

EyeTrackR: An R Package for Extracting and Visualizing Data from Mobile and Static Eye Trackers

Chunyang Li*

Jürgen Symanzik†

Abstract

Eye tracking has been used in many scientific fields, such as behavioral sciences, education, marketing, and sports. Visualization usually plays an important role in the analysis of eye tracking data. So far, there existed no R package that contains a whole collection of eye tracking data processing and visualization tools. In this article, we introduce an R package, **EyeTrackR**, for processing and visualizing data from mobile and static eye trackers. The main functionalities in **EyeTrackR** are: (i) Automatic extraction of mobile eye tracking data; (ii) Definition of Areas of Interest (AOIs) and data summarization; (iii) Common eye tracking visualization tools; (iv) Linked microposter plots, linked timeline microposter plots, and linked scanpath microposter plots. Our primary application in this article is for people looking at scientific posters. The plots from (iv) are based on the idea of linked micromap plots, a plot type that displays geographical distributions by linking statistical summaries with smaller groups of geographical regions. They have been specifically developed and implemented to visualize how people are looking at different components of scientific posters.

Key Words: Graphics, Mobile Eye Tracker, Linked Microposter Plots, Linked Timeline Microposter Plots, Linked Scanpath Microposter Plots.

1. Introduction

Eye tracking is the process of measuring where people are looking at with an eye tracker device, either a mobile eye tracker or a static eye tracker. Static eye trackers are based on a desktop, hence they are often used to study eye motion on a computer screen. Mobile eye trackers are fixed on a user's head, so they are not limited within a restricted area and can be used for a variety of activities, such as playing soccer, driving, etc. Both types of current eye trackers are using the corneal reflection of an infrared light emitting diode to illuminate and generate a reflection off the surface of the eye (Cooke, 2005). This approach is able to track pupils precisely, therefore eye tracking data, that is, meaningful scene videos indicating where people are looking at, are generated.

Eye tracking technology has become more and more affordable and accessible nowadays (Gould and Zolna, 2010) and has been adopted in a variety of research fields, including research for posters and related media. Barber et al. (2008) investigated posters in a computer simulated outdoor environment in order to “provide common measurement framework for poster panel visibility across settings and perspectives” with an eye tracking approach. Andersson (2010) looked at the effect of visual in-store advertisement designing on customers' decisions on purchasing, with participants' eye movement data recorded sitting in front of a computer screen. Using a mobile eye tracker, Foulsham and Kingstone (2011) investigated how people were looking at posters in an indoor environment, but with a focus on academic posters from psychology. However, none of the existing literature on eye tracking

*Utah State University, Department of Mathematics and Statistics, 3900 Old Main Hill, Logan, UT 84322, USA. E-mail: lichunyang1990@hotmail.com

†Utah State University, Department of Mathematics and Statistics, 3900 Old Main Hill, Logan, UT 84322, USA. E-mail: symanzik@math.usu.edu

for posters has specifically discussed eye tracking visualization or adopted any new visualization techniques.

Visualization tools are among the most important tools to explore eye tracking data. Common statistical graphics are frequently used for eye tracking visualization. These graphics are mostly used to present the data from the eye tracking metrics or the raw eye tracking data. These graphics include line charts, bar charts, scatter plots, and box plots.

Other plot types, specifically for eye tracking data, include attention maps, timelines with either point data or Area of Interest (AOI) data, and scanpath visualization.

- Attention maps are usually made of heat maps or hot spot maps with a Gaussian kernel function. They describe the spatial distribution of eye tracking data and the hot spots of the map point out “the regions attracted people’s gazes” (Holmqvist et al., 2011). With the hot spots overlaid on top of an image, it is quite obvious at which areas people are looking most. However, the sequential order of where one is looking is not shown in attention maps.
- Timelines are typically used to visualize temporal data (Blascheck et al., 2014). The AOI timelines are one of the most commonly used timeline plots for eye tracking data. An AOI timeline is the visualization of eye tracking data divided by AOIs: the horizontal axis shows the AOIs and the vertical axis shows the time. In this representation, both the start and end times, as well as the temporal sequence of changes between AOIs can be identified in the same plot. However, an AOI timeline does not present the spatial locations of each AOI from the plot underlying graph, photo, or poster.
- The term “scanpath” was first introduced by Noton and Stark (1971) to describe the chain of fixations and saccades. In visual representations of scanpaths, circles are used to represent fixations and lines are used to represent saccades (Goldberg and Helfman, 2010). The radius of the circle indicates the duration of the fixation. The numbers in the circles indicate the sequential order of the fixations. Scanpath visualizations give the sequence of one’s eye movements, however, when the viewing patterns become more complex, it is very difficult to perceive the visual patterns.

However, there is no plot type specifically designed for visualizing how people are looking at posters. To overcome the disadvantages of the commonly used eye tracking data visualization techniques, Li and Symanzik (2016) initially proposed the linked microposter plot, as well as the implementation in R (R Core Team, 2016), to visualize eye movement data on posters, based on the idea of the linked micromap plot (Carr and Pickle, 2010). In this article, the linked microposter plot is further extended to a linked timeline microposter plot and a linked scanpath microposter plot to add the time information in the plot and to overcome the shortcomings of the AOI timelines and scanpath visualization.

