COMPUTER GRAPHICS
forum

DOI: 10.1111/j.1467-8659.2010.01651.x

COMPUTER GRAPHICS forum
Volume 29 (2010), number 6 pp. 1830–1841

Using a Visual Attention Model to Improve Gaze Tracking Systems in Interactive 3D Applications

S. Hillaire[1,2,3], G. Breton[1], N. Ouarti[2], R. Cozot[4,2] and A. Lécuyer[2]

[1]Orange Labs, Rennes, France
[2]INRIA, Rennes, France
[3]INSA, Rennes, France
[4]University of Rennes 1, Rennes, France

Abstract

This paper introduces the use of a visual attention model to improve the accuracy of gaze tracking systems. Visual attention models simulate the selective attention part of the human visual system. For instance, in a bottom-up approach, a saliency map is defined for the image and gives an attention weight to every pixel of the image as a function of its colour, edge or intensity.

Our algorithm uses an uncertainty window, defined by the gaze tracker accuracy, and located around the gaze point given by the tracker. Then, using a visual attention model, it searches for the most salient points, or objects, located inside this uncertainty window, and determines a novel, and hopefully, better gaze point. This combination of a gaze tracker together with a visual attention model is considered as the main contribution of the paper.

We demonstrate the promising results of our method by presenting two experiments conducted in two different contexts: (1) a free exploration of a visually rich 3D virtual environment without a specific task, and (2) a video game based on gaze tracking involving a selection task.

Our approach can be used to improve real-time gaze tracking systems in many interactive 3D applications such as video games or virtual reality applications. The use of a visual attention model can be adapted to any gaze tracker and the visual attention model can also be adapted to the application in which it is used.

Keywords: gaze tracking, visual attention model, saliency, first person navigation, virtual environment

ACM CCS: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality

1. Introduction

Gaze trackers are systems used to compute the gaze position of a human [GEN95]. The majority of gaze trackers are designed to compute the gaze position onto a flat screen. Since their creation in the late 19th century, before the computer existed, these systems have advanced considerably [GEN95]. The interest in these systems has grown thanks to their usefulness in several domains: from human studies in psychology to VR systems, as an aid for people with disabilities or to accelerate the rendering process.

Many gaze estimation methods have already been proposed. However, many of them suffer from their complex calibration procedures, their intrusiveness [KBS93], their cost [Tob] or their cumbersomeness [BF03]. These systems are often accurate but, for the reasons aforementioned, cannot be sold on the mass market for daily use. Today, it would be valuable to have a low-cost eye-tracking system relying for instance on a basic web cam connected to a PC, and usable in various conditions, without the need for operational expertise. It could be valuable for interactive 3D applications such as video games, virtual reality applications, etc.

Table 1: *Summary of existing gaze tracking systems.*

Category	Reference	Hardware	Intrusive	Horizontal accuracy (degree)	Vertical accuracy (degree)	Limitations
Intrusive trackers	Kaufman *et al.* [KBS93]	Electrodes	Yes	1.5 to 2	1.5 to 2	Intrusive
	Duchowski *et al.* [DMC*02]	Helmet with two screens	Yes	0.3	0.3	Intrusive and expensive
	Beymer *et al.* [BF03]	Use of two steerable cameras	No	0.6	0.6	Expensive and cumbersome
	Tobii [Tob]	Dedicated capture system	No	0.5	0.5	Expensive
Remote gaze trackers	Yoo *et al.* [YC06]	Infra-red LED and CCD camera	No	1.0	0.8	User must stay between 30 to 40 cm from the screen
	Hennessey *et al.* [HNL06]	Infra-red LED and CCD camera	No	1.0	1.0	Infra-red light
	Guestrin *et al.* [GE06]	Two lights and one CCD camera	No	0.9	0.9	Needs the use of 2 specific light sources
	Yamazoe *et al.* [YUYA08]	CCD camera	No	5.0	7.0	Low accuracy
ANN-based gaze trackers	Baluja *et al.* [BP94]	640 × 480 CCD camera	No	1.5	1.5	Non robust calibration
	Piratla *et al.* [PJ02]	640 × 480 webcam	Yes	Not available	Not available	Non robust calibration
Visual objects trackers	Lee *et al.* [LKC09]	No hardware	No	Object based	Object based	Depends on the VE and user's task

In this paper, we present a novel way of improving the accuracy of any gaze tracking system by using a visual attention model. Our algorithm uses an uncertainty window, which is defined by the accuracy of the gaze tracker, in which more coherent gaze positions can be determined using a saliency map [Itt05] encoding visually salient areas.

In the remainder of this paper, after exposing the related work, we describe the low-cost gaze tracking system we used to compute gaze positions using a web cam and an artificial neural network. This gaze tracker is just an example used to demonstrate the efficiency of our methods. The accuracy and usability of this system is briefly discussed. Then, our novel approach which uses visual attention models to improve gaze tracking accuracy is presented. Finally, we report on two experiments conducted to assess the benefits of the proposed method. The paper ends with a general discussion and conclusion.

2. Related Work

Over the last decade many gaze tracking systems have been developed for various applications. Table 1 summarizes the existing gaze tracking systems, considering the required hardware and their current accuracy.

Intrusive gaze tracking systems are generally restrictive as users have to wear heavy and uncomfortable equipment. As an example, Kaufman *et al.* [KBS93] use electrooculography to measure the eyes' muscular activity. This method requires the user to wear electrodes. Another technique requires the user to wear induction coil contact lenses [GEN95]. The gaze direction can be computed by measuring the high-frequency electro-magnetic fields produced by these lenses. Both these techniques require the user's head to stay still. To overcome this problem, Duchowski *et al.* [DMC*02] propose a helmet with an embedded screen for each eye. Two gaze trackers based on the pupil-cornal reflection (P-CR) method are used (one for each eye). Intrusive systems are precise enough to be interesting for a research purpose, however, as shown in Table 1, few gaze tracking systems are intrusive and the current trend is towards the development of non-intrusive systems.

A new kind of gaze tracker has recently emerged: remote gaze tracker. Remote gaze tracking systems are 'systems that operate without contact with the user and permit free head movement within reasonable limits without losing tracking' [BMMB06]. These systems use either a high-resolution camera or low-resolution web cam and allow users to feel more free because they do not have to wear any devices. However, they are generally less accurate than other gaze trackers. Therefore, a lot of research is still going on to improve remote gaze tracking systems. Beymer and Flickner [BF03] propose a multi camera system tracking first the head of the user using a camera with a wide field of view, then, one of his eyes using a steerable high resolution narrow camera.

