

Development and Evaluation of a Gaze Feedback System Integrated into EyeTrace

Kai Otto

Perception Engineering, University of Tübingen
Tübingen, Germany
kai.otto@student.uni-tuebingen.de

David Geisler

Perception Engineering, University of Tübingen
Tübingen, Germany
david.geisler@uni-tuebingen.de

Nora Castner

Perception Engineering, University of Tübingen
Tübingen, Germany
castnern@informatik.uni-tuebingen.de

Enkelejda Kasneci

Perception Engineering, University of Tübingen
Tübingen, Germany
enkelejda.kasneci@uni-tuebingen.de

ABSTRACT

A growing field of studies in eye-tracking is the use of gaze data for realtime feedback to the subject. In this work, we present a software system for such experiments and validate it with a visual search task experiment. This system was integrated into an eye tracking analysis tool. Our aim was to improve subject performance in this task by employing saliency features for gaze guidance. This realtime feedback system can be applicable within many realms, such as learning interventions, computer entertainment, or virtual reality.

CCS CONCEPTS

• **Human-centered computing** → **Interaction paradigms**; • **Applied computing** → **E-learning**;

KEYWORDS

E-Learning, eyetracking, gaze-based feedback

ACM Reference Format:

Kai Otto, Nora Castner, David Geisler, and Enkelejda Kasneci. 2018. Development and Evaluation of a Gaze Feedback System Integrated into EyeTrace. In *ETRA '18: 2018 Symposium on Eye Tracking Research and Applications, June 14–17, 2018, Warsaw, Poland*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3204493.3204561>

1 INTRODUCTION

In many real-world situations, we are confronted with visual displays where information needs to be extracted and interpreted. Often, it is almost unavoidable that important details are missed, as the human eye and brain can only process parts of the screen at a time [Jarodzka et al. 2012]. We are limited due to the fovea restricting our field of view: where we see sharply within two degrees [Holmqvist et al. 2011]. Therefore, visual search merges our

fixations in order to perceive all present information. To stimulate perception, we can learn how to spread our attention more effectively. Effective gaze guiding can be implemented as graphical user interfaces or other visual feedback forms and has shown promising outcomes in a range of professions. For instance, in air traffic control [Mackworth 1948], piloting a vehicle [Wetzel et al. 1998], and reading medical imagery [Jarodzka et al. 2012]. Here, task detection and interpretation under certain circumstances is not only time consuming to learn, but can also be safety critical.

Gaze guidance or supportive highlighting of on-screen information can help in a number of scenarios. For instance, teaching systematic search of medical x-ray images [Kok et al. 2016; Kundel and La Follette Jr 1972; Van der Gijp et al. 2017]. Additionally, in air traffic control simulation, where stimuli is dynamic, gaze guidance highlights the relevant information as it appears [Mackworth 1948].

These attentional guiding systems not only highlight relevant information areas, but also needs to account already perceived information [Jarodzka et al. 2013]. From the literature, it is known that eye tracking offers insight into a user's perception through their gaze behavior [Holmqvist et al. 2011]. Thus in this work, we focus on effective visualization of online gaze behavior. Specifically, realtime gaze feedback that visualizes already viewed regions and incorporates more information from the periphery.

2 RELATED WORK

Employing eye movement data in the educational context has offered insight into how to model gaze. Most notable are the eye movement modeling examples (EMMEs); Where visual guidance to directly influence gaze behavior was employed by Jarodzka et al., in order to increase subjects' interpretation performance of medical records [Jarodzka et al. 2012] and a biological classification task [Jarodzka et al. 2009, 2013]. For [Jarodzka et al. 2012], eye movement data of experts were visualized by blurring areas they did not look at: i.e. non-relevant information. For [Jarodzka et al. 2009], experts' gaze was visualized as yellow circles on a stimulus image. For both studies, the model example incorporated gaze data post hoc.

Qvardfort and colleagues [Qvarfordt et al. 2010] found that applying white circular occlusions to fixations from a previous free-viewing over the stimuli was able to reduce the workload during a visual search task while increasing the true positive rate of targets

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ETRA '18, June 14–17, 2018, Warsaw, Poland

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5706-7/18/06...\$15.00

<https://doi.org/10.1145/3204493.3204561>

found [Qvarfordt et al. 2010]. Thus, participants were able to notice areas that they did not search in the free-viewing phase because the already search areas were occluded.

