Preprint

T. C. Kübler, K. Sippel, W. Fuhl, G. Schievelbein, J. Aufreiter, R. Rosenberg, W. Rosenstiel, and E. Kasneci. *Analysis of eye movements with Eyetrace*.

This is the author's "personal copy" of the submitted version of the paper, to **Communications in Computer and Information Science (CCIS), Biomedical Engineering Systems and Technologies**
8th International Joint Conference, BIOSTEC 2015, Lisabon, Portugal, January 12-15, 2015, Revised Selected Papers

Analysis of eye movements with Eyetrace

Thomas C. Kübler[1,3*], Katrin Sippel[1*], Wolfgang Fuhl[1], Guilherme Schievelbein[1], Johanna Aufreiter[2], Raphael Rosenberg[2], Wolfgang Rosenstiel[1], and Enkelejda Kasneci[1]

[1] Computer Engineering Department, University of Tübingen, Germany
thomas.kuebler, katrin.sippel, wolfgang.fuhl, wolfgang.rosenstiel, enkelejda.kasneci@uni-tuebingen.de
[2] Department of Art History, University of Vienna johanna.aufreiter, raphael.rosenberg@univie.ac.at
[3] Study Course Ophthalmic Optics/Audiology, University of Applied Sciences Aalen

Abstract. In the time of affordable and comfortable video-based eye tracking, the need for analysis software becomes more and more important. We introduce Eyetrace, a new software developed for the analysis of eye-tracking data during static image viewing. The aim of the software is to provide a platform for eye-tracking data analysis which works with different eye trackers, offering thus the possibility to compare results beyond the specific characteristics of the hardware devices. Furthermore, by integrating various state-of-the-art and new developed algorithms for analysis and visualization of eye-tracking data, the influence of different analysis steps and parameter choices on typical eye-tracking measures is totally transparent to the user. Eyetrace integrates several algorithms to identify fixations and saccades, and to cluster them. Well-established algorithms can be used side-by-side with bleeding-edge approaches with a continuous visualization. Eyetrace can be downloaded at http://www.ti.uni-tuebingen.de/Eyetrace.1751.0.html and we encourage its use for exploratory data analysis and education.

1 Introduction

Eye-tracking technology has found its way into many fields of application and research during the last years. With ever cheaper and easier to use devices, the traditional usage in psychology and market investigation was accompanied by new application fields, especially in medicine and natural sciences. Together with the number of devices, the number of software for the analysis of eye tracking data increased steadily (e.g., SMI begaze, Tobii Analytics, D-Lab, NYAN, Eyeworks, ASL Results Plus, or Gazepoint Analysis). Major brands offer their individual analysis software with ready-to-run algorithms and preset several parameter settings for their eye-tracker device and typical applications. All of them share common features (such as visualizing gaze traces, attention maps and calculating area of interest statistics) and distinguish in minor features. A major

[*] Contributed equally to this paper

restriction in their usability are the licensing regulations. Supplying a class of students with licenses for home use, post-hoc data analysis years after recording the data may be difficult due to the financial effort associated with the licensing regulations. Besides that, these applications usually can not be extended by custom algorithms and specialized evaluation methods. Not few studies reach the point where the manufacturer software is insufficient and its extension is not possible. Thus, the recorded data has to be exported and loaded into other programs, e.g., Matlab, for further processing. Furthermore, individual calculations are often non-opaque or not documented in the necessary detail in order to allow comparison between studies conducted with different eye-tracker devices or even between different recording software versions.

Eyetrace supports a range of common eye-trackers and offers a variety of state-of-the-art algorithms for eye-tracking data analysis. The aim is not only to provide a standardized work-flow, but also to highlight the variability of different eye-tracker devices as well as different algorithms. For example, instead of just finding all fixations and saccades in the data, we enable the data analyst to test whether the choice of parameters for the fixation filter was adequate. Our approach is driven by continuous data visualization such that the result of each analysis step can be visually inspected. Different visualization techniques are available and can be active at the same time, i.e. a scanpath can be drawn over an attention map with areas of interest highlighted. All visualizations are customizable in order to visualize grouping effects, being distinguishable on different backgrounds and for color-blind persons.

We realize that no analysis software can provide all the tools required for every possible study. Therefore, the software is not only extensible but also offers the possibility to export all data and preliminary analysis results for usage with common statistics software such as Gnu R or SPSS.

