RGBD Based Gaze Estimation via Multi-task CNN

Dongze Lian,[1*] Ziheng Zhang,[1*] Weixin Luo,[1] Lina Hu,[1] Minye Wu,[1]
Zechao Li,[2] Jingyi Yu,[1] Shenghua Gao[1†]
[1]ShanghaiTech University, [2]Nanjing University of Science and Technology
{liandz, zhangzh, luowx, huln, wumy}@shanghaitech.edu.cn,
{zechao.li}@njust.edu.cn, {yujingyi, gaoshh}@shanghaitech.edu.cn

Abstract

This paper tackles RGBD based gaze estimation with Convolutional Neural Networks (CNNs). Specifically, we propose to decompose gaze point estimation into eyeball pose, head pose, and 3D eye position estimation. Compared with RGB image-based gaze tracking, having depth modality helps to facilitate head pose estimation and 3D eye position estimation. The captured depth image, however, usually contains noise and black holes which noticeably hamper gaze tracking. Thus we propose a CNN-based multi-task learning framework to simultaneously refine depth images and predict gaze points. We utilize a generator network for depth image generation with a Generative Neural Network (GAN), where the generator network is partially shared by both the gaze tracking network and GAN-based depth synthesizing. By optimizing the whole network simultaneously, depth image synthesis improves gaze point estimation and vice versa. Since the only existing RGBD dataset (EYEDIAP) is too small, we build a large-scale RGBD gaze tracking dataset for performance evaluation. As far as we know, it is the largest RGBD gaze dataset in terms of the number of participants. Comprehensive experiments demonstrate that our method outperforms existing methods by a large margin on both our dataset and the EYEDIAP dataset.

Introduction

Gaze estimation is an important task and has wide applications in human-computer interaction (Majaranta and Bulling 2014), visual behavior analysis (Morimoto and Mimica 2005) and psychological studies (Rayner 1998). More recent studies have focused on appearance-based estimation (Zhang et al. 2015; Krafka et al. 2016; Zhang et al. 2017; Zhu and Deng 2017) as a vehicle for general gaze estimation. Unlike model-based methods, appearance-based methods achieve satisfactory performance and at the same time maintain a user-friendly data acquisition procedure, *i.e.*, without imposing additional priors on face poses or conducting elaborate system calibrations. Despite being a desirable gaze estimator, existing solutions are still sensitive to head pose, illumination inconsistencies, occlusions, low image quality, *etc*. In particular, accuracy in gaze estimation still varies significantly across subjects.

In light of the recent success of Convolutional Neural Networks (CNNs) in various computer vision tasks, a number of approaches have been proposed to leverage CNNs for appearance-based gaze estimation (Krafka et al. 2016; Zhang et al. 2017). They observe that the gaze point of a person depends on 3D eye position centered at the camera and gaze direction whereas the gaze direction further depends on the head and eyeball poses (Zhang et al. 2017) [1]. The 3D eye position imposes significance because even with the same gaze direction, a change of distance between the eye and the screen would change the gaze point on the screen target, as shown in Figure 1. To obtain 3D eye positions, (Krafka et al. 2016; Zhang et al. 2017) employed face grid to indicate the face region in the image. In reality, faces of different subjects can vary greatly in size and shape that a uniform grid is insufficient to describe.

Another challenge lies in head pose representations. A number of previous approaches (Krafka et al. 2016; Zhu and Deng 2017; Ranjan, De Mello, and Kautz 2018) attempt to encode head pose information by analyzing the RGB color image, *e.g.*, to introduce gaze transform layers or to use branched head pose models. Improvements brought by these approaches are still limited due to the lack of depth information. In this paper, we introduce depth-based approach to simultaneously address the 3D eye positions and head pose representation problems.

Using RGBD images for gaze tracking is not completely new (Mora and Odobez 2013; Xiong et al. 2014). However, the latest approaches only use a sparse set of points with depth information due to the limitations of the depth sensor. These points, although useful, are not sufficiently robust to reconstruct the head pose. Ideally, data-driven methods can more robustly infer the pose by leveraging learning-based approaches. However, so far only a single RGBD dataset is readily available to the public and the dataset itself only contains a very small number of participants - 16 with 12 males and 4 females. The study by Krafka *et al.* shows that more participants can clearly improve gaze tracking performance (Krafka et al. 2016). On the dataset front, we first present a

*The authors contribute equally.
†Corresponding author.

[1]The eyeball pose has also been denoted as the eyeball movement (Zhu and Deng 2017).

