Improved Itracker Combined with Bidirectional Long Short-Term Memory for 3D Gaze Estimation using Appearance Cues

Xiaolong Zhou[a,b], Jianing Lin[a], Zhuo Zhang[a], Zhanpeng Shao[a], Shenyong Chen[a,c], Honghai Liu[d]

[a] *Zhejiang University of Technology, Hangzhou 210023, China*
[b] *Quzhou University, Quzhou 324000, China*
[c] *Tianjin University of Technology, Tianjin 300384, China*
[d] *University of Portsmouth, Portsmouth, UK*

Abstract

Gaze is an important non-verbal cue for speculating human's attention, which has been widely employed in many human-computer interaction-based applications. In this paper, we propose an improved Itracker to predict the subject's gaze for a single image frame, as well as employ a many-to-one bidirectional Long Short-Term Memory (bi-LSTM) to fit the temporal information between frames to estimate gaze for video sequence. For single image frame gaze estimation, we improve the conventional Itracker by removing the face-grid and reducing one network branch via concatenating the two-eye region images. Experimental results show that our improved Itracker obtains 11.6% significant improvement over the state-of-the-art methods on MPIIGaze dataset and has robust estimation accuracy for different image resolutions under the premise of greatly reducing network complexity. For video sequence gaze estimation, by employing the bi-LSTM to fit the temporal information between frames, experimental results on EyeDiap dataset further demonstrate 3% accuracy improvement.

Keywords: Gaze estimation, CNN, RNN, LSTM

*Corresponding author: Shengyong Chen, Email: sy@ieee.org

1. INTRODUCTION

Gaze estimation is to speculate the gaze direction or a gaze point for a particular plane. It has been viewed as an important clue to speculate on the target's attention, which has been widely applied in many human-computer interaction-based fields. Recently, many gaze estimation methods have been explored, however, existing gaze estimation systems [1, 2, 3, 4, 5] have the following defects: redundant calibration process, complex system settings, limitations of lighting conditions and the non-universal calibration for different subjects as well as low tolerance to head movement, which limit the application of gaze estimation.

Gaze estimation methods can roughly be classified into two categories: model-based methods and appearance-based methods. The model-based methods simulate the eye-gaze through a three-dimensional model and estimate the gaze direction by applying the calibrated eye parameters to the gaze model [1, 5, 6, 7, 8, 9]. The appearance-based methods try to map features extracted from face or eye images to gaze direction or gazing point, which can be further classified into manual-feature-driven methods and data-driven methods.

Traditional manual-feature-driven gaze estimation methods [2] normally achieve the position of gaze point by mapping the eigenvector formed by the local features such as corneal reflections or eye corners as well as iris contours to the final target. Cai et al. [10] replaced the infrared reflection point with the eye corner, hence reduced the complexity of the device. At the same time, they improved the iris center localization method and simplified the differential-operator. The eigenvector composed of the eye corner and iris center was used for the establishment of subsequent regression equation. Feng et al. [3] preprocessed the pixel values of the whole eye region to form the feature, and applied the eigenvectors generated in the calibration process to linearly fit the eigenvectors in real-time prediction. The fitted parameters were then combined with the corresponding calibration points to get the predicted result. Kacete et al. [11] used random forest regression to perform gaze estimation and combined with depth information to improve the result under large head pose. Wang et al. [12]

proposed a k-Nearest Neighbor (KNN) method based on the head pose and iris center position. They used head pose and iris center position as the criterion of classification and trained the class-independent regression model to fit the mapping relationship on the corresponding data. Although some traditional appearance-based methods can achieve high accuracy, they tend to be poorly tolerant to various head poses, illumination changes and need person-specified calibration.