ScreenMasker [Orlov and Bednarik 2016] is an open source software by Orlov and Bednarik intent on developing a customizable system that visualizes gaze behavior. Their gaze contingent system creates a pattern mask over the on screen stimuli. Then, it uses gaze coordinates from the eye tracker to subtract the pattern, or unmask, where the subject is gazing in real time. For this system, an NVIDIA graphics card with CUDA framework was used and was shown to perform with very low latencies [Orlov and Bednarik 2016]. Thus, offering low to none temporal offset that could disturb a user.

We propose a platform integrated in a publicly available eye tracking analysis tool. The multiple plugins integrated offer an experimental center and a realtime gaze feedback option. Our system was tested and capable of running on a standard computer.

3 SOFTWARE DEVELOPMENT

Eyetrace [Kübler et al. 2015] is a software providing state-of-the-art algorithms for eye tracking data visualization, statistical analysis, event detection, AOI generation, saccade clustering, and scanpath analysis and supports a variety of eye trackers. All algorithms are parameterizable and the parameters together with the visualization and statistics can be exported. Therefore, we decided to extend this existing software, which is publicly available at <http://www.ti.uni-tuebingen.de/Eyetrace.eyetrace.0.html>

For our experiment (detailed in section 4), we used the EyeTribe eye tracker [Ooms et al. 2015] since it was already supported by Eyetrace. We extended this plug-in to support online usage whereas previously, only recording and importing the eye tracking data was available. The developed application interface also allows for extending the plugin to other eye trackers and online calibration.

3.1 The Experimenter

The *Experimenter* plug-in for EyeTrace was developed for creating and performing remote eye tracking experiments. The central part is the Designer widget, shown in figure 1, which has the following capabilities:

- Create and modify the experiment design where each index block is highly customizable (Figure 1 area 1 and 2).
- Import and export experiment designs as CSV file (Figure 1 area 2).
- Record subject data together with name, group and dominant eye (Figure 1 area 3).
- Select the Eyetracker to be used (Figure 1 area 4).
- Select an interruption key (Figure 1 area 5).
- Start/cancel the experiment run (Figure 1 area 5).

In the *Experimenter*, a researcher can manually organize an experiment design offering customization of stimuli, time of presentation, gaze feedback, and keypress interruptions. These experimental designs can be exported and saved as a CSV file, for additional data collection. Additionally, the ability to import experiment designs in CSV file format allows for the option of autogenerating randomized experiment designs with a simple script in any programming language with CSV parsing libraries or text editor.

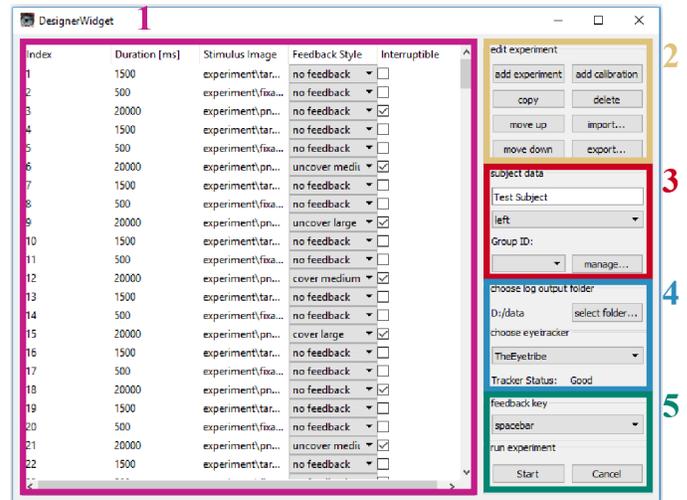


Figure 1: Designer Widget GUI. Here, experiments can be designed, and managed. The workflow of the experiment is organized (1) and can be modified (2) and each participant's data is defined (3). For each experiment, an eye tracker is selected (4) as well as a key for interruption (5).

Each step of the experiment design can either be a calibration, or a stimulus presentation/recording. In calibration, the eyetracker's calibration from the API is employed. In stimulus presentation, the durations (in milliseconds), the filepaths (if none is chosen, a white screen will be displayed), and whether the step is interruptible through keypress are customizable. The interruptible option is optimal for experimental designs where reaction-time or decision-making tasks are evaluated. Additionally, the researcher can also present the online gaze feedback for any number of stimuli, as described further in section 3.2.