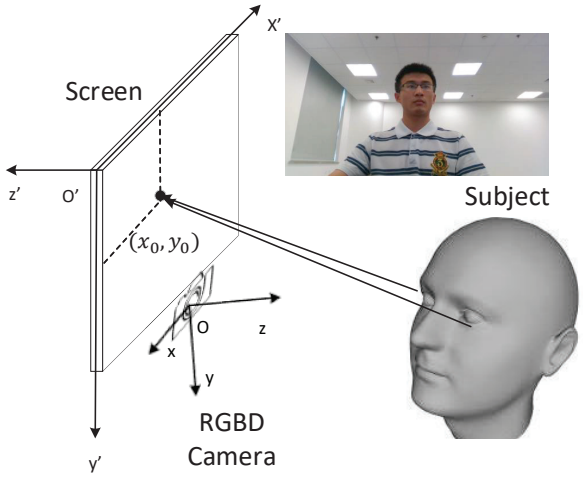

Figure 1: Our data acquisition system.

much larger RGBD gaze tracking dataset. Our dataset consists of 218 participants with a total over 165K images, probably the largest RGBD gaze dataset readily available to the research community.

On the algorithm front, we observe that (Zhu and Deng 2017) has demonstrated the effectiveness of estimating eyeball pose and head pose using separate CNNs. (Krafka et al. 2016) has further shown the benefits of separately extracting features from eye and face images. When combined with 3D eye positions features extracted using face grid, (Krafka et al. 2016) achieves the state-of-the-art performance. In a similar vein, we propose to decompose gaze point estimation into four separate modules: eyeball pose estimation, head pose estimation, 3D eye positions estimation, and gaze point estimation. Both head pose and 3D eye positions estimation exploit depth information. We notice that the captured raw depth images are of high noise and contain holes caused by occlusions, specularity of eyeglasses, depth range limitations, *etc*. We hence build a network to simultaneously remove noises and compensate for holes.

The contributions of this paper can be summarized as follows: 1) We build a large-scale RGBD gaze tracking dataset, to facilitate the exploration of data-driven approaches for gaze tracking; 2) We propose to decompose gaze point estimation into eyeball pose, head pose, and 3D eye position estimation and design a CNN based multi-task learning network to simultaneously refine depth maps and predict gaze point. Specifically, we present a generator network for both depth image refinement and head pose feature extraction. The generator is partially shared across multiple modules and can extract the effective head pose information; and 3) We conduct extensive experiments to show that our new technique outperforms existing state-of-the-art methods by a large margin in gaze tracking.

Related Work

RGB Image Based Gaze Tracking

Generally, gaze estimation can be categorized into model-based and appearance-based methods (Hansen and Ji 2010). Model-based methods (Zhu, Ji, and Bennett 2006) utilize geometric eyeball models and features for gaze estimation. Appearance-based methods (Lu et al. 2014b; 2014a) directly extract eye or face image information as feature vectors and learn a mapping from the feature vectors to gaze points. Previous research has mainly relied on hand-crafted features. Due to their success, CNNs have also been introduced to gaze tracking. In (Zhang et al. 2015), head poses were encoded as extra information used for gaze tracking. Krafka *et al.* (Krafka et al. 2016) proposed the implicit extraction of eyeball pose, head pose and eye coordinates from eye images, face images and a face grid. Zhang *et al.* (Zhang et al. 2017) proposed a spatial weights CNN for gaze point estimation directly from single RGB face images. It is worth noting that even though both head pose and eyeball pose can be inferred from face images, the input resolution of these faces is fixed because of computational costs, therefore eye areas are small, in addition to which CNN pooling operations cause information loss. Thus, such global face input solutions may not be a good choice for gaze point estimation. Recently, Zhu *et al.* (Zhu and Deng 2017) explained the within-subject and cross-subject ambiguity of extracting head pose based on facial landmarks, and proposed the encoding of head pose and eyeball pose with two separate CNNs. Ranjan *et al.* (Ranjan, De Mello, and Kautz 2018) designed a branched gaze network with different head poses.

RGBD Image Based Gaze Tracking

To recover a 3D model of head pose more accurately, RGBD cameras have been used in some works (Mora and Odobez 2013; Xiong et al. 2014). Previous works (Xiong et al. 2014) used eye or facial landmark locations in 3D space as seen by RGBD cameras, and predicted the gaze point through a mapping function learned from a personal calibration step. Although depth cameras were utilized to collect data and head poses were obtained (Sugano, Matsushita, and Sato 2014; Zhang et al. 2015), only six points on the face were used to establish the head coordinate system, insufficient to recover a 3D model of the face. Our method strengthens the importance of the facial 3D model and instead inputs the overall head depth into the network architecture to enhance performance.

Gaze Tracking Dataset

Because the deep learning method for gaze tracking is a data-driven appearance-based model, a large amount of publicly available gaze datasets have been proposed (Mora, Monay, and Odobez 2014; Sugano, Matsushita, and Sato 2014; He et al. 2015; Zhang et al. 2015; Huang, Veeraraghavan, and Sabharwal 2015; Krafka et al. 2016). Most of these datasets, however, only used single faces or eye images, or a combination thereof, which ignored the distance between participants and the screen target. This lack of depth information causes difficulties in estimating head poses and

Table 1: Statistics of our dataset with some publicly available datasets. Abbreviations: *cont.* for continuous, *illum.* for illumination.

