12-2017

Extracting and Visualizing Data from Mobile and Static Eye Trackers in R and Matlab

Chunyang Li
Utah State University

EXTRACTING AND VISUALIZING DATA FROM MOBILE AND STATIC EYE TRACKERS IN R AND MATLAB

by

Chunyang Li

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Mathematical Sciences

(Statistics)

Approved:

Jürgen Symanzik, Ph.D. Major Professor	Adele Cutler, Ph.D. Committee Member
D. Richard Cutler, Ph.D. Committee Member	John R. Stevens, Ph.D. Committee Member
Breanna Studenka, Ph.D. Committee Member	Mark R. McLellan, Ph.D. Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2017

ABSTRACT

Extracting and Visualizing Data from Mobile and Static Eye Trackers in R and Matlab

by

Chunyang Li, Doctor of Philosophy

Utah State University, 2017

Major Professor: Jürgen Symanzik, Ph.D.
Department: Mathematics and Statistics

Eye tracking is the process of measuring where people are looking at with an eye tracker device. Eye tracking has been used in many scientific fields, such as education, usability research, sports, psychology, and marketing. Eye tracking data are often obtained from a static eye tracker or are manually extracted from a mobile eye tracker. Visualization usually plays an important role in the analysis of eye tracking data. So far, there existed no software package that contains a whole collection of eye tracking data processing and visualization tools. In this dissertation, we review the eye tracking technology, the eye tracking techniques, the existing software related to eye tracking, and the research on eye tracking for posters and related media. We then discuss the three main goals we have achieved in this dissertation: (i) development of a Matlab toolbox for automatically extracting mobile eye tracking data; (ii) development of the linked microposter plots family as new means for the visualization of eye tracking data; (iii) development of an R package for automatically extracting and visualizing data from mobile and static eye trackers.

(149 pages)

PUBLIC ABSTRACT

Extracting and Visualizing Data from Mobile and Static Eye Trackers in R and Matlab

Chunyang Li

Eye tracking is the process of measuring where people are looking at with an eye tracker device. Eye tracking has been used in many scientific fields, such as education, usability research, sports, psychology, and marketing. Eye tracking data are often obtained from a static eye tracker or are manually extracted from a mobile eye tracker. Visualization usually plays an important role in the analysis of eye tracking data. So far, there existed no software package that contains a whole collection of eye tracking data processing and visualization tools. In this dissertation, we review the eye tracking technology, the eye tracking techniques, the existing software related to eye tracking, and the research on eye tracking for posters and related media. We then discuss the three main goals we have achieved in this dissertation: (i) development of a Matlab toolbox for automatically extracting mobile eye tracking data; (ii) development of the linked microposter plots family as new means for the visualization of eye tracking data; (iii) development of an R package for automatically extracting and visualizing data from mobile and static eye trackers.

I dedicate this work to my mother, Lun Huang, and father, Yongzhi Li, who convinced me that accomplishing a doctorate degree is my fate.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Jürgen Symanzik for his patience in helping me with this research and my career as a statistician. I am very grateful for him spending a tremendous amount time with me. I appreciate his awesome example of academic excellence, enthusiasm, and work ethic. I greatly appreciate his sincere encouragement and support in all my decisions. I appreciate his excellent mentorship on me that he always helps me find what I want to achieve and helps me to achieve my goals.

I would like to thank my committee members. First, I would like to thank Dr. Adele Cutler, who was also my master's advisor. I sincerely appreciate her help and advice in my career and choices. I appreciate our long conversations before I had to make important decisions. I would like to thank Dr. Breanna Studenka for her help in this eye tracking study. I appreciate her meeting with me and helping me with everything in the lab. I would like to thank Dr. John Stevens for helping me with my course work and encouragement during my graduate study. I would like to thank Dr. Richard Cutler for his support for this eye tracking study. Most of the posters used in this study are from Dr. Richard Cutler's multivariate statistics class.

At last, I would like to thank my family and friends for giving me a tremendous amount of support.

Chunyang Li

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

This chapter provides an introduction to eye tracking, eye tracking visualization techniques, eye tracking software development, literature on eye tracking for posters and related media, and an introduction on image processing.

1.1 Introduction to Eye Tracking

This section introduces the eye tracking technology in Section 1.1.1, the underlying terminology in Section 1.1.2, and main application areas for eye tracking in Section 1.1.3.

1.1.1 Technology

This section introduces eye tracking equipment, such as the development of the eye tracking technology, working principles, different eye trackers, the leading manufacturers in the eye tracking field, setting up and calibration, and data recording.

Eye Tracking Equipment

Development: Eye trackers were first built in the late 1800s, however, they were very expensive and not comfortable for the participants (Holmqvist et al., 2011). Since Dodge and Cline (1901) used light reflected from the cornea, improved the precision, and reduced the invasiveness of the eye tracker in 1901, eye tracking techniques have been developed rapidly during the past century. A number of different techniques were developed and the eye trackers achieved high precision. But those techniques were either uncomfortable or expensive, and some of them were even mechanically complicated. In the late 1900s, companies driven by engineers, such as Applied Science Laboratories (ASL) and Tobii, began to offer eye tracking hardware to researchers, making eye tracking techniques more accessible and versatile. Holmqvist et al. (2011) gave a comprehensive review on the history of eye trackers as well as on the principles of how they work.

Working Principle: Eye trackers nowadays are mostly using the corneal reflection of an infrared light emitting diode to illuminate and generate a reflection off the surface of the eye (Cooke, 2005). The cornea is the part that covers the outside of the eye and reflects light. The corneal reflection is the brightest reflections of all the reflections from one's eye. Eye trackers based on this system are able to track pupils precisely, taking small head movements into account, thus making this video-based pupil and corneal reflection tracking method the dominating eye tracking method since the early 1990s. The pupil either appears dark or bright in the eye image, depending on the eye camera's focal axis. While the dark-pupil technique is most commonly used, both dark-pupil and bright-pupil techniques give the same data quality. The eye tracker we used in this dissertation research is based on the bright-pupil technique. The geometric centers of the pupil and corneal reflection are calculated to estimate where people are looking at. For this estimation, some examples of how points in the tracked area correspond to specific pupil and corneal reflection relations are needed. These examples are provided to the eye tracker by the process called calibration.

Monocular versus Binocular Eye Tracking: The majority of eye trackers are monocular and record data from one eye only. Some eye trackers are binocular and track both eyes. Monocular eye trackers are dominating. According to Holmqvist et al. (2011), there are two reasons why: First, even though it is not always the case, the movements of both eyes are approximately the same, making it unnecessary to measure both eyes simultaneously. The exceptions occur when recording children, participants with neurological dysfunctions, etc. Second, it is cheaper to manufacture monocular eye trackers. Therefore, unless it is for some special purposes, most of the eye tracking experiments are conducted using monocular eye tracking systems. The eye tracker we used in this dissertation research is monocular as well.

Sampling Frequency: The sampling frequency represents the speed of the eye tracking system and is one of the most important properties of the eye trackers. It is measured in hertz (Hz), which indicates how many times the eye tracker records the eye movements of the participant each second. The existing eye trackers' sampling frequencies range from a few Hz up to over 1000 Hz. Higher speed eye trackers are much more expensive, therefore, one needs to select the system depending on the research requirements. For research that requires more detailed study or detection of eye movement, a higher frequency eye tracker has to be used. In other cases, the low sampling frequency can possibly be compensated with more data. Typically, a system that has less than 250 Hz is considered a low-frequency tracker, otherwise, it is of high frequency (Holmqvist et al., 2011). The one we are using has a frequency of 30 Hz, i.e., it records 30 images of the eye movements each second.

Taxonomy: There exist two main categories of eye tracking equipment, static eye trackers and mobile eye trackers. Each of them requires different set-ups. Eye trackers from both categories follow the same basic working principle and data are usually stored in video format and a series of x and y coordinates.

- **Static Eye Trackers:** Static eye trackers are based on a desktop, hence they are often used to study eye motion on a computer screen. They are usually of lower price and higher durability and use a higher sampling frequency compared to mobile eye trackers. The series of x and y coordinates corresponding to the specific pixel on the computer screen are defined and positioned as part of the system set-up and calibration. Therefore, the x and y coordinates exactly refer to where the participant is looking at on the computer screen. This property of the static eye tracker makes the data analysis relatively simple. There are two types of static eye trackers: remote trackers and head mounted trackers. Remote trackers are mounted remotely on the computer, while head mounted trackers are mounted on the users' heads.

  Remote eye trackers measure the eye movement with a camera typically mounted underneath a computer monitor, without any contact to the participant. For some remote trackers, the participant's head has to keep mostly still during the eye tracking process, because the pupil reflections from multiple angles are not measured with the remote system (Cooke, 2005). Some remote trackers offer a head tracker with face recognition software to compensate for head movements, thus head movements are allowed and calculated for each devices (Applied Science Laboratories, 2013b). Remote eye trackers make it possible to record eye tracking data on infants and even on animals.

  Head mounted trackers are fixed on a user's head. These systems are able to measure eye movement from different angles, making it possible for users to move their head. The coordinates in the data file refer only to positions in the video scene and don't indicate where the participant is looking at on the screen. Some systems contain magnetic head tracking which calculates the motion of the head and adds that into the motion of the eye. These systems generate useful data files that allow for automatic data analysis, even if very large head movements occur. Head mounted systems are a good choice when the participant is looking at multiple surfaces or when it is required to move within a restricted area.

- **Mobile Eye Trackers:** Mobile eye trackers are head mounted as well, but they are more light weight and wireless. They are also known as wearable eye trackers. Users are not limited within a restricted area, so these eye trackers can be used for a variety of activities, such as playing soccer, driving, etc.

  Figure 1.1 shows the whole set of an ASL mobile eye tracker. There is a portable data transmit unit (DTU), a laptop with wireless reception connected to the DTU, and a pair of eye tracking glasses with optics. Babcock and Pelz (2004) described the steps of building a mobile eye tracker and explained the working principles of each part of the hardware. The eye tracking glasses are the main part of the mobile eye tracker. Figure 1.2 shows the components of the mobile eye tracker's head mounted optics. The eye camera records the tracked eye through the reflection of the "hot" mirror. The scene camera is aimed forward and records the environment observed by the participant. The eye and scene video can be recorded on an SD card by the DTU, can be transmitted directly to the PC, or both (Applied Science Laboratories, 2012).

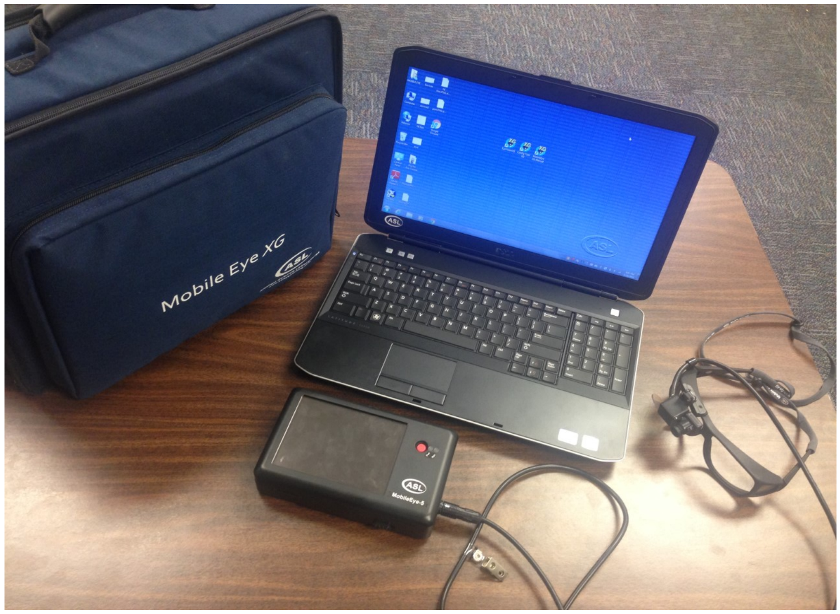

Fig. 1.1: An ASL mobile eye tracker.

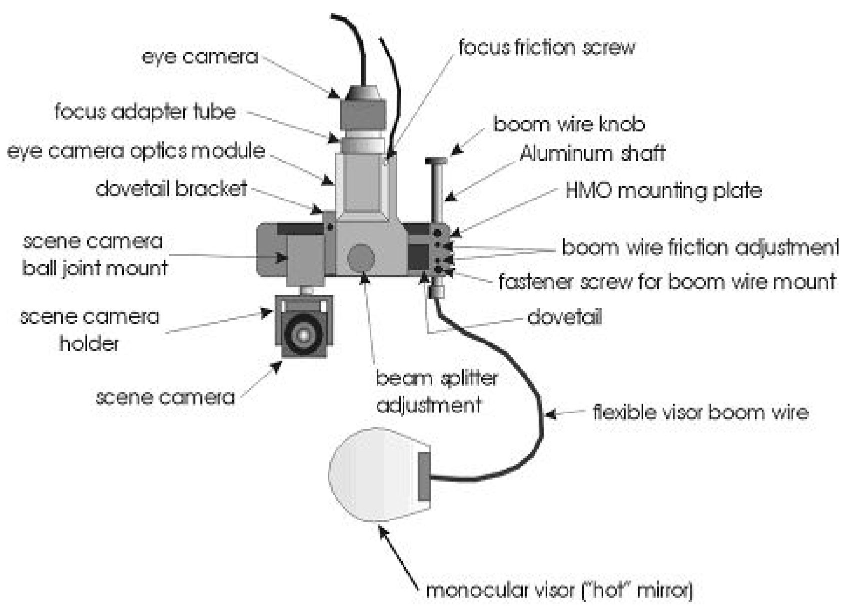

Fig. 1.2: Components of mobile eye tracker's head mounted optics. (Previously published in The DFKI Evaluation Center for Language Technology (2006)).

Manufacturers

There exist several companies that are building eye tracking hardware. According to Holmqvist et al. (2011), there were 23 companies that sold video based eye tracking systems in spring 2009. Over the past years, some manufacturers vanished, some new ones started. The leading manufacturers are Applied Science Laboratories (ASL), Tobii, SensoMotoric Instruments (SMI), and SR Research.

- **Applied Science Laboratories:** ASL was founded by M.I.T scientists in 1962 and is located in Bedford, Massachusetts. The first video based eye tracker was developed by ASL in 1974. ASL manufactures both, static and mobile eye trackers, that could be used in academic research, market research, and industrial markets. Their eye tracking data analysis software is provided together with the hardware (Applied Science Laboratories, 2013d).

- **Tobii:** Tobii was founded in Sweden in 2001 and developed rapidly during the past years. Offices have been built in the US, Germany, Japan, and China. Tobii has received world-wide recognition by building the first product that has assisted communication with eye tracking, and launching the first computer with built-in eye tracking. Nowadays, Tobii is collaborating with computer gaming companies, such as SteelSeries, Ubisoft, and Avalanche Studios, to integrate eye tracking into gaming accessories (Tobii, 2015a).

- **SensoMotoric Instruments:** SMI was founded in 1991 in Berlin, Germany. SMI provides products based on computer vision applications with a focus on eye tracking equipment. Its major fields of expertise are: eye and gaze tracking systems for research and industry application, high speed image processing and eye tracking. SMI products include data analysis software and mobile and static eye tracking systems (SensoMotoric Instruments, 2016).

- **SR Research:** SR Research manufactures EyeLink, including EyeLink 1000 Plus and EyeLink II (SR Research, 2013). There are many different options that one can choose to build an EyeLink 1000 Plus eye tracker, e.g. different options on a host computer, camera mount, and other accessories. EyeLink II used binocular eye monitoring. It has a frequency of 500 Hz, which has the fastest data rate and highest resolution of any head mounted video-based eye tracker. The EyeLink II can be used as either static eye tracker or mobile eye tracker. No mirrors are used in the EyeLink II system, which makes it relatively easy to set up.

Setting-up and Calibration

Preparation is an essential phase for conducting an eye tracking experiment. The participants need to be recruited following the ethics. The visual content, also known as the stimuli, need to be prepared, the eye tracker need to be set up and calibration has to be done to guarantee the experiment will work. The stimuli are mostly shown on the computer screen. They are typically images or videos. The stimuli can also be real-world situation. This is usually the case with mobile eye tracker. The setting up for different eye trackers is slightly different. There is usually an instruction manual on how to set up the specific eye tracker from the manufacturer. Once the equipments are set up and connected, we put the participant into the eye tracker and conduct the calibration. Calibration is typically done on a 2D area with some predefined points. The video quality should be checked and the calibration should be validated afterwards. Then the participants are instructed about the task and the recording can be started.

Setting up the equipment and follow the recording procedures correctly are important to make sure that the data obtained are of best quality. The steps for setting up eye tracking equipment and calibration are as follows:

- The visual content, also known as the stimuli, need to be prepared first. The stimuli are mostly shown on the computer screen. They are typically images or videos. The stimuli can also be real-world situation. This is usually the case with mobile eye tracker.
- Set up the eye tracker.
- Conduct calibration: Calibration is typically done on a 2D area with some predefined points. The video quality should be checked and the calibration should be validated afterwards.

Data Recording

Data recording is pivotal for all eye tracking based research. The data generated typically include a video, and a file that contains the x and y coordinates, pupil radius, mouse cursor position, etc. As stated in Section 1.1.1, the x and y coordinates are meaningful for most static eye trackers that are not influenced by the head movement and 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 are meaningless. In these situations, the video data are used for further analysis.

1.1.2 Terminology

Raw eye tracking data are usually not directly used in research. They are summarized with eye tracking metrics, based on which visualization techniques are applied and statistical analyses are conducted. Different eye tracking research uses different eye tracking metrics depending on the purpose of the research. A comprehensive review and detailed explanations of eye tracking metrics and terms can be found in Holmqvist et al. (2011). This subsection introduces the most frequently used eye tracking terminology and metrics according to the review of Holmqvist et al. (2011).

- **Fixation:** Fixation is the most frequently used metric in eye tracking data analysis. It refers to the state when the eye remains stable for a period of time. A fixation lasts from tens of milliseconds up to several seconds and aggregates an area of about 20 to 50 pixels on a computer screen. It can be detected using a velocity threshold or threshold of dispersion over a short period of time (e.g., 100-200 ms).

- **Area of Interest:** The Area of Interest (AOI) is defined by researchers as one or more area of the visual environment, based on the research interest. It is also known as the Region of Interest (ROI).

- **Gaze:** Gaze is the collection of spatial locations of a series of consecutive fixations within an AOI. It typically includes several fixations and some relatively short saccades between these fixations within an AOI (Jacob and Karn, 2003). When a fixation happens outside the AOI, the gaze ends.

- **Saccade:** A saccade is the rapid movement of the eye from one fixation to another. Saccades are the fastest movements the human body can conduct, and usually take 30-80 ms to complete. During most of the saccade, we are typically unaware of any visual information. The micro-movement during a fixation is called microsaccades and is used to understand human neurology. Saccades are not normally performed by taking the shortest path between two points. Instead, they take several shapes and curvatures. Therefore, researchers measure the amplitude, duration, and velocity of saccades.

- **Smooth Pursuit:** When the eyes are following some moving objects, e.g., an airplane flying across the sky, the slow movement of the eyes is called smooth pursuit. The velocity of smooth pursuits is around 10 to 30 degrees per second.

- **Scanpath:** A scanpath is a sequence of alternated fixations and saccades. It is the term originally introduced by Noton and Stark (1971) to refer to a fixed path that shows the characteristics of a specific participant and viewing pattern. Unlike a gaze, the saccades between fixations are relatively longer for a scanpath and a scanpath doesn't have to occur within one AOI.

1.1.3 Applications

Eye tracking has been widely used in various fields, both in academia and practices. Some of the main fields that adopted eye tracking are Education, Usability Research, Sports, Psychology, and Marketing.

Education

Eye tracking has been applied in different aspects of education, such as solving problems, classroom presentations, reading, and looking at graphics. With the implementation of eye tracking technology, researchers and educators have been able to confirm their research ideas on how to deliver knowledge more efficiently.

- **Problem Solving.**

  Nyström and Ögren (2012) used eye tracking to measure how students visually process vector calculus problems with illustrations and the same problems without illustrations. Line charts on proportion of fixations were used to visualize and compare all the students' different visual patterns in one illustrated and the same non-illustrated problem. The eye tracking data showed how the illustrations captured students' attention during the problem solving, however, the illustration didn't improve the overall performance. Tsai et al. (2012) also used eye tracking techniques to explore students' visual attention to solving multiple-choice problems.

- **Classroom Presentations.**

  Slykhuis et al. (2005) applied eye tracking technology to examine students' attentions on complimentary or decorative photographs on educational scientific PowerPoint presentations with and without audio narration. The mean and standard deviation of fixations on photographs were summarized and gazepaths were plotted to compare average students' visual patterns on slides with decorative photographs and complementary photographs, as well as slides with and without audio narration. Line charts were plotted to compare the fixation order of slides with and without audio narration. The eye tracking technique made it possible to confirm that students' devote more attention to highly relevant photographs on the PowerPoint slides. Similar research has been conducted, e.g., by Yang et al. (2013) who investigated university learners' visual attention to the PowerPoint presentations in a real classroom with eye tracking techniques.

- **Reading.**

  Eye tracking has been used extensively in reading since the 1970s, e.g., Adler-Grinberg and Stark (1978). According to Rayner (1998), research in this field that applied eye tracking technology includes language processing in reading, individual differences in reading, speed reading, dyslexia, music reading, visual search, etc. A comprehensive review of eye tracking for the past twenty years in reading and information processing can be found in Rayner (1998).

- **Looking at Graphics.**

The application of eye tracking on comparing different statistical graphics were introduced in Goldberg and Helfman (2010a) and Zhao et al. (2013). Goldberg and Helfman (2010a) examined the effectiveness of bar charts, line charts, and spider graphs with eye tracking data on information delivering. Scanpaths were used to visualize different visual patterns of the participants looking at the three different types of graphics. Zhao et al. (2013) assessed how lineup plots, such as histograms, densities, scatterplots and dotplots, were viewed with the eye tracking data. The viewing patterns were compared and visualized with scanpaths, timelines, dot plots, and bar charts.

Usability Research

Jacob and Karn (2003) provided a comprehensive review of eye tracking in human-computer interaction and usability research before 2003. Granka et al. (2004) employed eye tracking techniques “to gain insight into how users browse the presented abstracts and how they select links for further exploration”. Bar charts were used to visualize the time spent viewing each abstract together with the frequency the abstracts were selected. Cooke (2005) illustrated the application of eye tracking in reading behaviors online, searching, and scanning online information, as well as web page design. Google has also incorporated eye tracking into user studies to explore how users scan search results since 2005, when they obtained a Tobii eye tracker (Granka and Rodden, 2006). As the eye tracking technology has been more commonly applied in usability research, there are more and more references explaining how eye tracking can be used for usability research.

Sports

Many disciplines in sports have adopted eye tracking technology to study the basic technical mistakes in hand-eye coordination and how to optimize performance (Tobii, 2015b). These sports include soccer, table tennis, shooting, hockey, and baseball. Barfoot et al. (2012) used eye tracking systems to measure visual focus in archery. Du Toit et al. (2009) applied eye tracking in soccer to determine the visual skills of soccer players. More detailed reviews about eye tracking in sports can be found in Applied Science Laboratories (2013c).

Psychology

A variety of psychology fields have employed eye tracking to understand how people gather information visually and how information is processed. Eye tracking has helped researchers to reveal Autism at an early stage (Navab et al., 2012). Boraston and Blakemore (2007) provided a review on the application of eye tracking in autism studies with a focus on investigating gaze behavior of individuals with autism.

Marketing

Eye tracking is also a popular technique in marketing. There exist several references related to marketing using eye tracking technologies, such as Piqueras-Fiszman et al. (2012), and Purucker et al. (2013). Piqueras-Fiszman et al. (2012) defined the AOIs on the products and let the participants look at the images on the computer screen with a static eye tracker. Fixation data was analyzed in their study with a conjoint analysis to determine the attributes that affect the consumers' willingness to purchase the goods. Purucker et al. (2013) gave an overview of the literature using eye tracking for marketing and they proposed using scan statistics to analyze eye tracking data instead of using region of interests data analysis. The clusters of eye tracking data were identified and statistical tests were conducted on the spatial temporal data to compare designs of the car (Purucker et al., 2013).

Others

Other fields that made use of eye tracking technology include the military, civilian armed forces, 3D games and videos. More details can be found in Applied Science Laboratories (2013a).

1.2 Eye Tracking Visualization Techniques

This section summarizes the visualization techniques used for eye tracking data. There are numerous ways of visualizing eye tracking data. Blascheck et al. (2014) gave an overview of visualization of eye tracking data and classified the visualization techniques into three main categories: point-based techniques, AOI-based techniques, and techniques using both. Different taxonomies of plots are also illustrated in Blascheck et al. (2014). We summarize the main visualization tools for eye tracking data based on different graph categories. Common statistical graphics, scanpath visualization, timelines, and attention maps are the four main graph categories used for eye tracking data. The descriptions of the four graph categories and the plots belonging to each category are introduced in the following subsections.

1.2.1 Common Statistical Graphics

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.

- *Line charts* are used to visualize the proportion of participants looking at the AOIs, fixation duration versus the order of occurrence, etc. (Holmqvist et al., 2011).

