Learning to Personalize in Appearance-Based Gaze Tracking

Erik Lindén
Tobii
elin@tobii.com

Jonas Sjöstrand
Tobii
jsjd@tobii.com

Alexandre Proutiere
KTH Royal Institute of Technology
alepro@kth.se

Abstract

*Personal variations severely limit the performance of appearance-based gaze tracking. Adapting to these variations using standard neural network model-adaption methods is difficult. The problems range from overfitting, due to small amounts of training data, to underfitting, due to restrictive model architectures. We tackle these problems by introducing SPatial Adaptive GaZe Estimator (*SPAZE*). By modeling personal variations as a low-dimensional latent parameter space,* SPAZE *provides just enough adaptability to capture the range of personal variations without being prone to overfitting. Calibrating* SPAZE *for a new person reduces to solving a small and simple optimization problem.* SPAZE *achieves an error of 2.70° degrees on the MPIIGaze dataset, improving on the state-of-the-art by 14 %.*

We contribute to gaze tracking research by empirically showing that personal variations are well-modeled as a 3-dimensional latent parameter space for each eye. We show that this low-dimensionality is expected by examining model-based approaches to gaze tracking.

1. Introduction

Video-based gaze tracking deals with the problem of determining the gaze of a person's eye given images of the eyes. By "gaze" one usually means the point on a two-dimensional screen where the person is looking (2D gaze), but sometimes one wishes to determine the complete gaze ray in 3D space, originating from the eye and directed towards the screen gaze point. We refer to this (five-dimensional) quantity as *3D gaze*. Gaze tracking has numerous applications: It is used as a communication aid for people with medical disorders. Experimental psychologists use it to study human behavior. In the consumer market, it is used for human-computer interaction and when used in virtual reality, it can reduce the computational requirements through foveated rendering, that is, rendering in high resolution only where the person is looking.

Gaze tracking techniques can be categorized into model-based and appearance-based methods [8]. Model-based methods use image features such as the pupil center and the iris edge, combined with a geometric eye model to estimate the gaze direction. Some model-based methods also use the corneal reflections from one or more light sources. These reflections are known as *glints* and the light sources are known as *illuminators*, typically light-emitting diodes. Model-based methods can be implemented with small amounts of training data, since they make simplifying assumptions. For example, they might model the pupil as a dark ellipse. However, the same assumptions make them unsuited to handle large variations in appearance.

Appearance-based methods, on the other hand, do not rely on hand-crafted features. Instead, they estimate the gaze direction directly from the eye images. This requires a larger amount of training data, but makes it possible to track gazes despite appearance variations. Since appearance-based methods do not require an explicit feature-extraction step, they are believed to work better than model-based methods on low-resolution images [23]. Recent research on appearance-based methods using convolutional neural networks [2, 18, 11, 25, 24, 21, 3] has focused on challenging in-the-wild scenarios with person-independent models using low-resolution web camera images, and provided promising results.

In this paper, we propose a new approach to personal calibration in appearance-based gaze tracking: SPAZE (SPatial Adaptive GaZe Estimator). We analyze the performance of SPAZE on both low-resolution webcam images and on high-resolution images from near-infrared (NIR) cameras with active illumination. SPAZE consists of three steps: 1) From an image of the face, we extract three normalized images, two high-resolution images of the eyes and one low-resolution image of the face. 2) The three images are fed into separate convolutional neural networks. 3) The outputs of the networks are combined with person-specific calibration parameters in a fully connected layer. The calibration parameters are tuned by having the person look at known gaze targets. Previous methods for personal calibration have trained person-specific models [23, 25] or seen calibration

as a post-processing step [11]. This contrasts with SPAZE where we include calibration in the learning process, as a set of latent parameters for each person.

We aim to estimate gaze rays in 3D space. To make data collection simpler, we aim to learn this without ground truth for the 3D eye positions. We only require ground truth for the 3D gaze targets. By estimating 3D gaze and having a geometry-agnostic personal calibration, we hope to learn a single model that can be used with different camera/screen geometries.

SPAZE is evaluated on three datasets: 1) a large Tobii dataset from high-resolution, near-infrared cameras with active illumination, 2) GazeCapture [11] from iPhones and iPads and 3) MPIIGaze [25] from laptop webcams. Our experiments demonstrate that:

- SPAZE is as accurate as model-based gaze tracking methods on high-resolution, near-infrared images.
- Our calibration achieves state-of-the-art performance on both within-MPIIGaze evaluations and on cross-dataset evaluations, transferring from GazeCapture to MPIIGaze.
- SPAZE works without the help of a head pose estimator, using only a tight crop around the eyes.
- Personal variations are well-modeled as a 3-dimensional latent parameter space for each eye.

The contributions and outline of the paper are as follows: In Section 2, we review related work on appearance-based gaze tracking and approaches to personal calibration. In Sections 3 and 4, we describe SPAZE and the datasets we used. In Section 5, we present experimental results. We find that personal variations can be modeled as a 3-dimensional latent parameter space for each eye. In Section 6, we argue why this is expected, reviewing typical model-based methods for gaze tracking. In Section 7, we discuss the implications of our results for appearance-based gaze tracking.

