This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Monocular Free-head 3D Gaze Tracking with Deep Learning and Geometry Constraints

Haoping Deng*
Nanjing University
hoping_tang@outlook.com

Wangjiang Zhu†
Tsinghua University
wangjiang88119@gmail.com

Abstract

Free-head 3D gaze tracking outputs both the eye location and the gaze vector in 3D space, and it has wide applications in scenarios such as driver monitoring, advertisement analysis and surveillance. A reliable and low-cost monocular solution is critical for pervasive usage in these areas.

Noticing that a gaze vector is a composition of head pose and eyeball movement in a geometrically deterministic way, we propose a novel ***gaze transform layer*** *to connect separate head pose and eyeball movement models. The proposed decomposition does not suffer from* ***head-gaze correlation overfitting*** *and makes it possible to use datasets existing for other tasks. To add stronger supervision for better network training, we propose a two-step training strategy, which first trains sub-tasks with rough labels and then jointly trains with accurate gaze labels. To enable good cross-subject performance under various conditions, we collect a large dataset which has full coverage of head poses and eyeball movements, contains 200 subjects, and has diverse illumination conditions.*

Our deep solution achieves state-of-the-art gaze tracking accuracy, reaching 5.6° *cross-subject prediction error using a small network running at 1000 fps on a single CPU (excluding face alignment time) and* 4.3° *cross-subject error with a deeper network.*

1. Introduction

Gaze tracking has long been used in medical diagnoses [8], psychological studies [31] and human-computer interaction [11, 21, 27]. Recently, gaze tracking has also been applied to VR/AR devices for controlling [29] and foveated rendering [30]. Most existing gaze tracking systems need fixed head or head mounted devices. In fact, free-head 3D gaze tracking has wider applications in areas such as driver monitoring [41, 4, 12, 50], advertising [2, 52] and surveillance [28].

Some works have extended fixed-head solutions to allow for a small range of head movements by adding compensations [18, 20], but they cannot handle large head movements. Recent works [38, 51, 13] learned gaze as a joint function of head and eye features. However, their methods ignored the deterministic geometric constraints among head pose, eyeball movement and gaze vector, and forced this relationship to be learned from data, making it prone to overfitting the head-gaze correlation in training set.

In this work, we propose to separately model head pose and eyeball movement with two CNN networks, and then connect them with a "gaze transform layer". This decomposition avoids head-gaze correlation overfitting and makes it possible to take advantage of existing data for other tasks. For example, we can use head pose datasets to aid our head pose estimation and use fixed-head gaze tracking datasets for our eye model training.

For head pose estimation, we find that facial landmark based methods are ambiguous at both the within-subject level (as sensitive to facial expression) and the cross-subject level (as inter-person differences). So we propose to regress head poses directly from head image patches.

In order to apply stronger supervision to network training, we further propose a coarse-to-fine two-step training strategy which first trains separate models with coarse head pose and eyeball movement labels, then does joint training with accurate gaze labels. During the second step, the head pose and eyeball movement models adjust slightly to achieve cross-subject consistent head pose while keeping the composed gaze vector fixed.

To achieve robust cross-subject predictions in the wild, we need a large and diverse training set. In this work, we build a 200-subjects dataset with full coverage of different head poses and eyeball movements, various lighting conditions, and includes glasses occlusions and reflections.

Our contributions can by summarized as follows: 1). We analyze head-gaze correlation overfitting and head pose ambiguity issues which are ignored in previous works [38, 51,

*This work was done when Haoping Deng was an intern at SenseTime.
†This work was done when Wangjiang Zhu worked as the gaze project leader at SenseTime.

13]; 2). We propose separate head and eye modelings and a coarse-to-fine training strategy to solve the two issues; 3). A large gaze tracking dataset is introduced, which has a good coverage of different head poses, eyeball movements, lighting conditions, and glasses occlusions and reflections.

2. Related Works

Gaze tracking has been studied for decades [6, 3]. Most early works are **model-based** as they build geometric models of the eyeball [47, 32, 48] or iris contour [42, 45]. Model-based gaze tracking has been successfully used in many commercial gaze tracking systems. However, most of them need expensive specialized equipments with infrared lighting, and all of them have limited working distance.

Because of these limitations, recent research works focus more on **appearance-based** methods as they have the potential to predict gaze direction with low-resolution images given a large volume of training data. Appearance-based methods learn a mapping function from image patches to gaze predictions. Different mapping functions have been explored, including Neural Networks [1, 46, 51, 13], local interpolation [39, 19], Gaussian process regression [37], and random forests [38]. Zhang et al. [51] showed that Convolutional Neural Network (CNN) can achieve better performance than other mapping methods.

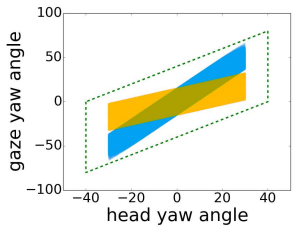
Given those previous studies and the success of deep learning, we use deep CNNs for gaze mapping in this work.