Eyetrace originates from a collaboration of the department of art history at the University of Vienna [2, 12, 15] and the computer science department at the University of Tübingen. It consists of the core analysis component and a pre-processing step that is responsible for compatibility with many different eye-tracker devices as well as data quality analysis. The software bundle, including Eyetrace and EyetraceButler is written in C++, based on the experience of the previous version (EyeTrace 3.10.4, developed by Martin Hirschbühl with Christoph Klein and Raphael Rosenberg) as well as other eye-tracking analysis tools [21].

We are eager to implement state-of-the-art algorithms, such as fixation filters, clustering algorithms, and data-driven area of interest annotation and share the need to understand how these methods work. Therefore, implemented methods as well as their parameters are transparent to the user and well documented including references to the original work introducing them.

Eyetrace is available free of charge for non commercial research and educational purposes. It can be employed for analysis of eye-tracking data in scientific studies, in education and teaching.

2 Data Preparation

In order to make the use of different eye-tracker types convenient, recordings are preprocessed and converted into an eye-tracker independent format. The preprocessing software EyetraceButler handles this step and can also be used to split a single recording into subsets (e.g., by task or stimulus) and to perform a data quality check.

EyetraceButler provides a separate plug-in for all supported eye-trackers and converts the individual eye-tracking recordings into a format that holds information common to almost all eye-tracking formats. More specifically, it contains the x and y coordinates for both eyes, the width and height of the pupil as well as a validity bit, together with a joint timestamp. For monocular eye-trackers or eye-trackers that do not include pupil data the corresponding values are set to zero. A quality report is then produced containing information about the overall tracking quality as well as individual tracking losses (Fig. 1). Especially for demanding tracking situations, the quality report enables distinction between an overall low tracking quality and the partial loss of tracking for a time slice (such as at the beginning or end of the recording).

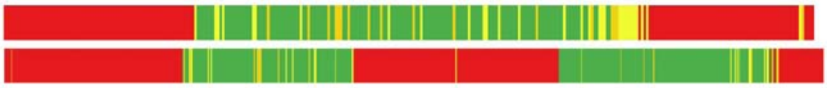


Fig. 1. Quality analysis for two recordings with a binocular eye-tracker. The color represents measurement errors (red), successful tracking of both eyes (green) and of only one eye (yellow) over time. It is easy to visually assess the overall quality of a recording as well as the nature of individual tracking failures.

2.1 Supplementary Data

In addition to the eye-tracking data, arbitrary supplementary information about the subject or relevant experimental conditions can be added, e.g. gender, age, dominant eye, or (subject's) patient's status. This information is made available to Eyetrace along with information concerning the stimulus viewed. Based on this information, the program is able to sort and group all loaded examinations according to these values.

2.2 Supported Eye-Tracker Devices

The EyetraceButler utilizes slim plug-ins in order to implement new eye-tracker profiles. To date, plug-ins for five different eye-trackers are available, among them devices by SMI, Ergoneers, TheEyeTribe as well as a calibration-free tracker recently developed by the Fraunhofer Institute in Ilmenau.

3 Data Analysis

3.1 Loading, Grouping and Filtering Data

Data files prepared by the EyetraceButler can be batch loaded into Eyetrace together with their accompanying information such as the stimulus image or subject information. Visualization and analysis techniques can handle subjects grouping by any of the arbitrary subject information fields. For example attention maps can be calculated separately for each subject, cumulative for all subjects or by subject groups. This allows to compare subjects with healthy vision to a low vision patient group or to compare the viewing behavior of different age groups. Adaptive filters are provided to select the desired grouping and individual recordings can be included or excluded from the visualization and analysis process.

3.2 Fixation and Saccade Identification

One of the earliest and most frequent analysis steps is the identification of fixations and saccades. Their exact identification is essential for the calculation of many eye movement characteristics, such as the average fixation time or saccade length.

Eye-tracking manufacturers often offer the possibility to identify fixations and saccades automatically. However, this filtering step is not as trivial as the automated annotation may suggest. In fact, different algorithms yield quite different results. By offering a variety of calculation methods and making their parameters available for editing, we want to bring to mind the importance of the right choice of parameters. Especially when it comes to identifying the exact first and last point that still belong to a fixation and the merging of subsequent fixations that come to fall to the same location, relevant differences between algorithms and a high sensitivity to parameter changes can be observed.