- *Bar charts* are an intuitive technique to display eye tracking metrics. An example of employing bar charts can be found in Convertino et al. (2003), where the percentage of fixation time for four different visualization techniques are shown with a bar chart to compare the eye tracking data of people looking at a parallel coordinates plot, a scatter plot, and a geographic map.

- *Scatter plots* are employed when the relationship between two variables or two metrics of eye tracking data or the direct encoding of eye movements are of interest (Card et al., 2014; Berg et al., 2009). Berg et al. (2009) visualized amplitude and velocity measurements of saccadic movements for both, humans and monkeys, to show the differences between visions of the two species. In the visualization tool developed by Card et al. (2014), scatter plots were used to directly indicate individual eye movements.

- *Box plots* are used to show the statistical distributions of eye tracking data. Law et al. (2004) compared the difference in proportion of eye gaze on surgery tools between experts and novices during the surgery process with a box plot. Eye tracking metrics, such as fixation rate and fixation duration, can also be presented with box plots (Sharif and Maletic, 2010).

1.2.2 Scanpath Visualization

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, 2010b). The length of the line and the radius of the circle indicate the duration of the saccade and fixation. Figure 2.18 shows the scanpath representation, also known as gaze plot. 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.

Fig. 1.3: Scanpath. (Previously published on `https://bluekiteinsight.com/blog/how-eye-tracking-works/`).

1.2.3 Timelines

Point Data

Timelines are typically used to visualize temporal data (Blascheck et al., 2014). Goldberg and Helfman (2010b) added a time expansion to the visual scanpath representations, in which fixation positions are separated into x and y time expansion graphs. Figure 1.4 shows that the time expanded representations separated the tangled scanpaths. Figure 1.5 is an improved version of the time expanded representation. Multiple viewers' viewing patterns can be compared in the same plot (Grindinger et al., 2010).

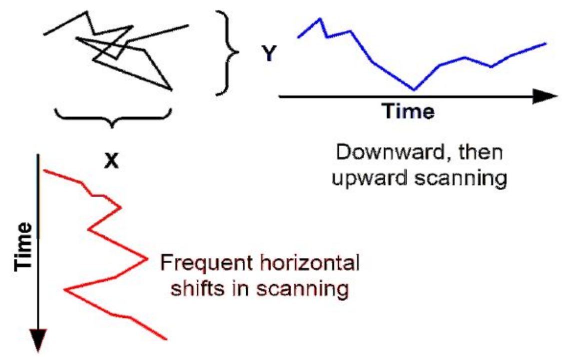

Fig. 1.4: Time expanded x and y representations of scanpaths. (Previously published as Figure 6 in Goldberg and Helfman (2010b), page 205).

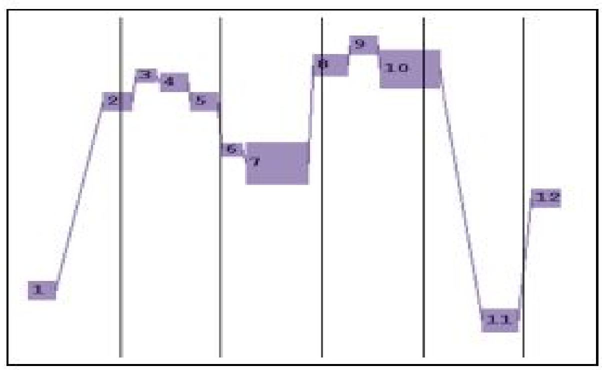

Fig. 1.5: Time-projected scanpath visualization, where the y-axis denotes vertical gaze position and the x-axis denotes time. "Vertical markers denote one-second intervals". The numbers represent the visual sequence and the size of the rectangles represent the relative length of time of the fixations. (Previously published as Figure 1 in Grindinger et al. (2010), page 101).

As an improvement, Li et al. (2010) proposed a Space-Time-Cube representation, making it possible to visualize eye movements with a timeline in both, horizontal and vertical, direction. A Space-Time-Cube representation is basically a 3D plot, in which the x and y axes represent horizontal and vertical movements and the z axis indicates the time (Li et al., 2010). Figure 1.6 compares a timeline, a scanpath, and a space-time-cube (STC) representation, where a timeline only gives the temporal pattern, a scanpath revels the spatial pattern but it is of potential overlap, and a STC representation shows both spatial and temporal pattern.

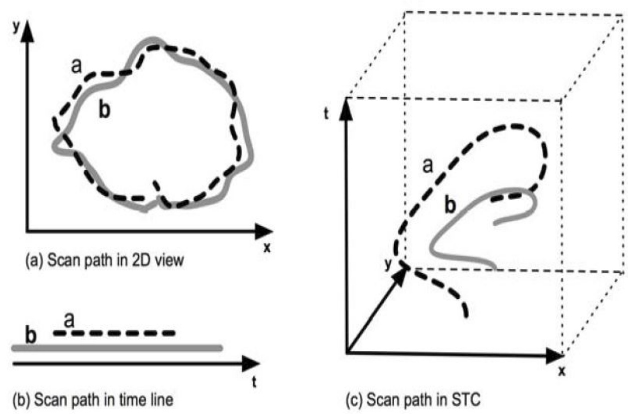

Fig. 1.6: Comarison of a timeline, a 2D scanpath, and a space-time-cube (STC) representation: the data are eye movements from two participants. (Previously published as Figure 3 in Li et al. (2010), page 302).

Area of Interest Data

Timelines are also used to visualize eye tracking data divided by AOIs. Räihä et al. (2005) used time plots to visualize scanpaths of AOIs. Followed by this idea, Raschke et al. (2012) proposed three visualization techniques using timelines combined with bar charts and line charts as parallel scanpath visualization. The three types of plots are gaze duration sequence diagrams, fixation point diagrams, and gaze duration distribution diagrams. With this parallel scanpath visualization technique, various properties of scanpaths, such as fixations, gaze durations and eye shift frequencies can be displayed in the same diagram (Raschke et al., 2012). According to Raschke et al. (2014), the three types of parallel scanpath visualization techniques are described below:

- In a *gaze duration sequence diagram*, the horizontal axis shows the area of interest (AOI) 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. Figure 1.7 shows a gaze duration sequence diagram.

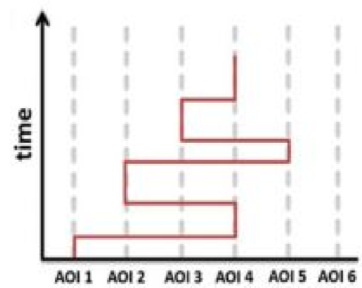

Fig. 1.7: Gaze duration sequence diagram. (Previously published as Figure 6c in Raschke et al. (2014), page 400).

- In addition to a gaze duration sequence diagram, a *fixation point diagram* adds single fixations inside AOIs as filled circle, e.g. AOI 2 starts with a fixation and there are three fixations during the gaze duration in AOI 2. The centers of respective gaze lines are connected. With more information added in the fixation point diagram, the frequency and number of fixations can be studied during a gaze duration in an AOI. Figure 1.8 shows an example of a fixation point diagram.

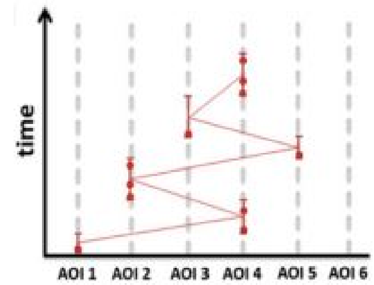

Fig. 1.8: Fixation point diagram. (Previously published as Figure 6d in Raschke et al. (2014), page 400).

- A *gaze duration distribution diagram* uses a line plot overlaid on top of a bar chart that shows the summation of the percentage of gaze durations in the areas of interest. A filled circle is the midpoint of a gaze duration and all the circles are connected to show the sequence of the gaze. An example of a gaze duration distribution diagram is shown in Figure 1.9

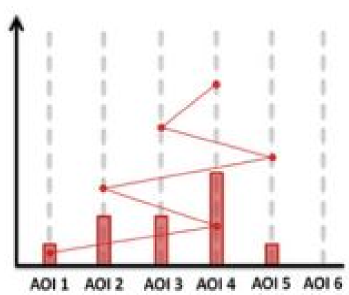

Fig. 1.9: Gaze duration distribution diagram. (Previously published as Figure 6e in Raschke et al. (2014), page 400).

Interactive versions of these diagrams also exist. Details are explained in Raschke et al. (2014).

1.2.4 Attention Maps

Holmqvist et al. (2011) provided a comprehensive review of attention maps. Attention maps are usually made of heat maps or hot spot maps with 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). Figure 1.10 left shows an example of an attention map. With the hot spots overlayed on top of a face, 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.

There are some other plot types developed based on attention maps. These plot types include difference maps and cluster images. The difference map is the map produced by taking the difference between two images with different view patterns (Caldara and Miellet, 2011). Figure 1.10 shows the plots that compare the difference between two viewers. A cluster image shows the percentages of participants who have viewed certain regions, instead of how long each section has been viewed by one participant (Andersson, 2010).

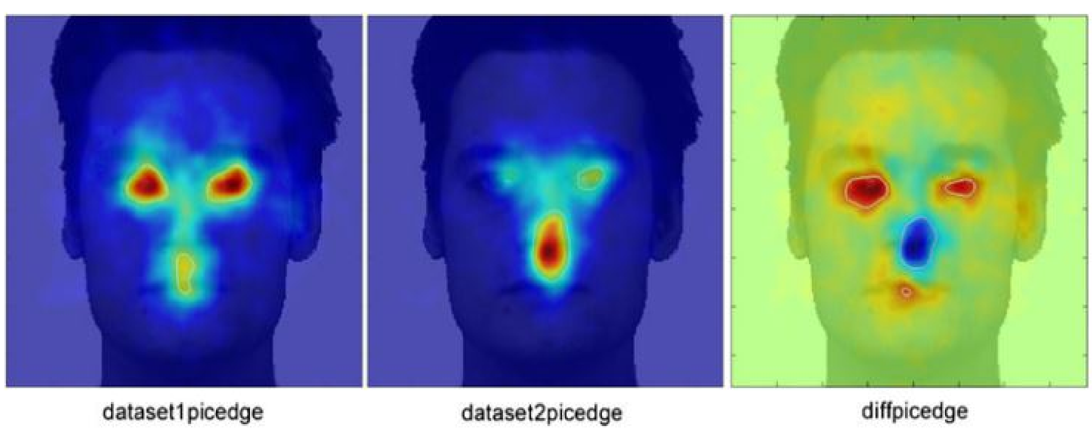

Fig. 1.10: Statistical fixation maps for two datasets and for their difference. (Previously published as Figure 4 in Caldara and Miellet (2011), page 870).

Other than the four most commonly used plot types summarized in this section, some other plot types are also used for eye tracking visualization: Foulsham and Kingstone (2011) used a schematic representation to summarize gaze transitions. Burch et al. (2014) overlayed attention maps with a hierarchical diagram to visualize the clusters of visual patterns and saccades.

1.3 Eye Tracking Software Development

This section provides an overview on the eye tracking software development. Most of the existing eye tracking software is developed with R packages, Matlab toolboxes and functions, and Python packages. A few software products, such as EyeC, iComp, and eTaddyare, are developed in other environments. To the best of our knowledge, the software reviewed in this chapter has been developed for data from static eye trackers or for data extracted from mobile eye trackers.

1.3.1 CRAN R Packages

Four R packages that are introduced in this section are available on CRAN.

gazepath: Gazepath Transforms Eye-Tracking Data into Fixations and Saccades

The *gazepath* R package (van Renswoude, 2015) provides a non-parametric speed-based approach to transform eye tracking data into fixations and saccades. The main function is called "gazepath", which conducts the transformations. The input dataframe is four by n, which includes the x and y coordinates, the stimuli name, and the distance to the screen. The gazepath function then transforms the input data frame in terms of fixations and saccades, based on the eye movement speed measured in degree/s.

saccades: Detection of Fixations in Eye-Tracking Data

This *saccades* R package (von der Malsburg, 2015) is using a velocity-based algorithm to detect eye fixations in raw eye-tracking data. The saccades are labeled when the velocity of the eye movement exceeds a certain threshold. According to (von der Malsburg, 2015):"Anything between two saccades is considered a fixation. Thus the algorithm is not appropriate for data containing episodes of smooth pursuit eye movements." This package is based on a data frame that consists of four columns: the x and y coordinates, the trial, and the time.

Three main functions are included in this package:

- *Calculate.summary* provides summary statistics such as the average number of fixations in trials, the average spatial dispersion in the fixations, and the average peak velocity that occurred during fixations.

- *Detect.fixations* converts the raw dataset to a data frame containing the statistics of detected fixations: the duration, the start and end time, the x and y coordinates, horizontal and vertical peak speed, and the standard deviation.

- *Diagonostic.plot* creates an interactive plot to show the raw samples and the detected fixations, so that the data can be screened and possible problems with the fixation detections can be diagnosed.

eyetracking: Eyetracking Helper Functions

Two functions are available in the eyetracking R package (Hope, 2012): One takes the x and y coordinates and returns the physical distance from the subject to the point on the screen; The other one takes two pairs of x and y coordinates and returns the angle subtended by the two points.

bdots

This *bdots* R package (Seedorff et al., 2015) analyzes differences among time series curves with a modified p-value technique proposed by Oleson et al. (2015). A 4-parameter logistic or an asymmetric Gaussian function is used to fit the eye tracking data.

1.3.2 Github R Packages

Three additional R packages that are introduced in this section are available on github.

eyeTrackerR

The *eyeTrackerR* R package (Godwin, 2012) is designed to generate fixation reports for eye tracking data. The two available functions are getting means for measures and events and contingencies. This package is at a very early stage of development.

Eyelinker

The *Eyelinker* R package (Barthelme, 2016b) is designed for reading data produced by Eyelink, which is a type of eye tracker manufactured by SR Research. It transforms Eyelink.asc files into R data structures containing raw traces, saccades, fixations, and blinks. Timeline plots can be produced with the Eyelinker R package.

eyetrackingR

This *eyetrackingR* R package (Dink and Ferguson, 2015) provides a connection of visualization and hypothesis testing tools for eye tracking data . The analyses include window analysis, growth curve analysis, onset contingent analysis, and estimating divergences. Plots are created with the *ggplot2* R package (Wickham, 2009).

- Window analysis is a collection of statistical tests and mixed-effects models on eye tracking data. It provides an initial look at the data and ascertains whether a subject looked at one object more than at another object.
- Growth curve analysis show the change of data over time. The data are summarized into time-bins and proportion-looking for each of the areas of interest (AOI) are calculated. Curves are fitted over the timecourse of the trial (See Figure 1.6) and bends in these curves are statistically assessed.
- Onset contingent analysis is used to examine how quickly participants looked to the referent AOI, i.e., to calculate the reaction times.
- Divergence is used to ascertain when a condition, such as an animated or inanimated image, had a significant effect during a trial.

1.3.3 Eyetracking in Matlab

Five Matlab toolboxes or functions are introduced in this section.

EyeMMV Toolbox

This *EyeMMV* Matlab toolbox (Krassanakis et al., 2014) can be used to detect fixation events, metrics analysis, data visualization, and ROI analysis. The fixation identification is based on an algorithm that has two spatial parameters and one temporal constraint. The visualization tools available in the *EyeMMV* toolbox are heat map, scanpath visualization and space-time-cube (See Chapter 2 for details). Krassanakis et al. (2014) further described the details of the functionalities in this toolbox.

Eyelink Toolbox

The *Eyelink* Matlab toolbox provides an interface for Matlab, a visual display programming toolbox called *Psychophysics* toolbox, and Eyelink (Cornelissen et al., 2002). Brainard (1997) introduced the *Psychophysics* toolbox, which interfaces Matlab and video display hardware.

GazeAlyze Toolbox

The functionalities of the *GazeAlyze* Matlab toolbox (Berger et al., 2012) include 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. Fixations, saccades, and movement path parameters are analyzed for ROIs and are included in the spread sheets generated. Graphical user interfaces are designed to control all the functions.

iMap Toolbox

iMap was introduced by Caldara and Miellet (2011) as an improvement of attention maps. *iMap* generates three-dimensional fixation maps with fixation data smoothed by convolving Gaussian kernels (See Section 1.2). The Gaussian kernel functions are either weighted by the number of fixations or their durations. *iMap* also produces difference maps (See Section 1.2) and uses robust statistics to compare conditions. Holmqvist et al. (2011) explained three ways of measuring the difference in attention maps in Section 11.3.4 of their book. In iMap, a priori segmentation of images or AOIs are not required. This characteristic overcomes the fact that it is difficult to define AOIs in some images, because the criteria can be too subjective. Matlab functions have been developed to generate iMap and compute the eye tracking data measures like number of fixations, the total fixation duration, the path length, and the mean saccade length.

Region of Interest Based Eyetracking Analysis

The Brain Imageing & Analysis Center (2014) provided functionalities for Matlab to let users draw Region of Interests (ROI) and create gaze paths. The gaze paths show how much the eye tracking data fall into the ROI.

1.3.4 Eyetracking in Python

Two Python packages are introduced in this section.

GazeParser

The *GazeParser* Python package (Sogo, 2013) provides a module to control the SimpleGazeTracker (Sogo, 2015), an open-source video-based eye-tracking application, and also various functions for data analysis, such as detecting saccades and fixations, plotting and comparing scan paths, calculating saccade trajectory curvature, etc.

PyGaze

PyGaze (Dalmaijer et al., 2014) combines several existing Python eye tracking packages including *PyGame*, *PsychoPy*, and *pylink*. EyeLink, SMI and Tobii eye tracking systems are supported with PyGaze. The basic functionality of PyGaze is to display visual stimuli and assess saccade and gaze behavior in real time. The documentation of the functionalities is explained in Dalmaijer (2013-2016).

1.3.5 Other Eye Tracking Software

EyeC

To compare the differences of visual patterns between subjects and groups of subjects, Ristovski et al. (2013) developed EyeC, an interactive visual analysis tool using coordinated views. Histograms of fixation durations for multiple people are created in the same plot with user selected AOIs according to the heat maps created (See Section 1.2). The interactive feature of scarf plots allows users to highlight a fixation and all other fixations over the same AOI will be highlighted automatically. Tree-structured visual representations are used for scanpaths, also with an interactive feature, to compare the differences between subjects. Ristovski et al. (2013) provided a detailed illustration of this visualization tool.

iComp

iComp is a visualization tool developed by Heminghous and Duchowski (2006), which implements quantitative scanpath comparisons and clustered eye gaze data of all viewers for each image display.

eTaddy

eTaddy (eyeTracking Analysis, conDuction and Designtool for user studies) was introduced by Raschke et al. (2014) to embed eye tracking metrics, statistical tests, and interactive parallel scanpath visualizations, introduced in Section 1.2.3.

1.4 Eye Tracking for Posters and Related Media

This section provides an overview on eye tracking usage on posters and related media. The literature includes eye tracking on posters in both simulated outdoor and indoor environments with computer screens. Commercial posters were investigated in both environments. Academic posters were studied with a mobile tracker in an indoor environment in a mock poster session to understand the attention of people during a poster session.

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. The influence of viewing distance and poster panel orientation were also investigated. According to Barber et al. (2008), the participants' eye movement data were recorded while they were looking at photographs of scenes with one or more poster panels. A 60 Hz remote static eye tracker was used to record the participant looking at the pictures on the computer screen. The x and y coordinates were obtained from the eye tracker to indicate gaze points. Poster panels were categorized into four environments in terms of their settings. The categories include driver-roadside, pedestrian-roadside, retail, and tube/rail. The eye fixations were classified as hits if they were on a poster panel. Therefore, the hit rates for the poster panels were calculated. The hit rates along with the poster panel properties, such as mean orientation of the scene, mean panel area, count of panels and mean panel distance, were summarized and compared in different scenes and each design category of interest.

Andersson (2010) looked at the effect of visual in-store advertisement designing on customers' decisions on purchasing. Eye tracking data were recorded to help answer the research question "How should the text and picture elements be positioned in in-store posters so that the message is understood as clearly as possible?" (Andersson, 2010). The participants' eye movement data were recorded while they were sitting in front of a computer screen. The author simulated two situations with computer images and let the participants look at the images presented on the computer screen given a certain amount of time. The first situation was when people were passing by the stores and the second situation was when people were in the store and paying attention to the posters. These situations were analyzed separately by Andersson (2010). Gaze plots were used to visualize experimental subjects' viewing patterns. Cluster images that show what components in the posters the participants have looked at were also used to compare different designs. Experimental subjects were interviewed about how many items they remembered in the posters. The research question was analyzed by evaluating people's visual patterns as well as summarizing the percentages of different items people remembered.

Foulsham and Kingstone (2011) also investigated how people were looking at posters in an indoor environment, but with a focus on academic posters from psychology. A mobile eye tracker was used to record participants' eye movements. However, to our best knowledge, it was not explained in the literature how the data recorded with the mobile eye tracker were extracted. Gaze transitions between different regions of the posters were visualized with arrows. Figure 1.11 shows how the eye tracking data were summarized in the visual representation. The summary statistics of gaze behaviors of all participants, such as, mean and standard deviation for proportion of fixation time on each components of the posters, probability of revisiting a poster, time spent on a poster, etc., were also summarized in tables. It was suggested in Foulsham and Kingstone (2011) that "participants spent the most time looking at introductions and conclusions". Larger posters and posters rated as more interesting (but not necessarily nicer in terms of aesthetics) were looked at for a longer time.

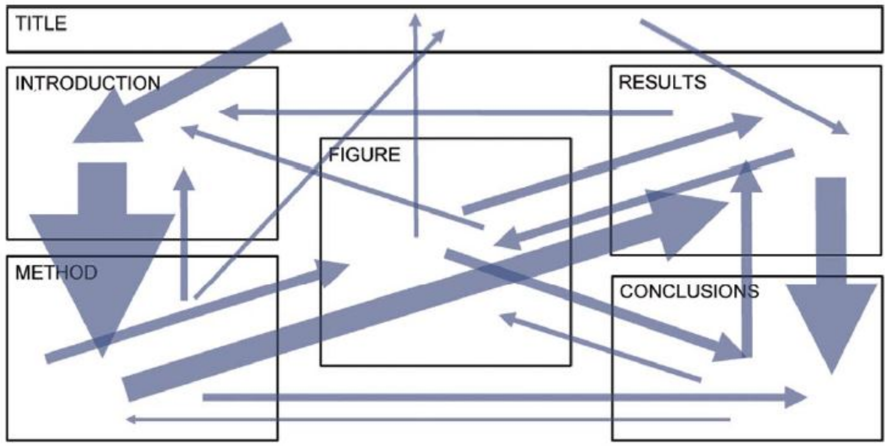

Fig. 1.11: Schematic representing the gaze transitions between regions across all participants and posters, where the size of the arrow represents the relative frequency. (Previously published as Figure 1 (b) in Foulsham and Kingstone (2011), page 1388).

Related research includes newspaper reading behavior (Holmqvist and Wartenberg, 2005) and looking at images (Judd et al., 2009). Holmqvist and Wartenberg (2005) used gaze plots as well as explorative multiple regression analysis to examine the effect of local design factors on readers' visual patterns. In Judd et al. (2009), eye tracking data were used to train a model of image saliency to predict fixation locations.

1.5 Image Processing

This section presents the terminology from image processing field in Section 1.5.1 and provides an overview of image matching algorithms in Section 1.5.2.

1.5.1 Terminology

The terminology from the image processing field that are used in this dissertation are summarized in this section:

- **Digital Image:**

  A digital image is a two-dimensional array that consists of small square regions known as pixels (Tanimoto, 2010). There are three basic types of images that are commonly used:

  - *Binary Image:*

    A binary image only consists of black and white color, therefore, there are only two possible values for each pixel. Such images are very efficient in terms of storage, because each pixel only needs one bit of storage space.

  - *Greyscale Image:*

    Each pixel of a greyscale image represents a shade of grey that typically ranges from 0 (black) to 255 (white).

  - *RGB Image:*

    Each pixel has a particular color, that is described by the amount of red, green, and blue, in an RGB image. Each of the red, green, and blue colors has a range of 0 to 255, which allows a large number of possible colors in an RGB image ($256^3 = 16,777,216$ possible colors). An RGB image is made of three matrices that represent the red, green, and blue values for each pixel. Each pixel on the image corresponds to three values.

- **Feature:**

  Features are typically noticeable specific locations in the images, such as mountain peaks, building corners, doorways, or interestingly shaped patches of snow. Szeliski (2011) provided an overview related to features. The most commonly used types of features are interest points or feature keypoints and edges. Interest points are often described by the appearance of patches of pixels surrounding the point location (See Figure 1.12 (a), (b), (d)). Edges can be good indicators of object boundaries and can be matched based on their orientations (See Figure 1.12 (c)).