Companies that design eye tracking technologies offer commercial software for eye tracking data analysis in general. ASL Results Plus GE provided by the Applied Science Laboratory (ASL) summarizes the eye tracking data by AOIs and also makes it possible to create bar chart and heat maps (Applied Science Laboratories, 2015). Tobii Pro Studio provided by Tobii also summarizes the eye tracking data by AOIs and offers visualization tools such as bar plots and timeline representations

(Tobii Technology, 2017). However, these software packages apply only to the eye tracking hardware from the manufacturer and are not supporting any other eye trackers (Zhegallo and Marmalyuk, 2015). Software for eye tracking data has been developed with R packages (von der Malsburg, 2015; Dink and Ferguson, 2015), **Matlab** toolboxes and functions (Krassanakis et al., 2014; Berger et al., 2012), **Python** packages (Dalmaijer et al., 2014), and in other environments (Heminghous and Duchowski, 2006), to detect eye movement events, to visualize and model eye tracking data, and to clean raw eye tracking data. These software developments make it possible to support eye tracking hardware from different manufacturers. Among these existing software packages, there are some **R** packages, **Matlab** toolboxes, and **Python** packages specifically designed to visualize eye tracking data: the **iMap** Matlab toolbox allows the generation of attention maps with Gaussian kernels (Caldara and Mielle, 2011); the **GazeParser** Python package provides various functions to detect saccades and fixations and plot scanpaths (Sogo, 2015); The **ETRAN** R package provides fixation detection, attention maps, and scanpath creation (Zhegallo and Marmalyuk, 2015).

However, none of the existing software developments offers the full combination of eye tracking data processing, visualization, and features specifically required for mobile eye tracking data from people looking at posters. In this article, we introduce our new **R** package, **EyeTrackR**, to process video data recorded from a mobile eye tracker, and to conduct different visualization tasks including the new plot types we introduced, i.e., the linked microposter plots, the linked timeline microposter plots, and the linked scanpath microposter plots.

The remainder of this article is structured as follows: We will discuss the mobile eye tracking device and how the data are collected in Section 2. The **EyeTrackR** R package and its functionalities are discussed in Section 3. We will finish with our conclusion and future work in Section 4.

2. Mobile Eye Tracking Device and Data Collection

The mobile eye tracker manufactured by ASL consists of a portable Data Transmit Unit (DTU), a laptop with a wireless reception connected to the DTU, and a pair of eye tracking glasses with optics. The eye tracking glasses are the main part of the mobile eye tracker. There are two cameras on the eye tracking glasses: one tracks the participant's eye and the other one records the environment observed by the participant. The data generated from such equipment typically include a scene video indicating where the participant is looking at (there is only one video generated for each recording, though we have two cameras), and a data file that contains the x and y coordinates, pupil radiuses, mouse cursor positions, etc. The x and y coordinates exactly refer to where the participant is looking at on the computer screen for most static eye trackers, therefore, they can be directly processed with the eye tracking data analysis software. However, for mobile eye trackers and some head mounted static eye trackers, the coordinates in the data file correspond to different coordinate systems, i.e., the coordinate system is changing for every single frame. In these situations, the videos first must be processed so that all coordinates are transformed to the same coordinate system in terms of a poster, thus the resulting data can be used for further analysis.

For each poster, the AOIs are defined in advance, such as the title, logos, multiple text areas, images, and tables. The AOIs on the posters can be defined automatically with R (see Section 3.2). Figure 1 shows the twelve defined AOIs of the poster used

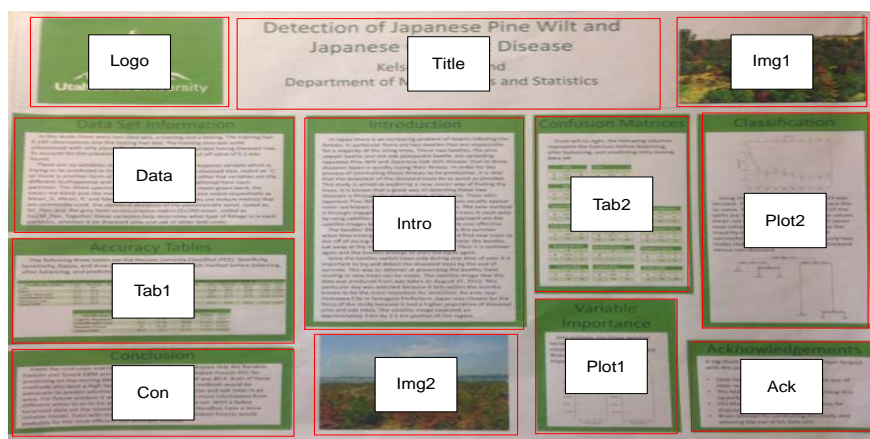


Figure 1: A clear photo of the poster in JPEG format with twelve AOIs

in this article. This poster shows a student’s course project from a multivariate statistics class at Utah State University. The red bounding boxes outline the defined AOIs. The areas that are not in the bounding box are defined as “Blank”. The poster is used to test the data processing results and the validity of the linked microposter plots. The participant is timed to look at eight of the AOIs for about five seconds and at four AOIs for about two seconds, using a 30 Hz mobile eye tracker from ASL that records 30 images per second. Overall, the participant looked at the poster for about 56 seconds, resulting in a total of 1680 video frames.