To date following algorithms are implemented in Eyetrace:

Standard Algorithm The standard algorithm for separating fixations and saccades is based on three adjustable values: The minimum duration of the fixations, the maximum radius of the fixations and the maximum number of points that are allowed to be outside this radius (helpful with noisy data). A time window of the minimum fixation duration is shifted over the measurement points until the conditions of maximum radius and maximum outliers are fulfilled. In the following step the beginning fixation is extended if possible until the number of allowed outliers has been reached. A complete fixation has been identified and the procedure starts anew. Every measurement point that was not assigned to a fixation is assigned to the saccade between its predecessor and successor fixation.

Velocity-Based Algorithm Since saccades show high eye movement speed while fixations and smooth pursuit movements are much slower, putting a threshold on the eye movement speed is a straight forward way of fixation filtering. Eyetrace currently implements three different variants of velocity based fixation identification. Each of the methods can filter short fixations via a minimum duration in a post-processing step.

Velocity Threshold by Pixel Speed [px/s] A simple threshold over the speed between subsequent measurements. If the speed is exceeded, the measurement belongs to a saccade, otherwise to a fixation. While a pixel per second threshold is easy to interpret for the computer, it is often not meaningful to the experimenter and therefore hard to choose.

Velocity Threshold by Percentile Based on the assumption that the velocity is bigger within saccades than within fixations, velocities are sorted by magnitude and a threshold is chosen by a percentile of the data selected by the user (usually 80-90%). An example of sorted distances between measurements:
1 3 7 11 12 13 18 21 21 22
Green distances are supposed to belong to fixations for a 60% percentile (6 out of 10 distances) and the value 18 would be chosen as velocity threshold.

Velocity Threshold by Angular Velocity [/s] This is the representation most common in the literature since it is independent of pixel count and individual viewing behavior. However, it also requires most knowledge about the data recording process in order to be able to convert the pixel distances into angular distances (namely the distance between viewer and screen, screen width and resolution). Suggested values for individual tasks can be found in the literature [1, 16]

Gaussian Mixture Model A Gaussian mixture model as introduced in [19] is also implemented in Eyetrace. This method is based on the assumption that distances between subsequent measurement points within a fixation form a Gaussian distribution. Furthermore distances between measurement points that belong to a saccade also form a Gaussian distribution, but with different mean and standard deviation. A maximum likelihood estimation of the parameters of the Mixture of Gaussians is performed. Afterward for each measurement point the probability that it belongs to a fixation or to a saccade can be calculated and fixation/saccade labels are assigned based on these probabilities (Fig. 2). The major advantage of this approach is that all parameters can be derived from the data. One could evaluate data recorded during an unknown experiment without the need to specify any thresholds or experimental conditions. The method has been evaluated in several studies [7–9].

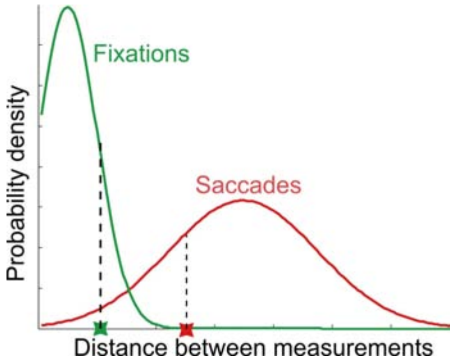


Fig. 2. Fit of two Gaussian distributions to the large distances between subsequent measurements within saccades and the short distances between fixations. Two sample points are shown, one with higher probability to belong to a fixation (left) and one with a higher probability for a saccade (right).

3.3 Fixation Clustering

After identification of fixations and saccades the fixations can also be clustered. Clustering fixations either by neighborhood thresholds or mean-shift clustering (as proposed by [17]) results in data-driven, automatically assigned areas of interest.

Fixation clusters can be calculated on the scan patterns of one subject or cumulative on a group of subjects.

