# Real-Time Gaze Tracking with a Consumer-Grade Video Camera

Andrea Keil
TU Braunschweig, Germany
andrea.keil@cg.cs.tu-bs.de

Georgia Albuquerque
TU Braunschweig, Germany
g.albuquerque@cg.cs.tu-bs.de

Kai Berger
TU Braunschweig, Germany
k.berger@cg.cs.tu-bs.de

Marcus Andreas Magnor
TU Braunschweig, Germany
m.magnor@cg.cs.tu-bs.de

**ABSTRACT**

Eye gaze can be a rich source of information to identify particular interests of human users. Eye gaze tracking has been largely used in different research areas in the last years, as for example in psychology, visual system design and to leverage the user interaction with computer systems. In this paper, we present an IR-based gaze tracking framework that can be easily coupled to common user applications and allows for real-time gaze estimation. Compared to other gaze tracking systems, our system uses only affordable consumer-grade hardware and still achieves fair accuracy. To evaluate the usability of our gaze tracking system, we performed a user study with persons of different genders and ethnicities.

**Keywords:** Eye tracking, gaze tracking, human-computer interaction, interactivity, application.

## 1 INTRODUCTION

For us humans, the eyes may be the most important source of information. Our gaze is the primary sense for experiencing our environment. Recent improvements in computer and image processing technologies have allowed our gaze to influence the environment and become an important branch of Human-Computer Interaction (HCI) via Eye and Gaze Tracking. Selecting icons simply by looking at them is a pleasant idea. This can be continued up to the complete controlling of a computer just by looking at the objects of interest instead of selecting them with the mouse. Moreover, this application can be necessary for handicapped people that could write words on a monitor keyboard by looking at the single characters and achieve interactivity with their environment.

As our gaze reflects our attention and our interests, gaze tracking can be used for perceptive tasks as well. What is the customer of the newly developed software application looking at immediately? Can he find all the functionality he needs, and where is he searching for them? When we look at homepages or online-shops or advertisement, what element is drawing our attention in the first place? All those questions may be answered when gaze can be tracked.



***WSCG'2010, February 1 – February 4, 2010***
*Plzen, Czech Republic.*


In this paper, we propose an IR-based gaze tracking system for user applications working in real time with consumer-grade hardware. Gaze tracking systems that use head-mounted devices are usually more accurate and allow free head movement compared to non-intrusive systems (that make no contact with the subject), but such systems are frequently uncomfortable and unnatural to the user. Therefore, we chose developing a non-intrusive gaze tracking system with a fixed camera. Our system consists of a simple setup and does not require expensive cameras.

In the following, we will give a short overview of the related work (see Section 2) and discuss the advantages and the liabilities of our system. In Section 3, our gaze detection pipeline is introduced and each step explained in detail. An application example is presented in Section 3.5 and in Section 4 we present a simple user study that was performed to evaluate the system usability. Finally, we conclude the paper in Section 5 with a short summary and possible future work.

## 2 RELATED WORK

The most popular technique for gaze tracking is the use of a single camera and IR light. The invisible light produces the corneal reflection depicting a strong reference point. Research efforts in this area are focused on simple calibration, accuracy and non-intrusive everyday execution.

Nahlaoui et al. [ANJKS05] developed a gaze tracking system using two IR point light sources for two reference points. Their field of application is the therapy of visually handicapped people. Zu et al. [ZFJ02] used two IR LED rings on a panel that can be turned on and

