

Link to data:

<https://atreus.informatik.uni-tuebingen.de/seafiler/d/8e2ab8c3fdd444e1a135/?p=%2Fsimulator>

Eye movement velocity and gaze data generator for evaluation, robustness testing and assess of eye tracking software and visualization tools.

Wolfgang Fuhl¹ and Enkelejda Kasneci¹

Eberhard Karls University, Sand 14, 72076 Tuebingen, Germany,
{wolfgang.fuhl, enkelejda.kasneci}@uni-tuebingen.de

Abstract. Eye movements hold information about human perception, intention, and cognitive state. Various algorithms have been proposed to identify and distinguish eye movements, particularly fixations, saccades, and smooth pursuits. A major drawback of existing algorithms is that they rely on accurate and constant sampling rates, impeding straightforward adaptation to new movements such as microsaccades. We propose a novel eye movement simulator that i) probabilistically simulates saccade movements as gamma distributions considering different peak velocities and ii) models smooth pursuit onsets with the sigmoid function. Additionally, it is capable of producing velocity and two-dimensional gaze sequences for static and dynamic scenes using saliency maps or real fixation targets. Our approach is also capable of simulating any sampling rate, even with fluctuations. The simulation is evaluated against publicly available real data using a squared error. The Matlab code for the simulator can be downloaded at <http://ti.uni-tuebingen.de/Projekte.1801.0.html> or used in EyeTrace.

Keywords: eye movement, simulation, generation, fixation, saccade, smooth pursuit, saliency maps, gaze mapping

1 Introduction

Eye movements hold valuable information about a subject, and his cognitive states [3, 16] and are also important for the diagnosis of defects and diseases of the eyes (many examples can be found in [20]). Therefore, the detection and differentiation of eye movement types have to be accurate. Most algorithms for eye movement detection apply different dispersion, velocity or acceleration thresholds and validate the detected eye movements based on their duration. This approach seems to be unsatisfactory [1] at its current state. This is partially due to unstable/dynamic sampling rates of eye tracking devices, task-specific sources of noise, the interpolation method applied to the data by the eye tracker, and several more [5, 8]. Depending on the task at hand, different thresholds are proposed in the literature [13]. It is especially difficult to adjust these thresholds for inconsistent sampling rates and noise which is not annotated by the eye tracker. Some commercial eye-tracker differ between tracking the eye and pupil

and re-detecting them after a tracking loss, where the latter requires significantly more processing time and thus results in a decreased frame rate. Therefore, the identification of eye movements is still a difficult task; it complicates to confidently generalize research findings across experiments [1].

We propose an eye movement simulator to generate data similar to the data of eye-trackers. This is especially useful if algorithms have to be evaluated or data is necessary to test the robustness of software working with eye tracking data. In addition, the generated data can be used to assess visualizations. The proposed simulator currently contains the following features:

- Generate velocity profiles of Saccades, Fixations and Smooth Pursuits based on scientific findings.
- Generate random sequences following predefined orders.
- Generate static and dynamic sampling rates.
- Supports any sampling rate.
- Generate gaze positions for static images using saliency maps.
- Generate new eye tracking data using real data mapped to a saliency map or to real fixation targets.
- Generate gaze positions for dynamic scenes like EPIC-Kitchens [6] using saliency maps or real fixation targets.

2 Related work

While there are well-established findings about the gaze signal itself, its synthesis is still challenging. In the Eyecatch [32] simulator, a Kalman filter is used to produce a gaze signal for saccades and smooth pursuits. While the signal itself was similar to real eye-tracking recordings, the jitter was missing. The first approach for rendering realistic and dynamic eye movements was proposed in [19], where the main focus was on saccadic eye movements. It also included smooth pursuits, binocular rotations (vergence) and the combination of eye and head rotations. The first data-driven approaches were proposed in [21] and [25]. Both simulate the head and eye movements together in order to generate eye-tracking data. The main disadvantage of [21] was that head motion seemed to trigger eye movement. In fact, the head orientation is only changed if the necessary amplitude of the eye is larger than a specific threshold [22] ($\approx 30^\circ$). Another data-driven approach was proposed in [18], where an automated framework for head motion, gaze, and eyelid simulation was developed. The framework generates data based on speech input using trained Gaussian Mixture Models. While this approach is capable of synthesizing nonlinear data, it only generates unperturbed gaze directions. The approach in [10] models eye rotations using specific eye related quaternions for oculomotor rotations as proposed in [29]. The main disadvantage of this approach is that the synthetic eyes cannot be rotated automatically. The approach in [31] produces gaze vectors and eye images to train machine learning approaches for gaze prediction, but does not synthesize realistic eye movements.