Standard Clustering Algorithm This greedy algorithm requires the definition of a minimum number of fixations that will be considered a cluster, the maximum radius of a cluster , and the overlap. Fixations are sorted in descending order of the number of included gaze points. Starting with the longest fixation, the algorithm iterates over all fixations, checking whether they fulfill the conditions of building a cluster with the biggest one. If the number of found fixations is sufficient, all found fixations are assigned to the same cluster and excluded from further clustering. If not, the first fixation cannot be assigned to any cluster and the algorithm starts again from the second longest fixation.

Mean-Shift Clustering The mean-shift clustering method assumes that measurements are sampled from Gaussian distributions around the cluster centers. The algorithm converges towards local point density maxima. The iterative procedure is shown in Fig. 3. One of the main advantages is that it does not require the expected number of clusters in advance but determines an optimal clustering based on the data.

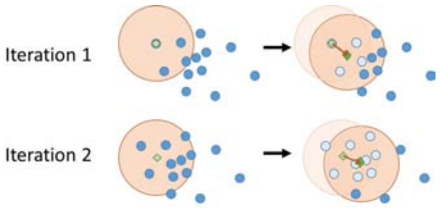


Fig. 3. Simplified visualization of the mean-shift algorithm for the first two iterations at one starting point. In each iteration the mean (green square) of all data points (blue circles) within a certain window around a point (big red circle) is calculated. In the next iteration the procedure is repeated with the window shifted towards the previous mean. This is done until the mean convergence.

Cumulative Clustering The clustering algorithms mentioned above can also be used on the cumulative data of more than one subject or more than one experiment condition. This way cumulative population clusters can be formed. They are more robust to noise and individual viewing behavior differences. The parameters of the algorithms are adapted for cumulative usage (e.g. the number of minimum fixations for the standard algorithm depends on the number of data sets used for cumulative analysis), but the way the methods work remain the same.

3.4 Areas of Interests (AOIs)

For the evaluation of gaze directed at specific regions, Eyetrace provides both the possibility to annotate top-down AOIs manually (via a graphical editor with only few mouse clicks) or to determine bottom-up clusters of high fixation density. Fig. 4 shows an example of manually defined AOIs and bottom-up defined clusters.

4 Data Visualization

The software allows simultaneous visualization of multiple scanpaths. These may represent different subjects, subject groups or distinct experiment conditions. The scan patterns are rendered in real-time as an overlay to an image or video

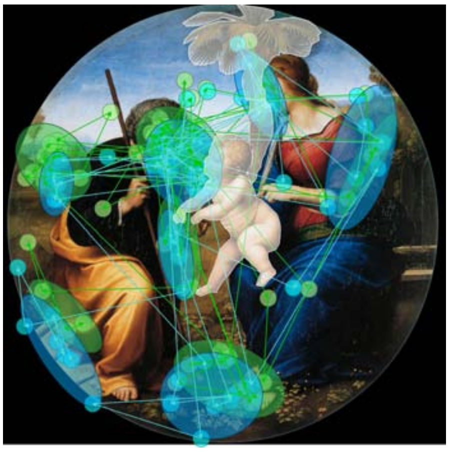


Fig. 4. Simultaneous overlay of multiple visualization techniques for one scanpath of an image viewing task. The background image is shown together with a scanpath representation of fixed-size fixation markers (small circles) and generated fixation clusters (bigger ellipses) for left (green) and right (blue) eye. AOIs were annotated by hand (marked as white overlay).

stimulus. Various customizable visualization techniques are available: Fixations that encode fixation duration in their circular size, elliptical approximations encoding spatial extend as well as attention and shadow maps. Exploratory data analysis can be performed by traversing through the time dimension of the scan patterns as if it was a video. Most of the visualizations are interactive so that placing the cursor over the visualization of e. g. a fixation gives access to detailed information such as its duration and onset time.

4.1 Fixations and Fixation Clusters

The visualization of fixations and fixation clusters has to account for their spatial and temporal information. It is common to draw them as circles of either uniform size or to encode the fixation duration in the circle diameter (see Fig. 6). Besides these options, Eyetrace can fit an ellipse to the spatial extend of all measurement points assigned to the fixation. The eigenvectors of these measurements point into the direction of highest variance within the data (see Fig. 5) and can therefore be used as major and minor axis for the ellipse fit. This visualization gives an excellent impression of measurement accuracy, since fixations are supposed to represent a relatively stable eye position and the variance of measurements within a fixation therefore stands representative for the measurement inaccuracy. When evaluating different fixation filters the circularity of resulting ellipses can be used as a quality measure: Realistic fixations are supposed to be approximately circular. Once fixations begin to grow ellipsoid, the choice of parameters for the selected filter is probably inadequate.