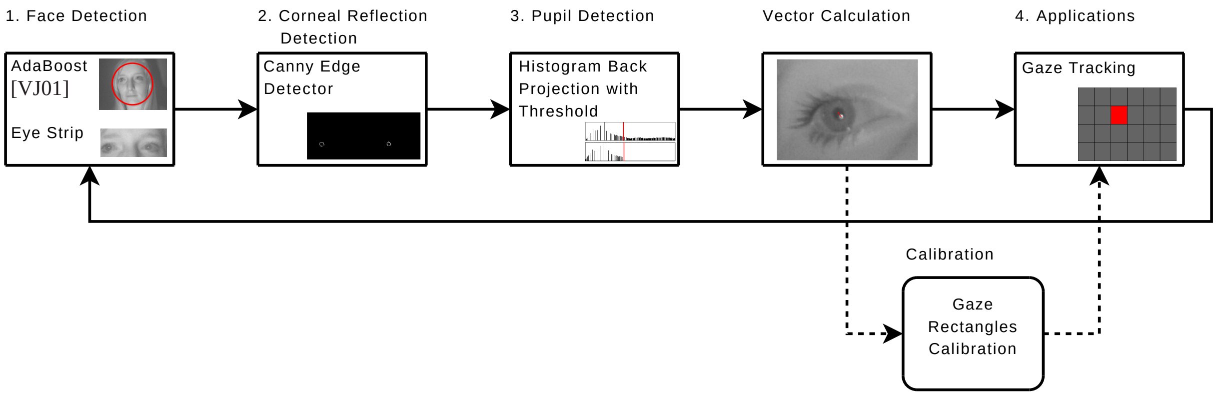



Figure 1: The pipeline of our system.

off alternately in order to receive bright and dark pupil illumination images in the even and odd field of the interlaced image. Other approaches apply a cornea model in order to estimate the gaze [LP07] or use face detection and exploit symmetry to find the gaze direction [MSWB04].

As the IR light shines directly toward the eye, glasses can cause bright extra reflections disturbing or preventing the detection of the corneal reflection. Ebisawa and Ohno et al. [Ebi98, OMY02] developed gaze tracking system that tackles glasses reflections.

A large group of similar approaches is the 3D approach for gaze tracking. They use a stereo camera system for 3D eye localization [OMK03, SL04, SEO96] and derive the 3D center of the corneal curvature in world coordinates. Other works [OM04, PCG+03] additionally use a steerable IR camera on a pan-tilt unit that follows the user so that free head movement is allowed.

Another possibility that allows free head movement is a head-mounted device that is always focused on the user's eye [LWP05, BP04]. Li et al. actually introduced a new pupil detection method called *Starburst* that requires a high eye image resolution, however. This is usually not available when using a standard web-cam.

Morimoto et al. [MAF02] introduced a novel approach for free head gaze tracking, that even works without calibration. This approach is based on the theory of spherical mirrors and the Gullstrand's eye model [Gul09].

In contrast to those proposed methods, our system does not use head-mounted devices because we aimed to develop a non-intrusive gaze tracking system. This decision also supports our goal to create a consumer-grade gaze tracking system that doesn't need fancy additional devices.

We do not use stereo cameras or cameras on a pan-tilt unit as we aimed for low-cost equipment. Instead, we use a single web-cam that is placed on a tripod near the user. The IR LEDs can be put anywhere near the monitor facing towards the user. This guarantees a very easy setup routine.

Furthermore, we apply two dimensional calibration and detection methods instead of 3D approaches where stereo cameras are necessary.

## 3 GAZE DETECTION PIPELINE

Fig. 1 depicts our **gaze detection pipeline** consisting of four main steps. As input, the pipeline receives an image of an ordinary web-cam adapted with an IR-filter. The first step in the pipeline (Step 1) in Section 3.1 reduces the region of interest performing a face detection in the input image, and in the detected face region, an eye strip is cut out so that further computations only occur in this strip. Step 2 in Section 3.2 is the detection of the corneal reflection, a small white reflection point that is produced by an IR light point source. In Step 3 in Section 3.3, the pupil of the human eye is searched in the direct neighborhood of the corneal reflection. From this two points, the *gaze vector* can be determined and the system is calibrated. Finally, the gaze vector is applied to the screen and the gaze can be estimated as Step 4 in Section 3.5.

### 3.1 Face Detection

Figure 2: A detected face via AdaBoost [VJ01].

In Step 1, a face is detected in the given input image. We use AdaBoost for this task, the adaptive boosting algorithm based on Haar-like features proposed in [VJ01] and extended in [LM02]. The in OpenCV implemented cascade was taken for our face detection. An example for the detection can be seen in Fig. 2.

Figure 3: The cut out eye strip from the detected face.

According to traditional rules of proportion, dividing the human face into four equal-sized rectangles, $1 \times 4$, we can determine that the eyes shall be positioned in the second division. From this division, the eye strip, Fig. 3, is generated. These four divisions are computed individually by the face extent given in each face detection step.

## 3.2 Corneal Reflection Detection

The *corneal reflection* is the first Purkinje image that is visible when IR light shines toward the human eye. It appears as a bright white spot in the adjacency of the pupil. There are four Purkinje images that describe where in the eye an object is reflected. The first Purkinje image is the reflection from the outer surface of the cornea, the second from the inner cornea's surface. The third Purkinje image is the reflection from the outer surface of the lens and the fourth from the outer lens' surface.