3. Functions in the EyeTrackR Package

In this section, we will present four functionality groups of functions contained in the **EyeTrackR** R package: (i) data processing, (ii) data summarization, (iii) common eye tracking visualization tools, and (iv) linked microposter plots. Data processing includes two functions that process the video record of a participant looking at a poster. Data summarization includes several functions that define AOIs and summarize the data by the AOIs defined. Common eye tracking visualization tools contains functions that create the commonly used eye tracking visualization techniques such as dot plots, bar charts, attention maps, etc. Linked microposter plots contains three functions that create the linked microposter plot, the linked timeline microposter plot, and the linked scanpath microposter plot. While functions from group (i) are primarily used for mobile eye tracking data, functions from the three other groups can be used for data from both mobile and static eye trackers. Table 1 summarizes the functions and their main functionalities.

Table 1: Functions in the **EyeTrackR** R Package

Function Name	Functionality Group
(i) Data Processing	
<i>GetVideoFrames</i>	Break AVI format video into individual frames.
<i>ExtractCoordinates</i>	Extract x and y coordinates in terms of the poster.
(ii) Data Summarization	
<i>ResizeImg</i>	Convert an input image into a user specified size.
<i>DrawAOIs</i>	Create rectangular AOIs and name the AOIs.
<i>GetAOITimelineData</i>	Categorize the coordinates by the visits of each AOI.
<i>GetPosterData</i>	Categorize the coordinates by each AOI.
(iii) Common Eye Tracking Visualization Tools	
<i>DrawEyeDotplot</i>	Create a dot plot indicating time spent on each AOI.
<i>DrawEyeBoxplot</i>	Create a box plot showing pupil radiuses in each AOI.
<i>DrawEyeBarplot</i>	Create a bar plot indicating time spent on each AOI.
<i>DrawEyeScatterplot</i>	Create a scatter plot showing focus points.
<i>DrawEyeHeatmap</i>	Create a heat map with Gaussian kernel function.
<i>DrawEyeAOITimelines</i>	Create an AOI timeline plot.
<i>DrawEyeScanpathMap</i>	Create a scanpath map.
(iv) Linked Microposter Plots	
<i>DrawEyeLMPlot</i>	Create a linked microposter plot.
<i>DrawEyeLTMPLOT</i>	Create a linked timeline microposter plot.
<i>DrawEyeLSMPlot</i>	Create a linked scanpath microposter plot.

3.1 Data Processing

Two functions are included in the package for data processing: *GetVideoFrames*, and *ExtractCoordinates*. Image processing and object detection algorithms are designed and applied to conduct the data processing. The details of the algorithms used in these functions are described in Li et al. (2017). The processing is conducted with the **Matlab** server through R using the **matlabr** package (Muschelli, 2016). **Matlab** 2014a or above has to be installed in the system and the right version of **Matlab** has to be specified in the **Matlab** path, in order to run the two functions.

The *GetVideoFrames* function breaks the AVI video into individual frames in all platforms. Other video formats, such as MP4, WMV, and MOV, can also be directly used in this function depending on the platforms one is using (The MathWorks Inc., 1994-2017). The user needs to specify the starting and ending seconds to break, the time interval between two video frames, and the output folder.

After the video has been broken into individual frames, individual images can be processed for the data extraction. As explained in Section 2, the locations of where the participant is looking at have to be extracted and transformed into coordinates with the same static coordinate system, such as the poster shown in Figure 1. To perform this task, image registration and object detection approaches are applied through the *ExtractCoordinates* function. A data set ready for further analysis is generated with the *ExtractCoordinates* function.

The *ExtractCoordinates* function extracts the x and y coordinates in terms of the poster from the video recorded with the mobile eye tracker. The main inputs of this function are a directory of the video frames and an electronic version or a clear photo of the poster in JPEG format (PNG and BMP format are also supported).

The output of the function is a CSV file with the x and y coordinates in terms of the poster.

3.2 Data Summarization

The data summarization functionality group includes the functions *ResizeImg*, *DrawAOIs*, *GetAOITimelineData*, and *GetPosterData* to prepare the raw eye tracking data for further analysis.

- *ResizeImg* is depending on the **imager** R package (Barthelme, 2016) and converts the input image into one with a user specified size. The function writes the output image into a JPEG file.
- *DrawAOIs* defines the AOIs with rectangles that are drawn by mouse clicking two vertices and the AOIs are named by the analyst through the **R** console right after each rectangle is drawn. The function creates a CSV file that contains names and border information of the AOI bounding boxes.
- *GetAOITimelineData* categorizes the eye tracking data by the visits of each AOI. The input files are a CSV file of the x and y coordinates and the pupil radius from the eye tracker, and the output CSV file from the *DrawAOIs* function. The pupil radius information is optional for the input. The user can specify a threshold for at least how many images (1/30 seconds for our eye tracker) the coordinates fall into a certain AOI to be considered as a visit. The output is a list that consists of a data frame with the length of visits for each visit, a list of eye movement speeds from one focus point to another in pixels for the visit at each AOI, and a list of pupil radiuses for each visit at the corresponding AOI (in case the pupil radius is provided in the CSV file).
- *GetPosterData* categorizes the eye tracking data by each AOI. It generates a data frame that summarizes how long the participant has spent and how many times the participant has visited each AOI, a list of eye movement speed from one focus point to another in pixels within each AOI and possibly one more list of pupil radiuses as described in the function *GetAOITimelineData*. The input files and options are also the two CSV files, i.e., the same as for the *GetAOITimelineData* function.

