Appearance-Based 3D Gaze Estimation with Personal Calibration

Erik Lindén
Tobii
elin@tobii.com

Jonas Sjöstrand
Tobii
jsjd@tobii.com

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

Abstract

We propose a way to incorporate personal calibration into a deep learning model for video-based gaze estimation. Using our method, we show that by calibrating six parameters per person, accuracy can be improved by a factor of 2.2 to 2.5. The number of personal parameters, three per eye, is similar to the number predicted by geometrical models. When evaluated on the MPIIGaze dataset, our estimator performs better than person-specific estimators.

To improve generalization, we predict gaze rays in 3D (origin and direction of gaze). In existing datasets, the 3D gaze is underdetermined, since all gaze targets are in the same plane as the camera. Experiments on synthetic data suggest it would be possible to learn accurate 3D gaze from only annotated gaze targets, without annotated eye positions.

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 wants 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*.

There are plenty of applications. Gaze tracking 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 estimation techniques can generally be divided into model-based and appearance-based methods [5]. 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 makes them unsuited to handle large variations in appearance.

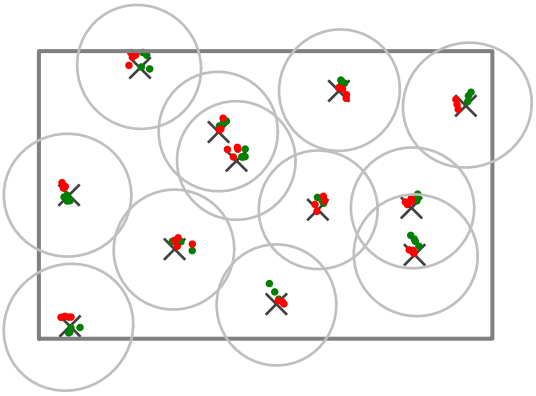


Figure 1: Result from an average performing recording in our dataset. The rectangle is a 19 inch laptop screen. Green and red dots are predicted gaze points, crosses are gaze targets. Around each target, we show a circle corresponding to 4.8°, the best result [15] on the MPIIGaze dataset [14]. MPIIGaze uses similar laptop screens, but is recorded with web cameras and the results are without personal calibration.

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 learn invariance to 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 [14]. Further, appearance-based methods may be more reliable even on high-resolution images, since they do not rely on specific

image features or assumptions about the eye geometry.

Recent research on appearance-based methods using convolutional neural networks has shown significant improvements in accuracy [9, 16, 15, 13, 1], and has focused on challenging in-the-wild scenarios with person-independent models using low-resolution web camera images. Given the potential benefits of appearance-based methods, we want to examine the performance in more constrained scenarios, with high-resolution near-infrared cameras, active illumination and personal calibration. See Figure 1 for an illustration of the improvement in accuracy these restrictions make possible.

Previous methods for personal calibration have trained person-specific models [14] or seen calibration as a post-processing step [9]. We want to include calibration in the learning process, as a set of latent variables for each person. As Deng et al. [1], we want to predict a gaze ray in 3D space, and to make data collection simpler, we want to learn 3D predictions without explicit 3D annotations, that is, with only 3D gaze targets and no 3D eye positions as ground truth. By predicting 3D gaze and having a generic personal calibration, we hope to learn a single model that can be used with different camera/screen geometries.

In this paper, we show the following:

- Appearance-based methods can be as accurate as model-based methods.
- Personal variations can be modeled as a low-dimensional latent parameter space where it is easy to find the optimal point for a given person.
- 3D gaze can be learned without explicit 3D annotations.

Here is the outline of the paper: First, in Section 2, we review related work on appearance-based gaze tracking and approaches to personal calibration. In sections 3 and 4, we describe our method and the three datasets we used. The experimental results are found in Section 5. One of our results is that the personal variations can be modeled as a low-dimensional latent parameter space, and experiments suggest that it is enough to have about three parameters per eye. In Section 6, we argue why this is expected. Finally, in Section 7, we discuss the implications of our results for appearance-based gaze tracking.

2. Related work

Appearance-based gaze tracking has received a lot of attention recently.

In 2014, Sugano et al. [11] used random forest regression to predict 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 needed to handle. For training, they augmented their dataset by rendering eye images from point clouds.