All of the aforementioned approaches have their origin in computer graphics with the goal to generate visually realistic head movement and gaze data. The

main application of those simulators is to produce realistic interacting virtual humans using parametric models [2, 24]. This leads to the disadvantage, that all movements in the generated data are perfect optimal representatives. In reality, the raw gaze data in eye movements contains noise introduced either through actual movements such as microsaccades or inaccuracies of the used eye-tracker. The first approach to simulate a realistic scan path, i.e., a sequence of fixations and saccades, on static images was proposed in [4]. They use a saliency map together with a unified Bayesian model to generate realistic random walks over a stimulus. A pure gaze data simulation approach including noise was proposed in [7]. Based on this approach, [9] further improves the noise synthesis by simulating jitter as a normal distribution.

Our approach combines the before mentioned publications by simulating velocity profiles of three eye movement types and map them to saliency maps. In contrast, our approach is capable of simulating noise as uniform and normal distribution and also allows to produce any sampling rate (static and dynamic). We also extended the mapping functionality to allow the usage of real fixation targets and use dynamic scenes instead of only static images.

3 Simulation

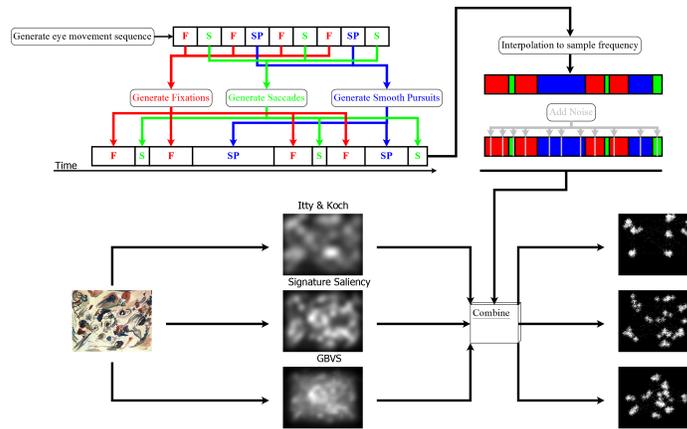


Fig. 1. Work-flow of generating eye movement data. First, a sequence of eye movement types is generated. In the second step, a model of each eye movement type is generated (F: Fixation, S: Saccade, SM: Smooth pursuit). This model allows for an almost infinite sampling rate, which is in the next stage interpolated to a target sampling rate (Red: Fixation, Green: Saccade, Blue: Smooth pursuit). Finally, noise is added on top of the signal (gray). The last stage is the mapping of the generated sequence to target locations.

The entire work-flow of the simulator is shown in Figure 1. Generating an eye movement velocity profile is done in four steps. The first step chooses a sequence of eye movement types (Fixation, Saccade, Smooth pursuit) without any time or velocity constraints. Afterward, each movement type in this sequence is assigned a velocity profile generated by preliminary set parameters. The mathematical model behind these profiles allows sampling at an extremely high, almost arbitrary rate. The target sampling rate is obtained by interpolating the computed frequency, which also allows for dynamically adjusting the target sampling rate. In the next step, noise is added which represents measurement errors or blinks. The generated velocity profile is then mapped to two-dimensional locations on a stimulus which are created using saliency maps ([12, 15, 14]) or real fixation targets. This also allows to remap real eye tracking data or use a dynamic stimulus taking into account the duration of the individual eye movement types. Each step of this eye movement simulator is described in the following subsections in more detail. The simulator also includes a random walker generator to model fixation direction [11]; saccade and smooth pursuit directions are generated randomly (but consistently within a movement) since this is stimuli- and task-dependent.

3.1 Eye movement sequence

Generating a sequence of eye movement types can be done either by sampling from a uniform distribution, setting it manually, or by following construction constraints. In case of the uniform distributed eye movements, the generator script randomly selects between three types of eye movements. If the amount of each type is specified a priori, the probability is automatically adjusted. This means that after each insertion the probabilities are computed based on the remaining quantity of each type to favor higher quantities. This process can also be constrained, e.g., by forcing the algorithm to insert a saccade after each fixation or before a smooth pursuit.

