Real-time Multi-map Saliency-driven Gaze Behavior for Non-conversational Characters – Supplementary Material –

Ific Goudé*, Alexandre Bruckert*, Anne-Hélène Olivier, Julien Pettré, Rémi Cozot, Kadi Bouatouch, Marc Christie, and Ludovic Hoyet

In this supplementary material, we provide additional information regarding the framework to facilitate replication. We also give more details regarding the VR-based user study conducted to collect ground-truth gaze activity for evaluating our model.

1 Saliency-driven gaze animation model

In this section, we provide additional technical details for the reproduction of our model. The useful notations and variables are presented in Table 1. As a reminder, our model relies on a *salicency model* in charge of assessing which elements in the character's field of view are more likely to attract its visual attention, a *saccadic model* that reproduces natural oculomotor characteristics (*e.g.*, duration of eye fixations, orientations and amplitudes of eye saccades, etc.), and a *eye-head animation model* that controls the eyes and head motion, while ensuring the realism of the kinematic variables. A full C++ implementation for Unreal Engine 5 is also made available online for the community[1].

Notations	
I_r	Image rendering of the scene: RGB 2D texture.
S_y	Saliency map: single-channel 2D texture.
S_a	Saccadic map: single-channel 2D texture.
F_{map}	Fixation map: single-channel 2D texture.
F_{point}	Fixation point: 4D vector representing the position and the duration of a fixation.
Animation Variables	
H_{angle}	The angle in degree between the head forward vector and the vector from the centre of the character's viewpoint to the fixation point.
T_h	Time delay between a new fixation point and the head movement with $T_h \sim \mathcal{N}(0.15, 0.1)$.
X_h	A threshold angle that triggers the head acceleration with $X_h \sim \mathcal{N}(60, 10) \leq 60$.
Y_h	A threshold angle that triggers the head deceleration with $Y_h \sim \mathcal{N}(15, 2)$.
R_{blink}	The probability of blinking at each frame.
T_{blink}	Time since last blink in seconds.

TABLE 1: Notations and variables used in our method.

1.1 Visual saliency model

The role of the *visual saliency model* is to assess which areas of the character's field of view are more likely to attract attention, solely based on the content of the image, and with no considerations of the human eye kinematics, or memory (*i.e.* bottom-up approach).

Given an image of the scene I_r rendered from the character's point of view, the *visual saliency model* predicts a saliency value, describing the visual attractiveness, for each pixel of the image. The output of the model is a scalar field of the size of the input image referred to as the **saliency map** S_y.

We use the deep saliency model MSINet [1], given its excellent performances (see for instance the MIT-Tuebingen benchmark of saliency models [2]), as well as its low number of parameters, enabling the model to be very efficient in terms of inference time. The choice of a deep learning model is straightforward considering the huge gap in performances between these data-oriented methods and more traditional image processing techniques [3]. The model is trained on the 10000 training images of the SALICON dataset [4], and later fine-tuned on the MIT dataset [5]. Both are composed of natural images, with their associated ground-truth saliency representations inferred from eye-tracking or mouse-tracking experiments. The weights are optimized using a combination of loss functions (CC, NSS and KL-divergence, see [6]) that compares the saliency maps of the prediction with the ground-truth. As an example, the **saliency map** predicted by our *visual saliency model* is illustrated in Figure 1b.

- * *both authors have contributed equally to the submission*
- *Ific Goudé, Anne-Hélène Olivier, Julien Pettré, Kadi Bouatouch, Marc Christie and Ludovic Hoyet are with Inria, Univ Rennes, CNRS, IRISA. E-mail: {name.surname}@inria.fr*
- *Alexandre Bruckert is with Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France. E-mail: alexandre.bruckert@univ-nantes.fr*
- *Rémi Cozot is with Littoral Opal Coast University*

1. https://github.com/igoude/saliency-driven-gaze

(a) Scene rendering I_r. (b) Saliency map S_y. (c) Saccadic map S_a. (d) Inhibition map IoR. (e) Fixation map F_{map}.

Fig. 1: The character's viewpoint overlaid with heatmaps where pixel values (from 0: blue to 1: red) correspond to the probability of shifting the gaze in the direction of the pixel. Here, F_{map} results from the blending of S_y, S_a, and IoR with coefficient weights equal to 0.9, 0.85, and 0.75 respectively.