3.3 Common Eye Tracking Visualization Tools

Common statistical graphics introduced in Section 1, such as dot plots, bar charts, and box plots, are available in the **EyeTrackR** R package to visualize how much time the participant has spent on each AOI. The plotting options are passed on from the general graphic options in R. Other frequently used eye tracking visualizing tools, such as eye scatter plots, attention maps, AOI timelines, and scanpath visualizations are also implemented in the **EyeTrackR** R package.

The scatter plot (Figure 2) shows where the focus points are and is used to directly indicate individual eye movements. There are 1508 focus points presented in Figure 2.

Figure 3 shows an example of an attention map created by the *DrawEyeHeatmap* function. With the hot spots overlaid on top of the poster, it is quite obvious at which areas the participant is looking most. However, the sequential order of where one is looking at can not be shown in such a figure.

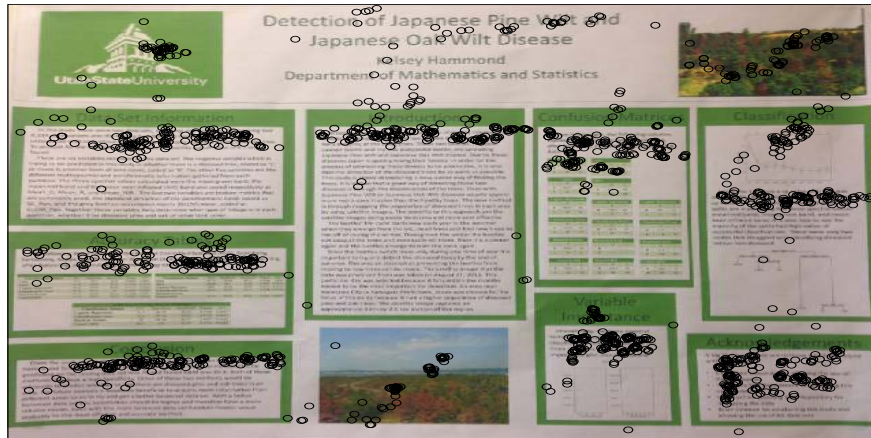


Figure 2: Scatter plot, overlaid on the photo of the original poster

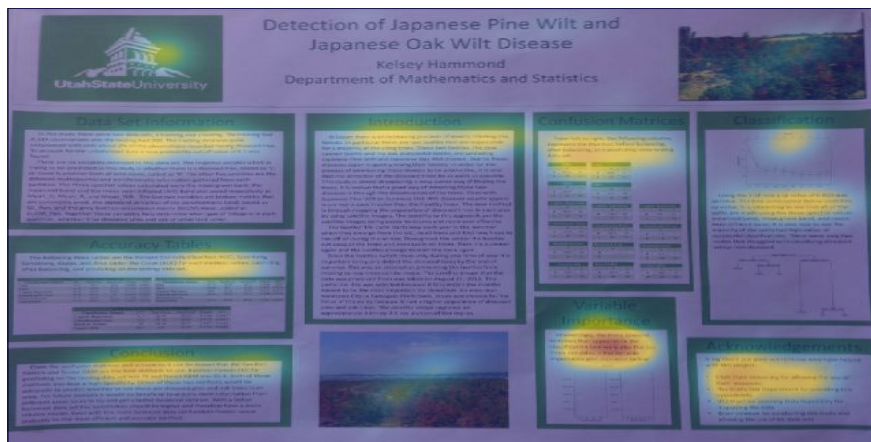


Figure 3: Attention map, overlaid on the photo of the original poster

Figure 4 shows the AOI timeline. We can see from the plot that the participant revisited the data and the title AOIs very briefly. The conclusion AOI has been visited twice as well, with a very short visit at the first time. However, the spatial information of the AOIs can not be shown in the AOI timeline. When the viewing patterns become complicated, it is very difficult to see the visual patterns from the AOI timeline.

Figure 5 shows the scanpath visualization. Fixations and saccades are identified with the **saccades** R package (von der Malsburg, 2015). The numbers in the circles indicate the sequential order of the fixations. Scanpaths give the sequence of one's eye movements, however, when the viewing patterns become more complex, the crossings and overlaps of scanpaths make it more difficult to perceive the visual patterns.

3.4 Linked Microposter Plots

The linked microposter plot is based on the idea of the linked micromap plot, a plot type that was first introduced in 1996 to highlight geographic patterns and associations among the variables in a spatial dataset (Carr and Pierson, 1996). It

