

# GazeAlyze: a MATLAB toolbox for the analysis of eye movement data

Christoph Berger · Martin Winkels ·  
Alexander Lischke · Jacqueline Höppner

© Psychonomic Society, Inc. 2011

**Abstract** This article presents GazeAlyze, a software package, written as a MATLAB (MathWorks Inc., Natick, MA) toolbox developed for the analysis of eye movement data. GazeAlyze was developed for the batch processing of multiple data files and was designed as a framework with extendable modules. GazeAlyze encompasses the main functions of the entire processing queue of eye movement data to static visual stimuli. This includes detecting and filtering artifacts, detecting events, generating regions of interest, generating spread sheets for further statistical analysis, and providing methods for the visualization of results, such as path plots and fixation heat maps. All functions can be controlled through graphical user interfaces. GazeAlyze includes functions for correcting eye movement data for the displacement of the head relative to the camera after calibration in fixed head mounts. The preprocessing and event detection methods in GazeAlyze are based on the software ILAB 3.6.8 Gitelman (Behav Res Methods Instrum Comput 34(4), 605–612, 2002). GazeAlyze is distributed free of charge under the terms of the GNU public license and allows code modifications to be made so that the program's performance can be adjusted according to a user's scientific requirements.

**Keywords** Eye movement analysis · Eye tracking · Fixations · Saccades · Heat maps · MATLAB · Computer software

## Introduction

In recent psychophysiological research, eye-tracking methodology has been widely used to obtain reaction parameters from eye movement data to analyze cognitive processes underlying visual behavior. Eye movements are classified into two basic components: the movements of the eye itself (called *saccades*) and the times between movements when the eye remains in a single position (called *fixations*). Fixations typically occur only with the reception of sensory information, and vision is generally suppressed during saccades (Matin, 1974). Eye movements are closely linked to visual attention; therefore, many studies have used the analysis of saccadic target regions to investigate the link between eye movements and covert attention. Some of these studies have shown that attention is always oriented in the same direction as saccades (Deubel & Schneider, 1996; Hoffman & Subramaniam, 1995; Irwin & Gordon, 1998; Kowler, Anderson, Doshier, & Blaser, 1995). Others have shown that attention and eye movements can be dissociated (Awh, Armstrong, & Moore, 2006; Hunt & Kingstone, 2003). A recent article (Golomb, Marino, Chun, & Mazer, 2011) discussed the representation of spatiovisual attention in retinotopically organized salience maps that were updated with saccadic movement. Other studies have investigated situations in which multiple objects must be tracked and have found that eye gaze is directed more toward the centroid of the target group than to any single target (Fehd & Seiffert, 2008; Huff, Papenmeier, Jahn, & Hesse, 2010). The relationship between attention and eye movements has

---

C. Berger (✉) · M. Winkels · A. Lischke · J. Höppner  
Department of Psychiatry and Psychotherapy,  
University of Rostock,  
Gehlsheimer Straße 20,  
18147 Rostock, Germany  
e-mail: christoph.berger@med.uni-rostock.de

A. Lischke  
Department of General Psychiatry, University of Heidelberg,  
Vößstraße 2,  
69115 Heidelberg, Germany

also been discussed in some literature reviews (Rayner, 2009; Theeuwes, Belopolsky, & Olivers, 2009). Research into visual behavior is motivated by applied psychological questions; the main topics of such research include the processing of facial expressions in social cognition (Brassen, Gamer, Rose, & Buchel, 2010; Domes et al., 2010; Gamer, Zurowski, & Buchel, 2010), the flexibility in the cognitive control of the oculomotor system analyzed in antisaccade paradigms (Liu et al., 2010; Stigchel, Imants, & Richard Ridderinkhof, 2011), and the experimental research of speech processing (Cho & Thompson, 2010; Myung et al., 2010). Eye movement data must undergo a principal processing pipeline including the following modules. The first module is preprocessing, such as blink detection and filtering. The second module is an event detection of patterns in eye movement, such as fixations and saccades. The third module is the generation of analysis parameters from these events for further statistical analysis. Finally, the fourth module includes different visualization and stimulus overlay techniques of the eye movement data, such as heat maps and gaze plots. In addition to this main data processing queue, there is also the separation of visual stimuli regions of interest (ROIs), in case these subregions have special psychological meanings.

While this processing queue has largely remained consistent since the beginning of eye-tracking research, the analysis paradigm can be further developed: Some of these different approaches to eye movement data processing are mentioned below. There exist various methods for detecting events in the raw data, such as algorithms for blink detection (Morris, Blenkhorn, & Farhan, 2002), fixation detection (Blignaut, 2009; Nystrom & Holmqvist, 2010; Salvucci & Goldberg, 2000; Widdel, 1984), and saccade detection (Fischer, Biscaldi, & Otto, 1993). The analysis of eye movement in response to dynamic visual stimuli and dynamic ROIs (Papenmeier & Huff, 2010; Reimer & Sodhi, 2006) requires the development of methods that can detect the pursuit of dynamic stimuli in the eye movement. Some researchers have focused on the model-based analysis of the eye-tracking data to explain sequences of eye movement patterns with assumptions for underlying cognitive models. (Goldberg & Helfman, 2010; Salvucci & Anderson, 2001)

The present article presents the software GazeAlyze, a MATLAB toolbox for the offline batch analysis of eye-tracking data. GazeAlyze was developed to analyze gaze data for static visual stimuli and to obtain parameters from fixations and saccades for further visualization and statistical analysis. GazeAlyze is suitable for application in research on visual attention in the cognitive processing of visual stimuli and for measurement of volitional control of behavior that is sensitive to fronto-striatal dysfunction, such as the antisaccade task.

GazeAlyze can utilize various algorithms according to choice and is an open-source software: Users can change and extend analysis steps. GazeAlyze is a universal analysis software and includes import capabilities for different eye-tracking data types and, therefore, is useful for data pooling. GazeAlyze also includes features of data quality analysis and artifact control, such as reports of global and trial-wise validation parameters or correction strategies to handle scan path drifts in fixed head settings. GazeAlyze stores the experimental settings in study-related control scripts for easy overview and editing.

Our intent was to design an application that allows the management of the entire offline processing and analysis of eye movement data (containing capabilities such as extensibility, universality, and quality control), as well as batch processing of file sets. GazeAlyze uses modules for blink detection, filtering, and event detection from ILAB 3.6.8 (Gitelman, 2002), a MATLAB toolbox for processing eye movement data. The GazeAlyze data structure can also be loaded into the ILAB software. GazeAlyze mainly expands the capabilities of ILAB by providing the following utilities: the batch processing of data file sets, the analysis and export of detected eye movement events such as fixations and saccades, and visualization capabilities in the form of heat maps. Besides ILAB, other free open-source software tools are available for offline eye movement analysis. There are two freeware tools programmed in C#, which run only on Windows computers. A very well-designed application is OGAMA (Vösskuhler, Nordmeier, Kuchinke, & Jacobs, 2008), which allows the analysis of gaze and mouse movement data together and the recording and storage of data in a Structured Query Language (SQL) database. The other C#-based software is ASTEF (Camilli, Nacchia, Terenzi, & Di Nocera, 2008), an easy-to-use tool for the investigation of fixations that implements the nearest neighbor Index as a measure of the spatial dispersion of the fixation distribution.

## Program description

GazeAlyze can be used as a tool for researchers with no specific programming skills, because the setting of all analysis parameters and all functionalities are controllable from graphical user interfaces (GUIs). All analysis parameters of specific experiments (i.e., definition of trial structure, definition of ROIs, statistics, plots, and heat maps settings) are saved in a single control script. This allows the comfortable handling of the analysis of different experiments.

Eye movement parameters of cognitive processing, visual search, or stimulus impact are estimated separately for ROIs by analyzing fixation, saccade, and movement path parameters. The analysis parameters and covariates are reported for

different trials in a single spread sheet for all data files included in the analysis. Heat maps may be generated over a collection of data files for the visualization of fixation number or duration. Fixation and movement path parameters from experimental periods with determined gaze can be used as correction parameters to minimize drift errors in the eye movement path due to shifts between the camera and head.

GazeAlyze was entirely written in the MATLAB scripting language. All functions are separated from the GUI, exist in separate m-files, and are called through a manager m-file. The GUI allows users to set analysis parameters and to call either a single or a set of processing steps. Through the General Public License (GPL), more experienced users are encouraged to adjust and extend the code for their needs.