2.1. Free-head 3D gaze tracking

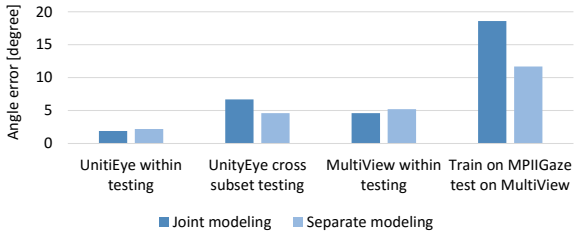
Free-head 3D gaze tracking outputs both the starting location, i.e. eye coordinates, and gaze direction in 3D space. This is different from early works [6, 3] which only predict gaze points on a screen and provide no information on eye locations and gaze vector. Kyle et al. [13] and Huang et al. [9] proposed free-head gaze tracking solutions by using a large volume of training data captured from daily life. Unfortunately, they can only predict gaze intersections on screen rather than 3D gaze vectors. Free-head 3D gaze tracking is necessary for applications which need to know what objects or regions a person is looking at. This can be achieved by calculating the intersection point between the 3D gaze vector and objects in 3D scene.

To handle different eye appearances at various head poses, previous works have evaluated compensation methods [18] and warping methods [20, 26]. Although those methods can help reduce model complexity, they cannot work well for large head movements. To overcome these limitations, recent works [38, 51, 9, 13] relies on large datasets to cover various head poses and gaze directions.

Our monocular free-head 3D gaze tracking solution estimates eye location by solving a PnP problem [16] and predicts a 3D gaze vector using the networks shown in Figure 4.



(a) Subsets from UnityEye [44]



(b) Testing errors.

Figure 1: (a) Two subsets sampled from UnityEye [44] for cross-set evaluation; (b) Compare joint modeling [51] with separate modeling on synthetic data and real life datasets.

2.2. Deep learning with geometry constraints

Recent years have witnessed great success of deep learning in image understanding, including image classification [14], object detection [33] and semantic segmentation [17]. However, when moving from the 2D image plane to the 3D world, we still need the help of classical geometric principles. Some recent works [53, 40, 15] have explored combining the multi-view geometric relationships with data-driven learning processes. Particularly, Zhou et al. [53] integrated the geometric constraints among depth map, camera motion and pixel correspondences with CNN network training.

For free-head gaze tracking, there is a deterministic relationship among head pose, eyeball movement, and gaze vector, as shown in Eq. (2). We propose to use this geometric relationship to decompose gaze estimation into separate head pose and eyeball movement predictions. Our solution is shown to be less prone to overfitting and can make use of existing datasets of other tasks.

3. Gaze Tracking Issues

In this section, we will analysis two critical issues in free-head gaze tracking, including head-gaze correlation overfitting and head pose ambiguity.

3.1. Head-gaze correlation overfitting

As a gaze vector is a combination of head pose and eyeball movement, it is correlated with one's head pose in nature. Figure 10 compares different head-gaze correlation

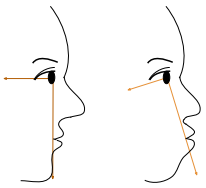


Figure 2: Within-subject head pose ambiguity: when a person pouts his/her mouth, the head pose will change even if the skull is fixed.

patterns of several gaze tracking datasets. As we can see, these datasets present various gaze-head distribution patterns, which depend on the data collection setting.

Previous works, such as [38, 51, 13], learned the gaze vector as a joint function of head and eye features, which can be denoted as $g = f(e, h)$, where e denotes the eye feature, h indicates the head pose label [51] or head features [13], and $f(.)$ is the joint function. In particular, both [51] and [13] defined $f(.)$ as a linear function, while [38] used a random forest regressor for $f(.)$.

We argue that learning $f(.)$ in a data-driven fashion is problematic, as the training set may exhibit biased head-gaze correlation pattern, thus leading to an overfitted $f(.)$.

To show the existence of this overfitting, we experimentally compare the performances of joint modeling and separate modeling under within-dataset and cross-dataset settings. For joint modeling, we follow [51] to model gaze as a linear function of the head pose label and eye patch features, using our tiny-AlexNet as the CNN structure; for separate modeling, we remove the linear function, and train a eyeball movement model with the same network structure as in joint modeling, then predict gaze vector by combining head pose and eyeball movement prediction following Eq. (2).

We do comparisons on both synthetic and real life data. For synthetic data, we first divide millions of eye images generated by UnityEye [44] into training and testing parts in a ratio of $4:1$, then choose two subsets with different yaw distributions from both training and testing sets respectively. Figure 1a illustrates the two subsets selected by different slopes ($\sqrt{3}$ and $1/\sqrt{3}$) in the yaw space. Both sets have head poses which range from $-30°$ to $30°$), and eyeball range of $30°$ around the middle line. Each subset in the training part contains approximately $300,000$ images. For realistic data, we train on MPIIGaze dataset [51] and test on UT Multiview dataset [38].

Figure 1b shows the results, from which we can see: 1) under the within-dataset setting, joint modeling [51] performs slightly better than separate modeling; 2) under the cross-dataset setting, joint modeling [51] has much lower performance than separate modeling. This is a clear sign of overfitting.

Figure 3: Cross-subject head pose ambiguity introduces inconsistent eye appearance. These images are sampled from MultiView [38], which share the **same head and eye labels**. However, they present different appearances: the first three are looking upward but the last two are looking forward.