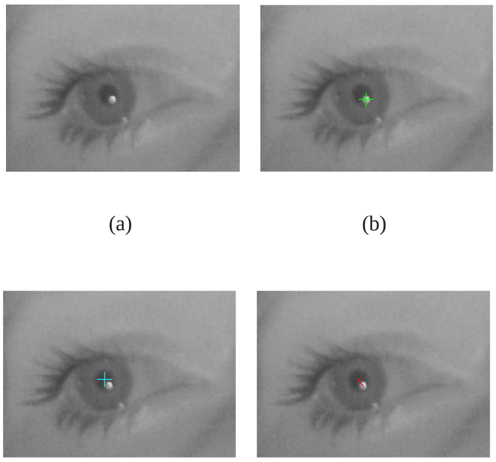
In Step 2, our system uses the Canny edge detector. It applies two thresholds to the eye strip image so that only a small number of white pixels remain in the eye strip image, Fig. 4. They mark edges of the corneal reflection. For the center of the corneal reflection, the centroid of those white pixels is taken, Fig. 5(b).

Figure 4: The output of the Canny edge detector, accentuated. The two corneal reflections produce white pixels in this image.

## 3.3 Pupil Detection

In Step 3, we assume that the pupil is near the corneal reflection. Thus, we search in the neighborhood of the earlier detected point. This is done via histogram back projection and thresholding the histogram. An input image for this method can be seen in Fig 6(a). The corresponding histogram is shown in Fig. 6(c). This histogram is thresholded with a low value and the darkest pixels are kept, Fig. 6(d). The threshold value was determined experimentally.

Via histogram back projection, only the pixels corresponding to the thresholded histogram are back projected into a new image. With this method, the darkest pixels from the input image that derive from the pupil are marked in the new image. As we assume the pupil in the adjacency of the corneal reflection, only pupil pixels are considered, Fig 6(b). Again, the centroid of these marked pixels is taken as the pupil center, Fig. 5(c).



(a) (b)

(c) (d)

Figure 5: Fig 5(a) shows the human eye reflecting IR light as the corneal reflection. In Fig 5(b) and Fig 5(c), a detected corneal reflection and pupil can be seen. A resulting gaze vector is shown in Fig 5(d).

## 3.4 Calibration Method

**Gaze Vector**

A gaze vector $\vec{v}_{input_i}$ with $i \in \mathbb{N}$ is computed between the corneal reflection's center and the pupil's center. The corneal reflection center is defined as the origin and the pupil center is defined as the end. If we assume that the human cornea is a perfect sphere and the user's head remains still, the position of the corneal reflection will not change when the user moves his or her eyes. But as the pupil position changes, the relative position between corneal reflection center and pupil center changes, too. This relative position is represented by the gaze vector that can be used to estimate the user's gaze.

An example of a gaze vector can be seen in Fig. 5(d). The calibration method works as follows: The user has to look at the four corners of the monitor. These four calibration vectors $\vec{v}_{calib_1}, \vec{v}_{calib_2}, \vec{v}_{calib_3}$ and $\vec{v}_{calib_4}$, are saved, where:

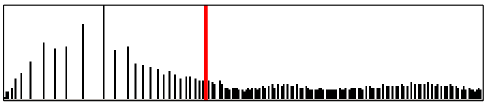
$$\vec{v}_{calib_i} = \begin{pmatrix} c_{i_\mathbf{x}} \\ c_{i_\mathbf{y}} \end{pmatrix}, 1 \leq i \leq 4, \tag{1}$$

with $\vec{v}_{calib_1}$ being the topleft calibration vector, $\vec{v}_{calib_2}$ being the topright calibration vector, $\vec{v}_{calib_3}$ being the bottomleft calibration vector and $\vec{v}_{calib_4}$ being the bottomright calibration vector.

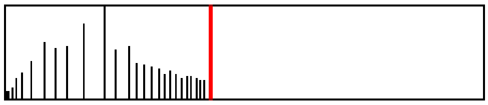
Although we are working on consumer-grade equipment that provides no high definition resolution, a short

(a) The input image.

(b) The back projected image based on the thresholded histogram.