The software ILAB 3.6.8 was the starting point in developing GazeAlyze. Because ILAB was written as a MATLAB toolbox, MATLAB was used as the platform for the open-source tool GazeAlyze. Even though MATLAB is only available commercially, it is commonly used in psychophysiological research. Many popular neuroscientific tools have been written for MATLAB; several examples are mentioned below: EEGLab (Delorme & Makeig, 2004), a toolbox for processing electrophysiological data; SPM8 (Wellcome Trust Centre for Neuroimaging, London, U.K.) a toolbox for processing brain imaging data; or the Psychophysics Toolbox (Brainard, 1997), a function set for accurately presenting controlled visual and auditory stimuli.

The software architecture and the data structure of GazeAlyze are similar to those of ILAB, because GazeAlyze uses the functions from ILAB for deblinking, filtering, and event detection, as mentioned in the Graphical User Interface section. Data imported in GazeAlyze contain the same structure as those in ILAB, but GazeAlyze extends this structure with a labeling of data segments into experimental trials and fixed gaze periods to obtain correction parameters. This labeling extends only the variable structure of ILAB and is implemented in the script GA-controller; therefore, imported GazeAlyze data can be loaded into ILAB, and ILAB import scripts can be used in GazeAlyze. The structure of event detection results and segmented data exported to the MATLAB variables and into Excel format is also the same in both toolboxes.

For all other functions besides ILAB-based preprocessing and event detection, GazeAlyze uses different code. The editor for generating ROIs, the ROI variable structure, and all GUI elements differ from ILAB.

### The GazeAlyze workflow

The program modularity of GazeAlyze is oriented toward the steps of the processing queue of eye movement data. This processing queue can be separated into four main parts. The workflow starts with the file format conversion

and import of eye movement data from various eye-tracking systems. For data import, it is also necessary to inform GazeAlyze about the trial structure of the data. Second, preprocessing is needed to remove artifacts (mainly lid blinks) from the data and to filter the noise generated by suboptimal pupil detection by the eye-tracking software. Third, eye movement events must be detected and described—in particular, the location, duration, and onset of fixations and saccades. Fourth, from these event properties and the preprocessed movement path, variables that can be used to operationalize different psychological aspects of the processing of visual stimuli are calculated (see Fig. 1).

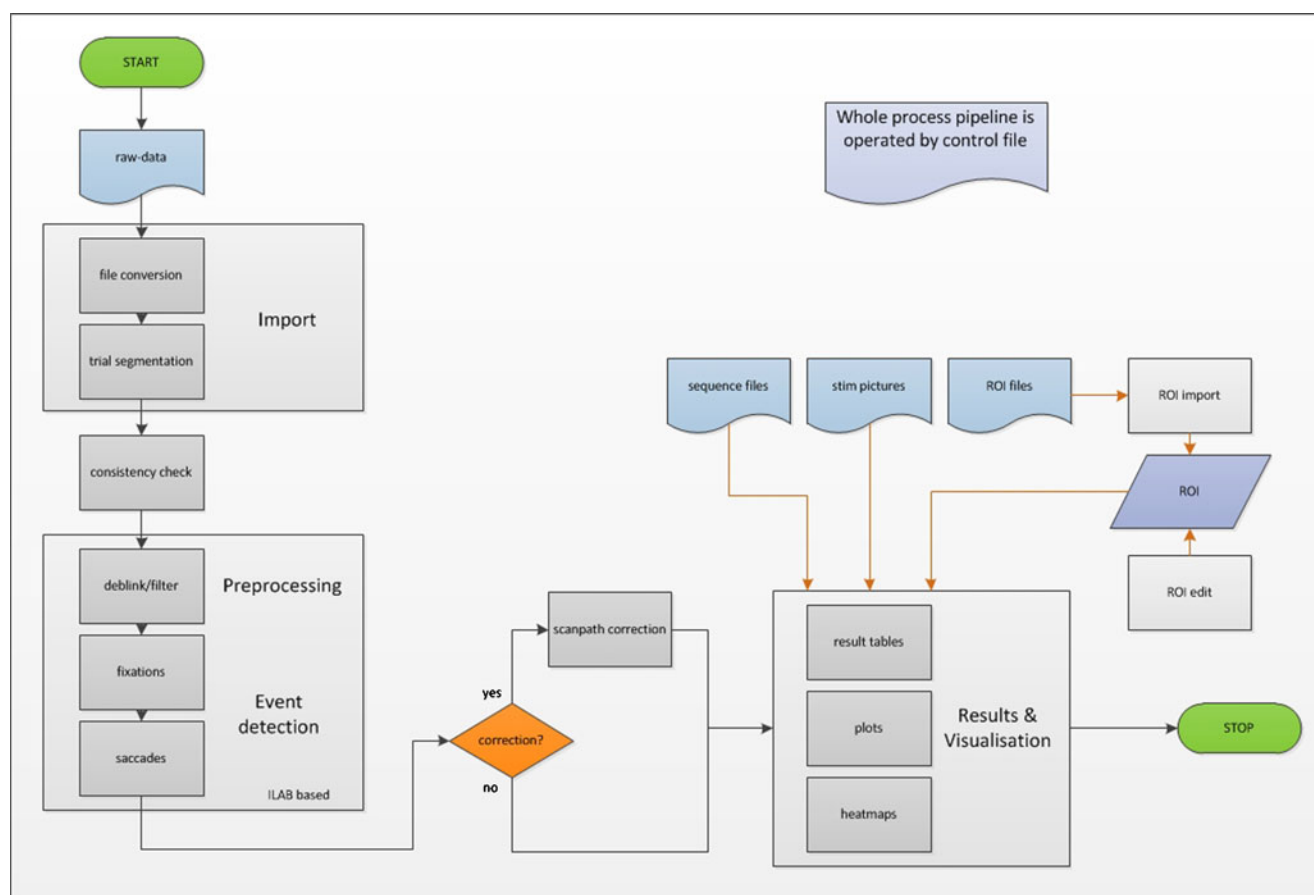
In addition to this, GazeAlyze contains several special functional modules. Eye-tracking systems that track only eye movement, such as fixed head or head-mounted systems, cannot separate these data from the movement of the head and are susceptible to a shift between the camera and head during the experiment, thereby producing deficient results. Therefore, in GazeAlyze, a correction function is implemented that generates correction parameters from periods in the experiment, during which the participant is asked to fixate on a special region. These correction parameters, such as the spatial median of movement path or fixation location, can be used to shift the eye-tracking data prior to the generation of parameters for further statistical analysis.

Plots can be generated of the movement path and the spatiotemporal fixation distribution for a quick data inspection after event detection. In ILAB, plots were originally available only in the MATLAB environment. In GazeAlyze, plots are exported into images in the .jpg file format. Another visualization function of GazeAlyze is the generation of heat maps of the spatial distribution of fixation count or duration. These heat maps can be generated with various color maps and scaling on basis of selectable participant and stimulus samples.

The managing of special ROIs inside the visual stimuli can be done via the ROI editor or the ROI import option. ROIs are important for marking regions with enclosed visual importance and separating movement analysis for these regions. It is possible to generate rectangular, elliptical, and polygonal ROIs with GazeAlyze; these ROIs can be imported from external ROI lists or obtained directly from pictures. Multiple ROIs can be tagged together and rotated by any angle.

### The application design

The program structure is described in Fig. 2. The handling of GazeAlyze is controlled by the GUI, where the user can set all analysis parameters and perform the processing of the eye movement data. The analysis parameters are saved in an experiment-specific control file. During a GazeAlyze



**Fig. 1** Flow chart of GazeAlyze processing pipeline

session, this control file is loaded from the hard disk, and all variables are stored in a structure of the main GUI panel. Via the GUI, the user has access to these experiment parameters. It is possible to edit them and to store them back into the control file on the hard disk.