Recently, a number of datasets have been proposed to provide a unified standard for gaze estimation evaluation as well as a reliable data source for data-driven methods. These methods such as Convolutional Neural Network (CNN) based methods have great potential to handle with many traditional challenges, since they use a data-driven off-line training instead of cumbersome personal calibration and only use a web-cam to avoid tedious system setup. Zhang et al. [13] used CNN to map the eye images and head poses to gaze vector, which showed that the CNN-based method has higher accuracy than classic methods for various illumination and appearance differences. Afterwards, they improved the basic network in [13] from Alexnet to vgg16 [14], and put the head pose information into the penultimate layer. It got a better estimation result but increased the scale of the network. Ranjan et al. [15] proposed a network based on Alexnet, maintained the previous network layers and trained the last two layers separately based on head pose. The final results indicated that this network was more robust to various head pose without increasing the network scale and inference complexity. Deng et al. [16] analyzed the over-fitting problem between head pose and gaze vector and proposed a two-step training structure. They first trained the separate model for head pose and eyeball movement, then aggregated them to estimate gaze vector from coarse to fine. This method had less potential for head-correlation over-fitting, but lacked evaluation on public datasets. Krafka et al. [17] separately input the whole face image and eye images to the corresponding branch and used face grid branch to locate face position in order to supply the location information for predicting gaze point on the screen. However, this method was only limited to the 2D gaze estimation, the performance on 3D was not evaluated. Zhang et al. [18] demonstrated that the entire face encoding the information of head pose and illumination could be beneficial for the final result, so they directly input the normalized face to the Alex-based network and proposed a spatial weights CNN to reduce redundant information in face region. This method has demonstrated more robust result under significant variation in illumination conditions but appears to be more complexity, which is not friendly enough to different hardware.

In order to improve the 3D gaze estimation accuracy while keeping hardware friendly, we propose an improved Itracker as a static model to predict the final result for a single image frame. For video sequence, we further employ a many-to-one bidirectional Long Short-Term Memory (bi-LSTM) to fit the temporal information bewteen frames. The main contributions of this paper are listed as following.

1) We improve the Itracker model [17] to predict the gaze of a single image frame. We analyze the role of face grid module in 2D gaze estimation and remove this module for our 3D gaze estimation. We concatenate the left and right eye images into a unified input with 6 channels. We reduce the network optimization parameters by nearly half without sacrificing accuracy.

2) We introduce the bi-LSTM to simulate the temporal sequence information and propose a gaze estimation method combining static and temporal models. To the best of our knowledge, this is the first time that a bidirectional recurrent neural network has been employed into gaze estimation to simulate temporal information.

3) We perform various evaluation on our proposed gaze estimation method on two publically available datasets: MPIIGaze [13] and EyeDiap [19]. Without any pretraining and data augmentation, we obtain a significant improvement of 11.6% over the recent state-of-the-art methods on MPIIGaze under the premise of greatly reduction on network complexity. Moreover, experimental results show that our method has high accuracy to various resolution degradation.

2. METHODOLOGY

In this section, we first introduce the steps of data preprocessing and then describe the network architecture for the proposed method, and finally describe the proposed temporal module as well as the implementation details. The overall architecture is shown in Fig.1.

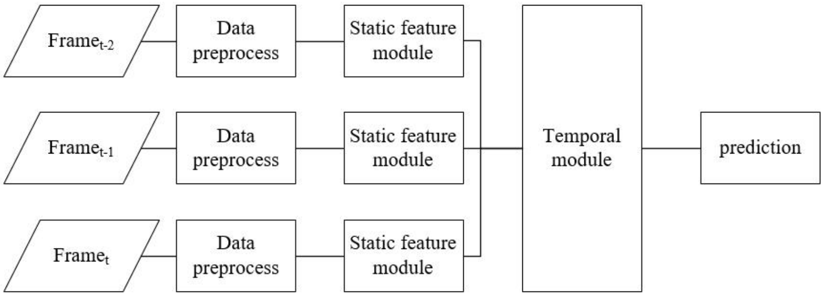


Figure 1: The overall architecture of proposed 3D gaze estimation method.