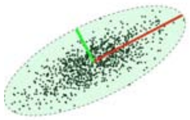
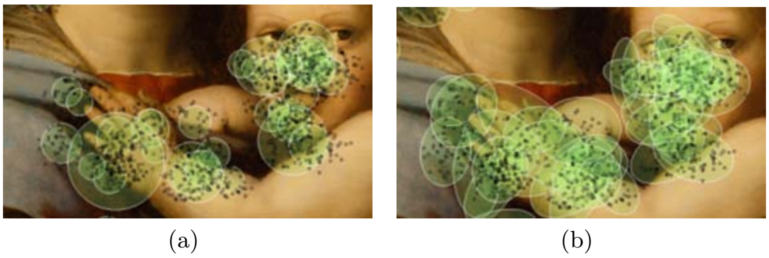


Fig. 5. The two eigenvectors of Gaussian distributed samples (that correspond to the directions of highest variance). These are used as the major and minor axes for an elliptic fit.



(a)

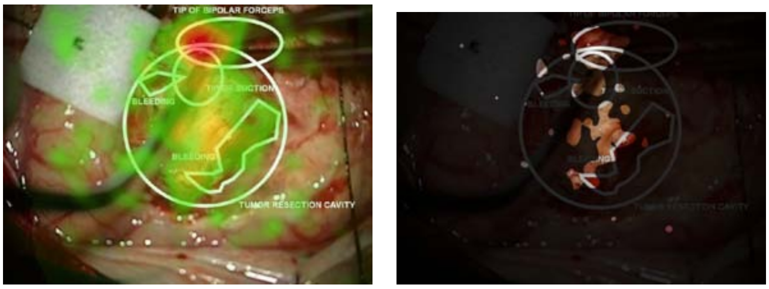
(b)

Fig. 6. (a) Fixation visualization as circles scaled by the respective dwell times, representing location and temporal information. (b) Elliptic fit to the individual measurement points assigned to the fixation. This visualization indicates the measurement accuracy and the quality of the fixation filter. Usually fixations are supposed to be circular, however wrong settings during the fixation identification step may cause adjacent saccade points to be clustered into the fixation and therefore deform the circle towards a more ellipsoid shape.

4.2 Attention and Shadow Maps

Attention maps are one of the most common eye-tracking analysis tools, besides the high number of subjects that have to be measured in order to get reliable results [14]. In order to enable fast attention map rendering even for a large number of recordings and high resolution, the attention map calculation utilizes multiple processor cores. Attention maps can be calculated for gaze points, fixations and fixation clusters. We provide the classical red-green color palette for attention maps as well as blue version for color-blind persons.

For the gaze point attention map each gaze point contributes as a two dimensional Gaussian distribution. The final attention map is calculated as the sum over all Gaussians. The Gaussian distribution is specified by the two parameters



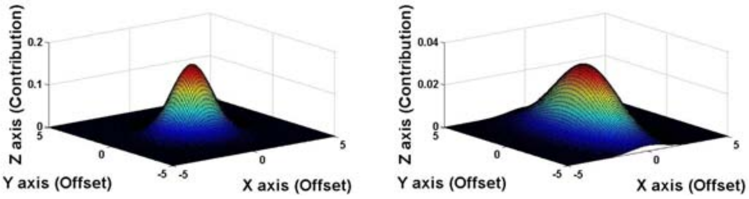
(a) (b)

Fig. 7. An attention map calculated for fixation clusters (a) and the corresponding shadow map (b). Data of microsureons during a tumor removal surgery lend from [5].

size and intensity which are adjustable by the user. This Gaussian distribution is circular because gaze points do not have information about orientation and size. For fixations and fixation clusters the elliptic fit is used to determine the shape and orientation of the Gaussian distribution. Fig. 8 shows an example of a circular Gaussian distribution (a) and a stretched, elliptical one (b).

$$P(x,y) = \frac{1}{\sigma_1 \sigma_2 2\pi \sqrt{1-\varrho}} e^{-\frac{1}{2(1-\varrho^2)}(\frac{x^2}{\sigma_1^2} + \frac{y^2}{\sigma_2^2} - \frac{2\varrho xy}{\sigma_1 \sigma_2})} \tag{1}$$

Equation 1 shows the Gaussian distribution in the two dimensional case. σ_1 and σ_2 are the variance in horizontal and vertical direction respectively (see Fig. 8). x and y are the offsets to the center of the Gaussian distribution (see Fig. 8). The correlation coefficient ϱ is zero in Fig. 8 to simplify the case.



