is labelled with its start time. Rows are grouped by the recording, pulse and eye columns where these exist, so the function works on the output of read_g3_gaze(), g3_epochs() and g3_pupil_long(). Columns named in by are added to these groups, and so are the groups of a grouped tibble.

Value

A tibble with one row per bin and group, starting with the grouping columns. Other numeric columns hold the mean of the available samples, and logical columns, such as `valid`, the proportion of `TRUE` samples. `samples` gives the number of samples in the bin. All other columns are dropped. A grouped tibble keeps its groups.

Examples

```
gaze <- read_g3_gaze(g3_example())
g3_downsample(gaze, bin = 0.1)

rec <- read_g3(g3_example())
ep <- g3_epochs(rec$gaze, rec$events, before = 0.5, after = 3)
g3_downsample(g3_pupil_long(ep), bin = 0.1)

gaze |>
  dplyr::mutate(participant = "P01") |>
  g3_downsample(bin = 0.1, by = "participant")
```

`g3_epochs` *Cut samples into epochs around sync port pulses*

Description

Returns the samples from `before` seconds before to `after` seconds after the onset of each sync port pulse, for example to analyse pupil responses to stimuli whose onset was marked with a sync pulse. With several recordings, each recording is cut at its own pulses.

Usage

```
g3_epochs(
  x,
  events,
  before = 0,
  after = 1,
  direction = c("out", "in"),
  pulses = NULL
)
```

Arguments

`x`	A data frame with a `time` column, such as the output of `read_g3_gaze()` or `read_g3_imu()`.
`events`	Event data from `read_g3_events()`.
`before, after`	Seconds to keep before and after each onset.
`direction`	`"out"` for the signal sent by the glasses, `"in"` for the signal received from external equipment.
`pulses`	Which pulses to use, as pulse numbers. Defaults to all.

Value

The selected rows of `x`, sorted by time within each epoch, with added columns `pulse` and `epoch_time` (seconds relative to the pulse onset). Samples can appear in more than one epoch if epochs overlap. If `x` is a grouped tibble, the result has the same groups.

Examples

```
rec <- read_g3(g3_example())
ep <- g3_epochs(rec$gaze, rec$events, before = 1, after = 5)
dplyr::count(ep, pulse)
```

---

g3_example *Path to the example recording*

Description

The package includes the first 30 seconds of a Glasses 3 recording at 50 Hz. The scene video is left out, and the participant name, serial numbers, recording ID and start time are replaced with placeholders.

Usage

```
g3_example()
```

Value

Path to the example recording folder.

Examples

```
g3_example()
```

---

g3_fill_gaps *Insert rows for missing samples*

Description

The glasses write a line even for samples where they detect no gaze, so the gaze data are usually complete. Samples can still be missing, for example after damaged lines were skipped. `g3_fill_gaps()` inserts a row for each missing sample, so that the data are evenly spaced again, as many preprocessing methods assume. The new rows hold `NA`, and their validity columns are `FALSE`. Values are not interpolated.

Usage

```
g3_fill_gaps(x, rate = NULL)
```

Arguments

x	Gaze data from read_g3_gaze().
rate	Sampling rate in Hz. By default it is worked out from the median interval between samples and set to 50 or 100 Hz when it is within 10% of either.

Details

With several recordings in x, each is filled separately. The recording column and columns you added yourself, such as a participant column, are copied into the new rows when they have a single value within the recording.

Value

x with the inserted rows, sorted by time, and a logical column inserted that is TRUE for the new rows. If x is a grouped tibble, the result has the same groups.

Examples

```
gaze <- read_g3_gaze(g3_example())
gappy <- dplyr::slice(gaze, -(100:109))
filled <- g3_fill_gaps(gappy)
nrow(gappy)
nrow(filled)
filled |>
  dplyr::filter(inserted) |>
  dplyr::select(time, left_pupil, valid)
```

---

g3_gaze_angles *Gaze angles and vergence*

Description

Converts the gaze direction of each eye to horizontal and vertical angles and adds the vergence angle between the two eyes.

Usage

```
g3_gaze_angles(gaze)
```

Arguments

gaze	Gaze data from read_g3_gaze().

Details

The glasses report gaze direction as a unit vector in a coordinate system centred on the scene camera, where x points to the wearer's left, y up and z forward (Tobii Pro Glasses 3 Developer Guide). Angles are relative to the z axis, so (0, 0) is straight ahead of the glasses.