2.1. 3D Gaze Estimation

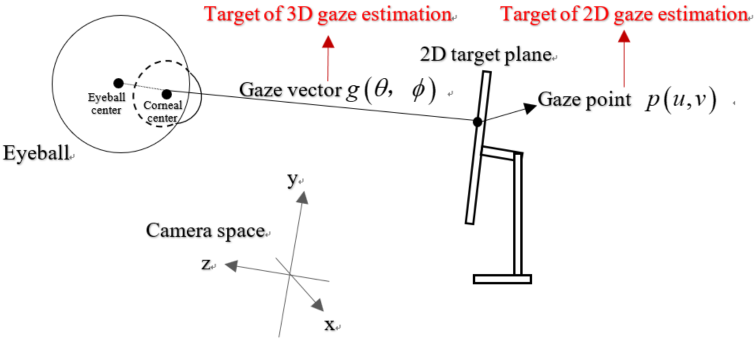

Target of 3D gaze estimation
Target of 2D gaze estimation
2D target plane
Eyeball center
Corneal center
Gaze vector $g(\theta, \phi)$
Gaze point $p(u,v)$
Eyeball
y
Camera space
z
x

Figure 2: The difference between 2D gaze estimation and 3D gaze estimation.

The purpose of 3D gaze estimation is to learn a function f to map the image I to a 3D gaze vector g, where $g = f(I)$. This vector is originated from corneal center or a reference point of the face. The 2D case can be obtained by intersecting the 3D gaze vector g with the specific 2D target plane. The difference between these two cases is shown in Fig.2. In this paper, we focus on estimating the 3D gaze vector g.

2.2. Data Preprocessing

Similar to [20], to weaken the effects of different head poses and various camera parameters on the final gaze estimation result, we make a certain perspective transformation on the original images so that we only need to train the model for gaze estimation under the specific virtual space. This process greatly reduces the complexity of the fitting problem as well as the potential model size.

The main process of data normalization can be divided into two steps. The first step is to rotate the camera by a conversion matrix so that the face reference point will be at the image center from a fixed distance, which could reduce the appearance variance. The second step is to transform the face into an image plane of a specific camera space through a transform matrix in order to reduce the negative effects of different camera configurations.

Assuming that a is the coordinate of the face reference point under camera space, we make the virtual camera face to the reference point by letting the z-axis of the virtual space be $v_z = \frac{a}{\|a\|_2}$, and then assuming that $H\left[h_x, h_y, h_z\right]$ is the rotation matrix of head pose, where h_x, h_y, h_z denote the coordinate of head in camera space. In order to make the x-axis parallel to the horizontal direction of head, we make $v_x = v_y \times v_z$, where $v_y = v_z \times h_x$. The rotation matrix R between original camera space and virtual camera space is computed as $R = \left[r_x, r_y, r_z\right]^T$. We assume that the distance between the virtual camera and the reference point is d. The conversion matrix $M = SR$ is used for the first step, where $S = diag\left(1, 1, \frac{d}{\|a\|_2}\right)$.

The second step is implemented by the warp matrix $W = C_o M C_n^{-1}$, where C_o is the intrinsic matrix of original camera and C_n is the intrinsic matrix of virtual camera that is determined by the size of output image.

Similar to the transformation of the image, we also need to convert the corresponding gaze label during training procedure. Let $g_n = Rg_o$, where g_n and g_o denote the normalized gaze label and the original gaze label, respectively. Then, we represent it by Euler angle to release the constraint relationship of unit vector. In test phase, for each prediction result, we need to convert them from virtual space back to the original camera space, so the result is obtained from $g_o = R^{-1}g_n$.