Soon thereafter, Zhang et al. [14] introduced MPIIGaze, a dataset with more than 200 000 images from 15 persons. The dataset includes camera calibration parameters and 3D gaze targets. They trained a light-weight convolutional neural network to predicting gaze angles from eye images.

In 2016, Krafka et al. [9] collected a dataset of almost 2.5 million images taken with smartphones in uncontrolled environments. They train 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 coordinates of the gaze targets.

In 2016, Wood et al. [12] introduced UnityEyes, a framework for generating synthetic eye images which look very realistic. With a k-nearest neighbor regressor, they obtained an accuracy of about 10° on the MPIIGaze dataset.

In 2017, Deng and Zhu [1] used deep learning to predict head pose and 3D gaze in a coordinate system following the head pose. They trained and evaluated on their own data and obtained a gaze direction accuracy of 4.3°.

2.1. Approaches to personal calibration

In model-based gaze tracking there is often some kind of personal calibration involved [5]. The personal 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 papers ignore calibration, though some authors have shown that person-specific estimators greatly outperform generic estimators.

For example, Sugano et al. [11] compared random forest regression with within-person training to the same method with cross-person training, finding errors of 3.9° and 6.5°. Zhang et al. [14] made the same comparison using a convolutional neural network on a different dataset, finding 3.3° versus 6.3°. Building on the idea of training person-specific estimators, Zhang et al. [13] devised a method for 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. But therein lies the crux, for it is unpractical to collect the vast amount of data needed to train a modern neural network from a single person.

One approach to solve this problem is to retrain not the complete network but only a part of it, typically the final layer. To our knowledge, the only implementation in that vein was done by Krafka et al. [9]. The final layer in their network was a fully-connected layer with input size 128 and output size 2 (gaze coordinates on screen). To allow for personal calibration, after training the network on a large

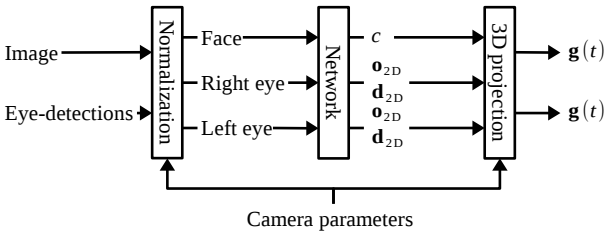


Figure 2: Network pre- and post-processing.

multi-subject dataset, they replaced the final layer with an SVR-model that was trained with calibration samples for each person while the weights of the rest of the network were kept fixed. With 13 calibration points the accuracy was improved with up to 20 percent, but with only four calibration points the accuracy was actually worse than without personal calibration, probably due to overfitting of the SVR-model.

3. Method

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

3.1. Image normalization

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.

To improve generalization, we normalize the images as described in [11]. For completeness, we briefly describe the method here.

By assuming that the face region has a short depth compared to the distance between the camera and the face, we can compensate for arbitrary scaling and camera rotation by a perspective image warp. This reduces the complexity of the gaze estimation 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 3a 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 right-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 $\mathbf{W} = \mathbf{C}_n\mathbf{R}\mathbf{C}_r^{-1}$, 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 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 px for the eye images and 224×56 px for the face image.

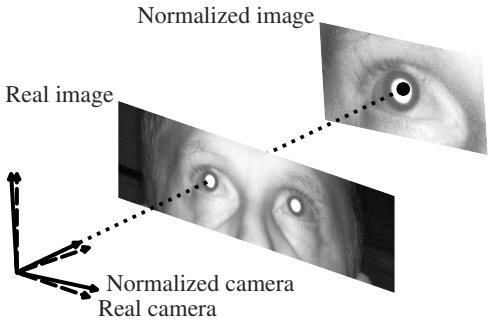
A gaze ray $\widehat{\mathbf{g}}(t) = \mathbf{o} + t\mathbf{d}$ is estimated in the normalized camera coordinate system and transformed back to the real camera coordinate system by $\mathbf{g}(t) = \mathbf{R}^{-1}\widehat{\mathbf{g}}(t)$.