(c) The corresponding histogram to the input image. The red line indicates the threshold value.



(d) The thresholded histogram. The red line indicates the threshold value.

Figure 6: The histogram back projection. The input image (Fig 6(a)) and its histogram (Fig 6(c)). It is thresholded and the resulting histogram (Fig. 6(d)) is back projected as a new image (Fig. 6(b)).

gaze vector can still be used in order to achieve fair accuracy. Therefore, a system of *gaze rectangles* is introduced that divides the screen in a given number of rectangles. This is defined as $A[col, row]$.

The rectangle width $rect_{width}$ and height $rect_{height}$ is computed according to the smallest span of x and y coordinates, respectively, as follows:

$$rect_{width} = \begin{cases} \dfrac{|c_{1_x} - c_{2_x}|}{columns}, & |c_{1_x} - c_{2_x}| < |c_{3_x} - c_{4_x}| \\ \dfrac{|c_{3_x} - c_{4_x}|}{columns}, & otherwise, \end{cases} \tag{2}$$

$$rect_{height} = \begin{cases} \dfrac{|c_{1_y} - c_{3_y}|}{rows}, & |c_{1_y} - c_{3_y}| < |c_{2_y} - c_{4_y}| \\ \dfrac{|c_{2_y} - c_{4_y}|}{rows}, & otherwise, \end{cases} \tag{3}$$

with *columns* being the total number of columns to display and *rows* the total number of rows to display. Both values can be determined by the user before starting the system. $columns \times rows$ defines the system's resolution.

### 3.5 Applications

In Step 4, the computed gaze vector (Section 3.4) is used for the gaze tracking application. The system will highlight only the gaze rectangle the user looks at. In order to determine which gaze rectangle has to be highlighted, the origin of each input vector is translated into the image origin in the upper left of the screen. This is achieved by subtracting the top left calibration vector $\vec{v}_{calib_1}$ from the input vector $\vec{v}_{input_i}$, Fig. 7. In this way, we have all input vectors' origins in the upper left of the screen. This results in new vectors $\vec{v}_{new_i}$.

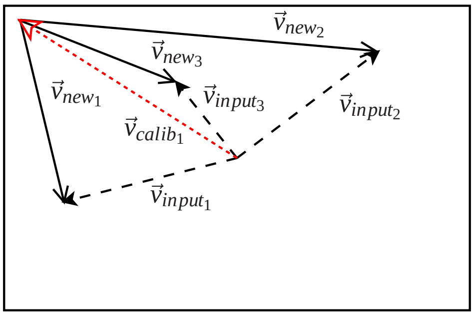



Figure 7: The calibration system that translates every input vector's origin into the upper left corner.

In order to determine the gaze rectangle the user is looking at, defined as $A[col, row]$, Equation 4 and 5 are computed.

$$col = \frac{|input_{i_x} - c_{1_x}|}{rect_{width}}, \tag{4}$$

$$row = \frac{|input_{i_y} - c_{1_y}|}{rect_{height}}, \tag{5}$$

with $i \in \mathbb{N}$, $input_{i_x}$ being the x component of the $i$-th input vector and $input_{i_y}$ the y component, respectively.

## 4 RESULTS

Our gaze detection system was developed in a Linux PC with an AMD Athlon 64 X2 Dual Core Processor 4600+. We used the Philips web-cam SPC 900NC with a resolution of $640 \times 480$ at a frame rate of 15 fps. It has an IR-sensitive CCD chip and was modified with an IR filter. The camera has a F/2.2 lens with an angular aperture of $55^\circ$ and a focal length of $4.5mm$. The system is implemented in C++ and the OpenCV library.

In our experiments, a $6 \times 4$ gaze rectangle grid is displayed on the screen so that the system has a resolution of 24 gaze regions. This can be seen in Fig. 8.