2.3. Network Architecture

In this paper, we propose a network architecture combined with bidirectional LSTM to incorporate temporal information for 3D gaze estimation. In [17], the authors input the face and the left and right eyes separately to a single branch of the network, and then mapped the merged features extracted from each branch to obtain the ultimate two-dimensional gaze point on the screen. We notice that a face grid module has been added to the network structure in [17], which had a greater impact on the final estimation results. Since the method in [17] needs to obtain the exact gaze point on the device, in addition to predicting the gaze direction, it is necessary to know exactly the head position in camera space. This information is primarily provided by the face grid module. Since we mainly discuss how to efficiently and accurately get the 3D gaze direction in this paper, the face grid module can be ignored in our topology. In the datasets, since each subject's gaze direction has a constraint that the gaze point is on the screen, which causes gaze direction be related to the relative position of the face and the camera. However, in the real-world application scenario, there is no such constraint, so it is easy to conclude that the gaze direction and the head position in camera space are independent. Therefore, we did not conduct experiments with or without face-grid, since even if relevant evaluations were performed on the datasets, the experimental results were meaningless to the final conclusions.

In order to reduce the network size as much as possible while ensuring accuracy, we concatenate the left and right eye images to form a single 6 channels input. In this way, we can successfully reduce nearly half the number of the network parameters (from approximately 3.6 million to 1.8 million) without any reduction in the final estimation performance.

In [18], the entire face already contains all the information needed for gaze estimation. It can also be seen from [14] and [21] that the additional face landmarks or head pose information has little impact on the final result. Because of the rich combination of hyper parameters, we cannot conclude that this weak improvement is attributed to the introduction of these additional structured information. So, in this paper, for simplicity we don't include these extra shape cues or head poses information in our network. All non-linear relationships are directly encoded by the network. The final static feature module is depicted in Fig.3.

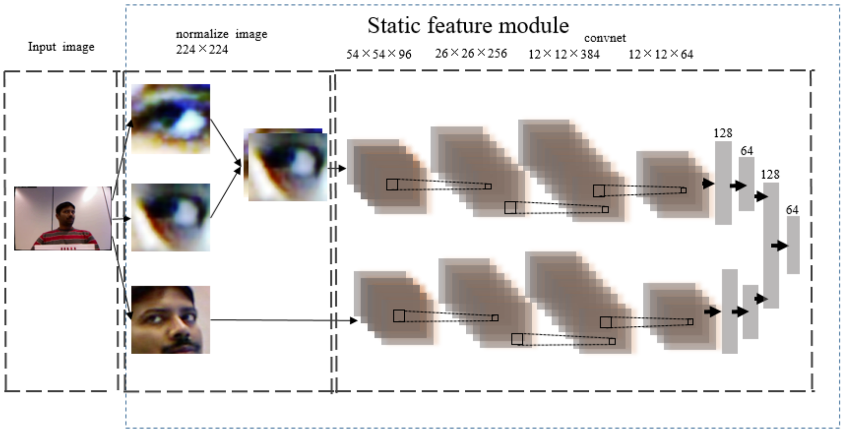


Figure 3: The static feature extraction module.

To the best of our knowledge, existing gaze estimation models rarely use the temporal information. In this paper, we consider the correlation of gaze direction between the consecutive frames. We use the bi-LSTM to simulate the temporal relationship to increase the accuracy and robustness of the network. The overall architecture is shown in Fig.1, which is divided into two modules: static module and temporal module. The static module learns features from the separate face and eyes appearance. It consists of a two-branch CNN and unified

FC layers. One branch CNN extracts the features from normalized face and the other from concatenated eyes image. The FC layers combine these two parts of features and learn a joint representation for the fused features. The learned features are then input to the many-to-one bi-LSTM. We finally use a linear regression to get the predicted result in normalized space from the hidden units in last time step.

2.4. Temporal Module Description

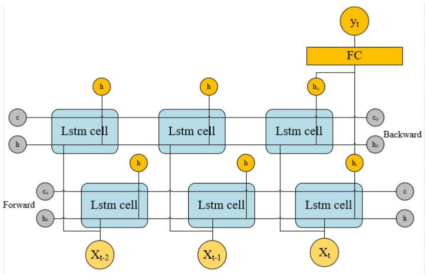


Figure 4: The temporal module.