2. Related work

Appearance-based gaze tracking has received much attention recently. Sugano *et al.* [17] used random forest regression to estimate gaze angles from eye images. They introduced the normalization technique we adopt in this paper, where the eye images are warped into a normalized camera view. This effectively reduced the appearance variations their regressor had to handle. For training, they augmented their dataset by rendering eye images from point clouds. Zhang *et al.* [23] introduced MPIIGaze, a dataset with 38 k images from 15 persons. The dataset includes camera calibration parameters and 3D gaze targets. They trained a light-weight convolutional neural network to estimate gaze angles from eye images. Krafka *et al.* [11] collected a dataset of 2.5 M images taken with smartphones in uncontrolled environments. They trained a convolutional neural network and without

personal calibration they obtained an accuracy of about 3° (2 cm) on a phone or tablet. While the dataset is large, it lacks camera calibration parameters and 3D ground truth for the gaze targets. Wood *et al.* [19] introduced UnityEyes, a framework for generating synthetic eye images which look realistic. With a k-nearest neighbor regressor, they achieved an accuracy of 10° on MPIIGaze. Park *et al.* [15] used a novel heat map approach to improve the regression performance, achieving 4.5° on MPIIGaze. Fischer *et al.* [5] trained an ensemble of 4 VGG networks, achieving 4.3° on MPIIGaze.

2.1. Approaches to personal calibration

In model-based gaze tracking there is often some kind of personal calibration involved [8]. The calibration parameters typically include the fovea offset for each eye, as discussed in Section 6. In appearance-based methods there is no explicit model of the eye, so it is not clear how to incorporate personal calibration. Most works ignore calibration, though some authors have shown that person-specific estimators greatly outperform generic estimators.

For example, Sugano *et al.* [17] compared random forest regression with person-specific training and cross-person training, finding errors of 3.9° and 6.5°. Zhang *et al.* [23] made the same comparison using a convolutional neural network on a different dataset, finding 3.3° and 6.3°. The comparison on MPIIGaze found 2.5° and 5.4° [25]. Building on the idea of training person-specific estimators, Zhang *et al.* [21] devised a method which helped to collect more training data for a specific person.

A neural network trained for a specific person will of course always outperform a generic network, assuming equal amounts of training data. However, it is unpractical to collect the vast amount of data needed to train a modern neural network from a single person.

One approach to this problem is to fine-tune some part of a generic network, typically the final layer. We adopt this approach as in iTracker by Krafka *et al.* [11], but with a critical difference. In iTracker, the network is initially trained to learn an average model for all persons present in the dataset. The calibration is just a post-processing step where the last fully connected layer with input size 128 and output size 2 is replaced by an SVR-model trained with calibration samples for a specific person (while the weights of the rest of the network are kept fixed). With 13 calibration points, the accuracy was improved with up to 20 %, whereas with only 4 calibration points, the accuracy was actually worse than without personal calibration, probably due to overfitting of the SVR-model. In iTracker, personal calibration is learnt from scratch for each person. This requires a large calibration parameter space, to support a sufficiently rich set of calibration mappings, and hence a large number of calibration samples. This contrasts with our method: We parameterize
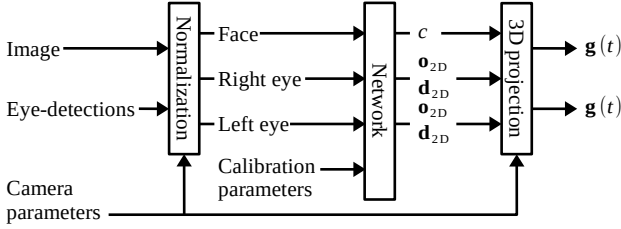
Figure 1: Network pre- and post-processing for image normalization and 3D gaze projection.

the set of possible calibration mappings by a neural network, and during the initial training, the network learns a suitable mapping from a small calibration parameter space. The high expressiveness of the neural network means it can potentially model many different calibration mappings, and the training must find a suitable one. This necessitates a large training set, to prevent overfitting. We have this large dataset, since we learn the calibration mapping from the whole training set, not just from one person.

Park *et al*. [16] adapted the method of iTracker by extracting 32 landmarks (corresponding to iris edges, pupils, eyelids). These features are then used to train a personal SVR as in iTracker.

Liu *et al*. [12] implemented personal calibration by training a differential neural network to estimate the difference in gaze direction between two images. When estimating the gaze direction in a novel image, they used a set of calibration images as *anchors*, estimating the gaze direction in the novel image relative the calibration images. This achieved 4.67° with 9 calibration samples on MPIIGaze. In an inversion of this approach, Yu *et al*. [20] used gaze redirection to synthesize additional person-specific training images. This extended calibration set was then used to fine-tune their network, achieving 4.01° with 9 calibration samples on MPIIGaze.

The state-of-the-art method for calibrated gaze tracking is FAZE (Few-shot Adaptive GaZE Estimation) by Park *et al*. [14]. FAZE uses an encoder/decoder structure to learn a compact and consistent gaze representation. This gaze representation is sent to a small multi-layer perceptron (64 hidden units). The perceptron is trained using model-agnostic meta-learning (MAML) to allow fast person-specific fine-tuning with minimal overfitting. FAZE achieved 3.14° with 9 calibration samples on MPIIGaze, when trained on GazeCapture. The performance was reduced when FAZE was trained on the much smaller MPIIGaze.