1.2 Saccadic model

Based on eye-tracking data on the one hand, and the **saliency map** generated at the previous step on the other, the objective is here to determine the next gaze fixation point for the virtual character. Our *saccadic model* is designed to reproduce the oculomotor biases exhibited by humans when visually exploring scenes (*e.g.*, duration of fixations, orientations and amplitudes of saccades). Our approach consists in generating a **saccadic map** that represents the likelihood of saccade orientations and amplitudes. Additionally, in order to include memory and temporal information, the Inhibition of Return (IoR) to observing objects in the scene is simulated thanks to an **inhibition map**. Those maps are combined with the **saliency map** to compute a **fixation map**, in turn used to determine the position of the next fixation point for the character. We will see in Section 1.2.4 that it is possible to influence the likelihood of the position of this fixation by other bias sources using **customization maps**, such as attraction to a focus point in the scene (*e.g.*, based on semantics or events), or favoring an idle gaze behavior.

1.2.1 Saccadic map (S_a)

A **saccadic map** captures the joint probability distributions of saccades orientations and amplitudes. As the human oculomotor bias depends on the task performed and the type of scene in which the task takes place, there should be as many saccadic maps as there are character categories and activities. However, exploring the effect of various saccadic maps is outside the scope of this work as we are targeting the animation of characters' gaze during free visual exploration tasks (generalization to other tasks is however discussed in Section 6).

A **saccadic map** is thus built in several steps. First, a dedicated eye-tracking dataset is recorded. The visual task performed by participants, as well as the type of scene in which this task takes place should match what the character is expected to do and its environment. Note that a number of datasets exist and can be employed here, *e.g.*, from visual exploration of natural scenes [4], [5]. Saccade parameters, *i.e.* orientation and amplitude, are then extracted from this dataset. As orientation and amplitude are not independent, they are aggregated together resulting in a joint distribution of probability. The **saccadic map** S_a is then the result of a Gaussian filter applied to the joint probability of saccades amplitude and orientation, and corresponds to the probability of producing a saccade in each point of the vision field relative to the vision center, as illustrated in Figure 1c.

1.2.2 Fixation duration

To provide realistic saccade frequencies, the fixations durations are also extracted from the eye-tracking dataset to establish the **distribution probability of fixation durations**. We found that a shifted gamma law $\Gamma(\alpha, \theta, loc)$ accurately captured this distribution in our situations ($\alpha = 1.2394$, $\theta = 0.1880$, and $loc = 0.08$), and therefore use these parameter values in the remaining of the paper.

1.2.3 Inhibition map (IoR)

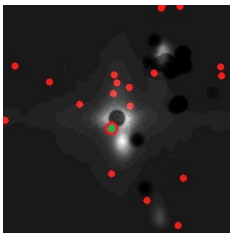
As the *saliency model* does not include any temporal information, there is a tendency for predicted fixations to repeatedly go back to the same salient elements of the scene. However, humans tend to explore their "visual environment by reducing the tendency to repeatedly sample a particular location" [7], which we model using an **inhibition map** IoR. This map is in charge of forcing the exploration by decreasing the probability of looking at recently observed elements, and is built by rendering a Gaussian of the size of the foveal vision (2° of angle) at the position of the previous fixations [8]. An inhibition point linearly decreases with time and completely disappears after 20 times the duration of the fixation has passed or after 15 new fixations. Note that a low value is applied to inhibition points, contrarily to other maps, as illustrated in Figure 1d.

1.2.4 Customization maps

As discussed previously, our model enables the user to provide additional **customization maps** which contribute to the final **fixation map**. The use of these maps can be extremely valuable to synthesize specific gaze behaviors, *e.g.*, to account for specific tasks. We present examples of two specific **customization maps** that we use for synthesizing our results in the main paper.

1.2.5 Maps Blending

The **fixation map** F_{map} represents the probability of looking in a direction at the end of a fixation, and results from the combination of the **saliency**, the **saccadic**, the **inhibition**, and the **customization maps**. F_{map} is the product of all the



(a) Random points in F_{map}.

(b) Raycasting process.

Fig. 2: Fixation computation process: (a) 20 random points (red dots) are distributed according to the probability distribution of F_{map}, the highest value (green dot) is selected. (b) Raycasting (red arrow) is used to hit the position (green cross) of the next fixation point F_{point} in the scene.

maps weighted by coefficients that adjust their respective contributions.