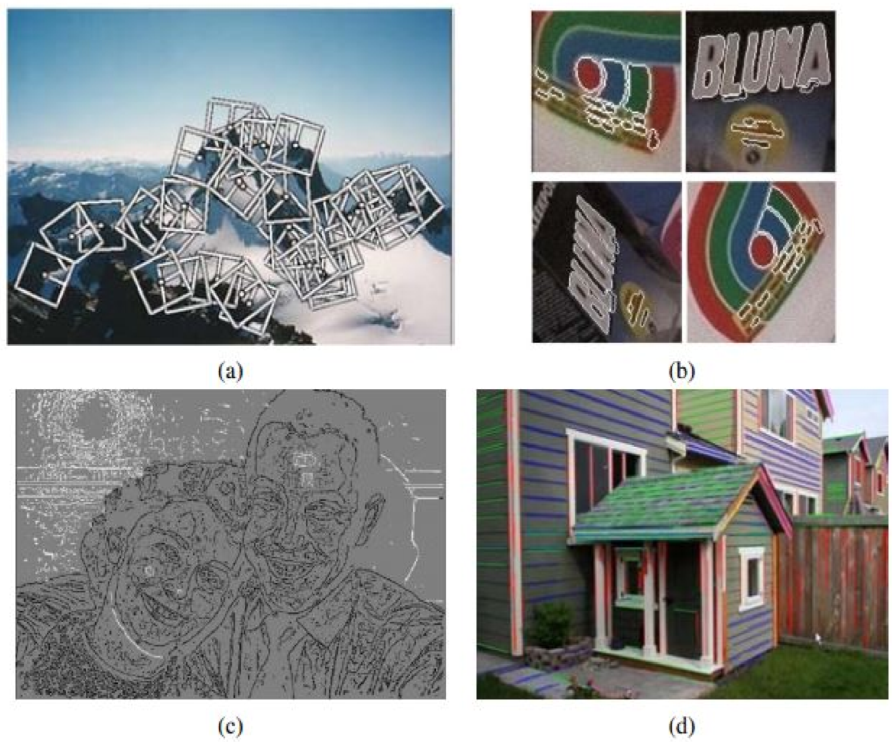

Fig. 1.12: A variety of features that can be used to describe and match images: (a) keypoint features; (b) region-like interest operators; (c) edges; (d) straight lines. (Previously published as Figure 4.1 in Szeliski (2011), page 212).

- **Template Image:**

  A template image is an image that consists of the pre-specified pattern or shape that is used for comparison with the original image (Institute of Electrical and Electronics Engineers, 1990).

1.5.2 Image Matching

Image matching is an important task in the computer vision field. There are two approaches to tackle this task: a feature-based approach and a template-based approach. The feature-based approach is effective when the template image has strong features. Given enough corresponding feature points, a transformation is then computed between two images. The most commonly used feature-based approaches are Scale Invariant Feature Transform (SIFT) (Lowe, 2004), Speeded up Robust Features (SURF) (Bay et al., 2008), Oriented fast and Rotated Binary robust independent elementary features (ORB) (Rublee et al., 2011), and Binary Robust Invariant Scalable Keypoints (BRISK). De Beugher et al. (2014) applied ORB to detect if one looked at specific objects in an eye tracking videostream. For template images without strong features, a template-based approach or template matching is a better option.

1.5.3 Speeded Up Robust Features (SURF)

The SURF algorithm was proposed by Bay et al. (2008) as a scale- and rotation-invariant detector and descriptor. The three main steps for the SURF algorithm are:

- Interest points, such as corners and T-junctions, are selected at different locations in the image. The detector is reliable such that the same physical interest points are detected under different viewing conditions.

- Constructing the local descriptors by representing the neighborhood of every interest point with a feature vector. The descriptor is invariant to view-point changes of the local neighborhood of the interest points, i.e., the descriptor is not changing based on the change of viewing distances and angles of the features detected.

- Matching of the descriptor vectors between different images based on the Euclidean distance between vectors. Filters, such as the random sample consensus (RANSAC) algorithm (Fischler and Bolles, 1981) or m-estimator sample consensus algorithm (Torr and Zisserman, 2000), are applied to remove outliers, thus the remaining matches more accurately correspond to the same scene viewed from different viewpoints.

Template Matching

Roberto (2009) provided a comprehensive overview of template matching. Template matching identifies the parts on an image that match a predefined template image. The template is put at every possible location, where some numeric measure of similarity is calculated between the template and the image segment it currently is overlapped with. The similarity can be measured with different metrics, e.g., sum of absolute differences (SAD), sum of squared differences (SSD), maximum absolute differences (MaxAD), image correlation, etc. SAD is one of the most commonly used metrics. Assuming the image is a greyscale image, SAD at location (x, y) of the original image is defined as follows:

$$SAD(x, y) = \sum_{i=0}^{nrow} \sum_{j=0}^{ncol} |I_o(x+i, y+j) - I_t(i, j)|,$$

Where $nrow$ and $ncol$ denote the number of rows and the number of columns of the template image, I_o denotes the intensity of the original image, and I_t denotes the intensity of the template image.

After obtaining the numeric measures of the similarities, the position that has the best similarity measure (highest for SAD) is the location of the pattern on the original image. This is considered as the naive template matching method. Advanced template matching algorithms, such as Korman et al. (2013), and Pereira and Pun

(2000), are able to locate the template on the images regardless of their orientation and size.

1.6 Goals of this Dissertation

Since eye tracking equipment has become more affordable and accurate nowadays, eye tracking has been employed in a variety of fields, as reviewed in Section 1.1.3. There exist several research projects on eye tracking usage on posters and related media (see Section 1.4). However, how people are looking at statistical posters hasn't been explored yet. Furthermore, most of the current research reviewed in Section 1.4 used the static eye tracker, which can provide meaningful x and y coordinates indicating exactly where one's visual focus is on a computer screen, as explained in Section 1.1.1. De Beugher et al. (2014) applied object, face, and person detection algorithms to automatically detect how often and how long a particular object was viewed. To the best of our knowledge, there is not any existing literature on automatic extraction of eye movement data for looking at posters from mobile eye trackers. Therefore, in this dissertation, we propose an automatic approach to extract mobile eye tracking data of where people are looking at on a poster. Advanced visualization techniques are developed to visualize eye tracking data more effectively. The **EyeTrackMat** Matlab toolbox and the **EyeTrackR** R package are developed for the data extraction and visualization. The three goals in this dissertation are described below.

1.6.1 Goal 1: Development of a Matlab Toolbox for the Extraction of Mobile Eye Tracking Data with an Application on People Looking at Scientific Posters

Eye tracking data is often obtained from a static eye tracker or manually extracted from a mobile eye tracker. We propose a new automatic way to extract meaningful eye movement data from a mobile eye tracker used by people that are looking at an object. In this dissertation, a scientific poster is used as an example. The local features of the poster and video frames are detected using the Speeded Up Robust Features (SURF) algorithm. The somewhat rotated recordings of posters in the frames of the eye tracking video are extracted and projected to a clearer version of the poster. The coordinates of the crosshair that represents the focus points of the human eye in the new coordinate system are detected using our new proposed heuristic object detection approach. Goal 1 of this dissertation is the development of our Matlab toolbox **EyeTrackMat**, with an application in extracting the eye tracking data of people looking at scientific posters from mobile eye trackers. Some basic visualization tools for eye tracking data are available in the toolbox to verify the validity of the data extracted. This work is presented in Chapter 2. A final version of this chapter will be submitted to an eye tracking journal such as the *Journal of Eye Movement Research*.

1.6.2 Goal 2: Development of the Linked Microposter Plots Family as New Means for the Visualization of Eye Tracking Data

Linked micromap plots have been widely used to visualize geographic patterns of regions and subregions. Based on the idea of linked micromap plots, we introduce three different types of linked microposter plots in Goal 2 of this dissertation (presented in Chapter 3) to visualize the eye movement pattern when people are looking at different components (such as headings, tables, figures, and different sections of text) of posters. These types are basic linked microposter plots, linked timeline microposter plots, and linked scanpath microposter plots. When using these the linked microposter plots, the eye tracking data of people looking at the various poster components can be better and more easily interpreted. The linked timeline microposter plots and the linked scanpath microposter plots are extensions of the basic linked microposter plots. The scanpath time series information is included in their visualizations. The linked microposter plot and its extensions provide several features that overcome some of the disadvantages of previously existing eye tracking data visualization methods. The linked microposter plots family can easily be extended to visualize how people look at webpages, power point slides, photos, etc. The preliminary work is published in Li and Symanzik (2016). A final version of this chapter will be submitted to a statistical visualization journal such as the *Journal of Computational and Graphical Statistics* (JCGS).

1.6.3 Goal 3: Development of an R Package for Extracting and Visualizing Data from Mobile and Static Eye Trackers

So far, there existed no R package that contains a whole collection of eye tracking data processing and visualization tools. Goal 3 of this dissertation (presented in Chapter 4) is the introduction of 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 dissertation is for people looking at scientific posters. The preliminary work is published in Li and

Symanzik (2017). An application on looking at power point slides to judge human postures by using the **EyeTrackR** R package for the data extraction and visualization is published in Symanzik et al. (2017b). A final version of this chapter will be submitted to a statistical software journal such as the *Journal of Statistical Software* (JSS).

CHAPTER 2
EYETRACKMAT: A MATLAB TOOLBOX FOR THE EXTRACTION OF MOBILE EYE TRACKING DATA WITH AN APPLICATION ON PEOPLE LOOKING AT SCIENTIFIC POSTERS

2.1 Introduction

Eye tracking is the process of measuring where people are looking at with an eye tracker device. Eye trackers were first built in the late 1800s and have been developed rapidly during the past century. Holmqvist et al. (2011) provided a comprehensive review on the history of eye trackers as well as on the principles of how they work. Eye trackers nowadays are mostly 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 meaningful scene videos indicating where people are looking at are generated.

Eye tracking techniques have been applied in a variety of research fields, such as behavioral sciences, education, marketing, and sports. There exist several literature reviews focusing on the application of eye tracking, e.g., Rayner (1998) provided a comprehensive review on eye tracking for the past twenty years in reading and information processing, and Jacob and Karn (2003) provided a comprehensive review of eye tracking in human-computer interaction and usability research. The eye tracking technology has become more and more affordable and accessible nowadays (Gould and Zolna, 2010). There also exists research on eye tracking 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, the eye tracking data in the existing research is often obtained from a static eye tracker or manually extracted from a mobile eye tracker. De Beugher et al. (2014) used the "oriented fast and rotated binary robust independent elementary features" (ORB-BRIEF) feature descriptor as an object detection technique to extract the objects of interest from an eye tracking video, making it possible to count how often and for how long a face or a person was viewed. However, there does not exist any literature for automatically extracting data from a mobile eye tracker for how people are looking at posters. We propose a new automatic method to extract eye movement data of where people are looking at the poster from a mobile eye tracker. We adopt and extend an algorithm that automatically extracts eye movement data from a recorded video. Our approach is based on feature detection and image registration algorithms.

Companies that design eye tracking technologies offer commercial software for eye tracking data analysis in general. However, this software typically only applies to the eye tracking hardware from the manufacturer and is not supporting any other eye trackers (Zhegallo and Marmalyuk, 2015). In addition, users have to manually go through every single video frame to extract data with the commercial software provided by the eye tracking company for mobile eye trackers. 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. However, none of the existing software developments has offered eye tracking data processing tools for mobile eye tracking data from people looking at an object, such as a poster. In this chapter, we introduce our new Matlab toolbox, **EyeTrackMat**, to process video data recorded from a mobile eye tracker, and to conduct some basic visualization tasks. Most of the existing Matlab tool boxes use a Graphical User Interface (GUI) for user friendly purpose, however, this design makes it more difficult for researchers to extend the toolboxes' functions for specific studies (Krassanakis et al., 2014). Our **EyeTrackMat** toolbox is made of a list of functions, therefore making it more convenient for people to modify them for any purpose of usage.

The remainder of this chapter is structured as follows: We will discuss the mobile eye tracking device and the collected data in Section 2.2. The data extraction procedures and algorithms are discussed in Section 2.3. The functionality of the **EyeTrackMat** toolbox is described in Section 4. An example of the application of the toolbox is introduced and the data quality is explored in Section 2.5. We will finish with our conclusion and discussion in Section 4.4.

2.2 Mobile Eye Tracking Device and Collected Data

There are two main types of eye trackers: static eye trackers and mobile eye trackers. 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. Figure 3.1 shows a full set of a mobile eye tracker equipment manufactured by Applied Science Laboratory (ASL). There is

Fig. 2.1: ASL mobile eye tracker equipment showing the DTU in the front, the eye tracking glasses to the right, and the laptop in the back.

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 (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 on the computer screen, etc. The ASL mobile eye tracker used in this chapter has a frequency of 30 Hz, i.e., the scene video data contains 30 frames per second. There is a label on each video frame indicating where the participant is looking at. The default and the nicest looking label of an ASL mobile eye tracker is a red crosshair. Other options include a circle and a huge cross that covers the whole screen. There are seven other options for the color of the crosshair: white, black, green, blue, magenta, cyan, and yellow. Since red is salient in most situations, a red crosshair is adopted in our data collections and illustration of the data extractions. The x and y coordinates are the center of the crosshair, which refers to where the participant is looking at in a particular video frame.

2.3 Data Extraction

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 can not be utilized directly, because they correspond to different frames, i.e., the coordinate system is changing for every single frame. Therefore, we use the object of interest (a scientific poster in this chapter) as a standard to unify the coordinate system. The coordinates from different coordinate systems are all transformed to the coordinates in the same coordinate system in terms of the object of interest, therefore, they become comparable and can be directly utilized. In this chapter, we transform all the coordinates to the same coordinate system in terms of a poster, thus the transformed coordinates can be used for further data analysis.

Given a video collected from a user wearing the mobile eye tracker to look at the poster and an electronic version or a clear photograph of the poster (the original poster), we are able to generate a CSV file with x and y coordinates in terms of the poster. Here, an electronic version of the original poster is used to illustrate the procedures. We have tested our approach on both, an electronic version and a clear photograph, for many posters and the results are similar. To complete this task, the video is first broken into a sequence of consecutive images or video frames. Figure 3.4 shows two examples of video frames extracted from a video of a user looking at the poster. As explained in Section 2.2, the location of the crosshair indicates the user's focus point. However, the coordinate systems in the two frames are different, therefore they need to be transformed into the same coordinate system. To transform the coordinates into the same coordinate system, the following three steps are applied: 1) Detecting the local features of the video frames and the original poster using speeded

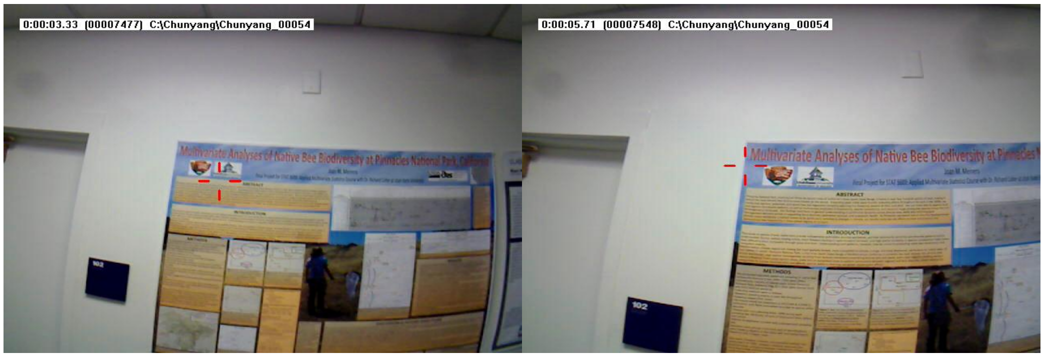

Fig. 2.2: Two examples of the original frames from the video

up robusted features (SURF); 2) Adopting an image registration approach to map the rotated and distorted poster onto the original poster; 3) Converting the given coordinates to the coordinate system in terms of the poster, or applying an object detection approach to locate the red crosshair in the unified coordinate system if a video is given without the coordinates of the crosshair.

2.3.1 Local Feature Detection

Local features are typically noticeable specific locations in the images, such as mountain peaks, building corners, doorways, or interestingly shaped patches of snow. "A local feature is an image pattern which differs from its immediate neighborhood" (Tuytelaars et al., 2008). A feature detector is able to provide a representation that allows local features to match effectively between images (Hassaballah et al., 2016), therefore, they should have the following properties:

- Robustness: The feature detector should be robust against scaling, rotation, and deformation;

- Repeatability: The features should be detected under different viewing conditions and different scenes, i.e., having the same locations when putting them back in the detected images;

- Informativeness: The detected features should be able to show enough variations, therefore can be used to distinguish different scene and objects, and be applied in tasks such as image registration;

- Quantity: A reasonable number of features should be detected in an image, i.e., the number of the features should be sufficient enough to provide a compact image representation that reflect the information content;

- Efficiency: The detection of features should be quick enough for time-critical applications.

The feature-based approach is effective when the template image has strong features. Since the poster images usually have salient local features, such as the textboxes, images, edges and corners, it is feasible to apply local feature detectors to extract features for image registration. Given enough corresponding feature points, a transformation is then computed between the images.

There are a variety of algorithms to detect and describe local features, such as scale invariant feature transform (SIFT) (Lowe, 2004), speeded up robust features (SURF) (Bay et al., 2008), oriented fast and rotated binary robust independent elementary features (ORB) (Rublee et al., 2011), and binary robust invariant scalable keypoints (BRISK) (Leutenegger et al., 2011). In this paper, SURF is used to conduct the feature detection task because of its efficiency and accuracy.

SURF was proposed by Bay et al. (2008) as a scale- and rotation-invariant detector and descriptor. The two main steps for the SURF algorithm are:

- Selection of interest points, such as corners and T-junctions, at different locations in the image. The detector is reliable that the same physical interest points are detected under different viewing conditions.

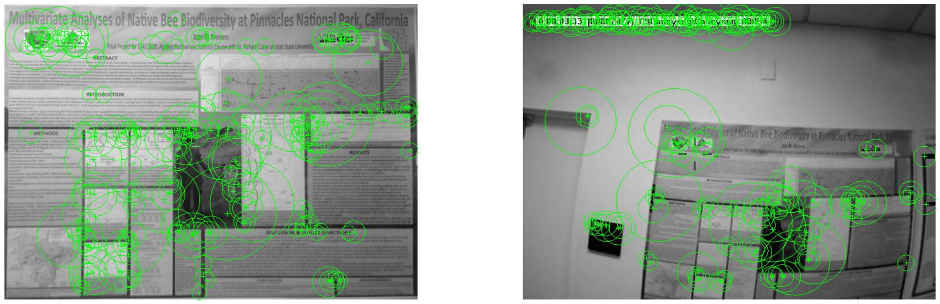

Fig. 2.3: The 200 most salient features on the original poster (left) and on one of the video frames (right)

- Construction of the local descriptors by representing the neighborhood of every interest point with a feature vector. The descriptor is invariant to view-point changes of the local neighborhood of the interest points.

Figure 2.3 (left) shows the 200 most salient features detected with the SURF algorithm on the original poster that is provided as a clear photograph of the poster, and Figure 2.3 (right) shows the 200 most salient features detected on the left video frame from Figure 3.4. The radius of the circle represents the scale (or the size) and the center represents the location of the feature.

2.3.2 Registration

After the feature descriptors are extracted, the descriptor vectors between different images are matched based on the Euclidean distance between vectors. Two feature vectors match when the distance between them is less than the matching threshold. The matching threshold can be adjusted. An increase of the matching threshold results in more matches; and vice versa. After the matching, filters, such as the random sample consensus (RANSAC) algorithm (Fischler and Bolles, 1981) or the m-estimator sample consensus algorithm (Torr and Zisserman, 2000), are applied to remove outliers. Thus, the remaining matches are more accurate and correspond

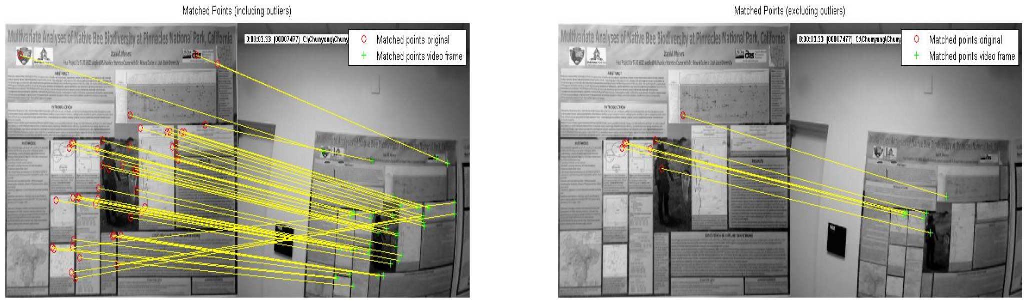

Fig. 2.4: All matched features (left) and matched features with outliers excluded (right)

to the same scene viewed from different viewpoints. We are using the m-estimator sample consensus algorithm to exclude outliers.

Figure 2.4 (left) shows the matched features using their descriptors: the round red circles indicate the feature points detected on the original poster and the green crosshairs indicate the feature points detected on the distorted poster in the video frame. Then outliers are removed using the statically robust m-estimator sample consensus algorithm, therefore spurious matches are eliminated and the remaining matches are relatively more reliable. Figure 2.4 (right) shows the matchings with the outlier removed from Figure 2.4 (left). After matching, the scale and angle of the distorted poster compared to the original poster are calculated. A geometric transformation is then performed on the distorted poster to recover the distortion from the original one. Figure 3.5 (right) shows the image registration result after the geometric transformation. The new coordinate of the crosshair in terms of the original poster is calculated and plotted in Figure 3.5 (left). The calculation methods are described in Section 2.3.3.

2.3.3 Determining the Coordinates in the New Coordinate System

If the coordinates of the crosshairs in the video frames are provided, the locations

Fig. 2.5: Image registration: the clear photograph of the poster with the calculated coordinate of the crosshair (left) and the transformed poster from the video frame (right)

of the crosshairs in terms of the poster can be calculated based on the scales and angles from the geometric transformation. However, if the eye tracking video is provided without the coordinates of the crosshairs, further processing is needed for the calculation. We developed a heuristic object detection approach to locate the crosshair based on its features.

Extracting the Crosshair Color

To detect the location of the red crosshair, the red objects are first extracted based on the red, green, and blue (RGB) values (Garg, 2015). However, there are several noises due to the other red components in the poster. Figure 2.6 (left) shows the extracted red components from the left video frame from Figure 3.4. To locate the crosshair, the noises are removed first and an object detection approach is applied thereafter.

When the poster has too many red components, changing the color of the crosshair may be necessary. If the crosshair has a different color, the objects with that color in the image are extracted instead. The user has to change the code provided by the toolbox function to extract a different color other than red, however, the working and coding principle remains the same. For all the posters we have experimented with, even with lots of red and orange (the colors that have very high "R" values) components in them, the following algorithm still works effectively.

Removing the Noises

For all the connected components of the image, bounding boxes are created and the length/width ratios are calculated. If the ratio is too small (smaller than 2) or too large (larger than 7), the connected component is removed. The lower and upper thresholds of the ratio, i.e., 2 and 7, are determined based on the shape of the legs of a standard crosshair. After removing the connected components whose shapes are too different from the legs of the crosshair, additional noises are removed, based on the major axis length of the ellipse that is tangential inside the bounding box. If the length of the major axis is too small (smaller than 20 pixels or half of the median of all the major axis lengths) or too large (larger than 80 pixels or two times the median of all the major axis lengths), the component is removed. The two thresholds, i.e., half and twice of the median (or 20 and 80 pixels), were determined based on empirical experiments. If there are several noises and most of their sizes are much smaller or much larger than the leg of the crosshair, a fixed threshold is used, such as (20, 80). If some of the noises have sizes larger than the crosshair and some have sizes smaller than the crosshair or if there are not many noises, a dynamic threshold is used, such as (0.5*Median, 2*Median). This process is repeated until all the major axis lengths are similar to a standard crosshair so that no additional components can be removed. The next step is to find the pairwise distances between the centroids of all the remaining components. In particular, the components whose pairwise distances with all other components are between 0.3 and 4 times the median of the major axis lengths are removed. Table 2.1 shows the summary of the thresholds applied in the

Table 2.1: Threshold applied in removing the noises (M is the median major axis length of all the remaining connected components)

Feature	Length/width ratio	Length	Distances
Threshold	(2, 7)	(20, 80) or (0.5*M, 2*M)	(0.3*M, 4*M)

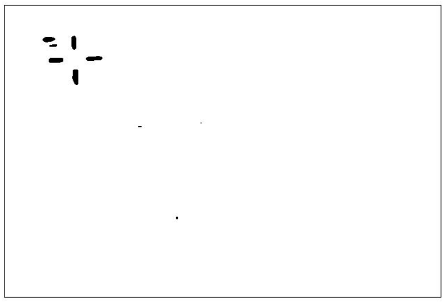

Fig. 2.6: Extracted red objects in the image before (left) and after (right) removing the noises. (The black connected components are the foreground and the background is white.)

noises filtering procedure. The thresholds can be changed easily in the code provided by the toolbox. Figure 2.6 (right) shows the remaining components after the noises removal procedure.

Finding the Center of the Crosshair

The orientations of the remaining components in the image are detected. Orientation is measured by the degree of the angle between the x-axis and the major axis of the ellipse that is tangent inside the bounding box of the component. The range of the angle is from -90 to 90 degrees. Figure 2.7 (left) shows the ellipse that covers the connected component and Figure 2.7 (right) shows how the orientation is measured: the angle between the longer solid line representing the major axis of the ellipse and the dashed line representing the x-axis.

