

# Geometric Generative Gaze Estimation (G<sup>3</sup>E) for Remote RGB-D Cameras

Kenneth Alberto Funes Mora

Jean-Marc Odobez

Idiap Research Institute, CH-1920, Martigny, Switzerland

École Polytechnique Fédéral de Lausanne, CH-1015, Lausanne, Switzerland

{kfunes, odobez}@idiap.ch

## Abstract

We propose a head pose invariant gaze estimation model for distant RGB-D cameras. It relies on a geometric understanding of the 3D gaze action and generation of eye images. By introducing a semantic segmentation of the eye region within a generative process, the model (i) avoids the critical feature tracking of geometrical approaches requiring high resolution images; (ii) decouples the person dependent geometry from the ambient conditions, allowing adaptation to different conditions without retraining. Priors in the generative framework are adequate for training from few samples. In addition, the model is capable of gaze extrapolation allowing for less restrictive training schemes. Comparisons with state of the art methods validate these properties which make our method highly valuable for addressing many diverse tasks in sociology, HRI and HCI.

## 1. Introduction

As a display of attention and interest, gaze is a fundamental cue in understanding people activities, behaviors, and state of mind, and plays an important role in many applications and research fields. In psychology and sociology, gaze information helps to infer inner states of people or their intention, and to better understand the interaction between individuals. In particular, gaze plays a major role in the communication process, like for showing attention to the speaker or indicating who is addressed, which makes gaze highly relevant for Human Robotics Interaction (HRI).

In another direction, in Human Computer Interfaces (HCI) gaze information coordinated with other user inputs can lead to the development of intuitive systems beneficial for instance for people with limited body mobility.

For these reasons, computer vision based gaze estimation has been studied for over 3 decades [6]. Many solutions have been proposed. Some achieve very high accuracy but require expensive and specialized hardware, like infrared setups or wearable sensors. A solution based on consumer hardware is needed.

To minimize intrusion and accommodate user's movement, remote cameras with wide field of view are preferred but lead to the challenge of low resolution imaging.

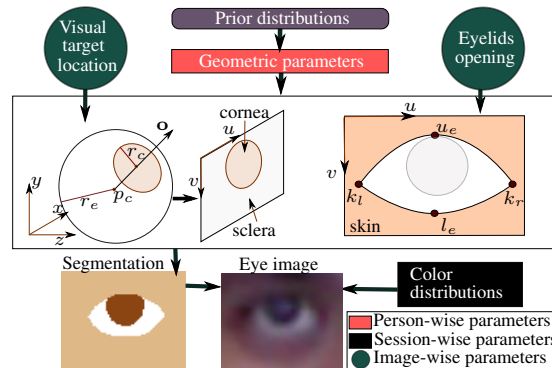


Figure 1: Method overview. The probabilistic generative process links the gazing at a visual target and eyelids movements with a semantic segmentation of the eye region and the resulting eye image observation. This process also depends on (and decouples) user specific parameters describing the eye and eyelid geometry, and ambient/session specific parameters (color distributions).

Appearance based methods, which learn a direct mapping between the eye image to the gaze parameters, avoid local features tracking which is problematic under low resolution conditions. However, they either require large sets of training data to handle eye image variations due to person's specific appearance, head pose, scale, illumination and eyelids movements when learning a general mapping, or require (less) per session training data resulting in overfitting to the person and conditions used during the training phase.

In this paper we propose a head-pose invariant gaze estimation method. It relies on an appearance generative process that model head-pose rectified eye images recovered thanks to the use of consumer RGB-D cameras.

The process is illustrated and briefly explained in Fig. 1, and has several advantages. Thanks to the use of an explicit geometric gaze model, it handles head pose and gaze direction in a unified framework, making it appropriate to reason in the 3D space and extrapolating to gaze directions not seen in the training data, which is useful for instance in inferring attention towards objects or people in HRI rather than only interpolating screen positions. The use of semantic regions (eyelids, cornea, sclera) allows to decouple the

gazing process and user geometry from the ambient conditions (color appearance), while avoiding the critical feature (iris/pupil) tracking problem of standard geometric methods. Overall, the method is able to span a large variety of people and conditions, allowing to easily adapt the model (to the user, viewing conditions) from a few training samples and use it to estimate gaze on unseen data.

Paper structure: Section 2 discuss related works. The RGB-D approach for head-free gaze estimation is described in Section 3. Our gaze model is detailed in Section 4, followed by the inference scheme in Section 5. Section 6 presents experiments and Section 7 concludes this paper.

## 2. Related Work

The recent survey by Hansen [6] provides a comprehensive overview on computer vision methods for gaze estimation. The most accurate techniques rely on eye geometry and on pupil center-corneal reflections detection under IR illumination [4]. They can accommodate head pose variation when using multiple light sources, but need specialized and usually costly IR hardware.

Natural light based methods have also been studied. Many proposals extract geometric models of the eyes, which can be an ellipse fitted to the pupil/iris [8], or complex shapes incorporating the eyelids [18]. Moriyama et al. [12] proposed a generative approach to segment an eye image into many detailed semantic regions (more than 10), but the actual gaze estimation was not investigated. Ishikawa et al. [7] built a geometric model of the eyeball and optical axis from image data, but relied on facial feature tracking and iris ellipse fitting. Closer to our work, Yamazoe et al. [17] proposed to fit such a model from ad-hoc segmentations of the eye images obtained through simple thresholding. However, at test time, they used the iris center derived from a fitted ellipse to infer gaze. Thus, all these methods require high contrast or high resolution images. In addition, further techniques are needed to infer the actual line of sight (LoS), specially for reasoning in 3D scenarios.

To avoid features tracking, there has been an increased interest on appearance based methods [1, 3, 9, 11, 15] that learn a direct mapping from the eye image to gaze parameters. Baluja and Pomerleau [1] trained a neural network but required thousands of training samples, while Williams et al. relied on semi-supervised Gaussian Process Regression (GPR) [16]. Sugano et al. [15] proposed taking user-computer interaction as training data. More recently, Lu et al. [9] proposed adaptive linear regression (ALR) which is based on sparse image reconstruction. They report high accuracy, even using low-resolution test images, but these images were artificially created from the same experimental session. Also, the method required a fixed head pose.

To remove this last constraint, Lu et al. subsequently proposed a GPR-based pose correcting scheme on top of their

fixed head pose model [11]. In another direction, Lu et al. [10] proposed to synthesize eye images as seen from different viewpoints given eye images from a single head pose and a few images from different head poses. More conveniently, Funes and Odobez [3] leveraged on RGB-D cameras to directly handle eye appearance variation by generating frontal looking eye images used as input to ALR.

Altogether, however, appearance based methods suffer from generalization problems. Either they require large amounts of training data [1, 15, 16] to handle variations due to eye shape, pose, illumination conditions, or they are trained from session dependent samples [3, 9, 11] to be used for interpolation. In both cases, the absence of an explicit geometric model make them rather inappropriate for adaptation to users or ambient conditions, or extrapolation, which is problematic when training from a few points on a screen and estimating gaze for different head poses.

Our generative approach has several advantages with respect to the aforementioned methods. Thanks to the use of a color-based semantic segmentation approach, it is suitable for low resolution imaging as compared to traditional geometric based methods, and decouples ambient conditions from the user specific geometry.

The model’s geometric prior makes it appropriate for training from a few samples and extrapolating to other conditions. This is valuable for reasoning in a 3D environment, a desired property in psychology, sociology and HRI while still appropriate for HCI applications.

## 3. Gaze estimation from RGB-D cameras

This section summarizes the gaze estimation method from RGB-D cameras. To acquire head-pose invariance we followed a similar procedure to [3]. In an offline step a 3D face mesh (template) is built for the user by fitting a 3D Morphable model (3DMM) [14] to depth data. Then, in an online stage, the following steps are executed:

- The 3D head pose  $\mathbf{p}_t$  is obtained by fitting, frame-by-frame, the personalized 3D mesh template to depth data using iterative closest points (ICP), resulting for frame  $t$  in the 3D head rotation and translation  $\mathbf{p}_t = \{\mathbf{R}_t, \mathbf{t}_t\}$ .
- Assuming a calibrated RGB-D setup, the RGB-D frame is transformed to a textured 3D mesh. We then re-render the texture, lying on the 3D data surface, using the inverse head pose parameters  $\mathbf{p}_t^{-1} = \{\mathbf{R}_t^\top, -\mathbf{R}_t^\top \mathbf{t}_t\}$ . This results in facial images as if the head was static and in front of the camera. The 3DMM defines a priori the eyes location referred to the head coordinate system. This position is used to crop eye images from the frontal looking facial texture, resulting in pose-rectified eye images.
- The gaze direction is estimated from the pose-rectified eye images using our proposed method (cf. Section 4).
- The gaze direction is transformed back to the world coordinate system, according to the head pose.

## 4. Geometric generative gaze model

Table 1: List of symbols related to our model

Symbol	Description
$I; (u, v)$	Image <i>index</i> and pixel <i>coordinates</i>
$p_c$	Eyeball rotation center
$\kappa = (\phi_\kappa, \theta_\kappa)$	Visual axis deviation
$d$	Nodal point distance from $p_c$
$\mathbf{a} := (\kappa, d)$	<i>Axial</i> parameters
$r_e, r_c$	Eyeball and cornea radii
$k_l = (k_{lu}, k_{lv})$	Left eye corner in image coordinates
$k_r = (k_{ru}, k_{rv})$	Right eye corner in image coordinates
$k_{lr} = (k_l, k_r)$	Left and right eye corners
$\mathbf{s} := (r_e, r_c, k_l, k_r)$	<i>Structure</i> parameters
$p$	Visual target 3D position
$\mathbf{o} = (\phi, \theta)$	Optical axis orientation
$u_e, l_e$	Upper and lower eyelid opening
$\mathbf{m} := (\mathbf{o}, u_e, l_e)$	<i>Movement</i> parameters
$\Lambda_l$	Class $l$ color distribution parameters
$c$	Observed color at pixel $u, v$
$\lambda \in \{0, 1\}$	Occlusion state for pixel $u, v$

### 4.1. Approach overview

The proposed approach is summarized as a block diagram in Fig. 1. Before describing the method, notice that all measures can be referred to a coordinate system fixed to the head (with the  $z$  axis directed towards the head front) due to the procedure described in Section 3. This makes it possible to deal with head fixed quantities. In addition, there is no scale ambiguity in the pose-rectified eye images as depth information provides the pixel size in meters.

The model is characterized by user specific parameters  $\mathcal{U} = \{p_c, r_e, r_c, \kappa, d, k_{lr}\}$ , which define the fixed eye geometry (all notations are defined in Table 1), and image specific parameters  $\mathbf{m} = \{\mathbf{o}, u_e, l_e\}$  related to the actual gaze activity: what is the person’s eye orientation (characterized by the optical axis  $\mathbf{o}$ ) and how are the eyelids open ( $u_e, l_e$ ).

As shown in Fig. 1, given these parameters, an eye and eyelid configuration can be specified, from which a semantic segmentation of the eye region can be generated. The generative process then further combines this segmentation with session dependent color model distributions, parametrized by  $\{\Lambda_l\}_{l=1..3}$ , to produce eye color-images.