Finally, a 3D representation of the eye is used jointly with the infra-red light glint to evaluate the user's gaze position. Tobii technology [Tob] proposes a non-intrusive gaze tracking system which enables some moderate movement of the user's head. It uses expensive dedicated tracking devices using infra-red lights, though further implementation details are not available [MM05]. Table 1 shows that these non-intrusive systems are very accurate but most of them require high expertise, are cumbersome [BF03] or very expensive [Tob].

Other remote gaze tracking systems have been designed to be used in everyday life by non-expert users with a simple and fast calibration process [GE06]. Some of the proposed systems [HNL06, YC06] still require infra-red LEDs but are able to achieve an accuracy of one degree under head movement. All the presented remote gaze trackers use a 3D representation of the eye or the P-CR method to compute the gaze direction. The system developed by Yamazoe *et al.* [YUYA08] is able to compute the gaze position without infra-red light nor a calibration sequence. This system is dedicated to everyday use since it uses a single video camera. It has a low accuracy of 5 degrees horizontally and 7 degrees vertically, but the results are promising and yet could be improved.

Few gaze tracking systems based on an Artificial-Neural-Network (ANN) have been proposed in the literature [BP94, PJ02]. Baluja and Pomerleau [BP94] propose to send a low resolution image of a single eye directly to an ANN. Piralta and Jayasumana [PJ02] compute features describing the current user's state, i.e. eyes' centre, distance between upper and lower eyelid, etc. These features are then used as the input of an ANN. Such systems only need a 640×480 web cam and represent the screen as a discretized two-dimensional grid.

Visual attention represents the capacity of a human to focus on a visual object. It is well known that human visual attention is composed of two components [Itt05]: *bottom-up* and *top-down* components.

The bottom-up component simulates the visual reflexes of the human visual system. Due to the structure of our brain and the fact that we only accurately perceive our environment within 2 degrees of our visual field [CCW03], the human visual system does not have the capabilities to analyze a whole scene in parallel. Actually, the human visual system can detect primitive features in parallel, defining salient areas in the visual field. Then, it uses a sequential visual search to quickly analyze a scene [TG80]. For example, when someone first looks at a scene, his/her gaze is first unconsciously attracted by visually salient areas to rapidly perceive the most important areas [IKN98]. Several visually salient features have been identified in previous research [TG80, IKN98]: red/green and blue/yellow antagonistic colours, intensities, orientations, etc. Inspired by the feature integration theory [TG80], bottom-up visual attention models have been developed to compute a saliency map from an image [IKN98] (for details on how to compute a saliency map, refer to Section 4.2). When a human looks at a picture without any task to do, the saliency value of each pixel of the saliency map represents its attractiveness, i.e. the higher saliency of an area, the more a human is likely to look at this area. Other features have been progressively added in the computation of saliency maps such as flickering [Itt05], depth [LKC09] or motion [YPG01].

Moreover, visual attention is not only controlled by reflexes resulting from visual stimuli, but also by the cognitive process that takes place in the brain, i.e. the top-down component. It is involved in the strategies we use to analyze a scene. For example, Yarbus [Yar67] has shown that the way people look at pictures strongly depends on the task they have to achieve. Furthermore, the top-down component is subject to the habituation phenomenon [LDC06], i.e. objects become familiar over time, and we become oblivious to them [NI05]. Several models have been proposed to simulate the top-down component using task-map [CCW03], habituation [LDC06], memory [NI05] or spatial context [LKC09].

Nowadays, visual attention models are used in various domains for several tasks. For example, they are used to accelerate the rendering of virtual environments, i.e. reflection, global illumination, using selective rendering [CCW03, LDC06, SDL*05, HMYS01], for realistic avatar animation [CMA03], mesh decimation [LVJ05], etc.

3. Our ANN-Based Gaze Tracker

We first propose a low-cost ANN-based gaze tracking system using a single web cam. This gaze tracker, on its own, is not necessarily new. We used it here as an example of a gaze tracking system that can be improved using a visual attention model.

In this section, we expose the hardware requirements, as well as the software architecture of our ANN-based gaze tracker. We detail the calibration sequence, i.e. how the ANN is trained, and its real-time use. Finally we report on an experiment conducted to measure the accuracy of this system.

3.1. ANN-based gaze tracking

Compared to previous gaze trackers based on ANN [BP94, PJ02], we propose to transform the captured images of the user's eyes into higher level primitives. Left and right intersections of the bottom and top eyelid of each eye are manually selected by the user in the video recorded by the web cam using two mouse clicks. Two points per eye are used to extract the image of each eye. During this procedure, the head is maintained in a constant position using a chin-rest. Each time a frame is received from the web cam, the images of the user's eyes are extracted and scaled to images of width W_e and height H_e. Contrarily to Baluja and Pomerleau [BP94] who send the picture of the eye directly to the ANN,

we propose to transform it in order to reduce the number of input of the ANN. First, we apply a contrast-limited adaptive histogram equalization filter to both images, previously transformed from a RGB format to an intensity format, in order to maximize their contrast. In order to reduce the amount of data sent to the ANNs, for each eye image, the pixels of each column and each row are added using Equations (1) and (2) to respectively obtain two arrays S_x (of size W_e) and S_y (of size H_e). After preliminary testing, we could notice that this computation of gaze positions was more stable than using the raw image

$$\forall i \in [1, W_e],\ S_x[i] = \sum_{j=1}^{H_e} eyeImage[i][j] \qquad (1)$$

$$\forall j \in [1, H_e],\ S_y[j] = \sum_{i=1}^{W_e} eyeImage[i][j]. \qquad (2)$$

Finally, for each eye, S_x and S_y have their values mapped from their range [*min value*, *max value*] to the range [0, 1]. This result is stored in S'_x and S'_y arrays. This mapping is important as it allows us to take advantage of the full working range of each neuron activation function. This latter is a linear activation function which works in the [0, 1] range.

For each eye, the arrays S'_x and S'_y are sent to the ANNs. Actually, two ANNs per eye are used : one computes the horizontal position of the gaze point based on S'_x and another computes the vertical position of the gaze point based on S'_y. After preliminary testing, we found that using the S'_x and S'_y arrays as input of two separate ANNs produces smoother estimations of the gaze position. We also found that $W_e =$ 40 pixels and $H_e = 20$ pixels make a suitable size for the scaled image of the eyes given the resolution of the web cam, i.e. 640 × 480, and learning capabilities of the ANN. Moreover, each ANN is composed of two hidden layers; each one containing twenty neurons. Using this architecture, our algorithm is able to evaluate a continuous gaze position on the screen contrary to previous ANN-based gaze trackers which represent the screen as a 2D grid [BP94, PJ02].