3. The SPAZE method

In this section, we describe the three main components of SPAZE: 1) image normalization, 2) the neural network and 3) 3D gaze projection. See Figure 1 for an overview of the data flow. We also describe our personal calibration method.

3.1. Image normalization

To improve generalization, we normalize the images as described by [22]. Unlike most other methods, we do not use a separate head pose estimator for image normalization and 3D gaze estimation. We have found head pose estimators to be fragile and they require pose-annotated training data. Our NIR datasets is also so tightly cropped to the eyes that most of the head is not visible. Instead we adopt a structure where eye detections are used for a rough pose estimation, for image normalization. The network is then asked to correct any errors (affecting gaze accuracy) in the rough estimate. The idea of head pose free methods has previously been investigated by [13, 3].

The input to the image normalization component is an image of a person's face and two points in the image defining where the eyes are. Those points are provided by an external eye detector. The output is three images: two high-resolution eye images centered at the eye detection points and one low-resolution face image centered at the midpoint between the eyes.

By assuming that depth variations in the face are small compared to the distance between the face and the camera, we can compensate for arbitrary scaling and camera rotation by a perspective image warp. This reduces the complexity of the gaze tracking problem, as the estimator does not need to handle arbitrary face rotations or scalings. However, due to imperfections in the normalization method, some rotation and scaling errors will remain.

Figure 2a illustrates the normalization. Given an input image $\mathbf{I}$ and a reference point (either an eye detection point or the midpoint between the eyes), we compute a conversion matrix $\mathbf{R}$. Its inverse $\mathbf{R}^{-1}$ is the matrix that rotates the camera so that it looks at the reference point and so that the interocular vector in the image becomes parallel to the camera x-axis. To make the eye appearance consistent, for the left-eye image we also let $\mathbf{R}^{-1}$ mirror the camera in the interocular direction after the rotation.

The conversion matrix $\mathbf{R}$ will map any 3D point in the real camera coordinate system into the normalized camera coordinate system. The same transform is applied to the image $\mathbf{I}$ using an image transformation matrix $\mathbf{C}_n\mathbf{R}\mathbf{C}_r^{-1}$, where $\mathbf{C}_r$ is the projection matrix of the real camera and $\mathbf{C}_n$ is the projection matrix of the normalized camera. $\mathbf{C}_n$ is selected as a scaling such that the interocular distance in the normalized image becomes 320 pixels for the eye images and 84 pixels for the face image. We use bilinear interpolation to implement the warping and crop out a $W \times H$ region in the normalized image, 224×112 pixels for the eye images and 224×56 pixels for the face image.

A gaze ray $\widehat{\mathbf{g}}(t) = \mathbf{o} + t\mathbf{d}$ is estimated in the normalized camera coordinate system and transformed back to the real
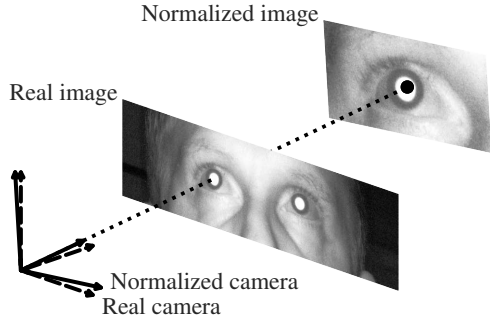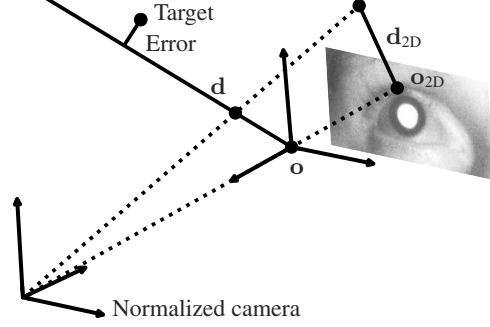
| (a) Image normalization | (b) 3D gaze projection |

Figure 2: (a) The image captured by the physical camera is warped into a normalized camera looking directly at the reference point, in this case the persons right eye. (b) The 2D gaze origin and gaze direction are combined with the corrected distance to form a gaze ray in 3D space. The miss distance between the gaze ray and the gaze target is the loss used to train the neural network.

camera coordinate system by $\mathbf{g}(t) = \mathbf{R}^{-1}\widehat{\mathbf{g}}(t)$.

3.2. 3D gaze projection

Here we describe how the output from the network is translated into a pair of gaze rays. The network has five outputs. For each eye, it predicts a 2D gaze origin $\mathbf{o}_{2D}$ and a 2D gaze direction $\mathbf{d}_{2D}$. It also generates a distance correction term, c, which is common to both eyes. We assume that the distance from eye to camera is approximately the same for both eyes, and our estimate of that distance will be called ρ. First, given the input image and the eye detections, we find a rough distance ρ_{rough} such that the separation between the eyes becomes 63 mm at that distance, approximately the average human interocular distance [6]. This distance is then corrected by the network by setting $\rho = c\,\rho_{rough}$. The rough distance will be unreliable, since it is based on only the eye detections, which are noisy. Further, it makes no allowance for head yaw. But since the same eye detections are used to normalize the images fed to the network, the network has an opportunity to spot misaligned eye detections and correct for them. Likewise, it can measure the head yaw and correct for it.