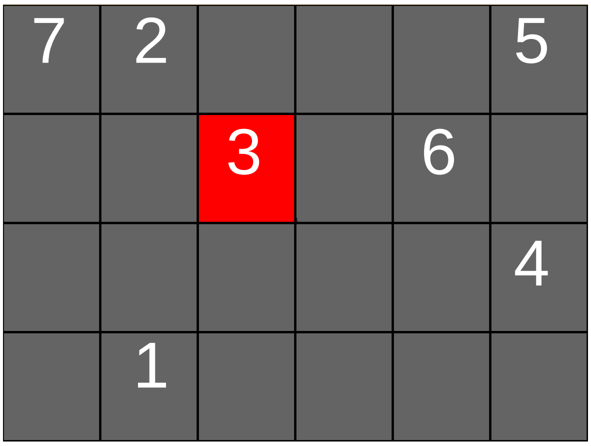



Figure 8: **User Study:** Displayed are seven target rectangles that the user study participants were asked to look at for five seconds. The graph in Fig. 9 shows the mean percentage of hits per target rectangle.

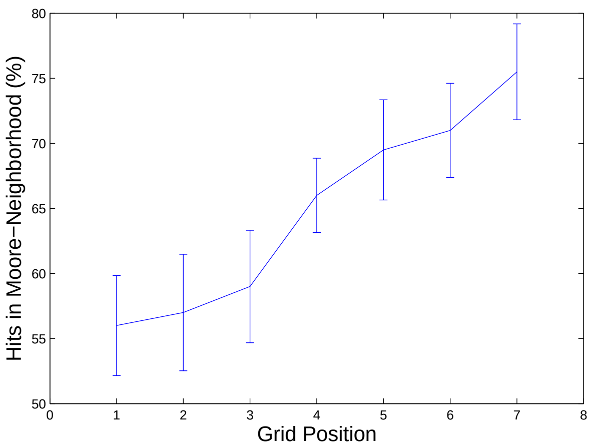



Figure 9: **User Study:** The graph depicts the results of the conducted user study with ten participants. For each of the seven target rectangles (see Fig. 8), the mean percentage of correct hits in the Moore neighborhood is plotted including its variance.

### User Study

To evaluate the accuracy of our gaze tracking system, a user study was conducted with ten participants of different gender and different origins. Although neither of them wore glasses, two wore contact lenses. The contestants sat down comfortably in front of the test system. They were asked to place their heads in the camera's field of view and hold their heads still for the test phase. The camera was placed on a tripod in front of the participants and a little to the left. The IR LEDs were positioned under the monitor facing the user. The calibration was performed by looking at the four corner points (Section 3.4). A test phase consisted of seven target gaze rectangles which the contestants were asked to look at. Per rectangle, their eye motion was recorded for five seconds. The target rectangles are shown in Fig. 8 and were located as follows: Two of them were corner gaze rectangles, number 5 and 7, two were located in the center of the screen, number 3 and 6, and three in the periphery, number 1, 2 and 4. These seven target rectangles provide well suited showcase results.

The graph in Fig. 9 depicts the average results of the proposed system. For each rectangle position in the grid, the mean percentage of correct hits in the Moore neighborhood of the target rectangle during the recording time is plotted. The vertical bars depict the variance of the contestants' results for each grid position.

Further results showed that every single targeted gaze rectangle was hit at least once.

### Infrared Safety

We are working on images that are taken under IR light that has a wavelength of $780nm$ up to $1400nm$. Therefore, an IRED array is used that consists of two IREDs with a wavelength of $880nm$ and a radiant intensity of $160mW/sr$ each, Fig. 10.

This results in an irradiance to the eye of $0.088mW/cm^2$ at a distance of $60cm$. At a distance of $30cm$, the irradiance is $0.355mW/cm^2$. In [SW80], $10mW/cm^2$ are given as a safe amount of IR light under chronic exposure. In [Lam77], the maximum permissible exposure for 16 minutes up to 8 hours is $0.717mW/cm^2$ using our IRED array.

Both values are well below both recommended safety levels.

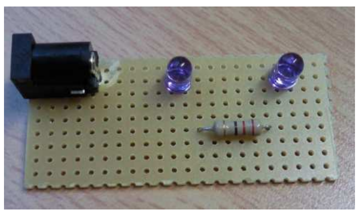


Figure 10: The IRED array we used for our system.

## 5 CONCLUSION AND FUTURE WORK

We proposed an IR-based gaze tracking system for user applications working in real time with consumer-grade hardware. The user study showed that it achieves fair accuracy. The system uses an easy calibration method that requires only four calibration points.

In future work, the corneal reflection and pupil detection should be improved for a more robust detection. This could be done with adaptive thresholding and more precise detection methods, e.g. ellipse fitting. The next step would be to refine the $6 \times 4$ grid if the vector had a much higher resolution. Therefore, we are planning to try other cameras for example with a higher resolution or a better image quality. As the system does not allow head movements, another goal in future work would be to use the face location provided by the face detection in order to compensate head movements.