3.2. Calibration sequence and gaze tracking

During the calibration sequence, each ANN is trained in order to compute one coordinate of the gaze point based on its associated eye image. For this aim, the user has to follow a target which moves across the entire screen in order to allow gaze tracking on the full screen surface. Finally, each ANN is trained using the retro-propagation algorithm [Wer90].

At the end of the ANN training, the real-time gaze tracking sequence is initiated. As explained before, the gaze tracker computes a gaze position on the screen for each eye. The final gaze position is computed as the mean of the two resulting gaze positions: this produces smoother gaze movements.

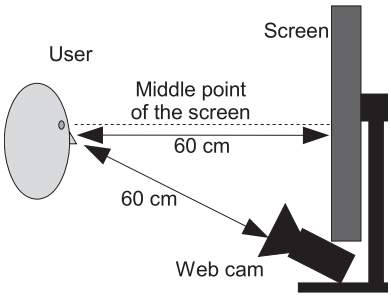

Figure 1: *Hardware setup.*

3.3. Environment and hardware setup

The ANN-based gaze tracker we propose only requires one web cam supporting video capture at a resolution of 640 × 480 pixels. This system is designed to compute the user's gaze position on a flat screen.

The user's head is expected to remain within the range of 40 to 80 cm in front of the screen as illustrated in Figure 1. Furthermore, during preliminary testing, we noticed that our system works better when the height of the eyes is at the level of the centre of the screen. For better performance, we recommend positioning the web cam under the screen and not over it. In this case, the eyes are more visible as they are not hidden by dense upper eyelashes. Currently, the system requires the user's head to stay in a constant position and orientation.

3.4. Accuracy

We assessed the accuracy of our ANN-based gaze tracking system by conducting an evaluation with 6 participants. During the test, they were positioned in front of a flat 19″ screen with a resolution of 1280 × 1024 pixels. They were at a distance of 60 cm from the screen, i.e. resulting in a field-of-view of 35 degrees. No sound was played. We used our ANN-based gaze tracking system to compute, in real time, the participants' gaze position. Their head and eyes were maintained in a constant position using a chin-rest. For each participant, after the calibration sequence, the experiment consisted in successively looking at nine white targets, each one lasting 3 seconds. We recorded the participants' gaze positions computed by the ANN-based gaze tracker together with the current real positions of the targets.

To assess the accuracy of the ANN-based gaze tracker, we computed the differences between the participants' gaze points measured by the gaze tracker and the real targets' positions on the screen. These differences correspond to the distances between the gaze points and targets on the horizontal and vertical axes in normalized screen coordinates. During this sequence, the first 100 milliseconds after each target changes were not taken into account in order to ignore

Table 2: *Mean and standard deviation (SD) for the horizontal and vertical accuracy of our ANN-based gaze tracker.*

	Horizontal accuracy (degree)	Vertical accuracy (degree)
Mean	1.476	1.135
SD	0.392	0.254

errors due to saccades. The mean accuracies are shown in Table 2. The accuracy is highly dependent on the shape of the user's eyes. For small and almost closed eyes, the accuracy can decrease to 1.81° whereas for users with wide open eyes, it can increase to 0.78°.

Since our system does not use infra-red light, the web cam needs ambient light to capture clear images of the user's eyes. Moreover, it requires the user's head to be maintained at a constant position, although previous ANN-based gaze trackers support head movements [BP94, PJ02] to some extent. However, it is used in this paper as an example of a gaze tracker that can be improved by using a visual attention model. To make this tracker suitable for a real use, e.g. for gaming, it could be improved by taking into account the eyes position in the video and yaw, pitch and roll angles of the head similarly to [PJ02].

Our ANN-based gaze tracking system has the advantage of computing a continuous gaze point position instead of a position in a two-dimensional grid representing the discretized screen [BP94, PJ02]. This system is sufficient to achieve various tasks in several environments such as in desktop operating systems or 3D virtual environments. However, it could be improved by taking advantage of the characteristics of the human visual system. This is addressed in the following section and it is considered the main contribution of the paper.

4. Using Visual Attention Models to Improve Gaze Tracking

We propose to improve the accuracy of gaze tracking systems by using a visual attention model.

4.1. General approach

The main idea of our approach consists of looking for salient pixels/objects located near the point given by the gaze tracker and considering that the user is probably looking at these pixels/objects. The global architecture of our algorithm is shown in Figure 2. It consists of two steps: (1) a global step, in which we compute the user's gaze position using, as an example, the ANN-based gaze tracker and (2) a refinement step, in which we compute a saliency map using a bottom-up visual attention model. Therefore, the method we propose to improve gaze tracking systems exploits characteristics of the bottom-up component of the human visual system. We shift the gaze point to the closest most salient pixel, corresponding to the precise/final estimation of the user's gaze point.

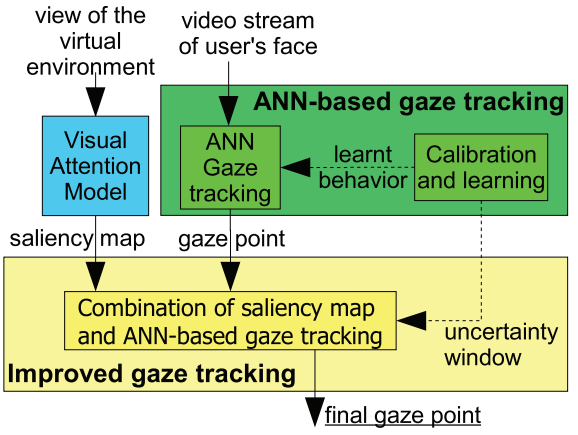

Figure 2: *Global architecture of our system combining classical ANN-based gaze tracking and visual attention model.*

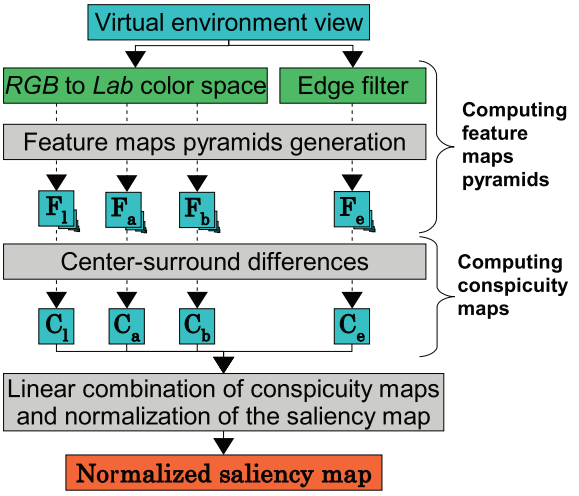

Figure 3: *Algorithm used to compute the saliency map.*

4.2. Computation of the saliency map

To compute the saliency map, we use the bottom-up visual attention model presented in Figure 3. It is inspired by Itti *et al.* [IKN98]. However, to reduce the computation time, it is implemented on GPU hardware using shaders.