To show what happened, we inspected the weights for head pose learned in the joint model [51] trained on MPIIGaze dataset [51]. We found that the weights are close to zero, indicating that gaze vectors are irrelevant to head poses. Of course, this is ridiculous. After checking the joint distribution of head pose and gaze direction as shown in Figure 10, we find this is not suprising given the horizontal distribution shape of MPIIGaze dataset [51].

In Section 6.3, we will further show that our separate head/eye modeling associated with the two-step training strategy achieves superior performance to joint modeling methods [51, 13].

3.2. Head pose ambiguity

It is intuitive to define head pose based on facial landmark locations. Sugano et al. [38] first recovered 3D facial landmarks from multi-view images captured simultaneously, then defined a head coordinate as shown in Figure 5, based on which head pose was defined as the rotation between camera coordinate and head coordinate. Zhang et al. [51] used the same definition of head pose, but cast head pose estimation as a PnP problem [16] as they used single camera for data collection. We argue that facial landmark based head pose estimation is sensitive to facial expression and face configuration, thus making head pose ambiguous at both within-subject level and cross-subject level:

1. **Within-subject ambiguity.** Facial landmark location will change with facial expressions, thus making the within-subject head pose unstable. Figure 2 illustrates that head pose changes when a person pouts his/her mouth, even if the skull and the eyeballs are fixed.

2. **Cross-subject ambiguity.** Inter-person face configuration differences will introduce more ambiguities. As shown in Figure 3, those differences lead to inconsistent eye appearances for the same eyeball movement label. This inconsistency will confuse data-driven learning processes. For monocular gaze tracking, this ambiguity will be even severer: for example, when solving the PnP problem for a long face with an average 3D landmark template, we will get a biased pitch angle.

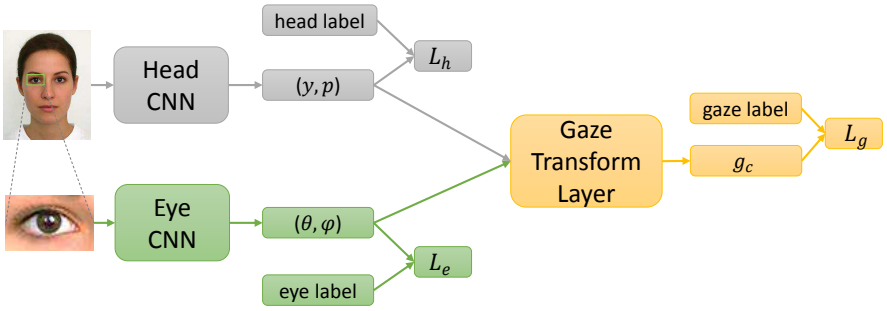


Figure 4: Our network structure. We use two CNNs to model head pose and eyeball movement respectively. A **gaze transform layer** is proposed to aggregate them into gaze prediction and can be trained in an end-to-end way. Gaze transform layer is defined in Eq. (2) and contains no learnable parameters.

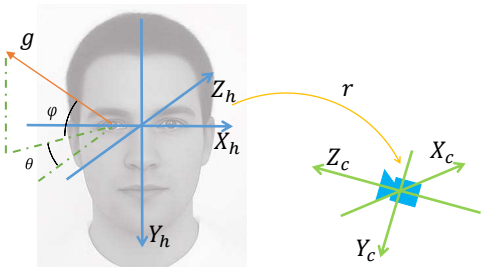


Figure 5: An illustration of head (blue) and camera (green) coordinate system. Gaze vector ($\mathbf{g}$) can be represented as $\mathbf{g_h}$ and $\mathbf{g_c}$ in head and camera coordinate system respectively. The relationship between $\mathbf{g_h}$ and $\mathbf{g_c}$ is defined by head pose $\mathbf{R}$, as illustrated in Equation 2.

4. Our Approach

This section will first introduce our gaze tracking approach which has solved the two issues analyzed in Section 3. Particularly, in order to avoid head-gaze correlation overfitting, we separately model head pose and eyeball movement and then aggregate them into gaze vector using a **gaze transform layer**; in order to overcome head pose ambiguity introduced by landmark based methods, we propose to regress head pose from appearance directly using a CNN network. Then a two-step training strategy is proposed for better network training.

Our separate modeling approach associated with the two-step training strategy has the following advantages:

1. **Less overfitting risk.** By using the deterministic geometric relationship, our model will not suffer from the head-gaze distribution bias in training set;

2. **More training data.** We can use existing head pose datasets (e.g. [5, 49]) for head pose pre-training, and use eye images from head mounted devices (e.g. [23, 22]) for eyeball movement pre-training.

3. **More explainable and flexible.** Providing separate head pose and eyeball movement predictions not only makes it easier for gaze tracking diagnoses and improvements, but also enables scenarios where only head poses are needed or only eyeball movements are needed (e.g. head mounted devices).

4.1. Gaze factorization

A gaze vector is composed of eyeball movement and head pose, as is illustrated in Figure 5. We use $\mathbf{g}$ to denote the physical gaze vector, which can be represented as $\mathbf{g_h}$ and $\mathbf{g_c}$ in head and camera coordinate system respectively.