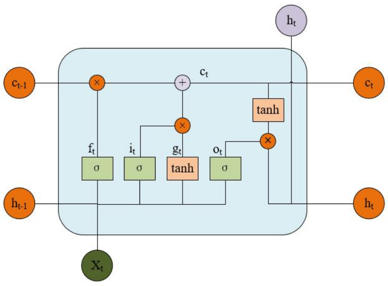


Figure 5: The structure of a single LSTM cell.

The overall structure of temporal module is shown in Fig.4. The LSTM structure contains a series of repeated LSTM cells, and the structure of a single LSTM cell is shown in Fig.5. Each LSTM cell contains three multiplicative units that represent the forget gate, the input gate, and the output gate. These multiplicative units allow LSTM memory cells to store and transfer information over a long period of time. The c and h respectively indicate the cell and hidden state. In Fig.4, (x_t, c_{t-1}, h_{t-1}) indicates the input layers and (h_t, c_t) indicates the output layers. Next, we briefly introduce how a standard LSTM cell generates outputs from inputs.

At time step t, the forget gate, the input gate and the output gate are represented as f_t, i_t, o_t respectively. The LSTM cell first uses the forget gate to filter out the information that needs to be discarded. The filtered information indicates some partial features extracted from the previous frame which is related to the former gaze direction but obviously irrelevant with the current target.

$$f_t = \sigma\left(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}\right) \tag{1}$$

where (W_{if}, b_{if}) and (W_{hf}, b_{hf}) respectively represent the weight matrix and bias term mapping input layer and hidden layer to the forget gate. The σ is the gate activation function, which is selected as the sigmoid function in this paper.

Then, the LSTM cell uses the input gate to incorporate valid information.

$$g_t = tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}\right) \tag{2}$$

$$i_t = \sigma\left(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}\right) \tag{3}$$

$$c_t = f_t c_{t-1} + i_t g_t \tag{4}$$

where (W_{ig}, b_{ig}) and (W_{hg}, b_{hg}) respectively represent the weight matrix and bias term mapping the input layer and hidden layer to the cell gate. (W_{ii}, b_{ii}) and (W_{hi}, b_{hi}) respectively represent the weight matrix and bias term mapping the input layer and hidden layer to the input gate.

Finally, the LSTM cell gets the output hidden layer from the output gate.

$$o_t = \sigma\left(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}\right) \tag{5}$$

$$h_t = o_t tanh\left(c_t\right) \tag{6}$$

where (W_{io}, b_{io}) and (W_{ho}, b_{ho}) respectively represent the weight matrix and bias term mapping the input layer and hidden layer to the output gate.

As shown in Fig.4, bi-LSTM contains a forward LSTM layer and a backward LSTM layer. In this paper, a sequence is composed of three image frames. The final gaze prediction is obtained by a fully connected layer. This layer maps the hidden layers got from forward and backward units of the last frame in time t to the final two-dimensional gaze vector g.

$$g = fc\left(h_t, h_{tr}\right) \tag{7}$$

2.5. Implementation Details

We use a reduced version of the convolution layers of Alexnet as the basic network for each branch. Each basic network has 4 convolution layers. The face branch is connected to a 128-dimensional FC layer followed by a 64-dimensional FC layer while the eyes branch is connected to a 64-dimensional FC layer. These two parts of the features are then combined through a 64-dimensional FC layer and regularized by a Dropout layer to prevent over-fitting problem. If it is a static model, the final prediction results could be obtained directly through a 2-dimensional FC layer. Else, the 64-dimensional features would be used as input to the temporal model. In this paper, we use bi-LSTM as the temporal model which has 1 LSTM layer and 32 hidden units.