First, from the 3D virtual environment image rendered from the current point of view, we compute four *feature* maps. Originally, Itti *et al.* [IKN98] also used four feature maps: red/green and blue/yellow antagonistic colours,

intensities and orientations. In this model, antagonistic colours were computed using simple colour differences. Lee *et al.* [LKC09] improved this computation by using the Hue value of the Hue-Luminance-Saturation colour space. In our case, we propose to use the *Lab* colour space which takes into account the human visual system [Rob90]. In this colour space, relative differences between colours are 'almost perceptually correct' [cie71]. Moreover, this colour space has the advantage of directly encoding red/green and blue/yellow antagonistic colours as well as intensity, i.e. respectively the a, b and L components. They correspond to F_a, F_b and F_l feature maps in Figure 3. In Itti *et al.* [IKN98], another feature map encoding the orientations in the image using a Gabor filter was computed. This filter is expensive to compute so we propose to use an edge filter as in Longhurst *et al.* [LDC06]. It results in the feature map F_e. These feature maps are directly computed in real-time on the GPU using a shader and stored in a single four-component texture.

Second, the feature maps need to be converted into *conspicuity* maps using the multiscale Centre-Surround difference operator as in [IKN98]. This operator aims at simulating the response of brain neurons which receive stimuli from the visual receptive fields. Originally, it needs a dyadic Gaussian feature map pyramid [IKN98]. In our case, we use the same approach as Lee *et al.* [LKC09] which consists of using a mipmap pyramid, containing the original feature maps and several down-sampled copies at a lower resolution, computed on the GPU to reduce computation time. The conspicuity maps, i.e. C_l, C_a, C_b and C_e in Figure 3, are finally computed using Equation (3) with i and $i + j$ being mipmap pyramid levels. The level i is a fine level and $i + j$ a coarser level of the pyramid

$$\forall x \in \{l, a, b, e\}, C_x = \frac{1}{6} \sum_{i=0}^{2} \sum_{j=3}^{4} \left| F_x^i - F_x^{i+j} \right|. \quad (3)$$

Finally, the normalized saliency map is computed by a linear combination of the four conspicuity maps using Equation (4) where S is the final saliency map, $\mathcal{N}$ a normalization operator and $w_x = M_x - m_x$ with M_x the maximum and m_x the mean of the values stored in the conspicuity map C_x. w_x is a factor promoting conspicuity map having small numbers of strong peaks in [IKN98]

$$S = \mathcal{N}\left(\sum_{x \in \{l,a,b,e\}} w_x \times C_x \right). \quad (4)$$

In order to compute the maximum and mean values of C_x, we do not iteratively read the entire conspicuity map using the CPU as this would be too expensive. Instead, we compute the maximum and mean by recursively down-sampling the conspicuity map by a factor of two until we reach the size of one texel which contains the final values. In this algorithm, at each step, and for each pixel of the coarser level, a fragment program computes the maximum and mean values of the conspicuity map's four corresponding pixels computed in the previous step. Once we have obtained these parameters, we can compute w_x for each conspicuity map. In the last step, the final saliency map is normalized using its maximum value (operator $\mathcal{N}$). To find this maximum, we also use this recursive algorithm.

As a result, using our algorithm, the saliency map is computed in real-time using GPU hardware. It takes 9.1 milliseconds for our algorithm to compute a 256×256 normalized saliency map on a nVidia GeForce 7900GTX. To sum up, our algorithm combines techniques of Longhurst *et al.* [LDC06] (orientation approximation by an edge filter) and Lee *et al.* [LKC09] (fast centre-surround operator) bottom-up visual attention models with the original model of Itti *et al.* [IKN98]. We have also accelerated the normalization process of the saliency map by using a pyramid algorithm taking advantage of the GPU hardware.

4.3. Final computation of the gaze position using a saliency map

Given that the accuracy of the gaze tracking system and the distance between the user and the screen are known, we can compute the accuracy of the gaze tracking system in screen coordinates. We define the accuracy Acc_x on the x-axis and Acc_y on the y-axis in screen coordinates. From these values, we can define an uncertainty window Wu. The dimension of Wu are $Wu_x = w_s \times 2.0 \times Acc_x$ on the x-axis and $Wu_y = w_s \times 2.0 \times Acc_y$ on the y-axis, with w_s being a scale factor. Assuming that the user is gazing inside Wu, we propose to improve the gaze tracker accuracy by searching inside Wu for potentially more coherent, i.e. salient, gaze points.

Itti [Itt05] has investigated the contribution of bottom-up saliency to human eye movements. He found that the majority of saccades were directed toward a minority of highly salient areas. Using a normalized saliency map, his experiment showed that 72.3% of the participants' gazes were directed towards an area of the screen containing pixels having a saliency value superior to 0.25. This disk area was centred on the participants' gaze point and has a diameter of 5.6 degrees of their field of view. He suggested that bottom-up saliency may provide a set of gaze locations and that the final gaze point is chosen according to a top-down process. In our case, we know in which area of the screen the user is gazing thanks to the gaze point estimated by the gaze tracker. Thus, we simply propose to search in this area for highly attractive, salient positions.

Based on Itti's work [Itt05], our algorithm takes into account a saliency threshold S_t. First, it searches inside the uncertainty window for the most salient position sp in the normalized saliency map. Second, if the saliency value of sp is greater than the threshold S_t, it sets the final gaze point on sp. On the contrary, if sp is lower than S_t, the gaze point remains unchanged.

