Face-Gazer: A Low-Cost Head-Mounted Eye Gaze Tracking System for Measuring Social Interactions

Brandon Jackson and Kerry Clavadetscher
Yale University
Department of Computer Science
51 Prospect Street
New Haven, CT 06520

CPSC 473
1 May 2013

Abstract

In this paper we present an inexpensive head mounted gaze tracker that can be assembled and configured with our gaze tracking software easily. In addition we develop a set of tools for analyzing social conversation gaze patterns 'in the wild.' We discuss the hardware and software decisions we made in building our product and potential applications to human-human, human-computer, and human-robotics interaction research.

Introduction

Gaze and Social Interactions The study of gaze patterns has made extraordinary impact in a diverse range of fields. Retailers study customers gaze patterns as they shop aisles or browse webpages. Neuroscientists examine how babies develop their visual system through gaze studies. In the field of robotics, gaze is of utmost importance. A robot that can determine the point of gaze of a human around it holds priceless information. And for robots trying to mimic natural human behavior, gaze patterns and gaze in coordination with other actions (head gestures, facial expressions, etc.) must be studied as gaze is an integral part of human social interaction.

Gaze-Tracking The most common and least invasive method of gaze tracking today is an Eye Gaze Tracker. At the most fundamental level an Eye Gaze Tracker is a camera, which detects a users eye and tracks the motion of the pupil(s) or iris(es). When used in coordination with knowledge about the users field of view, the position of the pupil can be used to determine where the user is looking. There is a broad application base for this type of system but our interest lies in the research that would be made possible in the field of social robotics.

Problems With Eye-Tracking Research The first problem is that the majority of labs use desk-mounted eye-gaze tracking systems. These make it easy to observe how people gaze at a stable scene, like a screen. However it is difficult to stage many kinds of experiences (such as authentic social interaction) when the subject's head is attached to a bite bar. Head-mounted tracking systems provide a way to observe gaze in natural situations. However most head-mounted systems available today are too expensive for the average social robotics or psychology lab to afford. A variety of gaze tracking systems are currently on the market ranging from the expensive ($10,000+) commercial systems to student-built designs ($100).

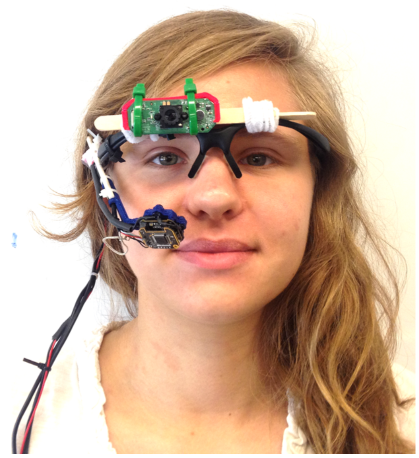

Figure 1: The Face-Gazer Apparatus

Previous Work In recent years there has been increased work in the field of head-mounted eye gaze tracking systems. In 2004, Babcock and Pelz developed a lightweight system designed to be worn in the field. The tradeoff for this enhanced mobility was that the analyses could not be performed in real time. (Babcock and Pelz 2004) There have been a number of domain-specific algorithms developed with the goal of facilitating real-time analysis(Morimoto and Mimica 2005). In 2006, a team of computer scientists led by Li developed a system called openEyes, which was used a novel pupil detection algorithm called Starburst to run in real time.(Li, Babcock, and Parkhurst 2006) Unfortunately neither of these systems are currently in development. At present there are two active groups working on building head-mounted eye gaze-tracking systems. The first is the ITU Gaze Tracking project. This group is focusing on building systems that can work with a variety of hardware setups, with the goal of providing assistive devices for disabled persons. (San Agustin et al. 2010) Kassner and Patera, architects by training, developed a system at MIT in 2012 and are actively building both the hardware and software required for a low-cost system.(Kassner and Patera 2012) Although their system is the easiest for others to reproduce, they have not yet developed any ways to analyze the data that they collect.

Motivation

Eye gaze tracking systems present many challenges for researchers, and our goal is to build both a scalable hardware and software system to collect eye gaze data, and in contrast with previous work, to build a set of tools that makes it easy to perform meaningful social analyses of this data.

