

Is the low-cost EyeTribe eye tracker any good for research?

Edwin S. Dalmaijer^a

^a *Department of Experimental Psychology, University of Oxford, Oxford, United Kingdom*

Correspondence

Correspondence concerning this article should be addressed to Edwin Dalmaijer, Dept. of Experimental Psychology, University of Oxford, Tinbergen Building, 9 South Parks Road, Oxford, OX1 3PS, United Kingdom. Email: edwin.dalmaijer@psy.ox.ac.uk

Pre-print note

This manuscript has been pre-published, to speed up scientific communications. This allows researchers to have access to the results before formal publication. However, it does mean this manuscript has not been peer-reviewed yet. The manuscript has been submitted for publication to a peer-reviewed venue.

Acknowledgements

The author is not in any way affiliated or sponsored by the EyeTribe or any other commercial party. This research was supported by a European Union FP7 Marie Curie ITN grant (606901). The author would like to thank Ignace Hooge, for his thorough review of an earlier version of this manuscript, and Benjamin de Haas, for his testing of the EyeTribe Toolbox for Matlab.

Abstract

Eye-tracking technology is becoming increasingly cheaper, both on the hardware and on the software front. Currently, the EyeTribe tracker is the most inexpensive commercial eye tracker in the world, at a price of \$99. The low costs make it a potentially interesting resource for research, but no objective testing of its quality has been performed yet. Here the EyeTribe tracker is compared with an EyeLink 1000, a high-quality video eye tracker. Results indicate that the spatial precision and accuracy are good enough for fixation checking, point-of-regard analyses, and pupilometry. However, the low sampling rate renders the device unsuitable for testing high-accuracy saccade metrics. Additionally, open-source toolboxes for Matlab and Python, and a plug-in for OpenSesame are presented, which can be used to interface with the EyeTribe tracker.

Keywords: eye tracking, pupilometry, fixations, saccades, open source

1. Introduction

Technology is becoming increasingly inexpensive and integrated in our daily lives. This trend is becoming apparent in eye tracking, a technique that tracks where individuals are gazing on their computer monitor or tablet. Eye tracking can be employed to allow users to navigate through user interfaces without having to use their hands, to play games in novel ways, and in assistive software for people with movement disabilities (e.g. to type by gaze). Recently, several high-profile technology companies have started to employ the technique in their (mobile) devices, e.g. in Samsung Galaxy smartphones and Google Glass. Furthermore, established and new manufactures of eye trackers have started to produce low-cost devices, e.g. GazePoint's GP3 tracker (\$495¹), Tobii's EyeX tracker (\$139), and the EyeTribe tracker (\$99).

Concurrent with this increase in availability of low-cost eye trackers, there is an emerging trend to publish software for (scientific) eye tracking on an open-source basis, making it freely available to all. Examples range from openEyes (Li, Babcock, & Parkhurst, 2006), a complete head-mounted device with tracking software; via ITU's Gaze Tracker (San Agustin et al., 2010; San Agustin, Skovsgaard, Hansen, & Hansen, 2009) and GazeParser (Sogo, 2013), packages designed to analyse (web)camera footage of eyes; to PsychoPy (Peirce, 2007, 2009), OpenSesame (Mathôt, Schreij, & Theeuwes, 2012) and PyGaze (Dalmaijer, Mathôt, & Van der Stigchel, 2013), stimulus generators that support existing (mostly commercial) eye trackers.

In the current article, these trends converge in new open-source software for the inexpensive EyeTribe tracker, which is tested for its potential for research. The aim is to explore in what kind of research the EyeTribe can be employed, and to provide researchers with tools to do so.

1 Listed prices are excluding VAT and shipping expenses, except from GazePoint's GP3 for which taxes but not shipping costs are included.

2. Methods

2.1 Apparatus and Participants

Data was collected by sequentially tracking eye movements in a group of five participants (including the author), who provided written informed consent before participating. In two consecutive sessions, an EyeTribe tracker collected data at 60 Hz (binocular), and an EyeLink 1000 at 1000 Hz (monocular), with participants' heads placed on a chin rest. With the exception of a forehead rest, which was only employed when using the EyeLink 1000, the setups were identical for both testing sessions.

Concurrent tracking by both devices of the same eye movements proved to be impossible, due to the mutually exclusive way in which both devices work. The EyeLink 1000 expects a single source of infrared light, whereas the EyeTribe tracker requires two. Using both devices at the same time results in deterioration of results of both, and often to a failure to calibrate at least one.

The EyeLink 1000 system was selected as a comparison for its high spatial and temporal accuracy, which make it one of the best video eye trackers that are currently available. EyeLinks have been the work horses in a lot of studies that require high-accuracy measurements of saccade characteristics, such as microsaccades (Laubrock, Kliegl, Rolfs, & Engbert, 2010; McCamy, Najafian Jazi, Otero-Millan, Macknik, & Martinez-Conde, 2013), fixation stability (Bethlehem et al., 2014), saccade trajectories (Van der Stigchel, Mulckhuyse, & Theeuwes, 2009), and subtle changes in pupil size (Mathôt, Dalmaijer, Grainger, & Van der Stigchel, in press).

2.2 Procedure

Both of the eye trackers were controlled by the same PyGaze script running on a Clevo W150ER (Intel Core i7-3610QM @ 2.30 GHz, 16 GB, Windows 7), using a ViewSonic P227f CRT monitor (21 inch, 60 Hz, 1024x768 pixels, 36.24x27.58 degrees of visual angle) that was placed 62 cm from the eyes of a participant. Calibrations were performed using nine-point grids for both trackers. Results were compared in custom offline analyses scripted in Python, using open-source libraries NumPy and SciPy (Oliphant, 2007), and Matplotlib (Hunter, 2007). All of the experimentation and analysis software is available to download and review on GitHub: github.com/esdalmaijer/EyeTribe_test.

2.3 Experiment 1: Precision and Accuracy Validation

An eye tracker's performance is usually assessed using two metrics: accuracy and precision. Accuracy (or systematic error) reflects the eye tracker's ability to measure the point of regard, and is defined as the average difference between a test stimulus position and the measured gaze position (Holmqvist, Nyström, & Mulvey, 2012). Precision refers to the eye tracker's ability to produce consistent measurements, and is often estimated by calculating the Root Mean Square noise (Holmqvist et al., 2012).

To establish these measures, gaze position was recorded during two-second periods of fixating a target stimulus. The targets appeared consecutively, with an inter-trial interval of one second, on locations that were different from the calibration grid. The target grid spanned 25.81 degrees of visual angle horizontally, and 19.50 degrees vertically (centred around the display centre).

The validation grid is displayed in Figure 1. Each target was a dot with a diameter of 0.44 degrees. Targets were displayed individually for 2000 ms, followed by an inter target interval of 1000 ms in which no target was visible.

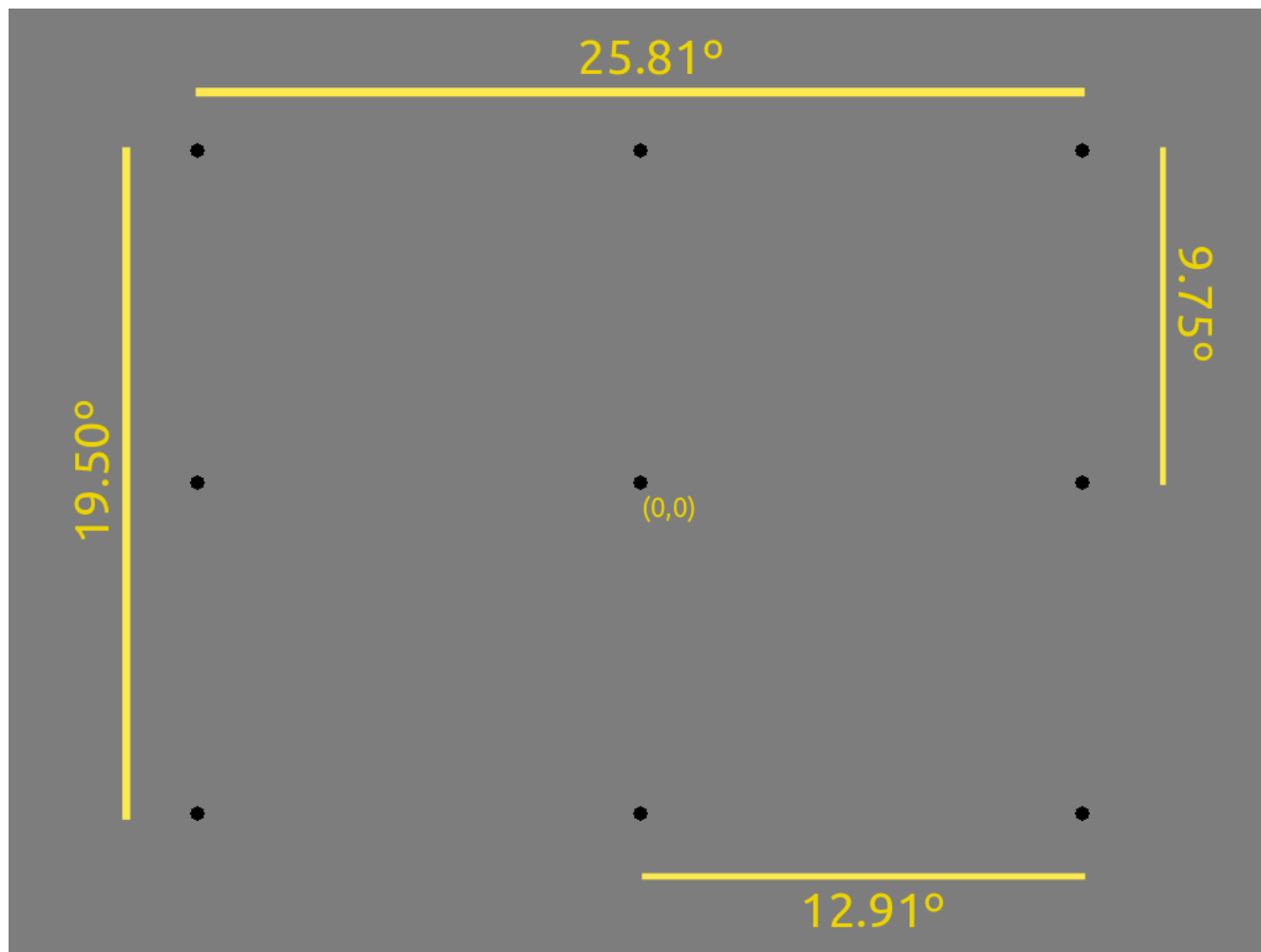


Figure 1 – Validation display. For illustrative purposes, all targets are displayed in one figure. In reality, each target was shown individually for 2 seconds, with an inter target interval of 1 second. Annotated numbers indicate the distances in degrees of visual angle. The central dot is at the display centre (0,0).