3.2. 3D gaze projection

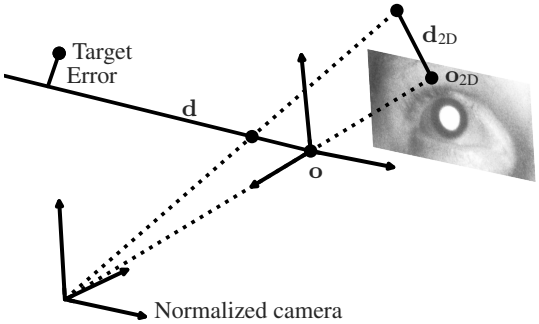
Here we describe how the output from the network is translated into a pair of 3D gaze rays. The network has five outputs. For each eye, it predicts a 2D gaze origin $\mathbf{o}_{2D}$ and a 2D gaze direction $\mathbf{d}_{2D}$. It also predicts a distance correction term, c, which is common to both eyes. We assume that the distance from camera to eye is approximately the same for both eyes, and our estimate for it 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 [3]. This distance is then corrected by the network by letting $\rho = c\rho_{\text{rough}}$. The rough distance will be unreliable, since it is based only on 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 3D gaze ray is computed for a single eye, see Figure 3b for an overview of the process. 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 gaze direction in 3D, we first construct a set of orthonormal basis vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$, where $\mathbf{z}$ points from the gaze origin to the camera and $\mathbf{x}$ is orthogonal to the y-axis of the normalized camera coordinate system. The gaze direction is then computed as

$$\mathbf{d} = \begin{bmatrix} | & | \\ \mathbf{x} & \mathbf{y} \\ | & | \end{bmatrix} \mathbf{d}_{2D} + \begin{bmatrix} | \\ \mathbf{z} \\ | \end{bmatrix}$$



(a) Image normalization



(b) 3D gaze projection

Figure 3: (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.

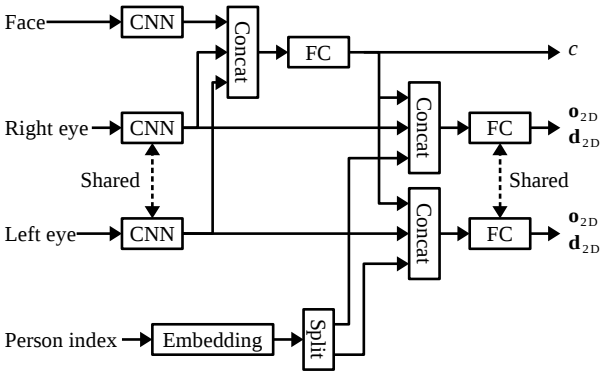


Figure 4: Network architecture.

3.3. Network architecture

Here we describe the input to the network, how the output is predicted and the loss function. See Figure 4 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. We have separate convolutional networks for the eyes and the face. Both networks are the convolutional part of ResNet-18 [6]. The output from all convolutional networks, both eyes and the face, are concatenated and fed to a fully connected module, which predicts the distance correction c. The rationale is to provide the distance estimation with fine features from the eye images and and a full face view to guide head yaw estimation.

The output from each eye convolution is 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 modules can be described as: FC(3072)-BN-ReLU-DO(0.5)-FC(3072)-BN-ReLU-DO(0.5)-FC($\{4, 1\}$) where FC is fully connected, BN is batch normalization [7] and DO is dropout [10]. The output is either the 2D gaze origin and 2D gaze direction, or the distance correction.

Initially we predicted the gaze origins and directions using information from both eyes. That improved performance, but made the predictions for the two eyes highly correlated, as all training data have both eyes looking at the same point. The rationale for providing the distance correction c to the modules that predict the gaze directions is to allow it to use features that require accurate distance information, as is the case with pupil-center/corneal-reflection gaze mapping [5, 4]. We do not provide personal calibration parameters to the distance estimation module, as it is typically impossible to detect distance errors from calibration data collected at one distance, which is what we have.

The network is trained to minimize the miss distance between the gaze ray and the 3D gaze target, see Figure 3b. In addition to the gaze loss, other parameters are regularized to help the training. Specifically, a hinge loss is applied to the 2D gaze origin $\mathbf{o}_{2D}$ to penalize if it moves outside the eye image. A similar hinge loss is applied to the distance correction c, penalizing changes in distance by more than 40 %.