Value

gaze with these columns added, in degrees:

left_azimuth, right_azimuth Horizontal angle, positive to the wearer's right.

left_elevation, right_elevation Vertical angle, positive up.

vergence Angle between the left and right gaze directions. Larger values mean the eyes converge on a closer point.

Samples without a direction for an eye get NA. The result is a tibble, and if gaze is a grouped tibble, it has the same groups.

Examples

```
gaze <- g3_gaze_angles(read_g3_gaze(g3_example()))
gaze |>
  dplyr::select(left_azimuth, left_elevation, vergence) |>
  summary()
```

---

g3_pupil_long *Pupil data in long format*

Description

Reshapes the pupil columns of gaze data to one row per sample and eye, which suits plotting and mixed-effects models.

Usage

```
g3_pupil_long(gaze)
```

Arguments

gaze	Gaze data from read_g3_gaze(), possibly with added columns.

Value

A tibble with the non-eye columns of gaze (such as recording, time, pulse or epoch_time, if present), plus eye ("left" or "right") and pupil. Groups of a grouped tibble are kept if their columns are still in the result.

Examples

```
gaze <- read_g3_gaze(g3_example())
g3_pupil_long(gaze)
```

g3_syncport *Extract sync port pulses*

Description

Turns the sync port signal changes in the event data into one row per pulse, where a pulse runs from the signal going high (1) to it going low (0) again. With several recordings, pulses are numbered within each recording.

Usage

```
g3_syncport(events, direction = c("out", "in"))
```

Arguments

events	Event data from read_g3_events().
direction	"out" for the signal sent by the glasses, "in" for the signal received from external equipment.

Value

A tibble with columns recording (if events has one), pulse (running number), onset, offset and duration (seconds). offset is NA if the recording ended while the signal was high. Groups of a grouped tibble are kept if their columns are still in the result.

Examples

```
events <- read_g3_events(g3_example())
g3_syncport(events)
```

list_g3_recordings *Find recordings in a folder*

Description

Searches a folder, such as the root of the Glasses 3 SD card, for recording folders.

Usage

```
list_g3_recordings(path, recursive = TRUE)
```

Arguments

path	Folder to search.
recursive	Search subfolders.

Value

A tibble with one row per recording: `recording` (the folder name, as used in the `recording` column by the readers), `path`, `name`, `participant`, `created` and `duration`. Pass it to `read_g3_gaze()`, `read_g3_imu()` or `read_g3_events()` to read all recordings at once.

Examples

```
recordings <- list_g3_recordings(dirname(g3_example()))
recordings
read_g3_gaze(recordings)
```

---

plot_g3_gaze *Plot gaze position*

Description

Plots the 2D gaze point in scene camera coordinates, with the top of the video frame at the top of the plot. The frame shows the edges of the scene video, and gaze points outside it are left out. With several recordings in `x`, each gets its own panel.

Usage

```
plot_g3_gaze(
  x,
  type = c("points", "density"),
  from = NULL,
  to = NULL,
  bins = 40
)
```

Arguments

`x`	Gaze data from `read_g3_gaze()` or `g3_epochs()`, or a `g3_recording` from `read_g3()`.
`type`	`"points"` for individual samples, shaded from light (early) to dark (late), or `"density"` for a 2D histogram.
`from, to`	Time window in seconds. Defaults to all the data. For epochs, the window is in seconds from the pulse onset.
`bins`	Number of bins along the horizontal axis when `type = "density"`. The vertical axis gets 9/16 as many, so that the bins are square in the video.

Value

A ggplot object.

Examples

```
rec <- read_g3(g3_example())
plot_g3_gaze(rec)
plot_g3_gaze(rec, type = "density")
```

`plot_g3_pupil` *Plot pupil diameter over time*

Description

Draws pupil diameter for each eye over time, with sync port pulses as shaded areas. With several recordings in `x`, each gets its own panel, with its own time axis.

