EUROGRAPHICS 2013 / I. Navazo, P. Poulin
(Guest Editors)

Volume 32 (2013), Number 2

Gaze-driven Object Tracking for Real Time Rendering

R. Mantiuk[1], B. Bazyluk[1] and R. K. Mantiuk[2]

[1] West Pomeranian University of Technology in Szczecin, Poland
[2] Bangor University, United Kingdom

Abstract

To efficiently deploy eye-tracking within 3D graphics applications, we present a new probabilistic method that predicts the patterns of user's eye fixations in animated 3D scenes from noisy eye-tracker data. The proposed method utilises both the eye-tracker data and the known information about the 3D scene to improve the accuracy, robustness and stability. Eye-tracking can thus be used, for example, to induce focal cues via gaze-contingent depth-of-field rendering, add intuitive controls to a video game, and create a highly reliable scene-aware saliency model. The computed probabilities rely on the consistency of the gaze scan-paths to the position and velocity of a moving or stationary target. The temporal characteristic of eye fixations is imposed by a Hidden Markov model, which steers the solution towards the most probable fixation patterns. The derivation of the algorithm is driven by the data from two eye-tracking experiments: the first experiment provides actual eye-tracker readings and the position of the target to be tracked. The second experiment is used to derive a JND-scaled (Just Noticeable Difference) quality metric that quantifies the perceived loss of quality due to the errors of the tracking algorithm. Data from both experiments are used to justify design choices, and to calibrate and validate the tracking algorithms. This novel method outperforms commonly used fixation algorithms and is able to track objects smaller then the nominal error of an eye-tracker.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.6]: Methodology and Techniques—

1. Introduction

The human gaze is probably the best pointing device, which is faster, more intuitive and require less effort than a computer mouse or a touch screen. Our visual system relies on constantly changing gaze, which scans the scene to create a percept of it. Given the importance of eye motion, it is disappointing that neither display devices, nor rendering methods make use of this property of the visual system. However, if we could precisely predict the gaze position, we could not only gain a very effective pointing device, but also enhance display with gaze contingent capabilities. For example, accommodation related focal cues could be simulated to enhance the visual experience [HLCC08] without a need for multi-focal displays [AWGB04].

The cost of eye tracking is falling (a do-it-yourself device can be constructed for less than 30 EURs [AMB10, MKNB12]) and so obtaining the gaze position information is affordable. Our observation is that it is not the cost of the eye-tracking systems, but their accuracy that is the main limiting factor. This is hard to overcome as most eye-trackers rely on faint corneal reflections and are affected by head movements, lids occluding the pupil, variation in lighting, shadows, and sunlight interfering with the infrared (IR) light sources. But even if a perfect registration of the eye position and orientation was possible, eye movements do not strictly follow the attention patterns of the visual system [HNA*11]. Even if an observer is focused on a single point, the eye will wander around that point because of the saccadic eye movements and tremors.

In this work we propose a gaze-tracking algorithm that combines eye-tracker data with information about the 3D scene and any animation in the scene to greatly improve the accuracy and stability of eye-tracking. Our method consists of a probabilistic model, which assigns a likelihood that one of the predefined target points is attended. The likelihood relies on the consistency of the gaze scan-paths with

the position and velocity of a target. The temporal characteristic of eye fixations is imposed by a Hidden Markov model, which steers the solution towards the most probable fixation patterns. The derivation of the algorithm is driven by the data from two eye-tracking experiments: the first experiment provides a large quantity of eye-tracker readings for 3D animations, together with the positions of the object that should be tracked. The second experiment is used to derive a quality metric scaled in just-noticeable-differences (JNDs) that quantifies the perceived loss of quality due to the errors of the tracking algorithm. The data from both experiments is used to justify design choices, calibrate and validate the tracking algorithms.

The main contributions of this work are to:

- Demonstrate that the accuracy of eye-tracking (Section 4.2) and the existing state-of-the-art of fixation estimators is insufficient for real-time computer graphics applications (Section 7).
- Derive a JND-scaled quality metric for 3D eye-tracking applications from experimental data (Section 5).
- Propose a new gaze-tracking method that utilises the information about a 3D scene and animation within the scene. The 3D scene serves as prior knowledge for the algorithm and improves the accuracy greatly above the eye-tracking error (Section 6.1).
- Demonstrate successful use of eye-tracking in the simulation of focal depth cues, as a pointing device in a computer game, and a saliency model (Section 8).

2. Background

2.1. Human eye movement and visual fixation

The field of view for both eyes spans more than 180° horizontally and 130° vertically, although, we are able to see details only in the fovea — the 2° patch of the retina located in the middle of the macula. The eye muscles enable fast gaze shifting to orient the eye such that the object of interest is projected onto the fovea. There are five types of eye movements: *saccades* - fast movements used to reposition the fovea, *smooth pursuits* - active when eyes track moving targets, *vergence* - used for depth perception to focus the pair of eyes over a target, *vestibular ocular reflex* - used to compensate head movement, and *optokinetic reflex* - used to account for the motion of the visual field [Duc07, Sun12].

From the gaze tracking technology perspective, the most important are saccadic movements and smooth pursuits (see Figure 1). Saccades last 10–100 msec and are too short for the brain to process the images transmitted by the visual system [RMB94]. In a smooth pursuit the eye follows the object of interest and matches its velocity. In our proposed tracking system we focus especially on smooth pursuits as it is common in graphics to follow moving objects.

Between saccades or during smooth pursuits the eyes tend to remain fixated for a 200-400 msec [SEDS81] on the most significant areas of an image (called the Region-of-Interest, ROI). After that, the eye moves towards a new zone of interest. This *fixation* period allows the brain to process information and see images.

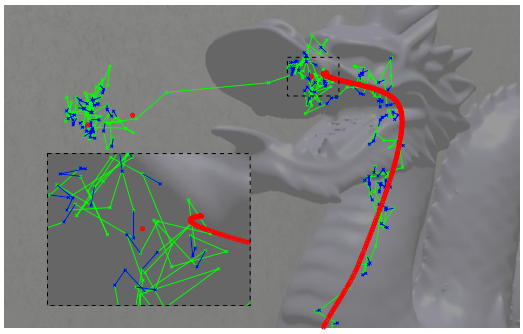

Figure 1: *Types of eye movement: the fast saccades over 20 [deg/s] are marked in green, the slower stabilised saccades suggesting existence of fixations are depicted as the blue lines and crosses, the red circles show the locations of the reference marker on which an observer was asked to look at (the marker is initially moving from the bottom to the top and then to the left). The movement of the slow saccades (fixations) defines direction of the smooth pursuit. The bright green lines represent gaze data captured with an eye tracker.*

2.2. Eye tracking

The gaze directions (or gaze positions) are captured by *eye trackers*. These devices do not measure the fixations but only collect "raw" gaze points that can be used to estimate the position of a fixation. We review fixation estimation algorithms in Section 3.2.