The absolute value of the orientations of the components are binned by 10 degrees from 0 to 90 degrees. There are nine bins in total. The components with

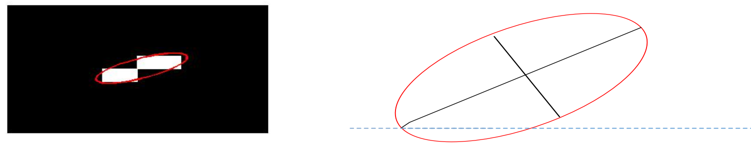

Fig. 2.7: The connected component (left) and its definition of orientation (right)

the orientations falling into the same bin are considered to have similar orientations. (Their orientations are different by 10 degrees at most.) If there are four remaining components that are adjacent and two of their pairs have similar orientations, then the location of the crosshair is the spatial mean of the two centroids of the pair that has the smaller difference in their orientations. If there are more than two remaining components in a bin, i.e., at least two components that are not the legs of the crosshair are in the same bin, the orientation differences between the components are calculated and compared with the differences in other bins. If there are noises, the differences in the orientations of the components are most likely to be larger than the differences in the bins with just the crosshair. Figure 2.8 (top left) shows the remaining components after removing the noises. There are four horizontal components and two vertical components that are in two different bins. The pairwise differences of the four horizontal components' orientations sum up to be 7.89 degrees, which is larger than the difference between the two vertical components' orientation (0.99 degrees). The spatial location of the crosshair is then determined by averaging the centroids of the two adjacent vertical legs.

If there are three legs visible on the poster, the location of the crosshair is calculated by taking the spatial average of the legs that have similar orientations, either the two vertical legs or the two horizontal legs. Figure 2.8 (top right) shows how the spatial location of the crosshair is determined by averaging the centroids of the two vertical legs.

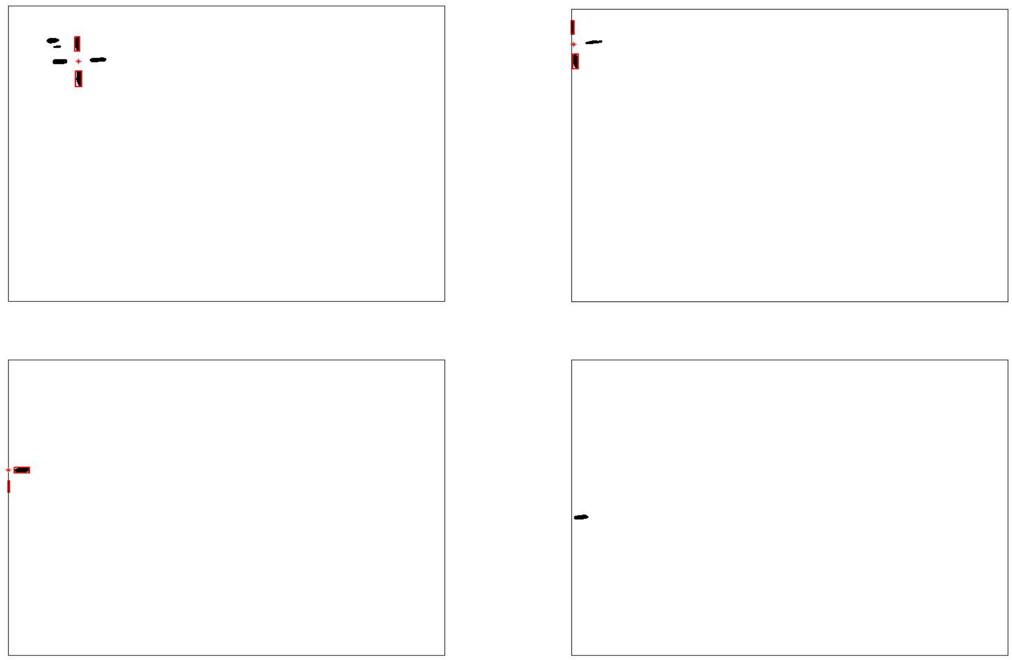

Fig. 2.8: Detecting the location of the crosshair with four visible legs (top left), with three visible legs (top right), with two visible legs (bottom left), and with one leg (bottom right). The red bounding boxes are indicating the legs that are used to calculate the center of the crosshair. The red mark in the middle is the calculated coordinates of the crosshair by averaging the centroid of the two legs with bounding boxes.

If there are two legs visible on the poster, the intersection point of the two perpendicular axes is the spatial center of the crosshair. Figure 2.8 (bottom left) shows how the spatial location of the crosshair is determined by taking the intersection point of the axes of the two vertical legs. If the intersection point is beyond the poster boundary, then the crosshair is considered invisible on the poster. If there is only one leg visible on the edge of the poster, as shown in Figure 2.8 (bottom right), the crosshair is considered to be invisible on the poster.

The images in this section, except Figure 3.5, are generated for illustration purpose, and are not an output of the **EyeTrackMat** toolbox functions.

2.4 Toolbox Functionality

The Matlab toolbox **EyeTrackMat** provide functions to extract eye tracking data from mobile eye tracker using the approach described in Section 2.3. The *GetVideoFrame* function is firstly used to break the video into individual video frames and then the *ExtractData* or *ExtractData2* functions are used to extract data from the video frames. Commonly used eye tracking visualization tools are available in the **EyeTrackMat** toolbox to visualize the data extracted from the video.

The AVI video that is recorded by the ASL mobile eye tracker is broken into consecutive video frames with the *GetVideoFrame* function (Zheng, 2014). Other video formats, such as MP4, WMV, or MOV can also be directly used in this function depending on the platforms one is using (The MathWorks Inc., 1994-2017). The user can specify the starting and ending time, the interval in seconds between each frames, and the output directory for the video frames. The *GetVideoFrame* function returns the video frames in the specified output directory.

ExtractData and *ExtractData2* are written to conduct the data extraction. *ExtractData* is used when the coordinates of the scene are unknown and *ExtractData2* is used when the coordinates of the scene are given. Users need to specify the directory of the video frames, the file that contains the clear image or photograph of the poster, the output CSV file, the location where to store the matched images, and the starting and ending iterations, i.e. from which video frame to which video frame the extraction should be conducted. For *ExtractData2*, users have to input the coordinates of the crosshair in terms of the scene. The output of the two functions include the matched images compared with the clear version of the poster that has the extracted coordinate labeled for quality check purposes, and a CSV file of the coordinates of the crosshair in terms of the poster. According to our empirical experiments, the default settings of the two functions are able to extract meaningful data sets from the participants looking at the scientific posters. However, the matching parameters (see Section 2.3.2) and the color parameters (see Section 2.3.3) can be modified based on the code provided by the toolbox.

The visualization functions *heatmap_generator* and *scanpath* are taken from Krassanakis et al. (2014) to generate attention maps and scanpath visualizations. We added the poster as the background image of the scanpath visualization and made the sequences of the scanpath visible. The *Scatterplot* function is used for the scatterplot creation.

Figure 2.9 shows the flow chart of the data processing in the **EyeTrackMat** toolbox and the structures of the functionality: The raw video data need to be broken into a folder of video frames. Data extraction functions can then be applied. With a csv file generated, data visualization tools are adopted for the data validation and exploration.

2.5 Application

The functionality of **EyeTrackMat** toolbox is presented in the following application.

2.5.1 Subjects

Data were collected on three participants, one computer science PhD student specialized in image processing (P1), one PhD student in Electronic Engineering (P2), and one computer science undergraduate student (P3). All three participants had normal or corrected-to-normal vision.

2.5.2 Procedures

A poster on medical image processing was set up on the wall in the eye tracking

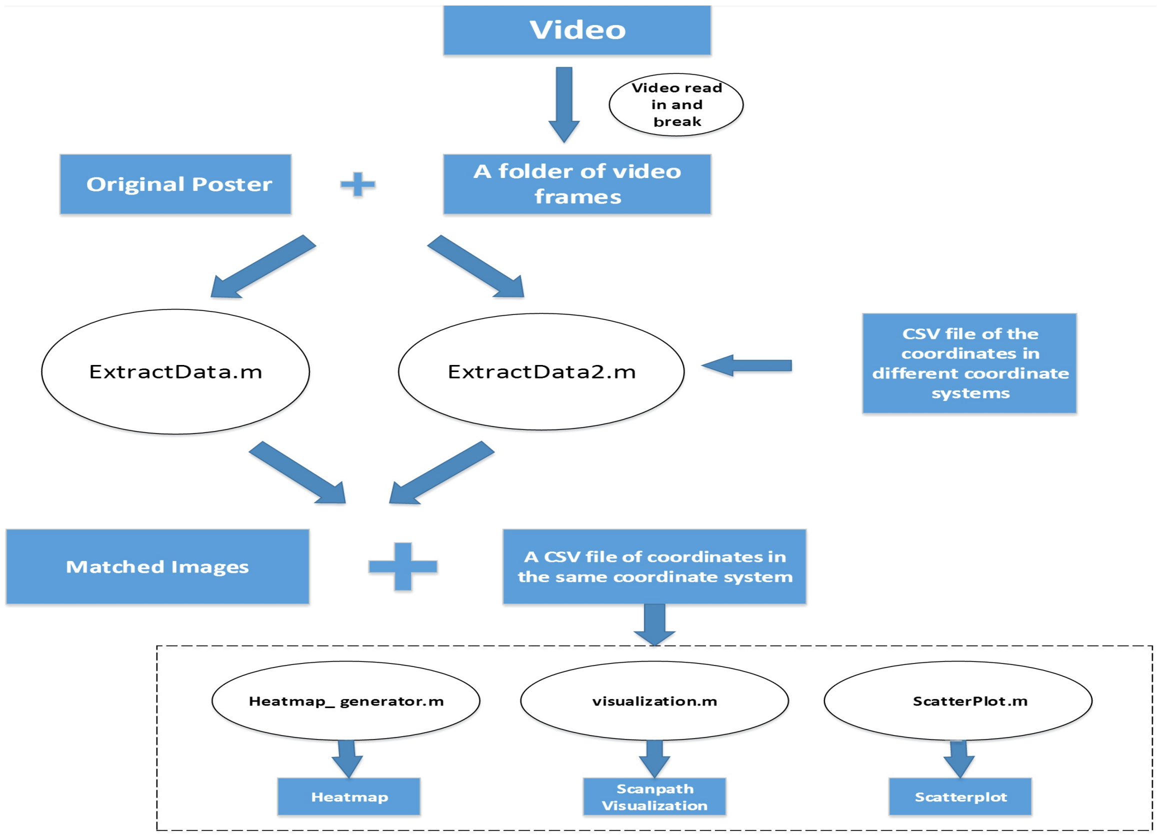

Fig. 2.9: Flow chart of the **EyeTrackMat** processing procedures

lab. Figure 2.10 shows a clear image of the poster. The dominating color of the poster is very close to red, the color of the crosshair that we are using; therefore, more noises from this poster are expected. We used a 30 Hz mobile eye tracker manufactured by ASL. The participant stood one to two meters away from the poster and was allowed to move around. The participant was firstly looking at the poster freely for 90 seconds. Then one of the authors read one question and let the participant look back at the poster to find the answer. Three questions were asked. One video was recorded for each of the participants, so we recorded three videos in total. Each video is about 3 minutes long overall. Therefore, we have around $3 * 60 * 30 = 5400$ video frames for each participant.

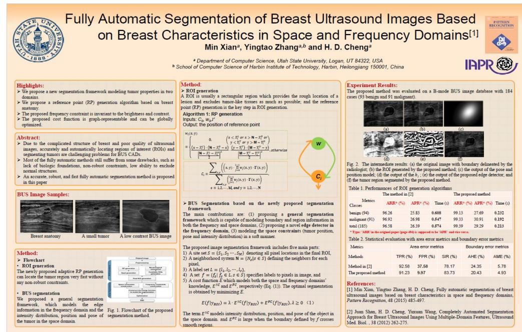

Fig. 2.10: Poster used in this application

2.5.3 Data Quality

A total of 200 video frames were randomly selected from $2700 * 3$ video frames of the 90 seconds of free viewing time for each participant. Based on a stratified sampling technique, 66 to 68 samples were taken from each participant. Among the 200 video frames, there are 29 missing coordinates (14.5%) according to our data extraction results based on the object detection approach. Since we are able to obtain a CSV file of the original coordinates of the crosshairs in terms of the video frame, the coordinates in terms of the poster can also be calculated based on the scales and angles from the geometric transformation for a comparison of the results. There are 30 missing coordinates (15%) according to the data extraction results. The missing data is reported slightly differently for the two data extraction approaches. In this case, the discrepancy is because when the crosshair is on the edge of the poster, the object detection approach is still able to identify the remaining crosshair while the transformed coordinate is outside the poster boundary (see Figure 2.11).

Missing Data

The reasons for the missing data have been investigated. 16 out of the 29 frames don't have the crosshair on the poster (Figure 2.12 (left)), or the center of the crosshair falls outside of the poster (Figure 2.12 (right)). In these situations, we assume that

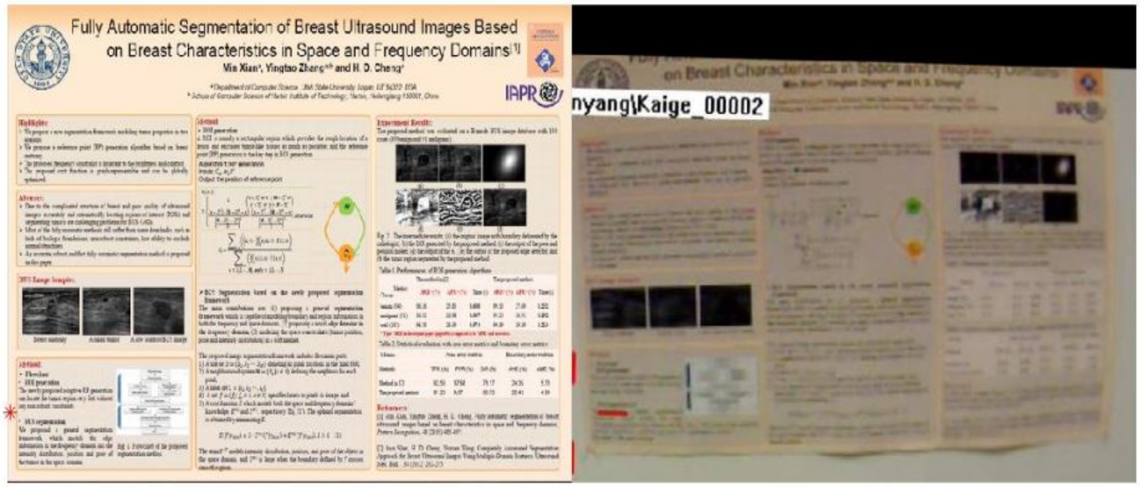

Fig. 2.11: The coordinates transformation identifies that for this video frame, the focus point is not on the poster, but the object detection identifies that the focus point is on the poster.

the participant is blinking or not looking at the poster. The rest of the missing data (13 out of 29) occurred because the image matching algorithm is not robust against the distortion of the image due to the visual curvature. Figure 2.13 (left) shows the original video frame and Figure 2.13 (right) shows the matching result. As the matching is not very accurate in this case, we fail to extract the coordinates of the crosshair.

Missing data could be imputed by taking the average of the coordinates of the previous and the next video frame, or by duplicating the coordinates obtained from the previous or the next video frame. Other missing data imputation approaches could also be explored. Since we have a large enough sample size, the missing data are deleted for further data analysis in the following application. There are still about 5400*0.85 = 4590 video frames left after removing the missing data.

Extraction Error

The extraction error is defined as the Euclidean distance between the calculated

Fig. 2.12: Missing crosshair: the poster on the left does not have a crosshair and the poster on the right has a crosshair but with only one leg on the poster.

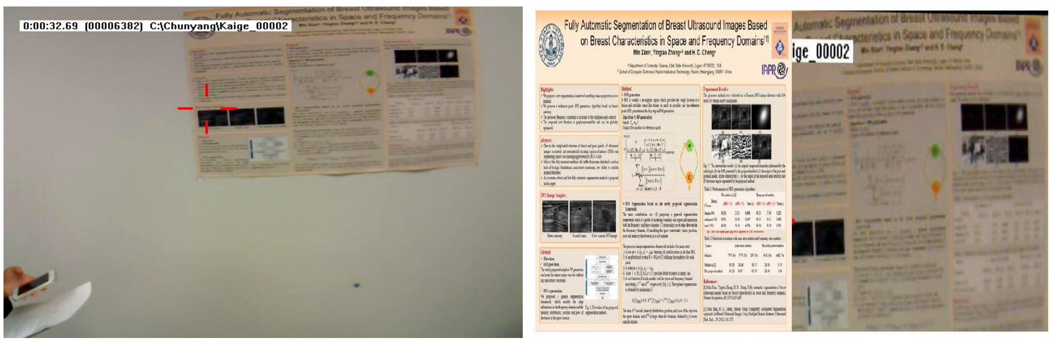

Fig. 2.13: Missing coordinate: there is a crosshair on the poster (left), however, due to the inaccurate matching in this case, we fail to detect the coordinates of the crosshair (right).

coordinates and the true location of the crosshair. The true locations of the crosshairs for 200 video frames were manually labeled by one of the authors of this article. The dimension of every video frame and matched image is $640 * 480$ pixels (width*height).

For the object detection approach, the average of the extraction error is 20.8 pixels, with a median of 15.9 pixels and a standard deviation of 21.2 pixels. Figure 2.14 shows the accuracy of the algorithms: The red star shows the coordinates calculated by the software, and the blue dot shows the true location labelled by one of the authors. The smaller orange circle is drawn with the purple dot as the center and the median extraction error (15.9 pixels) as a radius. The larger purple circle is drawn with the purple dot as the center and the mean extraction error (20.8 pixels) as a radius. The two circles show visually about how much the extracted coordinates are

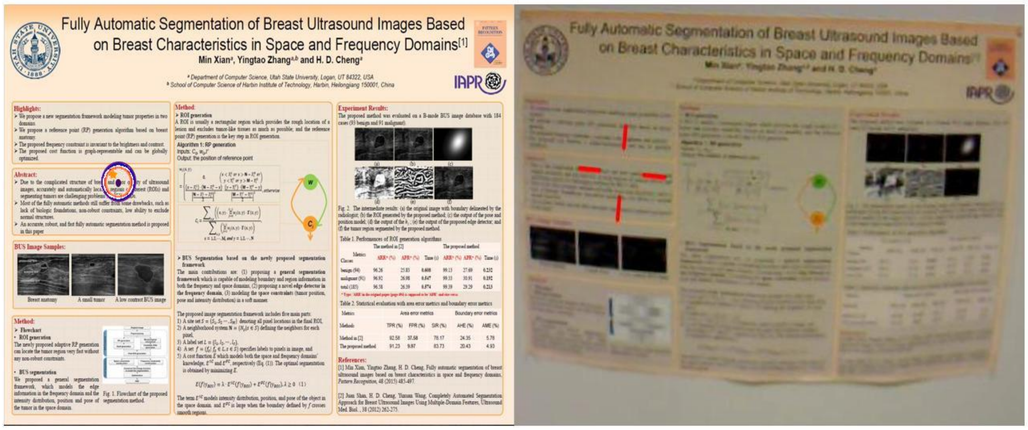

Fig. 2.14: Poster (left) with labeled true location (purple dot), the mapped location (orange star), the median accuracy (smaller orange circle), and the mean accuracy (larger purple circle), and the matched video frame (right).

away from the true locations on average or in terms of the median extraction error. The distribution of the errors after removing one outlier are shown in a histogram in Figure 2.15 (left). There was one outlier whose extraction error is 614.7 pixels, due to the failure of detecting the crosshair.

For calculations based on the coordinates transformation, the average of the extraction error is 20.1 pixels, with a median of 17.7 pixels and a standard deviation of 21.8 pixels. The histogram of the errors after removing the outlier is shown in Figure 2.15 (right). The distributions of the extraction errors based on the two estimation approaches was similar.

More experiments on posters with different colors and contents were also conducted and the data quality was similar.

2.5.4 Results

Figure 3.6 shows the scatterplots of where the focus points are and Figure 3.10 shows the heat maps of the viewing patterns for the 90 seconds of free views from the

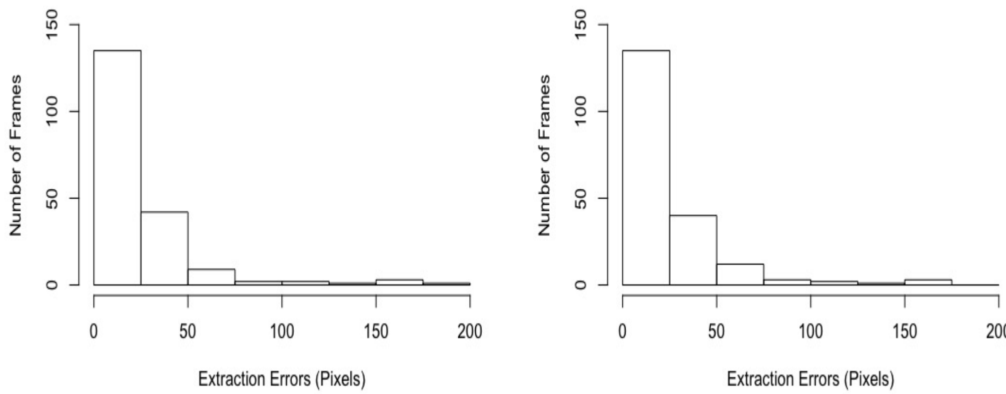

Fig. 2.15: The distribution of the extraction error: the errors from the calculation based on the object detection approach (left) and the errors from the calculation directly based on the geometric transformation (right)

three participants. It seems that the two PhD students (P1 and P2) were able to focus on more parts of the poster within the limited amount of time than the undergraduate student (P3). P1 is even trying to read some of the more detailed formulas, as we can see there are some light hot spots on the formulas in Figure 3.10 (left).

For the first question, the PhD student specialized in image processing (P1) didn't even look back at the poster and immediately gave the answer. It seems that the PhD student in Electronic Engineering (P2) looked all over the poster to find the answer, while the undergraduate student (P3) just looked at the key words in the title and gave the answer.

We used scanpath visualizations to visualize how the participants search for answers of the three questions. Here we only show the results for Question two. Results for Questions 1 and 3 are not presented in this chapter. Question 2 asked to judge whether the following statement is true or not: "There are three main parts for the proposed image segmentation framework." An example of a scanpath visualization presenting how the participants are searching answers for this question is shown in Figure 2.18. For this question, both PhD students (P1 and P2) searched

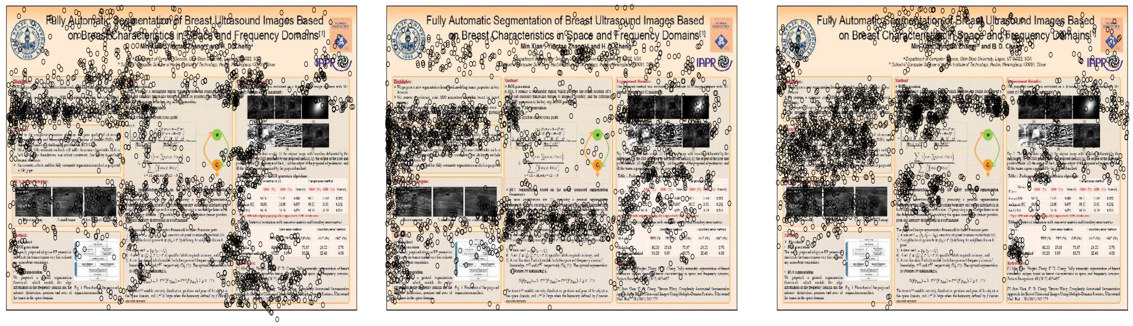

Fig. 2.16: Scatterplot for 90 seconds of free viewing: P1 (left), P2 (middle), and P3 (right)

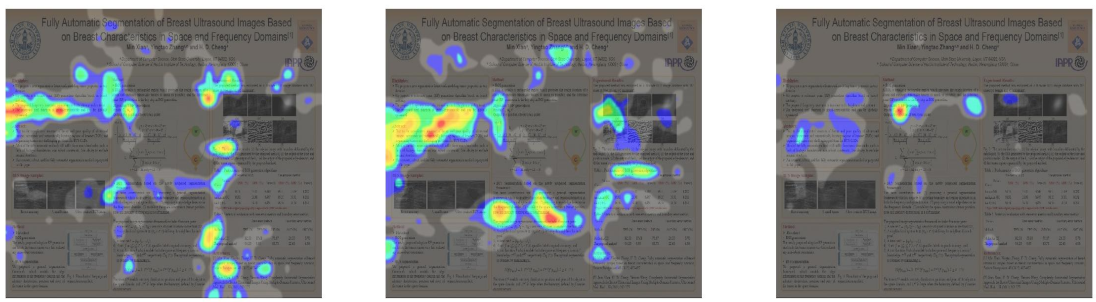

Fig. 2.17: Heatmap for 90 seconds of free viewing: P1 (left), P2 (middle) and, P3 (right)

more parts on the poster in order to find the answer, while the undergraduate student seemed to look at only a smaller part of the poster. According to the fixation detection algorithm from Krassanakis et al. (2014), there are no meaningful fixation points detected for the undergraduate participant (P3). Therefore, only the viewing patterns from P1 and P2 are presented in Figure 2.18. The answer to the question is located at the bottom of the middle column of the poster that is highlighted in a black box shaded in a light gray color. Only P2 eventually found the correct answer. P1 only briefly looked at the area and P3 didn't look over the area at all. Therefore, both of them gave the wrong answer.