Usage

```
plot_g3_pupil(
  x,
  events = NULL,
  eyes = c("left", "right"),
  from = NULL,
  to = NULL,
  direction = c("out", "in"),
  smooth = NULL
)

## S3 method for class 'g3_recording'
plot(x, ...)
```

Arguments

`x`	Gaze data from `read_g3_gaze()` or `g3_epochs()`, or a `g3_recording` from `read_g3()`.
`events`	Optional event data from `read_g3_events()`. Sync port pulses are drawn as shaded areas. Taken from `x` if `x` is a `g3_recording`. Not used for epochs.
`eyes`	Which eyes to plot.
`from, to`	Time window in seconds. Defaults to all the data. For epochs, the window is in seconds from the pulse onset.
`direction`	Sync port direction to show; see `g3_syncport()`.
`smooth`	Width in seconds of a running median drawn over the raw trace, or `NULL` (the default) for none. Missing samples are ignored, and the line is left out where more than half of the window is missing. For epochs, each pulse is smoothed before the mean is taken. Only the plot is smoothed.
`...`	Passed to `plot_g3_pupil()`.

Details

If x comes from g3_epochs(), the time axis is the time from the pulse onset instead. Each eye gets its own panel, with a thin line for every pulse and a thick line for the mean across pulses.

The eye colours and the theme are described in theme_g3().

Value

A ggplot object. The plot() method for recordings from read_g3() draws this plot and returns the recording invisibly.

Examples

```
rec <- read_g3(g3_example())
plot_g3_pupil(rec)
plot_g3_pupil(rec$gaze, eyes = "left", from = 5, to = 15)
plot_g3_pupil(rec, smooth = 1)

epochs <- g3_epochs(rec$gaze, rec$events, before = 0.2, after = 1)
plot_g3_pupil(epochs)
```

---

read_g3 *Read a Tobii Pro Glasses 3 recording*

Description

Reads a recording folder as written to the SD card by the Tobii Pro Glasses 3. The folder contains recording.g3, gazedata.gz, imudata.gz, eventdata.gz and a meta folder. The scene video is not needed.

Usage

```
read_g3(path, data = c("gaze", "imu", "events"))
```

Arguments

path	Path to a recording folder.
data	Which data streams to read. Any of "gaze", "imu" and "events".

Details

read_g3() reads one recording. To read several recordings into one data frame, give their paths to read_g3_gaze(), read_g3_imu() or read_g3_events().

Value

An object of class g3_recording: a list with elements info (see read_g3_info()), gaze (see read_g3_gaze()), imu (see read_g3_imu()) and events (see read_g3_events()). Streams not requested are NULL.

See Also

`list_g3_recordings()` to find recordings on an SD card.

Examples

```
path <- g3_example()
rec <- read_g3(path)
rec
summary(rec)
rec$gaze
plot(rec)
```

`read_g3_events` *Read event data*

Description

Parses `eventdata.gz`, which holds sync port signals and any events sent to the glasses through the API during the recording. Lines that cannot be parsed are skipped with a warning.

Usage

```
read_g3_events(path)
```

Arguments

`path`	Path to a recording folder or directly to an `eventdata.gz` file. Several paths, or the output of `list_g3_recordings()`, read several recordings into one tibble.

Value

A tibble with columns `recording` (see `read_g3_gaze()`), `time` (seconds since the start of the recording), `type` (for example `"syncport"`), `direction` and `value` (for sync port events; `NA` otherwise), and `data`, the event payload as a JSON string, so that other event types are kept unchanged.

Examples

```
read_g3_events(g3_example())
```

`read_g3_gaze` *Read gaze and pupil data*

Description

Parses `gazedata.gz` into one row per sample. Samples where the glasses detected no gaze are kept, with `NA` values, so that the time series stays complete. Lines that cannot be parsed, such as the cut-off last line of an incomplete file, are skipped with a warning; see `g3_fill_gaps()` to replace them with empty rows. A file whose compressed data are damaged cannot be read and gives an error.

Usage

```
read_g3_gaze(path)
```

Arguments

`path`	Path to a recording folder or directly to a `gazedata.gz` file. Several paths, or the output of `list_g3_recordings()`, read several recordings into one tibble.

Value

A tibble with columns:

recording Name of the recording folder. If two recordings in `path` have folders of the same name, the full path is used instead.

time Seconds since the start of the recording.

gaze2d_x, gaze2d_y Gaze point in normalised scene camera coordinates, with (0, 0) at the top left of the video frame and (1, 1) at the bottom right.

gaze3d_x, gaze3d_y, gaze3d_z Point where the gaze of the two eyes meets, in millimetres from the scene camera.

left_origin_x, ..., right_direction_z Position of each eye in millimetres from the scene camera, and its gaze direction as a unit vector.

left_pupil, right_pupil Pupil diameter for each eye in millimetres, the unit given in the Tobii Pro Glasses 3 Developer Guide.

valid_left, valid_right Whether the glasses reported data for that eye.

valid Whether the sample has a gaze point. The glasses can estimate one from a single eye, so this can be `TRUE` when one eye is invalid.