The grid spans a relatively small part of the area in which healthy individuals are able to move their eyes. EyeLink 1000 devices remain accurate far beyond these boundaries, and EyeTribe devices *might* be accurate beyond these extends as well, if used under optimal conditions. However, researchers that plan to use an EyeTribe tracker for the recording of eye movements into highly peripheral parts of the visual field, are encouraged to thoroughly test the data quality produced by their setup and participants.

2.3 Experiment 2: Free Face Viewing

Fixations are periods of relatively stable gaze, in which individuals are assumed to attend to the stimulus at their point of regard. For this reason, analysing fixation patterns is interesting in psychological research. Examples can be found in studies of neglect syndrome, where the amount of fixations to the disregarded field increases as its relative salience increases (Bays, Singh-Curry, Gorgoraptis, Driver, & Husain, 2010); and in studies of individuals with autism, who show different gaze patterns than healthy controls towards faces (less focus on faces and eyes), while at the same time showing differences from healthy controls in activity in brain areas associated with face and emotion processing (Dalton et al., 2005).

A common method in these types of studies, is to assign Areas of Interest (AOI) to an image, e.g. for different facial features. The AOIs used for the current study are displayed in Figure 2. The number of fixations, the average fixation duration, and the total amount of time spent looking at a specific AOI (*total dwell time*) can be used as indices of how much an individual attended a specific feature. The image used in this manuscript are from the NimStim stimulus set of facial expressions (Tottenham et al., 2009).

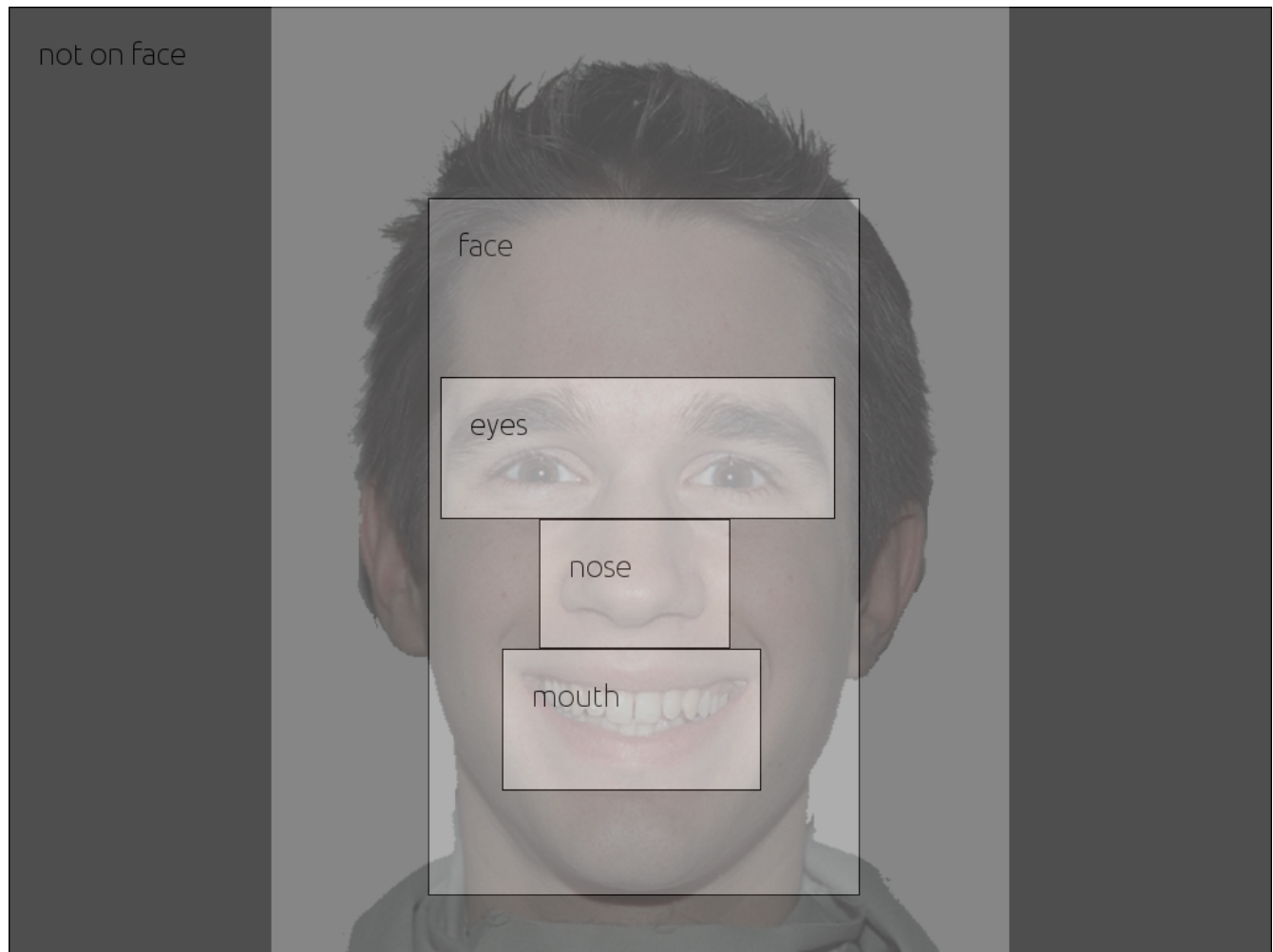


Figure 2 – Face stimulus, with corresponding regions of interest superimposed on the image.

Data was collected on two consecutive viewings of the same image, using an EyeLink 1000 on the first and an EyeTribe tracker on the second viewing. The image spanned 22.87 degrees in width and 27.59 in height, and was centred on the display.

2.4 Experiment 3: Pupilometry

The pupillary light response was tested in a total of 50 trials for each tracker, in which black and white screens alternated while a central grey fixation dot (0.44 degrees in diameter) was always visible. This design resulted in 25 trials per condition, during which pupil size was measured for a baseline period of 200 ms preceding the screen luminance change, and during the two following seconds. After this, a one second inter-trial interval occurred, in which no data was collected and the display remained unchanged.

2.5 Experiment 4: Saccadometry

For each eye tracker, eye movements were recorded during 50 saccades from a fixation cross (0.44 degrees in diameter) to a target (0.44 degrees in diameter), which were separated by 18.6 degrees. At the start of each trial, a fixation cross appeared. After 1000 ms and after the participant fixated the cross, a target dot appeared. The trial ended 500 ms after a participant's gaze position came within 2.2 degrees of the target, and was followed by a 1000 ms inter trial interval.

3. Results

3.1 Precision and Accuracy

Tracker accuracy was calculated for each participant by the averaging of all horizontal and vertical distances between validation target locations and the gaze samples obtained when a participant was fixating these targets, excluding missing data. The resulting values were averaged across all participants.

Precision was determined by calculating the horizontal and vertical root mean square (RMS) noise (that is: the RMS of the inter-sample distances) over all samples obtained when a participant was fixating target stimuli, excluding missing data. The resulting values were averaged across participants. Results are summarised in Table 1, and collected samples are visualised in Figure 3.

Table 1

Accuracy (mean difference between target location and observed point of regard in degrees of visual angle) and precision (RMS noise in degrees of visual angle). Data obtained from an EyeLink 1000 and an EyeTribe tracker, using nine calibration points (see Figure 2), with a viewing time of two seconds each. Calibration points spanned a range from -12.91 to 12.91 horizontally, and -9.75 to 9.75 vertically around the display centre.

	EyeLink 1000	EyeTribe
accuracy		
<i>horizontal</i>	0.02001	-0.05152
<i>vertical</i>	0.13224	0.18999
precision		
<i>horizontal</i>	0.00005	0.00105
<i>vertical</i>	0.00011	0.00333