We will now describe how a gaze ray is computed for a single eye, see Figure 2b for an overview of the procedure. The 3D origin of the gaze ray, $\mathbf{o}$, is computed by back-projecting the 2D gaze origin $\mathbf{o}_{2D}$ through the normalized camera to the distance ρ. To compute the 3D direction of the gaze ray, $\mathbf{d}$, we first construct a set of orthonormal basis vectors $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$, where $\mathbf{z}$ points from the 3D gaze origin to the camera and $\mathbf{x}$ is orthogonal to the y-axis of the normalized camera coordinate system. The 3D gaze direction $\mathbf{d}$ is then computed from the 2D gaze direction $\mathbf{d}_{2D}$ as $\mathbf{d} = [\mathbf{x}\,\mathbf{y}]\,\mathbf{d}_{2D} + \mathbf{z}$.
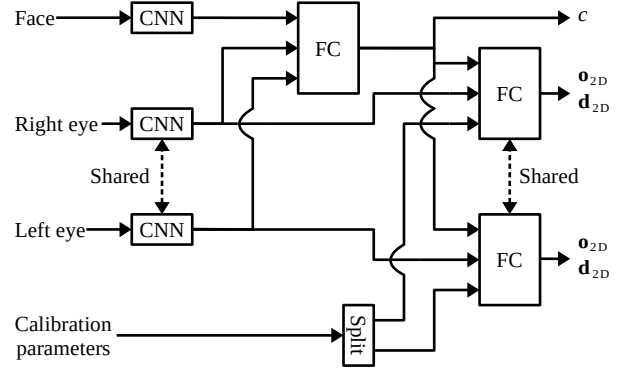

Figure 3: Network architecture.

3.3. Network architecture

Here we describe the input to the network and how the output is generated. See Figure 3 for an overview of the network architecture. We feed three images to the network: both eyes at high resolution and the face at low resolution. One convolutional network is applied separately to each eye. Another convolutional network is applied to the face. Both networks are the convolutional part of ResNet-18 [9].

The three outputs are concatenated and fed to a fully connected module. The fully connected module outputs the distance correction c. The rationale for using both the eyes and the face for the distance estimation is that the eyes could provide accurate interocular distance measurements, while the face could improve head pose estimation.

The convolutional network output for each eye is then concatenated with a set of N personal calibration parameters and the distance correction. This combined feature vector is

fed to a fully connected module. The fully connected module outputs the 2D gaze origin $\mathbf{o}_{2D}$ and the 2D gaze direction $\mathbf{d}_{2D}$. The same module is used for both eyes.

The fully connected modules can both be described as: FC(3072)→BN→ReLU→DO→FC(3072)→BN→ReLU→DO→FC(1 or 4) where FC is fully connected, BN is batch normalization and DO is dropout.

Initially we generated both gaze rays using information from both eyes. That gave better performance, but made the gaze rays for the two eyes highly correlated, as all training data have both eyes looking at the same point. As we want to support medical applications, where it is important to observe differences between the eyes, we choose this segregated approach. In a consumer application, it would be more appropriate to modify the network architectures to estimate a joint gaze direction, originating from between the eyes, as suggested by Zhang *et al*. [24].

The rationale for providing the distance correction c when estimating the gaze direction is to allow the use of features that require accurate distance information. This is the case with pupil-center/corneal-reflection gaze mapping [8, 7]. We do not input personal calibration parameters into the distance-estimation module, since it is typically impossible to detect distance errors from calibration data collected at a single distance, which is what we have.

3.4. Training and calibration

Personal variations are modeled by assigning $2N$ calibration parameters to each person, N for each eye. During training, the training set $\mathcal{D}$ consists of triples (X, t, m), where X is a triple of images (face, right eye, left eye), t is the 3D gaze target and m is the index of the person in the images. With M persons in the training set, we solve the optimization problem[1]

$$(\theta_{\mathrm{opt}}, P_{\mathrm{opt}}) = \operatorname*{argmin}_{\theta, P} \sum_{(X,t,m) \in \mathcal{D}} \mathrm{loss}(g_\theta(X, P_{m,\cdot}), t)$$

over all weights θ of the neural network g and, simultaneously, over all M-by-$2N$ matrices P of calibration parameters. After training, P_{opt} is discarded. The optimization method and the loss function will be described in a moment.

To calibrate for a new person, we collect a small calibration set $\mathcal{D}_{\mathrm{cal}}$ of (image triple, gaze target) pairs. The calibration procedure then finds the person's $2N$ calibration parameter vector defined as

$$\operatorname*{argmin}_{p \in \mathbb{R}^{2N}} \sum_{(X,t) \in \mathcal{D}_{\mathrm{cal}}} \mathrm{loss}(g_{\theta_{\mathrm{opt}}}(X, p), t).$$

---

[1]Note that our objective function includes person-specific parameters P_m, which contrasts with existing methods [11] whose objective function does not account for personal variations (with our notations, it would be $\sum_{(X,t) \in \mathcal{D}} \mathrm{loss}(g_\theta(X), t)$).