Dataset	#Participants	#Poses	#Targets	Illum.	#Images	#Views	Modality
UT-Multiview	50	8 + synth.	160	1	64,000	8	RGB
OMEG	50	3 + cont.	10	1	45,000	1	RGB
MPIIGaze	15	cont.	cont.	cont.	213,659	1	RGB
TabletGaze	51	cont.	35	cont.	videos	1	RGB
iTracker	1474	cont.	cont.	cont.	2,445,504	1	RGB
Free-head	200	cont.	cont.	cont.	240,000	12	RGB
ShanghaiTechGaze	137	cont.	cont.	cont.	233,796	3	RGB
EYEDIAP	16	cont.	cont.	2	videos	1	RGBD
ShanghaiTechGaze+ (ours)	218	cont.	cont.	cont.	165,231 pairs	1	RGBD

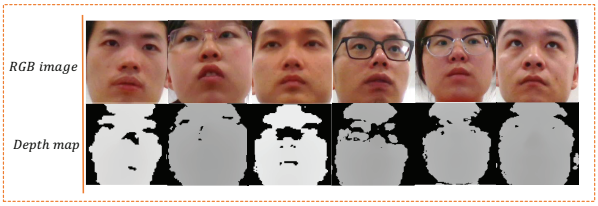

Figure 2: Some images captured by our system.

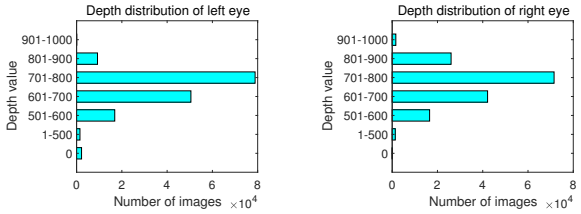

Figure 3: Depth distribution of left and right eyes in our dataset.

eye coordinates. In (Sugano, Matsushita, and Sato 2014; Lian et al. 2018), multi-view cameras were proposed to capture eye images, and the UT Multiview dataset and ShanghaiTechGaze dataset was built. Depth information was implied by the multi-view data, but this information was not obvious. Mora *et al.* (Mora, Monay, and Odobez 2014) have built a dataset based on RGBD by using a Kinect camera, but the participants were too few (only 16 participants). Krafka *et al.* (Krafka et al. 2016) showed that more participants would boost the performance of person-independent gaze tracking. Thus we build a large RGBD gaze dataset with 218 participants and 165,231 images. Our dataset will facilitate the study of data-driven-based approaches for gaze tracking, and we will release our dataset to the community in the future.

Our Proposed RGBD Dataset

Data Collection System

Our data collection system is depicted in Figure 1. Specifically, we use a 27-inch Apple iMac machine as a display. We use an Intel RealSense SR300 as our RGBD camera, which is placed under the iMac. We use the RGB images and their associated depth images captured by the RealSense SR300 to build our dataset. We fix the system on a desk in a room with normal lighting conditions. A chair is placed in front of the system, and all participants can move this chair to any comfortable position.

Data Acquisition Procedure

In the data acquisition phase, the participant is required to click a white dot displayed randomly on the screen. This clicking action makes the participant concentrate on the dot. After clicking, a blue dot will be generated in the position of the click. The distance between the white dot and the blue dot helps us to judge the reliability of the corresponding sample, because it is possible that the participant was distracted while clicking, causing an inaccurately placed blue dot. Finally, we drop incorrect dots exceeding a certain threshold. After which we then use the coordinates of the white dots as the ground-truth. When clicked, RGB and depth images of the participant are recorded together through the Intel RealSense SR300; the depth image is then resampled to be the same size as the RGB image (1920 $\times$ 1080 pixels) through a built-in program. The procedure is divided into 16 sessions and each session contains 50 dots.

There are 218 participants in our experiments (141 males, 77 females, aged between 19 and 37 years old). All participants have normal or corrected-to-normal vision. There are about 600-800 RGB and depth image pairs for each participant and their associated gaze point coordinates on the screen. In total, our dataset consists of 165,231 RGB/depth image pairs. We further use the images corresponding to 159 participants (119,318 RGB/depth image pairs) as training data and use the data corresponding to the remaining 59 participants as test data (45,913 RGB/depth image pairs). We list the statistics about the RGBD gaze tracking dataset and other gaze tracking datasets in Table 1.