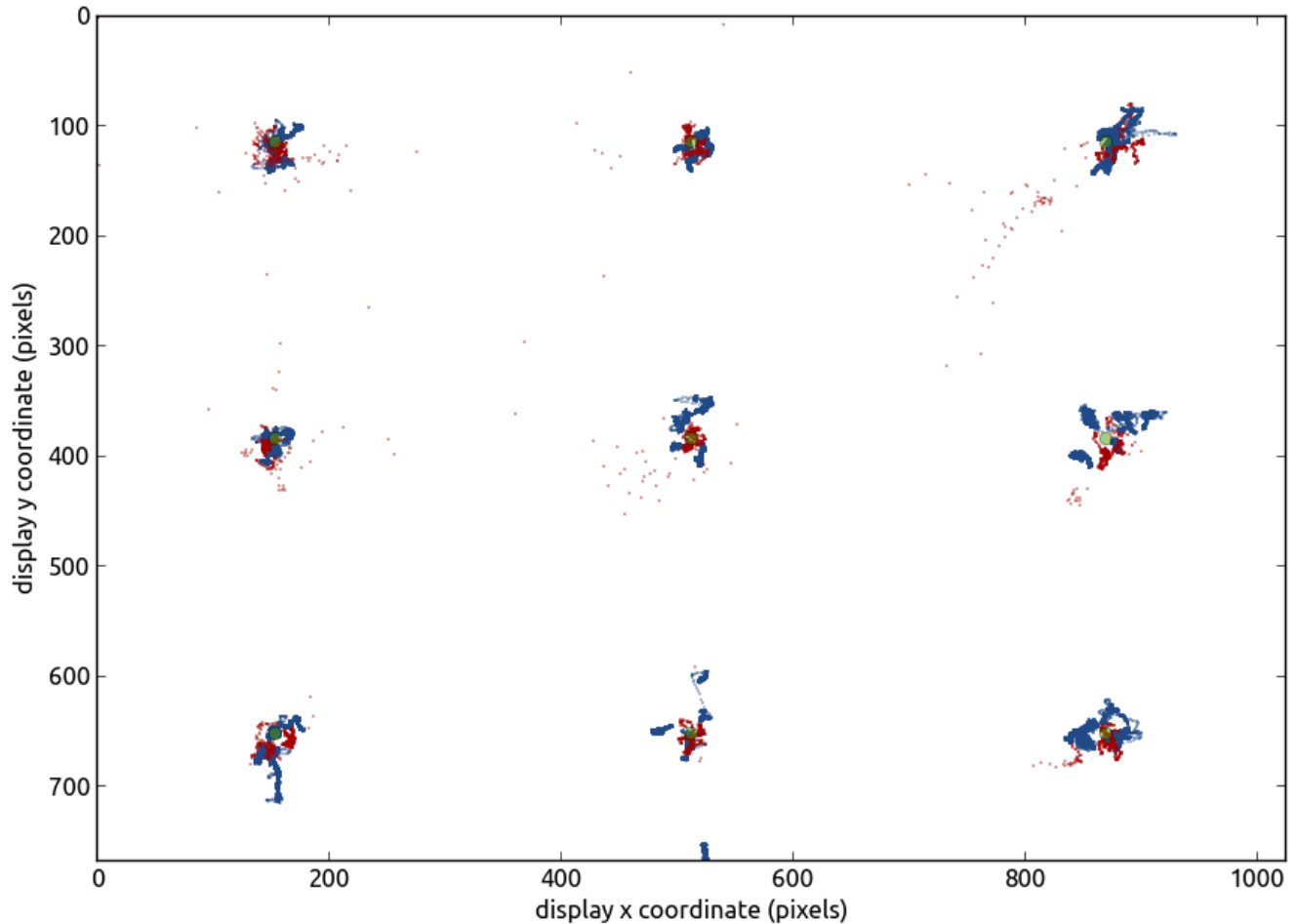


Figure 3 – All samples obtained from five participants using an EyeLink 1000 (blue dots) and an EyeTribe tracker (red dots) during the presentation of nine single targets (2000 milliseconds each), collapsed into a single graph for improved figure clarity. Green dots indicate true target locations.

3.2 Free Viewing Fixation Analyses

Fixations were defined as a period of stable gaze (within 1 degree of visual angle) that lasted for at least 50 milliseconds. Detection of fixations in data of both eye trackers was performed using exactly the same script, using the same parameters, and without resampling any of the input. The number of fixations, the average fixation duration, and the total dwell time were calculated for each participant, and then averaged across participants. The results of the AOI analyses of these indices are summarized in Table 2.

Table 2

Mean number of fixations, mean fixation duration (milliseconds), and mean total dwell time (milliseconds) for all regions of interest in the image shown in Figure 2. Data was collected during 10 seconds of free viewing. Note that the EyeLink 1000 and EyeTribe recordings did not co-occur: these are separate viewings of the same image.

<i>EyeLink 1000</i>	eyes	nose	mouth	face	not on face
N fixations	8.8	2.4	3.6	4	5.6
% of total	36.07	9.84	14.75	16.39	22.95
fixation duration	379	268	487	409	300
dwell time	3434	613	1717	1507	1613
% of total	38.65	6.90	19.33	16.96	18.16
<i>EyeTribe</i>					
N fixations	12.4	3.4	2.6	2.4	6.6
% of total	45.26	12.41	9.49	8.76	24.09
fixation duration	388	313	329	246	233
dwell time	4365	1013	1046	702	1474
% of total	50.76	11.78	12.16	8.16	17.14

The same fixation data can be visualised in different ways, for a more qualitative analysis. This is demonstrated in Figure 4, where the top row (A and B) shows results from EyeLink 1000 data, and the bottom row (C and D) shows results from EyeTribe data. In the left column (A and C), all fixations from all participants are plotted as transparent green dots, with dot size indicating fixation duration (the larger a dot is, the longer the fixation lasted). The right column (B and D) shows heatplots of all fixations of all participants, where 'hotter' colours indicate areas of the image in which more and longer fixations occurred. A slight upward drift seems to be present in the EyeTribe data (Figure 4C-D), compared to the EyeLink 1000 data (Figure 4A-B), which might explain some of the differences between the AOI analyses for both trackers.

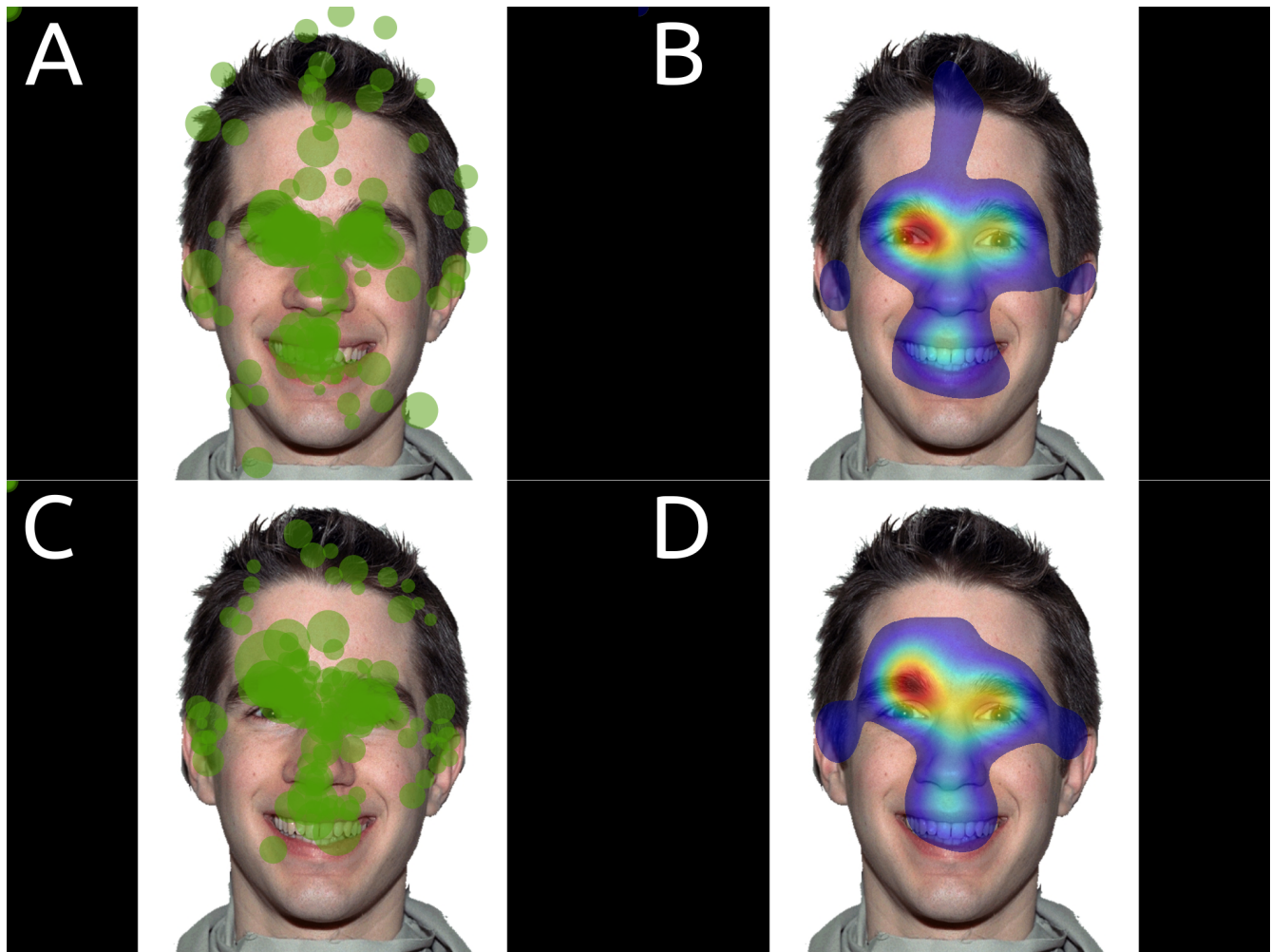


Figure 4 – Fixation plots (A and C) and heatmaps (B and D) the gaze behaviour of five participants measured using an EyeLink 1000 (A and B), and an EyeTribe tracker (C and D) in two different sessions. Green opaque dots indicate all fixations of five participants, with larger dots denoting longer fixation durations. Heatplots indicate fixation behaviour of all five participants, with hotter colours indicating more and longer fixations.

3.3 Pupilometry

Pupil diameter was recorded in consecutive testing sessions of each participant, on alternating dark and bright displays. Missing data and blinks were removed from the data, and were interpolated using a cubic spline (cf. Mathôt et al., in press), or a linear function when there were too little data points for a cubic spline interpolation. A Hampel filter (Pearson, 2002) was applied to remove outliers. Finally, the pupil signal was divided by the

median pupil signal in the baseline period, which resulted in the proportional change in pupil diameter relative to baseline.

To test for significant differences, a paired-samples t-test was performed for each timepoint, with a Bonferroni-corrected alpha of 0.000025 (an alpha of 0.05 corrected for 2000 comparisons in the EyeLink 1000 data; the same corrected alpha was applied to the 120 comparisons in the EyeTribe data). This correction is unusually stringent, yet due to the robustness of the pupillary light response, a clear significant effect is shown to be present from around 500 ms after a display luminance switch. Figure 5 shows the results of the pupil analyses of both eye trackers for a single (representative) participant. The traces for the other participants are included in the supplementary material.

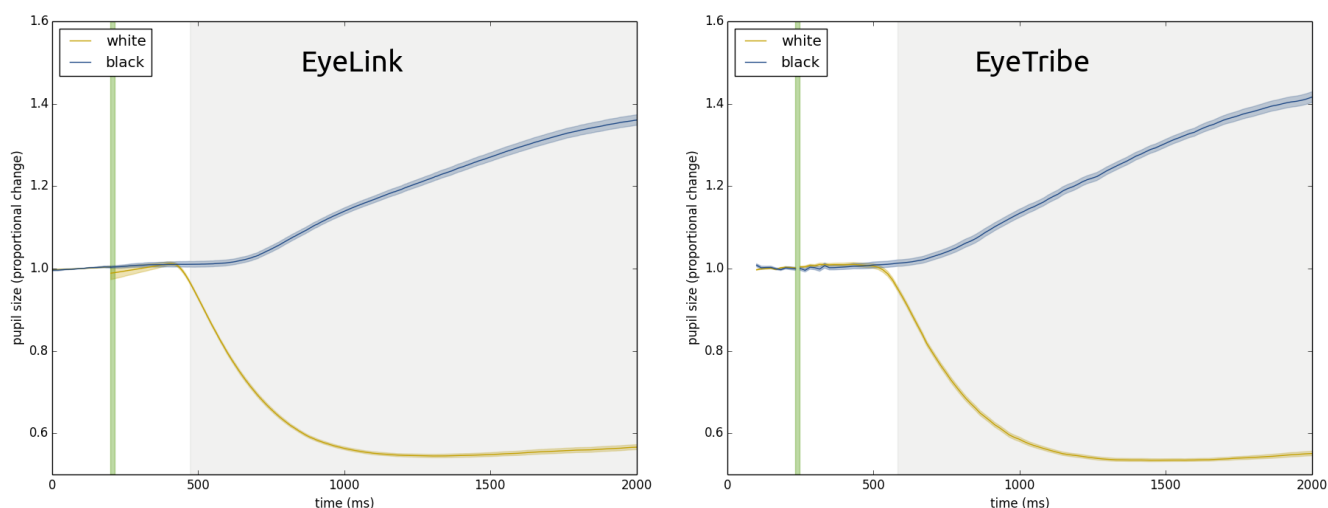


Figure 5 – Proportional change in pupil diameter after the onset (green shading) of either a dark (blue line) or a bright (yellow line) background. Lines represent the pupil response averaged over the mean pupil responses from a single (representative) participant, obtained with 25 trials for each luminance. Blue and yellow shading indicates the standard error of the mean. Grey shading indicates a significant difference at $p < 0.000025$ (timepoint-wise related-samples t-tests).