Our probabilistic model is thus able to compute the likelihood of such eye images, which constitute our observations. Hence, during a training phase, user parameters can be learned by maximizing the likelihood of gaze annotated training samples, while at test time, the image optimization leads to the actual estimation of  $\mathbf{m}$ , and thus the LoS.

In the following, we describe more precisely the different elements of our model: the eye geometric model, the parametric segmentation function, the definition of the likelihood, and our generative model.

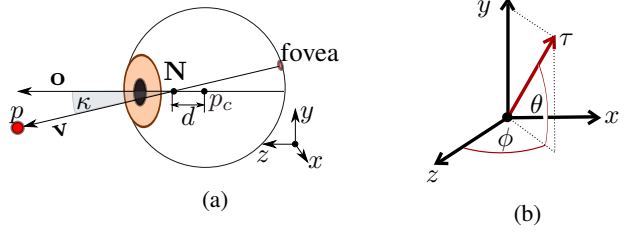


Figure 2: **a)** Eye geometry with optical ( $\mathbf{o}$ ) and visual ( $\mathbf{v}$ ) axis definition. **b)** spherical parametrization of an axis “ $\tau$ ”.

### 4.2. Eye geometric model

Fig. 2a illustrates the geometric eye model we use [6]. The process of gazing a visual target  $p \in \mathbb{R}^3$  consists of rotating the eyeball around the point  $p_c \in \mathbb{R}^3$  such that the *visual axis* ( $\mathbf{v}$ ) intersects  $p$ . The visual axis is the line connecting the fovea (the point of highest visual acuity in the retina) and the nodal point N. It differs from the *optical axis* ( $\mathbf{o}$ ), which is the line connecting the center of rotation  $p_c$  and the pupil center. We parametrize these axis by two angles (as in Fig. 2b). As the eye is a rigid body, the angular difference between these axis is fixed and can be represented by the person dependent angles  $\kappa = (\phi_\kappa, \theta_\kappa)$ <sup>1</sup>:

$$\mathbf{v} = \mathbf{o} + \kappa \quad (1)$$

Thus, implicitly, if the “axial” parameters  $\mathbf{a} := (\kappa, d)$  are known, then the eye rotation ( $\mathbf{o}$ ) can be defined as a function of the position of  $p$ . We denote this process as:

$$\mathbf{o}(p) = (f_\phi(p; \kappa, d, p_c), f_\theta(p; \kappa, d, p_c)) \quad (2)$$

### 4.3. Parametric segmentation function

An eye image is segmented into three regions: the cornea<sup>2</sup>, sclera and skin. The central rectangle in Fig. 1 shows our parametric segmentation: assuming that the user eye geometric parameters  $\mathcal{U}$  are known, then a given eye orientation  $\mathbf{o}$  define a cornea-sclera segmentation, obtained as the orthogonal projection of the 3D cornea contour into the  $xy$  plane, followed by a transformation to image coordinates  $uv$ . This is possible due to the eye image rectification procedure described in Sec. 3 which provides a mapping of the 3D data into the rectified eye image coordinates.

To define the segmentation of the skin region, we rely on a set of parameters characterizing the eyelids structure (eye corners  $k_l$  and  $k_r$ ) and another set controlling the eyelids opening. We take a simple approach, shown in the right part of Fig. 1 where the upper and lower eyelids are quadratic bezier curves sharing the eyelids corners  $k_l$  and  $k_r$ .

The vertical position of the inner control points are denoted as  $u_e$  and  $l_e$ . They define the eyelids opening, and thus, the skin segmentation. The skin class overrides the sclera and cornea regions in the overall segmentation.

<sup>1</sup>This representation ignores eye torsion. Even though it is known that the eyes rotate according to Listing’s and Donder’s laws, this simplification was shown to have little impact on gaze estimation [5].

<sup>2</sup>We define here “cornea” as the region composed of the pupil and iris.





The inferred  $\mathbf{o}$  is used to estimate the direction of the visual axis (cf. Eq. 1). We resorted to Variational Bayes (VB) as an approximate inference method to address the complexity of our model. We summarize the main points below and details are provided in the supplementary material.

**Variational Bayes.** Let  $\mathbf{X}$  denote the observed data and  $\mathbf{Z}$  to be the latent variables to infer. In VB the posterior  $p(\mathbf{Z}|\mathbf{X})$ , which might not be possible to estimate analytically, is approximated by some proposal distribution  $q(\mathbf{Z})$ .

This leads to the definition of the *variational lower bound*  $\mathcal{L}(q)$ , a functional whose maximization with respect to  $q$  is equivalent to a minimization of the Kullback-Leibler divergence between  $q(\mathbf{Z})$  and  $p(\mathbf{Z}|\mathbf{X})$  [2]. The optimal  $q^*(\mathbf{Z})$  is then used as a substitute of the posterior.

**Proposal distribution.** We define  $q(\mathbf{Z})$  with the following parametric form:

$$q(\mathbf{Z}) = \mathcal{N}(\mu_d, \sigma_d) \mathcal{N}(\mu_{\phi_\kappa}, \sigma_{\phi_\kappa}) \mathcal{N}(\mu_{\theta_\kappa}, \sigma_{\theta_\kappa}) \mathcal{N}(\mu_{r_e}, \sigma_{r_e}) \\ \mathcal{N}(\mu_{r_c}, \sigma_{r_c}) \mathcal{N}(\mu_{p_{cx}}, \sigma_{p_{cx}}) \mathcal{N}(\mu_{p_{cy}}, \sigma_{p_{cy}}) \mathcal{N}(\mu_{p_{cz}}, \sigma_{p_{cz}}) \\ \mathcal{N}(\mu_{k_{lu}}, \sigma_{k_{lu}}) \mathcal{N}(\mu_{k_{lv}}, \sigma_{k_{lv}}) \mathcal{N}(\mu_{k_{ru}}, \sigma_{k_{ru}}) \mathcal{N}(\mu_{k_{rv}}, \sigma_{k_{rv}}) \\ \prod_I [\mathcal{N}(\mu_\phi, \sigma_\phi) \mathcal{N}(\mu_\theta, \sigma_\theta) \mathcal{N}(\mu_{u_e}, \sigma_{u_e}) \mathcal{N}(\mu_{l_e}, \sigma_{l_e}) \prod_{u,v} q(\lambda)],$$

where we omit the image and pixel indices to avoid clutter.

Every continuous random variable has been defined as a univariate Gaussian. The motivation for this  $q(\mathbf{Z})$  is that it is possible to compute the derivatives of  $\mathcal{L}(q)$  with respect to the Gaussian parameters. Following [13] we compute the derivatives using Monte Carlo expectations<sup>3</sup> to address the complex relations in our model (cf. Eq. 2 and Eq. 3).

A factorized  $q(\mathbf{Z})$  also allows to optimize  $\mathcal{L}(q)$  in an iterative fashion, where one factor is optimized at the time, leading to an increase of  $\mathcal{L}(q)$  until global convergence.

The only non continuous variable is  $\lambda$ . It can be shown that the optimal  $q(\lambda)$  is a Bernoulli distribution with  $P(\lambda = 1) = \omega$ , where  $\omega$  is given by

$$\omega = \frac{\hat{\lambda}}{(1 - \hat{\lambda})^\epsilon \prod_I p(c|\Lambda_I) \mathbb{E}_{\mathbf{m}, \mathbf{s}, p_c} [S_{u,v}(l; \mathbf{m}, \mathbf{s}, p_c)] + \hat{\lambda}} \quad (4)$$

Notice that  $\mathbb{E}_{\mathbf{m}, \mathbf{s}, p_c} [S_{u,v}(l; \mathbf{m}, \mathbf{s}, p_c)]$  can be interpreted as the *expected segmentation* of an image. According to Eq. 4 an outlier is considered as a color observation which is either unlikely for any class, or that is likely for a given class but is spatially incoherent w.r.t. the geometric model.

**Efficient group factor optimization.** We can optimize  $\mathcal{L}$  efficiently by defining Jacobians over groups of variables (e.g.  $\mathbf{J}_a = [\frac{\partial \mathcal{L}}{\partial \mu_{\phi_\kappa}}, \frac{\partial \mathcal{L}}{\partial \sigma_{\phi_\kappa}}, \frac{\partial \mathcal{L}}{\partial \mu_{\theta_\kappa}}, \frac{\partial \mathcal{L}}{\partial \sigma_{\theta_\kappa}}, \frac{\partial \mathcal{L}}{\partial \mu_d}, \frac{\partial \mathcal{L}}{\partial \sigma_d}]^\top$ ). This is efficient in terms of derivatives computation, as we found that their Monte Carlo expectations require group sampling rather than univariate sampling, due to complex dependencies in Eq. 2 and Eq. 3.

<sup>3</sup>All expectations are defined with respect to  $q(\mathbf{Z})$

Gradient ascent is then used to find the optimal Gaussian parameters of the corresponding factor of  $q(\mathbf{Z})$  (e.g.  $q(\mathbf{a})$ ).

### Inference algorithms.

*Training.* Our overall inference method is given in Algorithm 1. This method finds the person-specific geometry from a set of eye images and their corresponding  $p$ .

*Test phase (Gaze inference).* At test time, the geometry is fixed and we only optimize w.r.t. the test image's  $q(\mathbf{m})$  and outliers in an iterative fashion. In this case the visual target location  $p$  is unknown; as we assume a uniform prior over  $p$ , its influence on  $p(\mathbf{o}|\cdot)$  becomes uninformative. The inferred mode of  $q^*(\mathbf{Z})$  can then be used to derive the MAP visual axis, leading to the 3D line of sight for the given image.

---

#### Algorithm 1 Geometric generative gaze model inference.

---

Set initial  $q(\mathbf{Z})$  from the prior distribution parameters.

**repeat**

- Optimize  $\mathcal{L}$  w.r.t. eye corners and all eyelids opening:  $q(k_{lu})q(k_{lv})q(k_{ru})q(k_{rv}) \prod_I q(u_e^I)q(l_e^I)$
- Optimize  $\mathcal{L}$  w.r.t. eyeball geometry and orientation:  $q(r_e)q(r_i)q(p_{cx})q(p_{cy}) \prod_I q(\mathbf{o}^I)$
- Optimize  $\mathcal{L}$  w.r.t. axial parameters and eyeball depth:  $q(\mathbf{a})q(p_{cz})$
- Update outliers  $q(\lambda_{u,v}^I)$  for all pixels using Eq. 4

**until** Convergence

**Return**  $q^*(\mathbf{Z})$

---

## 6. Experiments

To validate our model, we first studied its behavior using synthetic data. We then compared it against representative geometric and appearance approaches on real data to validate its advantages and added properties.

### 6.1. Experiments on synthetic data

To validate our method we created synthetic data using the generative process described in Section 4.5. Examples are shown in Fig. 4; their resolution in pixel is  $55 \times 40$ .

Synthetic data allows us to study the inference scheme and the observability of the gaze model parameters by comparing the parameters inferred by G<sup>3</sup>E to their true values.