3.2 Fixation

Fixations are generated based on two probability distributions which can be specified and parametrized. The first distribution determines its duration, the second the consistency of the fixation. For the duration and consistency, the minimum and maximum can be set. As distributions, the simulator provides Normal and Uniform random number generation. For the Normal distribution, the standard deviation can be specified. consistency describes the fluctuations in the velocity profile and is used as such in the entire document.

In Figure 2, two artificially generated fixations are shown. The consistency was set to one degree per second and the standard deviation for the normal distribution to two (Figure 2 (a)). As can be seen in the figure, the Uniform distribution looks more similar compared to real data although we have set the consistency very high with one degree per second.

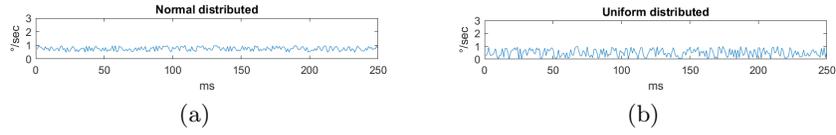


Fig. 2. Generated fixation based on a Normal (a) and Uniform (b) distribution.

3.3 Saccade

The most complex part of the eye movement generator is the saccades. For the length, we follow the same approach as for the fixations, in which a minimum and maximum length have to be set. The selectable distributions are Normal and Uniform. The result of the length also influences the maximum speed of the saccade. Therefore, the two random numbers are multiplied (both in the range between zero and one). This means that shorter saccades are limited to lower maximal velocities. To generate the velocity profile, minimum, maximum and the distribution type have to be set.

The most characteristic property of a saccade is its velocity profile. In our simulator, this is generated as a Gamma distribution. Therefore, the minimum and maximum skewness have to be specified. In [30] it was found that the Gamma function can be considered suitable to approximate saccade profiles (yet not perfect). To achieve more realistic data, a consistency minimum, maximum and distribution can be specified. This generates the jitter along the velocity profile.

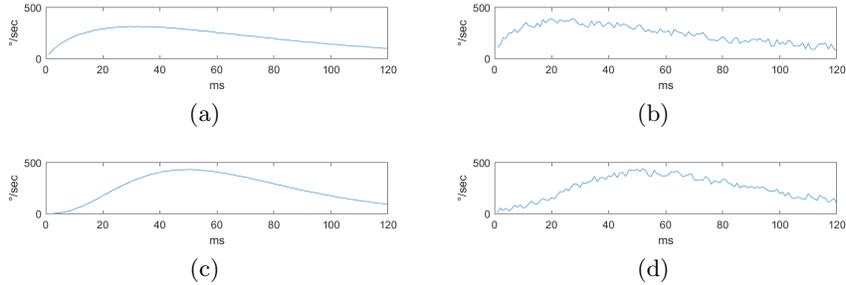


Fig. 3. Generated saccades with jitter (b,d) and without (a,c). For (a) and (b), the distribution was skewed to the left. In (c) and (d), Gamma distribution was only slightly skewed.

Figure 3 shows some generated saccades of fixed length. We simulated two large and two slightly left skewed saccades. The maximum velocity was selected from a range between 300 and 500 degrees per second. As can be seen from the Figure, the profile contains on- an offset of a saccade. The profile itself is smooth and follows the Gamma distribution. Post-saccadic movement is as of now missing in the simulator. In Figure 3(b) and (d), a small amount of jitter

was added to simulate measurement inaccuracy. This usually occurs through the approximation on image pixels or ellipse fit inaccuracy in pupil detection.

3.4 Smooth pursuit

For generating smooth pursuits we also simulate the onset following the findings in [23]. The authors did not provide a final function for the description of the velocity profile but visualized and described it precisely. The shape of the onset of a smooth pursuit follows a nonlinear growing function similar to the sigmoid function. While this equation is not scientifically proven, our framework allows to simply replace it once a better model is available. The most complex part of the pursuit model is the onset, followed by a regular movement.