3.4 Saccade Metrics

Saccades were defined as periods of at least 20 ms in which gaze velocity exceeded 20 degrees of visual angle per second. Saccade trajectories and velocity profiles of a single (representative) participant are plotted in Figure 6 (plots from the other participants are included in the supplementary material). From the graphs, it is immediately obvious that the EyeTribe velocity profiles and trajectories look 'edgy', a clear sign of under-sampling, resulting in low-quality trajectory and velocity data.

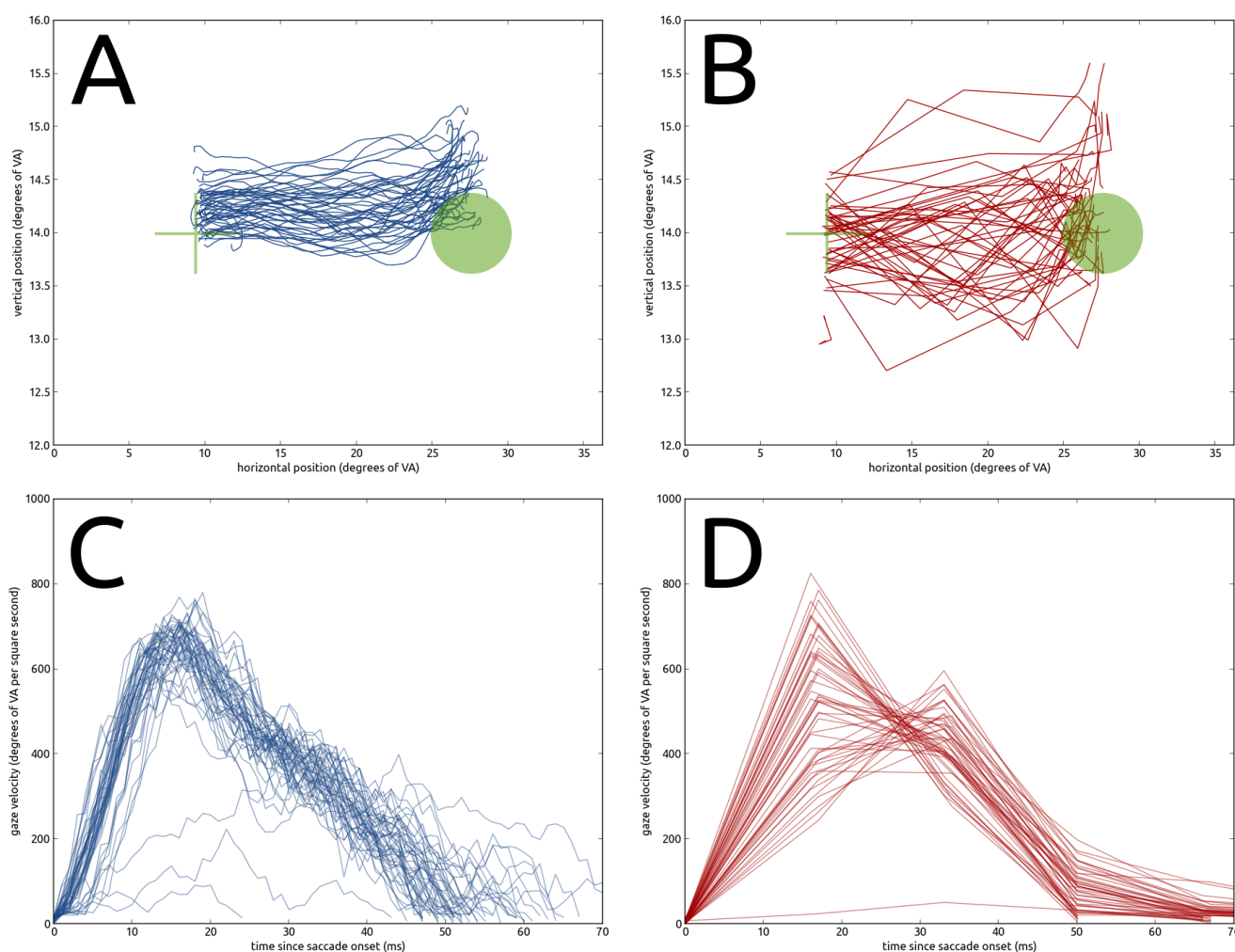


Figure 6 – Trajectories (A and B) and velocity profiles (C and D) for saccades collected from a single (representative) participant, using an EyeLink 1000 (A and C, blue lines) and an EyeTribe tracker (B and D, red lines). Solid lines represent individual trials (N = 50), transparent green objects indicate position and height of stimuli.

The saccade starting and landing error were defined as the average Euclidean distance between the fixation stimulus centre and the saccade starting position, and the average Euclidean distance between the target stimulus centre and the saccade ending position, respectively. The saccade curvature was defined as by Van der Stigchel, Meeter, & Theeuwes (2006). The peak velocity was defined as the maximum value in the velocity trace (i.e. the peak from each single line in Figure 6C and 6D), and the mean velocity as the average of all values in the velocity trace.

For each participant, each saccade's absolute starting and landing error, amplitude, duration, curvature, mean velocity, and peak velocity were calculated from data obtained from the EyeLink 1000 and from the EyeTribe tracker (50 saccades each, for each participant). The results were averaged across participants, and are summarized in Table 3. They give an indication of how well the EyeTribe approximates EyeLink 1000 performance.

Table 3

Summarized saccadometry indices that were calculated using data from an EyeLink 1000, and from an EyeTribe tracker. Reported values are averages of five participants, who each made 50 saccades. Standard deviations are reported between round brackets.

	EyeLink 1000	EyeTribe
starting error (degrees of visual angle)	0.79 (0.24)	2.62 (3.71)
landing error (degrees of visual angle)	2.33 (0.80)	6.60 (8.36)
amplitude (degrees of visual angle)	16.56 (0.62)	15.25 (3.75)
duration (milliseconds)	66.68 (8.68)	114.04 (19.32)
curvature (degrees of visual angle)	0.36 (0.45)	1.85 (2.37)
mean velocity (degrees of visual angle / second)	37.84 (3.44)	93.59 (4.31)
peak velocity (degrees of visual angle / second)	849.23 (384.45)	797.15 (482.51)

It is clear from the results that rather large discrepancies exist between the two eye trackers for some saccadometry measures, but not for others. The fixation and landing error, as well as the saccade amplitude seem to be accurately measured by both systems. However, an overestimation of saccade duration, as well as underestimations of curvature and velocities result from analyses of the EyeTribe data, compared to the EyeLink 1000 data.

3.5 Sampling

The EyeTribe's specifications promise a 60 Hz sampling rate, which should result in a steady 16.67 ms inter-sample time. However, this does not always seem to be true. For the EyeTribe, 99.08% of 88561 collected inter-sample times are actually near 16.67 ms. In contrast, 100% of 1641356 collected EyeLink inter-sample times are the expected 1 ms at a sampling rate of 1000 Hz. See Table 4 for the absolute and relative frequencies of inter-sample times obtained from the EyeTribe.

Out of 89121 samples collected from all five participants using an EyeTribe tracker, 6182 (6.94 %) were invalid (missing). In contrast, out of 1641921 samples collected from all participants using an EyeLink 1000, only 23047 (1.40 %) were invalid.

Table 4

Spread of 13825 inter-sample times obtained from an EyeTribe tracker running at 60 Hz. Ranges are almost equally wide, with a narrower band around the expected sampling time (16.67 ms). Ranges include the endpoint, but not the starting point (except for the range starting at 0, where 0 was included).

inter-sample time range (ms)	N	frequency (%)
0.00 – 14.67	0	0.00
14.67 – 18.67	87 750	99.08
18.67 – 35.00	785	0.89
35.00 – 50.00	7	0.01
> 50.00	19	0.02

4. Software

To facilitate researchers that want to use the EyeTribe in psychological research, the author developed toolboxes and plug-ins for several software packages that are often used to create experiments (Python, Matlab, and OpenSesame). Example scripts for the presentation of a gaze contingent dot in Python and Matlab are included in the Appendices.

4.1 EyeTribe API and PyTribe

The EyeTribe provides direct support for C#, C++, and Java. Furthermore, an open API is provided to create custom software using TCP socket communication through other programming languages. A popular programming language that provides this option is Python (Van Rossum & Drake, 2011), an interpreted language that is open-source, widely used, and versatile.

Here PyTribe is presented, a Python wrapper that captures all the functionality of the EyeTribe API. It can be used to develop Python-based applications, and is relatively user-friendly. An example of an experiment can be found in Appendix 1, which uses PyGame for the presentation of a gaze-contingent dot.

One downside of PyTribe (and of using the EyeTribe API directly) is that users are required to implement their own calibration and validation routines, which requires skill and is time consuming. Therefore, PyTribe has been integrated within existing experimentation software (listed in the following paragraphs), so that researchers can create experiments without each having to reinvent the wheel.

4.2 Python

The first of these implementations is in PyGaze (Dalmaijer, Mathôt, et al., 2013), a toolbox for experiment programming in Python that focuses on user-friendliness and script readability. PyGaze has routines for the presentation of visual and auditive stimuli, and for response collection using keyboards, computer mice, joysticks, webcams, and BioPac MP150 systems. Crucially, PyGaze offers support for multiple types of eye trackers from several manufacturers: EyeLink (SR Research), SensoMotoric Instruments, and Tobii.