The left plot of Fig. 5 shows the parameter estimation errors as a function of the number of training samples, where each parameter is inferred separately while the other parameters are set to their true values. We can conclude the following: i) almost all parameters can be well estimated, and

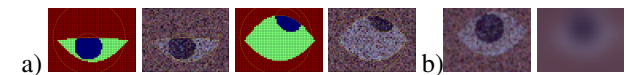


Figure 4: a) Synthetic data samples (drawn segmentation and the generated image from color sampling). b) Sample (left) smoothed by a gaussian filter of  $\sigma = 3.0mm$  (right).

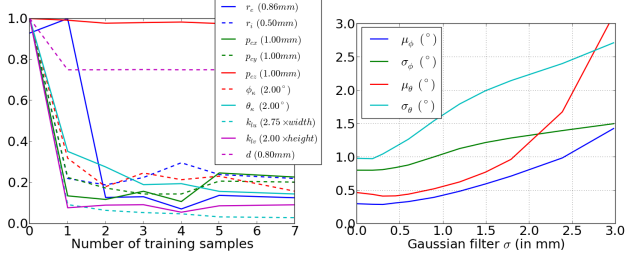


Figure 5: Left. Parameter estimation error vs. number of eye training samples. The y axis scale is given in the legend of each parameter. Right. Mean and standard deviation (derived from the inferred  $q(\mathbf{o})$ ) of the gaze estimates  $\mathbf{o} := (\phi, \theta)$ , vs. the standard deviation of the Gaussian blurring filter ( $1mm = 1.68pixels$ ). For the gaze means, the deviation from their true values is plotted. For each experiments, averages over 500 runs are reported.

this requires only a few samples; ii)  $d$  and the eyeball depth  $p_{cz}$  are difficult to infer, due to their small impact on  $\mathbf{o}$ . Nevertheless, this means that their impact on gaze estimation is small. iii) The visual axis  $\kappa$  angle parameters, which are important for accurate gaze estimation but are often neglected, are well constrained by the image likelihood and the known object position  $p$ , and can thus be inferred.

The right plot of Fig. 5 shows a similar experiment: we evaluate the gaze estimation accuracy (given true geometric parameters) as a function of image resolution simulated through blurring (see Fig. 4). Notice the high robustness w.r.t. resolution due to the optimization of a global image likelihood measure. We also show the estimated variances, which correctly reflect the uncertainty of the gaze estimates.

These results suggest that given proper training data, our approach can potentially have highly accurate gaze estimation ( $< 2^\circ$  error) under poor sensing conditions.

## 6.2. Real data collection

We collected RGB-D data using a Kinect sensor. To collect ground-truth, we asked participants to gaze at a visual target while either keeping their head fixed, or in second phase, asking them to rotate the head (observing  $\pm 30^\circ$  pose ranges for head yaw and elevation) while gazing. Each phase lasted a few minutes. As target, we used either a point displayed on a 24" flat screen or a moving floating target located between the participant and camera. The screen was calibrated with the camera so that 2D screen coordinates are interpreted as a 3D point. The floating target was automatically tracked to find its 3D position.

The distance of the participant to the screen and sensor was  $\approx 85cm$ . In the experiments using the floating target, people distance to the sensor ranged between 1m and 1.5m, resulting in eye image sizes between  $20 \times 14$  and  $13 \times 9$ . In our experiment, the pose-corrected image were upsampled to a fix resolution of  $55 \times 40$  pixels with known pixel size ( $0.595mm/pixel$ ). As performance measure, we used the

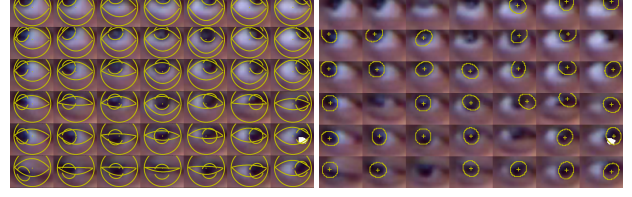


Figure 6: Left: Geometric fitting given by G3E. Right: Ellipse fitting given by the Starburst algorithm on training data collected using the floating target.

angular gaze error, defined as the angle between the estimated line of sight (LoS) and the vector pointing from the LoS's origin to the (known) visual target's 3D position ( $p$ ).

## 6.3. G<sup>3</sup>E inference and geometric methods

We illustrate the inference process output for the training samples shown in Fig. 6. The result of the training can be visualized using the mode of  $q^*(\mathbf{Z})$  (MAP estimate) and by overlaying the contours of the associated segmentation and eyeball structure, as shown in Fig. 6 (left).

Our method follows properly the position of the eyelids and eye orientation despite the low resolution and the sometimes unclear boundaries between eye regions.

As a qualitative comparison, we tested the Starburst algorithm [8] on the same data. This approach is representative of the geometric paradigm, which relies on fitting an ellipse to the cornea from the voting of thresholded gradients, estimated along rays from an initial estimation of the cornea region center. As initialization we set the true center value. Despite this, and parameters tuning, we obtained the results shown in Fig. 6. The low recall and unaccurate estimation demonstrate the important difficulties of ellipse fitting, which is a critical step for many geometric gaze estimation methods, e.g. [7]. Notice that our approach does not have this limitation as it avoids local feature computations.

## 6.4. Appearance based methods

We compared our approach to Funes and Odobez's method [3]. To our knowledge, this is the only method using RGB-D data, for which the eye images pose-rectification was proposed. This method in turn uses Adaptive Linear Regression (ALR) for gaze estimation from the low-resolution pose-rectified images. ALR is a state-of-the-art appearance based method proposed by Lu et al. [9].

By comparing to [3] we indirectly compare to [9] within the pose-rectified eye images context. In the following, when referring to ALR, we implicitly refer to [3]. Our intention is to contrast to the appearance based paradigm. We now describe the result of experiments designed to raise awareness of the limitations of appearance based methods.

**Number of training samples.** As concluded in Section 6.1 our model is adequate for training from few data. We validated this on real data, and contrasted to ALR, as shown in

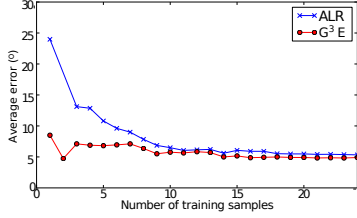


Figure 7: Average gaze error as a function of the number of training samples. Computed on test data from a participant gazing at a floating target with a fixed frontal pose.

Fig. 7, which presents a typical error curve obtained for a participant in functions of the number of training samples.

As shown in Fig. 7, ALR need to cover densely the gaze space with the training samples in order to achieve lower errors. This is a limitation of appearance based methods.

**Gaze extrapolation.** As our method is based on a explicit eye model, we argue it can extrapolate to gaze directions outside the training set. To illustrate this, we conducted an experiment where we used the same 49 samples restricted to gaze yaw and elevation angles within the range  $[-15^\circ, 15^\circ]$  to train the ALR and G<sup>3</sup>E models.

Fig. 8 shows the gaze tracking results on a test sequence, where our claim is validated. ALR, as any interpolation based method, is not able to estimate gaze outside the range of directions used for training, thus causing the saturations observed in Fig. 8. This is not a limitation of our method.

**Gaze estimation across different sessions.** In this experiment we collected data with the floating target in two different sessions (A and B) performed 6 months apart, each for two participants. Across sessions there is a drastic change in the illumination and distance to the camera (see Fig. 9).

We then learned an ALR and a G<sup>3</sup>E model using the data from session A. For the G<sup>3</sup>E approach, applying directly the learned model -including the color distributions from session A- on session B results in large errors (40.2° and 38.2° for the respective participants). This is due to the obvious color mismatch between the two conditions. However, we can easily leverage on the important property of our model which is the decoupling of the ambient conditions from the person-specific geometry. By learning the color distribu-

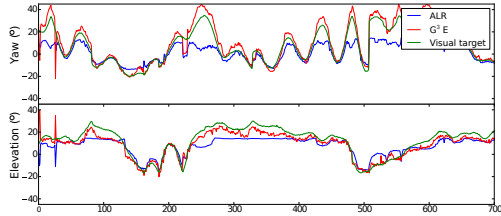


Figure 8: Estimated eye rotation (°) on a test sequence, with training samples restricted to the  $[-15^\circ, 15^\circ]$  range.

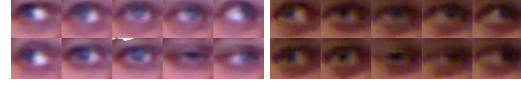


Figure 9: Eye image samples across different sessions. Participant 1 in session A (left) and session B (right).

Table 2: Mean angular error (°) when training the model on session A and estimating gaze on session B. See text.

Method	Participant 1	Participant 2
ALR	21.7	25.0
G <sup>3</sup> E	8.0	5.5

tions of session B using color samples picked from a single eye image in session B, we quickly obtain an adapted model that result in very good performance, as shown in Table 2.

On the other hand, given the lack of geometrical model, ALR does not offer much flexibility for adaptation. Even if it relies on normalized features that should be robust to global illumination variation within the eye image [9], results in Table 2 shows that the Session A model is not appropriate for Session B, demonstrating that session changes go beyond simple illumination and contrast corrections.

The automatic learning of color distributions for G<sup>3</sup>E is left for our future work, but this experiment already validates its potential for cross-session adaptation.

## 6.5. Screen gazing evaluation

We evaluated the performance of our method for the screen target estimation task, where we considered both the fixed and moving head pose case for the five participants.

Notice that, due to the proximity to the depth sensor, there is regularly missing depth data, which affects the pose-rectified eye image, as it is visible in Fig. 10.

In our method, we address this problem by forcing the pixels to be outliers (i.e. setting  $\omega \sim 1$  for missing pixels). However, as ALR does not provide a straightforward way to handle missing data, we do not report ALR results here.

Results are summarized in Table 3. Given the quality of the input data and that head pose variation was within a range of  $\pm 30^\circ$  for yaw and elevation, the performance are highly promising. To illustrate this, we provide in Fig. 10 an example of the setup together with qualitative segmentation results for the 5 participants.

They show that our method has a good behavior at test time, although we observe on the bottom right a problematic situation for our approach which is extreme gazing down, where the cornea region gets heavily occluded by the eyelid.

Table 3: Gaze angular median error (°) for people looking at screen targets.