Face-Gazer

Our product is a headset (safety glasses frame) with two webcams attached - one forward facing camera capturing the user's field of view and the other just below the right eye capturing the position of the pupil (see Figure 1). The camera below the eye is centered between four IR LED's which provide off axis illumination of the pupil. The two video streams are captured and transmitted via USB to a laptop computer which performs the necessary processing. We take the video from the eye camera and detect the center of the pupil using extensive blurring, thresholding and ellipse fitting. We then convert this position into a point of gaze coordinate in the world camera video using a map constructed from an initial calibration process. Lastly, we detect faces in the world camera video and compute the distance between the nearest face and the point of gaze to analyze the time spent looking at faces while wearing the device.

Materials & Methods

Hardware

A pair of safety glasses with removable lenses served as the base of the headset apparatus. Two cameras were attached to custom-made mounts, and then affixed to the base.

Cameras Two cameras were required: the *eye camera* used to measure the position of the pupil and the forward-facing *world camera* used to capture the scene the user is looking at. The Microsoft Lifecam HD-6000 was selected as the eye camera due to its extremely small form-factor. The Logitech C510 as the world camera due to the built-in image adjustment system which enhaces the vividness of the recording. To make it easier to attach the cameras to the headset apparatus, their cases were carefully removed.

Seeing in Near-Infrared An essential part of the system's pupil detection is its ability to exploit the unique reflectance properties of the human pupil in the near-infrared (NIR) range of light. The pupil acts as a sink in the NIR range, and is far darker than the rest of the eye when irradiated. To take advantage of this, the infrared filter from the eye camera was removed. This was the trickiest part of the hardware construction process, and required the usage of a knife to cut away part of the lens fixture. With the filter removed, the eye camera was able to detect both visible and NIR light.

LED Illumination Four NIR LEDs were used to illuminate the eye. Two main methods for infrared illumination are used in gaze tracking systems: bright (or on axis) illumination and dark (off axis). Because bright illumination often requires low light settings we chose to place the LEDs off axis from the camera and obtain a dark image of the pupil. This four-LED dark illumination technique was inspired by Hua et al (Hua, Krishnaswamy, and Rolland 2006).

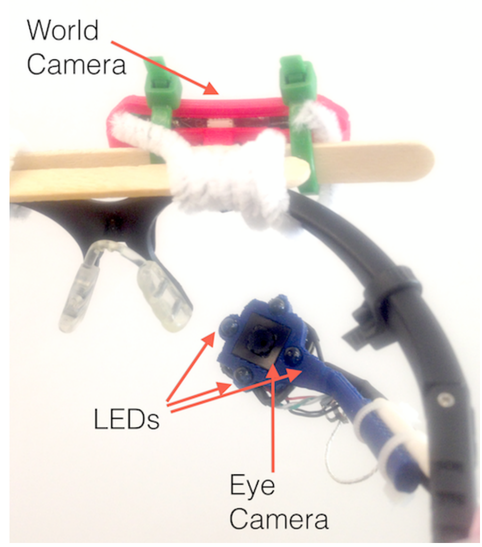

Figure 2: The Headset Apparatus

Mounting the Cameras The cameras were mounted to the base using a combination of 3D-printed parts, popsicle sticks, lollipop sticks, and pipe cleaners. The plans for the 3D printed parts were originally developed by the Pupil project, and are now freely available online.(Kassner and Patera 2012) The world camera was affixed directly above the wearer's right eye. The eye camera was attached to an arm, which was custom-designed to house the camera's principal component board. The plans from the Pupil project were modified to include four holes on each side of the camera, where the NIR LEDs were inserted.

Software

With our hardware in place the goal of our software system was to take at any point in time two input images (one from the eye camera and one from the world camera) and output a single coordinate (the point of gaze) in the world image and to do this in real time. The major steps needed to accomplish that goal were as follows:

- Detect the pupil in the eye image

- Accurately determine its center
- Map the center to a point in the world image to get the point of gaze

The flow chart in Figure 3 depicts these steps as the "Gaze Point Determination."