Using PyTribe, the EyeTribe tracker has been incorporated into the PyGaze toolbox, so that it can be initialized and calibrated with calls to a single method. Additionally,

methods for drift checks, starting and pausing of data recording, event logging, real-time sampling of gaze position and pupil size, and online detection of saccades, fixations and blinks are provided. To increase the usability of PyGaze-based scripts, users can switch between eye trackers by changing the name of a single constant. This means that no reprogramming of tasks is required when changing eye-trackers. An example PyGaze script for the EyeTribe is provided in Appendix 2.

4.3 OpenSesame

Recently, PyGaze-based plug-ins for OpenSesame (Mathôt et al., 2012) have been created. OpenSesame is an open-source, graphical experiment builder that resembles proprietary experiment builders like Experiment Builder (SR Research), E-Prime (Psychology Software Tools), and Presentation (NeuroBehavioral Systems). Earlier plug-ins provided support for EyeLink and SMI systems (Dalmaijer, Van der Stigchel, et al., 2013). With the new PyGaze plug-ins, additional support is realised for Tobii and, crucially, EyeTribe devices. An example of how a fully functional eye-tracking experiment looks in OpenSesame's *drag-and-drop* environment is provided in Figure 7.

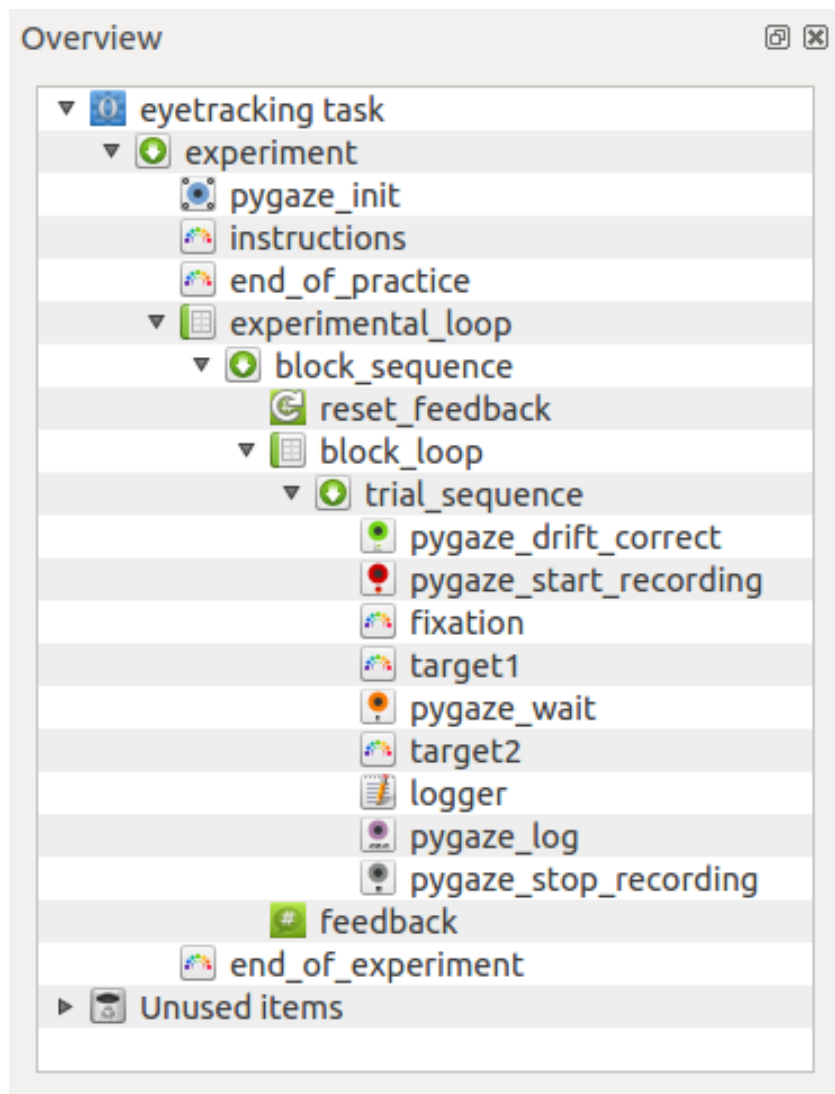


Figure 7 – Overview of an eye-tracking experiment created in the Graphical User Interface (GUI) of the OpenSesame experiment builder.

4.4 Matlab

Although Matlab (the MathWorks Inc.) provides native functionality for TCP communication, it lacks the multithreading functionality that is required for the collection and storage of data from the EyeTribe tracker in the background, while keeping the connection with the device open. Therefore, a non-native Matlab solution was developed in the form of a custom server that communicates with the EyeTribe server and collects data on Matlab's behalf (see Figure 8). From Matlab, commands can be send to this second server

to calibrate the tracker, to start or pause recording, and to obtain real-time samples of gaze position and pupil size. The visuals in the *EyeTribe Toolbox for Matlab* are coded using the popular *Psychophysics Toolbox for Matlab* (Brainard, 1997). An example script is provided in Appendix 3.

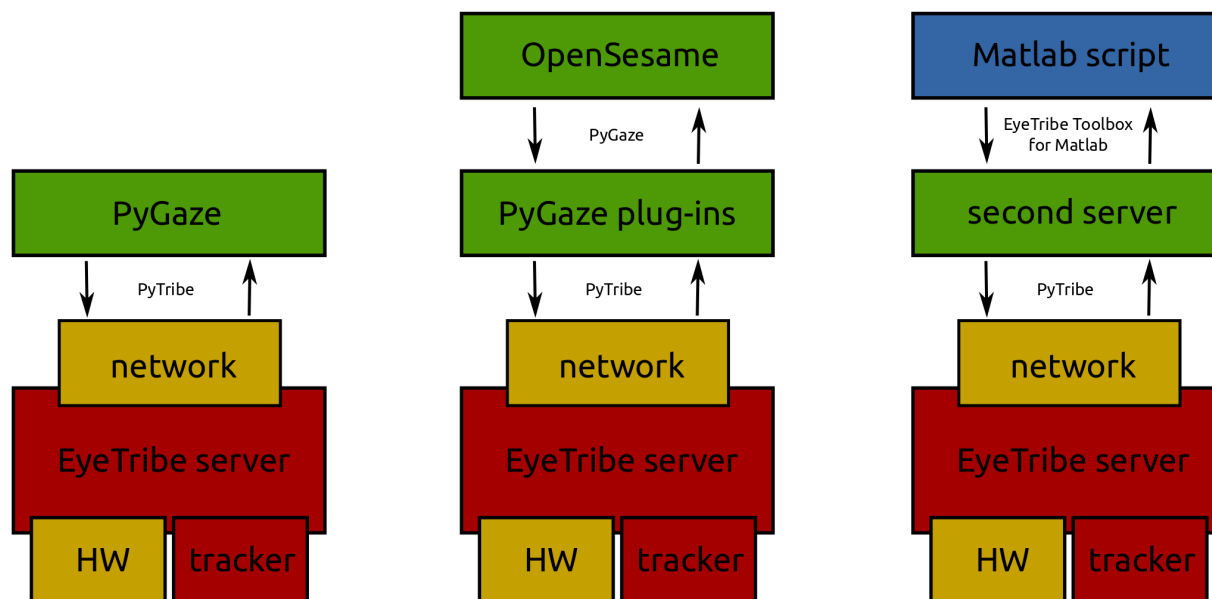


Figure 8 – Illustration of the communication hierarchy in PyGaze, OpenSesame, and the EyeTribe Toolbox for Matlab. Communication between a computer (yellow components) and an EyeTribe tracker are handled by the EyeTribe Server application (red components). When using Python-based applications (green components), a PyGaze EyeTracker instance is responsible for keeping the connection open, and for the collection and storage of data from the EyeTribe Server. For Matlab (blue components), a second server application continuously monitors the EyeTribe Server, to collect and store data and to keep the connection open. From Matlab, commands can be sent to the second server, which will then handle the request. Illustrations based on EyeTribe's system architecture overview: dev.theeyetribe.com/dev/.

4.5 Supported platforms

The EyeTribe Server is only available for Windows (7 and 8) and Max OS X. The device requires a USB 3.0 connection, making it unusable on any system that does not have such a port. All Python-based applications mentioned in this article are platform-independent, and

will run on Windows and OS X. Linux support is not available at the time of writing, but will be once a version of the EyeTribe Server will be released for Linux (this is dependent on the EyeTribe's manufacturer). The Matlab toolbox currently is restricted to Windows, due to its dependency on the custom second server. However, the source code for this second server is freely available online, and can be compiled to a standalone app. Alternatively, it can be run from source with a local Python installation.

4.6 Availability

All the new software mentioned in this article is hosted on GitHub, where it can be accessed, downloaded, and reviewed by anyone with a computer and an internet connection. This allows for large scale, international cooperation according to the *Best Practices of Scientific Computing* as formulated by Wilson et al. (2014). The software is released under General Public Licenses, version 3 (Free Software Foundation, 2007), assuring that it will always be free to download and use for everyone.

The EyeTribe's Tracker API has been free up until now, with no signs of that changing. However, the SDK (including the EyeTribe Server and a User Interface tool) can only be downloaded by individuals who have registered their details on the EyeTribe website (which one can do without buying a device).

5. Discussion

In this article, the recently introduced EyeTribe tracker is discussed. Due to its low price (its marketed as \$99 dollars), the eye tracker seems to be a promising resource for psychological research. Toolboxes and plug-ins to integrate the EyeTribe tracker with popular experimentation software (Python, Matlab, and OpenSesame) are introduced. Furthermore, the first objective tests of the EyeTribe's performance in a broad range of tests (that are often employed in psychological research) are performed. The results show that the EyeTribe can be used in research, but not in all kinds.

The presented software can be used to compliment currently existing packages that are popular among researchers: Matlab and the Psychophysics Toolbox (Brainard, 1997), Python and PsychoPy (Peirce, 2007) or PyGaze (Dalmaijer, Mathôt, et al., 2013), and experiment-builder OpenSesame (Mathôt et al., 2012). This allows researchers to continue using the software of their preference, with as little new functions to learn as possible. The presented EyeTribe toolbox for Matlab works in a slightly roundabout way, which might be improved in the future, but which should result in better timing accuracy than using Matlab's native multiprocessing and TCP functionality.