For each pixel (x, y) of a map M, which value $M(x, y)$ is defined between 0 and 1, and a weighting coefficient $w \in [0, 1]$, the weighted map $\bar{M}$ is equal to:

$$\bar{M}(x, y) = (w \times M(x, y)) + (1 - w), \quad (1)$$

where a weight w equal to 1 preserves the values of the map, while a weight w equal to 0 should not influence the product of probabilities, and then sets all the values of the map to 1 (*i.e.* right part of Equation 1).

Finally, the combination of all maps is given by:

$$F_{map}(x, y) = \prod_{i=0}^{N} \bar{M}_i(x, y), \quad (2)$$

where N is the number of weighted maps $\bar{M}_i$, and F_{map} is the resulting **fixation map**. Figure 1 presents the principal maps described before, while the method is designed so that users can easily define their own **customization maps**.

1.2.6 Fixation computation

F_{map} is then used to compute the position of the next fixation in the character's field of view. To this end, P pixels are randomly selected, according to the probability distribution of F_{map}, and the one with the highest value (*i.e.*, highest probability of fixation) is then selected to be the next fixation point. In our results, P is set to 20, as illustrated in Figure 2.

The 3D position (x, y, z) of the fixation in the virtual scene is then identified by casting a ray originating from the optical center of the camera rendering the scene (*i.e.*, the character's viewpoint) and directed to the selected fixation pixel (Figure 2b). Defining the fixation point in the 3D world space, instead of keeping it in the 2D view, resolves issues regarding object pursuit and movement coherency, as discussed by Peters et al. [9]. Finally, we set a fixation duration d from the probability distribution function of the found shifted gamma law presented in Section 1.2.2. The eyes and head of the autonomous character are finally animated to make it focus on the **fixation point** $F_{point}(x, y, z, d)$, as detailed below.

Algorithm 1 Eye-head animation algorithm

```
if new F_point then
    Start moves eye
    Wait T_h seconds
    if H_angle > X_h then
        Start moves head
    end if
end if
if H_angle < Y_h then
    Stop moves head
end if
```

1.3 Eye-head animation model

Our *eye-head animation model* controls the eye and head movements between the previous and the next fixation points in a realistic way. Given the 3D position and the duration of the next fixation point $F_{point}(x, y, z, d)$, as well as the position and orientation of the character's eyes and head, it guarantees that the angles formed by the body, head and eyes are realistically distributed and that the kinematic variables follow realistic courses. The model also controls the eyelid animation so as to generate realistic eye blinks.

First, the animation of the eyes and head is controlled by a set of kinematic variables (see Table 1) derived from the study of Siden-mark and Gellersen [10] who measured eye, head and torso coordination during gaze shifts in Virtual Reality. Eyes and head movements are controlled according to the method summarized in Algorithm 1. Once a new fixation is computed, the *animation model* first initiates the movement of the eyes towards the fixation point, which is followed after T_h seconds by a coordinated movement of the head if H_{angle} is greater than a threshold X_h. The model therefore updates the orientation of the eyes and head at each frame, until a new fixation point is computed. Each eye moves at a constant angular velocity of 100 °/s in the direction of the fixation point. The head moves with a linear angular acceleration of 30 °/s^2, and at a maximum angular velocity of 40 °/s in the direction of the fixation point, until H_{angle} reaches a threshold Y_h that triggers the deceleration to zero speed. In addition, the motion range of the eyes is limited to 50° in any direction while the motion range of the head is limited to 80° horizontally and 60° vertically. To account for eye contribution to be larger in downwards direction than upwards, we oriented the forward vector of the character's head 10° downwards. Additionally, a subsequent part of the gaze realism is ensured by blinks. Our blink model is largely inspired by previous work [11], [12], [13], and follows the method described in Algorithm 2. It is also important to mention that the remaining of the body animation is not affected by our method. In our examples and experiments, body animations were therefore controlled by Unreal Engine using animations from Mixamo.

Finally, after the specified fixation duration, the entire gaze simulation process starts again, beginning with the rendering of the maps affected by the character's viewpoint as explained next.

Algorithm 2 Eye blink animation algorithm

```
if new F_point then
    R_blink = 0.05
else
    if T_blink < 2 then
        R_blink = T_blink/2 × 0.0009 + 0.0001
    else if T_blink < 3 then
        R_blink = (T_blink − 2) × 0.009 + 0.001
    else
        R_blink = 0.01
    end if
end if
if R_blink > R then                          ▷ with R = [0, 1]
    Blink during S ms                        ▷ with S = [100, 400]
end if
```

1.4 Maps rendering

Given the 3D position of the character's head and the direction of its gaze, the *maps rendering* process initiates our gaze simulation by providing to the *saliency model* the rendered image of the scene from the character's point of view. The *maps rendering* also provides the spatial information necessary for the computation of the **inhibition** and the **customization maps**.