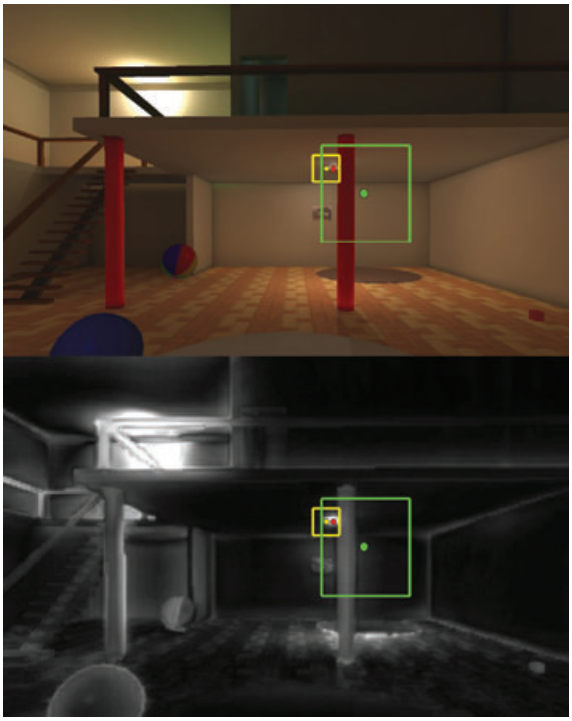

Figure 4: *Combination of low-cost gaze tracking and saliency map to improve performance. Top: view of the scene, bottom: the corresponding saliency map. In yellow, the gaze point and the uncertainty window of the Tobii system (used as theoretical gaze information). In green, the gaze point and the uncertainty window of the low-cost gaze tracker. In red, the gaze point computed by combining a low-cost ANN-based gaze tracker and saliency map.*

Following Itti's work [Itt05], an efficient threshold value for S_t would be 0.25 [Itt05]. However, this value can be adapted according to the application for which the tracker is used. For example, setting S_t to a value of 0 will always set the gaze point position to the most salient pixel inside *Wu*. In the experiment, we present in Section 5, we expose results for several threshold values and several sizes of uncertainty window.

In our model, we could have included a duration of fixation. However, Itti [Itt05] has shown that it is not correlated to saliency values at the level of the gaze point. Moreover, to the best of our knowledge, no other research has found a correlation between saliency maps and gaze duration. Instead, in order to avoid an instantaneous jump between the point estimated by the gaze tracker alone and the gaze tracker improved by the saliency map, the final gaze position estimation is low-pass filtered using a cut-off frequency of 4 Hz.

The use of this algorithm is illustrated in Figure 4. In this case, the gaze point estimated by the ANN-based gaze tracker is far from that estimated by the accurate Tobii system. However, when the ANN-based gaze tracker is combined with a saliency map using our method, the final gaze point is inside the 'Tobii zone'. The Tobii zone takes into account the accuracy of the Tobii system. It is a window centred on the gaze point computed by the Tobii gaze tracker. The size of this zone is defined by both the accuracy of the gaze tracker (0.5° [Tob] for the Tobii system) and the distance of the user from the screen (60 cm). On average, during our two experiments, the sizes of *Wu* when using our ANN-based gaze tracker were 300 × 340 pixels for a 1280 × 1024 screen. For the Tobii gaze tracker, the uncertainty window sizes were 70 × 70 pixels.

5. Experiment 1: Influence of our Algorithm on the Accuracy of Gaze Tracking during Free Navigation in a Virtual Environment

Our first experiment aimed at measuring to what extent our algorithm can improve the accuracy of gaze tracking systems during free navigation in a 3D virtual environment without a specific task. We computed the participants' gaze positions using three different systems: (1) ANN-based gaze tracker, (2) ANN-based Gaze Tracker improved by the bottom-up Visual Attention Model (GTVAM) and (3) a Tobii gaze tracker which is used to compute the 'ground truth' gaze position of the user (Tobii).

Ten naïve participants (9 males, 1 female) with a mean age of 25 (SD = 2.4) participated in our experiment. They were all familiar with the first-person navigation paradigm and had normal vision.

During this experiment, we used the ANN-based gaze tracker described in Section 3 and the Tobii ×50 gaze tracker [Tob]. The ANN-based gaze tracker could be combined with a visual attention model as described in Section 4. We tested the performance of our algorithm under several conditions, i.e. with different values for the saliency threshold S_t and scale factor of the uncertainty window w_s.

Participants were positioned in front of a flat 19″ screen at a resolution of 1280 × 1024. They were at a distance of 60 cm from the screen and no sound was played. Their heads and eyes were maintained in a constant position using a chin-rest. The virtual environment was rendered in real-time at a constant frame-rate of 50 Hz. It represented the interior of a house as shown in Figure 5.

5.1. Procedure

For each participant, the experiment consisted in visiting the 3D virtual environment freely. They navigated using a first-person navigation paradigm using a keyboard to control their motion on the horizontal plane or climb stairs, and the mouse to look around.

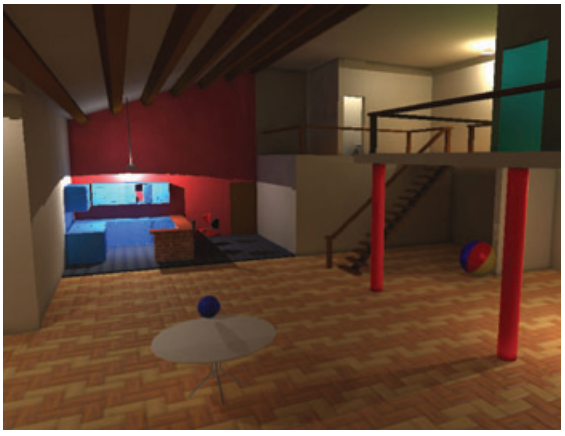

Figure 5: *3D virtual environment used for the experiment.*

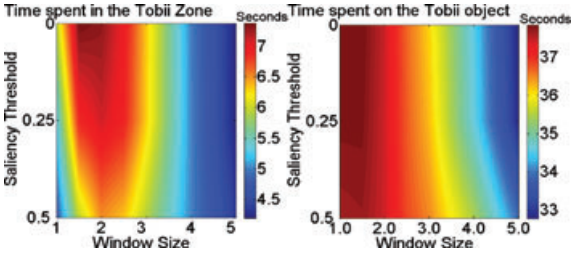

Figure 6: *Left: time spent looking inside the Tobii zone with our approach (GTVAM) for several values of S_t and w_s. Right: time spent looking at the same virtual object as detected with the Tobii system for several values of S_t and w_s.*

Table 3: *Mean performance of our approach (GTVAM) as compared to the ANN-based gaze tracker alone and saliency map alone (i.e. using the whole image on screen).*

	Saliency map alone	ANN alone	GTVAM ($S_t = 0.0$, $W_s = 1.5$)
Time inside Tobii zone	4.4% (2.64s)	5.3% (3.19s)	12.3% (7.37s)
Time on same object as Tobii	23.5% (14.07s)	37.9% (22.75s)	63.1% (37.85s)

The experiment was divided into two parts. The first part consisted in the calibration of the Tobii and the ANN-based gaze tracking system. The training sequence of the ANN lasted 30 seconds. Then, the second part of the experiment began. During this part, the participants were free to navigate around the 3D virtual environment. It is important to stress that since we only tested the bottom-up component of the human visual system (visual reflexes only), the navigation duration was short (1 minute) and no particular task was given to the participants.