The most popular eye trackers employ the pupil-corneal reflection (P-CR) technique. An eye tracker usually consists of an IR camera and an IR light source, which are directed at the eye. The camera captures the image of the eye (see example in Figure 2) with the dark circle of the pupil (or white depending on the location of the light source [MM05]) and the bright corneal glint, which is a reflection of the infrared light from the outer surface of the cornea (this reflection is also called the first Purkinje image [MM05]). The pupil follows the gaze direction during eye movement while the corneal reflection stays in the same position. The relative position between the reflection and the centre of the pupil is used to estimate the gaze direction. Such an estimate is robust to small head movements.

Modelling the geometric mapping between the registered features of the eye and the display coordinates is difficult because of the initial position of the head is unknown and eye movement is complex. Therefore, the majority of eye trackers employ a non-linear approximation technique, in which

an observer is asked to look at a set of target points displayed one by one in different locations of the screen. The relation between the relative position of the pupil centre and the known position of the target points is used to fit the coefficients of a pair of second-order polynomials [MM05].

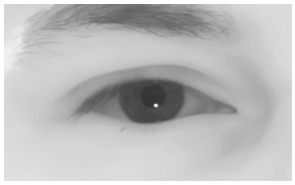

Figure 2: *Image of the dark pupil and the bright corneal reflection taken in the infrared light spectrum.*

The *accuracy of an eye tracker* refers to the difference between the actual and captured gaze direction measured in the degrees of viewing angle. The average accuracy of a typical P-CR eye tracker is close to 0.5° (see discussion in Section 4.2). This roughly corresponds to a circular region of 40 pixels diameter on a 22" display of 1680x1050 pixels observed from 65 cm. There are other gaze tracking techniques that offer better accuracy but require using chin rests or bite bars, or are intruisive, requiring electrodes around the eye, or a coil embedded into a contact lens (see [Duc07, MM05] for the review). The intruisive techniques are not suitable for casual end-user applications.

3. Related work

In this section we review existing applications of eye tracking and commonly used fixation algorithms.

3.1. Gaze tracking in computer graphics

Information about the gaze direction with the model of the visual acuity degradation (eccentricity-dependent CSF) can be used to reduce the complexity of computation in the parafoveal and peripheral regions of vision. This property is applied in view-dependent polygon simplification techniques that vary the level-of-detail (LOD) [LH01, MD01]. It can be also used to reduce sampling in ray casting [MDT09], volume rendering [LW90] or for a gaze-dependent ambient occlusion rendering [MJ12].

Gaze tracking can simulate a number of visual phenomena that depend on gaze direction and are difficult to reproduce on a display. For example, the blurring due to accommodation of the eye can be simulated by rendering scenes of reduced depth-of-field (DoF), focused at the current gaze position (see Section 8.1). Local light adaptation can be simulated in tone-mapping that adapts to the gaze position [RFM*09].

Jacob [Jac93] studied the application of eye trackers as the HCI interface, especially for people with disabilities. In this work we demonstrate the use of eye tracker as an intuitive and very fast controller in a computer game (see Section 8.2 and [Sun12]).

For most of the mentioned applications, a high accuracy for the gaze direction estimation was not crucial. They were tested on rather simple scenes with objects occupying a large area. However, most practical applications require correct identification of objects whose dimensions are below the accuracy of an eye tracker. This issue has been noticed in [HLCC08] where a so called "autofocus" technique was developed to supplement eye-tracker data with a computational attention model [Itt00] in a region of inaccurate gaze estimation. Such computational attentions models, however, were also found unreliable for scenes with complex contextual information.

3.2. Fixation techniques

The Dispersion Threshold Identification (I-DT) algorithm [Wid84] estimates fixation points in screen coordinates from eye-tracker gaze points. Since fixation points tend to cluster closely together because of their low velocity, I-DT identifies fixations as groups of consecutive points within a particular spatial *dispersion* window. Additionally, the *duration* of a fixation is limited to a time window ranging from 150 to 400 msec. The dispersion is usually measured in terms of centroid-distance — the average distance between the gaze points and their centroid computed for a time window of duration equal to a *window length*.Another fixation technique, called Velocity-Threshold Identification (I-VT) [EV95], separates fixation points and saccadic points based on their point-to-point velocity. If such velocities are less than a chosen *velocity threshold*, consecutive gaze points are collapsed into a fixation. Extension of this algorithm uses two-state hidden Markov models (HMMs) [SA98], in which the states correspond to saccades and fixations, which differ in their the velocity distributions. Although this technique improves the accuracy of fixation, the velocity-based algorithms still tend to produce inconsistent results at or near threshold values (it is difficult to find the proper threshold) [SG00] or for slow eye movements [ULIS07].

Generally, the main drawback of the fixation algorithms is that their accuracy depends on the selected parameters, which in turn depend on a scene content [Bli09]. Poorly selected parameter values can completely change the results of identification [SSC08]. The optimal parameters vary with an observer or even an observation session [Duc07, Bli09]. Interestingly, it has been shown that various fixation algorithms generate different results for the same gaze data [SSC08]. Eye trackers manufacturers often use proprietary techniques in their systems that are tuned for a particular application [Kar00]. Finally, the fixation techniques are not suitable for capturing the smooth pursuit movement, which is common in computer graphics applications.

4. Eye-tracking accuracy in 3D applications

The accuracy of eye-tracking is usually reported for simple static scenes, where an observer is asked to look at the spot target, and the distribution of the eye-tracker readings determines the error. Furthermore, the error is usually reported for the data processed with one of the fixations algorithms, which reduces the noise levels as compared to raw eye-tracker readings. However, we found that this testing scenario is not representative of dynamic 3D scenes with a multitude of moving objects, occlusions and camera movements. To collect more representative data, we measured eye-tracker readings for complex animated scenes, where the observers were instructed to follow a given target. The results will serve us as a data set for: a) more representative estimate of eye-tracker accuracy (Section 4.2); b) calibration and testing existing fixation algorithms for complex animated scenes (Section 7); and c) the basis for modelling, testing and calibration of a new algorithm (Section 6.1).