2.6 Conclusion and Discussion

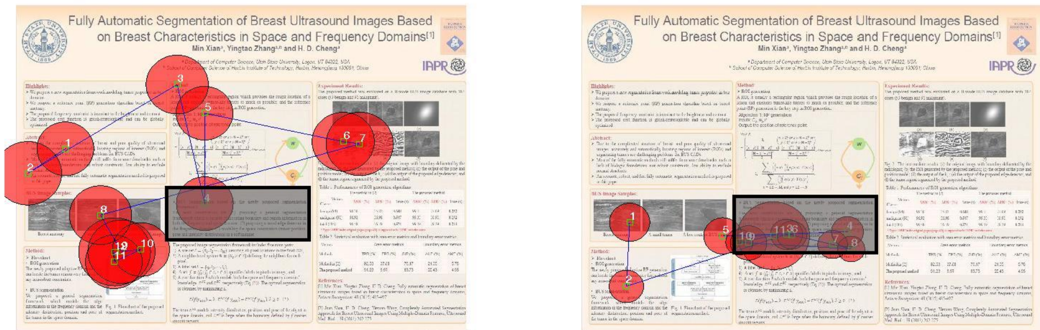

Fig. 2.18: Scanpath for Question 2: P1 (left), P2 (right)

In this chapter, we proposed the application of feature detection and image registration approaches to automatically extract data from mobile eye trackers. In addition, we proposed a heuristic object detection approach to find the focus point of people looking at posters when only the video data from a mobile eye tracker is provided. Our proposed methods enable researchers to automatically obtain the coordinates in the same coordinate system indicating where people are looking at a poster, so that they don't have to manually look at every single video frame to decide the focus points on the poster. A case study was provided to illustrate the functionality of the toolbox and data quality of the proposed approach.

More experiments have been conducted on different types of posters. The posters with the height larger than the width usually generated better results, due to a smaller viewing distortion from the video. Scientific posters mostly have strong features and the default set of our function can conduct the image matching task. However, when our extract data functions were applied to some relatively simple designed posters or images, the threshold for detecting the features needed to be lowered in order to return to more blobs of features, thus generating meaningful matches. If necessary, the threshold for matching can be lowered as well. For the posters containing too many red components that influence the data extraction results, we recommend to change the default color of the crosshair to green or some other color that shows up less in the poster.

Future work will be to further improve the data quality by making the image matching more robust against distortions. Also, other applications can be explored, such as verifying the saliency algorithm with how people are looking at images, and application in marketing research by analyzing video data of how potential customers look at commercial posters or power point slides. Furthermore, other objects can be thought of as a poster, therefore, this automatic data extraction approach could be applied in extracting data of how people are looking at other objects, such as a person's face or body, a machine, or a tool, etc. Symanzik et al. (2017b) extended the application on looking at power point slides to judge human postures by using the **EyeTrackMat** toolbox. More visualization tools can be found in our R package **EyeTrackR** that is soon to be released on the Comprehensive R Archive Network (CRAN) (Li and Symanzik, 2017; Symanzik et al., 2017b).

CHAPTER 3
THE LINKED MICROPOSTER PLOTS FAMILY AS NEW MEANS FOR THE VISUALIZATION OF EYE TRACKING DATA

3.1 Introduction

Eye tracking is the process of measuring where people are looking at with an eye tracker device. Eye trackers were first built in the late 1800s and have been developed rapidly during the past century. Holmqvist et al. (2011) provided a comprehensive review on the history of eye trackers as well as on the principles of how they work. Eye trackers nowadays are mostly 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 meaningful scene videos indicating where people are looking at are generated.

Eye tracking techniques have been applied in a variety of research fields, such as education, usability research, sports, psychology, and marketing. There exist several literature reviews focusing on the application of eye tracking, e.g., Rayner (1998) provided a comprehensive review on eye tracking for the past twenty years in reading and information processing, and Jacob and Karn (2003) provided a comprehensive review of eye tracking in human-computer interaction and usability research. Software for eye tracking data has been developed with R (R Core Team, 2016) packages (von der Malsburg, 2015; Dink and Ferguson, 2015), Matlab toolboxes and functions (Krassanakis et al., 2014; Berger et al., 2012), Python packages (Sogo, 2013; 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.

The eye tracking technology has become more and more affordable and accessible nowadays (Gould and Zolna, 2010). There exists some research on eye tracking 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 for posters has specifically discussed eye tracking visualization or adopted any new visualization techniques. In this chapter, we propose three different types of linked microposter plots to visualize eye tracking data of how people are looking at posters, recorded with a mobile eye tracker. These types are basic linked microposter plots, linked timeline microposter plots, and linked scanpath microposter plots. Linked timeline microposter plots and linked scanpath microposter plots are extensions of basic linked micromap plots, and are used to visualize the scanpath of the eye tracking data. Therefore, time series information of eye tracking data is included in these visualizations. Basic linked microposter plots were first introduced in Li and Symanzik (2016). Software implementations of these three types of linked microposter plots can be found in the **EyeTrackR** R package (Li and Symanzik, 2017).

The remainder of this chapter is structured as follows: We will discuss eye tracking technology and eye tracking data collection and processing in Sections 4.2 and 3.3, respectively. Eye tracking data visualization, the development of the linked microposter plot, and the construction of the linked microposter plot will be discussed in Sections 3.4, 3.5, and 3.6, respectively. The resulting plots will be presented in Section 3.7. The linked timeline microposter plot and the linked scanpath microposter plot will be introduced and presented in Section 3.8. We will finish with our conclusion and outline our future work in Section 4.4.

3.2 Eye Tracking Devices

There are two main types of eye tracking devices: static eye trackers and mobile eye trackers. 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. Figure 3.1 shows all components of the mobile eye tracker equipment manufactured by Applied Science Laboratory (ASL). There is 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

Fig. 3.1: ASL mobile eye tracker equipment showing the DTU in the front, the eye tracking glasses to the right, and the laptop in the back.

systems, i.e., the coordinate system is changing for every single frame due to the change of scences. In these situations, the videos first must be processed so that all the coordinates are transformed to the same coordinate system in terms of a poster, thus the resulting data can be used for further analysis.

3.3 Data Collection and Processing

The two co-authors of this chapter looked at a series of statistical and other scientific posters, using a 30 Hz mobile eye tracker from ASL that records 30 images per second. For each poster, the areas of interest (AOIs) were defined in advance, such as the title, logos, multiple text areas, images, and tables. For this chapter, two of the posters, the resulting videos, and data files have been used. The data for Poster 1 is based on a controlled experiment. The data for Poster 2 is based on a "free-viewing" experiment where no instructions were given to the participant. The AOIs of the posters can be automatically defined with the **EyeTrackR** R package. The rectangles that represent the AOIs are drawn by mouse clicking two vertices and they are named by the analyst. Figure 3.2 shows the twelve defined AOIs of Poster 1 and Figure 3.3 shows the nine defined AOIs of Poster 2. The red bounding boxes outline the defined AOIs. An additional AOI, called "Blank", contains all the empty space between these main AOIs. Poster 1 is used to test the data processing results and the validity of the linked microposter plot. The participant is timed to look at eight of the AOIs for about six seconds and at four of the AOIs for about two seconds. Poster 2 is used for a “free-viewing” experiment where it was left to the participant to look over the poster with the general goal to understand as much of this poster as possible. The data and video for both posters were processed in the same way. Overall, the participants looked at Poster 1 for about 56 seconds and at Poster 2 for about 80 seconds, resulting in a total of 1680 and 2400 video frames, respectively.

For further analysis, the video is broken into a sequence of consecutive video frames, also via the **EyeTrackR** R package. Figure 3.4 shows an example of one of the video frames extracted from the video recorded on Poster 1. The red crosshair indicates where the participant is looking at. Since the scene is changing all the time (because of movements and head movements), the coordinates of the crosshair correspond to different coordinate systems and are not comparable, therefore they need to be unified to the same coordinate systems. Image registration and object detection approaches are used to automatically extract the location of the focus point at the center of the crosshair. Li et al. (2017) provided the **EyeTrackMat** Matlab toolbox to conduct the video data processing task and Li and Symanzik (2017) made this functionality more widely available as part of the **EyeTrackR** R package.

Figure 3.5 shows the result of the automatically extracted focus point overlaid on the poster: the image on the right shows where the crosshair is in the original video frame and the image on the left shows the automatically extracted focus point that is overlaid with a black circle on Poster 1. Figure 3.6 shows all the automatically extracted focus points overlaid on Poster 1.

3.4 Eye Tracking Data Visualization

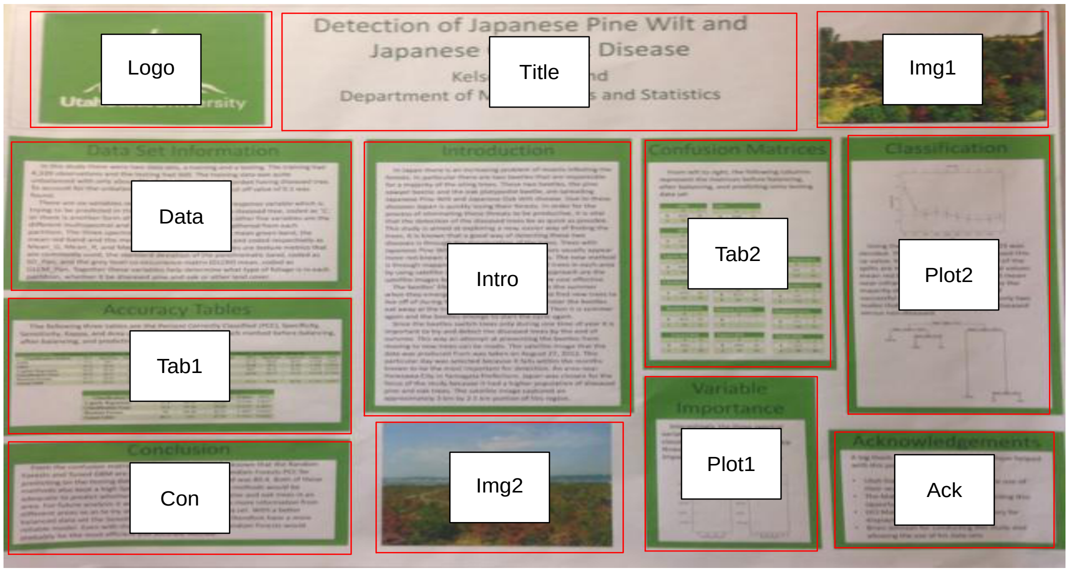

Fig. 3.2: Poster 1 with twelve AOIs (shown inside the red bounding boxes). The abbreviations Img, Tab, Intro, Con, and Ack refer to the images, tables, introduction, conclusion, and acknowledgement in the poster.

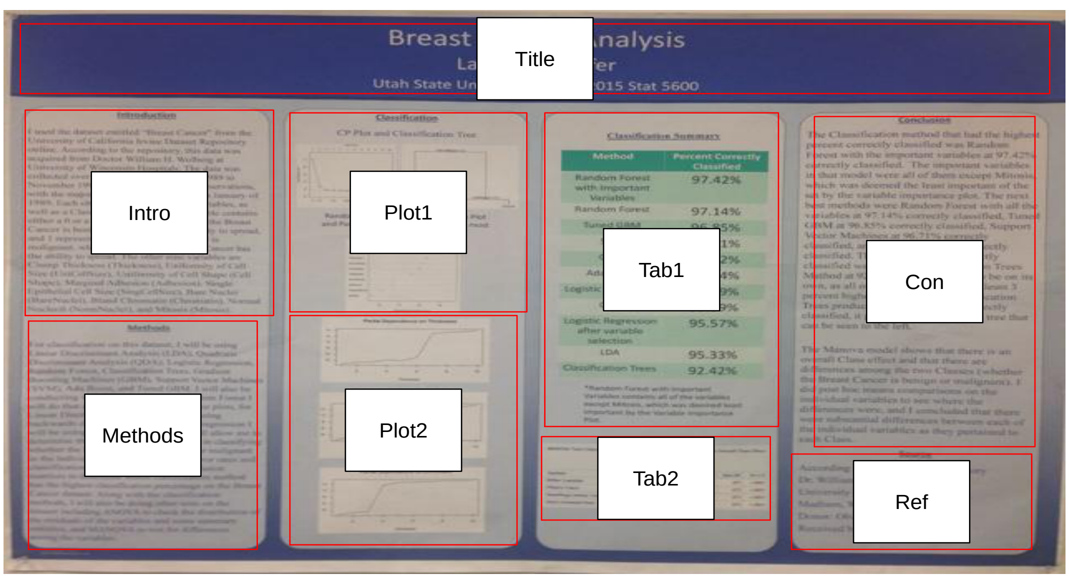

Fig. 3.3: Poster 2 with nine AOIs (shown inside the red bounding boxes). In addition to the abbreviations used in Figure 3.2, Ref is used to refer to the references in this poster.

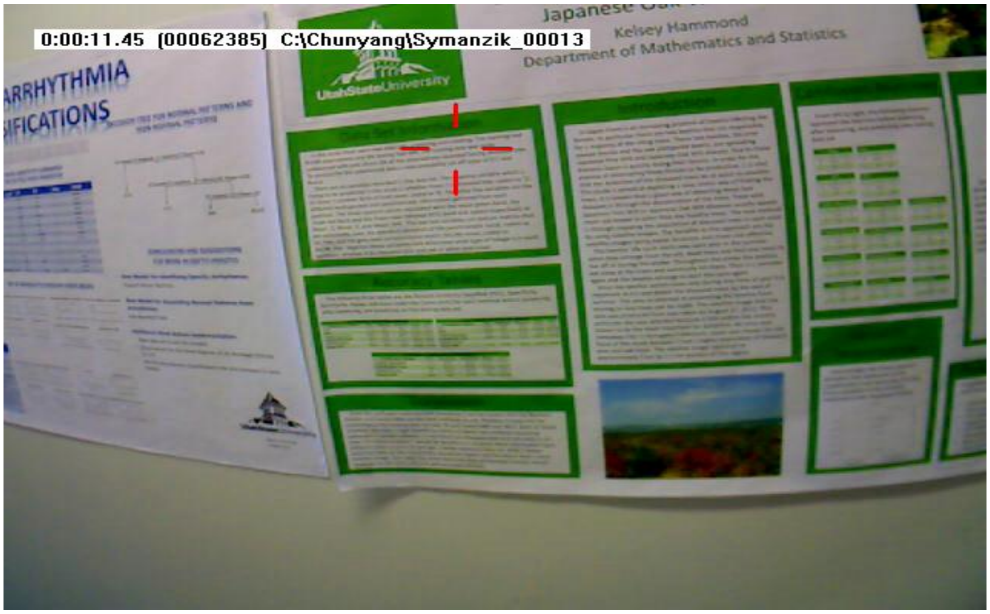

Fig. 3.4: One of the original video frames from the video of the viewing of Poster 1. The red crosshair shows the focus point of the participant in this video frame.

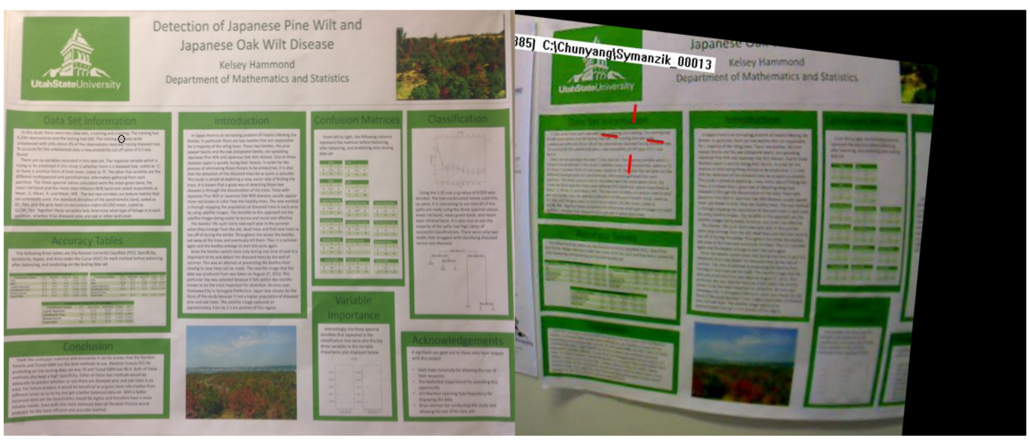

Fig. 3.5: Automatically extracted focus point overlaid on Poster 1 (left), based on the video frame of the viewing of Poster 1 shown in Figure 3.4 (right).

Graphical methods are among the most important tools to explore eye tracking data. Common statistical graphics, such as dot plots, bar charts, and box plots, are frequently used to visualize eye tracking data. Figures 3.7 and 3.8 show examples of using dot plots and bar charts to visualize how much time the participant has spent

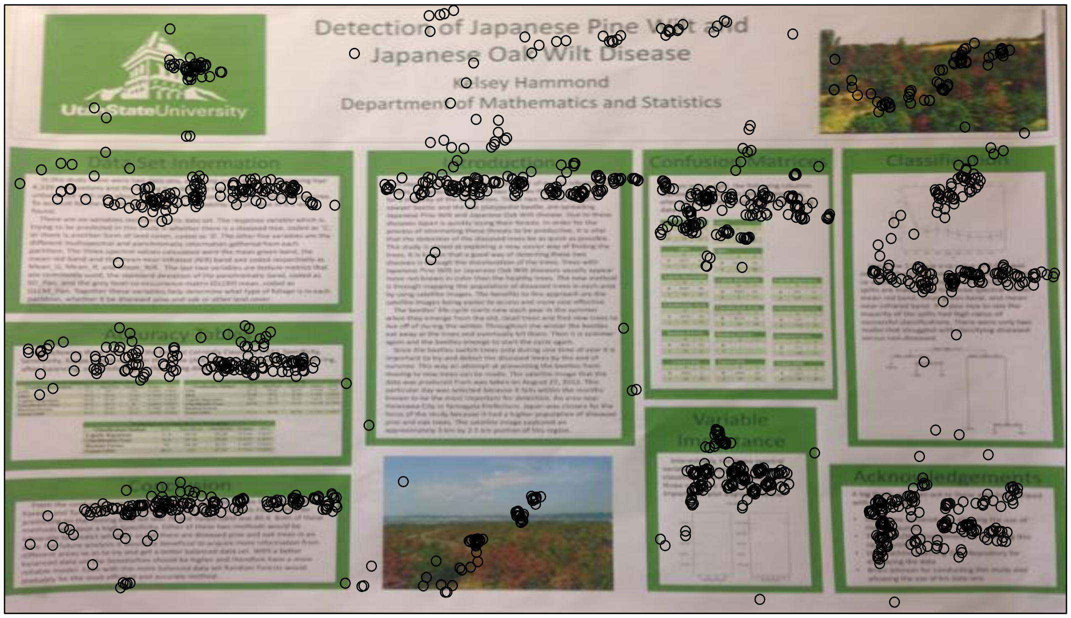

Fig. 3.6: All extracted focus points overlaid on Poster 1, based on the video frames of the viewing of Poster 1.

on each AOI of Poster 1. As a reminder, data from this experiment were obtained by looking at eight of the AOIs for about six seconds and at four of the AOIs for about two seconds. Both figures reveal that the actual looking times for these AOIs differ up to one second from the intended looking time. This may be because of frames where no crosshair could be detected or the image could not be matched properly with the poster. Figure 3.9 shows an example of using box plots to visualize the pupil radius in each AOI of Poster 1. The participant's pupil dilated looking at Image 1 and the variation of the pupil radiuses is high looking at Blank area. Pupil radiuses are measured in pixels on the eye camera and are only comparable within the same experiment with the same calibration. Though it hasn't been fully established what the changes of pupil radiuses really mean, pupil dilation has been shown to be an indication of changes in light, arousal, cognitive and emotional events, and the difficulty of the task at hand (Fong, 2012). These three graphs present the eye tracking data statistics, however, they are not overlapped with the poster. Therefore, it is more difficult to obtain further insights of the viewing patterns of the participant.

Attention maps are also frequently used for eye tracking data visualization (Holmqvist et al., 2011). Attention maps are usually based on heat maps or hot spot maps, using a Gaussian kernel function. Figure 3.10 shows the attention map for Poster 1. With the hot spots overlaid on the poster, it is quite obvious at which areas the participant is looking most frequently: the hot spots mostly appeared on the top part of each AOI, which can be explained by the viewing time limitations for each AOI. Because the participant was reading from top to bottom and left to right in each AOI, only the first few lines of text could be read in the allowed amount of time. However, only one variable at a time can be visualized in a single attention map. Also, converting numeric values into a few colors results in an immediate loss of information.

The term "scanpath" was first introduced by Noton and Stark (1971) to describe the chain of fixations and saccades. Fixation is the state when the eye remains stable for a period of time, and a saccade is the rapid movement of the eye from one fixation to another. In visual representations of scanpaths, circles are used to represent fixations and lines are used to represent saccades (Goldberg and Helfman, 2010b). The radius of the circle indicate the duration of the fixation. Figure 4.4 shows the scanpath map for Poster 1. 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.

The AOI timeline (Figure 4.3) shows both the start and end times, as well as the temporal sequence of changes between AOIs. The horizontal axis shows the AOIs and the vertical axis shows the time. We can see from the plot that the participant

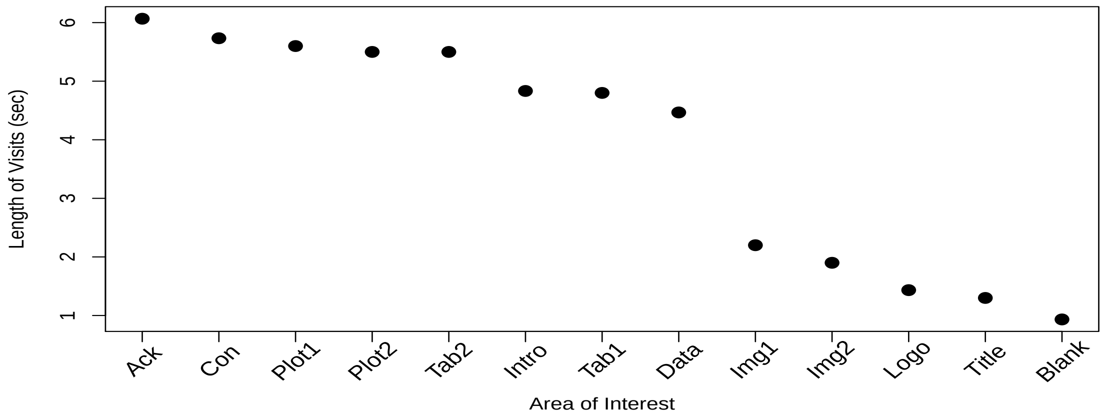

Fig. 3.7: Dot plot: visualizing the length of visits in each AOI for Poster 1.

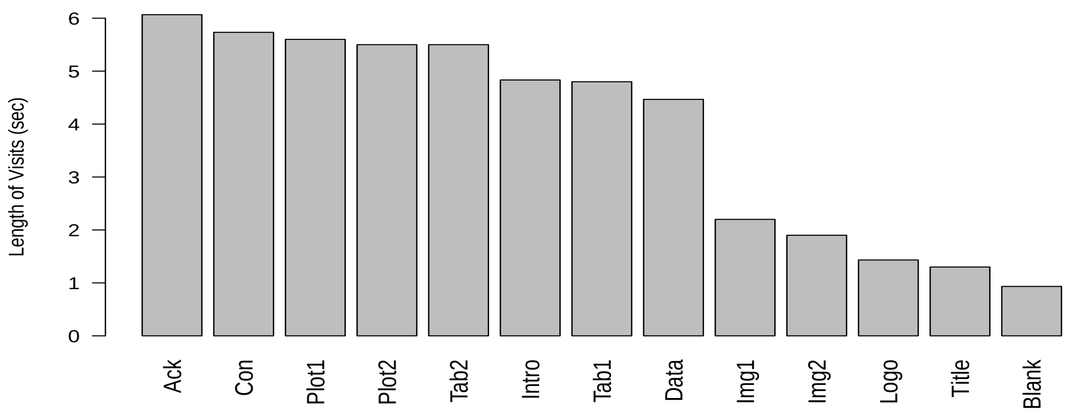

Fig. 3.8: Bar chart: visualizing the length of visits in each AOI for Poster 1.

revisited the data and the title very briefly. The conclusion has been visited twice as well, with a very short visit at the first time. However, an AOI timeline does not present the spatial locations of each AOI from the plot.

There does not exist any type of plot that is specifically designed to visualize how people are looking at posters. To overcome the shortcomings of the commonly used eye tracking data visualization techniques, we introduce a linked microposter

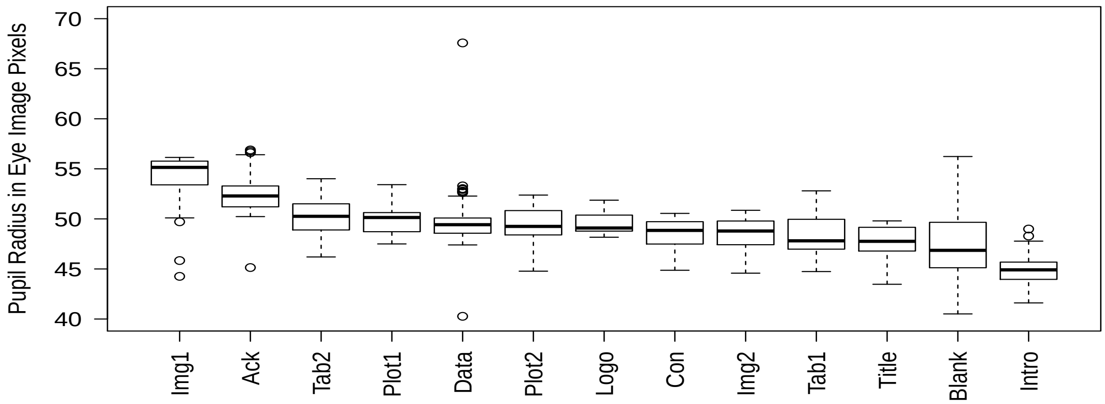

Fig. 3.9: Box plot: visualizing the pupil radius in each AOI for Poster 1.