5.2. Results

During the experiment, we recorded the participants' gaze positions using the accurate Tobii ×50 gaze tracker (computing gaze position at 50 Hz) and the ANN-based gaze tracker alone. We also recorded positions and movements in the virtual environment of the virtual camera, as well as position and orientation of dynamic objects. Then, offline, we applied our method designed to improve gaze tracking by replaying the recorded sequences of each participant.

As mentioned before, the two main parameters of our model are the uncertainty window scale factor w_s and the saliency threshold S_t. To assess the influence of these parameters on the accuracy, we tested several values for these parameters: $w_s \in \{1, 1.5, 2, 2.5, 3, 4\}$ and $S_t \in \{0, 0.25, 0.5\}$.

We searched for the couple of parameters (w_s*,S_t*) which maximizes the time spent by the computed gaze point inside the Tobii zone. We found that the best values were the same in the two conditions, i.e. the window size w_s* equals to 1.5 and the saliency threshold S_t* equals to 0.0 (see Figure 6).

Then, we compared the performances of our method with performance obtained in two other conditions: ANN alone and saliency map alone. We tested whether the time spent inside the Tobii zone (ZONE) is significantly different for the GTVAM condition as compared with ANN alone and saliency map alone conditions. The same procedure was performed based on the time spent on the same object as the one detected by the Tobii (OBJECT). To compute the object visible at a position on the screen, we used an item buffer containing the unique ID of the visible object at each pixel. The results are summarized in Table 3. To test whether the differences observed are significant or not, we used paired Wilcoxon tests. We first compared ANN alone and GTVAM gaze tracking conditions using the ZONE and OBJECT measures. We found that the distributions are significantly different for the ZONE ($p < 0.01$) and OBJECT ($p < 0.01$) measures. Then, we compared Saliency map alone and GTVAM conditions. We found that the distributions are significantly different for the ZONE ($p < 0.01$) and OBJECT ($p < 0.01$) measures. These results show that our method is able to increase the accuracy of gaze tracking systems.

5.3. Discussion

First, this experiment could be considered as an introduction to a methodology to compute optimal values for the parameters of our algorithm, i.e. S_t and w_s. The values we found can be considered as good indicators for implementing our

approach in similar configurations. Of course, for determining optimal values adapted to another configuration (screen size, application, etc), another procedure could be achieved using the same methodology as described in this paper.

Second, in this study, we defined two measures to assess the validity of using a saliency map computed from a bottom-up visual attention model to improve gaze tracking: time spent by the final gaze point inside the Tobii zone (ZONE) and time spent on the same object as Tobii (OBJECT).

We found that the ZONE and OBJECT measures were significantly improved when using the GTVAM condition as compared to the ANN alone condition. Besides, the accuracy of the Tobii system, which is 0.5 degree of angle, might have reduced the efficiency of our method. The gaze point computed by this system, considered as the ground truth, might sometimes not be located on the object actually gazed by the user. This has been especially observed when the user is gazing at the border of the screen.

Moreover, the uncertainty window may sometimes contain several salient areas that are competing for the final gaze position and the user may not look at the most salient pixel in the normalized saliency map. We can illustrate this with the condition where the higher value in the normalized saliency map of the whole screen is considered as the gaze point, i.e. Saliency map alone condition in Table 3. In this case, the selected gaze position is inside the Tobii zone only 4.4% of the global time. This correlates with Itti [Itt05] results that show that the user does not constantly look at the highest salient pixel. In a virtual environment (VE), a task may always be implied and this would reduce the attractivity of some salient areas [SSWR08], something that the model we used does not take into account for the moment. Furthermore, with our technique, it could be impossible to fixate on some non-salient areas. The use of top-down visual attention components could remove these two inherent problems. Components such as habituation [LDC06], spatio-temporal context [LKC09] or task relevance of objects [CCW03] could be added to our algorithm.

Taken together, our results suggest that the straightforward method we propose can significantly increase gaze tracker performance, especially in the case of object-based interaction.

6. Experiment 2: Influence of our Algorithm on the Accuracy of Target Selection Task during a Video Game

Our second experiment aimed at measuring to what extent our algorithm can improve gaze tracking systems. For this aim, we have developed a simple video game, involving a selection task, in which users play using only their eyes. We again compared three different gaze tracking approaches: (1) ANN-based gaze tracker, (2) ANN-based gaze tracker improved by the bottom-up visual attention model (GTVAM) and (3) Tobii gaze tracker.

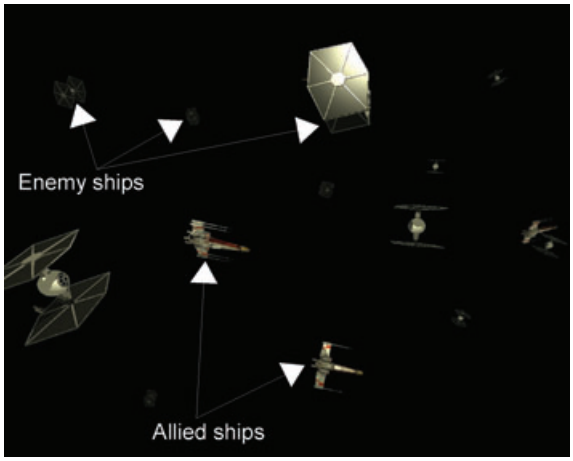

Figure 7: *Screenshot of the game used where players have to look at space ships to destroy them.*

Ten naïve participants (9 males, 1 female) with a mean age of 26.8 (SD=3.2) participated in our experiment. These people did not participate in the previous experiment and had normal vision.

6.1. Procedure

The same apparatus as described in Section 5 was used. For each participant, the task consisted in destroying ships flying through space as shown in Figure 7. They just had to look at one ship to destroy it automatically after a dwell-time of 600 ms. Participants were asked to destroy only enemy ships and not allied ships. Participants had 90 seconds to destroy as many enemy ships as possible. If all enemy ships were destroyed before the end, the game was automatically stopped.