Dataset Analysis

Since the participants can move their heads freely, our dataset contains faces at various depths. It is also worth noting that the depth images also have noise and black hole effects due to occlusion, illumination, the specularity of glasses, and out-of-valid range issues, as shown in Figure 2. The distribution of eyes at different distances is shown in

Figure 3 (The statistics are calculated by averaging the depth values around eye areas, excluding invalid values). The variable distances increase the complexity of the gaze point regression model, which makes gaze tracking in our dataset more challenging.

Method

As mentioned previously, gaze point estimation can be decomposed into eyeball pose, head pose, and 3D eye position estimation. Depth images provide head pose and 3D eye position information, which facilitates gaze tracking. The original depth images, however, sometimes contain considerable noise and black holes, limiting gaze tracking performance. In this section, we propose a CNN-based multi-task learning framework to simultaneously refine depth images and predict gaze points.

Figure 4 shows the overall network architecture. Our network predicts gaze points with the following steps: 1) We first extract eyeball pose features from two single-eye images. 2) We extract head pose features from RGB and depth images. In order to obtain more accurate head poses, we refine the depth map with a GAN. Here we utilize a generator network architecture which uses both RGB and depth images for depth image refinement, where the feature maps in the generator are also used to generate head pose features[2]. 3) We use the depth values of eye regions in the refined depth maps and eye coordinates in the original image to encode the 3D eye position. 4) We concatenate eyeball pose features, head pose features, as well as the 3D eye position features together and feed them into a network containing some fully connected layers for gaze point estimation.

Depth Image Refinement Based on GAN

Previous research has shown the practicability of single RGB image-based depth image generation (Eigen, Puhrsch, and Fergus 2014), which suggests that RGB images contain some depth information. In our setting, we have observed depth images with noise, as well as their corresponding RGB images. Since we do not have an accurate ground-truth depth for RGB images, we cannot directly learn a mapping with CNNs, as has been done in (Eigen, Puhrsch, and Fergus 2014). Thus we propose to use RGB and depth images together to synthesize a better depth map within a GAN (Goodfellow et al. 2014) framework.

Generator. In the generator, we feed RGB images and depth images into two separate CNNs to extract features, then we combine both features and use a decoder to generate a refined depth image. The detailed network architecture of the generator is shown in Figure 5. Since the depth map contains noise and black holes, we restrict the intensity of the synthesized depth to be consistent with its ground-truth for the non-black-hole areas. Because the measurement error of depth camera is different among all depth maps, we hope that CNNs would automatically fill the black holes. The RGB image contains content information and the depth image contains structure information. It is reasonable to infer partial structure from the content of RGB face. The CNN architecture itself seems to be a strong prior to regress more realistic images from non-realistic ones with reasonable supervision, which is similar to (Ulyanov, Vedaldi, and Lempitsky 2017). In addition, the middle feature will also be considered as the head pose information to regress the gaze point because the synthesized depth map information is from it totally. We denote $\Omega = \{(x, y)|I_i^d(x, y) \neq 0\}$, and I_i^d, I_i^{RGB} as the i^{th} RGBD image pair ($i = 1, \ldots, M$), where M is the total number of RGB/depth image pairs. Then the adversarial loss is given by

$$\ell_g = E[\log(D(G(I^d, I^{RGB})))] \sim G \tag{1}$$

Here G is the generator and D is the discriminator. Except for adversarial loss, we also use an additional L1 loss as the reconstruction loss to guide the training process, given by

$$\ell_{l1} = \frac{1}{M} \sum_i \|G(I_i^d(\Omega), I^{RGB}) - I_i^d(\Omega)\|_1 \tag{2}$$

Discriminator. The discriminator module in Figure 4 consists of simple convolution layers. It takes the synthesized depth maps and real depth images as its input to make them indistinguishable. The loss function of the discriminator D is

$$\ell_d = E[\log(D(I^d))] + E[\log(1 - D(G(I^d, I^{RGB})))] \sim D \tag{3}$$

Such synthesized depth maps help to obtain more accurate depth values from eye regions, which promotes the performance of gaze point estimation. In addition, when the synthesized depth maps are of a high-quality, the head pose features can help to predict gaze points more accurately.

Gaze Point Estimation Network

When a person stares at a point on the target screen, the position of the gaze point is geometrically determined by three factors: eyeball pose, head pose, as well as 3D eye positions, as shown in Figure 1. Instead of directly regressing gaze points from RGBD face images, following the work of (Krafka et al. 2016; Zhang et al. 2017), we propose that these factors be predicted separately, after which they can be combined together for gaze point estimation.

Eyeball pose estimation. Although a single eye contains eyeball pose information, two single-eye images can enhance estimation performance (Krafka et al. 2016) because two eyes are expected to stare at the same gaze point on the screen target. In this paper, we employ the shared ResNet-34 (He et al. 2016) as the eyeball pose extractor and take two single-eye images as inputs of it.