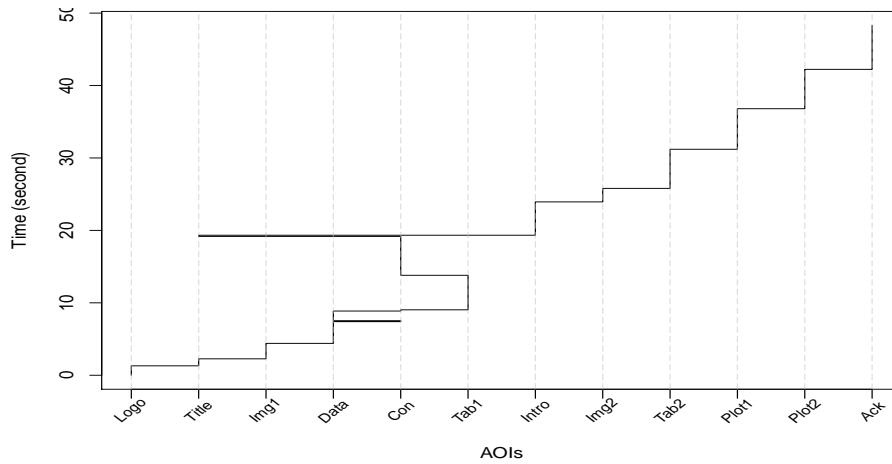


Figure 4: AOI timeline

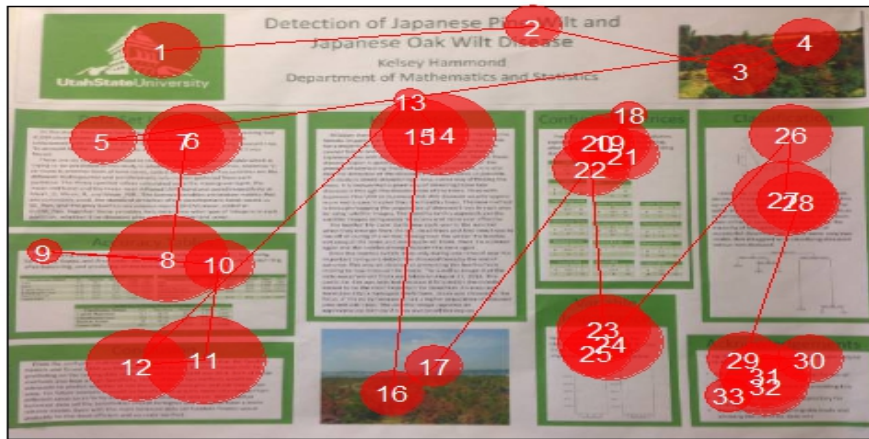


Figure 5: Scanpath map, overlaid on the photo of the original poster

has been widely used to display geospatially-indexed summary statistics. According to Carr and Pickle (2010), the micromap plot can represent any two-dimensional space, not just latitude-longitude on the Earth's surface. Based on this idea, we can think of a poster as a map and the AOIs of the poster as the different countries or states shown in a geographic map. The AOIs of the posters are the figures, tables, text areas, titles, etc. The length of time spent and the number of times each AOI is visited, eye movement speed and pupil radiuses are some of the variables of interest. Variables can be visualized with different plot types, such as dot plots, bar charts, and box plots, in different statistical data columns, all linked to the original poster and not isolated as in Figures 2, 3, 4, and 5. The same colors are used to link the areas in the small posters (microposters), the names of the AOIs, and the statistical data columns. Compared with an attention map or a scatter plot, the linked microposter plot is exploring selected AOIs, instead of simply looking at the hot spots. See Li and Symanzik (2016) for more details.

To create the linked microposter plot, the poster image, the AOI data created from the *DrawAOIs* function, and the summarized list of the eye tracking data from the *GetPosterData* function need to be provided. The input argument *columns.att* describes the content of each column in the plot. The column definition contains

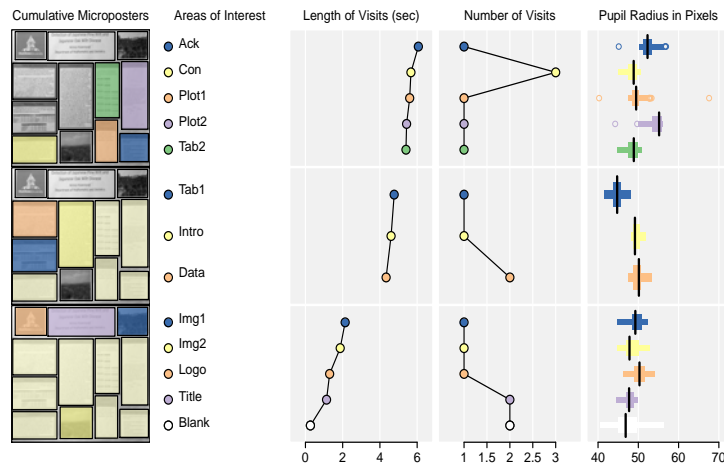


Figure 6: Linked Microposter Plot with three statistics columns

the number of the column, the data to be shown in this column, and the sizes and content of the labels, such as column title and axis labels. The *cumulate* option highlights the AOIs cumulatively from top to bottom across the panels with light yellow. Each list of the *columns.att* list describes a column in the plot. The structure is inspired by the **micromap** R package (Payton et al., 2015).

The function *DrawEyeLMPlot* allows the user to create the linked microposter plot. The user needs to read in the output files from the functions described in Section 3.2, and to specify plot types, the variables to visualize in each statistical column, and other options, such as the titles and panel width. The user can also change options like the highlighting colors, sorting variable, layout arrangements, and having a median row or not, if a median row is applicable. The *panel.types* input argument specifies the column types, including the poster, legend, boxplot, and dotplot. *panel.width* defines the width of each column. Once the panels and the layout are specified, defaults are acceptable for other details.

Figure 6 shows the linked microposter plot for the poster. The first column shows the microposters, the second column shows the color legend, and the third column shows the AOI names. The last three columns are the statistics columns. The gray AOIs are the AOIs that are not of interest in the corresponding panel. The light yellow shaded AOIs are the AOIs that have been investigated in the previous microposters above the current microposter. The rows are sorted by the length of visits. Each dot that represents an AOI is horizontally aligned with its AOI name and linked through color with the AOI on the microposter. The AOIs are separated into three perceptual groups. Perceptual groups typically contain between two and six of the AOIs. Such a design helps the readers focus on the values of a few mapped AOIs at once and allows to quickly identify clusters of mapped AOIs with similar values of the sorting variable (Carr and Pickle, 2010).