3.4. Calibration

We assign N calibration parameters to each person and eye. This is implemented by giving each person in the training set an index and using an embedding layer in the network. The parameters are initialized to zero and during training the network learns both the parameter values for each person and an efficient representation of the personal

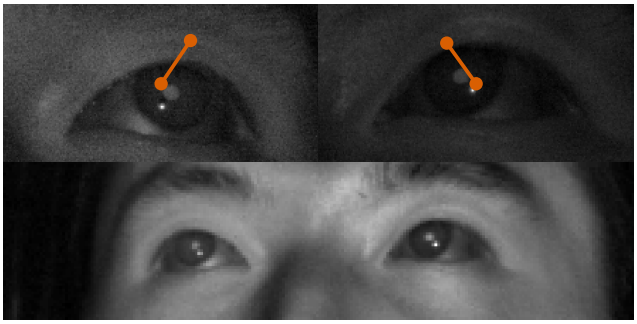


Figure 5: The three images are inputs to the network. The network predicts the gaze origin and gaze direction for each eye.

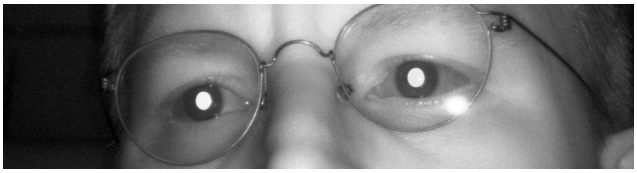


Figure 6: NIR region-of-interest image (contrast enhanced).

variations. When the network is trained, the embedding can be replaced by a simple vector. To calibrate for a new person, a few images are collected with the person looking at known targets and an optimizer adjusts the calibration parameter vector to minimize the gaze error.

3.5. Training

We train using Adam [8] with a learning rate of 10^{-3}. The eye detections are jittered for data augmentation in training, but not in test. Specifically, we randomly offset the detections in a disk with a radius equal to 4 % of the interocular distance. The test results are reported on the model with the lowest validation error.

4. Datasets

We use three different datasets for our experiments: an internal dataset from Tobii with near-infrared illumination, the public MPIIGaze dataset, and synthetic images generated with UnityEyes.

4.1. NIR dataset

We use a large internal dataset at Tobii for our calibration experiments. The dataset was collected on cameras with an infrared illuminator mounted very close to the camera. This produces a bright-pupil effect [5], 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

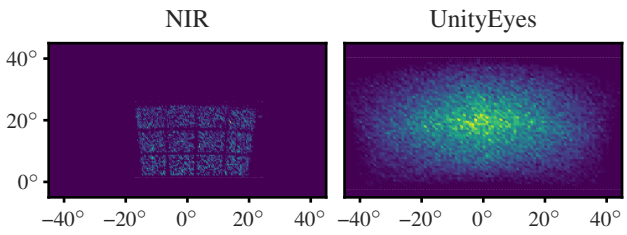


Figure 7: Distribution of gaze angles relative the camera.

system. The training subset was collected over a period of several years and contains 426 535 images from 1824 persons. The majority of the training data have gaze targets on a regular lattice.

For validation and testing, we have two subsets with 200 persons each. These subsets were 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 px and it captured a region-of-interest of 1150×300 px. 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 6 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 evaluation. 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 evaluation recordings, the screen was divided into a 3×4 grid and a gaze target was placed randomly in each grid cell, see Figure 7. The screen brightness was also randomized. For validation and testing, we first calibrate on the calibration recording and then report the error on the two evaluation recordings. As the setup was quite controlled, we believe head yaw angles were on the order of 10°.

4.2. MPIIGaze

For comparison with existing methods, we use the MPIIGaze dataset [14]. As MPIIGaze only contains 15 persons, we do leave-one-person-out training. The images from the first day with at least 100 images are used for calibration. The remaining images are used for testing. Since the number of images varies from person to person, we weight the results so that each person contributes equally. We use the provided (original) eye detections but not the head pose information.

4.3. UnityEyes

The NIR dataset and MPIIGaze have all gaze targets in the same plane as the camera. We found that this makes it impossible to learn meaningful distance corrections or gaze origins. The network can always compensate for an incor-

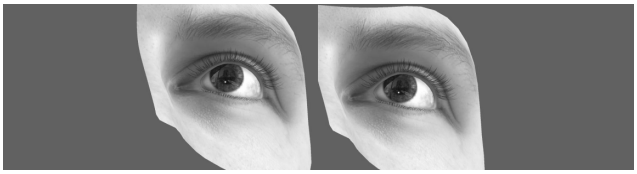


Figure 8: UnityEyes composition.

rect distance or origin by modifying the gaze direction. The gaze ray will still intersect the plane of the gaze targets at the correct point, but the ray is only correct at that point. However, if gaze targets are placed at various z-depths in the camera coordinate system, it is no longer possible to compensate distance errors with the gaze direction. The distance corrections and gaze origins could be trained with head pose annotated data, but we have found it hard to measure the true positions with the required accuracy, less than a centimeter, without a highly restrictive data collection setup.