Head pose	Participant					Avg
	1	2	3	4	5	
Fixed	2.9	2.7	3.1	2.5	5.9	3.4
Moving	9.3	5.5	3.6	4.6	8.6	6.3

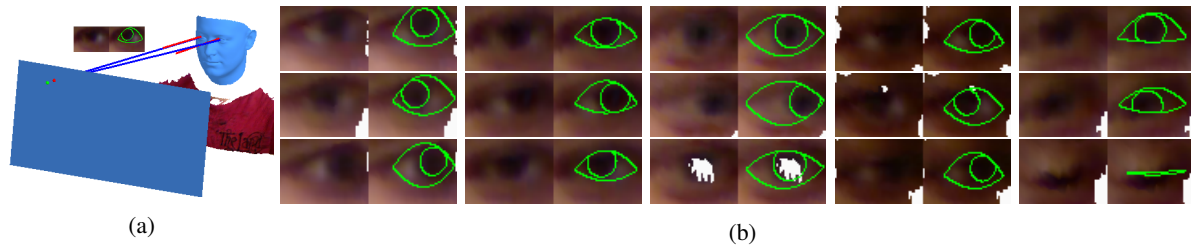


Figure 10: Screen gaze estimation task. **a)** RGB-D frame. The user’s facial 3D template is rendered with the estimated head pose. The blue lines and green dot on the screen are the ground truth. The red lines correspond to the estimated lines of sight and the red dot is the screen intersection (for the left eye). Video results are provided in the supplementary material. **b)** Test left eye images ( $\approx 20$  pixels eye width prior to pose rectification). Each column is for a different participant. White pixels are missing data. Contours represent the mode of  $q^*(Z)$ .

## 7. Conclusions

We have proposed a novel method for head pose invariant gaze estimation from RGB-D cameras. We call it geometric generative gaze estimation ( $G^3E$ ). It is based on a geometric understanding of the 3D gaze action and generation of eye images, formalized as a generative process.

We developed an inference technique, based on Variational Bayes, to find the person specific geometric parameters from training data (i.e. gaze annotated eye images).

We have shown that our method has many advantages with respect to previous approaches. Due to priors on the geometric parameters it is adequate for training from a few samples. Our model is able to extrapolate outside the training data, unlike previous approaches based on appearance. It has also proven adequate for low resolution imaging. Finally, our model correctly decouples the person specific geometry from the observed pixel values, which are dependent on the ambient conditions. This is adequate for adaptation and estimating gaze in different situations. These advantages were validated using both synthetic and real data.

We believe our method has an important potential for gaze estimation in many different scenarios, making it relevant for HCI, HRI, sociology and psychology.

**Acknowledgments** The authors gratefully acknowledge the financial support from the Swiss National Science Foundation (Project: 200021\_130152, TRACOME) [www.snf.ch](http://www.snf.ch).

## References

- [1] S. Baluja and D. Pomerleau. Non-Intrusive Gaze Tracking Using Artificial Neural Networks. Technical report, CMU, 1994. **2**
- [2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, Oct. 2007. **5**
- [3] K. A. Funes Mora and J.-M. Odobez. Gaze estimation from multimodal Kinect data. In *Computer Vision and Pattern Recognition Workshops*, pages 25–30, June 2012. **2, 6**
- [4] E. D. Guestrin and M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *Trans. on bio-medical engineering*, June 2006. **2**
- [5] E. D. Guestrin and M. Eizenman. Listing’s and Donders’ laws and the estimation of the point-of-gaze. In *Symp. on Eye Tracking Research & Applications*, Austin, TX, 2010. **3**
- [6] D. W. Hansen and Q. Ji. In the eye of the beholder: a survey of models for eyes and gaze. *IEEE trans. on pattern analysis and machine intelligence*, 32(3):478–500, Mar. 2010. **1, 2, 3**
- [7] T. Ishikawa, S. Baker, I. Matthews, and T. Kanade. Passive Driver Gaze Tracking with Active Appearance Models. In *Proc. World Congress on Intelligent Transportation Systems*, pages 1–12, Oct. 2004. **2, 6**
- [8] D. Li, D. Winfield, and D. Parkhurst. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. *Computer Vision and Pattern Recognition Workshops*, 3:79–79, 2005. **2, 6**
- [9] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Inferring human gaze from appearance via adaptive linear regression. In *Int. Conf. on Computer Vision*, Barcelona, Nov. 2011. **2, 6, 7**
- [10] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Head pose-free appearance-based gaze sensing via eye image synthesis. In *Int. Conf. on Pattern Recognition*, Nov. 2012. **2**
- [11] F. Lu, O. Takahiro, Y. Sugano, and Y. Sato. A Head Pose-free Approach for Appearance-based Gaze Estimation. In *Proc. of the British Machine Vision Conference*, 2011. **2**
- [12] T. Moriyama and J. Cohn. Meticulously detailed eye model and its application to analysis of facial image. *Int. Conf. on Systems, Man and Cybernetics*, 1:629–634, 2004. **2**
- [13] M. Oppel and C. Archambeau. The variational gaussian approximation revisited. *Neural Comput.*, Mar. 2009. **5**
- [14] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In *Proceedings of Advanced Video and Signal based Surveillance*, Genova, Italy, 2009. IEEE. **2**
- [15] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike. An incremental learning method for unconstrained gaze estimation. In *ECCV*, pages 656–667. Springer, 2008. **2**
- [16] O. Williams, A. Blake, and R. Cipolla. Sparse and semi-supervised visual mapping with the S3GP. In *Computer Vision and Pattern Recognition*, pages 230–237, 2006. **2**
- [17] H. Yamazoe, A. Utsumi, T. Yonezawa, and S. Abe. Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions. In *Symp. on Eye Tracking Research & Applications*, New York, 2008. **2**
- [18] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, Aug. 1992. **2**



# Supplementary material for “Geometric Generative Gaze Estimation (G<sup>3</sup>E) for Remote RGB-D Cameras”

Kenneth Alberto Funes Mora and Jean-Marc Odobez

## Introduction

In this document we describe in detail the inference method for the paper: “Geometric generative gaze estimation (G<sup>3</sup>E) for remote RGB-D cameras”. We will first define a set of functions related to the geometric model and the parametric segmentation function and we’ll then describe the inference methodology.

For convenience we include the graphical model in Fig. 1 and the list of parameters in Table 1. It is important to mention that the “hat” ( $\hat{\cdot}$ ) notation represents the hyperparameters of a prior distribution, e.g.  $\hat{d} := (\hat{\mu}_d, \hat{\sigma}_d)$ . This notation will be used throughout this document.

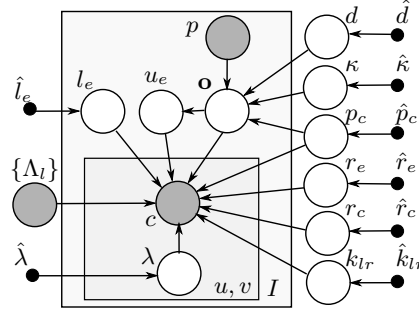


Figure 1: Graphical representation of the geometric generative gaze estimation model.

Table 1: List of symbols related to our model

Symbol	Description
$I; (u, v)$	Image <i>index</i> and pixel <i>coordinates</i>
$p_c$	Eyeball rotation center
$\kappa = (\phi_\kappa, \theta_\kappa)$	Visual axis deviation
$d$	Nodal point distance from $p_c$
$\mathbf{a} := (\kappa, d)$	<i>Axial</i> parameters
$r_e, r_c$	Eyeball and cornea radii
$k_l = (k_{lu}, k_{lv})$	Left eye corner in image coordinates
$k_r = (k_{ru}, k_{rv})$	Right eye corner in image coordinates
$k_{lr} = (k_l, k_r)$	Left and right eye corners
$\mathbf{s} := (r_e, r_c, k_l, k_r)$	<i>Structure</i> parameters
$p$	Visual target 3D position
$\mathbf{o} = (\phi, \theta)$	Optical axis orientation
$u_e, l_e$	Upper and lower eyelid opening
$\mathbf{m} := (\mathbf{o}, u_e, l_e)$	<i>Movement</i> parameters
$\Lambda_l$	Class $l$ color distribution parameters
$c$	Observed color at pixel $u, v$
$\lambda \in \{0, 1\}$	Occlusion state for pixel $u, v$

The rest of this document is organized as follows. Section 1 describes the used geometric model. Section 2 provides further detail on the parametric segmentation function. Section 3 introduce generalities of Variational Bayes which are necessary to obtain the inference method. Section 4 discusses the update equation for the outliers, which has a different treatment as it is not continuous. Section 5 describes the updates applied to the continuous variables together with the list of gradients used for optimization. Section 6 summarizes the inference algorithm. To conclude, Section 7 discusses implementation issues.

## 1 Eye geometric model

Fig 2 depicts the used geometric model. This is a standard representation of the human eyeball geometry. If all geometric parameters are known then, from the position of  $p$ , we can estimate the necessary eyeball orientation, characterized by the optical axis ( $\mathbf{o}$ ), such that  $\mathbf{v}$  intersects the visual target. This process is denoted as follows:

$$\mathbf{o}(p) = (f_\phi(p; \kappa, d, p_c), f_\theta(p; \kappa, d, p_c)), \quad (1)$$

for which we need to find the functions  $f_\phi$  and  $f_\theta$ . First we define the transformations between the vectorial and angular representation of an “axis” where  $\tau = (\phi_\tau, \theta_\tau)$  represents the angles of the axis (as in Fig. 2b) and  $\tau_v \in \mathbb{R}^3$  is the equivalent 3D vector:

$$\tau_v = \Psi(\tau) = \begin{pmatrix} \cos(\theta_\tau) \sin(\phi_\tau) \\ \sin(\theta_\tau) \\ \cos(\theta_\tau) \cos(\phi_\tau) \end{pmatrix}$$

The inverse transformation (assuming  $\tau_v$  is a unit vector) is given by

$$\tau = \Psi^{-1}(\tau_v) = \begin{pmatrix} \tan^{-1}(\tau_{vx}/\tau_{vz}) \\ \sin^{-1}(\tau_{vy}) \end{pmatrix}$$

Then the problem consist in finding the axis orientation  $\mathbf{o}$  (or eye rotation) such that the following equation is satisfied:

$$p = p_c + d\Psi(\mathbf{o}) + d_p\Psi(\mathbf{o} + \kappa)$$

Where  $d_p$  is the distance between  $\mathbf{N}$  and  $p$ . It can then be shown that the necessary eye rotation is:

$$f_\phi(p; \kappa, d, p_c) = \tan^{-1} \left( \frac{p_x - p_{cx}}{p_z - p_{cz}} \right) - \tan^{-1} \left( \frac{d_p \cos(\theta_\kappa) \sin(\phi_\kappa)}{d + d_p \cos(\theta_\kappa) \cos(\phi_\kappa)} \right) \quad (2)$$

and:

$$f_\theta(p; \kappa, d, p_c) = \sin^{-1} \left( \frac{p_y - p_{cy}}{|p - p_c|} \right) - \sin^{-1} \left( \frac{d_p \sin(\theta_\kappa)}{|d\mathbf{z} + d_p\Psi(\kappa)|} \right) \quad (3)$$

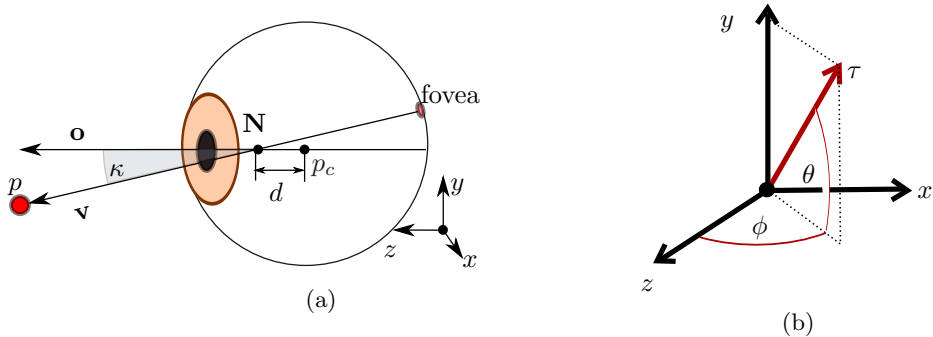


Figure 2: a) Eye geometry with optical ( $\mathbf{o}$ ) and visual ( $\mathbf{v}$ ) axis definition. b) spherical parametrization of an axis “ $\tau$ ”.