Unfortunately, the viewing sequences can not be shown in the linked microposter plot. To overcome this disadvantage, we developed the linked timeline microposter plot and the linked scanpath microposter plot to visualize the timeline and scanpath of the eye movements, respectively. Figure 7 shows an example of the linked timeline microposter plot sorted by the temporal viewing sequence, i.e., the data is presented based on the timeline of the eye tracking data. The numbers labeled inside the

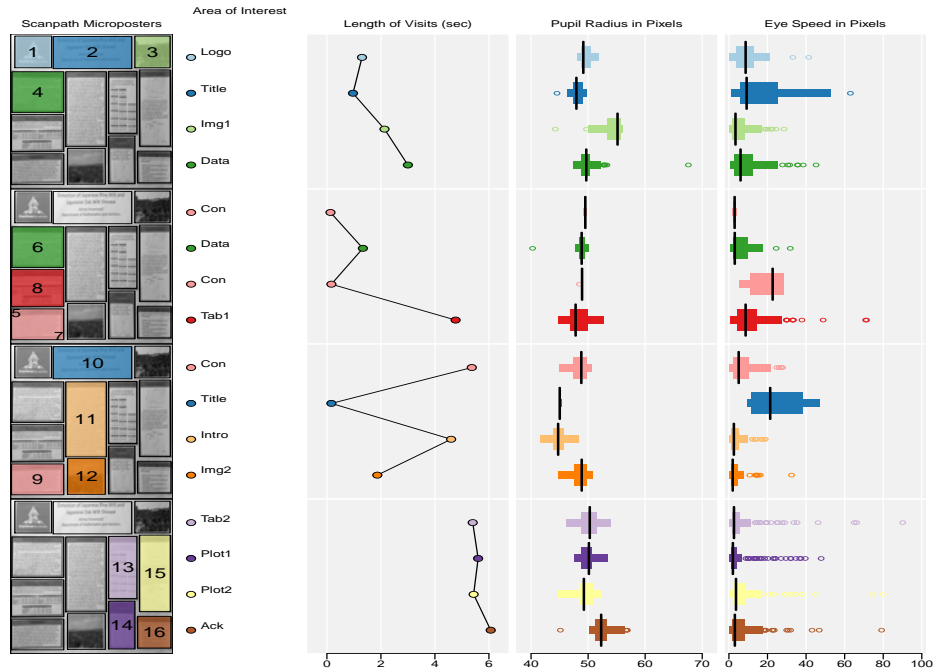


Figure 7: Linked Timeline Microposter Plots sorted by viewing sequence

AOIs indicate the viewing sequence. Each color represents an AOI. If the number of AOIs is more than the number of available colors, the colors are recycled, i.e., some AOIs are shaded with the same color. The AOIs can also be colored based on the panels similar to the linked microposter plot. Figure 8 shows the linked scanpath microposter plot. Instead of defining the AOIs, fixations are detected with the **saccades** R package. The statistical panels are based on the statistical summary of the fixations. Compared with an AOI timeline (Figure 4) and the scanpath visualization (Figure 5), the linked timeline microposter plot and the linked scanpath microposter plot can visualize multiple variables in one plot. Without all the AOIs and scanpath circles tangled in one single plot, it is much easier to see the visual patterns. For example, the revisits in Figure 4 of the title AOI is not very intuitive from the plot as well as how long the participant has spent on the revisit. In Figure 7, the revisit of the title is easier to view as the title is highlighted in blue and the color showed up twice labeled with number 2 and 10 in the microposter column. The length of the revisit is about 0.2 seconds from the linked timeline microposter plot, which is hardly readable from the AOI timeline.

The *DrawEyeLTMPlot* and *DrawEyeLSMPlot* functions have similar options as the *DrawEyeLMPlot* function. For the *DrawEyeLTMPlot* function, the additional options include *colorby*, which determines assigning each AOI a different color or just assigning different colors in the same panel. For *DrawEyeLSMPlot*, the additional options include the frequency of the eye tracker and the lambda that determines the threshold for the fixation detection.

The layout of the panels can be user-defined for the three functions, i.e., one can specify how many data points are in each panel. A median row can be added if a median is available. Below is an example of a self-defined layout with a median row enabled, where there are three groups of data points and each row has 6, 1,