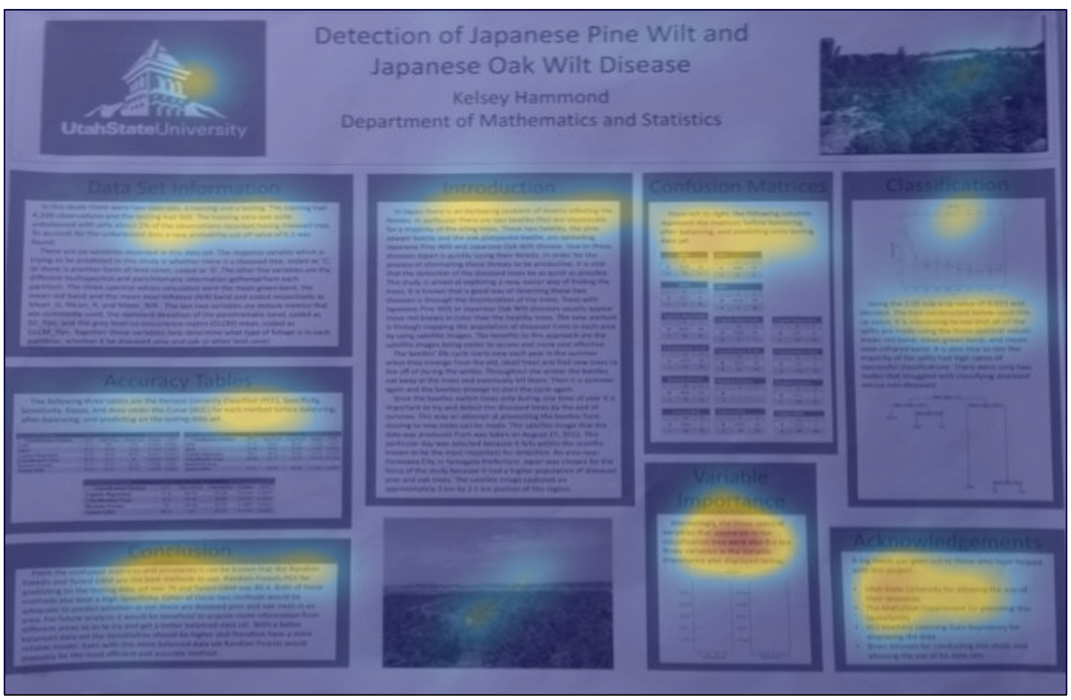

Fig. 3.10: Attention map: hot spots that attract the participant's attention for Poster 1.

plot to visualize eye tracking data on posters. Further, a linked timeline microposter plot and a scanpath microposter plot are extended to include scanpath time series

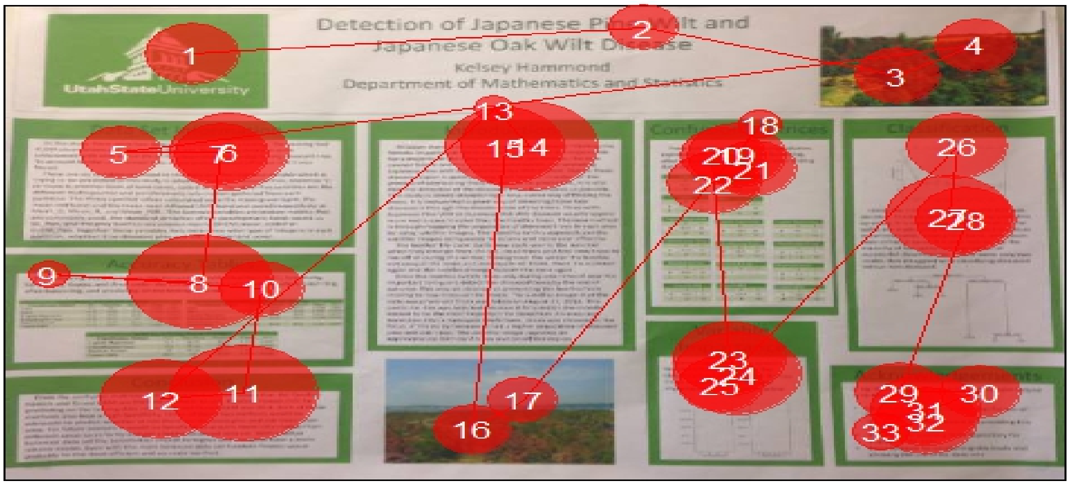

Fig. 3.11: Scanpath map: the viewing sequences of the participant looking at Poster 1.

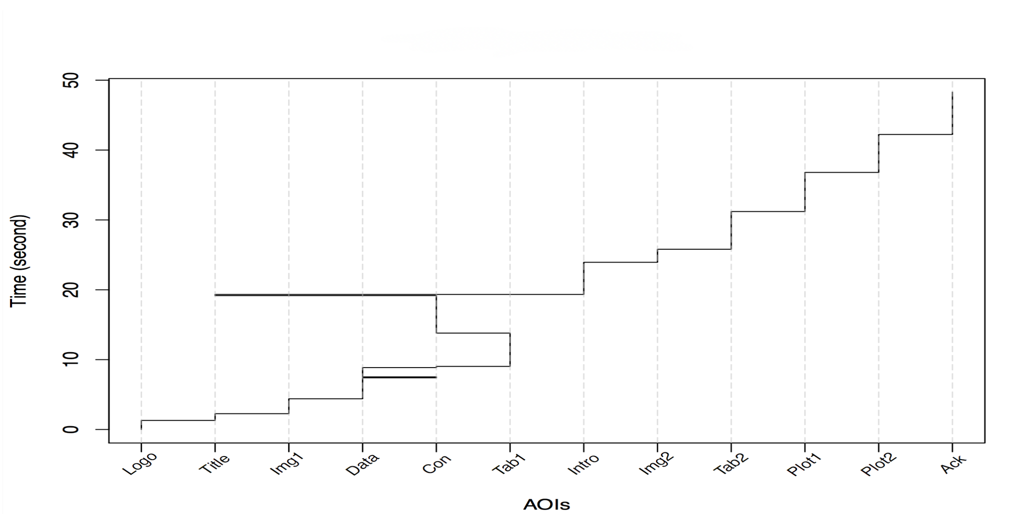

Fig. 3.12: AOI timelines: the temporal sequence of changes in viewing between AOIs for Poster 1.

information in the plot.

3.5 The Development of Linked Microposter Plots

The linked microposter plot is based on the idea of linked micromap plots, 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 has been widely used to display geospatially-indexed summary statistics. For some in-depth discussion of linked micromap plots, the reader is referred to Symanzik and Carr (2008), Carr and Pickle (2010), Symanzik et al. (2014), and Symanzik et al. (2017a). According to Carr and Pickle (2010), the linked micromap plots 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. 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, and pupil radiuses are some of the variables of interest. Other variables, such as eye movement speed, can also be visualized using linked microposter plots. 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 3.7, 3.8, and 3.9.

Compared to an attention map, linked microposter plots can be used to explore selected AOIs, instead of simply looking at the hot spots. Rather than focusing on a single detailed poster, there are multiple small posters (microposters) shown in linked microposter plots. The same colors are used to link the areas in the microposters, the names of the AOIs, and the statistical data columns. Providing small microposters on the sides, the linked microposter plot reveals the location patterns where one (or multiple) participants look at most on a poster.

3.6 Linked Microposter Plots Construction

Computer code to construct linked micromap plots has been available since their introduction in 1996, as summarized in Symanzik and Carr (2013). Major R code was provided in 2010 in support of Carr and Pickle (2010). With the advancement of the R computing environment, more advanced R code for the production of linked micromap plots has been developed. Two R packages, *micromap* (Payton et al., 2015b,a), and *micromapST* (Carr and Pearson, 2015; Pickle et al., 2015), have also been developed to make it easier for non-experts to produce linked micromap plots. However, the *micromapST* R package is focused on linked micromap plots for the United States. The *micromap* R package can be used for any geographic regions, but it requires new geographic shapefiles. Therefore, we developed our R code in the **EyeTrackR** R package for the construction of linked microposter plots based on the original R code for linked micromap plots provided by Carr and Pickle (2010).

The state border data for linked micropmap plots is replaced with the R border data generated by user-defined AOIs. The nation border for linked micromap plots is changed to the border of the whole poster. The poster image is used as the background image in each linked microposter plot. For better showing of the colors that link the various columns, the poster image is changed to a grayscale version.

The variables investigated in our experiments for this chapter are the length of visits, number of visits to each AOI, and the pupil radius in pixels. Length of visits is how long the participant has spent looking at each AOI. Number of visits is how many times the participant has looked at each AOI.

3.7 Linked Microposter Plots Interpretation

Figure 3.13 shows linked microposter plots for Poster 1. 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 shaded 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 several perceptual groups (three in Figure 3.13 and two in Figure 3.14.). Perceptual groups typically contain between two and five of the AOIs. Such a design helps the reader to 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).

As stated in Section 3.3, the video of Poster 1 is recorded for a controlled experiment where the participant is looking at eight AOIs for about six seconds and at four AOIs for about two seconds. Figure 3.13 verifies that the length of visits for eight AOIs are around six seconds and around two seconds for the remaining AOIs other than the blank AOI. Although all of the AOIs are supposed to be visited only once, several of the number of visits are bigger than one. This is because the participant's visual focus point is moving from one AOI to another and may pass through other AOIs, resulting in an increased number of visits for those AOIs. To take some of these situations into account, if the length of visits is less than a threshold of 1/10 second, it does not count as a visit. The threshold can be changed by the analyst. The right-most data column in Figure 3.13 shows the pupil radiuses for each AOI visualized via boxplots. The participant's pupil dilated while looking at the acknowledgement and at Image 1 of the poster, possibly because the participant saw someone he knows in the acknowledgement during the six seconds of reading and the image attracted him shortly for the two seconds of looking at it.

Linked microposter plots for Poster 2 are shown in Figure 3.14. This figure shows some spatial clusters of the eye tracking data that we would not be able to see in the simpler row-labelled plot designs shown in Figures 3.7, 3.8, and 3.9. We can see that the top microposter highlights the main content in the center regions of the poster,

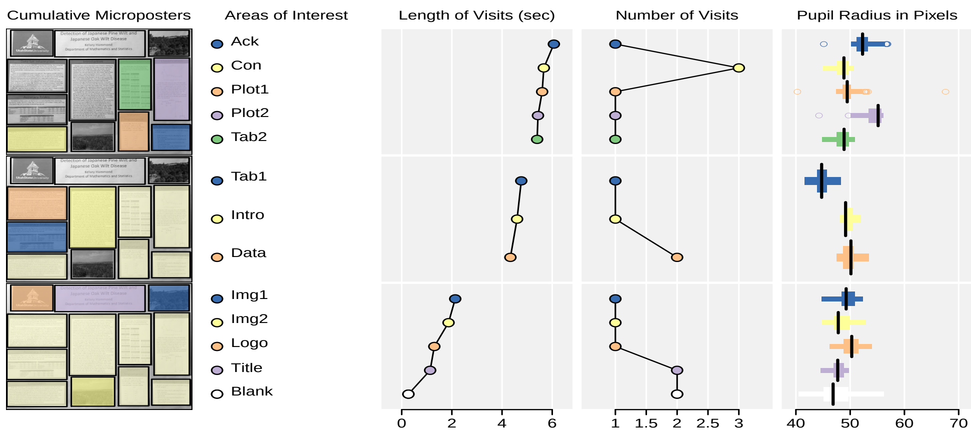

Fig. 3.13: Linked microposter plots of the eye tracking data for the AOIs for Poster 1.

while the bottom microposter highlights the parts in the corners of the poster. This indicates that the participant has spent most of the time on the main content of the poster, i.e., both tables, the introduction, the conclusion, and one of the plots of the poster.

We are able to visualize and compare multiple variables via several statistics columns in linked microposter plots, making it easier to identify the relationship between these variables. In Figure 3.14, the number of visits and the length of visits doesn't seem to have a strong association. In fact, a numerical assessment confirms that the correlation coefficient r is 0.032 between these two variables. Pupil radius seems to be negatively associated with the number of visits and the length of visits, with $r = -0.43$ and -0.64 respectively (r is calculated based on the median pupil radius in each AOI.).

3.8 Linked Timeline Microposter Plots and Linked Scanpath Microposter Plots

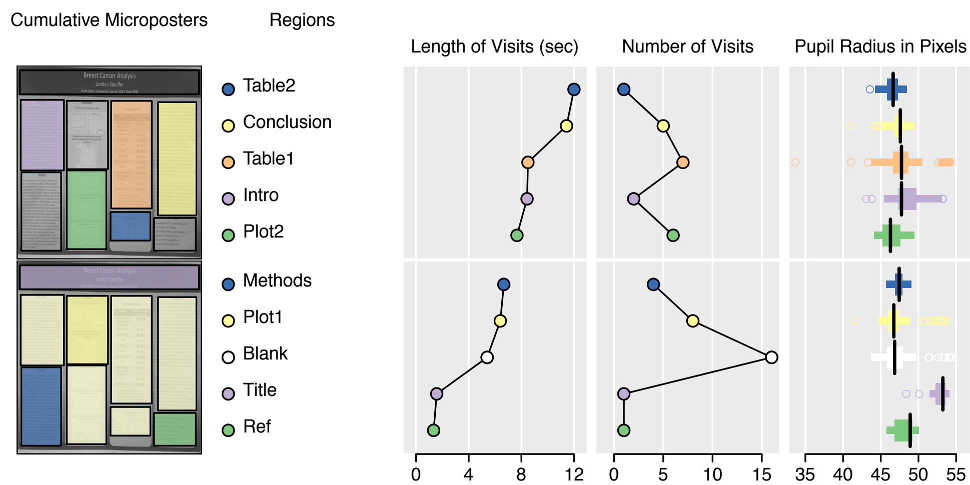

Fig. 3.14: Linked microposter plots of the eye tracking data for the AOIs for Poster 2.

Linked timeline microposter plots and linked scanpath microposter plots are extensions of a basic linked microposter plots. Linked microposter plots have many advantages over the commonly used eye tracking visualization techniques, however, the viewing sequences can not be shown in basic linked microposter plots. To overcome this disadvantage, we developed linked timeline microposter plots and linked scanpath microposter plots to visualize the timeline and the scanpath of eye movement. Linked timeline microposter plots are inspired by the idea of AOI timelines and linked scanpath microposter plots are inspired by the scanpath map.

3.8.1 Linked Timeline Microposter Plots

Compared with an AOI timeline, linked timeline microposter plots also show the spatial locations of the AOIs. Thus, the spatial clusters of the participant's viewing patterns can be identified. When there are many revisits of the AOIs, an AOI timeline is becoming complicated and confusing. Since linked timeline microposter plots only focus on several AOIs at a time in a microposter, an increased number of visits does not affect the quality of the visualization. Further, with more quantitive variables presented in one single plot, readers are able to explore the relationship among several variables.

Figure 3.15 shows an example of linked timeline microposter plots sorted by the viewing sequence. The numbers labeled inside the AOIs indicate the viewing sequence. Each color represents an AOI. We can see from the plot that the participant is looking from the top left to the top right and then from top to bottom column by column. While it is quite difficult to extract the exact viewing sequence of the AOIs from Figure 4.3, the sequential top-down arrangement immediately reveals the viewing sequence of the AOIs in Figure 3.15 respectively. Ideally, each of the AOI should have a different color, making it easier to identify the revisits of the AOIs. For example, in Figure 3.15, the light red, dark green and dark blue showed up twice in the microposters. Therefore, it is apparent that the bottom left AOI (the conclusion section) colored in light red has been visited three times, although the visits number 5 and 7 are just very quick glances. The visits of the data section (colored in dark green) are split into visits 4 and 6, due to the participant's glance at the conclusion section. The participant's eye passed by the title (colored in dark blue) again while his focus point moves from the first column to the second column on the poster. If the number of AOIs is more than the number of colors, the colors are recycled, i.e., some AOIs are shaded with the same color.

Linked timeline microposter plots can also be sorted by any of the other variables shown in the statistical columns, such as the length of visits and pupil radius. Therefore, the relationship between the viewing sequence and any other variable can be explored. Figure 3.16 shows an example of sorting the plot by the median of the pupil radius in a descending order. The plot shows that the participant's pupil radius is larger at the beginning and the end of the recording. Since it is a simple test recording, the participant is possibly excited at the beginning and the end of the test, while

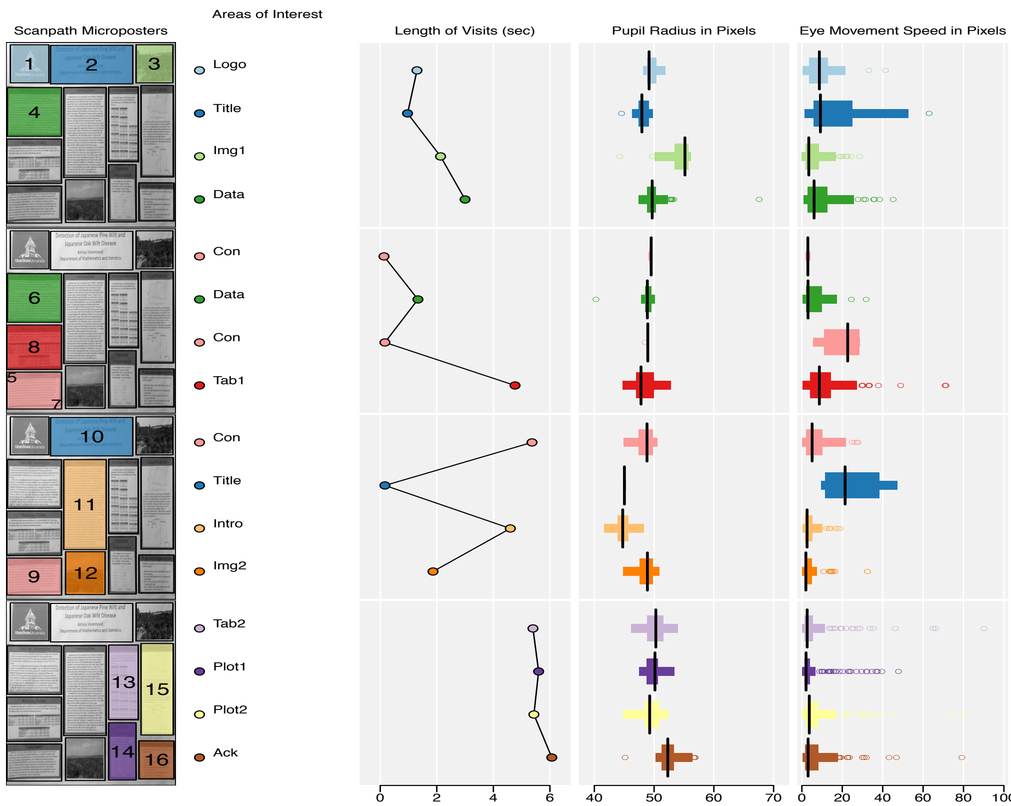

Fig. 3.15: Linked timeline microposter plots sorted by viewing sequence for Poster 1 feeling bored in the middle of the test. The pupil radius seem to be slightly negatively correlated with the eye movement speed, with $r = -0.39$. The eye movement speed is calculated by taking the distance between the consecutive locations.

3.8.2 Linked Scanpath Microposter Plots

Figure 3.17 shows an example of linked scanpath microposter plots sorted by the fixation sequences. 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, including the duration of the fixations (length of visits) and the pupil radius for each fixation.

Compared with a scanpath plot shown in Figure 4.4, the linked scanpath microposter plots in Figure 3.17 are focusing on a few fixations at a time, instead of

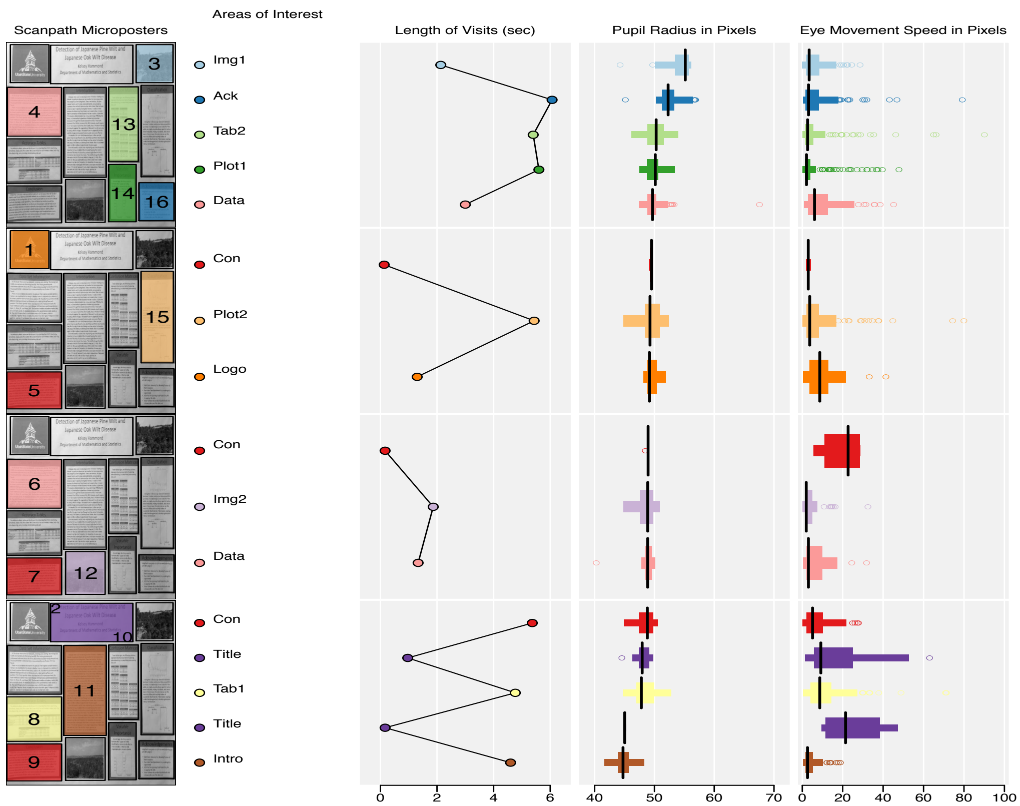

Fig. 3.16: Linked timeline microposter plots sorted by pupil radius for Poster 1 presenting the fixations all at once in a single graph. This feature makes even more complicated visual patterns easier to understand. The scanpath map uses the radius of a circle to indicate the duration of the fixation. When the duration of a fixation is very small or very large, corresponding circle in the scanpath map becomes unclear or overlapping. Using a dot plot in a separate panel to visualize the duration of the fixations, and linking to the fixations via color overcomes this issue. The capability of visualizing multiple variables in one single graph also shows the advantage of linked scanpath microposter plots over the commonly used scanpath map.

From Figure 3.17, we can see fixation points 8 and 10, labeled in blue and light red, last relatively longer than the other fixation points. The corresponding pupil radiuses also look relatively larger. However, there seem to be weak negative

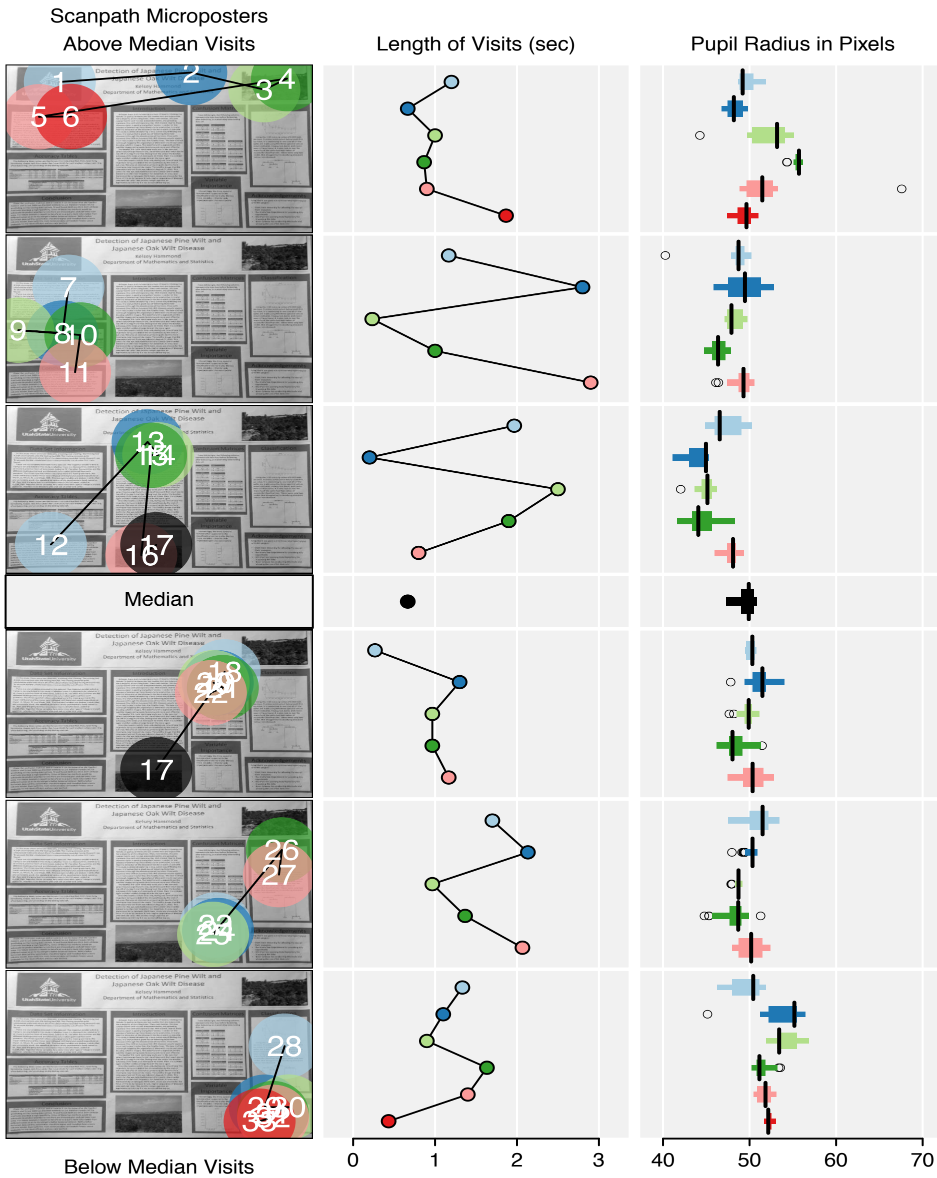

Fig. 3.17: Linked scanpath microposter plots sorted by fixation sequences for Poster 1

correlation between the median pupil radius and the duration of fixation, with $r = -0.15$.