Head coordinate system. We define the rough head coordinate system based on facial landmarks as in [38, 51]. It is necessarily rough at this stage due to inherent ambiguities of the head pose. The accurate head coordinate system that is within-subject stable and cross-subject consistent will be learned automatically by our gaze transform layer in the second training step.

Eyeball movement. As shown in Figure 5, eyeball movement is defined as the angles between the gaze vector $\mathbf{g}$ and head coordinate axes, and is denoted as (θ, ϕ), where θ and ϕ mean horizontal and vertical rotation angles respectively. In fact, eyeball movement depicts a gaze vector in head coordinate system as:

$$\mathbf{g_h} = [-\cos(\phi)\sin(\theta), \sin(\phi), -\cos(\phi)\cos(\theta)]^T \quad (1)$$

Head pose. Head pose reflects the rotation matrix $\mathbf{R}$ between the head and camera coordinate systems. As explained in Section 5.2, after data normalization, the 3DOF head pose can be normalized to 2DOF as (y, p), where y is the yaw and p is the pitch angles in the normalized spherical coordinate system. (y, p) and $\mathbf{R}$ can be interconverted as shown in [38]. Herein we denote this as $\mathbf{R} = f(y, p)$.

dataset	#subject	#pose	#target	#image
[24]	20	1	16	videos
[43]	20	19	2-9	2220
[35]	56	5	21	5880
[25]	16	cont.	cont.	videos
[38]	50	8+syn.	160	64,000
[51]	15	cont.	cont.	213,659
Ours	**200**	**cont.**	**cont.**	**240,000**

Table 1: Comparison with 3D annotated Datasets.

Gaze transform layer. The gaze vector $\mathbf{g_c}$ is defined in camera coordinate system, and the transformation between $\mathbf{g_c}$ and $\mathbf{g_h}$ is defined by the head pose $\mathbf{R}$:

$$\mathbf{g_c} = \mathbf{R}\mathbf{g_h}, \tag{2}$$

We build a neural network layer named the "gaze transform layer" to encode the transformation from $\mathbf{g_h}$ to $\mathbf{g_c}$. The forward operation of this layer is described in Eq. (2), and the backward operation is defined as:

$$\begin{aligned} \frac{\partial \mathbf{g_c}}{\partial y} &= \frac{\partial \mathbf{g_c}}{\partial \mathbf{R}}\frac{\partial \mathbf{R}}{\partial y}, \quad \frac{\partial \mathbf{g_c}}{\partial p} = \frac{\partial \mathbf{g_c}}{\partial \mathbf{R}}\frac{\partial \mathbf{R}}{\partial p}, \\ \frac{\partial \mathbf{g_c}}{\partial \theta} &= \frac{\partial \mathbf{g_c}}{\partial \mathbf{g_h}}\frac{\partial \mathbf{g_h}}{\partial \theta}, \quad \frac{\partial \mathbf{g_c}}{\partial \phi} = \frac{\partial \mathbf{g_c}}{\partial \mathbf{g_h}}\frac{\partial \mathbf{g_h}}{\partial \phi}. \end{aligned} \tag{3}$$

Note that our gaze transform layer contains no learnable parameters; it simply acts as a gradient conductor between $\mathbf{g_c}$ and the four inputs.

4.2. Training strategy

Given the network structure shown in Figure 4, we can directly use gaze label and apply gaze loss L_g to guide the learning process of the two CNN networks. In fact, this training strategy works fairly well as shown in Figure 11. Noticing that we can also add supervision information for the two separate CNNs, we hypothesize that adding stronger supervision at these mid-points may lead to better network learning. However, as analyzed in Section 3.2, facial landmark based methods could not provide accurate, cross-subject consistent head pose labeling. So we propose a two-step training strategy to first train the two separate CNNs (L_e + L_h) with rough labels, and then fine-tune the whole network using gaze label (L_g) to achieve within-subject stable and cross-subject consistent head pose.

Two-step training. In the first step, we train eyeball movement and head pose models using the rough labels estimated from facial landmarks detected by [54]; In the second step, we fine tune the 2-branch network shown in Figure 4 using only accurate gaze labels.

dataset	distribution	device ind.	3D anno.
[9]	wide	no	no
[13]	wide	no	no
[38]	narrow	yes	yes
[51]	narrow	yes	inaccurate
Ours	**wide**	**yes**	**yes**

Table 2: Attribute comparison for some recent datasets.

In the first step, parameters of the two CNNs are optimized to give a coarse estimation of head poses and eyeball movements, so they can achieve fairly good though still inaccurate gaze predictions. In the second step, the parameters will adjust slightly to achieve within-subject stable and cross-subject consistent head pose. Experimental results in Figure 11 show that this two-step training strategy significantly improves gaze accuracy.

We need to point out that our second training step is different from previous joint modeling methods such as [51, 13]. This is because our gaze is represented as a fixed function of head pose and eyeball movement, and the gaze transform layer contains no learnable parameters.