4.1. Data collection procedure

Apparatus. Our experimental setup consists of the P-CR RED250 eye tracker [SMI09] controlled by the proprietary SMI iViewX software (version 2.5) running on a dedicated PC. RED250 eye tracker captures locations of the corneal reflection (the first Purkinje image) and centre of the pupil for both eyes. The data is collected at 250 Hz, but we noticed that some samples are lost and the effective operation frequency is about 20% lower. The RED250 eye tracker is mounted under a 22" Dell E2210 LCD display with the screen dimensions 47.5x30 cm, and the native resolution of 1680x1050 pixels (60Hz). Another PC (2.8 GHz Intel i7 930 CPU equipped with NVIDIA GeForce 480 GTI 512MB graphics card and 8 GB of RAM, Windows 7 OS) was used to run our evaluation software, and to store experimental results.

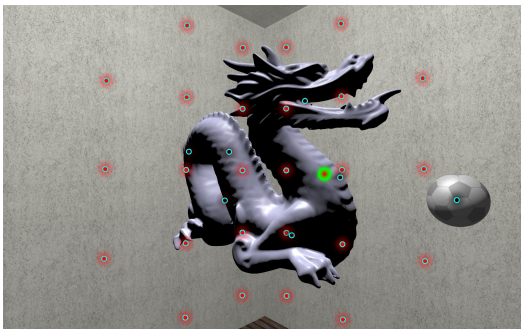

Figure 3: *An example frame from the test animation. The green circle with a red centre located in the middle of the dragon denotes the reference target. Other circles show locations of defined targets of potential focusing (these markers are not displayed during the experiment).*

Stimuli and procedure. An observer was shown one of three animations, labelled as A, B, and C, each with different objects and marker paths (the animations are included in the supplementary materials). The observer was asked to follow with the eyes the colour marker moving in the 3D scene (see example in Figure 3) while the eye-movements were recorded. The marker followed moving objects to instigate smooth motion pursuit, it jumped from one object to another at various depth levels to cause saccades, it stayed still for a longer moment, moved behind occluding objects for a short time-periods and also jumped from background to foreground objects. For every animation a set of target points of attention was distributed over the scene to act as potential fixation targets. We specified 8 targets for animation A, 9 for B, and 110 for animation C. The last case seems to be the most realistic for practical applications, although a smaller number of targets is also convenient to restrict the number of potential fixation targets.

Each observer sat at a distance of 65 cm from the display. The distance was restricted by a chin-rest. The actual experiment was preceded by a 9-point calibration and validation procedure. This procedure took about 20 seconds and involved looking at the markers displayed at 9 different points of the screen. This data was used to compute the coefficients of the polynomial mapping eye tracker camera coordinates to the display screen coordinates. Values for two eyes were averaged. We decided not to use the SMI iViewX proprietary calibration procedure because it caused a decrease in the accuracy of the eye tracker to average error equal to 2.9°. Every experimental session was preceded by a short training session where no data was recorded.

Participants. Gaze points were collected from 39 individual observers (age between 22 and 42 years old, 36 males and 3 females). From that group a different set of 20 observers was allocated to each of the tested scenes. All participants had normal or corrected to normal vision. No session took longer than 4 minutes. The participants were aware that the gaze position is registered, but they were naive about the purpose of the experiment.

4.2. Eye tracker noise

The specification of the RED250 eye tracker reports the mean error equal to 0.5°. But we managed to achieve such accuracy only in a separate experiment with static targets (16 regularly spaced points on the screen). For these measurements, the procedure was repeated 10 times for every participant so that the training effect was likely to improve the results.

In our experiment with moving targets the mean error averaged over all sessions was equal to 1.83° (std=1.07°). This error, corresponding to a circular region of diameter 150 pixels, is too high to effectively use eye-tracking in most computer graphics applications where important objects are often smaller than the error region. The error for individual observers was even higher, with the worst case equal to 3.59°.

In the best session the error was equal to 0.8°. Accuracy was calculated as the mean difference between the target and gaze points and converted into an angular measure based on the distance between the eye and the eye tracker [HNA*11]. Blinks and outliers with the error above 5° were removed from the calculations [TOB11].

5. Quality metric for eye-tracking

The experiment presented in the previous section gave us an objective measure of eye-tracking accuracy. We found, however, that such an objective measure poorly corresponds with the subjective experience of using a gaze-contingent application. We implemented gaze-contingent simulation of a depth-of-field (DoF) effect (more on that in Section 8.1) and experimented with different fixation algorithms. Our observation was that several factors affect the quality of eye-contingent DoF rendering. One obvious factor is the *error rate* (E_{rate}), which is the percentage of time a wrong object is tracked. However, in practice minimising the error rate does not necessary improve the quality of the DoF rendering. This is because the fixation algorithms calibrated for a low error rate usually produce rapid changes of fixation from one object to another, which are very distracting. Such rapid changes can be described by the number of times per second an algorithm changes fixation to a wrong object, which we call *error frequency* (E_{freq}). Therefore, a good fixation algorithm should minimise both error rate and error frequency. But, in order to do that, it is necessary to know what is the relative importance of each factor and how to combine both of them to form a single quality metric for DoF rendering. Such a quality metric is very important as we will use it to both calibrate (optimise) and evaluate fixation algorithms.

5.1. Quality assessment experiment

To find a quality metric, we conducted a quality assessment experiment. In the experiment the observers compared nine animations (each lasting 18 seconds) that differed in the way in which the fixation point deviated from the reference course. Each animation contained a scene in which a green ball was moving around the dragon figure (see Figure 4). The DoF algorithm was meant to focus on the green ball using a simulated fixation data. The lost of correct fixation was simulated to generate both long but stable (low error frequency, high error rate) and frequent but short (high error frequency, low error rate) focusing errors due to inaccurate fixation.

Each observer, sitting at a distance of approximately 65 cm from the display, was asked to compare the quality of two animations displayed sequentially and choose the one he preferred. All pairs of animations were compared in this forced-choice experiment giving 36 comparisons for each session. The data was collected for 7 observers between 24 and 42 years old, six males and one female. All participant had normal or corrected to normal vision. No session took longer than 30 minutes. Otherwise, the setup was identical as for the eye-tracking data collection experiment (see Section 4.1).