(a) (b)

Fig. 8. Two Gaussian distribution calculated with $\sigma_1 = \sigma_2 = 1$ (a) and $\sigma_1 = 1$ and $\sigma_2 = 5$ (b).

A variant of the attention map is the shadow map that reveals only areas frequently hit by gaze (see Fig. 7(b)). Its calculation is identical to that of the attention map with the difference of a smoothing step in order to show the border regions with higher sensitivity. This is done by calculating the n-th root of each map value where n is a user-defined parameter that regulates the desired smoothing.

4.3 Saccades

Saccades are typically visualized as arrows or lines connecting two fixations.

Besides this, a statistical evaluation can be visualized as a diagram called anglestar. It consists of a number of slices and a rotation offset. A slice of the anglestar codes in its length the number of saccades with the same angular orientation as the slice (e.g. if the slice represents the angles between 0° and 45° the number of saccades within that angle range contribute to that slice) to the horizontal axis is considered. The extend of a slice from the center of the star can represent the quantity, summed length or summed duration of the saccades towards that direction. Fig. 9 shows a diagram where the extension of the slices is based on the summed length of the saccades.

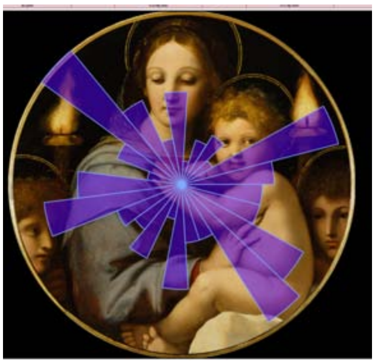
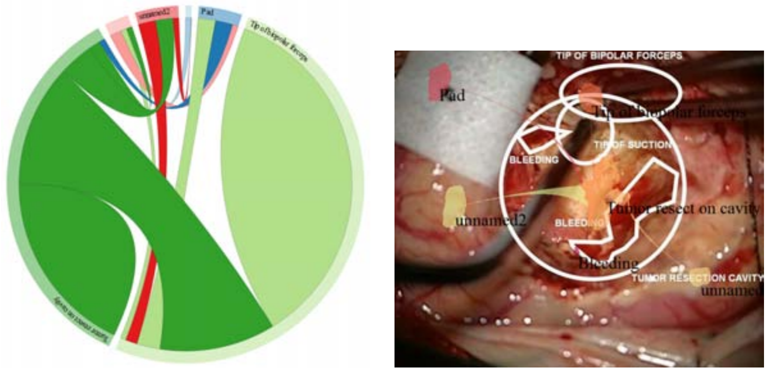


Fig. 9. Representation of an anglestar.

4.4 Fixations Clusters and AOI Transitions Diagram

For some evaluation cases it is interesting in which sequence attention is shifted between different areas. The AOI transitions diagram (Fig. 10) visualizes the transition probabilities between AOIs manually annotated and/or converted from fixation clusters during a specific time period. The color of the transition is inherited from the AOI with most outgoing saccades. Hovering the mouse over an AOI shows all transitions from this AOI and hides the transitions from all other AOIs. Hovering the cursor over a specific transition displays an information box containing the number of transitions in both directions. Fig. 10(b) visualizes the transitions directly on the image.



(a) (b)

Fig. 10. (a) Abstract diagram of the transitions between AOIs. The graphic is interactive and can blend out irrelevant edges if one AOI is selected. (b) AOIs as defined by cumulative fixation clusters and corresponding transition frequencies as an overlay over the original image. Eyetrace offers interaction with the graphic via the cursor in order to show only the currently relevant subset of transitions from and to one AOI.

5 Data Export

5.1 Statistics

General Statistics Independent of all other calculations it is possible to calculate some general gaze statistics. These include the horizontal and vertical gaze activity, minimum, maximum and average speed of the gaze. These statistics shine a light on the agility and exploratory behavior of the subjects and can be exported in a format ready to use in statistical programs such as JMP or SPSS.