3.9 Conclusion and Future Work

In this chapter, we proposed linked microposter plots family as new means to visualize how people are looking at a poster. We also demonstrated how the linked microposter plots family are able to more effectively visualize eye tracking data, compared to the commonly used eye tracking visualization tools. The linked microposter plots family have overcome the disadvantages of the commonly used eye tracking data visualization tools, making it possible to present multiple variables at one single plot accurately as well as their relationships. With the perceptual groupings, readers can quickly identify clusters of mapped AOIs with similar values of the sorting variable. Basic linked microposter plots are able to present the eye tracking statistics together with the spatial information. Linked timeline microposter plots and linked scanpath microposter plots add the time information in the plot, in addition to the eye tracking statistics. They make the tangled visual clusters and their statistics more clearly shown in one plot, compared with the AOI timelines and the scanpath map.

The *EyeTrackR* R package provides functions to extract and summarize the raw eye tracking data from a mobile eye tracker and to create linked microposter plots, linked timeline microposter plots, and linked scanpath microposter plots, in addition to data summarization and common eye tracking visualization tools (Li and Symanzik, 2017). The **EyeTrackR** R package will be released in late 2017 or early 2018.

Three types of linked microposter plots can be extended to visualize how people look at webpages, power point slides, photos, etc. Also, displaying data for multiple participants via single linked microposter plots and their extensions could be investigated in the future.

CHAPTER 4

EYETRACKR: AN R PACKAGE FOR EXTRACTING AND VISUALIZING DATA FROM MOBILE AND STATIC EYE TRACKERS

4.1 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 mostly 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 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 overlayed 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, 2010b). The the radius of the circle indicate the duration of the fixation. The numbers in the circles indicate the sequential order of the fixations. Scan path visualization 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 chapter, 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 plot 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 Miellet, 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 chapter, 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 linked scanpath microposter plots.

The remainder of this chapter is structured as follows: We will discuss the mobile eye tracking device and how the data are collected in Section 4.2. The **EyeTrackR** R package and its functionalities are discussed in Section 4.3. We will finish with our conclusion and future work in Section 4.4.

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 4.3.2). Figure 2.10 shows the twelve defined AOIs of the poster used in this chapter. This poster is a student's course project from 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.

4.3 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. Table 4.1 summarizes the functions and their main functionalities.

Table 4.1: Functions in the **EyeTrackR** R Package

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

Below is the intial set up code in order to run the examples in the following subsections:

```
library(EyeTrackR)
library(matlabr)
library(RColorBrewer)
data('locations_testing', 'timelinedata.all', 'AOIName', 'posterdat.all')
```

4.3.1 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.

```
options(matlab.path = "/Applications/MATLAB_R2016b.app/bin")
have_matlab()
```

The *GetVideoFrames* function breaks the AVI video into individual frames in all platforms. Other video formats, such as MP4, WMV, 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 4.2, 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 2.10. 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. The code below illustrates how to apply the *ExtractCoordinates* function.

```
ExtractCoordinates(framedir = system.file("extdata", "Frames",
                                          package = "EyeTrackR"),
            poster = system.file("extdata", "poster_colored.jpg",
                                 package = "EyeTrackR"),
            coordinate  = NA,
            outputcsv = "locations.csv",
            outputimg = "MatchImg",
            ibegin = 1,
            iend = 5)
```

4.3.2 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, 2016a) 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 AOIs 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.

The example function calls are below:

```
ResizeImg(poster = system.file("extdata", "poster_colored.jpg",
                                package = "EyeTrackR"),
         size_x = 500,
         size_y = 400,
         resized_poster = "poster_colored2.jpg"
        )


DrawAOIs(boxes = 3,
         poster = system.file("extdata", "poster_colored.jpg",
```

```
                                    package = "EyeTrackR"))
```

```
timelinedata.all <- GetAOITimelineData(posterVBorders = AOIName,
                                        locations = locations_testing
                                       )
```

```
posterdat.all <- GetPosterData(posterVBorders = AOIName,
                               locations = locations_testing
                                )
```

4.3.3 Common Eye Tracking Visualization Tools

Common statistical graphics introduced in Section 4.1, 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 4.1) shows where the focus points are and is used to directly indicate individual eye movements. There are 1647 focus points presented in Figure 4.1.

```
DrawEyeScatterplot(poster = system.file("extdata", "poster_colored.jpg",
               package = "EyeTrackR"),
               locations = locations_testing[, 1:2])
```

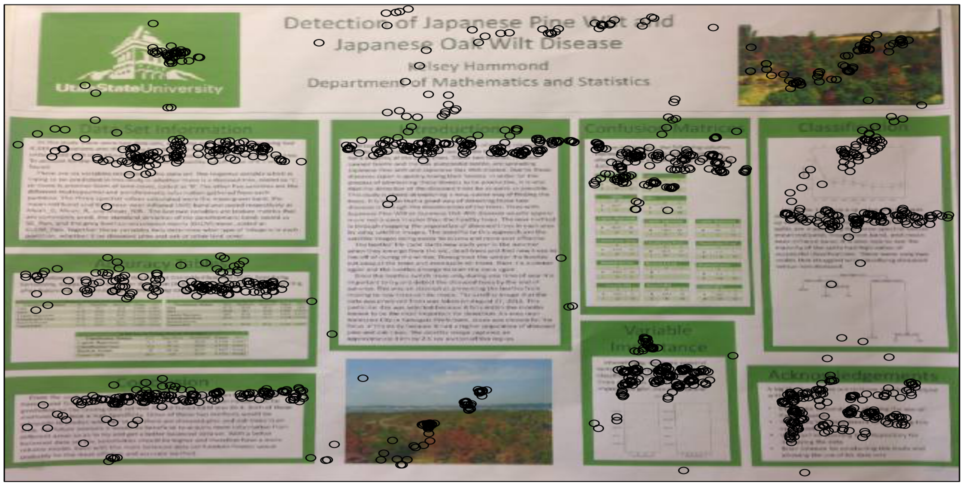

Fig. 4.1: Scatterplot, overlaid on the photo of the original poster

Figure 4.2 shows an example of an attention map created by the *DrawEye-Heatmap* function. With the hot spots overlayed 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.

```
DrawEyeHeatmap(poster = system.file("extdata", "poster_colored.jpg",
package = "EyeTrackR"),
locations = locations_testing,
bandwidth = c(60, 60))
```

Figure 4.3 shows the AOI timeline. We can see from the plot that the participant revisited the data and the title very briefly. The conclusion 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.

```
DrawEyeAOItimelines(timelinedata = timelinedata.all[[1]], label.cex = 0.8)
```

Figure 4.4 shows the scanpath visualization. Fixations and saccades are identified with the **saccades** R package (von der Malsburg, 2015). The numbers in the circles

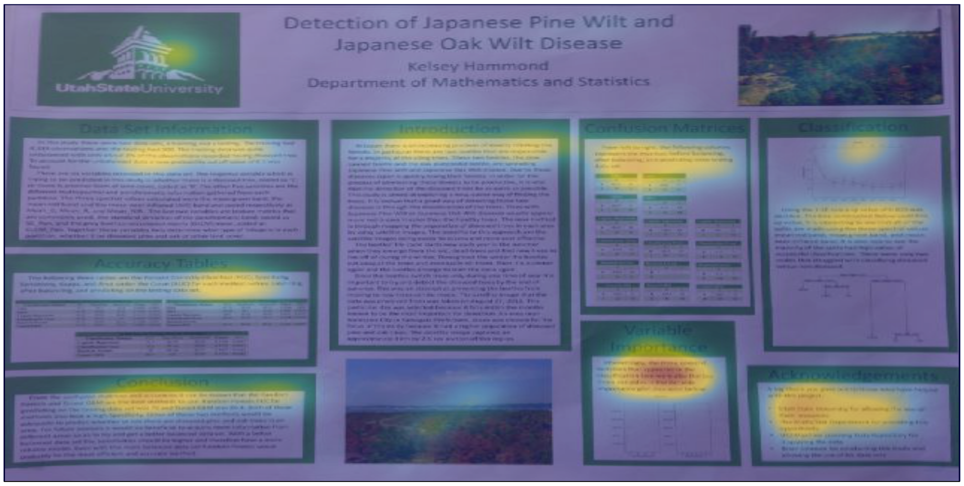

Fig. 4.2: Attention map, overlaid on the photo of the original poster

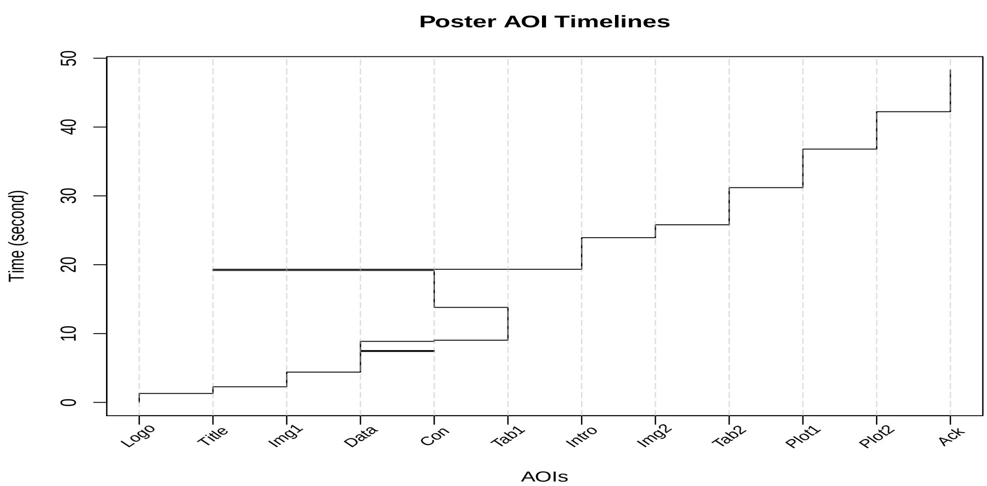

Fig. 4.3: AOI timelines

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.

```
DrawEyeScanpathMap(poster = system.file("extdata", "poster_colored.jpg",
                                        package = "EyeTrackR"),
                   locations = locations_testing
                   )
```

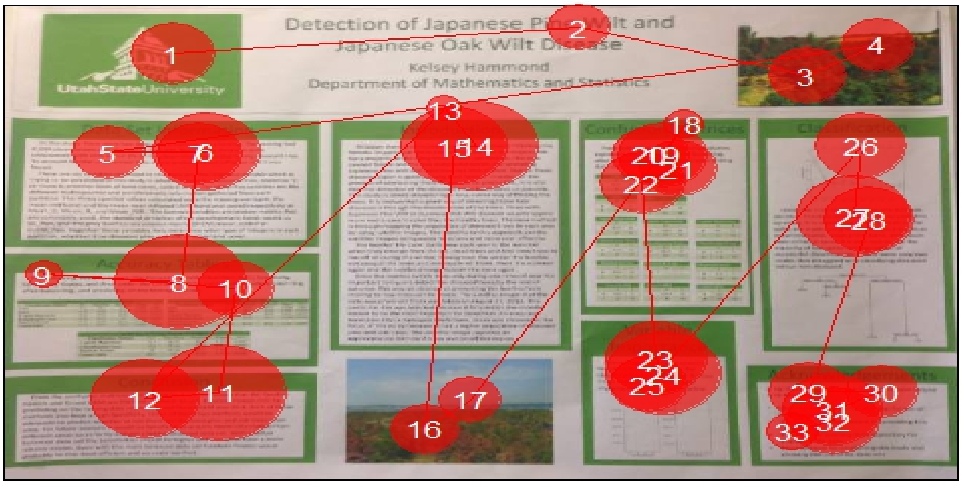

Fig. 4.4: Scanpath map, overlaid on the photo of the original poster

4.3.4 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 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 4.1, 4.2, 4.3, and 4.4. 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 the number of the column, the data to be shown in this column, and the sizes and content of the labelings, 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 code below creates the list for the *columns.att* argument prepared for the main function *DrawEyeLMPlot*. The structure is inspired by the **micromap** R package (Payton et al., 2015b).

```
columns.att <- list(
  list(col.num = 1,
       cumulate = TRUE),
  list(col.num = 2,
       header = "Areas of Interest",
       header.size = 0.7,
       point.size = 0.96,
       text.size = 0.7,
       text.font = 1),
  list(col.num = 3,
       panel.data = "visits",
       header = "Length of Visits (sec)",
       text.size = 0.7,
       point.size = 0.96,
       axis.ticks = NA,
       axis.labels = NA),
  list(col.num = 4,
```

```
        panel.data = "visits.num",
        header = "Number of Visits",
        text.size = 0.7,
        point.size = 0.96,
        axis.ticks = NA,
        axis.labels = NA),
   list(col.num = 5,
        panel.data = "pupildat",
        header = "Pupil Radius in Pixels",
        text.size = 0.7,
        axis.ticks = NA,
        axis.labels = NA))
```

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 4.3.2, 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.

```
DrawEyeLMPlot(poster.loc = system.file("extdata", "poster_colored.jpg",
                                       package = "EyeTrackR"),
        data = posterdat.all,
        posterVBorders = AOIName,
        panel.types = c("poster", "legend", "dot", "dot", "boxplot"),
        panel.width = c(2.9, 2.7, 3, 3, 3),
        columns.att = columns.att)
```

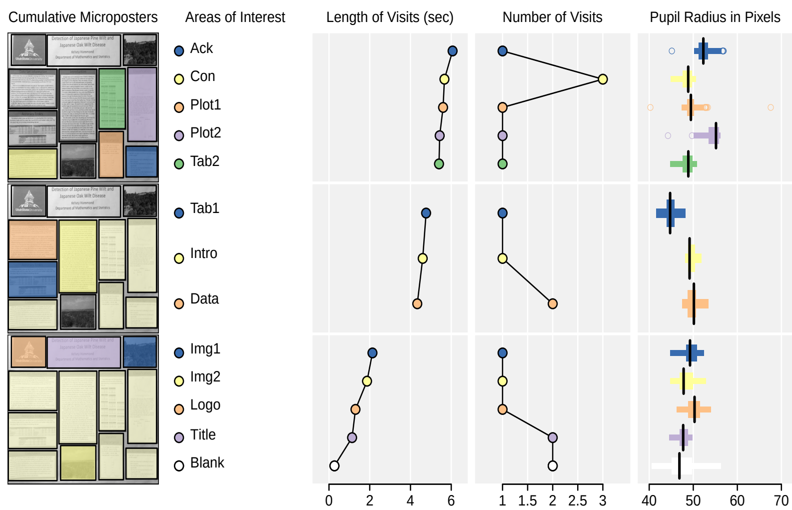

Fig. 4.5: Linked Microposter Plot

Figure 4.5 is created with the code above. Figure 4.5 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 five 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).

However, the viewing sequences can not be shown in the linked microposter plot. To overcome this disadvantage, we developed the linked timeline microposter plot

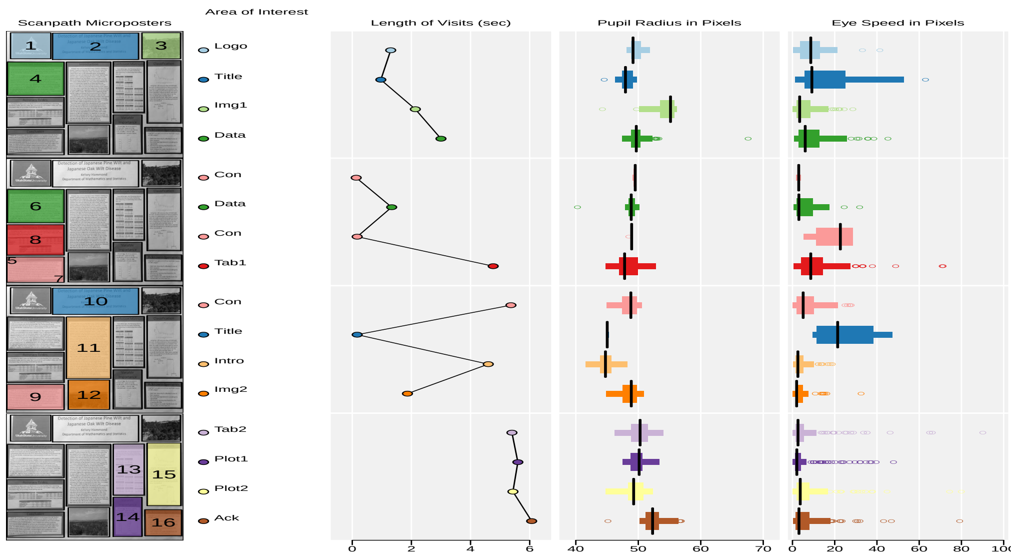

Fig. 4.6: Linked Timeline Microposter Plots sorted by viewing sequence

and the linked scanpath microposter plot to visualize the timeline and scanpath of the eye movements, respectively. Figure 4.6 shows an example of the linked timeline microposter plot sorted by the temporal viewing sequences, i.e., the data is presented based on the timeline of the eye tracking data. The numbers labeled inside the AOIs indicate the viewing sequences. 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 4.7 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 4.3) and the scanpath visualization (Figure 4.4), 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 4.3 of the title is not very intuitive from the plot as

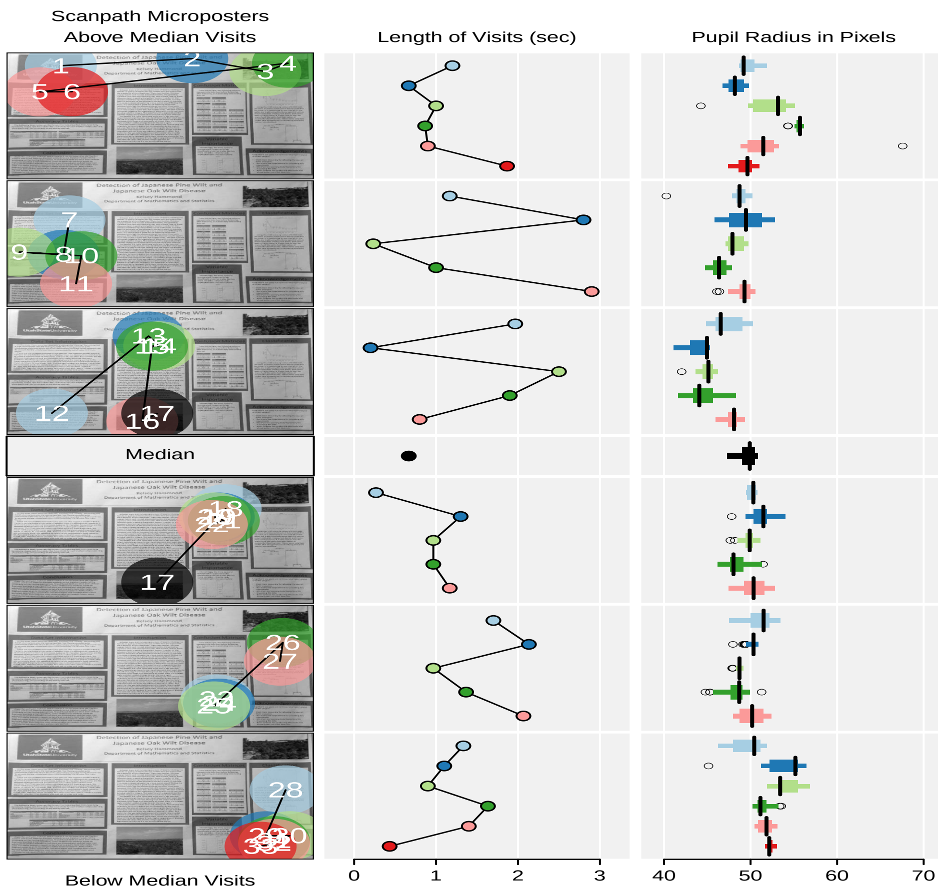

Fig. 4.7: Linked Scanpath Microposter Plots sorted by viewing sequence

well as how long the participant has spent on the revisit. In Figure 4.6, 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 self-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, and 6 data points. The data presented in each colomn can be self-defined and the colors can also be changed. Figure 4.8 shows the linked microposter plot created.

```
DrawEyeLMPlot(poster.loc = system.file("extdata", "poster_colored.jpg",
                                       package = "EyeTrackR"),
              grayscale = TRUE,
              AutomaticLayout = FALSE,
              data = posterdat.all,
              posterVBorders = AOIName,
              Layout = c(6, 1, 6),
              panel.types = c("poster", "legend", "dot",
                              "dot", "boxplot", "boxplot"),
              panel.width = c(2.9, 2.7, 3, 3, 3, 3),
              hdColors = c(rev(brewer.pal(6, "Accent")), "#000000"),
              columns.att = conlums.att,
              main.title = ""
)
```

4.4 Conclusion and Future Work

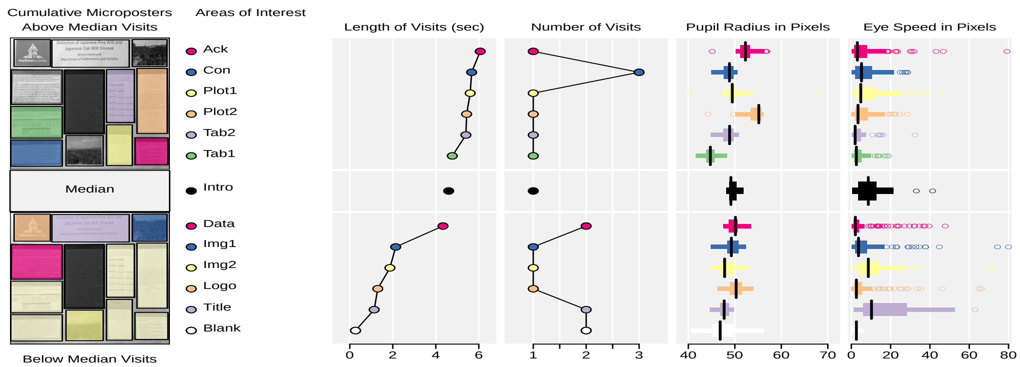

Fig. 4.8: Linked Microposter Plot

In this chapter, we presented the **EyeTrackR** R package and how it can be applied to process and visualize eye tracking data. 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, 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. (2017b) has 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 on 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.

CHAPTER 5
CONCLUSION

In this dissertation, we presented an approach to automatically extract data from mobile eye trackers. Feature detection and image registration are adopted to process the consecutive images from an eye tracking video. A heuristic object detection approach is designed to find the focus point when only the video data from a mobile eye tracker is provided. Our proposed methods enable researchers to automatically obtain the coordinates in the same coordinate system indicating where people are looking at a poster, so that they don't have to manually look at every single video frame to decide the focus points on the poster. We incorporated the data extraction in our **EyeTrackMat** Matlab toolbox.

We proposed the linked microposter plots family as new means to visualize how people are looking at a poster. Compared to the commonly used eye tracking visualization tools, the linked microposter plots family is able to more effectively visualize eye tracking data. The linked microposter plots family overcomes the disadvantages of the commonly used eye tracking data visualization tools, making it possible to present multiple variables in a single plot accurately as well as the relationships of these variables. With the perceptual groupings, readers can quickly identify clusters of mapped AOIs with similar values of the sorting variable. We implemented the data extraction, commonly used eye tracking techniques, and the linked microposter plots family in our **EyeTrackR** R package. The visualization tools in the R package are applicable for visualizing both the extracted mobile eye tracking data and the data from static eye trackers.

Future work includes displaying data for multiple participants via a single linked microposter plot, linked timeline microposter plot, or a linked scanpath microposter plot. Dynamic microposter plots could also be implemented in the future, such as through an R shiny app.