All head mounted (and even non-head mounted) systems follow this 3 step approach, only varying in the details of each step. Though we experimented with a variety of methods (as explained below) we present the following as the best combination of accuracy, robustness, and speed:

- For detecting the pupil we follow a standard thresholding approach but make use of extensive blurring as we discovered this to be the most effective method for throwing out incorrectly classified pupil candidates in the eye image.
- For accurately determining the center we fit an ellipse to the edges of all points identified as belonging to the pupil in the threshold image.
- For mapping this center to a world coordinate we employ a calibration scheme which acquires a list of pupil centers and corresponding world coordinates. We then fit separate bivariate polynomials to the x world coordinate and y world coordinate.

Detecting the Pupil

There are a variety of pupil detection schemes in use for existing eye trackers today but most of them (and especially ones that make use of IR illumination) use some kind of thresholding approach. Initially this is the method we chose to estimate the pupil location.

Thresholding We simply converted the eye image to grayscale and then filtered the image by sending any pixels darker than some threshold to 0 and anything brighter to 1. The result was a binary image that (ideally) contained a black ellipse on a white background, and our assumption was that anything black in the image belonged to the pupil (see threshold image in Figure 4)

Selecting the Threshold The threshold was selected by adding 25 to the minimum of the light intensity distribution. We quickly discovered this method was not robust enough to deal with the variety of users and lighting it would be exposed to. For example, the glasses frame or darker eyelashes or shadows created by sunlight would often appear in the thresholded image even after adjusting the threshold value. The problem then became determining which pixels in the binary threshold image belonged to the pupil and which to non-pupil objects like eyelashes.

Hough Transforms We then attempted to estimate the center in another way (we thought if we could do this then we might find the pupil pixels in our thresholded image using the distance to this center estimation or size of nearby connected components or even by implementing a sophisticated algorithm like RANSAC (wirski, Bulling, and Dodgson 2012)). The center estimation we did fairly successfully through a combination of Hough transforms (which find circle contours in an image) and template matching (finds the region of an image that best matches a smaller template image). We achieved the best results when we used the Hough transform to first detect all circles and then template matched* around those centers. Once we had this estimate of the pupil center we then attempted to classify nearby pixels as either belonging to the pupil or not. We tried several methods including simple cropping and finding the largest connected component near the center. However, none of the methods we tried returned only the pupil pixels consistently. This was because the pupil was often segmented due to occlusion from bright eyelashes or even worse, the pupil was connected to a dark crease in the eyelid.

Blurring The final solution we came upon derived from the realization that we could get rid of the dark creases around the pupil simply by blurring. Blurring would have the effect that skinny lines (like creases) and edges would become brighter but large dark areas (the pupil) would maintain their darkness. When we tested this theory it had the desired effect the small dark features disappeared from our threshold image and we were able to obtain pixels only belonging to the pupil. In fact we tried blurring the entire image from the very start and discovered if we used a large enough Gaussian (radius of extent 21 pixels) we didnt need to use the Hough transform or template matching because the only pixels left in our threshold image were the pupil pixels.The downside to this approach is that blurring removes pupil edges that would otherwise help give an accurate pupil center. With this threshold image containing only the pupil pixels we were able to easily determine a center for the pupil.

Accurate Pupil Center Determination Since our threshold image only contained dark pixels corresponding to the pupil, estimating a center was trivial. We ran a Canny edge detector over the threshold image and fed those edge points

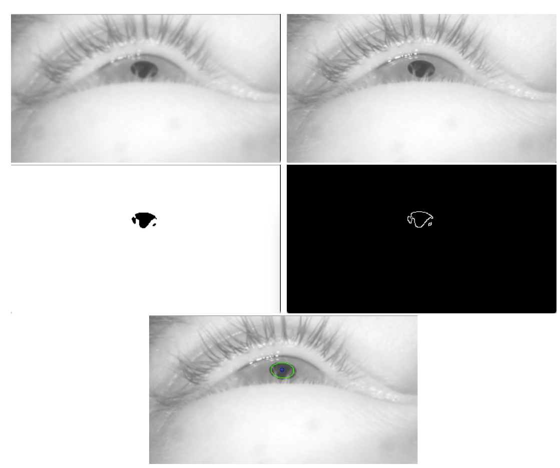

Figure 4: Processing stages (from left to right): original image, blurred image, threshold image, Canny edges, fitted ellipse with center in blue