AOI Statistics Numerous gaze characteristics can be calculated for AOIs, such as the total number of glances towards the AOI, the time of the first glance, glance frequency, total glance time, the glance proportion towards the AOI in respect to the whole recording and the minimum, maximum and mean glance duration. Glance in this case means that a sequence of gaze points are located inside the AOI, no matter if they belong to a saccade or a fixation. These statistics are a supplement to the AOI transitions diagram and can also be exported.

5.2 Visualization

The transition diagram as well as every visualization can be exported either lossless as vector graphics or as bitmaps (png, jpg). Eyetrace provides the option to export the information about the subject (e.g. age, dominant eye) and the parameters used for calculation and visualization as a footer in the exported image. That way results can be reproduced and understood based solely on the exported image.

5.3 Evaluation Results

After calculating fixations, fixation clusters or cumulative clusters Eyetrace provides the possibility to export them as a text file.
Fixations are exported in a table including the running number, the number of included points, x and y coordinate, radius and if calculated the id of the cluster this fixation belongs to. The text file for the clusters and cumulative clusters include an ID number, the number of fixations contained, mean x, mean y and the radius.

6 Conclusion and Outlook

Eyetrace is a well-structured software tool, advantageous enough to be employed for academic research in a number of application fields but with its convenient handling nonetheless usable for persons without broad eye-tracking experience, e.g., for teaching students. The major advantages of the software are the flexibility of algorithms and their parameters as well as their actuality with respect to the state-of-the-art.

Eyetrace has already been tested on data of various research projects, ranging from the viewing of fine art [2, 12, 15] recorded via a static binocular SMI infrared eye-tracker to on-road and simulator driving experiments [10, 20] and supermarket search tasks [11, 18] recorded via a mobile Dikablis tracker by Ergoneers.

In our future work, we will not only extend the EyetraceButler for applicability to further eye-tracking device, but also extend the general and AOI-based statistics calculations, integrate and implement new calculation and visualization algorithms and make the existing ones more interactive. A special focus will be given to the analysis and processing of saccadic eye movements as well as to the automated annotation of AOIs for dynamic scenarios [13] and non-elliptical AOIs. In addition, we plan on including further automated scanpath comparison metrics, such as MultiMatch [3] or SubsMatch [13].

Another relevant area is for our future work will be the monitoring of vigilance and workload during visual tasks. Especially for medical applications such as reaction or stimulus sensitivity testing, the mental state of the subject is of great importance. Available data, such as the pupil dilation, fatigue waves [6], saccade length differences [4], and blink rate may give important insight into the recorded data and even yield e.g. cognitive workload weighted attention maps.

Acknowledgments. We want to thank the department of art history at the university of Vienna for the inspiring collaboration. The project was partly financed by the the WWTF (Project CS11-023 to Helmut Leder and Raphael Rosenberg).