To test the feasibility of learning 3D gaze without ground-truth eye positions, we generate synthetic images using the UnityEyes tool [12]. This lets us place gaze targets at different depths. The dataset defines gaze in terms of the *optical axis* of the eye [5], so we use no personal calibration.

We make the synthetic images similar to the NIR dataset, with the camera 20° below the face and with gaze points above the camera. Relative to the camera, the gaze angle range is 0° to 40° in pitch and ±40° in yaw. The head pose range is 10° to 30° in pitch and ±20° in yaw. Specifically, we set the fields in the tool to $[-20, 0, 10, 20]$ and $[0, 0, 10, 20]$. Since we know the geometry of the eye surface, we have the opportunity to add glints from a coaxial light source. We generate one million UnityEye images and split 80/10/10 for training/validation/test. To reduce compression artifacts, we generate the images at 1024×768 px and rescale them to match the camera of the NIR dataset. The eyes are placed at 65 cm, with gaze targets randomly placed at a z-depth of one of −30, 0 and +30 cm. We sample the interocular distance from $\mathcal{N}(63\,\mathrm{mm}, 3.5^2\,\mathrm{mm}^2)$ [3]. See Figure 8 for an example image. We use the center of the eyelid annotations as the eye position. To make the positions less perfect, we randomly offset them in a circular disk with a radius equal to 3 % of the interocular distance.

5. Experiments

While we minimize the gaze-ray miss distance, we report the error as an angle for easier comparison with previous work. For the NIR dataset, where we have no ground truth, we assume all eyes are at a distance of 65 cm from the camera. For MPIIGaze, we use the provided 3D eye positions to compute an angular error. While it is common to report the performance as the mean error, we have found that user

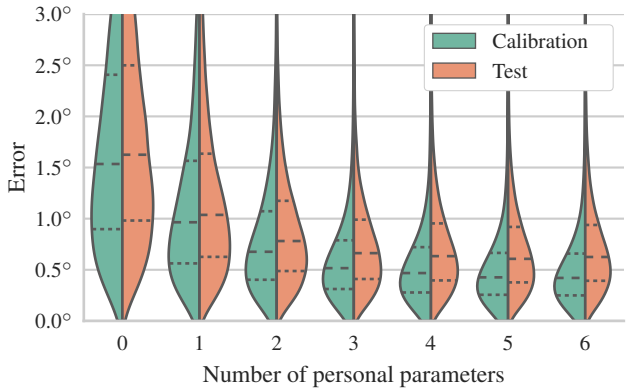


Figure 9: Distribution of angle errors as a function of the number of calibration parameters on the NIR dataset. Dashed lines show quartiles.

Method	Mean error
Generic CNN [14]	6.3°
Person-specific CNN [14]	3.3°
Generic CNN with calibration [our]	2.9°

Table 1: Results on MPIIGaze.

interface design tends to be driven by the worst-case errors. Therefore, we plot the full distribution of errors.

5.1. Calibration

We vary the number of personal calibration parameters and report the error on the calibration recordings and on the test recordings, see Figure 9. We want to point out that the absence of calibration does not add a fixed drop in performance compared to the calibrated case, but rather a scaling. In particular, we find that any given quantile of the error is approximately 2.5 times higher for $N = 0$ (uncalibrated) than for $N = 3$. As $N = 3$ seems to to provide near optimal performance, we use that for all other experiments. The mean error is 0.8°, which compares well to multi-camera, multi-illuminator systems [5, 2].

5.2. MPIIGaze

We compared our generic but calibrated neural network to other methods on the MPIIGaze dataset, see Table 1. Note that the test sets are slightly different, [14] uses 2×1500 eye images with external provided head pose, and the person-specific model is trained on 2500 images and tested on 500 images. Our method uses images from the first day (with at least 100 images) for calibration, tests on all other images and weight the results to account for the different number of images for different persons.