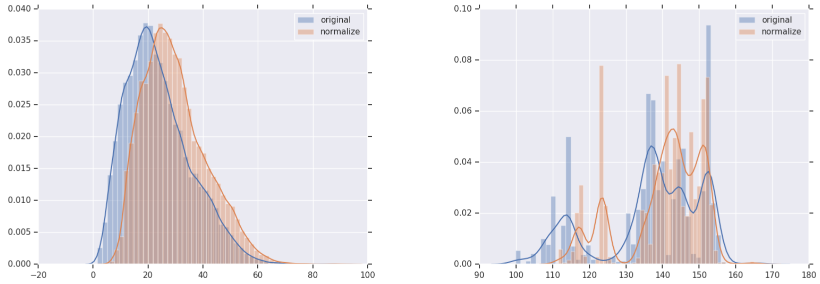
In the temporal model, we use a stage-wise training approach. We first train the static model from scratch and do not use any data augmentation processing to ensure the high reproducibility of the experimental results. We then treat the static model as a deep feature extractor whose parameters are frozen and no longer adjusted during the second training stage. We re-arrange the training data by a sliding window fashion. Every successive three frames form a sequence and the last frame of each sequence is treated as the ground truth.

We train the model using the Euclidean loss with the Adam optimizer. The basic learning rate is set to 0.0001 and the probability of dropout is set to 0.3. The batch size is 100 while the epoch is 20 for both static and temporal models.

3. EXPERIMENTS AND RESULTS

To validate the effectiveness of the proposed network for 3D gaze estimation, we evaluate the proposed method on two publicly available datasets: MPIIGaze [13] and EyeDiap [19]. First, we conduct cross person/group evaluation to show the basic performance of our method. Then, we conduct within person evaluation to demonstrate the potential accuracy of our method. Next, we perform ablation study to evaluate the role of each module in our network. Further, we conduct experiments with different resolutions to show the robust performance of our network to different resolution inputs. Finally, we combine the temporal model to explore the impact of temporal information on the estimation results.

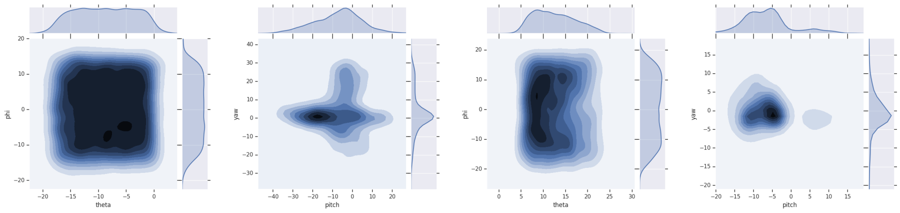
3.1. Datasets



(a) mean value(MPIIGaze)

(b) mean value(EyeDiap)

Figure 6: Distribution of images mean values on the MPIIGaze and filtered EyeDiap datasets.



(a) g(MPIIGaze)

(b) h(MPIIGaze)

(c) g(EyeDiap)

(d) h(EyeDiap)

Figure 7: Distribution of ground truth eye gaze g and head orientation h on the MPIIGaze and filtered EyeDiap datasets.

For the MPIIGaze dataset, we take the center of the six provided face landmarks as the start point of gaze vector as well as the facing point of the virtual camera. In data preprocessing step, to reduce the illumination variance, we apply adaptive histogram equalization on each input image. Fig.6a shows the mean