A call of a processing task from the GUI passes the structure of the experiment control variables to the controller script with the order to start a particular required processing task. The controller then calls the particular processing scripts designated to perform the task operations (i.e. importing, preprocessing, plotting, heat map generating, and analysis). It is possible to process the data from any point in the workflow without the need to run prior steps again (e.g., to run different visualization steps after the data are imported and preprocessed without preprocessing or importing the data again). It is also possible to batch task operations to perform a set of functions with the data. If a processing step is performed, the result will be stored in a subset of the variable structure in the GazeAlyze eye movement data file, as mentioned below:

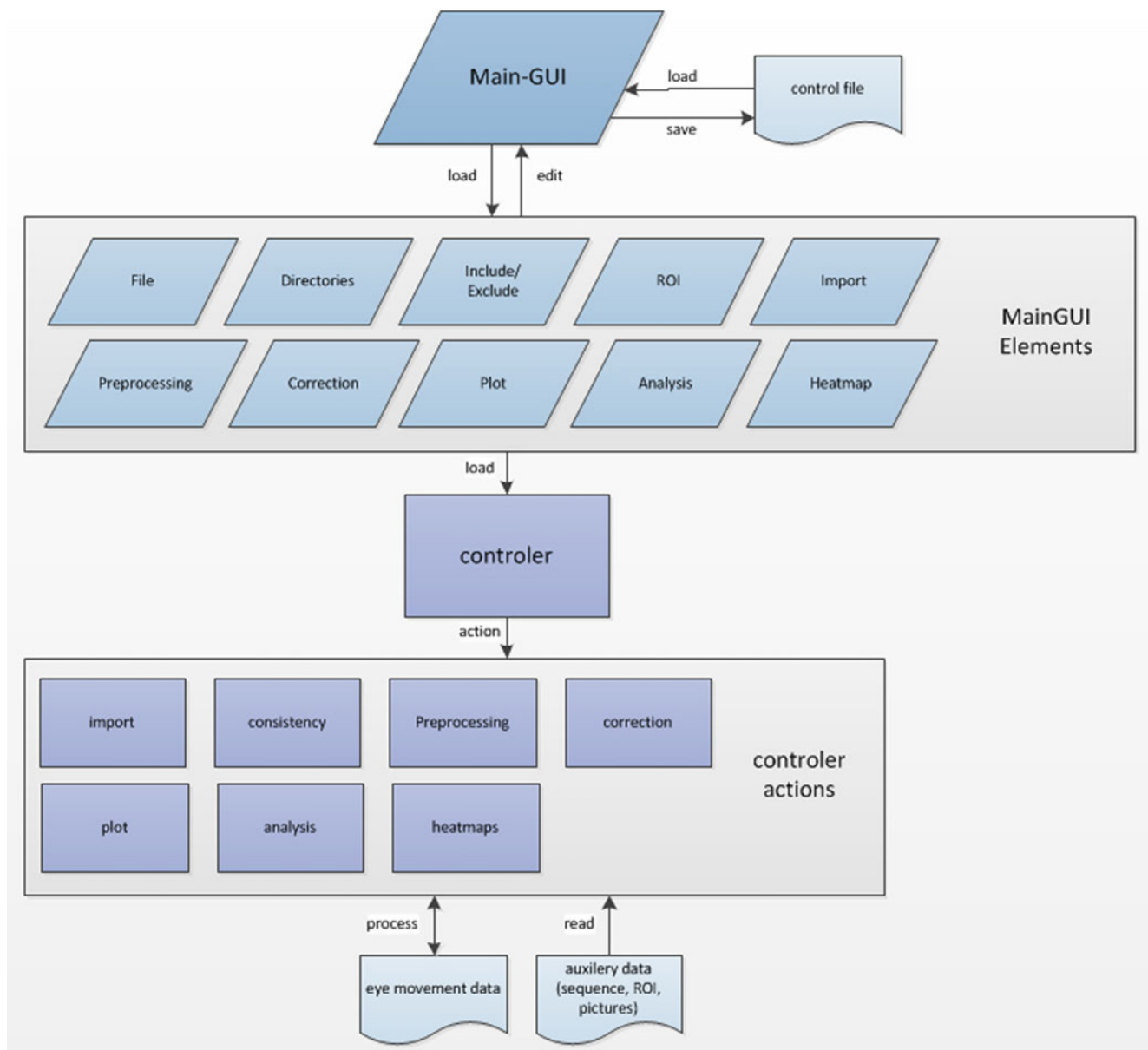
- **import:** all data from the external file for further conversion

- **ILAB:** includes the variable structure needed as input for deblinking, filtering, and event detection with ILAB
- **corr\_par:** result from correction parameter calculation
- **quality:** report of global data quality (signal-to-noise ratio and amount of blinks)
- **eyemove, fixvar, saccvar:** results from preprocessing and event detection.

### The graphical user interface

**Main panel** After typing "gazeaylze" into the MATLAB command window, the main panel appears, which contains three main parts. At the top, there is a menu for accessing all functions of GazeAlyze and some quick-start buttons for directly accessing the most frequently used functions. On the left, a list box contains an overview of all actual variable settings controlling a specific experiment. At the bottom of the main panel, a list contains the logged information about all actions performed since the last time the experiment control script was loaded (see Fig. 3).

**Include/exclude settings menu item** under this menu, there are three panels for the selection of a subset of experimental



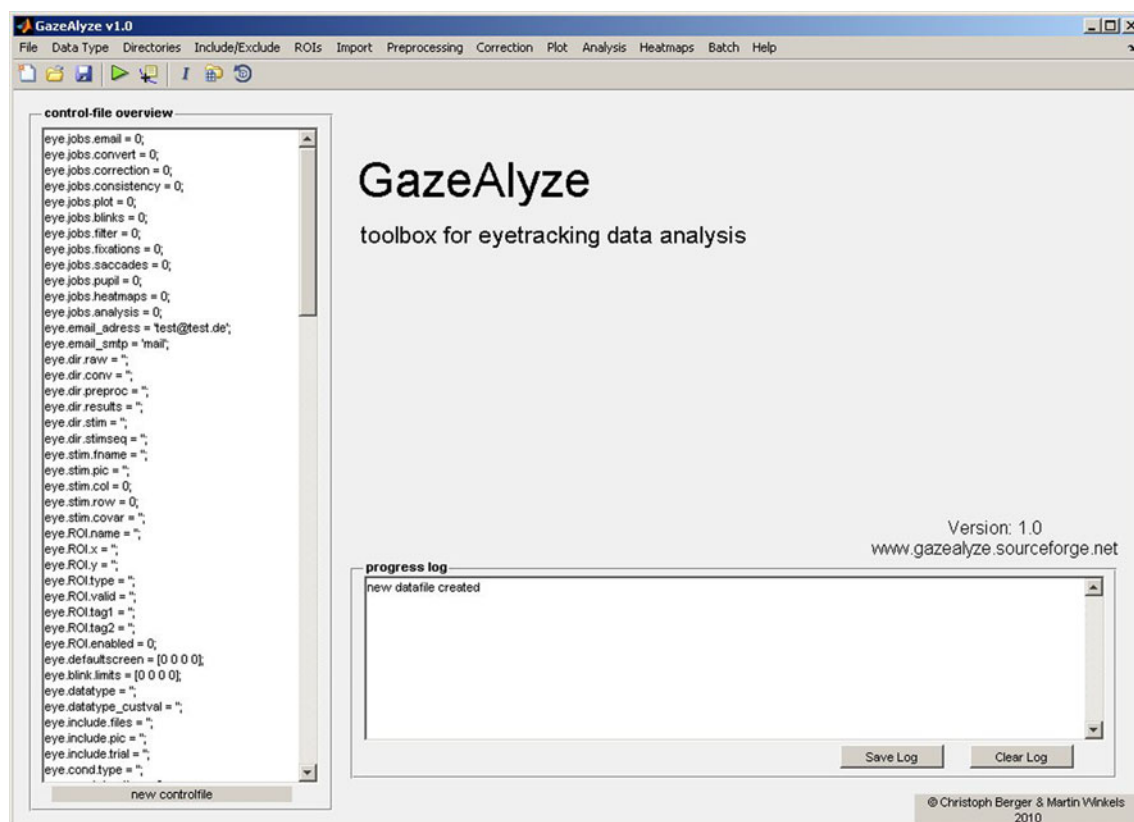
**Fig. 2** Structogram of GazeAlyze software architecture

items to perform the analysis process. With the first panel, a subset of files can be selected for the analysis. In this panel, the user can easily change between the two data input directories (the raw data file folder and the converted data file folder), because each directory can contain files from different participants. The raw data directory is the starting point for the import of external data files to convert the eye movement data into the GazeAlyze variable structure. The converted data file folder contains the GazeAlyze variable structure and is the starting point for any following stage of data processing. With the second panel a subset of picture stimuli can be selected for further processing such that only the trials affecting these pictures will be analyzed. With the

last panel, it is possible to include only some determined trials in the analysis, while excluding all other trials. In experiments with a large amount of trials and visual stimuli, this restriction avoids search operations through unnecessary data files or can reduce the analysis report to the relevant data.