The parameters that can be specified are the minimum and maximum length together with their distribution type. For the velocity and the length of the onset, the same parameters can be adjusted. To include the measuring error, the consistency parameters are also configurable. For the pursuit itself, we included linear growing, decreasing and constant profiles. In case of the growing, again the minimum, maximum and consistency function can be specified. Figure 4 shows

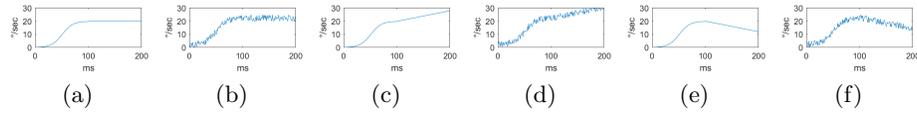


Fig. 4. Generated smooth pursuits with jitter (b,d,f) and without (a,c,e). For (a) and (b), the pursuit movement was constant. In (c,d) and (e,f) it was linear increasing and decreasing.

simulated smooth pursuits. For the visualization of the linear decreasing and increasing function, extreme values were used. The first column shows a smooth pursuit for a constantly moving object, which is often observed in laboratory experiments. The increasing and decreasing profiles are for objects which move further away or come closer to the subject with a constant speed. Other profiles may occur in real settings too, where the object has a slightly varying speed but these are future extensions of the generator and not part of this paper.

3.5 Sampling

After generating and linking the eye movements, they have to be interpolated to a sampling rate. This is necessary to simulate different recording frequencies. Here it is important to mention that not all modern eye trackers record at a constant frequency. On the one hand, image acquisition rates can vary depending on illumination changes that affect the aperture time of the camera and timestamps generated by the eye-tracker can vary in accuracy. On the other hand, image processing time, e.g. for eye and pupil detection, are not necessarily constant and

might change depending on how easy the pupil can be identified. For example, detection of the pupil is usually more time-consuming than keeping track of a previously detected pupil. Some systems, especially when running on mobile devices, may run into a state where frames are dropped in order to maintain real-time performance. We found systems where the timestamps are generated by the CPU time (which may be inaccurate for fast sampling rates) and even timestamps that are generated after image processing. Therefore, our simulator is capable of simulating varying sampling rates. The parameters for this step are the minimum and maximum sampling rate and also the consistency function. The interpolation itself computes the mean of all values from the last sampling position to the new sampling position.

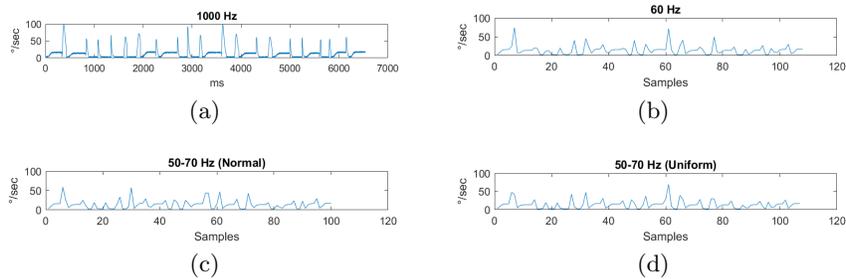


Fig. 5. Generated velocity profile of an eye movement sequence (a). In (b), the data is sampled at 60Hz without variations. (b) and (c) vary between 50 and 70 Hz with the Normal and the Uniform distribution.

In Figure 5(a) a generated velocity profile is shown. The initial sampling frequency was set to 1000 Hz but any other sampling rate is possible. For (b), a constant sampling frequency of 60 Hz was used. In (c), the sampling frequency varies between 50 and 70 Hz (with a mean of 60 Hz), wherein the Normal distribution was used as the random number generator. It differs significantly from the constant sampling rate in (a) and also has a different length. For (d), the sampling frequency also varied between 50 and 70 Hz with the difference that the Uniform distribution was used as the random number generator. The length is therefore similar to the constant sampling rate but it still differs especially for the saccadic peaks.

3.6 Noise

For generating noise, two distributions are used: one for the location where to place the noise in the data and the second for the velocity change to apply. Therefore, the user has to specify the types for both distributions and the minimum and maximum velocity of noise. The amount of noise is specified as a percentage of the samples that should be influenced.

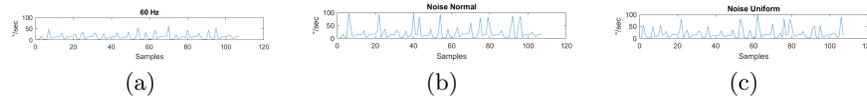


Fig. 6. Generated velocity profile of an eye movement sequence (a). In (b), noise is added based on a Normal distribution and in (c) a Uniform distribution was used.