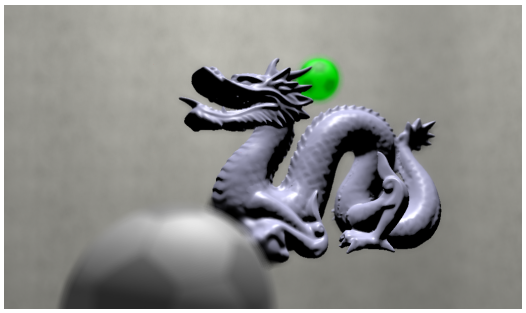

Figure 4: *An example frame from the real time simulation of the depth of field effect.*

5.2. Quality metric

To transform pair-wise comparison results into a quality scale, we used the Bayesian approach to JND scaling. The details of the approach can be found in [SF01]. In brief, the method maximises the probability that the collected data explains the experiment under the Thurston Case V assumptions [Eng00, ch. 8]. The optimisation procedure finds a quality value for each animation that maximises the probability, which is modelled by the binomial distribution. Unlike standard scaling procedures, the Bayesian approach is robust to unanimous answers, which are common when a large number of disparate conditions are compared, which was the case of our experiment.

After finding the JND-scaled quality values for each animation, we fitted the function, $Q(E_{rate}, E_{freq})$, explaining the relation between the error rate, error frequency and the quality expressed in JND units:

$$Q = -0.03312 \cdot E_{rate} - 4.358 \cdot {E_{freq}}^{0.4682} + 4.516. \quad (1)$$

The measured quality values and the fitted functions are shown in Figure 5. The plot clearly shows that the error frequency has a much larger impact on perceived quality than the error rate. The quality peaks sharply as the error frequency is reduced.

6. Gaze tracking for dynamic 3D scenes

We experimented with several fixation algorithms, but we found that none of them could produce satisfying real-time DoF simulation, regardless of the selected parameters (more on this in Section 7). This is mostly due to the limited accuracy of eye-trackers, as discussed in Section 4.2. However, in the case of our applications we have much more information than just the raw eye-tracker readings available to standard fixation algorithms. The 3D rendering delivers informa-

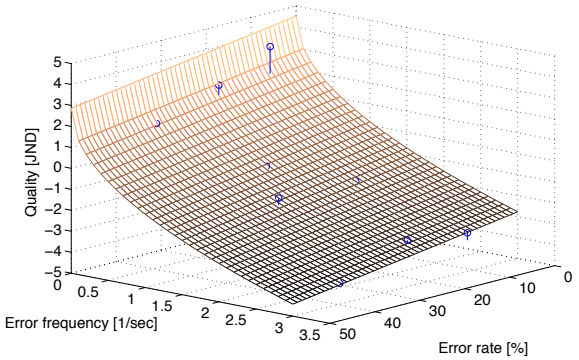

Figure 5: *Results of the quality experiment after JND-scaling are shown as black circles. The surface shows the best fitting function (SSE=0.9167). The lines are plotted between the measured JND-values and the fitted function.*

tion about objects in the scene, their positions and movement paths. In the following sections we derive a gaze-driven object tracking algorithm (GDOT) that combines the information from the eye-tracker and the 3D scene to improve the accuracy of eye-tracking. The goal is to simulate a dynamic and high quality DoF effect, but we also demonstrate the utility of this algorithm in other applications.

The overview of the tracking system is shown in Figure 6. The rendering engine supplies the gaze-tracking module with the current positions of the potential targets of attention and the eye-tracker sends the current gaze positions. Because the frequency of eye-tracker readings is higher than that of the display, gaze positions are resampled to match the display refresh rate, but also to reduce the noise in the eye-tracker readings and the inconsistencies in sampling. The targets are the spots that are likely to attract attention. Several targets are distributed over larger 3D objects and usually a single target is allocated to each smaller object. The task of the GDOT module is to select the most likely target of attention for the current time instance. This needs to be performed in an on-line system where only the information about past animation frames is available. The ID of the fixated target is passed back to the rendering engine, where it could be used for a desired application, such as DoF rendering.

The following subsections describe the details of the algorithm. The choice of the parameters is discussed in Section 7. For convenience, a MATLAB code of the off-line algorithm is included in the supplementary materials.

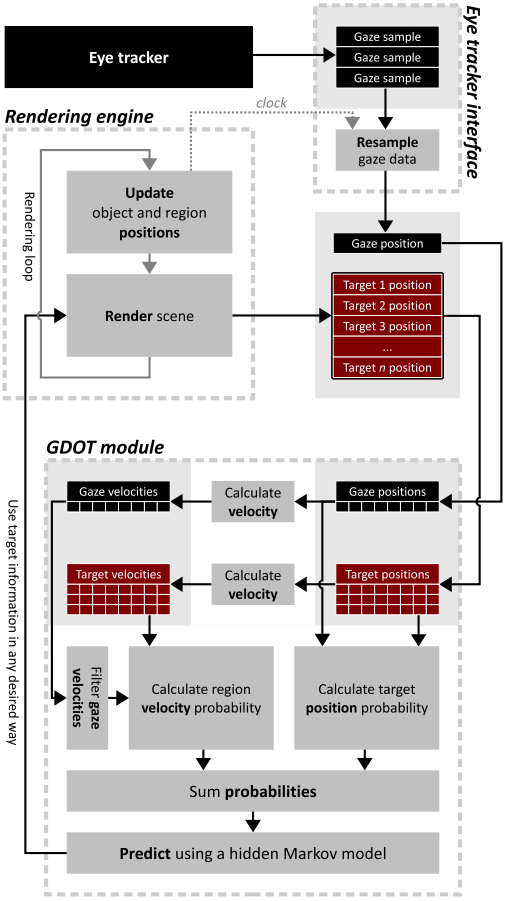

Figure 6: *The design of the gaze-tracking system, combining a 3D rendering engine with the gaze-directed target-tracking.*

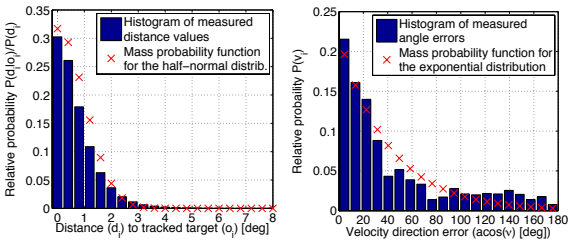

Figure 7: *Position (left) and velocity (right) error distribution for the tracked object.*

6.1. Gaze-driven object tracking

For a target to be attended, it must be close to the gaze point, and it should move with a similar velocity as the eye scan-path. Hence, the probability of fixating at the target i is:

$$P(o_i) = P(p_i \cup v_i) = 1 - (1 - P(p_i))\,(1 - P(v_i))\,, \qquad (2)$$

where $P(p_i)$ is the likelihood that the gaze point is directed at the target o_i (position is consistent) and $P(v_i)$ is the likelihood that the eye scan-path follows the target (velocity is consistent). The sum of probabilities ($\cup$) is used for the position and velocity consistency because it is likely that either position or velocity is inconsistent even if the target is at-

tended. The position inconsistency is often caused by imperfect eye-tracker calibration, so that the gaze-points always hit near but rarely at the target position. The velocity consistency is usually very low for stationary targets which have zero velocity while the eye-tracker readings indicate a constant movement of the eye. However, the consistency starts to be very high when the target moves and the smooth pursuit eye-motion is getting registered by the eye-tracker.