The EyeTribe's accuracy and precision seem to be quite good under optimal conditions (testing with a chin rest, in a dimly lit room, and no sources of infrared interference), and sufficient to do fixation research. Unsurprisingly, the EyeLink 1000's accuracy and precision are better.

The EyeTribe's sampling rate proved to be very constant (less than one percent of the samples were dropped), and showed signal loss in about seven percent of the tested samples. This is quite high in comparison with the EyeLink 1000, which showed between one and two percent signal loss.

Data from the EyeTribe tracker presented here, indicates that performance on fixation analyses and pupilometry grossly approximates that of data from an EyeLink 1000 (again, under optimal conditions). This is not surprising, as these kind of analyses do not require high-frequency sampling, but it is a reassuring result for researchers that are contemplating using the EyeTribe tracker in studies of fixation behaviour or pupil size modulations. However, researchers should be aware of using eye trackers with lower sampling frequencies in general, as the sampling rate is inversely related with the

measurement error of fixation and saccade durations, although this can be compensated by using more observations (Andersson, Nyström, & Holmqvist, 2010).

Researchers should be cautious when using an EyeTribe tracker to collect saccade metrics that do require high-frequency sampling. The results presented here show that saccade trajectories and velocities cannot be measured accurately due to undersampling. Somewhat surprisingly, simple velocity-based algorithms for saccade detection that work on high-frequency data, do a relatively good job on EyeTribe data as well: saccadic amplitude is comparable between analyses of EyeLink 1000 and EyeTribe data, but estimations of saccadic starting and ending positions appear to be less accurate when obtained from EyeTribe data.

In conclusion, the EyeTribe tracker can be a potentially valuable resource for psychological research. However, one should keep in mind its limitations. The take home message from the tests presented here, is that high-frequency eye trackers are preferred in studies of saccade metrics, and should generally be preferred if budget allows. However, researchers on a budget can use the EyeTribe for studies of fixation behaviour and pupil size.

References

- Andersson, R., Nyström, M., & Holmqvist, K. (2010). Sampling frequency and eye-tracking measures: How speed affects durations, latencies, and more. *Journal of Eye Movement Research*, 3(3:6), 1–12.
- Bays, P. M., Singh-Curry, V., Gorgoraptis, N., Driver, J., & Husain, M. (2010). Integration of Goal- and Stimulus-Related Visual Signals Revealed by Damage to Human Parietal Cortex. *Journal of Neuroscience*, 30(17), 5968–5978. doi:10.1523/JNEUROSCI.0997-10.2010
- Bethlehem, R. A. I., Dumoulin, S. O., Dalmaijer, E. S., Smit, M., Berendschot, T. T. J. M., Nijboer, T. C. W., & Van der Stigchel, S. (2014). Decreased Fixation Stability of the Preferred Retinal Location in Juvenile Macular Degeneration. *PLoS ONE*, 9(6), e100171. doi:10.1371/journal.pone.0100171
- Brainard, D. H. (1997). The Psychophysics Toolbox. *Spatial Vision*, 10(4), 433–436. doi:10.1163/156856897X00357
- Dalmaijer, E. S., Mathôt, S., & Van der Stigchel, S. (2013). PyGaze: an open-source, cross-platform toolbox for minimal-effort programming of eye-tracking experiments. *Behavior Research Methods*. doi:10.3758/s13428-013-0422-2
- Dalmaijer, E. S., Van der Stigchel, S., Van der Linden, L., Kruijne, W., Schreij, D., & Mathôt, S. (2013). OpenSesame Opens the Door to Open-Source and User-Friendly Eye-Tracking Research. *Journal of Eye Movement Research*, 6(3), 547.
- Dalton, K. M., Nacewicz, B. M., Johnstone, T., Schaefer, H. S., Gernsbacher, M. A., Goldsmith, H. H., ... Davidson, R. J. (2005). Gaze fixation and the neural circuitry of face processing in autism. *Nature Neuroscience*. doi:10.1038/nn1421
- Free Software Foundation. (2007). GNU General Public License. Retrieved from <https://gnu.org/licenses/gpl.html>

- Holmqvist, K., Nyström, M., & Mulvey, F. (2012). Eye tracker data quality: what it is and how to measure it (p. 45). ACM Press. doi:10.1145/2168556.2168563
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90–95. doi:10.1109/MCSE.2007.55
- Laubrock, J., Kliegl, R., Rolfs, M., & Engbert, R. (2010). When do microsaccades follow spatial attention? *Attention, Perception, & Psychophysics*, 72(3), 683–694. doi:10.3758/APP.72.3.683
- Li, D., Babcock, J., & Parkhurst, D. J. (2006). openEyes: a low-cost head-mounted eye-tracking solution (p. 95). ACM Press. doi:10.1145/1117309.1117350
- Mathôt, S., Dalmaijer, E. S., Grainger, J., & Van der Stigchel, S. (in press). The pupillary light response reflects exogenous attention and inhibition of return. *Journal of Vision*.
- Mathôt, S., Schreij, D., & Theeuwes, J. (2012). OpenSesame: An open-source, graphical experiment builder for the social sciences. *Behavior Research Methods*, 44(2), 314–324. doi:10.3758/s13428-011-0168-7
- McCamy, M. B., Najafian Jazi, A., Otero-Millan, J., Macknik, S. L., & Martinez-Conde, S. (2013). The effects of fixation target size and luminance on microsaccades and square-wave jerks. *PeerJ*, 1, e9. doi:10.7717/peerj.9
- Oliphant, T. E. (2007). Python for Scientific Computing. *Computing in Science & Engineering*, 9(3), 10–20. doi:10.1109/MCSE.2007.58
- Pearson, R. K. (2002). Outliers in process modeling and identification. *IEEE Transactions on Control Systems Technology*, 10(1), 55–63. doi:10.1109/87.974338
- Peirce, J. W. (2007). PsychoPy—Psychophysics software in Python. *Journal of Neuroscience Methods*, 162(1-2), 8–13. doi:10.1016/j.jneumeth.2006.11.017
- Peirce, J. W. (2009). Generating stimuli for neuroscience using PsychoPy. *Frontiers in Neuroinformatics*, 2. doi:10.3389/neuro.11.010.2008
- San Agustin, J., Skovsgaard, H., Hansen, J. P., & Hansen, D. W. (2009). Low-cost gaze

interaction (p. 4453). ACM Press. doi:10.1145/1520340.1520682

San Agustin, J., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D. W., & Hansen, J. P. (2010). Evaluation of a low-cost open-source gaze tracker (p. 77). ACM Press. doi:10.1145/1743666.1743685

Sogo, H. (2013). GazeParser: an open-source and multiplatform library for low-cost eye tracking and analysis. *Behavior Research Methods*, 45(3), 684–695. doi:10.3758/s13428-012-0286-x

Tottenham, N., Tanaka, J. W., Leon, A. C., McCarry, T., Nurse, M., Hare, T. A., ... Nelson, C. (2009). The NimStim set of facial expressions: Judgments from untrained research participants. *Psychiatry Research*, 168(3), 242–249. doi:10.1016/j.psychres.2008.05.006

Van der Stigchel, S., Meeter, M., & Theeuwes, J. (2006). Eye movement trajectories and what they tell us. *Neuroscience & Biobehavioral Reviews*, 30(5), 666–679. doi:10.1016/j.neubiorev.2005.12.001

Van der Stigchel, S., Mulckhuyse, M., & Theeuwes, J. (2009). Eye cannot see it: The interference of subliminal distractors on saccade metrics. *Vision Research*, 49(16), 2104–2109. doi:10.1016/j.visres.2009.05.018

Van Rossum, G., & Drake, F. L. (2011). *Python Language reference manual*. Bristol, UK: Network Theory Ltd.

Wilson, G., Aruliah, D. A., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., ... Wilson, P. (2014). Best Practices for Scientific Computing. *PLoS Biology*, 12(1), e1001745. doi:10.1371/journal.pbio.1001745

Appendix 1 – Python and PyTribe example

An example experiment (written in Python) that uses PyTribe directly, assuming that calibration has been performed before running the script (e.g. using the EyeTribe's GUI calibration software).

```
1 import os
2 import pygame
3 from pytribe import EyeTribe
4
5 # screen details
6 RESOLUTION = (1280,1024)
7
8 # files and paths
9 DIR = os.path.dirname(os.path.abspath(__file__))
10 LOGFILE = os.path.join(DIR, "example_data.txt")
11 IMGDIR = os.path.join(DIR, "imgs")
12 IMGNames = os.listdir(IMGDIR)
13
14 # start communications with the EyeTribe tracker
15 tracker = EyeTribe(logfilename=LOGFILE)
16
17 # initialize a new PyGame display Surface
18 pygame.init()
19 disp = pygame.display.set_mode(RESOLUTION, pygame.FULLSCREEN)
20
21 # start recording gaze data
22 tracker.start_recording()
23
24 # loop for 10 seconds
25 t0 = pygame.time.get_ticks()
26 while pygame.time.get_ticks() - t0 < 10000:
27
28     # draw the image
29     gazePos = tracker.sample()
30     # draw a new dot
31     disp.fill((128,128,128))
32     pygame.draw.circle(disp, (0,0,0), gazePos, 6)
33     # show the dot
34     pygame.display.flip()
35     tracker.log_message("flip")
36
37 # stop recording
38 tracker.stop_recording()
39
40 # close connection to the tracker
41 tracker.close()
42
43 # close the display
44 pygame.quit()
```


Appendix 2 – Python and PyGaze example

An example experiment (written in Python) that uses the PyGaze toolbox to interface with the EyeTribe tracker. Additional examples are available from www.pygaze.org.

```
1 from pygaze.libscreen import Display, Screen
2 from pygaze.libinput import Keyboard
3 from pygaze.eyetracker import EyeTracker
4 import pygaze.libtime as timer
5
6 # prepare objects
7 disp = Display(disptype="psycho", dispsize=(1920,1080))
8 scr = Screen(disptype="psycho", dispsize=(1920,1080))
9 kb = Keyboard()
10 tracker = EyeTracker(disp, trackertype="eyetribe")
11
12 # display instructions
13 scr.draw_text(text="Press any key to start the calibration.", fontsize=24)
14 disp.fill(scr)
15 disp.show()
16 kb.get_key(keylist=None, timeout=None, flush=True)
17
18 # calibrate the eye tracker
19 tracker.calibrate()
20
21 # start tracking
22 tracker.start_recording()
23
24 # loop for 10 seconds
25 t0 = timer.get_time()
26 while timer.get_time() - t0 < 10000:
27     # draw the dot
28     scr.clear()
29     scr.draw_fixation(fixtype="dot", pos=gazepos)
30     # present dot
31     disp.fill(scr)
32     timestamp = disp.show()
33     tracker.log("flip, time=%d" % timestamp)
34
35
36 # stop recording
37 tracker.stop_recording()
38
39 # close tracker connection and display
40 tracker.close()
41 disp.close()
```

Appendix 3 – Matlab, PsychToolbox, and EyeTribe Toolbox for Matlab example

An example script (written in Matlab) that uses the EyeTribe Toolbox for Matlab to interface with the EyeTribe tracker, and the Psychophysics Toolbox for Matlab to do the presentation of visual stimuli.