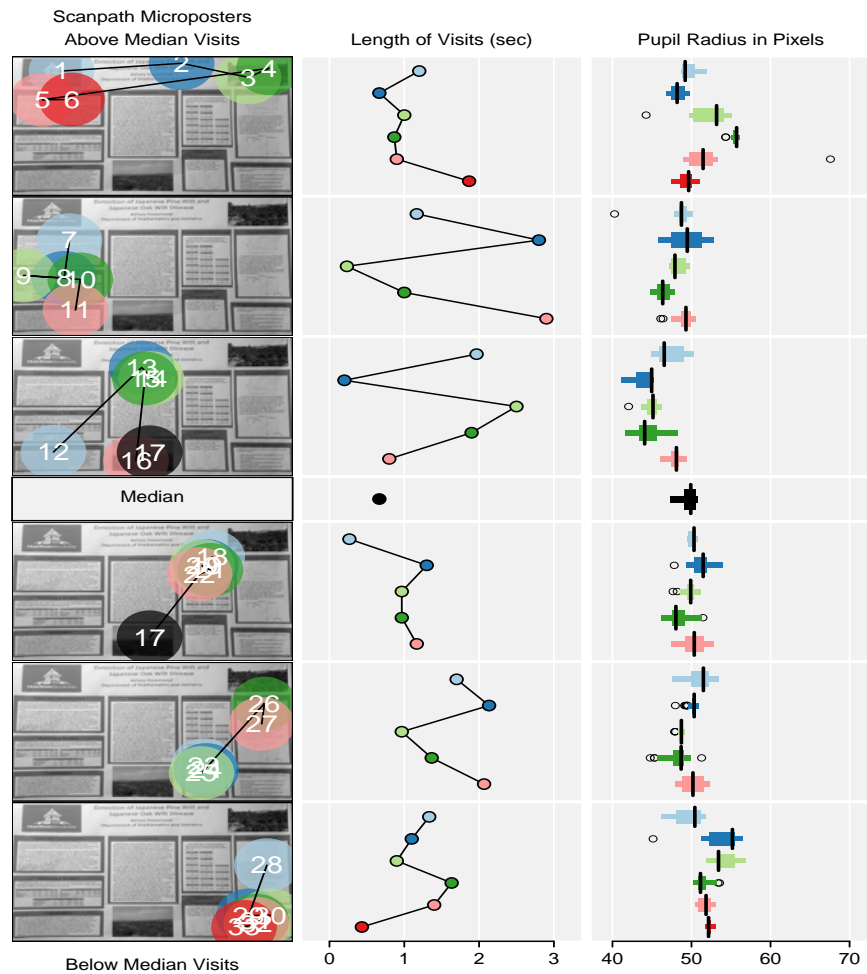


Figure 8: Linked Scanpath Microposter Plots sorted by viewing sequence

and 6 data points. The data presented in each column can be user-defined and the colors can also be changed. Figure 9 shows the resulting linked microposter plot that visually differs from the one shown in Figure 6, but is based on the same data.

4. Conclusion and Future Work

In this article, we presented the **EyeTrackR** R package and how it can be applied to process and visualize data from mobile and static eye trackers. If **Matlab** 2014a or above is installed, raw video data from a mobile eye tracker can also be processed with our **EyeTrackR** R package. The processed mobile eye tracking data can then be used for visualization and further analysis. For users without **Matlab** installed, the visualization tools can still be used to visualize processed mobile eye tracking data or data from static eye trackers from different manufacturers.

The commonly used existing eye tracking data visualization tools are implemented in the **EyeTrackR** R package. The new visualization tools we introduced,

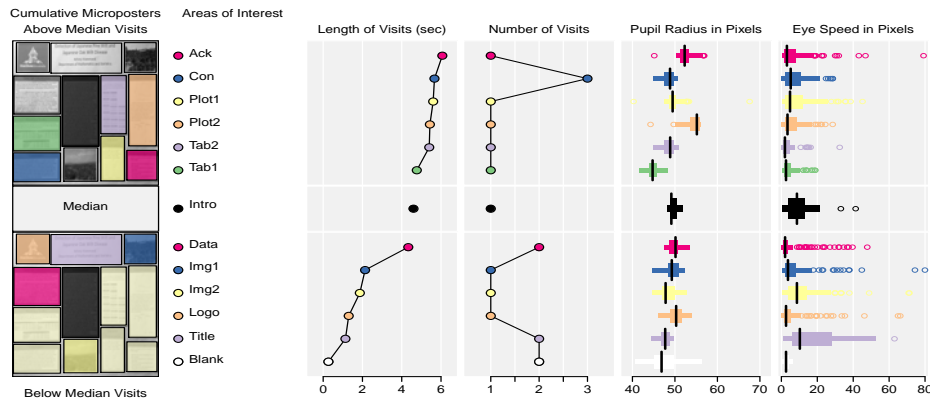


Figure 9: Linked Microposter Plot with four statistics columns

the linked microposter plot, the linked timeline microposter plot, and the linked scanpath microposter plot, are also implemented to visualize the eye tracking data. However, the application is not limited to visualize how people are looking at posters. The **EyeTrackR** video data processing, extraction, and visualization can also be applied when people are looking at power point slides, book chapters, commercial electronic posters, etc. Symanzik et al. (2017) extended the application on looking at power point slides to judge human postures by using the **EyeTrackR** R package for the data extraction and visualization. The visualization tools are also applicable for visualizing the extracted mobile eye tracking data or the data from static eye trackers.

Displaying data for multiple participants via a single linked microposter plot, linked timeline microposter plot, or a linked scanpath microposter plot will be added in the future. Dynamic microposter plots can also be implemented. The supporting **EyeTrackR** R package will be released on the Comprehensive R Archive Network (CRAN) in early 2018.

5. Acknowledgement

We would like to thank Bree Studenka for providing us with the mobile eye tracking device used for this research. We would also like to thank Kelsey Hammond for letting us use her poster.