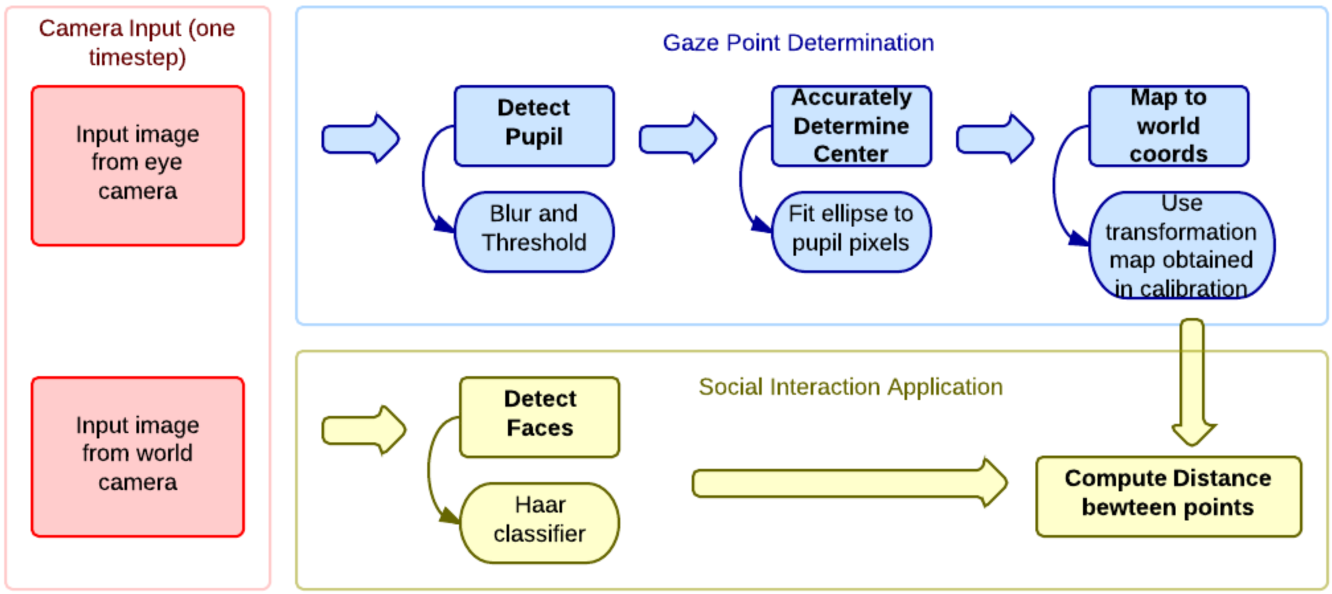

Figure 3: Software flow chart

into an ellipse fitting function. The center of the best fit ellipse became the pupil center.

The series of images in Figure 4 depicts the different stages of processing and the final estimated center.

Calibration

Collecting Calibration Data OpenCV provided built in recognition for our calibration target so obtaining the world coordinates for this point was easy.

Mapping From Pupil Coordinates to World Coordinates With an accurate estimation of the pupil center the task becomes finding the corresponding world coordinate. Some sophisticated systems in use today attempt to create a 3D model of the eyeball to make this mapping either automatic (i.e. to render calibration unnecessary) or at least more intuitive. However, accurate systems still require at least one calibration point (citation), so to avoid the complications involved with building an eye model we decided to drop this idea and focus on accurate calibration. We chose a calibration scheme that has the user stare at a calibration target* while moving their head at random. During this time we collect pairs of pupil center points and calibration target points. We then used a least squares method via SVD (Kassner and Patera 2012) to fit two polynomials to the data (one for each world coordinate) as follows:

$$x_{world} = a_0 + a_1 x_{eye} + a_2 y_{eye} + a_3 x_{eye}^2 + a_4 y_{eye}^2 + a_5 x_{eye} y_{eye}$$

$$y_{world} = b_0 + b_1 x_{eye} + b_2 y_{eye} + b_3 x_{eye}^2 + b_4 y_{eye}^2 + b_5 x_{eye} y_{eye}$$

We considered simple linear interpolation of nearest neighbors but this required more calibration points for a smooth fit. We wanted to avoid having the user stare for too long at the target since we discovered it to be a rather uncomfortable experience. Another method we tried for the mapping was using neural nets to learn the fitting function, but the results we were not able to obtain enough data points from the calibration for a feedforward network to correctly approximate the projection.