*Preprocessing menu item* The deblinking and filtering of the data and also the event detection are based on the software ILAB 3.6.8, which has been slightly adapted to the needs of GazeAlyze in terms of data organization and communication between GazeAlyze functions, as mentioned below. With the exception of progress bars, GUI





**Fig. 3** Main panel of the GazeAlyze GUI

elements of ILAB are not shown. All of the advanced settings belonging to the preprocessing steps based on ILAB can be achieved by direct editing of a MATLAB script corresponding to this menu item. Furthermore, the consistency check can be performed separately here. The consistency check is a new GazeAlyze feature and mainly determines whether an import step was performed correctly, there are data in the file, and the marker timing was correct. If an error is recognized during import, the affected files are stored separately. Because GazeAlyze uses a global struct to communicate between different scripts and functions, the ILAB analysis parameter are stored in this global struct. ILAB offers a velocity- and a dispersion-based algorithm to detect fixation events, which were transferred in their original functionality into GazeAlyze. The event detection output does not differ between ILAB and GazeAlyze.

*Trial import settings panel* In this panel, the user can comfortably adjust settings for the trial structure to be segmented from the raw data stream (see Fig. 4). In the upper part of the settings for the trial, timing marker information can be adjusted. It is possible to specify trials with a fixed duration relative to a single marker or to specify trials limited by start and stop markers. There can be various markers in the data stream, and the marker

strings that label the trials can be specified. It is possible to analyze a block set of trials separately that are labeled with only one marker per block.

In addition to the trial markers, it is often important to evaluate fixed-gaze periods in the raw data stream. In these fixed-gaze periods, the gaze is often focused on a determined region, such as a fixation cross, and the deviance from this region in the eye movement data can be used for measurements of data quality. The marker of these fixed-gaze periods can be determined separately in this panel.

*Directories panel* In the processing pipeline of eye-tracking data analysis, various data files are involved. Quality structured file handling is especially important for the batch processing of multiple data files. In the "directories" panel, all of the relevant file locations can be determined. The eye movement data processed by GazeAlyze are stored in their original file format in the directory for raw eye movement data. All other stages of processed eye movement data are stored in files containing the GazeAlyze variable structure in the directory for converted and further processed eye movement data. Furthermore, there is a directory for the export of data from various analysis steps, which contains subfolders for correction parameters in text file format, the

**Fig. 4** Trial import settings panel

variable worksheet for further statistical analysis, and visualized data such as heat maps and fixations or scan path plots.

Two other directories contain the visual stimuli that were presented in the experiment and the files that contain the information about the stimulus sequences. These sequence files determine which stimulus belongs to which trial and may also contain additional trial-specific covariates such as response information, which can also be reported together with eye movement parameters for further statistical analysis.

**Sequence file panel** In a separate panel accessible from the directory panel, the user can determine the structure of the sequence files. Sequence files are typically text files with header information such as the experiment name, the participant identifier, and the experiment date. Trial data are written in a table under each row belonging to a separate trial. The user must determine at which row to begin the trial table, which column contains the name of the

presented picture, and which columns should be reported for further statistical analysis. The other information needed is the format of the picture names.

**Correction setting panel** As described above, the fixed-gaze period can be analyzed to receive correction parameters for quality control and possible offline manipulation of the data. When the head is fixed and head-mounted eye-tracking cameras are used, as in the scanner during functional imaging, the camera position relative to the head may move. For example, the eyetracker belonging to the presentation system “Visua Stim” (Resonance Technology, CA) is mounted on a visor that the participant wears on the eyes during the scan and is only fixed on the participants head with a rubber strap. If the visor slips after the calibration period, the complete eye movement data will be shifted from the original position.

Therefore, in this section, the user can obtain information about a data shift and use this information to correct

the eye movement data before the statistical analysis is conducted. The basis of the data shift measurement is the fixed-gaze periods in the experiment when the region that the participant looked at was determined in a special task, such as looking at a fixation cross. The distance between gaze location measures and the location of this region is used to correct the data shift.

The user can select between different strategies to receive the gaze location measures. It is possible to select between the location of the longest fixation, the first or last fixation, the mean of all fixations, or the median of the scan path. It is also possible to mark the subsequent trial as bad, if the fixation period ends with a saccade. In this fixed-gaze period, it must be ensured that the gaze that will be analyzed to obtain the correction parameter is really directed toward the fixation object. GazeAlyze uses markers for the segmentation of the fixed-gaze periods from the entire data set. When a fixation object is shown over the entire duration of the fixed-gaze period and no more objects are presented, all methods will be good estimates for the amount of shift in the measured gaze. In scenarios in which the presentation time of the fixation object is shorter than the duration of the entire segment, it should be useful to have more focus on the analysis of fixations such as the mean or the last fixation than on the median of the scan path from the entire segment to calculate the shift between gaze and origin of the fixation object. It is also possible to evaluate fixed-gaze periods belonging to experimental trials and to determine the duration of this period independently of the trial duration.

Correction parameters are stored in the GazeAlyze variable structure in MATLAB format and as a table in text files. In this table, the first two columns refer to the first and the last trial to which the correction parameter applies, and the third and fourth columns list the shift in the  $x$  and  $y$  directions that must be subtracted from the eye movement data.

**Plot setting panel** Data plots can be used to visualize the eye movement data and events for each trial. In this panel, it is possible to select the plot of the fixations, the scan path of experimental or fixed-gaze trials, and the median of the scan path of the fixed-gaze trials. It's also possible to plot data based on scan path shift corrections and to plot only a certain amount of time of the trials.

**ROI editor panel** In this panel, GazeAlyze offers the opportunity to create and edit ROIs. ROIs are areas in the picture in which analysis steps were performed separately. ROIs can belong to a special visual stimulus or to all stimuli that were presented during the experiment. ROIs can be simple geometrical forms (rectangles and ellipses),

or they can be drawn in an arbitrary outline. It is also possible to rotate ROIs to various angles. An ROI is determined by the combination of its name and the stimulus to which it belongs. GazeAlyze also offers the opportunity to group different ROIs using the flag labels from the ROI description. Furthermore, it is also possible to analyze eye movement parameters for each ROI separately or for a group of ROIs with a particular flag name.

There are two ways to create an ROI with this panel (see Fig. 5). One way is to draw an ROI with the mouse on the field on the right in this panel. Alternatively, the position and the expansion of the ROI can be inserted into the fields on the left.

**ROI import from the file panel** In addition to creating new ROIs with the ROI editor panel, GazeAlyze can import an entire ROI table from an external text file. This ROI table typically contains multiple ROI groups with the same name but different positions, sizes, and picture origins. GazeAlyze can import tables where a group of columns contains the parameter of a group of ROIs. Starting from a row below heading information, each row contains the name of the original picture stimulus and the position and size of the ROI in this picture. It is also possible to import only a subset of the pictures in this list.

**Analysis setting panel** In addition to the batch processing of eye movement data, the ROI handling, and the heat map visualization, the main feature of GazeAlyze is that it is easy to obtain, calculate, and aggregate eye movement event parameters for further statistical analysis. The corresponding setting can be adjusted in this panel (see Fig. 6).

Future development of the analysis features of GazeAlyze is planned, because many extensions are possible. Currently, the following parameter can be exported from the data:

Parameters of cognitive processing:

- Absolute cumulative and mean fixation durations and numbers, relative fixation durations, and numbers as fixation parameter from an ROI, relative to the fixation parameters from another ROI or from the entire stimulus
- The fixation/saccade ratio

Parameters of visual search:

- Spatial extension of the scan path

Parameters of stimulus impact:

- Region of initial fixation
- Fixation order



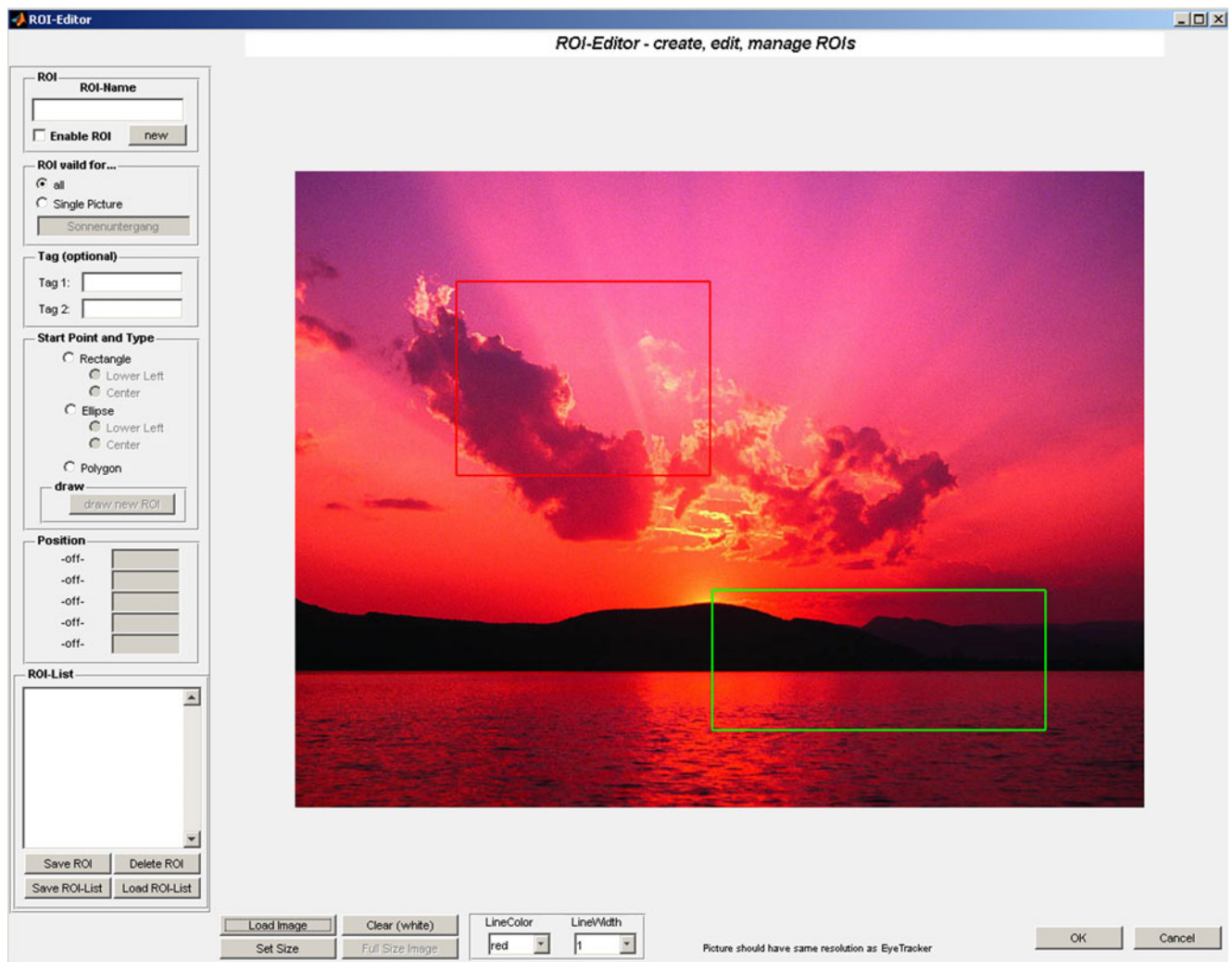


Fig. 5 ROI editor panel

- First saccade direction
- First saccade latency

In addition to these analysis parameters, several other settings relevant for further statistical analysis can be adjusted. The user can select the following actions:

- Report global data quality in the form of the ratio between data points and detected blink points and the signal-to-noise ratio
- Correct the data prior to analysis with the scan path shift correction parameters
- Report a trial validation measure. Trials are marked as invalid, if there is no accessible correction parameter because of noisy data and if the fixed-gaze period ends with a saccade, whether this should be reported as invalid too
- Report trial covariates from the sequence file
- Report the entire trial duration, especially from trials with different durations
- Sort the trials by the trial ore condition number in ascending order

It is also possible to weight or normalize the analysis parameters, relative to the spatial extent of the ROI, by multiplying them with a normalization factor. This feature can be useful for allowing a better comparison of analysis parameter from ROIs with different spatial extensions in some experimental settings. For example, with two photographs of social interaction scenes, both pictures have the same number of objects shown; however, the subset of objects that is more closely relevant to the social interaction differs between the pictures. ROIs that are drawn around these object subsets have different spatial extents. Object

Fig. 6 Analysis settings panel

count and complexity of visual content should be correlated with eye movement parameters of visual attention, assuming that the perception of more complex picture content requires more sensory effort for a person to perceive and orient him-/herself to the picture regardless of further picture properties, such as the social meaning of the object constellation. To compare ROIs of different relative complexity, it could be useful to take into account the spatial extent of the ROIs as operationalization of complexity. Weighting analysis parameters relative to the spatial extent removes the relative complexity of ROI objects from further statistical analysis of attention caused by picture content directly related to social cognition.

However, in scenarios in which the relative complexity does not differ between ROIs, this spatial weighting should not be applied and can lead to artifacts. For example, if two equally sized objects were shown in a picture at different depths, the closer object would occupy a larger area on the computer monitor. ROIs placed around these two objects to

distinguish them from other picture content will differ in size, since the closer object will define a larger ROI than will the farther object. In this case, the complexity between the two ROIs does not differ, and the distribution of gaze for the close and the far objects should be valued in the same manner without weighting them for spatial extent.

As a normalization factor, the following equation is used:

$$1 - \text{area\_ROI} / \text{area\_screen}.$$

As the spatial extent of the ROI decreases, the relevance of the related analysis parameters increases.

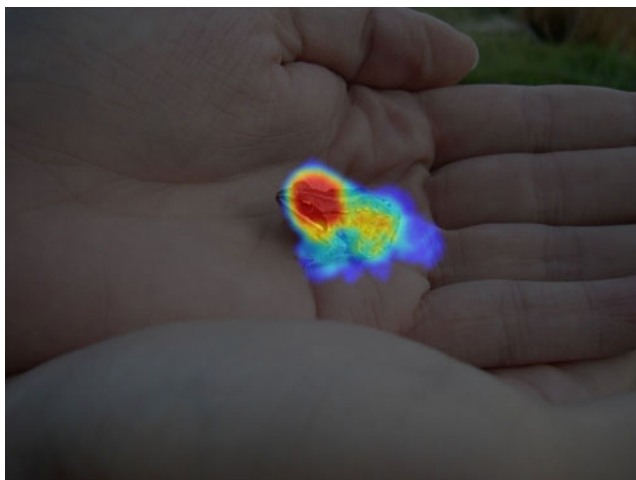
It is possible to obtain all of these parameters from a temporal segment and not from the entire trial. The start and the end of this analysis period inside the trial can be specified relative to the trial start or end as a percentage of the trial duration or in milliseconds.

**Batch job setting panel** All of the processing steps can be performed independently via the corresponding menu items or can be processed in immediate succession with the GazeAlyze batch job feature. The jobs included in the batch processing are selectable under this panel. It is also possible to send an e-mail when the batched job execution is ready.

**General settings panel for heat maps** GazeAlyze main visualization feature is the ability to present eye movement data as heat maps with variable layout settings. It is possible to visualize the amount and dispersion of fixation durations or fixation numbers. These data plots can be made as the visualization of a single participant or as the mean of these parameters over a selectable group of participants. Figure 7 show examples of visualization of fixation number as heat maps generated with GazeAlyze.

The workflow of heat map creation is outlined in Fig. 8. A group of three single fixations, each from a different participant (a), is grouped in (b). A circle of arbitrary radius is drawn around the fixation origins (c). These circles overlay with each other in some areas. A color map (d) can be defined to represent the amount of overlaying circle areas (e). In this example, the number of fixations is represented in the color map, but it is also possible to colorize the amount of added fixation duration with GazeAlyze. The part of the stimulus picture that is not covered by fixations can be overlaid with a unique color. This unique color and the colored circle areas can have some transparency (f). Finally, the heat map is smoothed to blur the edges of the different colors (g), and the stimulus picture is plotted into an image file with the screen pixel size (h) in the position in which it was shown during the eye movement recording.

In the general heat maps settings panel (see Fig. 9), a subset of files can be selected to be included in a heat map



**Fig. 7** Heat map example of the number of fixations

group. This selection can be made by direct input or by importing the filenames from a .csv file. Corrected or uncorrected data can be chosen to be visualized. In a second step, the user should select the global color settings for the heat map. The locations of the visualized data points are first plotted on an empty screen, and this plot is then laid over the stimulus picture that was presented during the experiment. For this overlay of the plotted data, the following adjustments can be set on this panel. First, the background color, as the color of the screen where no stimulus was presented, can be adjusted. Second, the basic overlay color, which overlays the entire stimulus where no fixations occurred, can be adjusted. Third, the data points, which are drawn as circular areas with a certain radius, can be adjusted. Fourth, the transparency of the data overlay can be adjusted, so that the stimulus picture is also visible under the heat map. Fifth, the filter strength for smoothing the edges of the overlaid data point circle areas, as well as the position of the stimulus picture on the screen, can also be adjusted.