A close distance between the eye-tracker gaze point and the target is the strongest indicator that an object is attended. However, care must be taken to properly model the likelihood that a particular distance indicates that an object is attended. Because the distance specifies all points on the circle around the target point, the circumference of the circle will grow relative to the distance and consequentially the probability of observing such a distance, $P(d_i)$. This probability is due to the geometric properties of the distance and has no importance for the selection of the attended object. In discrete terms the histogram of observing a distance for the target needs to be normalised by the area of the annulus covering all pixels belonging to a particular bin of the histogram. Such a normalised histogram is shown in the left of Figure 7. Such normalisation is the consequence of the Bayesian rule:

$$P(p_i) = \frac{P(d_i|o_i)\,P(o_i)}{P(d_i)} = \omega_p \exp\left(\frac{-d_i^2}{2\sigma_s^2}\right), \qquad (3)$$

where d_i is the Euclidean distance between the gaze point and object o_i, expressed in the screen coordinates. $P(d_i|o_i)$ is the probability of observing distance d_i between the gaze point and the object when the object is tracked. Figure 7 indicated that such probability can be well approximated by the half-normal distribution, with the parameter σ_s describing the magnitude of the eye-tracker error. ω_p is the importance of the position consistency relative to velocity consistency (from 0 to 1).

If the position consistency becomes unreliable, the object can still be tracked if its velocity is consistent with the smooth pursuit motion of the eye. The velocity computed directly from scan-paths is an unreliable measure as it is dominated by the eye-tracker noise, saccades and the tremor of the eye. Fortunately, the smooth pursuit motion operates over longer time periods and thus can be extracted from the noisy data with the help of a low-pass filter. For simplicity, we employ a box-filter. The choice of the filter length is discussed in Section 7.

We found that the consistency of velocity $P(v_i)$ is the most robust if it is defined in terms of the angle between the low-pass filtered gaze-path velocity vector $\mathbf{u_t}$ and the target velocity vector $\mathbf{v_{t,i}}$, and is independent of the magnitude of these velocities. The correlate of such an angle is:

$$\nu = \frac{\mathbf{u_t} \circ \mathbf{v_{t,i}} + \varepsilon}{||\mathbf{u_t}||\,||\mathbf{v_{t,i}}|| + \varepsilon}, \qquad (4)$$

where $\circ$ is a dot product and ε is a small constant (0.001), which prevents division by 0 when either a target or a gaze point are stationary. Based on our experimental data, shown on the right of Figure 7, the arccos of this correlate follows exponential distribution. Hence, the likelihood of consistent velocity is expressed by:

$$P(v_i) = \frac{P(\nu_i|o_i)\,P(o_i)}{P(\nu_i)} = \omega_v \exp\left(\frac{-\arccos(\nu)}{\sigma_v}\right), \qquad (5)$$

where σ_v describes the allowable magnitude of the angular error. Analogous to ω_p, ω_v is the importance of velocity consistency relative to the position consistency.

A naive target tracking algorithm could compute the probability for each target at a given point in time (or frame) and choose the object with the highest probability. This, however, would result in frequent shifts of tracking from one target to another. We experimented with temporal low-pass filters that could reduce the fluctuations of the gaze point readings $\mathbf{g_t}$, gaze point velocities $\mathbf{u_t}$ and corresponding probabilities $P(p_i)$ and $P(v_i)$. Such an approach is similar to integration in the I-DT algorithm. However, we found that such filtering introduces an additional time delay (phase shift), which cannot be avoided if the gaze-tracking is needed for an on-line and real-time system.

An elegant way to penalise implausible interpretation of the data without excessive time delay is to model the attention shifts between targets as a Hidden Markov process. Each state in such a process corresponds to tracking a particular target. Since the fixation cannot be too short, the probability of staying in a particular state is usually much higher than the probability of moving to another state (in our case 95% vs. 5%, refer to Figure 8). This way the eye is more likely to keep tracking the same target than to rapidly jump from one target to another.

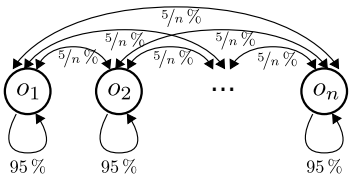

Figure 8: *State transitions for the Hidden-Markov-Model (HMM) of fixations. Each state o_1,...,o_n represents a single target. The probability of transition between targets is much lower than the probability of remaining in the same state (keeping focus on a target).*

The solution of the HMM gives the most likely sequence of states (tracked targets) at a given point in time. The strength of this approach is that the best sequence is selected irrespectively of the best sequence in the previous point in time, so that as the new data arrives, the decision to select a particular target in the previous instance of time can be revised in favour of a more likely sequence. Because of that,

it is advantageous to introduce a short delay between adding new probability values to the model and retrieving the resulting state. For the first order HMM, used in our solution, the best sequence can be efficiently computed using a dynamic programming algorithm called the Viterbi algorithm [Vit67].

6.2. Alternative tracking algorithms

Although the algorithm described in the previous section appears simple, it performed better than several more complex alternative algorithms that we tested against our data set. The alternatives included several temporal filters mentioned in the previous section, combination of these filters with the HMM, and the algorithm in which the gaze-points were classified into saccades and fixations using the HMM based I-VT algorithm. Only fixations were used to compute the gaze velocities. The calibration procedure reduced the temporal constants of all the filters to 0, indicating that the filters degraded resulting quality instead of improving it. The benefit of fixation/saccade segmentation was minimal as compared to the added complexity of the system. The Kalman filter is commonly used for the problems where a smooth trajectory needs to be estimated from noisy data [YLNP12]. However, the actual eye movements are erratic, with sudden saccadic movements occurring even when we consciously track a single object. Such movements would be very difficult to predict with the Kalman filter.

7. Evaluation and results

To fairly compare tracking and fixation algorithms, it is necessary to find the best set of parameters for each algorithm. To avoid over-fitting, we performed cross-validation with a random 50%/50% division of the experiment data (Section 4) into training and testing sets. The downhill simplex method with multiple starting points was used for a derivative-free optimisation of the parameters, which avoided local minima.