5. Data Collection

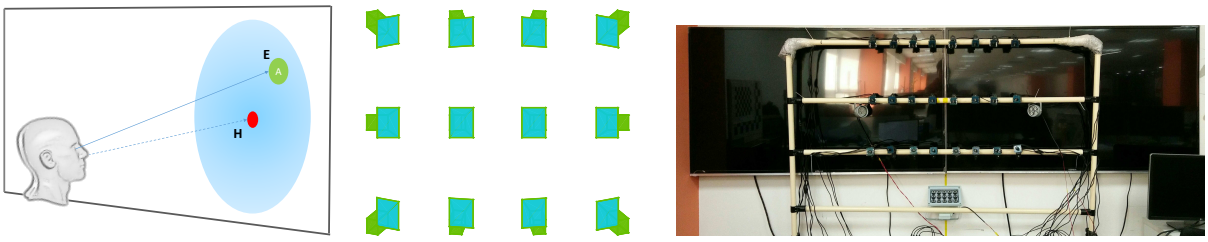
We argue that a good gaze tracking dataset should meet the following requirements:

1. **Wide distribution.** The dataset should have full coverage of head poses, eyeball movements and their combinations; it should include a large number of subjects for good cross-subject performances; and it should take illumination variations, occlusions and glasses reflection into consideration.

2. **Device independent.** A model learned from the training set should be easily deployed to other devices with distinct camera parameters and/or target screen.

3. **3D annotation.** Annotating head pose and gaze vector in the 3D space makes it possible to predict a 3D gaze vector rather than only gaze intersection with the screen. Thus the system can be used to predict gaze intersection with any surrounding objects.

Table 2 uses these criteria to compare our dataset with four largest ones. TabelGaze [9] and GazeCapture [13] collected a huge number of videos and images in daily life using tablet or smart phone. However, they are not device independent as lack of camera intrinsic parameters, and they could not be accurately 3D annotated because of using monocular camera for data collection. In UT Multiview dataset [38], images with fixed head pose were first collected, and then virtual cameras were used to synthesize



(a) head target and eyeball target (b) 12 cameras config (c) our collection environment

Figure 6: Our data collection setup. (a). Head pose targets and eyeball targets are used to guide head pose and eyeball movement respectively; (b). 12 synchronized cameras are used for wide head pose range and also provide 3D annotations; (c). Two TV screens are used for target presentation, and we add 3 extra light sources to generate random lighting combinations.

(a) one eye in 12 cameras (b) different eyeball angle

Figure 7: (a). Eye images by 12 synchronized cameras from different directions. (b). Example images showing large eyeball movement range.

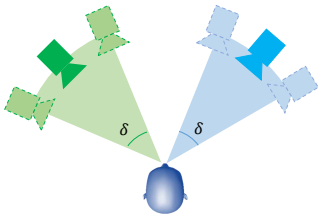


Figure 8: A person's head pose range δ is enlarged to $N\delta$ by N cameras with unparallel optical axes.

images from other view points. They have relatively large head pose and eyeball movement ranges, but the lighting conditions are almost fixed. In addition, no occlusions from glasses or hair is included due to the requirements of the 3D reconstruction algorithm. MPIIGaze dataset [51] collected a large number of pictures using laptop cameras during the subjects' daily life to include extreme lighting conditions and glasses occlusions. However, limited by the usage habits of the subjects, MPIIGaze dataset [51] contains a small gaze range as shown in Figure 10. In addition, because of the use of a monocular camera for data collection, they cannot get accurate 3D annotations.

Table 1 compares dataset size and diversity with existing 3D annotated datasets. As we can see, our dataset is large in both the subject number and the image number aspects.

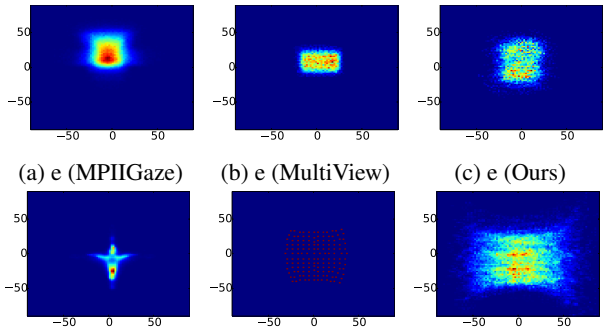
In the following, we will introduce our data collection setup and procedures to show how to achieve a large dataset with wide distribution of head poses, eyeball movements and illuminations.

5.1. Collection Procedure

Figure 6 shows the setup of our data collection system. We use two TV screens (60 inches, screen size of a single screen: 1345mm * 780mm) for target presentation, and a camera array for full head pose range coverage. The multi-camera systems are pre-calibrated, and the positions of TV screens in world coordinate are calibrated with a mirror [34]. We guide participants to adjust their head and eyeballs according to two types of targets, i.e., head pose targets and eyeball targets as shown in Figure 6a. In fact, during collection, we repeat: 1) showing a head pose target randomly and asking the participant to turn his/her head towards it; 2) then showing eyeball targets randomly around a previous head pose target and asking the participant to track them by moving his/her eyeballs. Note that, head pose targets play an ancillary role and will not be used for annotation generation, so the deviations during head pose turning will not influence our data accuracy.