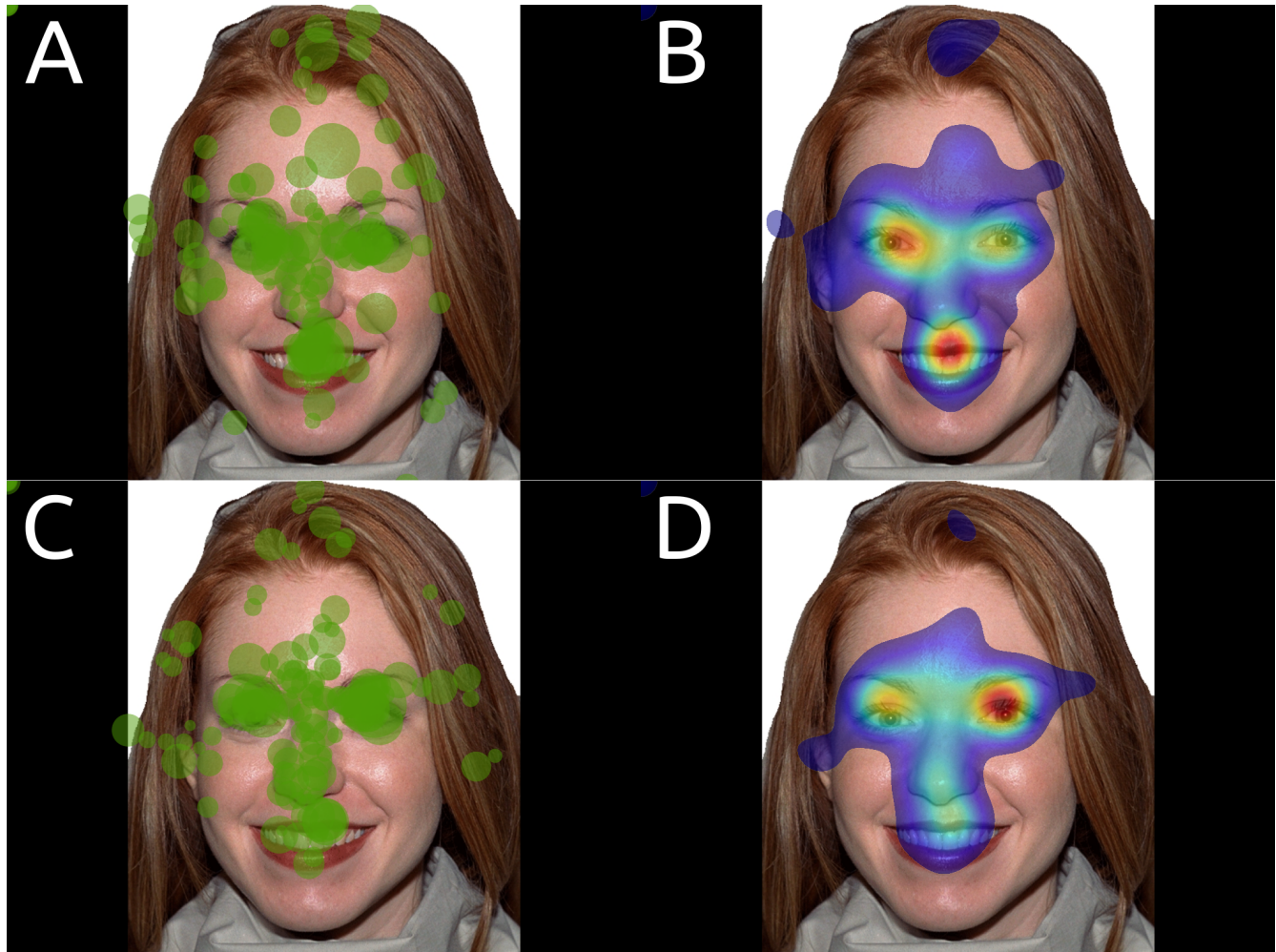
```

1  % don't bother with vsync tests for this demo
2  Screen('Preference', 'SkipSyncTests', 1);
3
4  % initialize connection
5  [success, connection] = eyetribe_init('subject-01');
6
7  % open a new window
8  window = Screen('OpenWindow', 0);
9
10 % calibrate the tracker
11 success = eyetribe_calibrate(connection, window);
12
13 % show grey window
14 Screen('FillRect', window, [128 128 128]);
15 Screen('Flip', window);
16
17 % start recording
18 success = eyetribe_start_recording(connection);
19
20 % loop for 10 seconds
21 t0 = getSecs()
22 while getSecs() - t0 < 10
23     % get new sample
24     [success, x, y] = eyetribe_sample(connection);
25     % draw a dot
26     Screen('FillOval', window, [0 0 0], [x-6; y-6; x+6; y+6]);
27     % display the dot
28     Screen('Flip')
29     success = eyetribe_log(connection, 'flip');
30 end
31
32 % stop recording
33 success = eyetribe_stop_recording(connection);
34
35 % close connection
36 success = eyetribe_close(connection);
37
38 % close window
39 Screen('Close', window);

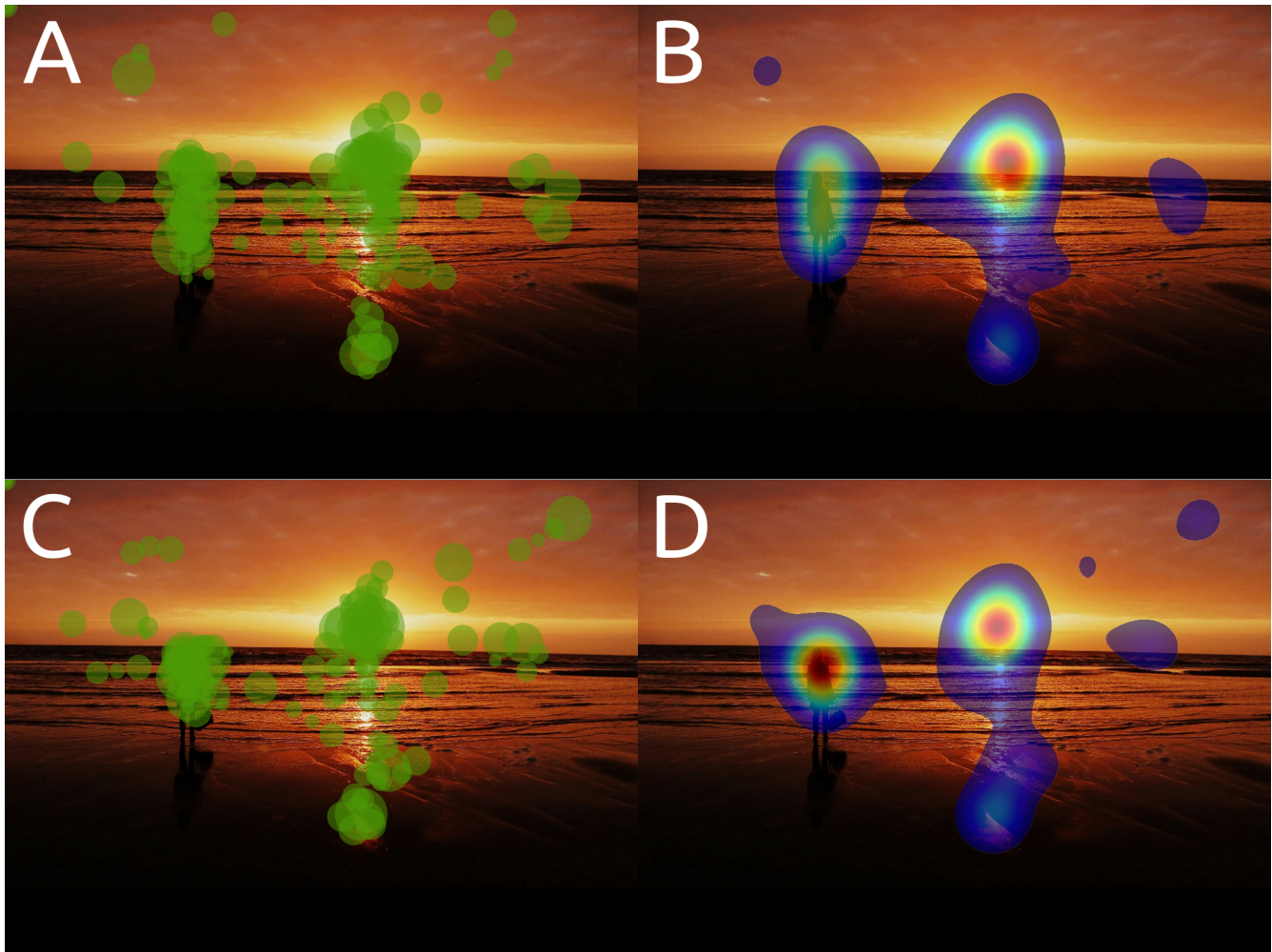
```