When a new fixation is required, a virtual camera positioned at the character's head location and oriented toward the character's eye direction renders the scene. The camera field of view is set to 70° of angle to cover most of recorded saccades amplitude. The format of the image given to the *saliency model* must fit with the input of the MSINet which is an 8bit 256×256 RGB image. As it has been trained on natural images, the quality of the scene rendering directly impacts the saliency prediction of the network. This point is also discussed in Section 6 of the main paper.

2 Eye-gaze data collection

For objectively evaluating our method in terms of similarity between simulated and real gaze behavior, we conducted a user study to collect ground-truth eye-gaze activity by immersing participants in virtual scenes using a Head Mounted Display embedded with an eye-tracker. This section provides additional details about the user study.

2.1 Materials & Methods

2.1.1 Apparatus

In this study, participants were immersed in a virtual environment using an HTC Vive Pro Eye (specifications: 90 Hz, 110° *fov*, 2880 × 1600 resolution). It is equipped with a built-in eye-tracker (120 Hz, 110° *fov*, 0.5°–1.1° accuracy), which can function as an assessment tool of saccadic eye movement [14]. The experiment was designed with Unreal Engine 5, using MetaHuman characters to populate the scenes when necessary, as well as the SRanipal SDK to capture participants' eye-tracking data. The experiment ran on a HP Z VR G2 backpack (specifications: NVidia RTX 2080, Intel Core i7-8850H processor, 32GB RAM).

2.1.2 Protocol

Upon arrival, participants were asked to fill in a consent form, during which they were presented the task to perform. They were then invited to sit on a chair and to wear the HTC Vive Pro Eye HMD. Through the experiment, participants were immersed in the virtual scenes presented in Figure 2 in the main paper (i.e. the *lobby*, the *waiting room*, the *bar*, and the *poker table*).

We first performed the eye-tracking calibration, after which participants were immersed in the *lobby*, which acted as an introduction scene, so that they could familiarize themselves with the virtual environment and VR. In this scene, we checked the eye-tracking accuracy by asking participants to focus on colored cubes positioned in front of them, which turned black after a short fixation time (500 ms). In case participants were not able to focus on all the squares, we performed a calibration again until obtaining the required eye-tracking accuracy. The other scenes (*waiting room*, *bar*, and *poker table*) were used for collecting ground-truth data. For each of the three scenes, we also included combinations of the two following situations:

1) *Populated*: the scene either included virtual characters or not. Our objective was to account for differences in gaze behaviors in the presence of other characters, as faces are known to have a high level of attraction.
2) *Event*: the scene either included a particular event or not. The event was specific to each scene: the door of the *waiting room* suddenly opening, a red exit light suddenly flashing in the *bar*, chips falling from the slot machine in the *poker room*. Our objective was to account for sudden gaze behaviors when an event occurs, and to demonstrate the adaptability of our model in such situations.

Participants therefore performed 12 trials (3 *Scenes* × 2 *Populated* × 2 *Events*) in random order, which they were instructed to freely explored for 30 seconds each. Between trials, participants went back into the *lobby* to verify the eye-tracking calibration.

Participants also completed a number of questionnaires. Prior to the experiment, they provided demographic information (age and gender), as well as their experience in video games (from 0: never played, to 5: regular player) and VR (from 0: never tried, to 5: regular user). To avoid any effect of motion sickness on our recorded data (even though the risks were very limited), we asked participants to fill in a Simulator Sickness Questionnaire (SSQ) [15] before and after the experiment. Potential motion sickness was also assessed twice through the experiment by asking them a verbal rating (FMS) [16] of their physical state (from 0: no symptoms, to 20: frankly sick). Finally, we measured Presence using the Slater-Usoh-Steed (SUS) questionnaire [17] which was completed at the end of the experiment.

2.1.3 Participants

Fifty unpaid participants, recruited via internal mailing lists among students and staff, volunteered for the experiment (25F, 25M; age: avg=28±6, min=20, max=55). They were all naive to the purpose of the experiment, had normal or corrected-to-normal vision, and gave written and informed

(a) Scene 1: The waiting room. (b) Scene 2: The bar. (c) Scene 3: The poker table.

Fig. 3: Examples of our saliency maps on static scenes.

consent. The study conformed to the declaration of Helsinki, and was approved by the local ethical committee (COERLE).