Note that the network weights are fixed. We used BFGS to solve this optimization problem, and as an initial guess we find a single calibration parameter vector that minimizes the error over the whole training set. This parameter vector is also used for the uncalibrated case.

In the experiments, we vary N and find that 3 parameters per eye are enough to provide an efficient person-specific gaze tracking. Modeling personal variations using such a low-dimensional latent parameter space contrasts with existing calibration approaches (e.g. in iTracker, a much higher number of parameters is used), but can be motivated and justified by existing model-based methods; see Section 6 for a complete discussion.

Finally, let us describe the loss function and the optimization method during training. The loss is the miss distance between the gaze ray and the 3D gaze target, see Figure 2b. We train using Adam [10] with a learning rate of 10^{-3} and a weight decay of 10^{-5}. We train for 30 epochs, decaying the learning rate by 10 every 10th epoch. The eye detections are jittered for data augmentation in training: We randomly offset the detections in a disk with a radius equal to 4% of the interocular distance.

With this setup, we found that the generated gaze rays would pass close to the gaze target, but the 3D gaze origin would be highly incorrect. The reason is that current datasets are collected with a geometry that makes the 3D gaze origin underdetermined. In the supplementary material, we use a synthetic dataset to investigate whether this limitation can be overcome by a different data collection strategy.

To prevent unphysical solutions when training on current datasets, we introduced two regularizing terms: 1) a hinge loss on the 2D gaze origin $\mathbf{o}_{2D}$, penalizing if it moves outside the eye image and 2) a hinge loss on the distance correction c, penalizing changes in distance by more than 40% in either direction.

4. Datasets

We use three datasets in our experiments: 1) a Tobii dataset from high-resolution, near-infrared (NIR) cameras, 2) GazeCapture and 3) MPIIGaze. In the supplementary material we investigate the effect of gaze target placement and glints using synthetic UnityEyes images.

4.1. NIR dataset

We use a large Tobii dataset for our calibration experiments. The dataset was collected with a near-infrared (NIR) eye tracker platform, with an infrared illuminator mounted very close to the camera. This produces a bright-pupil effect [8], the same effect that makes the eyes red in flash photography. Since the illuminator position coincides with the camera position, we can scale and rotate the normalized camera without changing the position of the illuminator in the camera coordinate system. The training set was collected

over a period of several years and contains 427 k images from 1824 persons. The majority of the training data have gaze targets on a regular lattice on the screen.

For testing, we have a set of 200 persons. This set was collected on 19 inch, 16:10 aspect ratio screens, with the camera placed at the bottom edge of the screen and tilted up 20°. The camera focal length was 3679 pixels and it captured a region of interest of 1150×300 pixels. The region of interest was kept aligned on the eyes using an eye detector. The persons sat at 65 ± 10 cm from the camera. See Figure 2a for an example image. Half of the recordings were made in Sweden, the other half in China.

There are three recordings for each person, one for calibration and two for test. From each recording, we extract 45 images, evenly distributed over gaze targets. The calibration recordings have gaze targets on a regular 3×3 lattice. For the test recordings, the screen was divided into a 3×4 grid and a gaze target was placed randomly in each grid cell. The screen brightness was also randomized. For testing, we first calibrate on the calibration recording and then report the error on the two test recordings. As the setup was quite controlled, we believe head yaw angles were on the order of $10°$.

4.2. RGB datasets

GazeCapture [11] contains 1.5 M images from 1471 persons, collected on iPhones and iPads. It does not provide camera calibrations. To apply SPAZE, we assume that all cameras have a horizontal field-of-view of 54.4° [1] and that the principal point is always in the middle of the image. As the eye detections are poor, we increase the area viewed by our eye images by 50 %. For testing, we use those 105 persons in the test set having at least 756 images. We report the same error metric as Krafka *et al.*, the on-screen miss distance, since GazeCapture does not provide the 3D data needed to compute angles.

MPIIGaze [21] is a widely used benchmark for gaze estimation. It consists of webcam images from 15 persons. We used the MPIIFaceGaze [24] subset, which has around 2500 images per person.

We adopt the experimental protocol of FAZE [14] where the last 500 images for each person are used for evaluation and k calibration samples are drawn randomly from the remaining images. For each k, we adaptively control the total number of trials to keep the uncertainty in the final error below $0.01°$.[2] For cross-dataset evaluation, we train on the training set of GazeCapture and test on MPIIGaze. For within-dataset evaluation, we perform leave-one-out training on MPIIGaze.

---

5. Experiments

While we minimize the gaze ray miss distance, we report the error as an angle for easier comparison with previous works. Since we estimate a 3D gaze ray, we compute the angle as follows: Let **ET** be the line through the true (annotated) 3D eye center and the 3D gaze target. Then we find the point **G** on the gaze ray $\mathbf{g}(t)$ where the line **GT** is orthogonal to **ET**. We measure the error as the angle between **ET** and **EG**.