We see that our generic CNN with a low-dimensional personal calibration, three parameters per eye, performs

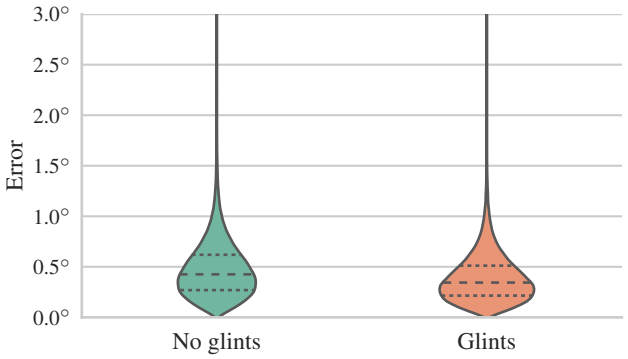


Figure 10: Gaze errors on the UnityEyes dataset, with and without glints.

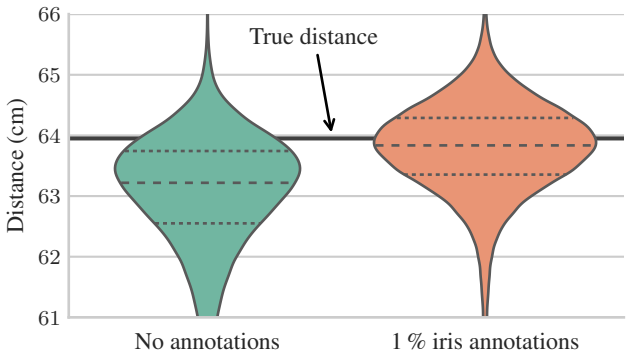


Figure 11: Estimated distances on the UnityEyes dataset, with and without iris annotations.

much better than a generic CNN, and is comparable to a person-specific CNN. The error for the uncalibrated model is about 2.2 times higher than the error for the calibrated model.

This is not a fair comparison, since training sets, test sets and network architectures differ between the three methods. These differences are necessary, since the three methods have different requirements, both in terms of the input data and the training sets. We do however believe these results support the idea that personal variations are well-described by a low-dimensional latent parameter space.

5.3. Distance estimation

As the NIR dataset and MPIIGaze only have gaze targets in the same plane as the camera, we could not learn meaningful distance estimates. The network could compensate for distance errors by changing the gaze direction. To investigate the feasibility of learning 3D gaze without annotated eye positions, we used the synthetic UnityEyes dataset, where the gaze targets could be located at different z-depths. We also compared the performance with and without glints, all else being equal. The results are shown in Figure 10. Compared to the NIR dataset, the errors are reduced by approximately a factor 1.7 without glints and 2.0 with glints. We see that the network has predicted correct gaze angles, even though it does not know the z-depth of the gaze targets.

While the gaze rays as such are close to the true gaze ray, there is nothing forcing the gaze origin to coincide with any physical feature on the eye, and the predicted gaze origins tend to be "floating" along the gaze ray. In an attempt to improve the prediction, we used the iris metadata to annotate a 1 % subset of the UnityEyes image with iris centers. We then added an additional term to the cost function, an L^2 loss on the distance between the iris center and the 2D gaze origin. The resulting distance estimates are shown in Figure 11. We see that even quite sparse iris annotations improve the consistency of the distance estimate.

6. Model-based gaze estimation

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 [4]. For a comprehensive review of model-based methods, we refer to Hansen and Ji [5].

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

Image processing methods detect the corneal reflection, 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 personal 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, these are three personal parameters. 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.

7. Conclusions

We propose a way to incorporate personal calibration into a deep learning model for video-based gaze estimation. Using our method, we show that an appearance-based gaze

tracking system with a single camera and a collocated illuminator can achieve performance similar to model-based, multi-camera, multi-illuminator systems. The number of personal parameters is low, about three per eye, and similar to the number predicted by geometrical models. When evaluated on the MPIIGaze dataset, our estimator performs better than deep learning methods with person-specific models.

Experiments on synthetic data suggest it would be possible to learn accurate 3D gaze (both origin and direction of gaze) without annotated eye positions.

Acknowledgement

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation

References

[1] 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. 2

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

[3] 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. 3, 6

[4] 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. 4, 7

[5] 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, 4, 5, 6, 7

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

[7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015. 4

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

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

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, Jan. 2014. 4

[11] 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*, pages 1821–1828, June 2014. 2, 3

[12] 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, 6

[13] 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. 2

[14] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, June 2015. 1, 2, 5, 6

[15] 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, 2

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