The calibration method requiring four calibration points could be simplified so that only two calibration points were necessary, namely two opposing corners of the monitor. The gaze rectangle properties could still be computed correctly with information of only two corners. Another calibration method assuming a linear correlation between input gaze vector and output gazing point could be applied as well. This method would require three calibration points for a $2 \times 2$ scale and rotation matrix and a two dimensional offset vector.

## REFERENCES


[ANJKS05] M.Y. Al Nahlaoui, K. Jostschulte, R. Kays, and J. Schmitz. Kostenguenstiges Eye-Trackingverfahren fuer die ambulante Sehtherapie. In *6. Wuerzburger Medizintechnik Kongress*, pages 1–6, 2005. In German.

[BP04] J.S. Babcock and J.B. Pelz. Building a lightweight eye-tracking headgear. In *Proceedings of the 2004 symposium on Eye tracking research & applications*, page 114. ACM, 2004.

[Ebi98] Y. Ebisawa. Improved video-based eye-gaze detection method. *IEEE Transactions on Instrumentation and Measurement*, 47(4):948–955, 1998.

[Gul09] A. Gullstrand. Appendix II. The optical system of the eye, von Helmholtz H. *Handbuch der Physiologischen Optik*, pages 350–358, 1909.

[Lam77] D. Lamarre. Development of criteria and test methods for eye and face protective devices. Technical report, 1977.

[LM02] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *IEEE ICIP*, volume 1, pages 900–903. Citeseer, 2002.

[LP07] E.C. Lee and K.R. Park. A study on eye gaze estimation method based on cornea model of human eye. *Lecture Notes in Computer Science*, 4418:307, 2007.

[LWP05] D. Li, D. Winfield, and D.J. Parkhurst. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *Proceedings of the IEEE Vision for Human-Computer Interaction Workshop at CVPR*, pages 1–8, 2005.

[MAF02] C.H. Morimoto, A. Amir, and M. Flickner. Free head motion eye gaze tracking without calibration. In *Conference on Human Factors in Computing Systems*, pages 586–587. ACM New York, NY, USA, 2002.

[MSWB04] J.J. Magee, M.R. Scott, B.N. Waber, and M. Betke. Eyekeys: A real-time vision interface based on gaze detection from a low-grade video camera. In *IEEE Workshop on Real-Time Vision for Human-Computer Interaction (RTV4HCI)*, pages 1–8. Citeseer, 2004.

[OM04] T. Ohno and N. Mukawa. A free-head, simple calibration, gaze tracking system that enables gaze-based interaction. In *Proceedings of the 2004 symposium on Eye tracking research & applications*, pages 115–122. ACM New York, NY, USA, 2004.

[OMK03] T. Ohno, N. Mukawa, and S. Kawato. Just blink your eyes: a head-free gaze tracking system. In *Conference on Human Factors in Computing Systems*, pages 950–957. ACM New York, NY, USA, 2003.

[OMY02] T. Ohno, N. Mukawa, and A. Yoshikawa. FreeGaze: a gaze tracking system for everyday gaze interaction. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 125–132. ACM New York, NY, USA, 2002.

[PCG+03] A. Perez, ML Cordoba, A. Garcia, R. Mendez, ML Munoz, JL Pedraza, and F. Sanchez. A precise eye-gaze detection and tracking system. In *11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 1–4. Citeseer, 2003.

[SEO96] A. Sugioka, Y. Ebisawa, and M. Ohtani. Noncontact video-based eye-gaze detection method allowing large head displacements. In *IEEE Int. Conf. on Medicine and Biology Society*, pages 526–528, 1996.

[SL04] S.W. Shih and J. Liu. A novel approach to 3-D gaze tracking using stereo cameras. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1):234–245, 2004.

[SW80] D.H. Sliney and M. Wolbarsht. *Safety with lasers and other optical sources: a comprehensive handbook*, page 147. Plenum Press New York, 1980.

[VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, 2001. CVPR 2001. In *Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages 8–14, 2001.

[ZFJ02] Z. Zhu, K. Fujimura, and Q. Ji. Real-time eye detection and tracking under various light conditions. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 139–144. ACM New York, NY, USA, 2002.