Head pose estimation. RGB images contain head pose information, while depth maps also contain 3D geometric information that is useful for pose estimation. Hence, we utilize both RGB images and depth maps of full faces to estimate head poses. Since the features at the bottom of the gen-

[2]Since the generator takes face and depth images as inputs, which contain eyeballs and the depths of eyeballs/faces, the features used for head poses contain some eyeball pose information and eye coordinate information. To maintain consistency with (Zhu and Deng 2017), we also refer to this feature as the head pose.

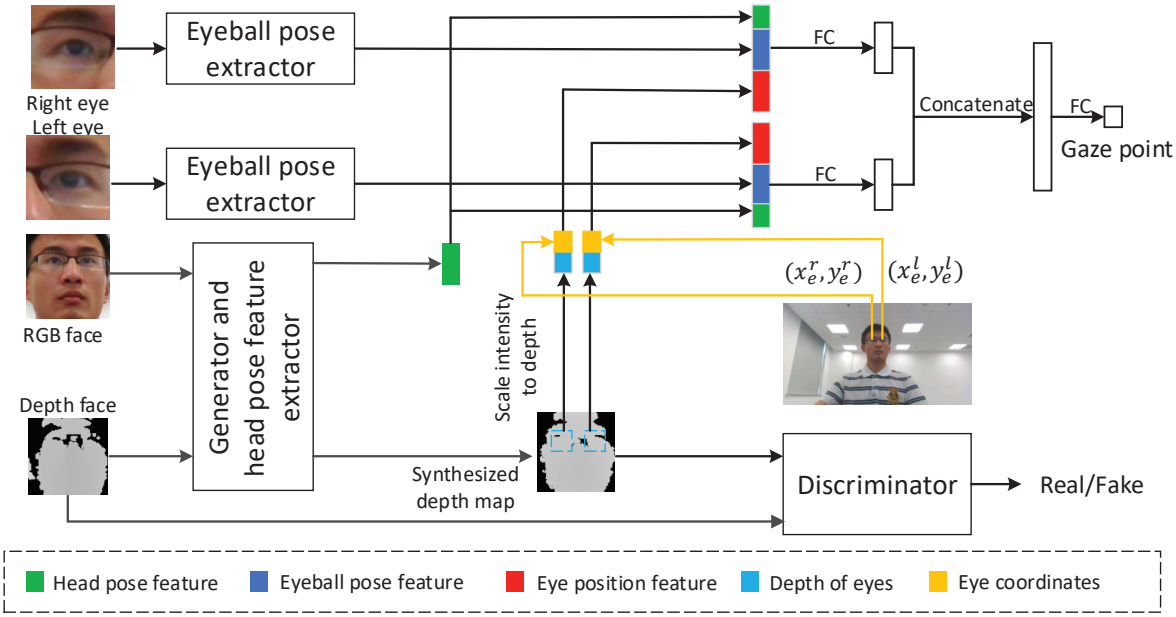

Figure 4: The network architecture for gaze point prediction. Eyeball pose features are extracted from two single-eye images. Head pose features are obtained from RGB and depth images. 3D eye positions are determined by eye coordinates and depth of eyes. Finally, all features are combined to predict gaze point (Best viewed in color).

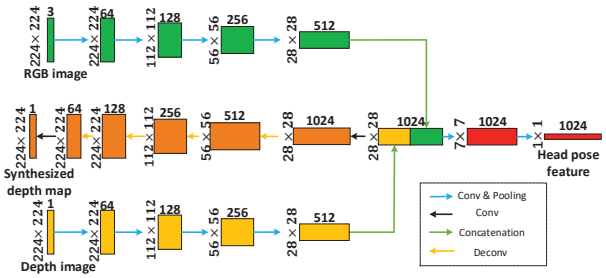

Figure 5: The network architecture of generator and head pose feature extractor.

erator already contain the content information and the structure information (from RGB and depth image), we reuse them to generate head pose features with an additional two convolutional layers, thereby helping to reduce the dimensionality of head pose features.

3D eye position extraction. In order to get 3D eye positions, we first use facial landmark detection methods to locate eye positions in RGB face images. We denote the coordinates of the left and right eye centers as (x_e^l, y_r^l) and (x_e^r, y_e^r), respectively, and denote the depth of these eyes as z_e^l and z_e^r, which can be obtained from the synthesized depth image. We can directly feed these coordinates to a neural network for gaze point estimation. It is based on the fact that the 3D eye position in the camera-centered world coordinate system can be derived based on the eye coordinates of images captured by the camera as well as their distances to the camera since the data acquisition system is fixed.