2.2 Collected Data

2.2.1 Eye-tracking data

During the experiment, gaze samples were recorded at 90 Hz. For each frame, we collected: the participant's head (HMD) position and rotation in the virtual space, the gaze direction vector (i.e., the combined direction of head and eye in the virtual space) and the eye openness (from 0: closed to 1: opened). We used raycasting along the gaze direction vector to record the exact 3D position of the point gazed at in the scene, as well as the unique ID of the corresponding object. Over all participants, we recorded a total of 1,469,643 gaze samples, which were then processed according to the following two steps.

First, we discarded gaze samples where tracking errors were raised by the SDK, usually occurring right after a blink (27,196 samples, ∼1.85%). We also discarded samples corresponding to blinks, which we defined as gaze samples where eye openness was below a threshold of 0.2 (31,673 points, ∼2.15%). Before the experiments, we also defined that the whole recordings of a specific scene for an observer would be discarded if more than 10% of the gaze samples were flagged as untracked or blink, which would be representative of eye-tracking issues for this specific scene. Two scene records over the 600 available were thus also discarded (7,264 points, ∼0.49%).

Then, we computed eye fixations from the remaining raw gaze samples, using a velocity-based algorithm [18]. First, the gaze vector coordinates are transformed into latitude and longitude on a unit sphere centered on the head position. Then, we compute the velocity between two successive gaze points by dividing the haversine distance $\Delta\sigma$ by the timestamp difference between the two recordings using Equation 3:

$$\Delta\sigma = 2\arcsin\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)} \tag{3}$$

where $\Delta\sigma$ is the distance between the two points on the sphere, $\Delta\phi$ is the difference in longitude, $\Delta\lambda$ the difference in latitude and ϕ_1 and ϕ_2 are the respective longitudes of the two points. The resulting velocities are then smoothed using a Savitzky-Golay filter of order 2 with a window length of 3.

All gaze samples with a velocity under 80 °/sec were flagged as being part of a fixation. The last step was then to aggregate the successive gaze samples flagged as fixations into a single fixation point, which gaze direction vector is set as the average of all the direction vectors of the gaze samples within that fixation. Finally, we removed fixations that lasted less than 80 ms, as it is considered to be the minimum amount of time required to process visual information and plan a new saccade [19]. We also removed saccades lasting more than 1.4 s, i.e. over 4 SDs away from the mean fixation duration. A total of 35,984 fixations over the whole dataset were thus extracted.

2.2.2 Scene saliency map

To perform an objective evaluation of the model, we need to rely on ground-truth visual saliency maps created from the collected eye-tracking data. Since the participants were seated, and mostly static, we could rely on the 360° saliency maps described for instance by David et al. [20]. However, this approach requires considering the scene as an omnidirectional image, and does not account for multiple points of view. Thus, we propose a more general way of defining visual saliency for a virtual scene that is independent of the viewpoint.

For each fixation, we trace a ray from the average head position recorded during the fixation and following the gaze direction vector to get the coordinates in the scene of the fixation point. A 3D isotropic Gaussian kernel is then applied to those points, with sigma (standard deviation) sets to 2° of visual angle to account to the whole foveal area.

We finally define the saliency value at any point of the scene $S(x, y, z)$ as the sum of all the Gaussian kernels at this location:

$$S(x,y,z) = \sum_{i\in\mathcal{F}} \exp\left(\frac{-(x-x_i)^2 - (y-y_i)^2 - (z-z_i)^2}{2\sigma_i^2}\right) \tag{4}$$

where $\mathcal{F}$ is the set of all fixation points, (x_i, y_i, z_i) are the 3D-coordinates of the i-th fixation point, and σ_i its associated sigma computed to represent 2° of visual angle.

A saliency map is then defined by taking sample locations in the virtual scene and evaluating the saliency value at these points. The question of which fixations to consider strongly depends on the considered scene, its dynamics and the evaluation method. For instance, to evaluate the global attention in a static scene, we consider all fixations to get a single saliency map, while for a dynamic scene, it would be necessary to define a sliding time window, and only consider fixations within this time frame.

To get sampling points of the scene, since participants are static in our case, we sample a first point by tracing a ray following the vertical $(1, 0, 0)$ vector starting from the average position of the participants heads, and iterate each half degree in latitude and longitude. Examples of such scene saliency maps, represented as heat-maps [21], are displayed in Figure 3.