References

1. Blignaut, P.: Fixation identification: The optimum threshold for a dispersion algorithm. Attention, Perception, & Psychophysics 71(4), 881–895 (2009)
2. Brinkmann, H., Commare, L., Leder, H., Rosenberg, R.: Abstract Art as a Universal Language? Leonardo 47(3), 256–257 (2014)
3. Dewhurst, R., Nyström, M., Jarodzka, H., Foulsham, T., Johansson, R., Holmqvist, K.: It depends on how you look at it: Scanpath comparison in multiple dimensions with MultiMatch, a vector-based approach. Behavior research methods 44(4), 1079–1100 (2012)
4. Di Stasi, L.L., McCamy, M.B., Macknik, S.L., Mankin, J.a., Hooft, N., Catena, A., Martinez-Conde, S.: Saccadic eye movement metrics reflect surgical residents' fatigue. Annals of surgery 259(4), 824–9 (Apr 2014), `http://www.ncbi.nlm.nih.gov/pubmed/24169184`
5. Eivazi, S., Bednarik, R., Tukiainen, M., von und zu Fraunberg, M., Leinonen, V., Jääskeläinen, J.E.: Gaze behaviour of expert and novice microneurosurgeons differs during observations of tumor removal recordings. In: Proceedings of the Symposium on Eye Tracking Research and Applications. pp. 377–380. ACM (2012)
6. Henson, D.B., Emuh, T.: Monitoring vigilance during perimetry by using pupillography. Investigative ophthalmology & visual science 51(7), 3540–3 (Jul 2010), `http://www.ncbi.nlm.nih.gov/pubmed/20207962`
7. Kasneci, E.: Towards the Automated Recognition of Assistance Need for Drivers with Impaired Visual Field. Ph.D. thesis, University of Tübingen, Wilhelmstr. 32, 72074 Tübingen (2013), `http://tobias-lib.uni-tuebingen.de/volltexte/2013/7033`
8. Kasneci, E., Kasneci, G., Kübler, T.C., Rosenstiel, W.: The applicability of probabilistic methods to the online recognition of fixations and saccades in dynamic scenes. In: Proceedings of the Symposium on Eye Tracking Research and Applications. pp. 323–326. ACM (2014)
9. Kasneci, E., Kasneci, G., Kübler, T.C., Rosenstiel, W.: Online Recognition of Fixations, Saccades, and Smooth Pursuits for Automated Analysis of Traffic Hazard Perception. In: Artificial Neural Networks, pp. 411–434. Springer (2015)
10. Kasneci, E., Sippel, K., Aehling, K., Heister, M., Rosenstiel, W., Schiefer, U., Papageorgiou, E.: Driving with binocular visual field loss? A study on a supervised on-road parcours with simultaneous eye and head tracking. PloS one 9(2), e87470 (2014)
11. Kasneci, E., Sippel, K., Aehling, K., Heister, M., Rosenstiel, W., Schiefer, U., Papageorgiou, E.: Homonymous Visual Field Loss and its Impact on Visual Exploration - A Supermarket Study. Translational Vision Science and Technology In Press (09 2014)
12. Klein, C., Betz, J., Hirschbuehl, M., Fuchs, C., Schmiedtová, B., Engelbrecht, M., Mueller-Paul, J., Rosenberg, R.: Describing Art–An Interdisciplinary Approach to the Effects of Speaking on Gaze Movements during the Beholding of Paintings. PloS one 9(12), e102439 (2014)
13. Kübler, T.C., Bukenberger, D.R., Ungewiss, J., Wörner, A., Rothe, C., Schiefer, U., Rosenstiel, W., Kasneci, E.: Towards automated comparison of eye-tracking recordings in dynamic scenes. In: EUVIP 2014 (12 2014)
14. Pernice, K., Nielsen, J.: How to conduct eyetracking studies. Nielsen Norman Group (2009)

15. Rosenberg, R.: Blicke Messen. Vorschläge für eine empirische Bildwissenschaft. Jahrbuch der Bayerischen Akademie der Schönen Künste 27, 71–86 (2014)
16. Salvucci, D.D., Goldberg, J.H.: Identifying fixations and saccades in eye-tracking protocols. In: Proceedings of the 2000 symposium on Eye tracking research & applications. pp. 71–78. ACM (2000)
17. Santella, A., DeCarlo, D.: Robust clustering of eye movement recordings for quantification of visual interest. Proceedings of the Eye tracking research & applications symposium on Eye tracking research & applications - ETRA'2004 pp. 27–34 (2004), http://portal.acm.org/citation.cfm?doid=968363.968368
18. Sippel, K., Kasneci, E., Aehling, K., Heister, M., Rosenstiel, W., Schiefer, U., Papageorgiou, E.: Binocular Glaucomatous Visual Field Loss and Its Impact on Visual Exploration - A Supermarket Study. PloS one 9(8), e106089 (2014)
19. Tafaj, E., Kasneci, G., Rosenstiel, W., Bogdan, M.: Bayesian online clustering of eye movement data. In: Proceedings of the Symposium on Eye Tracking Research and Applications. pp. 285–288. ACM (2012)
20. Tafaj, E., Kübler, T.C., Kasneci, G., Rosenstiel, W., Bogdan, M.: Online classification of eye tracking data for automated analysis of traffic hazard perception. In: Artificial Neural Networks and Machine Learning–ICANN 2013, pp. 442–450. Springer (2013)
21. Tafaj, E., Kübler, T.C., Peter, J., Rosenstiel, W., Bogdan, M., Schiefer, U.: Vishnoo: An open-source software for vision research. In: Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on. pp. 1–6. IEEE (2011)