To ensure robust gaze predictions at the testing stage, it is critical to get full coverage of eyeball movements, head poses and their combinations. To get high quality data without noise labeling, we need to make sure participants are looking at the appointed targets at the time of capture. We make an effort to assure these criteria in the following manner:

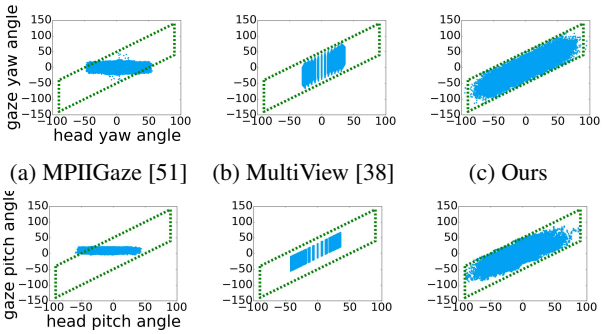
Full head pose coverage. Full head pose coverage is achieved by the combination of head movements and the use of multiple cameras. In fact, if N cameras are placed with unparallel optical axes, they will contribute to N samples with different head poses at a single point in time. If we further allow participants to move their head within a range, the N cameras will contribute to N regions within the head pose space. We give a 1D example to illustrate this in Figure 8: when the participant moves his head within a small range of δ, we can get $\delta \times N$ head pose range. Note that,



(a) e (MPIIGaze) (b) e (MultiView) (c) e (Ours)

(d) h (MPIIGaze) (e) h (MultiView) (f) h (Ours)

Figure 9: Eyeball movement (first line) and head pose (second line) distribution on MPIIGaze dataset [51], MultiView dataset [38] and ours.



(a) MPIIGaze [51] (b) MultiView [38] (c) Ours

(d) MPIIGaze [51] (e) MultiView [38] (f) Ours

Figure 10: Head-gaze distribution on MPIIGaze dataset [51], UT MultiView dataset [38] and our dataset. Blue dots represent data samples and the parallelograms in green dashed line illustrate the possible region for the distribution of eyeball movement ($[-50°, 50°]$) and head pose ($[-90°, 90°]$).

it is equivalent to have "fixed cameras with moving head" v.s. "moving cameras with fixed head", so by allowing head movements during data collection, we will get a continuous distribution of camera location.

Full eyeball movement coverage. The large TV screens make it possible to sample eyeball targets far from associated head targets, thus leading to a large eyeball movement range. Example images showing the large eyeball movement range can be observed in Figure 7b. To achieve full eyeball movement around each head pose, we need to symmetrically sample eyeball targets around each head pose target.

Figure 9 shows that our dataset has the largest head pose and eyeball movement range; and Figure 10 shows that we have full coverage of possible head-gaze distribution. In comparison, MPIIGaze [51] covers the smallest region due to relatively fixed laptop usage habits; and UT MultiView [38] has relatively large coverage by using multiple virtual cameras.

Lighting coverage. In addition to the natural lighting in the room, we add three light sources and control the voltage of each light source in a continuous and random way, thus leading to a huge number of lighting combinations.

Data verification. For head pose verification, we run online head pose tracking utilizing the face alignment method in [54] to ensure that the actual head pose is close to the appointed head pose target. If this check is not passed, the collection program will not start.

For gaze direction verification, we display changing characters at the eyeball target location, and ask participants to press a key immediately when a certain character shows up. This procedure effectively makes participants focus their sights on appointed locations.

5.2. Data processing

In general, object pose has 6 DOF relative to the camera, including 3 DOF of translation and 3 DOF of rotation. Normalizing training samples to a lower space can significantly reduce the required number of training samples, thus saving data collection costs. In this work, we follow [38] to do data normalization.

We detect facial landmarks as in [54] and reconstruct their 3D coordinates using multiple cameras. Note that we only need multi-view reconstruction during the data collection stage. At the testing stage, a monocular camera is enough. With 3D facial landmarks, we normalize eye patches following the method introduced in [38]. The underlying principle is that we can warp the original image following a homography transformation to simulate camera rotation (around its pinhole) and zoom in/out. Particularly, the normalization is done in two steps: First, rotate the camera around its pinhole to a status where 1) the x axis of head coordinate system lies in the camera's X-O-Z plane and 2) the eye center lies in the Z axis of the camera. Then, zoom the camera in or out to a new focal length to get a fixed magnification ratio (ratio of the size on image to the size in real 3D world).

After this normalization, the head pose is reduced from 6DOF to 2DOF. So we can represent the head pose using the longitude and latitude angles in the spherical coordinate system. Please refer to the original paper for more details.

6. Experiments

6.1. Network designing

For speed consideration, we propose a tiny-AlexNet network, which is a simplified version of the original

AlexNet [14]. Our tiny-Alexnet can run a forward pass within 1ms on a single Intel Xeon 2.4GHz CPU. In order to see the influence of deeper structure and more parameters, we compare tiny-AlexNet with BN-Inception [10] network. In the following experiments, if not explicitly specified, tiny-AlexNet is used as the default structure.

Our tiny-AlexNet has the same number of layers as the original AlexNet, but with fewer channels. We also change the Local Response Normalization [14] into batch normalization [10] due to its fast convergence speed and training stability. The detailed structure of tiny-AlexNet can be described as: C(3,1,6)-BN-PReLU-P(2,2)-C(3,1,16)-BN-PReLU-P(2,2)-C(3,1,24)-BN-PReLU-C(3,1,24)-BN-PReLU-C(3,1,16)-BN-PReLU-P(2,2)-FC(256)-PReLU-FC(128)-PReLU-FC(2), where C(k,s,c) means a convolution layer with kernel size k, stride s and channel number c; P(k,s) means max pooling with kernel size k and stride s; PReLU is defined by [7]; and FC means fully connected layer. Euclidean loss is used both for eyeball movement and head pose regression.