For which  $\mathbf{z} := [0, 0, 1]^\top$  and:

$$d_p = \sqrt{d^2 \cos^2(\theta_\kappa) \cos^2(\phi_\kappa) - d^2 + |p - p_c|^2} - d \cos(\theta_\kappa) \cos(\phi_\kappa) \quad (4)$$

## 2 Segmentation function

Here we briefly describe the parametric segmentation derived for our model. It is composed of two main elements: the cornea-sclera segmentation and the skin segmentation.

### 2.1 Cornea-sclera segmentation

Given the parameters  $r_c$ ,  $r_e$ ,  $p_c$  and  $\mathbf{o}$  we need to determine the contour of the cornea which is projected in the  $x, y$  plane and then transformed to image coordinates  $u, v$ . Using a parametric representation of the circumference of the cornea (limbus), where  $t \in [-\pi, \pi]$ , we obtain:

$$\hat{\mathbf{x}}(t) = \begin{bmatrix} r_c \cos(t) \\ r_c \sin(t) \\ \sqrt{r_e^2 - r_c^2} \end{bmatrix}$$

This is valid in the coordinate system located at the eyeball center  $p_c$ . If we apply a rotation of the eyeball given by  $\mathbf{o} = (\phi, \theta)$ , then the parametric contour is given by:

$$\hat{\mathbf{x}}(t; \mathbf{o}) = \begin{bmatrix} \cos(\phi) & -\sin(\theta) \sin(\phi) & \sin(\phi) \cos(\theta) \\ 0 & \cos(\theta) & \sin(\theta) \\ -\sin(\phi) & -\sin(\theta) \cos(\phi) & \cos(\phi) \cos(\theta) \end{bmatrix} \begin{bmatrix} r_c \cos(t) \\ r_c \sin(t) \\ \sqrt{r_e^2 - r_c^2} \end{bmatrix}$$

Expanding and now including the center of the eyeball  $p_c$  as a translation:

$$\hat{\mathbf{x}}(t; \mathbf{o}, p_c) = \begin{bmatrix} r_c \cos(\phi) \cos(t) - r_c \sin(\theta) \sin(\phi) \sin(t) + \sin(\phi) \cos(\theta) \sqrt{r_e^2 - r_c^2} \\ r_c \cos(\theta) \sin(t) + \sqrt{r_e^2 - r_c^2} \sin(\theta) \\ -r_c \sin(\phi) \cos(t) - r_c \sin(\theta) \cos(\phi) \sin(t) + \cos(\phi) \cos(\theta) \sqrt{r_e^2 - r_c^2} \end{bmatrix} + p_c$$

As we are mostly interested in the projection in the  $x, y$  plane, the final contour is given as:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} r_c \cos(\phi) \cos(t) - r_c \sin(\theta) \sin(\phi) \sin(t) + \sin(\phi) \cos(\theta) \sqrt{r_e^2 - r_c^2} + p_{cx} \\ r_c \cos(\theta) \sin(t) + \sqrt{r_e^2 - r_c^2} \sin(\theta) + p_{cy} \end{bmatrix} \quad (5)$$

Transformation to image coordinates is straightforward: the projection to the pose-rectified eye images is orthogonal and the pixel size is given (due to the available depth data and camera calibration).

### 2.2 Eyelids segmentation

The upper and lower eyelids are defined as quadratic Bezier curves. For example, the upper eyelid has as control points  $k_l$ ,  $\mathbf{c}_u$  and  $k_r$ , i.e. the eye corners and the middle point  $\mathbf{c}_u$ . We defined  $\mathbf{c}_u := [(k_{lu} + k_{ru})/2, u_e]$ , where  $u_e$  define the vertical position of the control point and thus, the eyelid opening. The lower eyelid is defined in the same manner, but using  $l_e$  to define the eyelid opening (vertical).

### 2.3 Final segmentation

A segmentation is therefore defined by a set of geometric parameters:  $r_c$ ,  $r_e$ ,  $p_c$ ,  $k_l$ ,  $k_r$ ,  $u_e$ ,  $l_e$  and  $\mathbf{o}$ . Using Eq. 5 we can query if a pixel  $u, v$  is inside the contour by simple comparisons. Similarly we can query if the pixel  $u, v$  is or not within the skin region by comparisons to the eyelids Bezier curves. Notice the skin class overrides the sclera and cornea classes, as the skin occludes these regions. This set of rules thus define the segmentation function  $S_{u,v}$ .

### 3 Variational Bayes

For an in-depth discussion of Variational Bayes (VB), please refer to [1], in this section we summarize the main generalities relevant to our method. In VB the posterior  $p(\mathbf{Z}|\mathbf{X})$  is approximated using the proposal distribution  $q(\mathbf{Z})$ . This approximation leads to the known relation:

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \text{KL}(q||p) \quad (6)$$

Where  $\text{KL}(q||p)$  is the Kullback-Leibler (KL) divergence between  $p(\mathbf{Z}|\mathbf{X})$  and  $q(\mathbf{Z})$ , and  $\ln p(\mathbf{X})$  is the log-marginal likelihood, which is a constant quantity under a fix model. In general, the *variational lower bound*  $\mathcal{L}(q)$  takes the following form:

$$\begin{aligned} \mathcal{L}(q) &= \int q(\mathbf{Z}) \ln p(\mathbf{X}, \mathbf{Z}) d\mathbf{Z} - \int \ln q(\mathbf{Z}) q(\mathbf{Z}) d\mathbf{Z} \\ &= \mathbb{E}_{\mathbf{Z}} [\ln p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_{\mathbf{Z}} [\ln q(\mathbf{Z})] \end{aligned} \quad (7)$$

Where  $p(\mathbf{X}, \mathbf{Z})$  is the complete joint distribution and expectations are defined with respect to  $q(\mathbf{Z})$ . The goal is to maximize the functional  $\mathcal{L}$  with respect to  $q(\mathbf{Z})$  as, due to  $\ln p(\mathbf{X})$  being constant, this is equivalent to minimize the KL divergence between  $p(\mathbf{Z}|\mathbf{X})$  and  $q(\mathbf{Z})$ . Therefore the optimal  $q(\mathbf{Z})$  becomes a tractable substitute for the posterior.

Assuming we have a training set  $\{(\mathbf{I}^i, p^i)\}_{i=1}^N$  of  $N$  pairs of eye images  $\mathbf{I}$  and visual targets  $p$ , indexed by  $i$ , the functional form of  $q(\mathbf{Z})$  is given by:

$$\begin{aligned} q(\mathbf{Z}) &= \mathcal{N}(\mu_d, \sigma_d) \mathcal{N}(\mu_{\phi_\kappa}, \sigma_{\phi_\kappa}) \mathcal{N}(\mu_{\theta_\kappa}, \sigma_{\theta_\kappa}) \mathcal{N}(\mu_{r_e}, \sigma_{r_e}) \mathcal{N}(\mu_{r_c}, \sigma_{r_c}) \mathcal{N}(\mu_{p_{cx}}, \sigma_{p_{cx}}) \\ &\quad \mathcal{N}(\mu_{p_{cy}}, \sigma_{p_{cy}}) \mathcal{N}(\mu_{p_{cz}}, \sigma_{p_{cz}}) \mathcal{N}(\mu_{k_{lu}}, \sigma_{k_{lu}}) \mathcal{N}(\mu_{k_{lv}}, \sigma_{k_{lv}}) \mathcal{N}(\mu_{k_{ru}}, \sigma_{k_{ru}}) \mathcal{N}(\mu_{k_{rv}}, \sigma_{k_{rv}}) \\ &\quad \prod_{i=1}^N [\mathcal{N}(\mu_{\phi}^i, \sigma_{\phi}^i) \mathcal{N}(\mu_{\theta}^i, \sigma_{\theta}^i) \mathcal{N}(\mu_{u_e}^i, \sigma_{u_e}^i) \mathcal{N}(\mu_{l_e}^i, \sigma_{l_e}^i) \prod_{u,v \in i} q(\lambda_i^{u,v})] \end{aligned} \quad (8)$$

Where  $\mathcal{N}(\mu_a, \sigma_a)$  is a simplification for the univariate normal distribution  $\mathcal{N}(a|\mu_a, \sigma_a)$ . Notice that  $q(\mathbf{Z})$  can be seen as a fully factorized distribution where  $q(\mathbf{Z}) = \prod_i q(\mathbf{Z}_i)$ . If we allow variations of  $\mathcal{L}$  around a single factor  $q_j = q(\mathbf{Z}_j)$  then Eq. 7 is equivalent to the following expression:

$$\mathcal{L}(q) = \mathbb{E}_{\mathbf{Z}_j} [\mathbb{E}_{\mathbf{Z}_{i \neq j}} [\ln p_j(\mathbf{X}, \mathbf{Z})]] - \mathbb{E}_{\mathbf{Z}_j} [\ln q_j(\mathbf{Z}_j)] + cst_j \quad (9)$$

Where  $cst_j$  is a constant with respect to  $\mathbf{Z}_j$ , and thus constant for any change of  $q_j$ . Notice that computations related to  $q_j$  in Eq. 9, ignoring  $cst_j$ , need only the additive terms from  $\ln p(\mathbf{X}, \mathbf{Z})$  which include the variable  $\mathbf{Z}_j$ . Here we denote these terms as  $\ln p_j(\mathbf{X}, \mathbf{Z})$ .

In standard VB,  $\mathcal{L}$  can be maximized in an iterative fashion, updating a factor per iteration until global convergence. If  $\mathcal{L}$  is optimized with respect to the factor  $q_j$ , while keeping the rest of factors unchanged, the optimal distribution for the factor  $\mathbf{Z}_j$  is:

$$\ln q_j^*(\mathbf{Z}_j) = \mathbb{E}_{\mathbf{Z}_{i \neq j}} [\ln p_j(\mathbf{Z}, \mathbf{X})] + cst_j \quad (10)$$

Depending on the definition of the model conditionals and priors Eq. 10 normally can be solved analytically. This is not the case for the continuous variables of our model due to the complex relations in  $f_\phi$ ,  $f_\theta$  and  $S_{u,v}$ . To solve this aspect we imposed a parametric form for the continuous factors: a univariate Gaussian. This leads to an optimization of  $\mathcal{L}$  with respect to the Gaussian parameters.