In some figures, we show shaded bands around the mean value. These show the *consistency* of the calibration, how much a user can expect the mean error to vary between repeated calibrations.[3]

5.1. Number of calibration parameters

We vary the number of calibration parameters (N), see Figure 4. In all cases, there is little improvement beyond 3 parameters. In (b) and (d), we see that fewer parameters are generally better for few calibration samples, which is expected since the low dimensionality provides regularization. We use $N{=}3$ in all subsequent experiments.

For the NIR dataset, (a), we found that calibration shifts the whole error distribution downwards. Specifically, any error quantile is reduced by a factor of 2.5. This suggests that personal variations are a dominating factor throughout the error distribution. Our calibrated mean error on the NIR dataset is $0.79°$ degrees (uncalibrated $1.85°$), which compares well with multi-camera, multi-illuminator systems [8, 4].

The improvement from $N{=}2$ to $N{=}3$ is much larger for GazeCapture-to-MPIIGaze, (c), than for within-dataset testing, (b) and (d). Looking closer at the GazeCapture results, we found that changing $N{=}2$ to $N{=}3$ mostly affected the error for tablets (21.0 mm $\rightarrow 17.0$ mm), while phones remained similar (12.5 mm $\rightarrow 12.3$ mm). Tablets generate larger gaze angles than phones, but tablets are a minority in GazeCapture. We suspect that with $N{=}2$, the network is forced to choose between offset and scaling (see Section 6). For small gaze angles, like on phones, offset might be more important than scaling. But for MPIIGaze, which has larger gaze angles, scaling might be more important than offset.

5.2. Comparison with state-of-the-art

We compare SPAZE with FAZE [14], a state-of-the-art method for calibrated appearance-based gaze tracking. To closer match the FAZE experimental protocol, we form a single gaze ray by averaging the origins and directions of the two individual gaze rays generated by our network.

See Figure 5 for the comparison with FAZE. Within-MPIIGaze training does not work well with FAZE, while
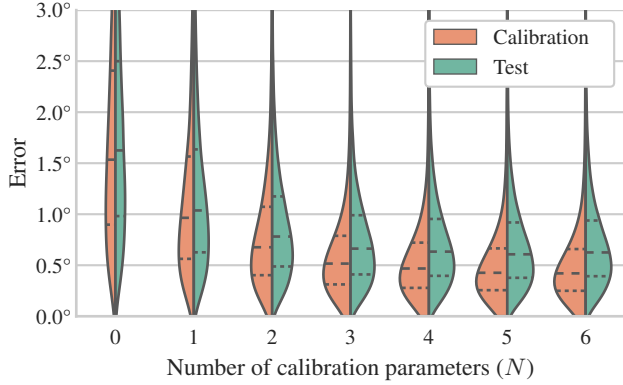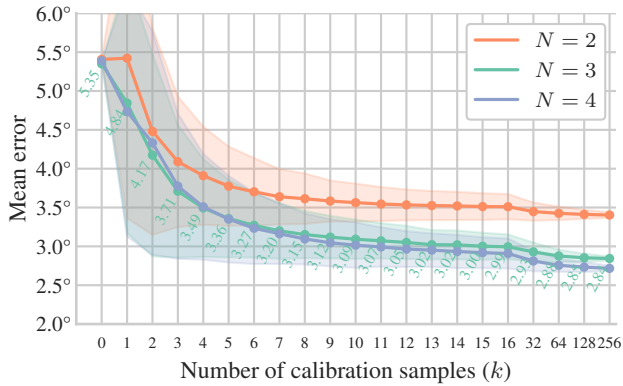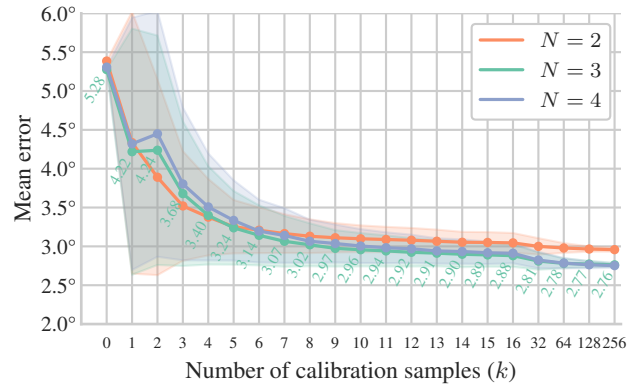

(a) NIR dataset
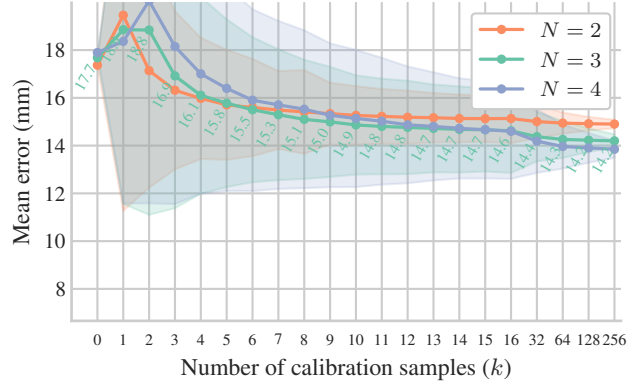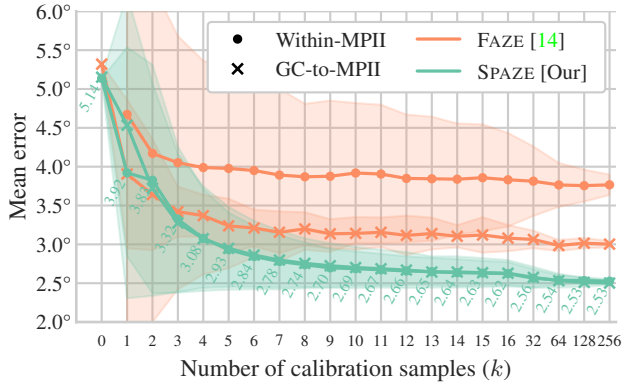