Supplementary Material

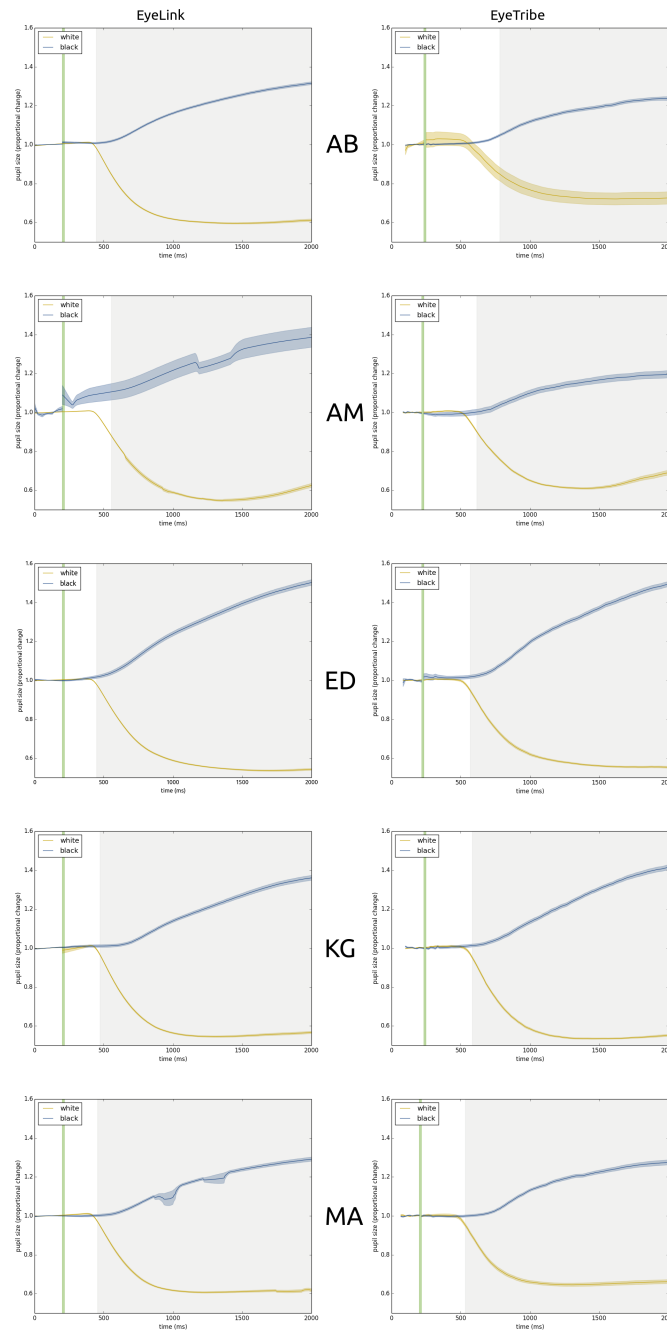
This section contains visualisations of analyses that were not included in the manuscript. These are also available from github.com/esdalmajjer/EyeTribe_test.



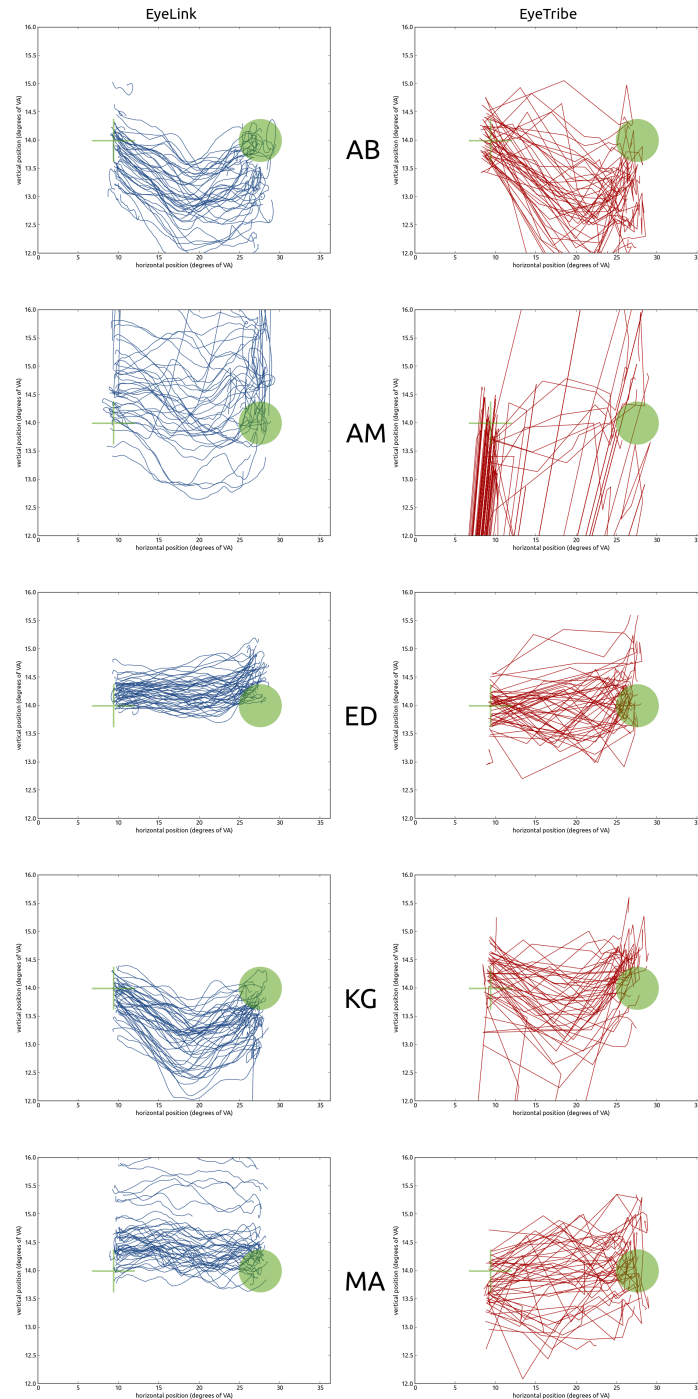
Supplementary Figure 1 - Fixation plots (A and C) and heatmaps (B and D) the gaze behaviour of five participants measured using an EyeLink 1000 (A and B), and an EyeTribe tracker (C and D) in two different sessions. Green opaque dots indicate all fixations of five participants, with larger dots denoting longer fixation durations. Heatplots indicate fixation behaviour of all five participants, with hotter colours indicating more and longer fixations.



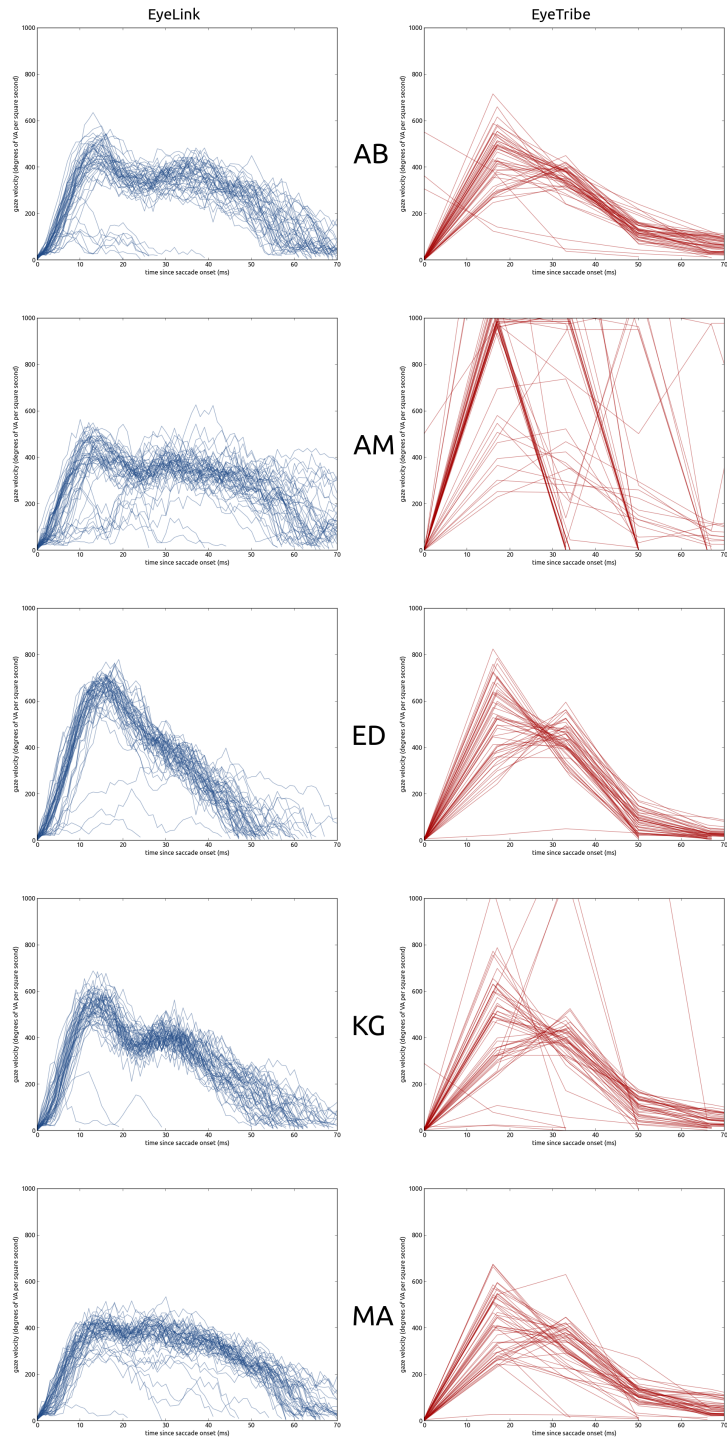
Supplementary Figure 2 - Fixation plots (A and C) and heatmaps (B and D) the gaze behaviour of five participants measured using an EyeLink 1000 (A and B), and an EyeTribe tracker (C and D) in two different sessions. Green opaque dots indicate all fixations of five participants, with larger dots denoting longer fixation durations. Heatplots indicate fixation behaviour of all five participants, with hotter colours indicating more and longer fixations.



Supplementary Figure 3 – Proportional change in pupil diameter after the onset (green shading) of either a dark (blue line) or a bright (yellow line) background. Lines represent the pupil response averaged over the mean pupil responses from single participants (each row is from a different participant), obtained with 25 trials for each luminance. Blue and yellow shading indicates the standard error of the mean. Grey shading indicates a significant difference at $p < 0.000025$ (timepoint-wise related-samples t -tests).



Supplementary Figure 4 – Trajectories for saccades collected from a single participants (each row is from a different participant), using an EyeLink 1000 (blue lines) and an EyeTribe tracker (red lines). Solid lines represent individual trials (N = 50), transparent green objects indicate position and height of stimuli.



Supplementary Figure 5– Velocity profiles for saccades collected from a single participants (each row is from a different participant), using an EyeLink 1000 (blue lines) and an EyeTribe tracker (red lines). Solid lines represent individual trials (N = 50).