In addition to the proposed GDOT, we tested commonly used I-DT and I-VT algorithms, introduced in Section 3.2, as well as a simple tracking scheme, which used non-processed eye-tracker raw data to select the closest target. We are not aware of any other method that would use 3D scene information to improve the accuracy of eye tracking. To our best knowledge, I-DT and I-VT are the two algorithms most often used in commercial eye-trackers. But it must be also noted that our technique is not directly comparable to I-DT and I-VT as these are context-free methods while our method requires the knowledge of the scene.

The optimisation of our algorithm did not include the width of the temporal velocity filter and the delay of the HMM retrieval (refer to Section 6.1) because both values are discrete and thus not suitable for the continuous optimisation method we used. Instead, we confirmed that there is little correlation between these two and other parameters and then performed an exhaustive search in the 2D parameter space. The highest quality was found for the temporal velocity filter of 120 frames (2 seconds), and the HMM retrieval delay of 18 frames (300 ms at 60 Hz).

Table 1 reports the best fitting parameters for each method and scene, as well as the result of global fitting when one set of parameters is used for all scenes. For each algorithm there is at least one parameter that differs significantly from one scene to another. This confirms the difficulty of finding a single set of parameters that would be suitable for a wide range of scenes. However, we observed that GDOT is less affected by this variability (refer to Figure 10).

Figure 9 compares the quality, error rate and error frequency (refer to Section 5) for all tested algorithms. The proposed GDOT has a far superior overall quality as compared with other algorithms, mostly because of the consistency of predictions (low error frequency), but also because it could track longer attended objects (smallest overall error rate). This result has been also confirmed informally by running our gaze-contingent applications (see Section 8) using different algorithms, and in each case only GDOT offered a sufficient level of performance.

Table 1: *Optimised parameters for fixation techniques computed for individual scenes and globally for all sessions.*

	param.	A	B	C	global
I-VT	*velocity [deg/sec]*	3.22	5.90	4.76	4.02
I-DT	*duration [sec]*	175	181	200	181
	dispersion [pixels]	245	173	178	208
	win. length [sec]	45.3	18.5	11.4	20.1
GDOT	σ_s *[pixels]*	469	244	443	465
	σ_v	0.65	0.68	0.38	0.73
	ω_p	1	1	1	1
	ω_v	0.54	0.39	0.70	0.41

7.1. Time-characteristic of object tracking

The time-characteristics of each algorithm, shown in Figure 10, provide better insight into why the proposed algorithm was judged as significantly better. Each plot shows one session from the experiment, where the ID of the coloured objected that observers were asked to follow is shown as a bold red line. The blue line is the ID of the object identified by the corresponding tracking/fixation algorithm.

The raw gaze data (top plot) resulted in a relatively low error rate, lower than for more advanced I-VT and I-DT algorithms. However, raw gaze points also gave an unacceptable level of flickering, resulting in much higher error frequency.

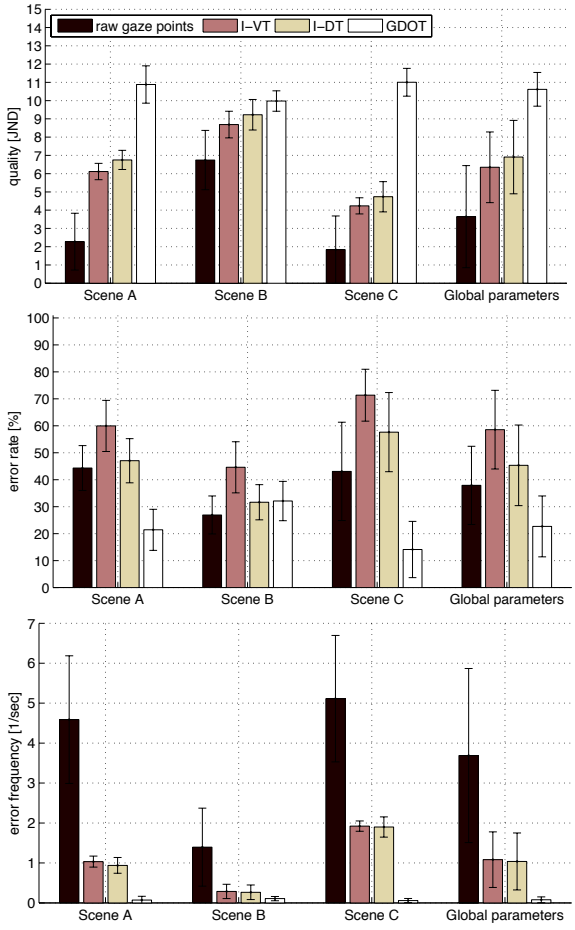

Figure 9: *Quality of tracking, error rate, and error frequency for individual scenes and for global parameters optimised for all scenes. The bars show the mean and standard deviation of 10 repetitions of 2-fold cross validation.*

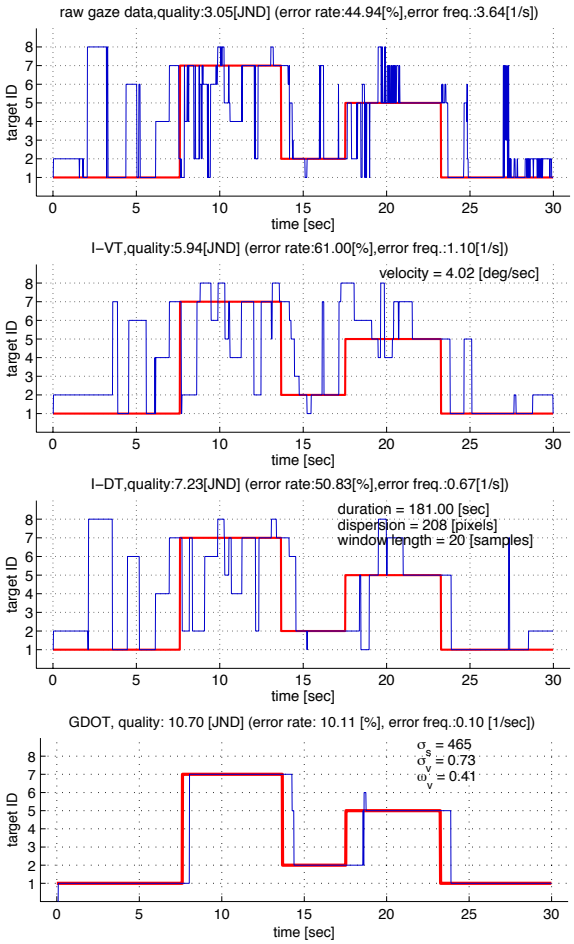

Figure 10: *Object tracking results for an example experimental session (scene A, observer mka-1). The red lines are the IDs of the reference targets, the blue lines denote the predicted target.*

This result motivated us to derive a JND-based quality metric in Section 5, which correctly reflects a much worse visual experience when using raw eye-tracker data.