Participants stared in front of the screen, interacting only with their eyes. The camera in the virtual environment was still. There were 15 enemy ships flying and following randomized straight trajectories: 5 ships flying at a distance of $D1 = 45$ m from the camera, 5 ships flying at a distance of $D2 = 80$ m and 5 ships flying at a distance of $D3 = 115$ m. The ships crossed the screen randomly from left to right, or from right to left. There were also 5 allied ships flying and following random trajectories at a distance between 45 m to 115 m. Once a ship left the screen, its new trajectory was computed and its *dwell-time* value was restored to 0 ms. In each condition, the final gaze point computed by the system is used to detect if the user is looking at a ship or not. To do so, we use a classical ray-triangle intersection algorithm at the level of the gazed pixel.

Table 4: *Mean and standard deviation for each measure of the game experiment for each gaze tracking condition.*

	Enemy ships destroyed		Allied ships destroyed		Game time (s)		Destroyed at $D1$		Destroyed at $D2$		Destroyed at $D3$		Time on no object (s)	
ANN	4.55	(2.5)	0.1	(0.3)	90.0	(0.0)	3.8	(1.8)	0.75	(1.3)	0.0	(0.0)	83.0	(2.3)
Tobii	11.6	(2.7)	0.05	(0.2)	85.3	(10.5)	5.0	(0.0)	4.4	(1.1)	2.2	(1.9)	74.1	(11.2)
GTVAM	15.0	(0.0)	0.75	(0.8)	33.2	(18.1)	5.0	(0.0)	5.0	(0.0)	5.0	(0.0)	21.2	(18.5)

During the experiment, the participants played the game six times, two times under each gaze tracking condition. The order of presentation of each gaze tracking condition was counterbalanced. At the end of each game, we recorded the game duration, number of enemy ships destroyed for each distance $D1$ to $D3$, number of allied ships destroyed and time spent gazing at no objects.

6.2. Results

Using the Wilcoxon paired test, we found that our technique (GTVAM) is significantly more efficient to destroy enemy ships than the ANN gaze tracker alone ($p < 0.01$). Surprisingly, our algorithm was found even more efficient than the Tobii ($p < 0.01$) gaze tracker. The mean and standard deviations for each measure are summarized in Table 4.

We could decompose the performance for each distance $D1$ (near), $D2$ and $D3$ (far). We found that GTVAM gave better performances than ANN alone for each distance $D1$, $D2$ and $D3$ (Wilcoxon paired test $p < 0.01$, $p < 0.01$, $p < 0.01$). This corresponds to three different sizes of ship on screen: 200, 100 and 40 pixels. We also found that the difference was not significant between GTVAM compared with Tobii for distance $D1$ and $D2$, but significantly different for distance $D3$($p < 0.01$). This result shows that the better performance of GTVAM compared with Tobii is due to the destruction of the farthest targets (smaller ships on screen).

Another way to evaluate the efficiency of our algorithm is to measure the time spent to complete the mission (destruction of the 15 enemy ships). The GTVAM gaze tracking condition is more efficient when compared with ANN alone (Wilcoxon paired test $p < 0.01$) and Tobii (Wilcoxon paired test $p < 0.01$) conditions. We also computed the time spent on no object and we found that GTVAM is more efficient than the ANN alone (Wilcoxon paired test $p < 0.01$) and Tobii (Wilcoxon paired test $p < 0.01$) conditions.

Our algorithm (GTVAM) is significantly less efficient concerning the destruction of allied ships as compared with ANN alone (Wilcoxon paired test $p < 0.01$) and Tobii (Wilcoxon paired test $p < 0.01$). However, the number of erroneous destructions in the case of GTVAM remains very low (on average less than 1 error per participant).

6.3. Discussion

Overall, using the ANN-based gaze tracking condition combined with our method (GTVAM), participants performed better in the game as compared to the ANN alone and Tobii gaze tracking conditions (Table 4). The GTVAM condition allowed the participants to finish the game faster than under the two other conditions. This is due to the fact that the game ends when all enemy ships are destroyed and this happened only when the GTVAM gaze tracker is used. It can be explained by the lower time spent on no objects in this condition as compared with the two other gaze tracking conditions. However, the number of allied ships destroyed erroneously is significantly higher compared to ANN alone and Tobii conditions. This is a well-know problem in gaze tracking called the *Midas Touch Problem* [Jac95], i.e. people just want to look at an item but it results in an unwanted action. This emphasizes the fact that the method we propose might result in erroneous, i.e. non-intentional, selection. A simple way to reduce these errors would be to ask the player to push a button to fire at enemies.

As shown by Sundstedt *et al.* [SSWR08], a saliency map alone is not sufficient to predict the users attention, especially when a task is explicitly given to the user. In our case, the black background probably helped the GTVAM model. If we had used a more complicated background, e.g. stars and planets, the GTVAM could have performed lower. To overcome this problem, it would be important to take into account the destruction task in the visual attention model such as in [CCW03, LDC06, SDL*05, LKC09]. In this case, we would give a high weight to the enemy ships, a lower weight for the allied ships and a slight weight for the decorative background.

Our analysis showed no significant difference in the number of enemy ships destroyed for near distances ($D1$ and $D2$) between the Tobii and GTVAM conditions. However, the GTVAM gaze tracker has a significantly higher number of far enemy ships ($D3$) destroyed. First, it shows that our algorithm can compensate for the low accuracy of a gaze tracker in such a case. Second, it suggests that it can compensate for the latency in the estimated gaze point position. This latency is due to the acquisition/processing of video and to the computations needed to evaluate the final gaze position on screen. Indeed, thanks to the uncertainty window, the method we propose is able to set the gaze point on salient

objects away from the current gaze point which is subject to latency.

To sum up, despite a small amount of erroneous selections, the GTVAM gaze tracking condition allowed participants to globally perform better in the game. The time to finish the game was the shortest and all enemy ships were destroyed. This experiment emphasizes the fact that the use of a saliency map can increase the tracker accuracy and can also compensate the latency of the tracking systems.

7. Conclusion

We have introduced the use of visual attention models to improve the accuracy of gaze tracking systems in interactive 3D applications.