References

- Andersson, P., April 2010. What is an effective layout for in-store posters? Case: accent — an accessories chain. Ph.D. thesis, Arcada University of Applied Sciences, Helsinki.
 URL https://www.theseus.fi/bitstream/handle/10024/14378/Andersson_Pauliina.pdf?sequence=1
- Applied Science Laboratories, 2015. ASL Results Plus GM.
 URL [http://www.razor3d.co.kr/eyetrac/pdf/ASL%20Results%20Plus%20GM%20\(Offline%20Analysis\).pdf](http://www.razor3d.co.kr/eyetrac/pdf/ASL%20Results%20Plus%20GM%20(Offline%20Analysis).pdf)
- Barber, P., Sanderson, M., Dickenson, A., 2008. Postar visibility research. School of Psychology Birkbeck College, London.

- URL <http://www.route.org.uk/document-library/postar-visibility-report-wave-4/>
- Barthelme, S., 2016. imager: Image Processing Library Based on 'CImg'. R package version 0.31.
URL <https://CRAN.R-project.org/package=imager>
- Berger, C., Winkels, M., Lischke, A., Höppner, J., 2012. GazeAlyze: a Matlab toolbox for the analysis of eye movement data. *Behavior Research Methods* 44 (2), 404–419.
- Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., Ertl, T., 2014. State-of-the-art of visualization for eye tracking data. In: Borgo, R., Maciejewski, R., Viola, I. (Eds.), *Proceedings of Eurographics Conference on Visualization (EuroVis)*.
- Caldara, R., Miellet, S., 2011. iMap: a novel method for statistical fixation mapping of eye movement data. *Behavior Research Methods* 43, 864–878.
- Carr, D. B., Pickle, L. W., 2010. *Visualizing Data Patterns with Micromaps*. CRC Press, Boca Raton, FL.
- Carr, D. B., Pierson, S. M., 1996. Emphasizing statistical summaries and showing spatial context with micromaps. *Statistical Computing & Statistical Graphics Newsletter* 7 (3), 16–23.
- Cooke, L., 2005. Eye tracking: How it works and how it relates to usability. *Technical Communication* 52 (4), 456–463.
- Dalmajjer, E. S., Mathôt, S., Van der Stigchel, S., Dec 2014. PyGaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments. *Behavior Research Methods* 46 (4), 913–921.
- Dink, J., Ferguson, B., 2015. eyetrackingR: An R library for eye-tracking data analysis. Github.
URL <http://www.eyetracking-r.com/>
- Foulsham, T., Kingstone, A., 2011. Look at my poster! Active gaze, preference and memory during a poster session. *Perception* 40 (11), 1387–1389.
- Goldberg, J. H., Helfman, J. I., 2010. Visual scanpath representation. In: *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. ACM, pp. 203–210.
- Gould, N., Zolna, J., Apr. 2 2010. Eye tracking and web usability: A good fit? *UX Magazine* (509).
URL <https://uxmag.com/articles/eye-tracking-and-web-usability-a-good-fit>
- Heminghous, J., Duchowski, A. T., July 2006. icomp: a tool for scanpath visualization and comparison. In: *ACM SIGGRAPH 2006 Research posters*.
- Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., Van de Weijer, J., 2011. *Eye Tracking: A Comprehensive Guide to Methods and Measures*, 1st Edition. Oxford University Press, Oxford, New York.

- Krassanakis, V., Filippakopoulou, V., Nakos, B., 2014. EyeMMV toolbox: An eye movement post-analysis tool based on a two-step spatial dispersion threshold for fixation identification. *Journal of Eye Movement Research* 7 (1), 1–10.
- Li, C., Symanzik, J., 2016. The linked microposter plot as a new means for the visualization of eye tracking data. In: 2016 JSM Proceedings. American Statistical Association, Alexandria, VA.
- Li, C., Symanzik, J., Zhang, B., 2017. EyeTrackMat: a Matlab toolbox for extraction of mobile eye tracking data with an application on people looking at posters (in preparation).
- Muschelli, J., 2016. matlabr: An Interface for MATLAB using System Calls. R package version 1.1.3.
URL <https://CRAN.R-project.org/package=matlabr>
- Noton, D., Stark, L., 1971. Scanpaths in eye movements during pattern perception. *Science* 171 (3968), 308–311.
- Payton, Q., Olsen, T., Weber, M., McManus, M., Kincaid, T., 2015. Micromap: Linked micromap plots. R version 1.9.2.
URL <https://CRAN.R-project.org/package=micromap>
- R Core Team, 2016. R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing.
URL <http://www.R-project.org/>
- Sogo, H., Oct 2015. Gazeparser/simplegazetracker.
URL <http://gazeparser.sourceforge.net/#>
- Symanzik, J., Li, C., Zhang, B., Studenka, B., McKinney, E., 2017. Eye tracking in practice: A first analysis of a study on human postures. In: 2017 JSM Proceedings. American Statistical Association, Alexandria, VA.
- The MathWorks Inc., 1994-2017. Videoreader.
URL <https://www.mathworks.com/help/matlab/ref/videoreader.html>
- Tobii Technology, 2017. Tobii pro studio.
URL <https://www.tobiiipro.com/product-listing/tobii-pro-studio/#Packages>
- von der Malsburg, T., 2015. saccades: Detection of fixations in eye-tracking data. R package version 0.1-1.
URL <http://CRAN.R-project.org/package=saccades>
- Zhegallo, A. V., Marmalyuk, P. A., 2015. Etran – R extension package for eye tracking results analysis. *Perception* 44 (8-9), 1–7.