For the majority of sessions the proposed GDOT correctly identified all targets and the dominant source of error was the delay when switching between targets. Part of the delay was caused by the fact that the observers in the experiment were not able to move their gaze instantaneously when the marker jumped from one object to another. Hence, no algorithm is able to give zero error for our data. However, the major part of the delay is due to the temporal velocity filter and the delay in the HMM state retrieval in the algorithm. The JND error measure strongly penalised wrong predictions as more disturbing than the delays, so that the optimisation resulted in a higher time-constants.

8. Applications

The proposed GDOT algorithm was tested in thee practical, graphics oriented applications: a gaze-dependent depth-of-field rendering, a gaze-contingent controller in a computer game, and an eye-tracker-supported saliency model. For all applications we associated tracked target points with moving objects, and used a regular grid of points for larger objects. We found this operation straightforward and requiring little effort.

8.1. Focal-cues induction

It has been shown that people prefer depth-of-field (DoF) visualisation actively controlled by the temporal gaze direction because it enhances immersion in the virtual environment [HLCC08, MBT11]. Such a gaze-contingent induc-

tion of focal-cues can be seen as a substitute for a multi-focal plane display [AWGB04], which poses a huge technical challenge to build. To achieve gaze-contingent DoF visualisation, we implemented the DoF rendering algorithm based on the reverse mapping technique with reduced colour leakage [PC83, MBT11]. A frame from such a DoF simulation is shown in Figure 4. As compared to other implementations of similar rendering, the GDOT algorithm significantly improves accuracy and stability of the intended focus plane identification, even for fast moving objects. We found that the existing solutions based on the I-DT result in frequent unwanted changes of focus, which are highly distracting for the observers.

8.2. Gaze-contingent controller in a computer game

Another appealing application of our tracking algorithm is using an eye-tracker as a game controller. Figure 11 shows a frame from a shooter-type game controlled with the help of our algorithm. A video clip is included in the supplementary video. During the gameplay, it is possible to display information about an opponent's spaceship just by looking at it. Also the camera starts to follow the attended target to create a more attractive way of aiming player's weapons. When experimenting first with the I-DT algorithm, it was found challenging to keep a constant focus on the fast moving objects. GDOT correctly identifies targets that are followed by the smooth pursuit motion of the eye. The proposed method is also able to identify the objects that are smaller than the radius of the eye tracker error.

Figure 11: *Screenshot from the* Invasion *game. The information about the spaceship is displayed only when a player is looking at it.*

8.3. Scene-aware saliency model

Saliency models predict the likelihood that a particular part of the scene will be observed or will remain unattended. The classical techniques predict saliency from low-level features such as luminance and colour contrast [Itt00]. However, these techniques can be unreliable because of the task-driven, top-down nature of the visual attention [TG80, THLB11].

Eye-tracking is believed to capture the attention patterns, which reflect both top-down and bottom-up attention processes. But in practice such data is very noisy, restricted to relatively large objects and regions, and is even less reliable when the scene is animated. Our algorithm can be used to determine the objects that are the most likely to be scrutinized and thus it predicts the saliency of objects rather than the saliency of pixels. This is an important distinction, because most applications of saliency require the knowledge of objects that are actually observed by the user, rather than the pixels which could be briefly scanned by the gaze. It must be noted, however, that our method will not detect the patters of the low-level attention mechanism that constantly scans a scene in a search for the best gaze allocation [THLB11].

9. Conclusions and future work

In this work we have shown how using information about a 3D scene and the motion of objects can greatly improve the accuracy of eye-tracking. The proposed probabilistic model aggregates the likelihood of attending each individual target across time using the prior knowledge about an eye-tracker error distribution and temporal fixation patterns of the eye. It is notable that the algorithm is motivated and then optimised by a JND-scaled quality metric, derived from experimental data. Such a metric was necessary as the quality of object-tracking cannot be explained by a simple objective measure. The proposed algorithm is shown to outperform standard fixation algorithms both in terms of objective measures as well as the JND-scaled quality metric.

As a future work, we will introduce our algorithm into novel applications, such as gaze-contingent simulation of afterimages [RE12]; locally adaptive tone-mapping [RFM*09]; a novel controller for gaming; and as a support input for other pointing devices, such as a mouse or a touchscreen. When used with stereo displays, the accuracy of tracking could be further improved by incorporating the information about the vergence of the eyes [DPHW11], so that the gaze position and velocity is registered in a 3D space instead of a display plane.

Acknowledgements

We would like to thank anonymous reviewers for their comments and suggestions and the volunteers who participated in the experiments. This work was partly supported by the Polish Ministry of Science and Higher Education through the grant no. N N516 508539.

References

[AMB10] AGUSTIN J., MOLLENBACH E., BARRET M.: Evaluation of a low-cost open-source gaze tracker. In *Proc. of ETRA 2010, Austin, TX, March 22-24* (2010), pp. 77–80. 1

[AWGB04] AKELEY K., WATT S., GIRSHICK A., BANKS M.: A stereo display prototype with multiple focal distances. *ACM Transactions on Graphics (TOG) 23*, 3 (2004), 804–813. 1, 10

[Bli09] BLIGNAUT P.: Fixation identification: The optimum threshold for a dispersion algorithm. *Attention, Perception, & Psychophysics 71* (2009), 881–895. 3

[DPHW11] DUCHOWSKI A. T., PELFREY B., HOUSE D. H., WANG R.: Measuring gaze depth with an eye tracker during stereoscopic display. In *Proceedings of Symposium on Applied Perception in Graphics and Visualization* (France, 2011), APGV'11. 10

[Duc07] DUCHOWSKI A. T.: *Eye Tracking Methodology: Theory and Practice (2nd edition)*. Springer, 2007. 2, 3

[Eng00] ENGELDRUM P. G.: *Psychometric scaling: a toolkit for imaging systems development*. Imcotek Press, 2000. 5

[EV95] ERKELENS C. J., VOGELS I. M. L. C.: The initial direction and landing position of saccades. *Eye Movements Research: Mechanisms, Processes and Applications* (1995), 133–144. 3