(b) GazeCapture (test)

(c) GazeCapture-to-MPIIGaze

(d) Within-MPIIGaze

Figure 4: Comparison of different numbers of calibration parameters (N).

Figure 5: MPIIGaze. Comparison with FAZE.

Method	Samples (k)	Their (°)	Our (°)
Diff-NN [12]	9	4.67	2.94 ± 0.22
GazeML [16]	20	4.6	2.82 ± 0.09
RedFTAdap [20]	1	4.97	4.12 ± 1.48
	5	4.20	3.16 ± 0.42
	9	4.01	2.94 ± 0.22
GazeNet+ [25]	2500	2.5	2.76 ± 0.00

Table 1: Comparison of SPAZE with other calibrated methods on MPIIGaze. SPAZE is evaluated using the experimental protocol of the compared method.

SPAZE performs equally well within MPIIGaze and transferred from GazeCapture to MPIIGaze. Looking at only the FAZE GazeCapture-to-MPIIGaze results, the two methods have similar uncalibrated performance, but SPAZE has a better mean error for $k > 3$. For $k=9$, we improve from 3.14° to 2.70°. FAZE has better consistency throughout, especially for small k. This consistency seems to come at the cost of underfitting. SPAZE does not consider the a priori likelihood

of the calibration parameters, it just drives the empirical calibration error as low as possible. This results in large variations for small k.

We also compare SPAZE with some other calibrated methods, see Table 1. SPAZE outperforms the other methods, approaching the performance of GazeNet+, which trains a person-specific network on 2500 images. Another advan-
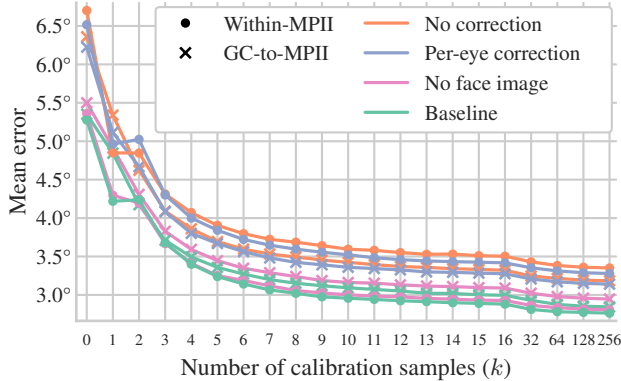
Figure 6: MPIIGaze. Different types of distance correction.

tage of SPAZE over most other methods is the absence of hyper-parameters; there is no learning rate or early stopping to tune. The calibration optimization uses the standard BFGS optimizer, operating on 2×3 parameters, and drives the calibration error to a local minima. Since only the fully connected layers depend on the calibration parameter, the calibration optimization does not need to evaluate the expensive convolutional layers, resulting in a fast calibration.

5.3. Distance correction

To validate our use of a distance correction derived from the two eye images and the face image, we perform an ablation study where we: 1) completely disable the distance correction and only use the rough distance from the eye detections, 2) estimate a per-eye distance correction, using only the image of that eye and 3) remove the face image input.

The results in Figure 6 show that our method is superior to the alternatives. Note that we do not want to feed the image of one eye to the network that is estimating the gaze direction of the other eye, to ensure separate estimates for the two eyes. The difference in performance with and without the distance correction is much smaller with calibration (0.59°) than without calibration (1.43°). This can be explained as follows: Even if the network cannot explicitly change the distance estimate, it can compensate for a flawed distance estimate by changing the gaze angle (see the supplementary material). This is why a per-eye distance correction performs similarly to no distance correction. Estimating the distance from a single eye image is more difficult than estimating the distance from both eye images (and the face image). Calibration can alleviate this difficulty by providing person-specific information on how to estimate the distance from the single eye image.

6. Low-dimensional parameter space for calibration: the case of model-based methods

To understand why it is reasonable to model personal variations as a low-dimensional latent parameter space, it helps

to look at a typical model-based method for gaze tracking. Here we will review the eye model described by Guestrin and Eizenman [7]. For a comprehensive review of model-based methods, we refer to [8].

We will describe a gaze mapping model called pupil-center/corneal-reflection (PCCR). Assume a system with one camera and a collocated illuminator. Further assume we know the distance to the eye, from head pose estimation, a second stereo camera or some other method.