**Heat map color/gradient settings panel** In this panel, the different colors used to visualize the different amounts of data focused on a region of the stimulus can be set. The color map can be chosen from a selection of preinstalled color maps, and each color can also be manually adjusted. The range and the gradient of the color map—that is, the number of data points that are focused to a region and presented with a single color—can also be set here. It is possible to choose whether the fixation duration or the fixation numbers must be plotted (see Fig. 10).

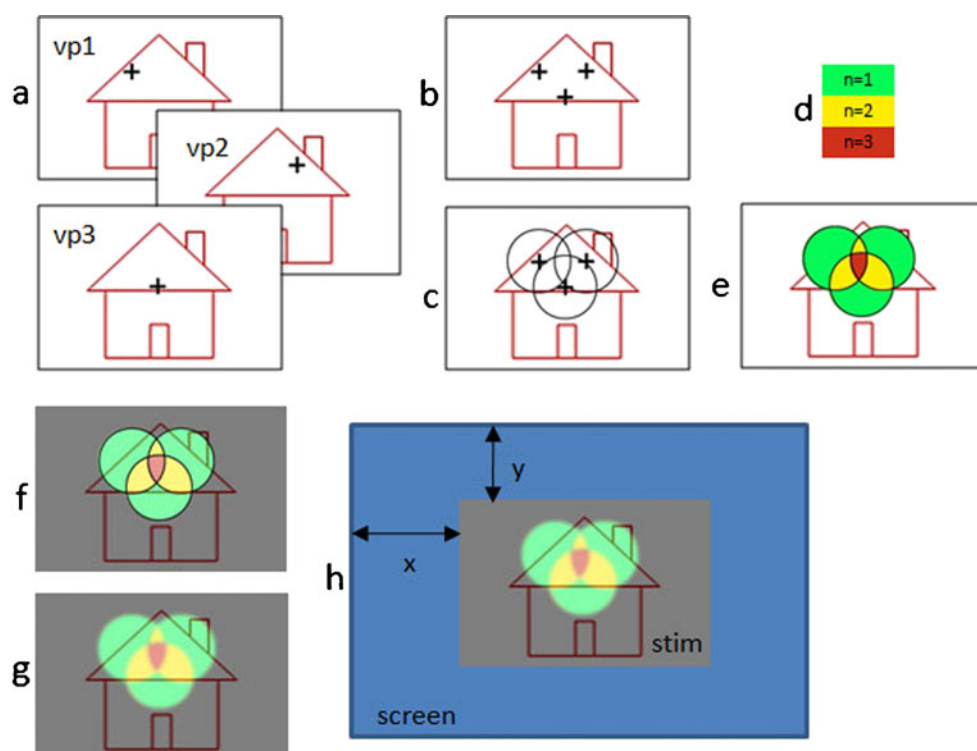
## System requirements

GazeAlyze requires less than 6 MB of free hard disk space. The program will run on any operating system supporting MATLAB (MathWorks Inc.) or later. The MATLAB image-processing toolbox (commercially available from MathWorks), is needed for the use of the visualization capabilities of GazeAlyze, such as the creation of plots and heat maps. For any other function in GazeAlyze, no special MATLAB toolbox is required. GazeAlyze has been tested in its current version 1.0.1 on MATLAB 2007a on Mac OS X, version 10.5 (Apple Inc., CA), Ubuntu 8.10 Linux, Microsoft Windows XP and Microsoft Windows 7 (Microsoft Inc., WA).

## Installation procedure

The installation routine involves downloading and unzipping the GazeAlyze package. GazeAlyze is then launched by typing “gazealyze” into the MATLAB command

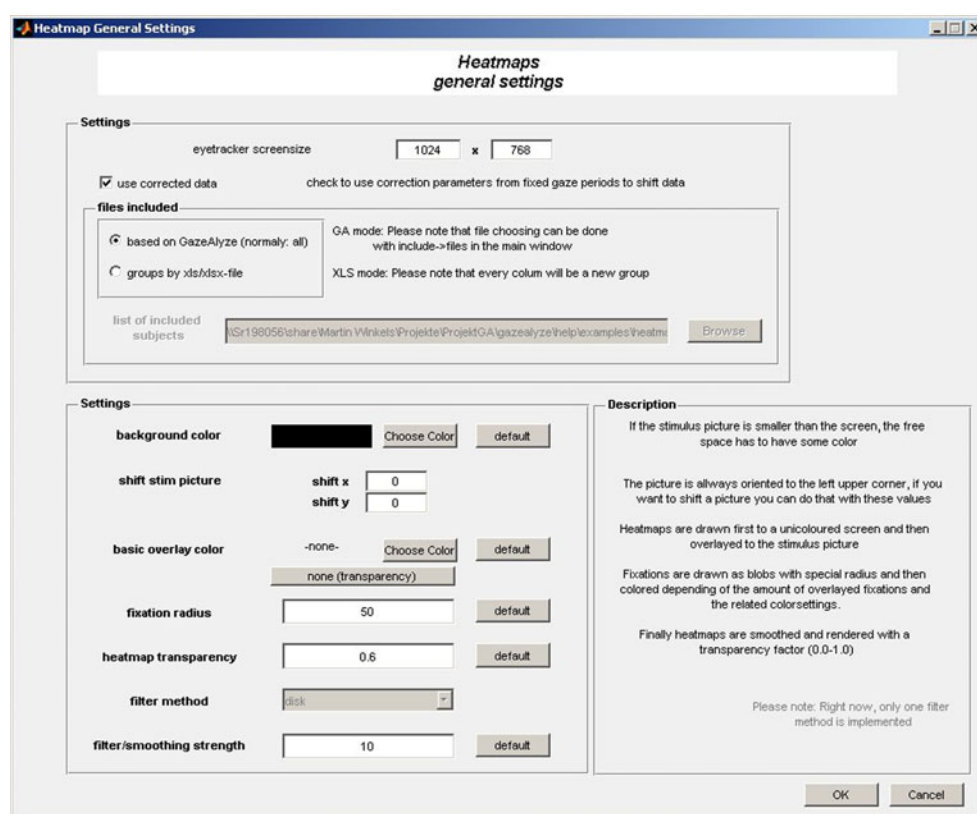
**Fig. 8** Heat map generating workflow



window. When starting GazeAlyze for the first time, it is necessary to start from the GazeAlyze directory. Then

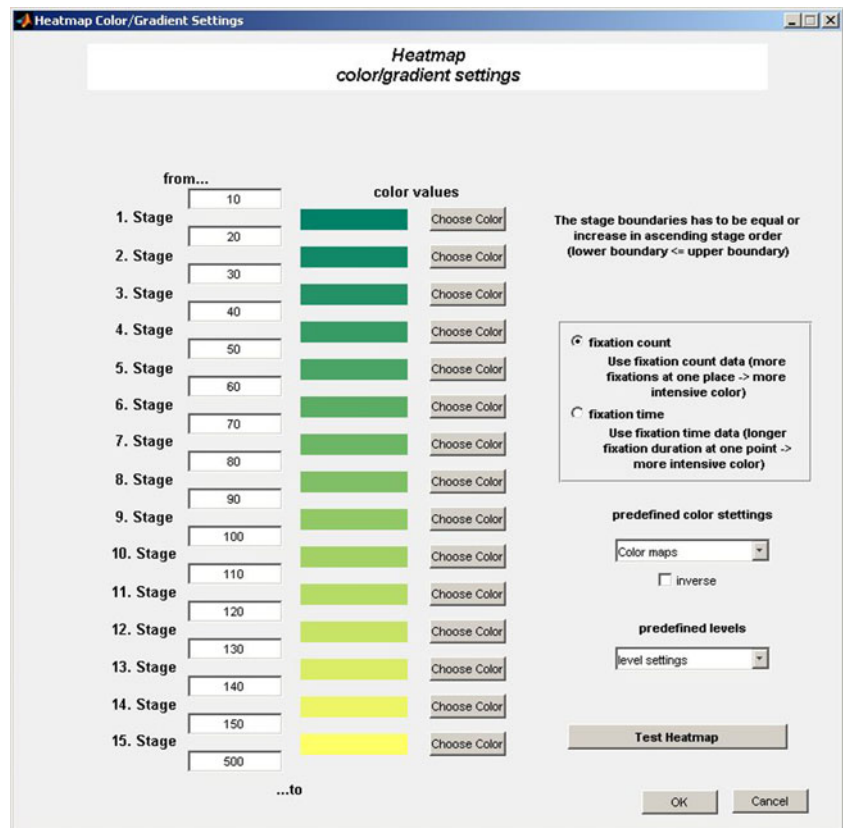
GazeAlyze can be started from any other working directory. After closing GazeAlyze, all subfolders will be automati-

**Fig. 9** General settings panel for heat maps





**Fig. 10** Heat map color/gradient settings panel



cally removed from the search path to prevent interferences with other scripts in the search path.