The 'Start' button in the Designer widget initiates the experiment: Data logging starts here as well. Stimuli are shown in the Presenter widget, a second full screen widget that gets called. Ideally, if the researcher has two monitors, the main window of Eyetrace and the Designer widget can be displayed to the researcher and the participant only sees the Presenter widget. The researcher can always cancel the experiment with the Designer widget's 'Cancel' button. Otherwise, the Experimenter runs through the designed stimuli list and terminates at the end, closing the Presenter widget.

On the data handling side, timestamps, gaze coordinates, and keypresses are recorded in a log file. Internally, they are stored in a data structure for the experimental session for later calculations and analyses.

3.2 Realtime Feedback

In the realtime feedback, the user's gaze data is visualized on the screen as he or she is performing a task. In order to achieve low and relatively constant response times of the feedback system, intermediate results are stored in a cache. Then, the system only has to process new gaze data when it is repeatedly called.

Triggered by a timer, every 7 ms (approximately 144 Hz) the screen drawing method of the Presenter widget gets called to update



Figure 2: Screenshot of an experiment trial, showcasing the ‘cover’ (a) and ‘uncover’ (b) feedback condition. Stimulus: Ilya Repin, “Unexpected Visitors”, 1884-1888. Oil on canvas. public domain <https://commons.wikimedia.org/wiki/>.

the screen content. This trigger calls the currently active realtime feedback implementation to draw over the stimulus.

Presently, two feedback algorithms are implemented, plus the default ‘no feedback’ condition. First, the ‘cover’ feedback occludes the user’s gaze coordinates on the stimulus with opaque circles as illustrated in figure 2a. Second, the ‘uncover’ feedback unoccludes a semitransparent cover in a similar manner to the former condition as illustrated in figure 2b. Essentially, this feedback is the complement of the former applied to the mask overlay. For both conditions, there is no decay of feedback for older gaze points.

Both feedback conditions use a white mask-like image overlaid over the original stimuli, and the feedback effects the masks’ alpha channel, meaning its opacity is changed. Each event where the Realtime Feedback class is called, the list of new gaze points is run through and circles are drawn on the mask overlay for each new gaze point coordinate. For a video recording illustrating both the ‘cover’ and the ‘uncover’ feedback methods, please refer to the supplementary materials.

In both feedback conditions, the mask is either transparent (for covered) or semi-transparent (for uncovered). the alpha channel on this map is then changed based on the gaze coordinates. The compositing method adds or subtracts the circle’s alpha values to the existing mask corresponding to gaze coordinates, giving the effect of decreasing or increasing transparency the longer the subject looks at a certain spot. However, a lower bound threshold is given to the circle’s alpha value to prevent any part of the stimulus becoming invisible. Each circle consists of a radial gradient, projecting outwards from the circle’s center to its edge. This effect makes the feedback appear smoother, removing distracting, sharp edges (see figure 2). The compositing method updates the mask with the new gaze points each time the trigger timer event takes place. Then, the mask gets drawn over the stimulus.

3.3 Gaze Behavior with Feedback

In order to evaluate our online gaze feedback system, we measured performance in a visual search task with the feedback as an independent variable. We propose that both the cover and uncover feedback conditions will affect gaze behavior compared to no feedback at all.

The cover feedback method could have two effects. One, subjects may be less likely to look a second time at areas of the stimulus they already looked at, as the saliency gets decreased after looking the

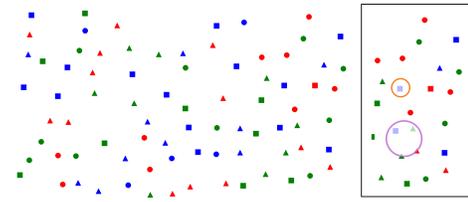


Figure 3: An example stimuli, as used in the experiment. In this case, subjects had to decide whether there is a red square visible or not. Stimuli was presented at full resolution. On the right, are the respective sizes for cover small (orange) and cover large (purple) conditions.

first time. Meaning their time to scan the image is shorter resulting in shorter reaction times. Two, subjects will have longer saccades, as the feedback includes distractors in the periphery of what is currently being fixated, which otherwise might have been the area of the next fixation.