[HLCC08] HILLAIRE S., LECUYER A., COZOT R., CASIEZ G.: Using an eye-tracking system to improve camera motions and depth-of-field blur effects in virtual environments. In *Proc. of IEEE Virtual Reality* (2008), pp. 47–50. 1, 3, 9

[HNA*11] HOLMQVIST K., NYSTROM M., ANDERSSON R., DEWHURST R., JARODZKA H., VAN DE WEIJER J.: *Eye Tracking: A comprehensive guide to methods and measures*. Oxford University Press, USA; 1 edition (November 1, 2011), 2011. 1, 5

[Itt00] ITTI L.: *Models of Bottom-Up and Top-Down Visual Attention*. PhD thesis, California Institute of Technology, Pasadena, California, Jan 2000. 3, 10

[Jac93] JACOB R. J. K.: Eye movement-based human-computer interaction techniques: Toward non-command interfaces. *Advances in Human-Computer Interaction* (1993). 3

[Kar00] KARN K. S.: Saccade pickers vs. fixation pickers: the effect of eye tracking instrumentation on research. In *Proceedings of the 2000 symposium on Eye tracking research & applications* (New York, NY, USA, 2000), ETRA '00, ACM, pp. 87–88. 3

[LH01] LUEBKE D. P., HALLEN B.: Perceptually-driven simplification for interactive rendering. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques* (London, UK, UK, 2001), Springer-Verlag, pp. 223–234. 3

[LW90] LEVOY M., WHITAKER R.: Gaze-directed volume rendering. In *Proceedings of the 1990 symposium on Interactive 3D graphics* (New York, NY, USA, 1990), I3D '90, ACM, pp. 217–223. 3

[MBT11] MANTIUK R., BAZYLUK B., TOMASZEWSKA A.: Gaze-dependent depth-of-field effect rendering in virtual environments. *Lecture Notes in Computer Science (Proc. of SGDA 2011) 6944* (2011), 1–12. 9, 10

[MD01] MURPHY H., DUCHOWSKI A. T.: Gaze-contingent level of detail rendering. In *Proceedings of the EuroGraphics Conference* (2001), EuroGraphics Associates. 3

[MDT09] MURPHY H. A., DUCHOWSKI A. T., TYRRELL R. A.: Hybrid image/model-based gaze-contingent rendering. *ACM Trans. Appl. Percept. 5* (February 2009), 22:1–22:21. 3

[MJ12] MANTIUK R., JANUS S.: Gaze-dependent ambient occlusion. *Lecture Notes in Computer Science (Proc. of ISVC'12 Conference) 7431*, I (2012), 523–532. 3

[MKNB12] MANTIUK R., KOWALIK M., NOWOSIELSKI A., BAZYLUK B.: Do-it-yourself eye tracker: Low-cost pupil-based eye tracker for computer graphics applications. *Lecture Notes in Computer Science (Proc. of MMM 2012) 7131* (2012), 115–125. 1

[MM05] MORIMOTO C. H., MIMICA M.: Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding 98*, 1 (2005), 4–24. 2, 3

[PC83] POTMESIL M., CHAKRAVARTY I.: Modeling motion blur in computer-generated images. *SIGGRAPH Comput. Graph. 17*, 3 (July 1983), 389–399. 10

[RE12] RITSCHEL T., EISEMANN E.: A computational model of afterimages. *Computer Graphics Forum 31*, 2pt3 (2012), 529–534. 10

[RFM*09] RAHARDJA S., FARBIZ F., MANDERS C., ZHIYONG H., LING J. N. S., KHAN I. R., PING O. E., PENG S.: Eye hdr: gaze-adaptive system for displaying high-dynamic-range images. In *ACM SIGGRAPH ASIA 2009 Art Gallery & Emerging Technologies: Adaptation* (New York, NY, USA, 2009), SIGGRAPH ASIA '09, ACM, pp. 68–68. 3, 10

[RMB94] ROSS J., MORRONE M., BURR D. C.: Changes in visual perception at the time of saccades. *TRENDS in Neurosciences 24*, 2 (1994), 113–121. 2

[SA98] SALVUCCI D. D., ANDERSON J. R.: Tracing eye movement protocols with cognitive process models. In *In Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (1998), Erlbaum, pp. 923–928. 3

[SEDS81] SALTHOUSE T. A., ELLIS C. L., DIENER C. L., SOMBERG B. L.: Stimulus processing during eye fixations. *Journal of Experimental Psychology: Human Perception and Performance 7*, 3 (1981), 611–623. 2

[SF01] SILVERSTEIN D., FARRELL J.: Efficient method for paired comparison. *Journal of Electronic Imaging 10* (2001), 394. 5

[SG00] SALVUCCI D. D., GOLDBERG J. H.: Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications (ETRA)* (New York, 2000), pp. 71–78. 3

[SMI09] SMI: *RED250 Technical Specification*, 2009. SensoMotoric Instruments GmbH. 4

[SSC08] SHIC F., SCASSELLATI B., CHAWARSKA K.: The incomplete fixation measure. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications* (2008), ACM, pp. 111–114. 3

[Sun12] SUNDSTEDT V.: *Gazing at Games: An Introduction to Eye Tracking*. Morgan & Claypool Publishers, 2012. 2, 3

[TG80] TREISMAN A., GELADE G.: A feature-integration theory of attention. *Cognitive Psychology 12* (1980), 97–136. 10

[THLB11] TATLER B. W., HAYHOE M. M., LAND M. F., BALLARD D. H.: Eye guidance in natural vision: reinterpreting salience. *Journal of vision 11*, 5 (May 2011). 10

[TOB11] *Accuracy and precision test method for remote eye trackers (version 2.1.1)*. Tech. rep., Tobii Technology, 2011. 5

[ULIS07] URRUTY T., LEW S., IHADADDENE N., SIMOVICI D. A.: Detecting eye fixations by projection clustering. *ACM Trans. Multimedia Comput. Commun. Appl. 3* (December 2007), 5:1–5:20. 3

[Vit67] VITERBI A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. on Information Theory 13*, 2 (1967), 260–269. 8

[Wid84] WIDDEL H.: Operational problems in analysing eye movements. *In A. G. Gale & F. Johnson (Eds.), Theoretical and Applied Aspects of Eye Movement Research. North-Holland: Elsevier Science Publishers B.V. 1* (1984), 21–29. 3

[YLNP12] YEO S. H., LESMANA M., NEOG D. R., PAI D. K.: Eyecatch: simulating visuomotor coordination for object interception. *ACM Trans. Graph. 31*, 4 (July 2012), 42:1–42:10. 8