Social Analysis

Face Detection Our original intent was to build a tracker that could be used in social robotics or social psychology experiments without distracting the user or requiring them to sit in front of a computer or be otherwise restricted. Such an experiement might be a question as simple as ”How much time do people spend looking at faces during conversation?” In order to prove that our tracker could be used for such an experiment we developed a face detection module using Haar cascade classifiers (Viola and Jones 2004) that detects faces in the world view cameara. We then keep track of the times when a user’s gaze is near a face.

Distance Metric We record the minimum distance between the point of gaze and the closest face in the world view. This record is then saved for postprocessing analysis.

Results

Our goal was to build a robust, inexpensive head mounted tracker that could be used for social robotics experiments. We built a sytem that can accurately determine the point of gaze of the viewer in the world camera and measure the distance between this point and the nearest face.

A Low-Cost Solution

We successfully constructed a robust apparatus using off-the-shelf components.

Table 1: Expenses

Item	Qty	Price
Microsoft Lifecam HD-6000	1	$25
Logitech C510	1	$35
Infrared LEDs	4	$4
Total	-	$64

The source code for the software toolkit and the modified CAD plans used to print the eye camera arm are open source, and available online at github.com/brandonjackson/gaze-tracker.

System Implementation

The system is able with enough calibration points (200+) to determine the point of gaze of the user with a resolution of 10-50 pixels. It also detects faces in the world camera. Figure 5 depicts a face detected in red and the point of gaze as a purple circle in the world image. The user was looking at the persons' face. As can be seen, there was a systematic offset in the y coordinate for this trial.

Discussion

Strengths & Innovations

Cost The cost of the system is much lower than other systems currently available on the market. It is comparable with other low-cost systems.

Arm Design with Integrated LEDs Our arm design and LED array provide an exceptionally stable image of the pupil. The arm design allows the eye camera to be placed securely below the right eye and almost out of the field of view of the user. It does not interfere substantially with the user's visual field and though the image of the eye is off axis it does not detract from determining the pupil center. The arm we designed also incorporates 4 IR LEDs, meaning the IR illumination is very strong and makes using a bright illumination approach unnecessary. In addition, the reflection of the four LED's in the first Perkinje image will provide useful information when for building a 3D model for future iplementations of this project.

Robustness and Simplicity The accuracy of our system is not yet comparable to commercial or even off-the-shelf systems which claim 1 degree of visual angle error. However, for the kinds of research we hope will precipitate (i.e. the question we asked - How long does a user stare at faces during a conversation?) the accuracy of our system suffices. In addition the simplicity of the code the method of blurring to obtain the ellipse center and our polynomial surface fitting to compute a calibration map make our software accessible and improveable (see potential improvements section).

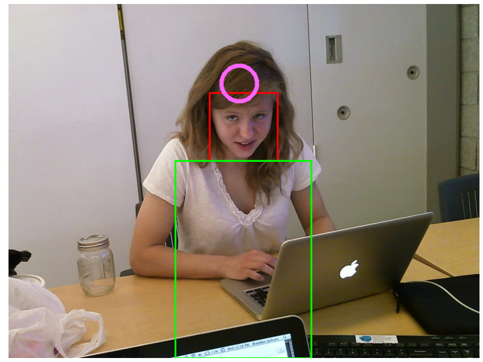

Figure 5: World View. The red box is a detected face. The green box is the body attached to the face. The purple circle is the point of gaze.

Overcoming the Unstable World View Problem Via Social Statistics Previous studies using desk-mounted trackers have been able to easily display data from an entire experimental trial on a single image, since the scene itself remains constant even as the eyes move. This is not possible with a head mounted system, since the headset is designed to move freely with its wearer. Thus it becomes difficult to combine data and generate heat maps and related measurements. The social statistics capabilities of the system are a novel solution to this problem. It is difficult for researchers to gain access to authentic social interactions in a laboratory/experimental setting. The headset allows researchers to capture these events 'in the wild.' Our system is capable of simplifying (and nearly automating) the process of collecting data about social interactions.