Additionally, the uncover feedback could result in subjects’ scanning behavior becoming more systematic because the saliency is being reduced, resulting in fewer inconsistent saccades across the stimulus. This systematic search behavior effect was also found in [Jarodzka et al. 2013], the spotlight condition (where non relevant information is blurred) resulted in faster, more efficient detection of relevant information.

4 METHODS

For our experiment, we used a Windows 10 computer with a 27 inch monitor (resolution 1920×1080 pixel) as display device, and the Eyetribe eyetracker. Evaluation of the Eyetribe with regards to scientific usability can be found in the literature [Dalmaijer 2014; Ooms et al. 2015].

The visual search task was performed with images consisting of either 80 distractors (target absent) or 79 distractors plus the target item (target present). An example image is shown in figure 3. In total, 100 images were generated¹. Order of stimuli presentation was randomized for each participant.

For each stimuli, the target item was centered on screen for 1.5 seconds. Following target presentation, a fixation cross was visualized for 0.5 seconds, and then the stimulus presentation. To signal their decision, participants pressed a button on the keyboard: Either keypress *m* (right index finger) for target present, or keypress *y* (left index finger) for target absent.

The feedback methods, as introduced in section 3.2, were parameterized the following way. For each of the two feedback types (uncovering, covering), we chose two diameters: 100 or 200 pixel diameter. The control condition was no gaze feedback. Conditions, in conjunction to order of stimuli, were randomized to allow for a within-subject evaluation.

A total of 18 participants (17 university students; five wore glasses) took part. They were positioned roughly 60 cm away from the screen. A 9-Point calibration was performed using the Eye-Tribe’s calibration software. Following the experiment, participants

¹The images had equal distribution of color and shape of the target item, and its absence/presence.

Table 1: Reaction times for correct answers when target was absent. t -test is between intervention and control condition. “*” indicates a significant result.

condition	μ [ms]	σ [ms]	t -value	p
Target absent				
control	5256.49	2715.04	n.a.	
uncover small	5144.21	2531.49	0.39	0.70
uncover large	5213.20	2430.17	0.16	0.88
cover small	5198.08	2370.27	0.21	0.83
cover large	4693.89	1897.47	2.28	0.02*

filled out a self-report regarding perceived performance and experience.

Eye movement data was evaluated in Eyetrace. Fixations were calculated with the I-DT algorithm [Salvucci and Goldberg 2000] with the following parameters: minimum duration of 50 ms, maximum radius of 20 pixel, maximum outliers of 0. Saccades were then calculated as the spatial representation between two fixations. In addition to eye movement data, response error rate and reaction time for were calculated. Reaction time was defined as time between onset of stimulus to keypress.

5 RESULTS

From the questionnaire responses, it was found that self reports of effectiveness and helpfulness for both feedback conditions did not significantly differ compared to the control condition of no feedback.

Participant behavior for each experimental trial where they responded correctly was evaluated. Only 4.3% of the total trials were excluded because they were incorrect responses. A low correlation ($r = 0.31$) ruled out any effect of target distance from center fixation cross on reaction time.

5.1 Performance

The reaction times for target absent trials ($\mu_{\text{absent}} = 5095.12$ ms, $SD_{\text{absent}} = 2405.44$ ms) were significantly longer than for the target present trials ($\mu_{\text{present}} = 2254.12$ ms, $SD_{\text{present}} = 1346.08$ ms: $t = -30.16$, $p < 0.001$).

Regarding feedback intervention and reaction time, it was found that when the target was absent, the cover large (200 pixel diameter) condition had significant differences in reaction time. The Welch’s unequal variances t -test² as shown in Table 1 found that this condition had significantly shorter reaction times ($t = 2.28$, $p = 0.023$).

When the target was present, reaction times were overall shorter, though there was no significant differences between feedback conditions here.

5.2 Gaze Behavior

Similar to reaction time, the effect of target absent or present on numbers of saccades was highly significant. Where there were more saccades when the target was present ($\mu_{\text{present}} = 16.75$, $SD_{\text{present}} =$

²Welch’s unequal variances t -test pools together all values for each condition, meaning sample sizes are larger, which increases the statistical power.