See the Tobii Pro Glasses 3 Developer Guide for the coordinate systems.

The glasses record gaze at 50 or 100 Hz, in the same file format. The rate is not stored in the recording, so functions that need it work it out from the timestamps.

Examples

```
gaze <- read_g3_gaze(g3_example())
gaze

recordings <- list_g3_recordings(dirname(g3_example()))
read_g3_gaze(recordings)
```

`read_g3_imu` *Read inertial measurement unit data*

Description

Parses `imudata.gz`. Accelerometer and gyroscope samples and magnetometer samples are stored on separate lines at different rates, so each row has `NA` for the sensors not sampled at that time. Lines that cannot be parsed are skipped with a warning.

Usage

```
read_g3_imu(path)
```

Arguments

`path`	Path to a recording folder or directly to an `imudata.gz` file. Several paths, or the output of `list_g3_recordings()`, read several recordings into one tibble.

Value

A tibble with columns `recording` (see `read_g3_gaze()`), `time` (seconds since the start of the recording), `accel_x`, `accel_y`, `accel_z` (accelerometer), `gyro_x`, `gyro_y`, `gyro_z` (gyroscope) and `mag_x`, `mag_y`, `mag_z` (magnetometer). Following the Tobii Pro Glasses 3 Developer Guide, acceleration is in metres per second squared, rotation in degrees per second and the magnetic field in microtesla.

Examples

```
read_g3_imu(g3_example())
```

`read_g3_info` *Read recording metadata*

Description

Reads `recording.g3` and the files in the `meta` folder.

Usage

```
read_g3_info(path)
```

Arguments

`path`	Path to a recording folder.

Value

A list of class `g3_info` with these elements:

- `name` and `uuid` of the recording
- `created`, the start time (POSIXct, UTC), and `timezone`
- `duration` in seconds
- `gaze_samples` and `gaze_valid_samples`, as counted by the glasses
- `participant`, as entered when recording
- `firmware` version
- `recording_unit` and `head_unit` serial numbers
- `files`, the names of the data files
- `scene_camera`, the camera calibration
- `meta`, all files in the `meta` folder as character strings

Examples

```
read_g3_info(g3_example())
```

theme_g3 *Theme for glasses3 plots*

Description

The ggplot2 theme used by all plots in the package, for giving your own plots the same look. It builds on `ggplot2::theme_minimal()` with light grey gridlines, grey axis text, the legend at the top left and bold left-aligned panel titles on a white background.

Usage

```
theme_g3(base_size = 11)
```

Arguments

`base_size`	Base font size in points.

Details

In the package plots, the left eye is drawn in slate blue (`#2A568C`) and the right eye in Indian red (`#EE6363`, R's `indianred2`). The pair passes checks for lightness, chroma, contrast against white and separation under simulated protanopia, deuteranopia and tritanopia. Means use darker steps of the same hues (`#1D4371` and `indianred3`).

Value

A ggplot2 theme.

Examples

```
library(ggplot2)
gaze <- read_g3_gaze(g3_example())
ggplot(gaze, aes(time, left_pupil)) +
  geom_line(colour = "#2A568C", na.rm = TRUE) +
  theme_g3()
```

Index

g3_add_syncport, 2
g3_downsample, 3
g3_epochs, 4
g3_epochs(), *3*, *9–11*
g3_example, 5
g3_fill_gaps, 5
g3_fill_gaps(), *13*
g3_gaze_angles, 6
g3_pupil_long, 7
g3_pupil_long(), *3*
g3_syncport, 8
g3_syncport(), *10*
ggplot2::theme_minimal(), *16*

list_g3_recordings, 8
list_g3_recordings(), *12–14*

plot.g3_recording *(*plot_g3_pupil*)*, 10
plot_g3_gaze, 9
plot_g3_pupil, 10
plot_g3_pupil(), *10*

read_g3, 11
read_g3(), *9–11*
read_g3_events, 12
read_g3_events(), *2*, *4*, *8–11*
read_g3_gaze, 13
read_g3_gaze(), *2–4*, *6, 7*, *9–12*, *14*
read_g3_imu, 14
read_g3_imu(), *2*, *4*, *9*, *11*
read_g3_info, 15
read_g3_info(), *11*

theme_g3, 16
theme_g3(), *11*