We have proposed an algorithm using a saliency map which is meant to improve the accuracy of any gaze tracking system such as the ANN-based gaze tracking system described here. It uses an uncertainty window defined by the gaze tracker accuracy and located around the gaze point given by the tracker. Then, the algorithm searches for the most salient points, or objects, located inside this uncertainty window, and determines a novel and, hopefully, better gaze point. We have presented the results of two experiments conducted to assess the performance of our approach. The results showed that the method we propose can significantly improve the accuracy of a gaze tracking system. For instance, during free first-person navigation, the time spent by the computed gaze point on the same object as the ground truth gaze point is increased by 66%. Thus, our approach seems especially interesting in the case of object-based interaction. It even performs better than the accurate Tobii gaze tracker in a game requiring the selection of small visual objects to destroy them. On average, using our approach, the player could destroy two times more small objects on screen than with the standard Tobii system.

Taken together, our results show a positive influence of our algorithm, i.e. of using visual attention models, on gaze tracking. Our approach could be used in many applications such as for video games or virtual reality. Our algorithm can be adapted to any gaze tracking system and the visual attention model can also be extended and adapted to the application in which it is used.

Future work could first concern the evaluation of our method under higher-level tasks by adding top-down components such as proposed in [PI07, SDL*05] or [LKC09]. Second, we could propose using this approach to design new techniques to accelerate [HMYS01] or to improve [HLCC08] the rendering of virtual environments.

References

[BF03] Beymer B., Flickner M.: Eye gaze tracking using an active stereo head. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition* (2003), 451–459.

[BMMB06] Bohme M., Meyer A., Martinetz T., Barth E.: Remote eye tracking: state of the art and directions for future development. In *Proc. of COGAIN* (2006), pp. 10–15.

[BP94] Baluja S., Pomerleau D.: *Non-Intrusive Gaze Tracking Using Artificial Neural Networks*. Tech. rep., Carnegie University, 1994.

[CCW03] Cater K., Chalmers A., Ward G.: Detail to attention: exploiting visual tasks for selective rendering. In *Proc. of the 14th Eurographics workshop on Rendering* (2003), pp. 270–280.

[CMA03] Courty N., Marchand E., Arnaldi B.: A new application for saliency maps: Synthetic vision of autonomous actors. In *Proc. of IEEE International Conference on Image Processing* (2003), pp. 1065–1068.

[DMC*02] Duchowski A., Medlin E., Cournia N., Gramopadhye A., Melloy B., Nair S.: Binocular eye tracking in vr for visual inspection training. In *Proc. of ACM Symposium on Virtual Reality Software and Technology* (2002), pp.1–8.

[GE06] Guestrin E., Eizenman M.: General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on Biomedical Engineering 53*, 6 (2006), 1124–1133.

[GEN95] Glenstrup A., Engell-Nielsen T.: Eye controlled media : present and future state. Master thesis, University of Copenhaguen, 1995.

[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.

[HMYS01] Haber J., Myszkowski K., Yamauchi H., Seidel H.-P.: Perceptually guided corrective splatting. *Computer Graphics Forum 20*, 3 (2001), 142–152.

[HNL06] Hennessey C., Noureddin B., Lawrence P.: A single camera eye-gaze tracking system with free head motion. In *Proc. of ACM symposium on Eye tracking research & applications* (2006), 87–94.

[IKN98] 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 (1998), 1254–1259.

[Itt05] ITTI L.: Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes. *Visual Cognition 12* (2005), 1093–1123.

[Jac95] JACOB R. J. K.: Eye tracking in advanced interface design. *Virtual Environments and Advanced Interface Design* (1995), 258–288.

[KBS93] KAUFMAN A., BANDOPADHAY A., SHAVIV B.: An eye tracking computer user interface. In *Proc. of IEEE Research Frontier in Virtual Reality Workshop* (1993), pp. 120–121.

[LDC06] LONGHURST P., DEBATTISTA K., CHALMERS A.: A gpu based saliency map for high-fidelity selective rendering. In *Proc. of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa* (2006), pp. 21–29.

[LKC09] LEE S., KIM G., CHOI S.: Real-time tracking of visually attended objects in virtual environments and its application to lod. *IEEE Transactions on Visualization and Computer Graphics 15*, 1 (Jan.-Feb. 2009), 6–19.

[LVJ05] LEE C. H., VARSHNEY A., JACOBS D.: Mesh saliency. In *Proc. of ACM Transactions on Graphics* (2005), 659–666.

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

[NI05] NAVALPAKKAM V., ITTI L.: Modeling the influence of task on attention. *Vision Research 45*, 2 (2005), 205–231.

[PI07] PETERS R., ITTI L.: Beyond bottom-up: incorporating task-dependent influences into a computational model of spatial attention. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR07)* (2007), pp. 18–23.

[PJ02] PIRATLA N. M., JAYASUMANA A. P.: A neural network based real-time gaze tracker. *Journal of Network and Computer Applications 25*, 3 (2002), 179–196.

[Rob90] ROBERTSON A. R.: Historical development of cie recommended color difference equations. *Color Research and Application 15*, 3 (1990), 167–170.

[SDL*05] SUNDSTEDT V., DEBATTISTA K., LONGHURST P., CHALMERS A., TROSCIANKO T.: Visual attention for efficient high-fidelity graphics. In *SCCG '05: Proceedings of the 21st spring conference on Computer graphics* (New York, NY, USA, 2005), ACM, pp. 169–175.

[SSWR08] SUNDSTEDT V., STAVRAKIS E., WIMMER M., REINHARD E.: A psychophysical study of fixation behavior in a computer game. In *APGV '08: Proceedings of the 5th symposium on Applied perception in graphics and visualization* (New York, NY, USA, 2008), ACM, pp. 43–50.

[TG80] TREISMAN A. M., GELADE G.: A feature-integration theory of attention. *Cognitive Psychology 12*, 1 (1980), 97–136.

[Tob] Tobii: http://www.tobii.com. Last accessed May 2009.

[Wer90] WERBOS P. J.: Backpropagation through time: what it does and how to do it. In *Proceedings of the IEEE 78*, 10 (1990), 1550–1560.

[Yar67] YARBUS D.: *Eye Motion and Vision*. Plenum Press, 1967.

[YC06] YOO D., CHUNG M.: Non-intrusive eye gaze estimation using a projective invariant under head movement. In *Proc. of IEEE International Conference on Robotics and Automation* (2006), pp. 3443–3448.

[YPG01] YEE H., PATTANAIK S., GREENBERG D. P.: Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Trans. Graph. 20*, 1 (2001), 39–65.

[YUYA08] YAMAZOE H., UTSUMI A., YONEZAWA T., ABE S.: Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions. In *Proc. of ACM symposium on Eye tracking research & applications* (2008), pp. 245–250.