Table 2: Mean and standard deviation for saccade length. t -test is between intervention and control condition. “*” indicates a significant result with $p < 0.05$, “” for significance level $p < 0.005$.**

condition	μ_{sac} [px]	σ_{sac} [px]	t -value	p
control	313.84	148.48	n.a.	
uncover small	329.26	157.14	-2.16	0.045*
uncover large	333.11	152.79	-2.52	0.022*
cover small	344.18	164.12	-3.11	0.006*
cover large	333.19	150.23	-3.31	0.004**

Table 3: Mean and standard deviation for number of fixations needed to complete the task for each of the five conditions, split for target is absent. t -test between control and intervention condition, “*” indicates a significant result.

condition	μ	σ	t -value	p
Target absent				
control	17.36	9.53	n.a.	
uncover small	17.19	10.99	0.22	0.83
uncover large	16.86	8.99	0.78	0.45
cover small	16.30	9.67	2.31	0.03*
cover large	16.08	8.07	1.61	0.13

9.32) than when the target was absent ($\mu_{\text{absent}} = 7.44$, $SD_{\text{absent}} = 3.47$: Welch’s unequal variances test, $t = 8.85$, $p < 0.001$). However, feedback conditions showed no significant effect on number of saccades.

More interesting, saccade length was affected by the feedback. Here, the Welch’s unequal variances t -test (values in table 2) also reported significant differences for feedback conditions compared to control, where the feedback conditions had longer saccade lengths.

Concerning fixations, fixation duration was not significantly different between control ($\mu_{\text{control}} = 118.47$ ms, $SD_{\text{control}} = 49.45$ ms) and all feedback conditions (For example, $\mu_{\text{coverLarge}} = 116.67$ ms, $SD_{\text{coverLarge}} = 40.34$ ms: $t = 0.99$, $p = 0.34$).

Similar to saccades and reaction times, the number of fixations was higher for target absent. Where for both controls: target present ($\mu_{\text{present,control}} = 7.22$, $SD_{\text{present,control}} = 3.07$) and target absent ($\mu_{\text{absent,control}} = 17.36$, $SD_{\text{absent,control}} = 9.53$).

There were no significant differences between feedback and control condition when the target was present. However, when the target was absent (see table 3), an effect for the cover small feedback (diameter 100px) condition was found. Although there were no significant differences, a trend can also be seen for less fixations in the cover and uncover large conditions compared to the control.

6 DISCUSSION

Generally, it took subjects longer to correctly decide if a target is absent, than it took them to decide if a target is present. The difference was highly significant; this experimental result reproduces a

well documented effect of target presence or absence on reaction time [Chun and Wolfe 1996; Wolfe et al. 1989].

Concerning how the intervention influenced subject behavior, we can see that even with our rather simple feedback methods, we were able to induce a change in subjects. An increase in periphery employed is apparent from the longer saccades for both covering and uncovering interventions. However, only the cover large condition, where a semi-transparent circle with a 200 pixel diameter overlaid on the gaze coordinates, increased reaction times when the target was accurately determined as absent. There was also a trend for less fixations when determining the target was absent for both cover and uncover (where the circle uncovers a semi-transparent overlay) large feedback conditions, though significantly less fixations were only found in the cover small (100 pixel diameter) feedback condition. Therefore, the realtime gaze based feedback algorithms developed for the system produced an effect on gaze behavior in the visual search task.

Interestingly enough, the self-reports from the participants did not indicate that the feedback helped or improved their performance. In contrast, their reaction times as well as their eye movement differences showed that gaze feedback indeed affected their behavior compared to no gaze feedback. Participants also reported that none of the feedback conditions were distracting in any way. Therefore, the gaze feedback system we developed appears to be unobtrusive, yet effective.

Regarding a more effective gaze model, the current experiment found that the large cover affected reaction time for target absent being correctly determined. However, overall correct detection was extremely high at 96%. Future work into effective gaze modeling could look into more complex visual search tasks to see whether gaze modeling improves performance.

Jarodzka and colleagues [Jarodzka et al. 2013] found that using either the spotlight condition (where non relevant information is blurred) or the dot condition for EMMEs were both effective in modeling gaze behavior. However, each condition affected a certain aspect of learning and performance, where the spotlight condition affected visual search and the dot condition affected interpretation [Jarodzka et al. 2013]. In our system, the uncover feedback algorithm is relatively similar to their spotlight condition, where both present a clear unaffected gaze area, and occlude the other areas ([Jarodzka et al. 2013]: blurring, ours: opaque mask). Additionally, [Jarodzka et al. 2013]'s dot condition is similar to our cover condition; however, ours covers the gaze area with a semi-transparent mask that does not hide the stimulus information underneath. Whether our online gaze feedback would be beneficial for learning environments is of great interest in future research.