Images are histogram equalized to alleviate illumination differences before inputting to the CNN. For tiny-AlexNet, images are first resized to 72×72 and then cropped to 62×62. For BN-Inception, images are first resized to 256×256 and then cropped to 224×224.

We train the network using classical Stochastic Gradient Descent (SGD). For the first training step (L_e and L_h), we use an initial learning rate of 0.001 and 25,000 iterations at a batch size of 256, and we decrease the learning rate by $1/10$ at the iterations of $12,000$ and $20,000$; For the second training step (L_g), we use the same parameters but skip the first $12,000$ iterations to jointly fine-tune the whole networks.

6.2. Two-step training strategy

In this section, we compare different training strategies and study whether the proposed two-step training strategy can improve gaze prediction accuracy.

Our two-step learning strategy first trains the 2-branch network with $L_e + L_h$, leading to a sub-optimal solution which suffered from head pose ambiguity. Then in the second step, only the gaze loss L_g is applied, as we have accurate gaze ground truth labeling.

Figure 11 compares the two-step training strategy with other different training strategies. From the figure, we can observe that: 1) head pose ambiguity has a significant impact on gaze accuracy with the worse case occurring when only $L_e + L_h$ is used; 2) simply using L_g for a one-step training also acquires relatively good performance, indicating that the 2-branch network can learn good eyeball movement and head pose models given only gaze supervision; 3) two-step training can significantly improve gaze prediction, especially for large BN-Inception network where more parameters have to be learned.

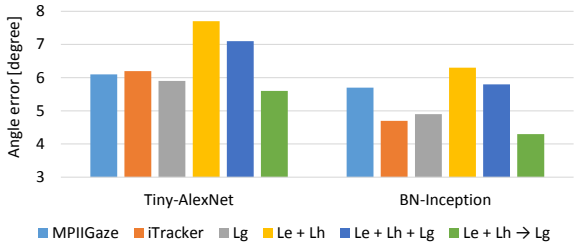


Figure 11: Performance comparison among MPIIGaze method [51], iTracker method [13] and our different training strategies. For iTracker [13] method we stack the feature vectors of eye and head, and then use a fully connected layer for gaze regression. As we can see our separate modeling associated with the two-step training strategy achieves the best performance.

6.3. Comparison with state-of-the-art

Since public 3D annotated datasets [38, 51] only provide eye patches rather than the whole face image, we cannot directly compare our method with other state-of-the-art methods on those datasets. So we compare our solution with state-of-the-art methods on our dataset. We randomly divide the 200 subjects into 5 groups and report the 5-fold cross validation results for different methods.

As Zhang et al. [51] has already shown, CNN-based gaze tracking outperforms other mapping functions (e.g. random forest [36], kNN, adaptive linear regression and support vector regression), herein we only compare with CNN-based gaze tracking. Particularly, we compare with the MPIIGaze method [51] which learns gaze as a linear function of head pose and eye features, and with the iTracker method [13] which learns gaze as a function of eye feature and head feature. Note that, for fair comparison, we change the network structure of [51] and [13] into the same as ours, i.e., tiny-AlexNet or BN-Inception. Figure 11 shows that our separate modeling associated with the two-step training strategy outperforms MPIIGaze method [51] and iTracker method [13].

7. Conclusions

In this work, we analyzed the head-gaze correlation overfitting and head pose ambiguity issues which are ignored by existing free-head gaze tracking works, then propose a solution using separate modeling and a two-step training strategy to solve these two issues. To enable free-head gaze tracking under various conditions, a large dataset is collected with full coverage of head pose, eyeball movements, illuminations and occlusions. Experiments validate that our methods are less likely to overfit. We achieve state-of-the-art gaze tracking accuracy.

References

[1] S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. Technical report, DTIC Document, 1994.

[2] A. T. Duchowski. A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers*, 34(4):455–470, 2002.

[3] O. Ferhat and F. Vilariño. Low cost eye tracking: The current panorama. *Computational intelligence and neuroscience*, 2016, 2016.

[4] L. Fletcher and A. Zelinsky. Driver inattention detection based on eye gazeroad event correlation. *The international journal of robotics research*, 28(6):774–801, 2009.

[5] N. Gourier, D. Hall, and J. L. Crowley. Estimating face orientation from robust detection of salient facial structures. In *FG Net Workshop on Visual Observation of Deictic Gestures*, volume 6, 2004.

[6] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(3):478–500, Mar. 2010.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[8] P. S. Holzman, L. R. Proctor, D. L. Levy, N. J. Yasillo, H. Y. Meltzer, and S. W. Hurt. Eye-tracking dysfunctions in schizophrenic patients and their relatives. *Archives of general psychiatry*, 31(2):143–151, 1974.

[9] Q. Huang, A. Veeraraghavan, and A. Sabharwal. Tabletgaze: A dataset and baseline algorithms for unconstrained appearance-based gaze estimation in mobile tablets. *arXiv preprint arXiv:1508.01244*, 2015.

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[11] R. Jacob and K. S. Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind*, 2(3):4, 2003.