Gaze point estimation. Given eyeball pose features, head pose features, and 3D eye position features, we concatenate all of this information and feed it to a neural network with two fully connected layers to conduct gaze point estimation. The loss function is as follows:

$$\ell_{gp} = \frac{1}{M}\sum_{i}^{M} \|\hat{\boldsymbol{p}}^{\boldsymbol{i}} - \boldsymbol{p}^{\boldsymbol{i}}\|_2^2 \tag{4}$$

where $\hat{p}_i$ denotes the predicted gaze point, and its ground-truth is p^i for the i^{th} training image pair ($i = 1, \ldots, M$).

Remarks. Since the generator is shared by both the gaze point prediction network and the depth image refinement network, the optimization of gaze point prediction enforces the depth of refined images and pose features to be correct, thus facilitating GAN training; in addition, the GAN discriminator also helps to generate better depth images for more accurate gaze tracking.

Implementation Details

For data preparation, we first use the *Dlib* library to detect faces and landmarks on our dataset. Then a square patch with a length of 1.5 times the distance between eye corners is cropped out. Face regions and eye regions are cropped from the original RGB images and depth images. All cropped images are resized to be 224×224 and are then fed into our proposed CNN architecture. We first separately pretrain the GAN and gaze estimation networks, and then we finetune the overall network to get optimal estimation performance with multi-task learning.

Experiments

Experimental Setup

Training setup. We implement our method with the PyTorch (Paszke et al. 2017) framework. The batch size for all of our experiments is 100 for training and 200 for testing. We use 8 NVIDIA Tesla k40m GPUs to train our network. Stochastic Gradient Descent (SGD) optimization algorithm is adopted to train our network.

Datasets. We evaluate our method on both our RGBD gaze dataset and EYEDIAP. Our RGBD gaze dataset is used for gaze point prediction, and we choose images corresponding 159 subjects from the total of 218 subjects as a training set, with the remaining used as a test set. The EYEDIAP dataset is used for gaze direction prediction. Since depth images also facilitate estimation of gaze direction, we also use them for performance evaluation. We follow the same strategy as (Zhang et al. 2017) to choose frame images and gaze points. After that, we divide the 14 participants into 5 groups and perform cross-validation.

Evaluation metrics. For gaze point estimation on our RGBD gaze dataset, we use the Euclidean distance metric to measure the distance between the prediction of our gaze point and its ground-truth.

$$d_e = \frac{1}{M}\sum_{i}^{M} \|\boldsymbol{p^i} - \hat{\boldsymbol{p}}^{\boldsymbol{i}}\|_2 \tag{5}$$

where M is the total number of images in our dataset, $\boldsymbol{p^i}$ is the ground-truth for the i^{th} image, and its prediction is $\hat{\boldsymbol{p}}^{\boldsymbol{i}}$.

For gaze direction estimation on the EYEDIAP dataset, the angle deviation between our estimation and its ground-truth is used for performance measurement, i.e.,

$$a_e = \frac{1}{N}\sum_{i}^{N} \arccos \frac{\langle \boldsymbol{a^i}, \hat{\boldsymbol{a}}^{\boldsymbol{i}} \rangle}{|\boldsymbol{a^i}||\hat{\boldsymbol{a}}^{\boldsymbol{i}}|} \tag{6}$$

where N is the total number of images in EYEDIAP. $\boldsymbol{a^i}$ is the ground-truth of the i^{th} image, and its prediction is $\hat{\boldsymbol{a}}^{\boldsymbol{i}}$. $\langle \boldsymbol{a^i}, \hat{\boldsymbol{a}}^{\boldsymbol{i}} \rangle$ refers to the inner product between $\boldsymbol{a^i}$ and $\hat{\boldsymbol{a}}^{\boldsymbol{i}}$. To be consistent with (Zhang et al. 2017), Equation (5) is in millimeters (mm), and Equation (6) is in degrees.

Table 2: Performance comparison of gaze point estimation on our dataset. (unit: mm)

Methods	d_e
Multimodal CNN (Zhang et al. 2015)	67.2
iTracker (Krafka et al. 2016)	55.5
iTracker* (Krafka et al. 2016)	47.5
Spatial weights CNN (Zhang et al. 2017)	60.6
Our method	**38.7**

Table 3: Performance comparison of gaze direction estimation on EYEDIAP. (unit: degree)

Methods	a_e
Multimodal CNN (Zhang et al. 2015)	10.2 (2.9)
iTracker (Krafka et al. 2016)	8.3 (1.7)
iTracker* (Krafka et al. 2016)	5.7 (1.1)
Spatial weights CNN (Zhang et al. 2017)	6.0 (1.2)
Ghiass *et al.* (Ghiass and Arandjelovic 2016)	7.2 (1.3)
Our method	**4.8 (0.7)**

Table 4: Network architecture evaluation on our dataset and EYEDIAP.