2.3 Questionnaire Analysis

To quantify a potential influence of scene realism on our eye-tracking dataset, we computed average SSQ (Simulator Sickness Questionnaire) and Presence scores. Regarding the effect of motion sickness, we performed a one-way ANOVA on the SSQ scores before and after the experiment. The results suggests that our experiment did not have any significant impact on motion sickness ($p > 0.05$), which scores were very low on average (0.35 ± 0.64 on a 0 to 6 scale). Similarly, the maximum of both FMS scores was also very low on average (1.3 ± 2.2 on a 0 to 20 scale). Overall, these results suggest that our experiment did not cause any significant discomfort to participants. Presence scores collected at the end of the experiment were above average (4.7 ± 1.7 on a 1 to 7 scale), and were in the range of what is commonly found in VR experiments.

REFERENCES

[1] A. Kroner, M. Senden, K. Driessens, and R. Goebel, "Contextual encoder–decoder network for visual saliency prediction," *Neural Networks*, vol. 129, pp. 261–270, 2020.

[2] M. Kümmerer, T. S. A. Wallis, and M. Bethge, "Saliency benchmarking made easy: Separating models, maps and metrics," in *Computer Vision – ECCV 2018*, 2018, pp. 798–814.

[3] A. Borji, "Saliency prediction in the deep learning era: Successes and limitations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 679–700, 2021.

[4] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1072–1080.

[5] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[6] A. Bruckert, H. R. Tavakoli, Z. Liu, M. Christie, and O. Le Meur, "Deep saliency models: The quest for the loss function," *Neurocomputing*, vol. 453, pp. 693–704, 2021.

[7] A. G. Samuel and D. Kat, "Inhibition of return: A graphical meta-analysis of its time course and an empirical test of its temporal and spatial properties," *Psychonomic bulletin & review*, vol. 10, no. 4, pp. 897–906, 2003.

[8] O. Le Meur and Z. Liu, "Saccadic model of eye movements for free-viewing condition," *Vision Research*, vol. 116, pp. 152–164, 2015.

[9] C. Peters, G. Castellano, M. Rehm, E. André, A. Raouzaiou, K. Rapantzikos, K. Karpouzis, G. Volpe, A. Camurri, and A. Vasalou, "Fundamentals of agent perception and attention modelling," in *Emotion-Oriented Systems*, 2011, pp. 293–319.

[10] L. Sidenmark and H. Gellersen, "Eye, head and torso coordination during gaze shifts in virtual reality," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 27, no. 1, pp. 1–40, 2019.

[11] L. Itti, N. Dhavale, and F. Pighin, "Realistic avatar eye and head animation using a neurobiological model of visual attention," in *Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation VI*, vol. 5200, 2003, pp. 64–78.

[12] C. Peters and A. Qureshi, "A head movement propensity model for animating gaze shifts and blinks of virtual characters," *Computers & Graphics*, vol. 34, no. 6, pp. 677–687, 2010.

[13] W. Steptoe, O. Oyekoya, and A. Steed, "Eyelid kinematics for virtual characters," *Computer animation and virtual worlds*, vol. 21, no. 3-4, pp. 161–171, 2010.

[14] Y. Imaoka, A. Flury, and E. D. de Bruin, "Assessing saccadic eye movements with head-mounted display virtual reality technology," *Frontiers in Psychiatry*, vol. 11, p. 922, 2020.

[15] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness," *The international journal of aviation psychology*, vol. 3, no. 3, pp. 203–220, 1993.

[16] B. Keshavarz and H. Hecht, "Validating an efficient method to quantify motion sickness," *Human factors*, vol. 53, no. 4, pp. 415–426, 2011.

[17] M. Usoh, E. Catena, S. Arman, and M. Slater, "Using presence questionnaires in reality," *Presence*, vol. 9, no. 5, pp. 497–503, 2000.

[18] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, 2000, p. 71–78.

[19] B. R. Manor and E. Gordon, "Defining the temporal threshold for ocular fixation in free-viewing visuocognitive tasks," *Journal of Neuroscience Methods*, vol. 128, no. 1, pp. 85–93, 2003.

[20] E. J. David, J. Gutiérrez, A. Coutrot, M. P. Da Silva, and P. L. Callet, "A dataset of head and eye movements for 360° videos," in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, p. 432–437.

[21] S. Stellmach, L. Nacke, and R. Dachselt, "Advanced gaze visualizations for three-dimensional virtual environments," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, 2010, p. 109–112.