#### Availability

GazeAlyze is distributed free of charge under the terms of the GNU General Public License as published by the Free Software Foundation. Under these terms of the GNU License, users are free to redistribute and modify the software. GazeAlyze is freely available for download at <http://gazealyze.sourceforge.net>.

#### Samples of typical system or program runs and comparison of correction methods

GazeAlyze is the standard tool in our laboratory for the analysis of eye movement data. The GazeAlyze software package also contains some sample data. These sample data include raw data files in text file format as Viewpoint eye movement data and the regarding control script with the trial marker settings. The performance of the different correction strategies was tested in an experiment to correct the eye movement data for scan path drifts induced by the shifting of the eyetracker camera after the calibration procedure, such as the use of the camera as a part of the

VisuaStim system (Resonance Technology Inc., USA) in functional imaging settings or in other head-mounted systems.

#### Materials and methods

**Participants** Sixty-four healthy adults (age:  $M = 25.83$  years,  $SD = 3.04$ ), who were enrolled in an ongoing study on emotion recognition abilities in males, participated in the validation study. All participants provided written informed consent and were paid for participation. The study was carried out in accordance with the Declaration of Helsinki and was approved by the ethical committee of the University of Rostock.

**Emotional recognition task** A dynamic emotion recognition task with neutral expressions that gradually changed into emotional expressions was used (Domes et al., 2008).

For this task, six male and six female individuals, each showing a neutral, sad, angry, fearful, and happy expression, were chosen from the FACES database (Ebner, Riediger & Lindenberger, 2010) Using Winmorph 3.01 (<http://www.debugmode.com/winnmorph/>), each emotional expression was morphed by steps of in 5% increments from 0% to 100% of intensity. For each individual, four sets



**Table 1** Comparison of corrected versus uncorrected data and the number of fixations involving the eye region of facial expressions

| Correction Method | Mean Difference to No Correction | Standard Error | Significance | 95% Confidence Intervall |             |
|-------------------|----------------------------------|----------------|--------------|--------------------------|-------------|
|                   |                                  |                |              | Lower Bound              | Upper Bound |
| lastfix           | 20.71*                           | 4.91           | .0001        | -33.68                   | -7.74       |
| mean              | 16.37*                           | 4.91           | .005         | -29.34                   | -3.40       |
| median            | 22.48**                          | 4.91           | .0001        | -35.45                   | -9.50       |

Note. ANOVA,  $N = 256$ , (64 probands, 4 events)

of emotional expressions (happy, sad, angry, and fearful) were created, each containing 21 images. Each image of a set of expressions was presented for 800 ms, ranging from 0% to 100% of intensity. Each set of 21 pictures was preceded by a presentation of a fixation cross in the middle of the screen for 1 s. Each image was presented on a 20-in. computer screen (screen size,  $30.6 \times 40.8$  cm; resolution,  $1,024 \times 768$  pixels) against a gray background at a viewing distance of 55 cm. The mean viewing angle for each image was  $16.93^\circ$  in the vertical direction and  $23.33^\circ$  in the horizontal direction.

The participants were instructed to press a stop button as soon as they became aware of the emotional expression. After stopping the presentation, a forced choice between the four emotions (happy, sad, angry, and fearful) was asked for.

**Eye tracking** Participants' eye movements were recorded with a remote infrared eyetracker (PC-60 Head Fixed; Arrington Research Inc.; Scottsdale, AZ) to record gaze changes during emotion recognition. Raw data were collected at a 60-Hz sampling rate with a spatial resolution of approximately  $0.15^\circ$  for tracking resolution and  $0.25^\circ$ – $1.0^\circ$  for gaze position accuracy. Eye movement data were analyzed with GazeAlyze. After filtering and deblinking of the raw data, fixations were coded with dispersion algorithm (Widdel, 1984) when the gaze remained for at least 100 ms within an area with a diameter of  $1^\circ$ . Image-specific templates were used to differentiate between fixations to distinct facial ROIs (eyes, mouth, forehead, and cheeks). For statistical testing, the number of fixations on the eye region was calculated. Regarding eye movement data for the correction strategies, the data recorded during the time of presentation of the fixation cross were used.

**Statistical analysis** Since the focus of the present study was on the eye-tracking data, data pertaining to the emotion recognition task will be presented elsewhere. The effect of three different scan path drift compensation methods (median of scan path, last fixation, mean fixation) to the mean number of fixations counted in the eye region over all emotions was tested with a one-way analysis of variance (ANOVA). The significance level for analysis was set at  $p <$

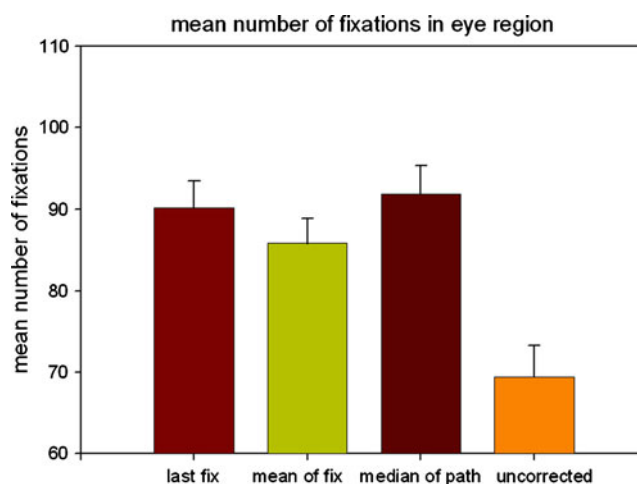
.05. All statistical analyses were performed using SPSS 17 (SPSS Inc., Chicago, IL).

## Results

The one-way ANOVA showed that there were significantly more fixations counted inside the eye region for all correction methods, as compared with no correction for the scan path drift,  $F(3, 1020) = 8.73$ ,  $p < .00001$ . Post hoc  $t$ -tests showed no evidence for a significant difference in the performance of correction methods (see Table 1 and Fig. 11).

## Discussion

This evaluation study showed that in experiments with fixed head settings, the amount of fixation parameters of attention into regions of high saliency can be improved via correction methods for scan path drifts. In interpreting the results as a successful correction of scan path drifts, there are two assumptions that have to be discussed in greater detail—namely, first, the correction methods do not create any artifacts



**Fig. 11** Mean number of fixations in the eye region of facial expressions

causing the results, and second, the participants originally looked at the ROIs and, during the presentation of the fixation cross, they originally looked at the fixation cross. There are four arguments to support these assumptions, as discussed below. The first two arguments concern the artifact-free application of the correction methods.

First, the correction method will be applied to the fixations at a time when they are already created. Applying or even calculating correction parameters does not change the overall number or duration of fixations. The calculation of fixations is the main part of the event detection. Furthermore, eye movement events, such as saccades and fixations, must be created first, and corrections will only be applied to them.

Second, the data used for the calculation of the correction parameter are independent of the data for event detection of trial fixations. Only the eye movement data from gazing at the fixation cross are used to calculate correction parameters and are measured at a different experimental time than the eye movement data from the trial. The origins of ROIs are independent of the origin of the fixation cross. We measured eye movement data during a simple procedure of one point calibration (gaze at a fixation cross) and calculated the shift of measured eye movement data to the origin of this calibration point. A similar calibration procedure was performed with the eyetracker recording software at the beginning of the experiment. This strategy was used to remap the gaze toward the original coordinates again. It is also possible to use more salient stimuli than a simple fixation cross to guide the gaze closer to the calibration object.

The last two arguments address the origin of the regions the participants looked at during the sampling of the eye movement data. First, during the fixation period, the participants were asked to look at the fixation cross. During the trial, they did not explicitly have the task of looking at the eyes but were told to look at the picture on the screen. Regarding the eye movement during the presentation of static faces, it was shown that the eyes are especially salient and are the main point of fixation (Emery, 2000; Itier & Batty, 2009). Second, no other objects were near the eye ROIs that might have attracted the gaze at all. For both reasons, it is most likely that participants do fixate directly on the eye region and that it is a more appropriate mapping of the participants gaze that fixations originally counted as "next to the eyes" after the correction are counted as "on the eyes."