Baselines	d_e	a_e
No head pose	54.0	7.8
No depth	46.7	5.7
No RGB	52.9	7.1
No decoder	44.2	5.3
Stacked RGB + depth	41.6	5.0
Our method	**38.7**	**4.8**

Performance Comparison

We compare our proposed method with state-of-the-art deep learning based methods for gaze point estimation on our dataset and gaze direction estimation on EYEDIAP, including:

- Multimodal CNN (Zhang et al. 2015): Normalized eye images and 3D head poses are fed into a CNN consisting of a LeNet feature extractor and a few fully connected layers to predict gaze direction.
- iTracker (Krafka et al. 2016): Eye images, faces, and face grids are fed into a multi-region CNN architecture. In the fully connected layer, all features are combined to predict gaze points.
- iTracker* (Krafka et al. 2016): We substitute the original feature extractor in iTracker with ResNet-34. All other parts are the same as with iTracker (Krafka et al. 2016).
- Spatial weights CNN (Zhang et al. 2017): It makes use of the full-face image as its input to learn the spatial weights CNN for gaze point and gaze direction prediction.

The experimental results of different methods are listed in Table 2 and Table 3, which show that our method outperforms all other methods for both gaze point and gaze direction estimation by a large margin. Specifically, we obtain the following findings from these results: i) Results of iTracker and iTracker* suggest that the architecture of the basic eyeball pose feature extractor is important for improving gaze tracking accuracy. ii) For a fair comparison, we improve the backbone of iTracker by using the same network as ours (ResNet-34) and denote this baseline as iTracker*. Our method still achieves the best performance, which is because: firstly, our decomposition strategy for gaze point estimation problem is effective and we introduce depth information into the network, which provides both head poses and 3D eye positions. Secondly, we utilize a generator and

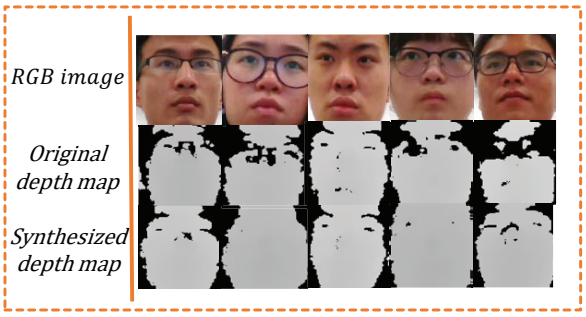

Figure 6: Visualization of original and synthesized face depth map.

apply GAN to refine the depth maps and predict gaze points. Although we do not geometrically model 3D faces, the depth data provide sufficient information, necessary head poses and 3D eye positions for improving gaze estimation.

In Table 3, we apply our methods to gaze direction estimation. The results on the EYEDIAP dataset are the average results based on 5-fold cross-validation, and the numbers in parentheses indicate the standard deviations of angles. Different from gaze point estimation, gaze direction estimation does not depend on the 3D eye positions, but the depth maps still provide the head pose information and help head pose estimation. Thus, with synthesized depths, our method still achieves state-of-the-art results for gaze direction estimation on EYEDIAP. The t-test [3] shows that the improvements are statistically significant.

Ablation Studies

In order to explore the effectiveness of different modules for gaze tracking, we report the performance after removing different components from our network. We conduct these experiments on our dataset and the EYEDIAP dataset. The results of all baseline methods are listed in Table 4. For gaze point estimation, our network combines head pose features, eyeball pose features and 3D eye position features. For gaze direction estimation, we only concatenate the head pose features and eyeball pose features.

First, we remove the head pose information, which means that the network only takes two single-eye images and 3D eye positions as inputs, which leads to the worst recorded performance. As mentioned before, head poses are necessary for gaze estimation, otherwise, the network will be over-fitting. Both RGB image and depth map provide head pose information. Next, we remove the RGB image and depth map, respectively. Removing depth input results in better performance over removing RGB input, which shows that RGB face data makes for better representation than facial depth data. It is also worth noting that the difference between removing depth and iTracker* (Krafka et al. 2016) lies in the 3D eye position encoding method. To obtain 3D eye position information, in (Krafka et al. 2016), a binary map (referred to as a face grid) is used to indicate face regions in the images, and the sizes of the faces and their positions in the images captured by the camera roughly encodes the 3D eye coordinates. Our accurate 3D eye positions slightly enhance the performance (46.7 mm *vs.* 47.5 mm), as shown in Table 4.

In addition, we also remove the decoder module during depth reconstruction, obtaining a worse performance than the depth reconstruction, which validates the effectiveness of depth refinement in our network. The network can learn a better representation of head pose feature due to the supervision of depth reconstruction. Finally, since we introduce the depth map to extract head pose feature, we also conduct the experiment about how to combine RGB image and depth map. We stack both together to extract the head pose feature and denoted it as Stacked RGB + depth, but this leads to a poorer performance than our method.