Figure 6 shows two types of Noise added to the velocity profile shown in (a). The amount of noise added was 10%. For the Normally distributed noise in (b) it can be seen that the peaks are mostly high. In comparison to it, the Uniform distributed noise in (c) produces more peaks of different heights.

3.7 Mapping

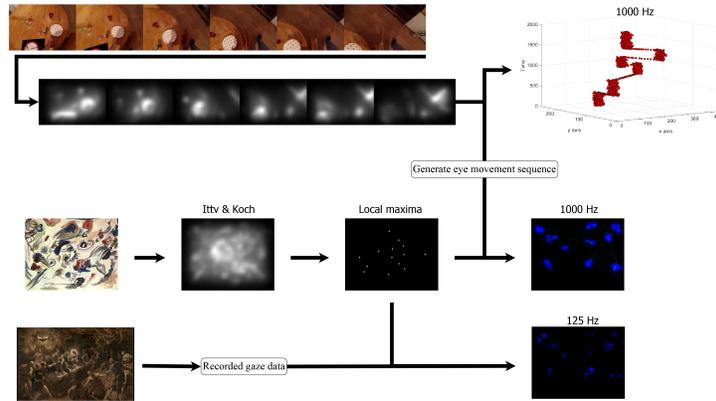


Fig. 7. Scenarios of mapping eye movement data. First, a video sequence is converted to saliency maps and afterward gaze data is mapped to them based on the frame rate. The central part shows the mapping of generated or real data to a new stimulus image. At the bottom real data is mapped to the same stimulus based on fixation locations and saccade durations. The upper series of images is taken from EPIC kitchen [6].

The mapping function of the simulator is used to produce spatial data out of the velocity profiles (Figure 7). This functionality can be used for simulated and real data (fixations, saccades, smooth pursuits). Therefore, possible fixation targets have to be identified for which our simulator includes three saliency maps ([12, 15, 14]). As locations, the local maxima of these saliency maps are used. In addition, a small random shift of the local maxima is also included as the possible target to simulated close consecutive fixations. Afterward, a type of eye movement is selected and randomly generated by a predefined parameter set including a duration range. For saccades and smooth pursuits, this duration

is interpolated to the distance between the last and new fixation target taking into account the speed of the individual sample points. In addition, a maximal deviation range can be defined based on which the gaze points differ to the straight line between both positions. For fixations, the scattering is generated based on the deviation parameter (inaccuracy of the measurement simulation). For smooth pursuits, both approaches are used to map the velocity profile between to locations. Since it is possible to generate a velocity profile out of real data to the aforementioned approaches can also be used to map real fixations, saccades, and smooth pursuits to new stimulus images. For the generation of new data out of real data for the same stimulus image, we propose to randomly select an eye movement type out of the real data and use the centers of all fixations as possible targets. An example can be seen at the bottom of Figure 7.

For dynamic scenes, the generated data can be mapped based on the same approach (local maxima of saliency maps). The only thing that differs is that the local maxima are time-dependent. This means that for a saccade the two locations have to be selected out of two different sets of local maxima which are computed based on the timestamps of frames in the video (Figure 7 top).

4 Evaluation

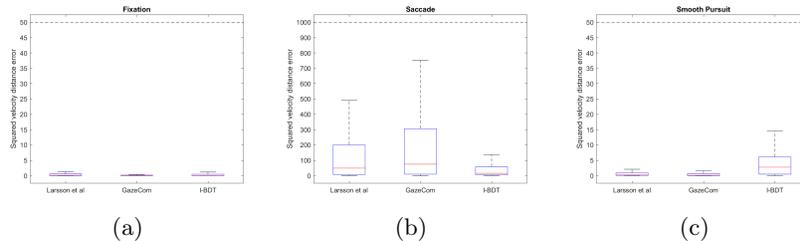


Fig. 8. Squared velocity error for the simulation per data set.

Figure 8 shows the per sample point squared error as whisker plots of our simulator in comparison to the publicly available datasets [17, 27, 28, 26]. The error was computed based on the squared difference between each sample. Therefore, we simulated each fixation, saccade, and smooth pursuit ten times with the same length as in the available data sets. For a fixation, the simulator got the information of the mean velocity and the standard deviation to generate a profile. The information of a saccade was the peak velocity and the position of this peak. For smooth pursuits, the simulator got the information of the mean velocity and the standard deviation.