Given the -univariate- Gaussian factor  $q(\mathbf{Z}_j) = \mathcal{N}(\mu_{\mathbf{Z}_j}, \sigma_{\mathbf{Z}_j})$  we followed the approach described in [2] where the derivatives can be computed as shown in Eq. 11 and Eq. 12.

$$\frac{\partial \mathcal{L}}{\partial \mu_{\mathbf{Z}_j}} = \mathbb{E}_{\mathbf{Z}} \left[ \frac{(\mathbf{Z}_j - \mu_{\mathbf{Z}_j})}{\sigma_{\mathbf{Z}_j}^2} \ln p_j(\mathbf{Z}, \mathbf{X}) \right] \quad (11)$$



$$\frac{\partial \mathcal{L}}{\partial \sigma_{\mathbf{Z}_j}} = \mathbb{E}_{\mathbf{Z}} \left[ \frac{((\mathbf{Z}_j - \mu_{\mathbf{Z}_j})^2 - \sigma_{\mathbf{Z}_j}^2)}{\sigma_{\mathbf{Z}_j}^3} \ln p_j(\mathbf{Z}, \mathbf{X}) \right] + \frac{1}{\sigma_{\mathbf{Z}_j}} \quad (12)$$

## 4 Outliers term

To obtain the outliers update we first identify the additive terms of  $\ln p(\mathbf{X}, \mathbf{Z})$  which include  $\lambda_i^{u,v}$  (i.e. the Bernoulli prior for  $\lambda$  and the image likelihood) and letting any other term being absorbed by  $cst$ :

$$\ln p_{\lambda_i^{u,v}}(\mathbf{Z}, \mathbf{X}) = \lambda_i^{u,v} (\ln \hat{\lambda} + \ln \epsilon) + (1 - \lambda_i^{u,v}) \left( \ln(1 - \hat{\lambda}) + \sum_l S_{u,v}(l; \mathbf{m}, \mathbf{s}, p_c) \ln p(c^{u,v} | \Lambda_l) \right) + cst \quad (13)$$

The optimal distribution for  $\lambda_i^{u,v}$  can be obtained by solving Eq. 10:

$$\begin{aligned} \ln q^*(\lambda_i^{u,v}) &= \mathbb{E}_{\mathbf{Z} \neq \lambda_i^{u,v}} \left[ \lambda_i^{u,v} (\ln \hat{\lambda} + \ln \epsilon) + (1 - \lambda_i^{u,v}) \left( \ln(1 - \hat{\lambda}) + \sum_l S_{u,v}(l; \mathbf{m}, \mathbf{s}, p_c) \ln p(c^{u,v} | \Lambda_l) \right) \right] + cst \\ &= \lambda_i^{u,v} (\ln \hat{\lambda} + \ln \epsilon) + (1 - \lambda_i^{u,v}) \left( \ln(1 - \hat{\lambda}) + \sum_l \mathbb{E}_{\mathbf{m}, \mathbf{s}, p_c} [S_{u,v}(l; \mathbf{m}, \mathbf{s}, p_c)] \ln p(c_{x,y} | \Lambda_l) \right) + cst \end{aligned} \quad (14)$$

By inspection it can be shown that Eq. 14 is equivalent to the following Bernoulli distribution:

$$q^*(\lambda_i^{u,v}) = (\omega_i^{u,v})^{\lambda_i^{u,v}} (1 - \omega_i^{u,v})^{(1 - \lambda_i^{u,v})} \quad (15)$$

Where

$$\omega_i^{u,v} = \frac{\hat{\lambda}}{(1 - \hat{\lambda})^{\frac{1}{\epsilon}} \prod_l p(c | \Lambda_l)^{\mathbb{E}_{\mathbf{m}, \mathbf{s}, p_c} [S_{u,v}(l; \mathbf{m}, \mathbf{s}, p_c)]} + \hat{\lambda}} \quad (16)$$

Notice that  $\mathbb{E}_{\mathbf{m}, \mathbf{s}, p_c} [S_{u,v}(l; \mathbf{m}, \mathbf{s}, p_c)]$  is an expectation with respect to  $q(\mathbf{m}_i)q(\mathbf{s})q(p_c)$ , which we compute using a Monte Carlo approximation.

## 5 Gaussian derivatives

In this section we list the derivatives for our model, obtained by solving Eq. 11 and Eq. 12 for each parameter.

### 5.1 Common expressions

We will find useful to define expressions which are derived from log of the conditionals related to the visual axis gazing action and image likelihood for a given sample  $i$ :

$$g_i(\mathbf{a}, p_c) := -\frac{1}{2\hat{\sigma}_o^2} \left( (f_\phi(p^i; p_c, d, \kappa) - \mu_\phi^i)^2 + (f_\theta(p^i; p_c, d, \kappa) - \mu_\theta^i)^2 \right) \quad (17)$$

$$h_i(\mathbf{m}, \mathbf{s}, p_c) := \sum_{u,v} (1 - \omega_i^{u,v}) \sum_{l=1}^3 S_{u,v}(l; \mathbf{m}, \mathbf{s}, p_c) \ln p(c_i^{u,v} | \Lambda_l) \quad (18)$$

For a random variable  $a$  with a prior Gaussian distribution  $\mathcal{N}(\hat{\mu}_a, \hat{\sigma}_a)$  and an associated proposal distribution  $q(a) = \mathcal{N}(\mu_a, \sigma_a)$  it can be shown that:

$$\mathbb{E}_a [\ln \mathcal{N}(\hat{\mu}_a, \hat{\sigma}_a)] = -\frac{(\mu_a^2 + \sigma_a^2 - 2\mu_a \hat{\mu}_a)}{2\hat{\sigma}_a^2} + cst \quad (19)$$

$$\mathbb{E}_a \left[ \frac{(a - \mu_a)}{\sigma_a^2} \ln \mathcal{N}(\hat{\mu}_a, \hat{\sigma}_a) \right] = -\frac{(\mu_a - \hat{\mu}_a)}{\hat{\sigma}_a^2} \quad (20)$$

$$\mathbb{E}_a \left[ \frac{\left( (a - \mu_a)^2 - \sigma_a^2 \right)}{\sigma_a^3} \ln \mathcal{N}(\hat{\mu}_a, \hat{\sigma}_a) \right] = -\frac{\sigma_a}{\hat{\sigma}_a^2} \quad (21)$$

For the special case of the term which correlates the eyelid opening with the eye elevation:

$$\mathbb{E}_{\theta^i, u_e^i} [\ln \mathcal{N}(a_u \theta^i + b_u, \hat{\sigma}_{u_e})] = -\frac{\mu_{u_e}^2 + \sigma_{u_e}^2 - 2a_u \mu_{u_e}^i \mu_{\theta}^i - 2b_u \mu_{u_e}^i + a_u^2 \mu_{\theta}^i{}^2 + a_u^2 \sigma_{\theta}^i{}^2 + 2a_u b_u \mu_{\theta}^i + b_u^2}{2\hat{\sigma}_{u_e}^2} \quad (22)$$

As described in the main paper, all parameters are divided in sub-groups. The overall optimization is done by iteratively optimizing over these sub-groups until global convergence. In the following we describe each of the groups with their corresponding bound and set of derivatives for the optimization. In each case, a gradient ascent method is used to find the optimal values.

## 5.2 Eye corners and eyelids opening

This group is optimized for the eyelids corners and the eyelids opening for all samples jointly, such that we optimize for the group of random variables:  $\mathbf{e} := (k_{lu}, k_{lv}, k_{ru}, k_{rv}, l_e^1, u_e^1, \dots, l_e^N, u_e^N)$ .

To this end we identify the additives terms of  $\ln p(\mathbf{X}, \mathbf{Z})$  which include  $\mathbf{e}$ , as in Eq. 9. In this way we obtain the variational lower bound in function of only the terms related to  $\mathbf{e}$  as follows:

$$\begin{aligned} \mathcal{L}(q) \approx & \mathbb{E}_{k_{lu}} [\ln \mathcal{N}(\hat{\mu}_{k_{lu}}, \hat{\sigma}_{k_{lu}})] + \mathbb{E}_{k_{lv}} [\ln \mathcal{N}(\hat{\mu}_{k_{lv}}, \hat{\sigma}_{k_{lv}})] + \mathbb{E}_{k_{ru}} [\ln \mathcal{N}(\hat{\mu}_{k_{ru}}, \hat{\sigma}_{k_{ru}})] + \mathbb{E}_{k_{rv}} [\ln \mathcal{N}(\hat{\mu}_{k_{rv}}, \hat{\sigma}_{k_{rv}})] + \\ & \ln \sigma_{k_{lu}} + \ln \sigma_{k_{lv}} + \ln \sigma_{k_{ru}} + \ln \sigma_{k_{rv}} + \sum_i \left[ \mathbb{E}_{l_e^i} [\ln \mathcal{N}(\hat{\mu}_{l_e}, \hat{\sigma}_{l_e})] + \mathbb{E}_{\theta^i, u_e^i} [\ln \mathcal{N}(a_u \theta^i + b_u, \hat{\sigma}_{u_e})] + \right. \\ & \left. \ln \sigma_{u_e}^i + \ln \sigma_{l_e}^i + \frac{1}{M} \sum_{k=1}^M h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k) \right] + K_{\mathbf{e}} \end{aligned} \quad (23)$$

Where  $K_{\mathbf{e}}$  is a constant. We can refer to Eq. 19 to solve the expectations for the terms related to the priors. In the case of the Jacobian, we develop Eq. 11 and Eq. 12, and taking into account the results shown in Eq. 20 and Eq. 21, we will define the eye corners Jacobian  $\mathbf{J}_{\mathbf{e}\mathbf{k}}$  and the -per sample  $i$ - eyelids opening Jacobian as  $\mathbf{J}_{\mathbf{e}}^i$ . These terms are computed as shown in Eq. 24 and Eq. 25:

$$\mathbf{J}_{\mathbf{e}\mathbf{k}} := \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \mu_{k_{lu}}} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{k_{lu}}} \\ \frac{\partial \mathcal{L}}{\partial \mu_{k_{lv}}} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{k_{lv}}} \\ \frac{\partial \mathcal{L}}{\partial \mu_{k_{ru}}} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{k_{ru}}} \\ \frac{\partial \mathcal{L}}{\partial \mu_{k_{rv}}} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{k_{rv}}} \end{bmatrix} \approx - \begin{bmatrix} \frac{(\mu_{k_{lu}} - \hat{\mu}_{k_{lu}})}{\hat{\sigma}_{k_{lu}}^2} \\ \frac{\sigma_{k_{lu}}}{\hat{\sigma}_{k_{lu}}^2} - \frac{1}{\sigma_{k_{lu}}} \\ \frac{(\mu_{k_{lv}} - \hat{\mu}_{k_{lv}})}{\hat{\sigma}_{k_{lv}}^2} \\ \frac{\sigma_{k_{lv}}}{\hat{\sigma}_{k_{lv}}^2} - \frac{1}{\sigma_{k_{lv}}} \\ \frac{(\mu_{k_{ru}} - \hat{\mu}_{k_{ru}})}{\hat{\sigma}_{k_{ru}}^2} \\ \frac{\sigma_{k_{ru}}}{\hat{\sigma}_{k_{ru}}^2} - \frac{1}{\sigma_{k_{ru}}} \\ \frac{(\mu_{k_{rv}} - \hat{\mu}_{k_{rv}})}{\hat{\sigma}_{k_{rv}}^2} \\ \frac{\sigma_{k_{rv}}}{\hat{\sigma}_{k_{rv}}^2} - \frac{1}{\sigma_{k_{rv}}} \end{bmatrix} + \frac{1}{M} \sum_{i=1}^N \sum_{k=1}^M \begin{bmatrix} \frac{(\mu_{k_{lu}}^k - \mu_{k_{lu}})}{\sigma_{k_{lu}}^2} \\ \frac{((k_{lu}^k - \mu_{k_{lu}})^2 - \sigma_{k_{lu}}^2)}{\sigma_{k_{lu}}^3} \\ \frac{(\mu_{k_{lv}}^k - \mu_{k_{lv}})}{\sigma_{k_{lv}}^2} \\ \frac{((k_{lv}^k - \mu_{k_{lv}})^2 - \sigma_{k_{lv}}^2)}{\sigma_{k_{lv}}^3} \\ \frac{(\mu_{k_{ru}}^k - \mu_{k_{ru}})}{\sigma_{k_{ru}}^2} \\ \frac{((k_{ru}^k - \mu_{k_{ru}})^2 - \sigma_{k_{ru}}^2)}{\sigma_{k_{ru}}^3} \\ \frac{(\mu_{k_{rv}}^k - \mu_{k_{rv}})}{\sigma_{k_{rv}}^2} \\ \frac{((k_{rv}^k - \mu_{k_{rv}})^2 - \sigma_{k_{rv}}^2)}{\sigma_{k_{rv}}^3} \end{bmatrix} h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k) \quad (24)$$

And the Jacobian related to the eyelids opening of a sample  $i$ ,  $\mathbf{J}_{\mathbf{e}}^i$  is given as:

$$\mathbf{J}_e^i := \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \mu_{u_e}^i} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{u_e}^i} \\ \frac{\partial \mathcal{L}}{\partial \mu_{l_e}^i} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{l_e}^i} \end{bmatrix} \approx - \begin{bmatrix} \frac{(\mu_{u_e}^i - (a_u \mu_{\theta}^i + b_u))}{\hat{\sigma}_{u_e}^2} \\ \frac{\sigma_{u_e}^i}{\hat{\sigma}_{u_e}^2} - \frac{1}{\sigma_{u_e}^i} \\ \frac{(\mu_{l_e}^i - \hat{\mu}_{l_e})}{\hat{\sigma}_{l_e}^2} \\ \frac{\sigma_{l_e}^i}{\hat{\sigma}_{l_e}^2} - \frac{1}{\sigma_{l_e}^i} \end{bmatrix} + \frac{1}{M} \sum_{k=1}^M \begin{bmatrix} \frac{(u_e^k - \mu_{u_e}^i)}{\sigma_{u_e}^i{}^2} \\ \frac{((u_e^k - \mu_{u_e}^i)^2 - \sigma_{u_e}^i{}^2)}{\sigma_{u_e}^i{}^3} \\ \frac{(l_e^k - \mu_{l_e}^i)}{\sigma_{l_e}^i{}^2} \\ \frac{((l_e^k - \mu_{l_e}^i)^2 - \sigma_{l_e}^i{}^2)}{\sigma_{l_e}^i{}^3} \end{bmatrix} h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k) \quad (25)$$

Which relies on the result in Eq. 22. Finally we can define the group Jacobian as  $\mathbf{J}_e = [\mathbf{J}_{e\mathbf{k}}^\top, \mathbf{J}_e^{1\top}, \dots, \mathbf{J}_e^{N\top}]^\top$ , used to optimize the eyelids corners and opening for all samples jointly.

Notice the Monte Carlo expectations are based on drawing  $M$  samples  $\{(\mathbf{m}^k, \mathbf{s}^k, p_c^k)\}_{k=1}^M$  from the current distribution  $q(\mathbf{s})q(p_c)q(\mathbf{m}^i)$  (this is done per sample  $i$ ) and notice that  $\mathbf{m}^k := (\phi^k, \theta^k, u_e^k, l_e^k)$ .

### 5.3 Eyeball geometry and orientation

The eyeball geometry and orientation are optimized jointly, i.e. with respect to the random variables  $\mathbf{g} := (p_{cx}, p_{cy}, r_e, r_c, \phi^1, \theta^1, \dots, \phi^N, \theta^N)$ . To this end we define the related variational lower bound by using only the additive terms of  $\ln p(\mathbf{X}, \mathbf{Z})$  which include  $\mathbf{g}$ , leading to:

$$\begin{aligned} \mathcal{L}(q) \approx & \mathbb{E}_{p_{cx}} [\ln \mathcal{N}(\hat{\mu}_{p_{cx}}, \hat{\sigma}_{p_{cx}})] + \mathbb{E}_{p_{cy}} [\ln \mathcal{N}(\hat{\mu}_{p_{cy}}, \hat{\sigma}_{p_{cy}})] + \mathbb{E}_{r_e} [\ln \mathcal{N}(\hat{\mu}_{r_e}, \hat{\sigma}_{r_e})] + \mathbb{E}_{r_c} [\ln \mathcal{N}(\hat{\mu}_{r_c}, \hat{\sigma}_{r_c})] + \\ & \ln \sigma_{p_{cx}} + \ln \sigma_{p_{cy}} + \ln \sigma_{r_e} + \ln \sigma_{r_c} + \sum_i \left[ \mathbb{E}_{\phi^i} [\ln \mathcal{N}(\hat{\mu}_{\phi^i}, \hat{\sigma}_{\phi^i})] + \mathbb{E}_{\theta^i} [\ln \mathcal{N}(\hat{\mu}_{\theta^i}, \hat{\sigma}_{\theta^i})] + \right. \\ & \left. \mathbb{E}_{\theta^i, u_e^i} [\ln \mathcal{N}(a_u \theta^i + b_u, \hat{\sigma}_{u_e})] + \ln \sigma_{\phi^i} + \ln \sigma_{\theta^i} + \frac{1}{M} \sum_{k=1}^M (g_i(\mathbf{a}^k, p_c^k) + h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k)) \right] + K_{\mathbf{g}} \quad (26) \end{aligned}$$

For which we have defined the expected eye orientation conditioned on the position of the visual target  $p^i$ :

$$\hat{\mu}_{\phi^i} = \mathbb{E}_{\mathbf{a}, p_c} [f_{\phi}(p^i, \mathbf{a}, p_c)] \approx \frac{1}{M} \sum_{k=1}^M f_{\phi}(p^i; \mathbf{a}^k, p_c^k) \quad (27)$$

$$\hat{\mu}_{\theta^i} = \mathbb{E}_{\mathbf{a}, p_c} [f_{\theta}(p^i, \mathbf{a}, p_c)] \approx \frac{1}{M} \sum_{k=1}^M f_{\theta}(p^i; \mathbf{a}^k, p_c^k) \quad (28)$$

Notice that Eq. 27 and Eq. 28 are key elements into the well constrained global optimization, as these values put constraints on the eye orientation conditioned by the visual target position, while comparing to image data. These quantities assume drawing a set of samples  $\{(\mathbf{a}^k, p_c^k)\}_{k=1}^M$  from  $q(\mathbf{a}, p_c)$ .

In the case of the eyeball location, in terms of the x and y coordinates, it takes information from both image and target data. We thus define the corresponding Jacobian  $\mathbf{J}_{p_{cxy}}$  as follows:

$$\mathbf{J}_{p_{cxy}} := \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \mu_{p_{cx}}^i} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{p_{cx}}^i} \\ \frac{\partial \mathcal{L}}{\partial \mu_{p_{cy}}^i} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{p_{cy}}^i} \end{bmatrix} \approx - \begin{bmatrix} \frac{(\mu_{p_{cx}}^i - \hat{\mu}_{p_{cx}})}{\hat{\sigma}_{p_{cx}}^2} \\ \frac{\sigma_{p_{cx}}^i}{\hat{\sigma}_{p_{cx}}^2} - \frac{1}{\sigma_{p_{cx}}^i} \\ \frac{(\mu_{p_{cy}}^i - \hat{\mu}_{p_{cy}})}{\hat{\sigma}_{p_{cy}}^2} \\ \frac{\sigma_{p_{cy}}^i}{\hat{\sigma}_{p_{cy}}^2} - \frac{1}{\sigma_{p_{cy}}^i} \end{bmatrix} + \frac{1}{M} \sum_{i=1}^N \sum_{k=1}^M \begin{bmatrix} \frac{(p_{cx}^k - \mu_{p_{cx}})}{\sigma_{p_{cx}}^i{}^2} \\ \frac{((p_{cx}^k - \mu_{p_{cx}})^2 - \sigma_{p_{cx}}^i{}^2)}{\sigma_{p_{cx}}^i{}^3} \\ \frac{(p_{cy}^k - \mu_{p_{cy}})}{\sigma_{p_{cy}}^i{}^2} \\ \frac{((p_{cy}^k - \mu_{p_{cy}})^2 - \sigma_{p_{cy}}^i{}^2)}{\sigma_{p_{cy}}^i{}^3} \end{bmatrix} (g_i(\mathbf{a}^k, p_c^k) + h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k)) \quad (29)$$

We define the eyeball radii Jacobian  $\mathbf{J}_r$  as follows:

$$\mathbf{J}_{\mathbf{r}} := \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \mu_{r_e}} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{r_e}} \\ \frac{\partial \mathcal{L}}{\partial \mu_{r_c}} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{r_c}} \end{bmatrix} \approx - \begin{bmatrix} \frac{(\mu_{r_e} - \hat{\mu}_{r_e})}{\hat{\sigma}_{r_e}^2} \\ \frac{\sigma_{r_e}}{\hat{\sigma}_{r_e}^2} - \frac{1}{\sigma_{r_e}} \\ \frac{(\mu_{r_c} - \hat{\mu}_{r_c})}{\hat{\sigma}_{r_c}^2} \\ \frac{\sigma_{r_c}}{\hat{\sigma}_{r_c}^2} - \frac{1}{\sigma_{r_c}} \end{bmatrix} + \frac{1}{M} \sum_{i=1}^N \sum_{k=1}^M \begin{bmatrix} \frac{(\mu_{r_e}^k - \mu_{r_e})}{\sigma_{r_e}^2} \\ \frac{((\mu_{r_e}^k - \mu_{r_e})^2 - \sigma_{r_e}^2)}{\sigma_{r_e}^3} \\ \frac{(\mu_{r_c}^k - \mu_{r_c})}{\sigma_{r_c}^2} \\ \frac{((\mu_{r_c}^k - \mu_{r_c})^2 - \sigma_{r_c}^2)}{\sigma_{r_c}^3} \end{bmatrix} h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k) \quad (30)$$