Evaluating the Quality of Synthesized Depth Maps

Due to the raw depth images directly obtained from depth sensor contain noises, it is hard to evaluate how accurate the generated depth map is. Alternatively, we evaluate the quality of synthesized depth maps through the performance of gaze point estimation indirectly. In this experiment, we use the same network architecture. In the testing phase, however, rather than using depth data from the refined depth map, we use the depth value of the original depth image as 3D eye positions for gaze tracking. The gaze point estimation error of this baseline on our dataset is 40.4 mm, which is worse than that when using the refined depth map (38.7 mm). The improvement of our method over this baseline quantitatively validates the better quality of synthesized depth maps over original maps. We also qualitatively show the improvement from original depth images to refined ones in Figure 6. Our synthesized depth map can complement the black holes caused by occlusion, the specularity of eyeglasses, and out-of-valid ranges of the depth camera partly.

Conclusion

In this paper, we decompose gaze point estimation into eyeball pose, head pose, and 3D eye position estimation. Depth is important to predict gaze because it complements head pose and 3D eye position information. Raw depth images directly obtained from depth sensors, however, contain noises, so we utilize a generator network and apply GAN to refine the depth maps and predict gaze points simultaneously. The entire network is combined via a multi-task learning framework. In addition, we build a large-scale RGBD gaze dataset. As far as we know, this is the largest dataset in terms of its number of subjects. Extensive experiments on our dataset and the EYEDIAP dataset show that our gaze tracking method outperforms all existing state-of-the-art methods by a large margin.

Acknowledgment

This project is supported by NSFC (No. 61502304).

[3] https://www.graphpad.com/quickcalcs/ttest1/?Format=SD

References

Eigen, D.; Puhrsch, C.; and Fergus, R. 2014. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, 2366–2374.

Ghiass, R. S., and Arandjelovic, O. 2016. Highly accurate gaze estimation using a consumer rgb-d sensor. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 3368–3374. AAAI Press.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Hansen, D. W., and Ji, Q. 2010. 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.

He, Q.; Hong, X.; Chai, X.; Holappa, J.; Zhao, G.; Chen, X.; and Pietikäinen, M. 2015. Omeg: Oulu multi-pose eye gaze dataset. In *Scandinavian Conference on Image Analysis*, 418–427. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Huang, Q.; Veeraraghavan, A.; and Sabharwal, A. 2015. Tabletgaze: unconstrained appearance-based gaze estimation in mobile tablets. *arXiv preprint arXiv:1508.01244*.

Krafka, K.; Khosla, A.; Kellnhofer, P.; Kannan, H.; Bhandarkar, S.; Matusik, W.; and Torralba, A. 2016. Eye tracking for everyone. In *CVPR*, 2176–2184.

Lian, D.; Hu, L.; Luo, W.; Xu, Y.; Duan, L.; Yu, J.; and Gao, S. 2018. Multiview multitask gaze estimation with deep convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 1–14.

Lu, F.; Okabe, T.; Sugano, Y.; and Sato, Y. 2014a. Learning gaze biases with head motion for head pose-free gaze estimation. *Image and Vision Computing* 32(3):169–179.

Lu, F.; Sugano, Y.; Okabe, T.; and Sato, Y. 2014b. Adaptive linear regression for appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(10):2033–2046.

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

Mora, K. A. F., and Odobez, J.-M. 2013. Person independent 3d gaze estimation from remote rgb-d cameras. In *ICIP*, 2787–2791. IEEE.

Mora, K. A. F.; Monay, F.; and Odobez, J.-M. 2014. 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*, 255–258. ACM.

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

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

Ranjan, R.; De Mello, S.; and Kautz, J. 2018. Lightweight head pose invariant gaze tracking. *arXiv preprint arXiv:1804.08572*.

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

Sugano, Y.; Matsushita, Y.; and Sato, Y. 2014. Learning-by-synthesis for appearance-based 3d gaze estimation. In *CVPR*, 1821–1828.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. S. 2017. Deep image prior. *CoRR* abs/1711.10925.

Xiong, X.; Liu, Z.; Cai, Q.; and Zhang, Z. 2014. Eye gaze tracking using an rgbd camera: a comparison with a rgb solution. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 1113–1121. ACM.

Zhang, X.; Sugano, Y.; Fritz, M.; and Bulling, A. 2015. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4511–4520.

Zhang, X.; Sugano, Y.; Fritz, M.; and Bulling, A. 2017. It's written all over your face: Full-face appearance-based gaze estimation. In *CVPRW*.

Zhu, W., and Deng, H. 2017. Monocular free-head 3d gaze tracking with deep learning and geometry constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3143–3152.

Zhu, Z.; Ji, Q.; and Bennett, K. P. 2006. Nonlinear eye gaze mapping function estimation via support vector regression. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, 1132–1135. IEEE.