The limitation of this dissertation is that the main application is focusing on looking at scientific posters. The data processing, extraction, and visualization can also be applied when people are looking at power point slides, book chapters, commercial electronic posters, etc. The application in looking at other objects could also be explored. Symanzik et al. (2017b) 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 preliminary results for two test participants are presented in Symanzik et al. (2017b) using the exploratory data analysis techniques introduced in this dissertation. In the upcoming study, "two groups of participants will be examined: one group with extensive yoga experience, and one group with minimal experience with actions that require stability (e.g., yoga, gymnastics, ballet dancing, etc.)." (Symanzik et al., 2017b). The participants are eqquiped with a mobile eye-tracking device that will allow us to see what they are looking at as they make stability judgments. The data collection is ongoing. The data processing, visualization, and analysis are based on this dissertation, and will be extended to another Ph.D. dissertation.

REFERENCES

Adler-Grinberg, D., Stark, L., 1978. Eye movements, scanpaths, and dyslexia. American Journal of Optometry & Physiological Optics 55 (8), 557–570.

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, June 2012. Eye tracker systems manual. Mobile Eye XG, Manual version 1.4.

Applied Science Laboratories, 2013a. Applied Science Laboratories Applications.
URL http://www.asleyetracking.com/Site/

Applied Science Laboratories, 2013b. D7 remote desktop eye tracking optics.
URL http://www.asleyetracking.com/Site/Products/EYETRAC7Series/RemoteDesktop/tabid/66/Default.aspx

Applied Science Laboratories, 2013c. Sports performance training and research.
URL http://www.asleyetracking.com/Site/Applications/SportsPerformance/tabid/263/Default.aspx

Applied Science Laboratories, 2013d. Where it all began...
URL http://www.asleyetracking.com/Site/Company/AboutASL/tabid/115/Default.aspx

Applied Science Laboratories, 2015. ASL Results Plus GM.
URL http://www.razor3d.co.kr/eyetrac/pdf/ASL%20Results%20Plus%20GM%20(Offline%20Analysis).pdf

Babcock, J. S., Pelz, J. B., 2004. Building a lightweight eyetracking headgear. In: Proceedings of the 2004 Symposium on Eye Tracking Research & Applications. ACM, pp. 109–114.

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/

Barfoot, K. M., Matthew, M., Callaway, A. J., Jul. 2012. Combined eeg and eye-tracking in sports skills training and performance analysis: an archery case study. In: World Congress of Performance Analysis of Sport IX, University of Worcester.
URL http://www.alpha-active.com/Alpha-Active_%20WCPAS9_Final.pdf

Barthelme, S., 2016a. imager: Image Processing Library Based on 'CImg'. R package version 0.31.
URL https://CRAN.R-project.org/package=imager

Barthelme, S., Feb. 23 2016b. New R package for Eyelink eye-trackers. R-bloggers.
URL http://www.r-bloggers.com/new-r-package-for-eyelink-eye-trackers/

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (SURF). Computer Vision and Image Understanding 110 (3), 346–359.

Berg, D. J., Boehnke, S. E., Marino, R. A., Munoz, D. P., Itti, L., 2009. Free viewing of dynamic stimuli by humans and monkeys. Journal of Vision 9 (5), 1–15.

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).

Boraston, Z., Blakemore, S.-J., 2007. The application of eye-tracking technology in the study of autism. The Journal of Physiology 581 (3), 893–898.

Brain Imageing & Analysis Center, Aug. 04 2014. Region of interest based eyetracking analysis.
URL `https://wiki.biac.duke.edu/biac:analysis:roi_et`

Brainard, D. H., 1997. The Psychophysics toolbox. Spatial Vision 10, 433–436.

Breeze, J., Dec. 8 2011. Eye tracking: Best way to test rich app usability. UX Magazine (505).
URL `https://uxmag.com/articles/eye-tracking-the-best-way-to-test-rich\-app-usability`

Burch, M., Schmauder, H., Raschke, M., Weiskopf, D., 2014. Saccade plots. In: Proceedings of the Symposium on Eye Tracking Research and Applications. ACM, pp. 307–310.

Caldara, R., Miellet, S., 2011. iMap: a novel method for statistical fixation mapping of eye movement data. Behavior Research Methods 43 (3), 864–878.

Card, S. K., Demiralp, Ç., Cirimele, J., 2014. The VERP explorer — a tool for applying recursion plots to the eye-movements of visual-cognitive tasks. In: 2014 Specialist Meeting — Spatial Search.

Carr, D., Pearson, J., 2015. Linked micromap plots for U. S. states. R version 1.0.5. URL `https://CRAN.R-project.org/package=micromapST`

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.

Convertino, G., Chen, J., Yost, B., Ryu, Y.-S., North, C., 2003. Exploring context switching and cognition in dual-view coordinated visualizations. In: Proceedings of the International Conference on Coordinated and Multiple Views in Exploratory Visualization. IEEE, pp. 55–62.

Cooke, L., 2005. Eye tracking: How it works and how it relates to usability. Technical Communication 52 (4), 456–463.

Cornelissen, F. W., Peters, E. M., Palmer, J., 2002. The Eyelink toolbox: eye tracking with Matlab and the Psychophysics toolbox. Behavior Research Methods, Instruments, & Computers 34 (4), 613–617.

Dalmaijer, E., 2013-2016. PyGaze: Open source eye-tracking software and more. URL `http://www.pygaze.org/docs/`

Dalmaijer, 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.

De Beugher, S., Brône, G., Goedemé, T., 2014. Automatic analysis of in-the-wild mobile eye-tracking experiments using object, face and person detection. In: 2014 International Conference on Computer Vision Theory and Applications (VISAPP). Vol. 1. IEEE, pp. 625–633.

Dink, J., Ferguson, B., 2015. eyetrackingR: An R library for eye-tracking data analysis. Github.
URL `http://www.eyetracking-r.com/`

Dodge, R., Cline, T. S., 1901. The angle velocity of eye movements. Psychological Review 8 (2), 145–157.

Du Toit, P. J., Kruger, P. E., Chamane, N., Campher, J., Crafford, D., Dec 2009. Sport vision assessment in soccer players. African Journal for Physical, Health Education, Recreation and Dance 15 (4), 594–604.

Fischler, M. A., Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the Association for Computing Machinery (ACM) 24 (6), 381–395.

Fong, J., Dec. 6 2012. The meaning of pupil dilation. The Scientist.
URL `http://www.the-scientist.com/?articles.view/articleNo/33563/title/The-Meaning-of-Pupil-Dilation/`

Foulsham, T., Kingstone, A., 2011. Look at my poster! Active gaze, preference and memory during a poster session. Perception 40 (11), 1387–1389.

Garg, S., Dec. 18 2015. Red object detect in live video using matlab. URL https://arduinopro.wordpress.com/2015/12/18/red-object-detector-in-live-video-using-matlab/

Godwin, H., 2012. R package for analysing eye-tracking data. Github. URL https://github.com/hjgodwin/eyeTrackR/

Goldberg, J. H., Helfman, J. I., 2010a. Comparing information graphics: A critical look at eye tracking. In: Proceedings of the 3rd BELIV'10 Workshop: Beyond Time and Errors: Novel Evaluation Methods for Information Visualization. ACM, pp. 71–78.

Goldberg, J. H., Helfman, J. I., 2010b. 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

Granka, L., Rodden, K., 2006. Incorporating eyetracking into user studies at google. In: Workshop Position Paper presented at Computer Human Interaction. URL http://static.googleusercontent.com/media/research.google.com/zh-CN//pubs/archive/34377.pdf

Granka, L. A., Joachims, T., Gay, G., 2004. Eye-tracking analysis of user behavior in www search. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 478–479.

Grindinger, T., Duchowski, A. T., Sawyer, M., 2010. Group-wise similarity and classification of aggregate scanpaths. In: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications. ACM, pp. 101–104.

Hassaballah, M., Abdelmgeid, A. A., Alshazly, H. A., 2016. Image features detection, description and matching. In: Hassaballah, M. (Ed.), Image Feature Detectors and Descriptors. Vol. 630. Springer, Cham, pp. 11–45.

Heminghous, J., Duchowski, A. T., July 2006. icomp: a tool for scanpath visualization and comparison. In: ACM SIGGRAPH 2006 Research Posters. No. 186.

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.

Holmqvist, K., Wartenberg, C., 2005. The role of local design factors for newspaper reading behaviour — an eye-tracking perspective. Lund University Cognitive Studies 127, 1–21.

Hope, R. M., 2012. eyetracking: Eyetracking Helper Functions. R package version 1.1.
URL `http://CRAN.R-project.org/package=eyetracking`

Institute of Electrical and Electronics Engineers, 1990. IEEE Standards Glossary of Image Processing and Pattern Recognition Terminology. IEEE.

Jacob, R. J. K., Karn, K. S., 2003. Commentary on Section 4. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In: Hyona, J., Radach, R., Deubel, H. (Eds.), The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research. Elsevier Science BV, Oxford, England, pp. 573–605.

Judd, T., Ehinger, K., Durand, F., Torralba, A., 2009. Learning to predict where humans look. In: IEEE 12th International Conference on Computer Vision (ICCV). pp. 2106–2113.

Korman, S., Reichman, D., Tsur, G., Avidan, S., 2013. Fast-match: Fast affine template matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2331–2338.

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.

Law, B., Atkins, M. S., Kirkpatrick, A. E., Lomax, A. J., Mackenzie, C. L., 2004. Eye gaze patterns differentiate novice and experts in a virtual laparoscopic surgery training environment. In: Proceedings of the 2004 Symposium on Eye Tracking Research & Applications. ACM, pp. 41–48.

Leutenegger, S., Chli, M., Siegwart, R. Y., 2011. Brisk: Binary robust invariant scalable keypoints. In: 2011 IEEE International Conference on Computer Vision (ICCV). pp. 2548–2555.

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, Boston, MA.

Li, C., Symanzik, J., 2017. EyeTrackR: An R package for extraction and visualization of eye tracking data from people looking at posters.

Li, C., Zhang, B., Symanzik, J., 2017. EyeTrackMat: a Matlab toolbox for extraction of mobile eye tracking data with an application on people looking at posters (in preparation).

Li, X., Çöltekin, A., Kraak, M.-J., 2010. Visual exploration of eye movement data using the space-time-cube. In: Fabrikant, S. I., Reichenbacher, T., van Kreveld, M., Schlieder, C. (Eds.), Geographic Information Science: Lecture Notes in Computer Science. Vol. 6292. Springer, Berlin, Heidelberg, pp. 295–309.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60 (2), 91–110.

McAndrew, A., 1 2004. An introduction to digital image processing with matlab notes for scm2511 image processing. School of Computer Science and Mathematics, Victoria University of Technology, 1–264.

Mirman, D., Dixon, J. A., Magnuson, J. S., 2008. Statistical and computational models of the visual world paradigm: Growth curves and individual differences. Journal of Memory and Language 59 (4), 475–494.

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`

Navab, A., Gillespie-Lynch, K., Johnson, S. P., Sigman, M., Hutman, T., 2012. Eye-tracking as a measure of responsiveness to joint attention in infants at risk for autism. Infancy 17 (4), 416–431.

Noton, D., Stark, L., 1971. Scanpaths in eye movements during pattern perception. Science 171 (3968), 308–311.

Nyström, M., Ögren, M., Aug. 30 2012. How illustrations influence performance and eye movement behaviour when solving problems in vector calculus. In: Lunds Tekniska Högskola (LTHs) 7: e Pedagogiska Inspirationskonferens.

URL https://www.lth.se/fileadmin/lth/genombrottet/konferens2012/23_Nystroem_OEgren.pdf

Oleson, J. J., Cavanaugh, J. E., McMurray, B., Brown, G., 2015. Detecting time-specific differences between temporal nonlinear curves: Analyzing data from the visual world paradigm. Statistical Methods in Medical Research (0), 1–22.

Payton, Q., Olsen, T., Weber, M., McManus, M., Kincaid, T., 2015a. Micromap: A package for linked micromaps. Journal of Statistical Software 63 (3).
URL https://www.jstatsoft.org/article/view/v063i02

Payton, Q., Olsen, T., Weber, M., McManus, M., Kincaid, T., 2015b. Micromap: Linked micromap plots. R version 1.9.2.
URL https://CRAN.R-project.org/package=micromap

Pereira, S., Pun, T., 2000. Robust template matching for affine resistant image watermarks. IEEE Transactions on Image Processing 9 (6), 1123–1129.

Pickle, L. W., Pearson, J. B., Carr, D. B., 2015. MicromapST: Exploring and communicating geospatial patterns in US state data. Journal of Statistical Software 63 (3).
URL https://www.jstatsoft.org/article/view/v063i03

Piqueras-Fiszman, B., Alcaide-Marzal, J., Spence, C., 2012. An application of eye-tracking technologies to study consumers'attention to packaging sensory attributes. XVI Congreso Internacional de Ingenieria de Proyectos Valencia, 1952–1963.

Purucker, C., Landwehr, J. R., Sprott, D. E., Herrmann, A., 2013. Clustered insights: Improving eye tracking data analysis using scan statistics. International Journal of Market Research 55 (1), 105–130.

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/

Räihä, K.-J., Aula, A., Majaranta, P., Rantala, H., Koivunen, K., 2005. Static visualization of temporal eye-tracking data. In: Costabile, M. F., Paternò, F. (Eds.), IFIP International Federation for Information Processing 2005, Lecture Notes in Computer Science 3585. Springer, Berlin, Heidelberg, and New York, pp. 946–949.

Raschke, M., Blascheck, T., Burch, M., 2014. Visual analysis of eye tracking data. In: Huang, W. (Ed.), Handbook of Human Centric Visualization. Springer, New York, pp. 391–409.

Raschke, M., Chen, X., Ertl, T., 2012. Parallel scan-path visualization. In: Proceedings of the Symposium on Eye Tracking Research and Applications. ACM, pp. 165–168.

Rayner, K., 1998. Eye movements in reading and information processing: 20 years of research. Psychological Bulletin 124 (3), 372–422.

Richardson, D. C., Dale, R., 2005. Looking to understand: The coupling between speakers' and listeners' eye movements and its relationship to discourse comprehension. Cognitive Science 29, 1045–1060.

Ristovski, G., Hunter, M., Olk, B., Linsen, L., 2013. Eyec: Coordinated views for interactive visual exploration of eye-tracking data. In: Information Visualisation (IV), 2013 17th International Conference. IEEE, pp. 239–248.

Roberto, B., 2009. Template matching techniques in computer vision: theory and practice. John Wiley & Sons.

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. Orb: an efficient alternative to sift or surf. In: 2011 IEEE International Conference on Computer Vision (ICCV). pp. 2564–2571.

Seedorff, M., Oleson, J., Brown, G., Cavanaugh, J., McMurray, B., 2015. bdots: Bootstrapped Differences of Time Series. R package version 0.1.2.
URL `http://CRAN.R-project.org/package=bdots`

SensoMotoric Instruments, 2016. SMI gaze & eye tracking systems.
URL `http://www.smivision.com/en/gaze-and-eye-tracking-systems/home.html`

Sharif, B., Maletic, J. I., 2010. An eye tracking study on the effects of layout in understanding the role of design patterns. In: 26th IEEE International Conference on Software Maintenance (ICSM). Timisoara, Romania, pp. 1–10.

Slykhuis, D. A., Wiebe, E. N., Annetta, L. A., Dec. 2005. Eye-tracking students' attention to powerpoint photographs in a science education setting. Journal of Science Education and Technology 14 (5-6), 509–520.

Sogo, H., 2013. GazeParser: an open-source and multiplatform library for low-cost eye tracking and analysis. Behavior Research Methods 45 (3), 684–695.

Sogo, H., Oct 2015. Gazeparser/simplegazetracker.
URL `http://gazeparser.sourceforge.net/#`

SR Research, 2013. Sr research: Fast, accurate, reliable eye tracking.
URL `http://www.sr-research.com/index.html`

Symanzik, J., Carr, D. B., 2008. Interactive linked micromap plots for the display of geographically referenced statistical data. In: Handbook of Data Visualization. Springer, Berlin, Heidelberg, pp. 267–294.

Symanzik, J., Carr, D. B., 2013. Linked micromap plots in R. In: Cho, S. H. (Ed.), Asian Regional Section of the IASC. Proceedings of IASC-Satellite Conference for the 59th ISI WSC & The 8th Conference of IASC-ARS, pp. 213–218.

Symanzik, J., Carr, D. B., McManus, M. G., Weber, M. H., 2017a. Micromaps. Wiley StatsRef: Statistics Reference Online, 1–11.

Symanzik, J., Dai, X., Weber, M. H., Payton, Q., McManus, M. G., 2014. Linked micromap plots for South America—general design considerations and specific adjustments. Revista Colombiana de Estadística: Current Topics in Statistical Graphics 37 (2), 451–469.

Symanzik, J., Li, C., Zhang, B., Studenka, B., McKinney, E., 2017b. Eye tracking in practice: A first analysis of a study on human postures. In: 2017 JSM Proceedings. American Statistical Association, Alexandria, VA.

Szeliski, R., 2011. Feature detection and matching. In: Computer Vision. Springer London, pp. 181–234.

Tanimoto, S., April 2010. Digital images. Department of Computer Science & Engineering, University of Washington, Seattle, WA.
URL `http://www.cs.washington.edu/research/metip/about/digital.html`

The DFKI Evaluation Center for Language Technology, Nov. 8 2006. Tips and tricks for mobile eye tracking.
URL `http://www.lt-eval.org/Seiten/Englisch/tips-tricks.htm`

The MathWorks Inc., 1994-2017. Videoreader.
URL https://www.mathworks.com/help/matlab/ref/videoreader.html

Tobii, 2015a. The history of Tobii.
URL http://www.tobii.com/group/about/history-of-tobii/

Tobii, 2015b. Sports research.
URL http://www.tobiipro.com/fields-of-use/human-performance/sports-research/

Tobii Technology, 2017. Tobii pro studio.
URL https://www.tobiipro.com/product-listing/tobii-pro-studio/#Packages

Torr, P. H., Zisserman, A., 2000. Mlesac: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding 78 (1), 138–156.

Tsai, M.-J., Hou, H.-T., Lai, M.-L., Liu, W. Y., Yang, F.-Y., 2012. Visual attention for solving multiple-choice science problem: An eye-tracking analysis. Computers & Education 58 (1), 375–385.

Tuytelaars, T., Mikolajczyk, K., et al., 2008. Local invariant feature detectors: a survey. Foundations and Trends® in Computer Graphics and Vision 3 (3), 177–280.

van Renswoude, D., 2015. gazepath: Gazepath Transforms Eye-Tracking Data into Fixations and Saccades. R package version 1.0.
URL http://CRAN.R-project.org/package=gazepath

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

Wickham, H., 2009. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.
URL http://had.co.nz/ggplot2/book

Yang, F.-Y., Chang, C.-Y., Chien, W.-R., Chien, Y.-T., Tseng, Y.-H., 2013. Tracking learners' visual attention during a multimedia presentation in a real classroom. Computers & Education 62, 208–220.

Zhao, Y., Cook, D., Hofmann, H., Majumder, M., Chowdhury, N. R., 2013. Mind reading using an eyetracker to see how people are looking at lineups. International Journal of Intelligent Technologies & Applied Statistics 6 (4), 393–413.

Zhegallo, A. V., Marmalyuk, P. A., 2015. Etran-r extension package for eye tracking results analysis. Perception 44 (8-9), 1–7.

Zheng, D., May 14 2014. getVideoFrames, MathWorks.
URL https://www.mathworks.com/matlabcentral/fileexchange/46615-getvideoframes-vid--startt--endt--step--savetodir

APPENDICES

VITA

Chunyang “Catherine” Li

Education

2013 - 2017

PhD in Statistics, Data Visualization with Specialty in Eye Tracking, Utah State University, Logan, Utah

2011 - 2013

Master’s Degree, Statistics, Statistical Learning with Specialty in Random Forests, Utah State University, Logan, Utah

2007 - 2011

Bachelor’s Degree in Economics, with Major Focus in Statistics, Southwestern University of Finance and Economics, Sichuan, China

Professional Experience

Decision Support Outcome Analyst Intern

May 2014 - Aug 2014, Intermountain Healthcare, Salt Lake City, Utah

- Constructed a new way of connecting the company’s database system more securely
- Developed a new weighted two-dimensional kernel density map to visualize the patients geographical data
- Estimated the effectiveness of the pain service program and patients’ safety programs

- Analyzed the patients' visits pattern using time series analysis techniques

Research Assistant

Jan 2013 - May 2014, Utah State University, Logan, Utah

- Literature review on volatility estimation models for high frequency data
- Developed linear regression model for interval data

Data Analyst

May 2013 - Aug 2013, Start-Smart K3-Plus Validation Study, Logan, Utah

- Analyzed the cost and effectiveness of an extended school year program
- Summarized and combined course schedules in Excel
- Cleaned data for household surveys using Stata and SAS
- Analyzed the influence of the program for each family and wrote analysis reports

Projects

Software Development for Eye Tracking Data

- Developed EyeTrackMat Matlab toolbox to extract and visualize eye tracking data from mobile eye trackers
- Developed EyeTrackR R Package for processing and visualizing eye tracking data of mobile and static eye trackers

Object Recognition in Images

- Preprocessed eye tracking images in R

- Used the combination of Support Vector Machine, Bagged Trees, Gradient Boosted Machine, Generalized Boosted Regression Models, and Random Forests to identify images

Driver Telematics Analysis Use Telematic Data to Identify a Driver Signature

- Extracted features from telematic data in R
- Used Random Forest probability estimation approach to predict the probability of a driver signature

Text Mining: Evaluate Spam Detection Software SpamAssassin

- Extracted hundreds of Junkmails filtered by SpamAssassin and rebuilt dataframes in R
- Analyzed the sending information in HTML format and the email contents
- Visualized the performance of SpamAssassin in R and wrote the assessment report

Computer Skills

Microsoft Office Suite with excellence in Excel

Specialty Software: R, Matlab, SAS, EViews, SPSS, Stata, Maple, and Latex

Programming Languages: Visual Basic, SQL, Python, and C++

Language Skills

English (fluent)

Chinese (native)

Teaching Experience

Graduate Instructor, Business Statistics, Introduction to Statistics, Introduction to Statistical Methods, and Beginning Algebra

Aug 2013 - December 2017, Utah State University, Logan, Utah

- Taught lectures to 20-50 students from all majors; was solely responsible for course content
- Designed and graded all quizzes, homework, and academic exams

Recitation Leader, Business Statistics, Introduction to Statistics, and Pre-Calculus Aug 2011 - May 2014, Utah State University, Logan, Utah

- Taught recitations to approximately 100 students per semester, and helped students with any questions
- Graded quizzes, homework, and academic exams

SAS Lab Tutor, Linear Regression and Time Series Analysis May 2011 - Aug 2011, Utah State University, Logan, Utah

- Helped students with SAS programming, homework problems and graded the homework
- Lectured when the professor was out of town

Selected Publications

- Sun, Y. and **Li, C.** Linear Regression for Interval-valued Data: A New and Comprehensive Model. Linear Regression Models, Analysis and Applications. Mathematics Research Developments. Editor: Vera L. Beck.

- **Li, C.** Probability Estimation in Random Forest, MS Thesis, Utah State University, May 2013

- **Li, C.** and Symanzik, J. The Linked Microposter Plot as a New Means for the Visualization of Eye Tracking Data. 2016 JSM Proceedings. American Statistical Association. Alexandria, VA.

- **Li, C.**, and Symanzik, J. EyeTrackR: An R Package for Extracting and Visualizing Data from Mobile and Static Eye Trackers. 2017 JSM Proceedings. American Statistical Association. Alexandria, VA

- Symanzik, J., **Li, C.**, Zhang, B., Studenka, B., Mckinney, E., Eye-Tracking in Practice: A First Analysis of a Study on Human Postures. 2017 JSM Proceedings. American Statistical Association. Alexandria, VA

- **Li, C.**, Symanzik, J., Merrill, A., and Zabriskie, B. (submitted). Identifying Geographical Clusters of High-Utilizers in HealthCare with Traditional and Weighted Hot Spot Mapping

- **Li, C.**, Symanzik, J., and Zhang, B. (in preparation) EyeTrackMat: a Matlab Toolbox for Extraction of Mobile EyeTracking Data with an Application on People Looking at Posters

Conference Presentations

EyeTrackR: An R Package for Extracting and Visualizing Data from Mobile and Static Eye Trackers, Joint Statistical Meetings. Baltimore, MD, 2017

The Linked Microposter Plot as a New Means for the Visualization of Eye Tracking Data, Joint Statistical Meetings. Chicago, IL, 2016

Linear Regression for Interval-valued Data: A New and Comprehensive Model, Joint Statistical Meetings. Boston, MA, 2014

Services/Leadership

Class Commissary, Activity Committee and Public Affair, 2009 - 2011

Activity Planner, Organization Department, Students' Union of the School, 2009 - 2010

Volunteer, the School of Statistics, Red Cross of China, 2008 - 2009

Volunteer Helper, AeA in Wenchuan Earthquake, Jun 2008 - Oct 2008

Awards

Dissertation Fellowship, 2016

Individual Scholarship, Innovation (Top 3% of the School), 2010

Second Prize Scholarship (Top 10% of the School), 2009

Best Design Award, Dormitory Insignia (Top 5% of the School), 2009

Merit Student (Top 10% of the School), 2008