[12] Q. Ji and X. Yang. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging*, 8(5):357–377, 2002.

[13] 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.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. *arXiv preprint arXiv:1702.02706*, 2017.

[16] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[18] F. Lu, T. Okabe, Y. Sugano, and Y. Sato. A head pose-free approach for appearance-based gaze estimation. In *BMVC*, pages 1–11, 2011.

[19] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Inferring human gaze from appearance via adaptive linear regression. In *2011 International Conference on Computer Vision*, pages 153–160. IEEE, 2011.

[20] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Head pose-free appearance-based gaze sensing via eye image synthesis. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1008–1011. IEEE, 2012.

[21] P. Majaranta and A. Bulling. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*, pages 39–65. Springer, 2014.

[22] M. Mansouryar, J. Steil, Y. Sugano, and A. Bulling. 3d gaze estimation from 2d pupil positions on monocular head-mounted eye trackers. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 197–200. ACM, 2016.

[23] C. D. McMurrough, V. Metsis, D. Kosmopoulos, I. Maglogiannis, and F. Makedon. A dataset for point of gaze detection using head poses and eye images. *Journal on Multimodal User Interfaces*, 7(3):207–215, 2013.

[24] C. D. McMurrough, V. Metsis, J. Rich, and F. Makedon. An eye tracking dataset for point of gaze detection. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 305–308. ACM, 2012.

[25] K. A. F. Mora, F. Monay, and J.-M. Odobez. Eyediap: a database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 255–258. ACM, 2014.

[26] K. A. F. Mora and J.-M. Odobez. Gaze estimation from multimodal kinect data. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–30. IEEE, 2012.

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

[28] S. S. Mukherjee and N. M. Robertson. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Trans. Multimedia*, 17(11):2094–2107, 2015.

[29] S. Nilsson. Interaction without gesture or speech–a gaze controlled ar system. In *Artificial Reality and Telexistence, 17th International Conference on*, pages 280–281. IEEE, 2007.

[30] D. Pohl, X. Zhang, and A. Bulling. Combining eye tracking with optimizations for lens astigmatism in modern wide-angle hmds. In *2016 IEEE Virtual Reality, VR 2016, Greenville, SC, USA, March 19-23, 2016*, pages 269–270, 2016.

[31] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372, 1998.

[32] M. Reale, T. Hung, and L. Yin. Viewing direction estimation based on 3d eyeball construction for hri. In *2010 IEEE Com-*

puter Society Conference on Computer Vision and Pattern Recognition-Workshops, pages 24–31. IEEE, 2010.

[33] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[34] R. Rodrigues, J. P. Barreto, and U. Nunes. Camera pose estimation using images of planar mirror reflections. In *European Conference on Computer Vision*, pages 382–395. Springer, 2010.

[35] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar. Gaze locking: passive eye contact detection for human-object interaction. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 271–280. ACM, 2013.

[36] L. B. Statistics and L. Breiman. Random forests. In *Machine Learning*, pages 5–32, 2001.

[37] Y. Sugano, Y. Matsushita, and Y. Sato. Appearance-based gaze estimation using visual saliency. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):329–341, 2013.

[38] Y. Sugano, Y. Matsushita, and Y. Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[39] K.-H. Tan, D. J. Kriegman, and N. Ahuja. Appearance-based eye gaze estimation. In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 191–195. IEEE, 2002.

[40] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[41] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi. Driver gaze tracking and eyes off the road detection system. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2014–2027, 2015.

[42] J.-G. Wang, E. Sung, and R. Venkateswarlu. Eye gaze estimation from a single image of one eye. In *ICCV*, pages 136–143. IEEE Computer Society, 2003.

[43] U. Weidenbacher, G. Layher, P.-M. Strauss, and H. Neumann. A comprehensive head pose and gaze database. In *Intelligent Environments, 2007. IE 07. 3rd IET International Conference on*, pages 455–458. IET, 2007.

[44] 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 and Applications*, pages 131–138, 2016.

[45] H. Wu, Q. Chen, and T. Wada. Conic-based algorithm for visual line estimation from one image. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 260–265. IEEE, 2004.

[46] L.-Q. Xu, D. Machin, and P. Sheppard. A novel approach to real-time non-intrusive gaze finding. In *BMVC*, pages 1–10, 1998.

[47] H. Yamazoe, A. Utsumi, T. Yonezawa, and S. Abe. Remote and head-motion-free gaze tracking for real environments with automated head-eye model calibrations. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–6. IEEE, 2008.

[48] 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 *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 245–250. ACM, 2008.

[49] X. Zabulis, T. Sarmis, and A. A. Argyros. 3d head pose estimation from multiple distant views. In *BMVC*, volume 1, page 6, 2009.

[50] Y. Zang and H. Liu. A real-time video-based eye tracking approach for driver attention study. *Computing and Informatics*, 31(4):805–825, 2012.

[51] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[52] X.-b. Zhang, C.-T. Fan, S.-M. Yuan, and Z.-Y. Peng. An advertisement video analysis system based on eye-tracking. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 494–499. IEEE, 2015.

[53] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[54] S. Zhu, C. Li, C. C. Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *CVPR*, pages 4998–5006. IEEE Computer Society, 2015.