As can be seen in Figure 8(b), the error for saccades was the largest. This is due to the noisy signal which is due to the inter sample velocity computation. Figure 9 shows some saccades which produced high squared errors. The red line

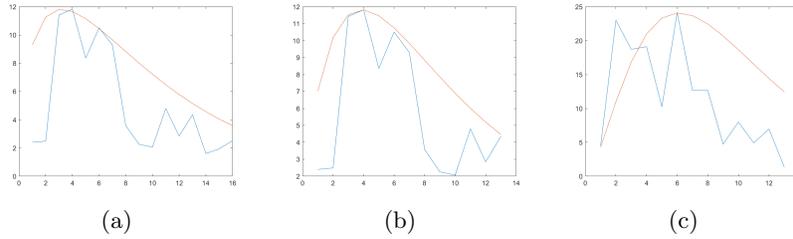


Fig. 9. Saccades with a high squared error. Red is the simulation and blue is the real data.

corresponds to the simulation result, whereas the blue line corresponds to the real data. As can be seen, the course of the velocity profile is well simulated, which is well in line with previous findings in [30]. The high errors originate mainly from measurement inaccuracies in the real data. This also highlights the difficulty in detecting eye movements in such a signal. For the data set from [26](I-BDT), the error for saccades was lowest. This is due to the low sampling rate of the used eye tracker (30 Hz), for which large fluctuations do not occur. This is similar to smoothing or using multiple samples for the velocity computation. In contrast, the smooth pursuits error was the largest in the I-BDT data set. This is because in such low sampling rates the onset of a smooth pursuit is hardly represented. Our simulator is capable of simulating this (sampling 3.5) but for the evaluation, it was not used. We only used the generators to simulate the eye movements.

5 Conclusion

We proposed a novel eye movement simulator which is capable of creating mappings to static images and dynamic scenes based on saliency maps or real fixation targets. Optionally the framework is also capable of remapping real eye tracking data onto new stimuli or generate a new scan path based on real data for the same stimuli. In addition, it can generate data for any static and dynamic sampling rate. The currently included eye movement types are Fixations, Saccades and Smooth Pursuits which can be parameterized. Variations and noise can be generated using different distributions for noise, sampling shift, eye tracker accuracy etc. Further research will be the extension of the simulator to be also capable of generating post saccadic, optokinetic and vestibular-ocular movement. In addition, smooth pursuits have to follow an object for which we want to include a key point registration and detection to compute possible locations for this type of eye movements in videos.

References

1. Andersson, R., Larsson, L., Holmqvist, K., Stridh, M., Nyström, M.: One algorithm to rule them all? an evaluation and discussion of ten eye movement event-detection