7 CONCLUSION

In this work, we introduced a novel software system for eyetracking experimentation, which allows realtime feedback to the subject. We successfully validated the implementation in a visual search task study. The current system was integrated into our eye tracking analysis tool EyeTrace. Now, this tool provides experimental design and testing in addition to the analysis and visualization of eye tracking data.

REFERENCES

- Marvin M Chun and Jeremy M Wolfe. 1996. Just say no: How are visual searches terminated when there is no target present? *Cognitive psychology* 30, 1 (1996), 39–78.
- Edwin Dalmaijer. 2014. *Is the low-cost EyeTribe eye tracker any good for research?* Technical Report. PeerJ PrePrints.
- Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. 2011. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford.
- Halszka Jarodzka, Thomas Balslev, Kenneth Holmqvist, Marcus Nyström, Katharina Scheiter, Peter Gerjets, and Berit Eika. 2012. Conveying clinical reasoning based on visual observation via eye-movement modelling examples. *Instructional Science* 40, 5 (2012), 813–827.
- Halszka Jarodzka, Katharina Scheiter, Peter Gerjets, Tamara van Gog, and Michael Dorr. 2009. How to convey perceptual skills by displaying experts' gaze data. In *Proceedings of the 31st annual conference of the cognitive science society*. 2920–2925.
- Halszka Jarodzka, Tamara van Gog, Michael Dorr, Katharina Scheiter, and Peter Gerjets. 2013. Learning to see: Guiding students' attention via a Model's eye movements fosters learning. *Learning and Instruction* 25 (2013), 62–70.
- Ellen M Kok, Halszka Jarodzka, Anique BH de Bruin, Hussain AN BinAmir, Simon GF Robben, and Jeroen JG van Merriënboer. 2016. Systematic viewing in radiology: seeing more, missing less? *Advances in Health Sciences Education* 21, 1 (2016), 189–205.
- Thomas C Kübler, Katrin Sippel, Wolfgang Fuhl, Guilherme Schievelbein, Johanna Aufreiter, Raphael Rosenberg, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2015. Analysis of eye movements with Eyetrace. In *International Joint Conference on Biomedical Engineering Systems and Technologies*. Springer, 458–471.
- Harold L Kundel and Paul S La Follette Jr. 1972. Visual search patterns and experience with radiological images. *Radiology* 103, 3 (1972), 523–528.
- Norman H Mackworth. 1948. The breakdown of vigilance during prolonged visual search. *Quarterly Journal of Experimental Psychology* 1, 1 (1948), 6–21.
- Kristien Ooms, Lien Dupont, Lieselot Lapon, and Stanislav Popelka. 2015. Accuracy and precision of fixation locations recorded with the low-cost Eye Tribe tracker in different experimental setups. *Journal of eye movement research* 8, 1 (2015).
- Pavel A Orlov and Roman Bednarik. 2016. ScreenMasker: An open-source gaze-contingent screen masking environment. *Behavior research methods* 48, 3 (2016), 1145–1153.
- Pernilla Qvarfordt, Jacob T Biehl, Gene Golovchinsky, and Tony Dunningan. 2010. Understanding the benefits of gaze enhanced visual search. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. ACM, 283–290.
- Dario D Salvucci and Joseph H Goldberg. 2000. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*. ACM, 71–78.
- A Van der Gijp, CJ Ravesloot, H Jarodzka, MF van der Schaaf, IC van der Schaaf, Jan PJ van Schaik, and Th J Ten Cate. 2017. How visual search relates to visual diagnostic performance: a narrative systematic review of eye-tracking research in radiology. *Advances in Health Sciences Education* 22, 3 (2017), 765–787.
- Paul A Wetzel, Gretchen M Anderson, and Barbara A Barelka. 1998. Instructor use of eye position based feedback for pilot training. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 42. SAGE Publications Sage CA: Los Angeles, CA, 1388–1392.
- Jeremy M Wolfe, Kyle R Cave, and Susan L Franzel. 1989. Guided search: an alternative to the feature integration model for visual search. *Journal of Experimental Psychology: Human perception and performance* 15, 3 (1989), 419.