Now, provided the result in Eq. 22, we define the per sample orientation Jacobian  $\mathbf{J}_{\mathbf{o}}^i$  as:

$$\mathbf{J}_{\mathbf{o}}^i := \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \mu_{\phi}^i} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{\phi}^i} \\ \frac{\partial \mathcal{L}}{\partial \mu_{\theta}^i} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{\theta}^i} \end{bmatrix} \approx - \begin{bmatrix} \frac{(\mu_{\phi}^i - \hat{\mu}_{\phi}^i)}{\hat{\sigma}_{\phi}^2} \\ \frac{\sigma_{\phi}^i}{\hat{\sigma}_{\phi}^2} - \frac{1}{\sigma_{\phi}^i} \\ \frac{(\mu_{\theta}^i - \hat{\mu}_{\theta}^i)}{\hat{\sigma}_{\theta}^2} - \frac{a_u(\mu_{u_e}^i - (a_u \mu_{\theta}^i + b_u))}{\hat{\sigma}_{u_e}^2} \\ \frac{\sigma_{\theta}^i}{\hat{\sigma}_{\theta}^2} - \frac{1}{\sigma_{\theta}^i} + \frac{a_u \sigma_{\theta}^i}{\hat{\sigma}_{u_e}^2} \end{bmatrix} + \frac{1}{M} \sum_{k=1}^M \begin{bmatrix} \frac{(\phi^k - \mu_{\phi}^i)}{\sigma_{\phi}^{i^2}} \\ \frac{((\phi^k - \mu_{\phi}^i)^2 - \sigma_{\phi}^{i^2})}{\sigma_{\phi}^{i^3}} \\ \frac{(\theta^k - \mu_{\theta}^i)}{\sigma_{\theta}^{i^2}} \\ \frac{((\theta^k - \mu_{\theta}^i)^2 - \sigma_{\theta}^{i^2})}{\sigma_{\theta}^{i^3}} \end{bmatrix} h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k), \quad (31)$$

where to evaluate the Monte Carlo expectations involved in the previous definitions we have drawn  $M$  samples  $\{(\mathbf{m}^k, \mathbf{s}^k, \mathbf{a}^k, p_c^k)\}_{k=1}^M$  from the current distribution  $q(\mathbf{m}^i)q(\mathbf{s})q(\mathbf{a})q(p_c)$ , i.e.  $M$  samples per training sample  $i$ .

We finally define the group Jacobian as  $\mathbf{J}_{\mathbf{g}} = [\mathbf{J}_{p_{cxy}}^\top, \mathbf{J}_{\mathbf{r}}^\top, \mathbf{J}_{\mathbf{o}}^{1\top}, \dots, \mathbf{J}_{\mathbf{o}}^{N\top}]^\top$  which is used to optimize jointly the eyeball geometry and the eye orientation per sample.

## 5.4 Axial and eyeball depth parameters

In this section we define the derivatives for the joint optimization of the axial parameters  $\mathbf{a} := (d, \phi_{\kappa}, \theta_{\kappa})$  and the eyeball depth  $p_{cz}$ . We will refer this group of variables as  $\mathbf{A} := (d, \phi_{\kappa}, \theta_{\kappa}, p_{cz})$ .

We thus identify the additives terms of  $\ln p(\mathbf{X}, \mathbf{Z})$  including  $\mathbf{A}$ , as in Eq. 9, we obtain the variational lower bound in function of only the terms related to  $\mathbf{A}$ :

$$\begin{aligned} \mathcal{L}(q) &\approx \mathbb{E}_{\phi_{\kappa}} [\ln \mathcal{N}(\hat{\mu}_{\phi_{\kappa}}, \hat{\sigma}_{\phi_{\kappa}})] + \mathbb{E}_{\theta_{\kappa}} [\ln \mathcal{N}(\hat{\mu}_{\theta_{\kappa}}, \hat{\sigma}_{\theta_{\kappa}})] + \mathbb{E}_d [\ln \mathcal{N}(\hat{\mu}_d, \hat{\sigma}_d)] + \mathbb{E}_{p_{cz}} [\ln \mathcal{N}(\hat{\mu}_{p_{cz}}, \hat{\sigma}_{p_{cz}})] + \\ &+ \ln \sigma_{\phi_{\kappa}} + \ln \sigma_{\theta_{\kappa}} + \ln \sigma_d + \ln \sigma_{p_{cz}} + \frac{1}{M} \sum_{i=1}^N \sum_{k=1}^M g_i(\mathbf{a}^k, p_c^k) + K_{\mathbf{A}} \end{aligned} \quad (32)$$

Where  $K_{\mathbf{A}}$  is a constant as it contains the terms not related to  $\mathbf{A}$ . We can refer to Eq. 19 to obtain the expectations for the terms related to the priors. In the case of the Jacobian, we develop Eq. 11 and Eq. 12, and taking into account the results shown in Eq. 20 and Eq. 21, we obtain  $\mathbf{J}_{\mathbf{a}}$  as follows:

$$\mathbf{J}_{\mathbf{a}} := \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \mu_d} \\ \frac{\partial \mathcal{L}}{\partial \sigma_d} \\ \frac{\partial \mathcal{L}}{\partial \mu_{\phi_{\kappa}}} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{\phi_{\kappa}}} \\ \frac{\partial \mathcal{L}}{\partial \mu_{\theta_{\kappa}}} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{\theta_{\kappa}}} \\ \frac{\partial \mathcal{L}}{\partial \mu_{p_{cz}}} \\ \frac{\partial \mathcal{L}}{\partial \sigma_{p_{cz}}} \end{bmatrix} \approx - \begin{bmatrix} \frac{(\mu_d - \hat{\mu}_d)}{\hat{\sigma}_d^2} \\ \frac{\sigma_d}{\hat{\sigma}_d^2} - \frac{1}{\sigma_d} \\ \frac{(\mu_{\phi_{\kappa}} - \hat{\mu}_{\phi_{\kappa}})}{\hat{\sigma}_{\phi_{\kappa}}^2} \\ \frac{\sigma_{\phi_{\kappa}}}{\hat{\sigma}_{\phi_{\kappa}}^2} - \frac{1}{\sigma_{\phi_{\kappa}}} \\ \frac{(\mu_{\theta_{\kappa}} - \hat{\mu}_{\theta_{\kappa}})}{\hat{\sigma}_{\theta_{\kappa}}^2} \\ \frac{\sigma_{\theta_{\kappa}}}{\hat{\sigma}_{\theta_{\kappa}}^2} - \frac{1}{\sigma_{\theta_{\kappa}}} \\ \frac{(\mu_{p_{cz}} - \hat{\mu}_{p_{cz}})}{\hat{\sigma}_{p_{cz}}^2} \\ \frac{\sigma_{p_{cz}}}{\hat{\sigma}_{p_{cz}}^2} - \frac{1}{\sigma_{p_{cz}}} \end{bmatrix} + \frac{1}{M} \sum_{i=1}^N \sum_{k=1}^M \begin{bmatrix} \frac{(d^k - \mu_d)}{\sigma_d^2} \\ \frac{((d^k - \mu_d)^2 - \sigma_d^2)}{\sigma_d^3} \\ \frac{(\phi_{\kappa}^k - \mu_{\phi_{\kappa}})}{\sigma_{\phi_{\kappa}}^2} \\ \frac{((\phi_{\kappa}^k - \mu_{\phi_{\kappa}})^2 - \sigma_{\phi_{\kappa}}^2)}{\sigma_{\phi_{\kappa}}^3} \\ \frac{(\theta_{\kappa}^k - \mu_{\theta_{\kappa}})}{\sigma_{\theta_{\kappa}}^2} \\ \frac{((\theta_{\kappa}^k - \mu_{\theta_{\kappa}})^2 - \sigma_{\theta_{\kappa}}^2)}{\sigma_{\theta_{\kappa}}^3} \\ \frac{(p_{cz}^k - \mu_{p_{cz}})}{\sigma_{p_{cz}}^2} \\ \frac{((p_{cz}^k - \mu_{p_{cz}})^2 - \sigma_{p_{cz}}^2)}{\sigma_{p_{cz}}^3} \end{bmatrix} g_i(\mathbf{a}^k, p_c^k) \quad (33)$$



Where in order to evaluate Eq. 32 and Eq. 33 we have drawn  $M$  samples  $\{(\mathbf{a}^k, p_c^k)\}_{k=1}^M$  from the current distribution  $q(\mathbf{a})q(p_c)$  to use a Monte Carlo expectation approximation.

## 6 Inference algorithm

As mentioned in the paper, the variational lower bound is optimized in an iterative manner. Each group of latent variables is optimized assuming the rest is fixed. This produces the following inference algorithm for the training stage:

---

**Algorithm 1** Geometric generative gaze model inference.

---

Set initial  $q(\mathbf{Z})$  from the prior distribution parameters.

**repeat**

- Optimize  $\mathcal{L}$  w.r.t. eye corners and all eyelids opening (c.f. Section 5.2)

$$q(k_{lu})q(k_{lv})q(k_{ru})q(k_{rv}) \prod_I q(u_e^I)q(l_e^I)$$

- Optimize  $\mathcal{L}$  w.r.t. eyeball geometry and orientation (c.f. Section 5.3)

$$q(r_e)q(r_i)q(p_{cx})q(p_{cy}) \prod_I q(\mathbf{o}^I)$$

- Optimize  $\mathcal{L}$  w.r.t. axial parameters and eyeball depth (c.f. Section 5.4)

$$q(\mathbf{a})q(p_{cz})$$

- Update outliers  $q(\lambda_{u,v}^I)$  for all pixels using Eq. 16

**until** Convergence

**Return**  $q^*(\mathbf{Z})$

---

Each optimization of a group of continuous variables is based on gradient ascent, and thus the need to compute the corresponding Jacobian.

**Test phase evaluations.** During test phase there is no knowledge on the position of  $p$ . Assuming that  $p$  could be everywhere with equal probability, during test we simply ignore the terms related to  $p(\mathbf{o}|p, \cdot)$ . The optimization is done jointly for the eyelids opening, and the eyeball orientation, which is nothing but a combination of Eq. 25 and Eq. 31, which assumes an uninformative prior on the eye orientation.

## 7 Efficient sampling: semi-integral likelihoods

Notice that Eq. 18 involves an integration over the image according to the segmentation parametrized by  $(\mathbf{m}, \mathbf{s}, p_c)$ . In order to compute such integration faster we can precompute semi-integral images as follows:

$$\mathcal{I}(u, v) = \sum_{j=1}^v (1 - \omega^{u,j}) \ln p(c^{u,j} | \Lambda_I) \quad (34)$$

Given that the segmentation is parametric, we can query per image column coordinate the boundaries of each region (cornea, sclera and skin) as row pixel coordinates. We then evaluate the integral of the image column by computing differences of  $\mathcal{I}$  at the boundaries.

This is highly valuable as likelihood image integrations from Eq. 18 are used throughout the inference process. In particular, it is where the main computational cost resides during test time.

This process, combined with a GPU implementation of the Monte Carlo expectation, leads to a fast implementation of the movement parameters inference at test phase.

## References

- [1] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, Oct. 2007.
- [2] M. Oppel and C. Archambeau. The variational gaussian approximation revisited. *Neural Comput.*, 21(3):786–792, Mar. 2009.