- algorithms. *Behavior Research Methods* **49**(2), 616–637 (2017)
2. Andrist, S., Pejsa, T., Mutlu, B., Gleicher, M.: Designing effective gaze mechanisms for virtual agents. In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. pp. 705–714. ACM (2012)
 3. Braunagel, C., Geisler, D., Rosenstiel, W., Kasneci, E.: Online recognition of driver-activity based on visual scanpath classification. *IEEE Intelligent Transportation Systems Magazine* **9**(4), 23–36 (2017)
 4. Campbell, D.J., Chang, J., Chawarska, K., Shic, F.: Saliency-based bayesian modeling of dynamic viewing of static scenes. In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. pp. 51–58. ACM (2014)
 5. Cornelissen, F.W., Peters, E.M., Palmer, J.: The eyelink toolbox: eye tracking with matlab and the psychophysics toolbox. *Behavior Research Methods, Instruments, & Computers* **34**(4), 613–617 (2002)
 6. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: Scaling egocentric vision: The epic-kitchens dataset. *arXiv preprint arXiv:1804.02748* (2018)
 7. Duchowski, A., Jörg, S., Lawson, A., Bolte, T., Świrski, L., Krejtz, K.: Eye movement synthesis with 1/f pink noise. In: *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*. pp. 47–56. ACM (2015)
 8. Duchowski, A.T.: A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers* **34**(4), 455–470 (2002)
 9. Duchowski, A.T., Jörg, S., Allen, T.N., Giannopoulos, I., Krejtz, K.: Eye movement synthesis. In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. pp. 147–154. ACM (2016)
 10. Duchowski, A., Jörg, S.: Modeling physiologically plausible eye rotations. *Proceedings of Computer Graphics International* (2015)
 11. Engbert, R., Mergenthaler, K., Sinn, P., Pikovsky, A.: An integrated model of fixational eye movements and microsaccades. *Proceedings of the National Academy of Sciences* **108**(39), E765–E770 (2011)
 12. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: *Advances in neural information processing systems*. pp. 545–552 (2007)
 13. Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., Van de Weijer, J.: *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford (2011)
 14. Hou, X., Harel, J., Koch, C.: Image signature: Highlighting sparse salient regions. *IEEE transactions on pattern analysis and machine intelligence* **34**(1), 194–201 (2012)
 15. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence* **20**(11), 1254–1259 (1998)
 16. Kübler, T.C., Rothe, C., Schiefer, U., Rosenstiel, W., Kasneci, E.: Subsmatch 2.0: Scanpath comparison and classification based on subsequence frequencies. *Behavior Research Methods* **49**(3), 1048–1064 (2017)
 17. Larsson, L., Nyström, M., Stridh, M.: Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit. *IEEE Transactions on Biomedical Engineering* **60**(9), 2484–2493 (2013)
 18. Le, B.H., Ma, X., Deng, Z.: Live speech driven head-and-eye motion generators. *IEEE transactions on Visualization and Computer Graphics* **18**(11), 1902–1914 (2012)
 19. Lee, S.P., Badler, J.B., Badler, N.I.: Eyes alive. In: *ACM Transactions on Graphics (TOG)*. vol. 21, pp. 637–644. ACM (2002)

20. Leigh, R.J., Zee, D.S.: The neurology of eye movements, vol. 90. Oxford University Press, USA (2015)
21. Ma, X., Deng, Z.: Natural eye motion synthesis by modeling gaze-head coupling. In: IEEE Virtual Reality Conference. pp. 143–150. IEEE (2009)
22. Murphy, H., Duchowski, A.T.: Perceptual gaze extent & level of detail in vr: looking outside the box. In: ACM SIGGRAPH conference Abstracts and Applications. pp. 228–228. ACM (2002)
23. Ogawa, T., Fujita, M.: Velocity profile of smooth pursuit eye movements in humans: pursuit velocity increase linked with the initial saccade occurrence. *Neuroscience research* **31**(3), 201–209 (1998)
24. Pejsa, T., Mutlu, B., Gleicher, M.: Stylized and performative gaze for character animation. In: Computer Graphics Forum. vol. 32, pp. 143–152. Wiley Online Library (2013)
25. Peters, C., Qureshi, A.: A head movement propensity model for animating gaze shifts and blinks of virtual characters. *Computers and Graphics* **34**(6), 677–687 (2010)
26. Santini, T., Fuhl, W., Kübler, T., Kasneci, E.: Bayesian identification of fixations, saccades, and smooth pursuits. In: Proceedings of the Symposium on Eye Tracking Research and Applications. pp. 163–170. ACM (2016)
27. Startsev, M., Agtzidis, I., Dorr, M.: Smooth pursuit. <http://michaeldorr.de/smoothpursuit/> (2016)
28. Startsev, M., Agtzidis, I., Dorr, M.: Manual & automatic detection of smooth pursuit in dynamic natural scenes. In: Proceedings of the European Conference of Eye Movements (2017)
29. Tweed, D., Cadera, W., Vilis, T.: Computing three-dimensional eye position quaternions and eye velocity from search coil signals. *Vision research* **30**(1), 97–110 (1990)
30. Van Opstal, A., Van Gisbergen, J.: Skewness of saccadic velocity profiles: a unifying parameter for normal and slow saccades. *Vision research* **27**(5), 731–745 (1987)
31. Wood, E., Baltrusaitis, T., Zhang, X., Sugano, Y., Robinson, P., Bulling, A.: Rendering of eyes for eye-shape registration and gaze estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3756–3764 (2015)
32. Yeo, S.H., Lesmana, M., Neog, D.R., Pai, D.K.: Eyecatch: simulating visuomotor coordination for object interception. *ACM Transactions on Graphics (TOG)* **31**(4), 42 (2012)