Image processing methods detect the corneal reflection, the glint, from the illuminator. The center of the pupil is also detected. The difference between these two points forms the pupil-center/corneal-reflection vector. If the cornea is assumed to be spherical, the cornea center lies directly behind the glint. The *optical axis*, a line passing through the cornea center and the pupil center, can then be computed if the distance between the person's cornea center and pupil center is known. This distance is one calibration parameter.

However, the optical axis is not the *visual axis*, the person's line of gaze. The fovea, the most sensitive part of the retina, is offset from the optical axis, and this offset, in two dimensions, differs from person to person.

Taken together, we have three parameters per eye for each person. The foveal offset roughly corresponds to shifting the gaze up-and-down and side-to-side, and the cornea-center/pupil-center distance scales the gaze around the optical axis.

Over the three datasets, NIR, GazeCapture and MPIIGaze, GazeCapture sees the least improvement from calibration, and the NIR dataset sees the greatest. The former also has the worst images, while the latter has the best. We do not believe that this is a coincidence. As measurement errors in the images decrease, the mismatch between a generic gaze tracking model and a specific person's eye geometry becomes the dominant source of errors.

7. Conclusions

We have presented SPAZE, a method for calibrated appearance-based gaze tracking. SPAZE achieves state-of-the-art results on MPIIGaze and is as accurate as model-based gaze tracking on high-resolution, near-infrared image. Our results show that accurate gaze tracking is possible without a separate head pose estimator. Our results also show that personal variations are well-modeled as a 3-dimensional latent parameter space for each eye.

References

[1] Apple Inc. iOS Device camera summary. Available at https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/Cameras/Cameras.html 6

[2] S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 753–760. Morgan-Kaufmann, 1994. 1

[3] H. Deng and W. Zhu. Monocular free-head 3D gaze tracking with deep learning and geometry constraints. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3162–3171, Oct. 2017. 1, 3

[4] O. Ferhat and F. Vilario. Low cost eye tracking: The current panorama. *Computational Intelligence and Neuroscience*, page 14, 2016. 6

[5] T. Fischer, H. Jin Chang, and Y. Demiris. RT-GENE: Real-time eye gaze estimation in natural environments. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[6] C. C. Gordon, C. L. Blackwell, B. Bradtmiller, J. L. Parham, P. Barrientos, S. P. Paquette, B. D. Corner, J. M. Carson, J. C. Venezia, B. M. Rockwell, M. Mucher, and S. Kristensen. 2012 Anthropometric survey of U.S. Army personnel: Methods and summary statistics. Technical Report NATICK/15-007, Natick MA: U.S. Army Natick Soldier Research, Development and Engineering Center, 2014. 4

[7] E. D. Guestrin and M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on Biomedical Engineering*, 53(6):1124–1133, June 2006. 5, 8

[8] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):478–500, 2010. 1, 2, 5, 6, 8

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 4

[10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[11] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 5, 6

[12] G. Liu, Y. Yu, K. A. Funes Mora, and J.-M. Odobez. A differential approach for gaze estimation with calibration. In *29th British Machine Vision Conference BMVC*, 2018. 3, 7

[13] F. Lu, T. Okabe, Y. Sugano, and Y. Sato. A head pose-free approach for appearance-based gaze estimation. In *British Machine Vision Conference, BMVC 2011, Dundee, UK, August 29 - September 2, 2011. Proceedings*, pages 1–11, 2011. 3

[14] S. Park, S. D. Mello, P. Molchanov, U. Iqbal, O. Hilliges, and J. Kautz. Few-shot adaptive gaze estimation. In *2019 IEEE International Conference on Computer Vision (ICCV)*, 2019. 3, 6, 7

[15] S. Park, A. Spurr, and O. Hilliges. Deep pictorial gaze estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[16] S. Park, X. Zhang, A. Bulling, and O. Hilliges. Learning to find eye region landmarks for remote gaze estimation in unconstrained settings. In *Proc. International Symposium on Eye Tracking Research and Applications (ETRA)*, pages 21:1–21:10, 2018. 3, 7

[17] Y. Sugano, Y. Matsushita, and Y. Sato. Learning-by-synthesis for appearance-based 3D gaze estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1821–1828, June 2014. 2

[18] K.-H. Tan, D. J. Kriegman, and N. Ahuja. Appearance-based eye gaze estimation. In *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision*, WACV '02, pages 191–, Washington, DC, USA, 2002. IEEE Computer Society. 1

[19] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138, 2016. 2

[20] Y. Yu, G. Liu, and J.-M. Odobez. Improving few-shot user-specific gaze adaptation via gaze redirection synthesis. In *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 7

[21] X. Zhang, M. X. Huang, Y. Sugano, and A. Bulling. Training person-specific gaze estimators from interactions with multiple devices. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2018. 1, 2, 6

[22] X. Zhang, Y. Sugano, and A. Bulling. Revisiting data normalization for appearance-based gaze estimation. In *Proc. International Symposium on Eye Tracking Research and Applications (ETRA)*, pages 12:1–12:9, 2018. 3

[23] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, June 2015. 1, 2

[24] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2299–2308, July 2017. 1, 5, 6

[25] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. MPIIGaze: Real-world dataset and deep appearance-based gaze estimation. *CoRR*, abs/1711.09017, 2017. 1, 2, 7