According to the data, there were no significant differences between the methods used for the calculation of the correction parameter. In the present experiment, there was a fixation cross shown over the entire duration of this fixed-gaze period, and no more objects were presented. For scenarios like this, all methods are good estimates of the amount of shift in the measured gaze. However, the fixation-based correction

methods may have some advantages over the median of the scan path when the fixation object occurs only in a part of the fixed-gaze data segment. Furthermore, for convenience, GazeAlyze also offers the opportunity to use fixation-based correction for such situations.

## Conclusion

GazeAlyze is currently in use in research laboratories at the University of Rostock and at the Free University of Berlin. It has proven to be useful in psychophysiological experiments of the analysis of visual attention in the research of social cognition. Furthermore, it has also been developed for the analysis of reflex suppression in antisaccade tasks. The easy handling of all processing settings through the GUIs, the capabilities of batch processing, and the wide range of analysis and visualization features allow the researcher to shorten the analysis time and provide all tools for handling the complexity of eye movement data analysis.

GazeAlyze offers a single software tool for managing all offline processing steps involved in the preparation of eye movement data for further statistical analysis: (from trial extraction during raw data file import over event detection to the aggregation of event parameters into a single table). Because all settings are separate from the software in experimental control files, setting information sharing and data handling are further facilitated.

GazeAlyze is being actively maintained and developed. New functionality, such as the extension of analysis features, (e.g., pupillometry) and the implementation of other kind of event detection algorithms, such as Glissade detection algorithms (Nystrom & Holmqvist, 2010), is expected in future versions.

**Acknowledgements** We would like to thank Evelyn Schnapka and Lars Schulze for their valuable suggestions and support during the development of the features of GazeAlyze.

**Conflict of interest statement** None declared.

## References

- Awh, E., Armstrong, K. M., & Moore, T. (2006). Visual and oculomotor selection: Links, causes and implications for spatial attention. *Trends in Cognitive Sciences*, 10, 124–130.
- Blignaut, P. (2009). Fixation identification: The optimum threshold for a dispersion algorithm. *Attention, Perception, & Psychophysics*, 71, 881–895.
- Brainard, D. H. (1997). The Psychophysics Toolbox. *Spatial Vision*, 10, 433–436.
- Brassen, S., Gamer, M., Rose, M., & Buchel, C. (2010). The influence of directed covert attention on emotional face processing. *NeuroImage*, 50, 545–551.

- Camilli, M., Nacchia, R., Terenzi, M., & Di Nocera, F. (2008). ASTEF: A simple tool for examining fixations. *Behavior Research Methods*, 40, 373–382.
- Cho, S., & Thompson, C. K. (2010). What goes wrong during passive sentence production in agrammatic aphasia: An eyetracking study. *Aphasiology*, 24, 1576–1592.
- Delorme, A., & Makeig, S. (2004). EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134, 9–21.
- Deubel, H., & Schneider, W. X. (1996). Saccade target selection and object recognition: Evidence for a common attentional mechanism. *Vision Research*, 36, 1827–1837.
- Domes, G., Czeschnek, D., Weidler, F., Berger, C., Fast, K., & Herpertz, S. C. (2008). Recognition of facial affect in borderline personality disorder. *Journal of Personality Disorders*, 22, 135–147.
- Domes, G., Lischke, A., Berger, C., Grossmann, A., Hauenstein, K., Heinrichs, M., et al. (2010). Effects of intranasal oxytocin on emotional face processing in women. *Psychoneuroendocrinology*, 35, 83–93.
- Ebner, N. C., Riediger, M., & Lindenberger, U. (2010). FACES—a database of facial expressions in young, middle-aged, and older women and men: Development and validation. *Behavior Research Methods*, 42, 351–362.
- Emery, N. J. (2000). The eyes have it: The neuroethology, function and evolution of social gaze. *Neuroscience and Biobehavioral Reviews*, 24, 581–604.
- Fehd, H. M., & Seiffert, A. E. (2008). Eye movements during multiple object tracking: Where do participants look? *Cognition*, 108, 201–209.
- Fischer, B., Biscaldi, M., & Otto, P. (1993). Saccadic eye movements of dyslexic adult subjects. *Neuropsychologia*, 31, 887–906.
- Gamer, M., Zurowski, B., & Buchel, C. (2010). Different amygdala subregions mediate valence-related and attentional effects of oxytocin in humans. *Proceedings of the National Academy of Sciences*, 107, 9400–9405.
- Gitelman, D. R. (2002). ILAB: A program for postexperimental eye movement analysis. *Behavior Research Methods, Instruments, & Computers*, 34, 605–612.
- Goldberg, J. H., & Helfman, J. I. (2010). *Scanpath clustering and aggregation*. Paper presented at the 2010 Symposium on Eye-Tracking Research & Applications, Austin, TX.
- Golomb, J. D., Marino, A. C., Chun, M. M., & Mazer, J. A. (2011). Attention doesn't slide: Spatiotopic updating after eye movements instantiates a new, discrete attentional locus. *Attention, Perception, & Psychophysics*, 73, 7–14.
- Hoffman, J. E., & Subramaniam, B. (1995). The role of visual attention in saccadic eye movements. *Perception & Psychophysics*, 57, 787–795.
- Huff, M., Papenmeier, F., Jahn, G., & Hesse, F. W. (2010). Eye movements across viewpoint changes in multiple object tracking. *Visual Cognition*, 18, 1368–1391.
- Hunt, A. R., & Kingstone, A. (2003). Covert and overt voluntary attention: Linked or independent? *Cognitive Brain Research*, 18, 102–105.
- Irwin, D. E., & Gordon, R. D. (1998). Eye movements, attention and trans-saccadic memory. *Visual Cognition*, 5, 127–155.
- Itier, R. J., & Batty, M. (2009). Neural bases of eye and gaze processing: The core of social cognition. *Neuroscience and Biobehavioral Reviews*, 33, 843–863.
- Kowler, E., Anderson, E., Doshier, B., & Blaser, E. (1995). The role of attention in the programming of saccades. *Vision Research*, 35, 1897–1916.
- Liu, C.-L., Chiau, H.-Y., Tseng, P., Hung, D. L., Tzeng, O. J. L., Muggleton, N. G., et al. (2010). Antisaccade cost is modulated by contextual experience of location probability. *Journal of Neurophysiology*, 103(3), 1438–1447.
- Matin, E. (1974). Saccadic suppression: A review and an analysis. *Psychological Bulletin*, 81, 899–917.
- Morris, T., Blenkhorn, P., & Farhan, Z. (2002). Blink detection for real-time eye tracking. *Journal of Network and Computer Applications*, 25, 129–143.
- Myung, J. Y., Blumstein, S. E., Yee, E., Sedivy, J. C., Thompson-Schill, S. L., & Buxbaum, L. J. (2010). Impaired access to manipulation features in apraxia: Evidence from eyetracking and semantic judgment tasks. *Brain and Language*, 112, 101–112.
- Nystrom, M., & Holmqvist, K. (2010). An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods*, 42, 188–204.
- Papenmeier, F., & Huff, M. (2010). DynAOI: A tool for matching eye-movement data with dynamic areas of interest in animations and movies. *Behavior Research Methods*, 42, 179–187.
- Rayner, K. (2009). Eye movements and attention in reading, scene perception, and visual search. *Quarterly Journal of Experimental Psychology*, 62, 1457–1506.
- Reimer, B., & Sodhi, M. (2006). Detecting eye movements in dynamic environments. *Behavior Research Methods*, 38, 667–682.
- Salvucci, D., & Anderson, J. R. (2001). Automated eye-movement protocol analysis. *Human Computer Interaction*, 16, 39–86.
- Salvucci, D., & Goldberg, J. (2000). *Identifying fixations and saccades in eye-tracking protocols*. Paper presented at the ACM Symposium on Eye Tracking Research and Applications—ETRA 2000, Palm Beach Gardens, FL.
- Stigchel, S. V., Imants, P., & Richard Ridderinkhof, K. (2011). Positive affect increases cognitive control in the antisaccade task. *Brain and Cognition*, 75, 177–181.
- Theeuwes, J., Belopolsky, A., & Olivers, C. N. (2009). Interactions between working memory, attention and eye movements. *Acta Psychologica*, 132, 106–114.
- Vosskuhler, A., Nordmeier, V., Kuchinke, L., & Jacobs, A. M. (2008). OGAMA (Open Gaze and Mouse Analyzer): Open-source software designed to analyze eye and mouse movements in slideshow study designs. *Behavior Research Methods*, 40, 1150–1162.
- Widdel, H. (1984). Operational problems in analysing eye movements. In A. G. Gale & F. Johnson (Eds.), *Theoretical and applied aspects of eye movement research* (pp. 21–29). New York: Elsevier.