Potential Improvements

Hardware Improvements

- Design permanent mounting system to attach world camera to headset
- Inserting over-exposed film between eye camera lens and CCD to reduce visual light in image
- Build recording system to run experiments offline and out of the laboratory

Gaze-Detection Software Improvements

- Use morphological open procendure to close holes in pupil region to obtain better accuracy in occluded images
- Dynamically threshold pupil histogram using a clustering approach
- Accumulate pupil positions over time to crop eye image to a region of interest to avoid detecting dark objects far from pupil
- Take advantage of Purkinje images to improve calibration
- Use RANSAC algorithm to fit point cloud for calibration transformation

Post-Processing Software Improvements

- Stitch together images from the world camera to create a panorama
- Face classification using Fisher Faces (Belhumeur, Hespanha, and Kriegman 1997)

Potential Experimental Uses

Autism Research The headset could be used continue the work of Fred Shic (Shic and Scassellati 2007) and Fred Volkmar (Klin A 2002), studying the perceptual impact of autism disorder. This headset enables the collection of eye gaze statistics in real-life social interactions, whereas the desk-mounted systems used in their work can only be used in conjunction with movies. Since the apparatus can be embedded in real-life situations, it also opens up a new realm of research: building systems that respond to behavior and help the user to adapt to social situations.

Classification of head movements The freely moving apparatus enables data to be collected about a broader range of social expressions beyond eye gaze tracking. One example is head movements that are made while the subject remains fixated on the same thing, such as turning one's head sideways.

Interfaces for the Disabled Gaze-controlled interfaces provide a new way for disabled patients to interact with the world. (Graffiti Research Lab 2009) San Agustin has developed a keyboard interface that enables patients with ALS to type with their eyes. (San Agustin et al. 2010)

Conclusion

As the price of hardware decreases, more energy should be devoting to building dedicated, domain-specific systems that can assist researchers in tackling difficult questions. Our system proves that reliable hardware and software can be built for relatively little cost and even without the most sophisticated software, one can equip themself with a useful tool for research.

References

Babcock, J. S., and Pelz, J. B. 2004. Building a lightweight eyetracking headgear. In *Proceedings of the 2004 symposium on Eye tracking research & applications*, ETRA '04, 109114. New York, NY, USA: ACM.

Belhumeur, P.; Hespanha, J.; and Kriegman, D. 1997. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(7):711–720.

Graffiti Research Lab. 2009. The EyeWriter project. http://eyewriter.org/.

Hua, H.; Krishnaswamy, P.; and Rolland, J. P. 2006. Video-based eyetracking methods and algorithms in head-mounted displays. *Optics Express* 14(10):43284350.

Kassner, M., and Patera, W. 2012. *Pupil: constructing the space of visual attention*. Ph.D. Dissertation, MIT, Cambridge, MA.

Klin A, J. W. 2002. VIsual fixation patterns during viewing of naturalistic social situations as predictors of social competence in individuals with autism. *Archives of General Psychiatry* 59(9):809–816.

Li, D.; Babcock, J.; and Parkhurst, D. J. 2006. openEyes: a low-cost head-mounted eye-tracking solution. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, ETRA '06, 95100. New York, NY, USA: ACM.

Li, D.; Winfield, D.; and Parkhurst, D. 2005. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops*, 79–79.

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

Ohno, T.; Mukawa, N.; and Yoshikawa, A. 2002. FreeGaze: a gaze tracking system for everyday gaze interaction. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, ETRA '02, 125132. New York, NY, USA: ACM.

San Agustin, J.; Skovsgaard, H.; Mollenbach, E.; Barret, M.; Tall, M.; Hansen, D. W.; and Hansen, J. P. 2010. Evaluation of a low-cost open-source gaze tracker. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, 7780. New York, NY, USA: ACM.

Shic, F., and Scassellati, B. 2007. A behavioral analysis of computational models of visual attention. *International Journal of Computer Vision* 73(2):159–177.

Villanueva, A., and Cabeza, R. 2008. A novel gaze estimation system with one calibration point. *Trans. Sys. Man Cyber. Part B* 38(4):11231138.

Viola, P., and Jones, M. J. 2004. Robust real-time face detection. *International Journal of Computer Vision* 57(2):137–154.

wirski, L.; Bulling, A.; and Dodgson, N. A. 2012. Robust real-time pupil tracking in highly off-axis images. In *Proceedings of ETRA*.