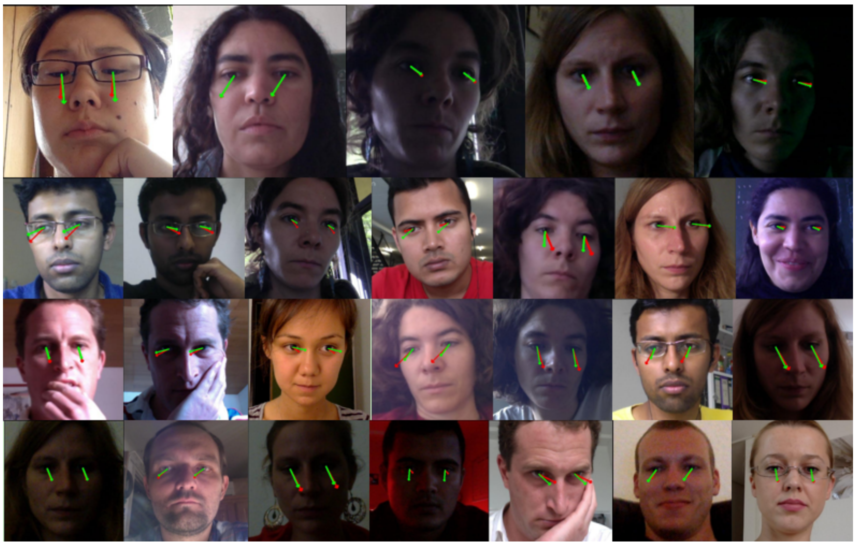


Figure 8: Some prediction results on the MPIIGaze dataset. Green and red lines indicate the predictions and the ground truth, respectively.

value changes before and after the equalization process. MPIIGaze dataset has a total of 15 participants. We perform leave-one-person-out cross validation on all participants to facilitate comparison with other methods. Fig.7a and 7b show the distribution of ground truth gaze angle and head poses in the MPIIGaze dataset.

For the EyeDiap dataset, we take the midpoint of the provided two iris centers as the origin of gaze vector as well as the facing point of the virtual camera. Similar with MPIIGaze, we apply adaptive histogram equalization to reduce illumination variance. Fig.6b shows the changes of pixel mean value after this operation. The gaze targets on this dataset fall into two categories: screen targets and floating targets. In order to facilitate comparison, we only use the screen targets for evaluation and sample one image per 15 frames from 4 VGA videos of each participant. We filter out frames that meet the following conditions: (1) The participant is not looking at the screen; (2) The annotation is not provided properly; (3) The gaze angle is violating the physical constraints

($|\theta| \leq 40^o, |\phi| \leq 30^o$). We divide 14 participants into 4 groups and perform leave-one-group-out cross validation on all groups. Fig.7c and 7d show the distribution of gaze angle and the distribution of head poses in the EyeDiap dataset.

3.2. Cross Person/ Group Evaluation and Within Person Evaluation

Table 1: COMPARISON RESULT WITH THE STATE-OF-THE-ART METHODS ON MPIIGAZE DATASET.

Methods	**3D degrees error**
Baltrusaitis T et al. 2016 [21]	9.96
Wood E et al.2016 [22]	9.58
Shrivastava A et al. 2016 [23]	7.8
Nie S et al. 2018 [24]	7.1
Zhang X et al. 2015 [13]	6
Krafka K et al. 2016 [17]	5.6
Zhang X et al. 2017 [14]	5.4
Zhang X et al. 2017 [18]	4.8
our static model	4.18

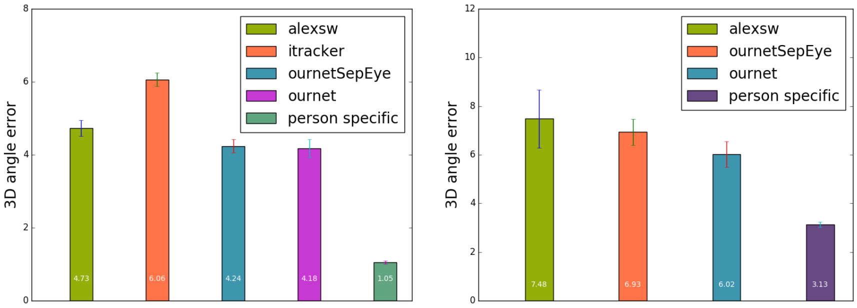
Table 2: COMPARISON RESULT WITH THE STATE-OF-THE-ART METHODS ON EYEDIAP DATASET.

Methods	**3D degrees error**
Krafka K et al. 2016 [17]	8.3
Park S et al.2018 [25]	7.4
Zhang X et al. 2017 [18]	6.0
Palmero C et al. temporal model 2018 [26]	3.4
Our static model	6.02
Our static + bi-LSTM model	5.84

In order to demonstrate the basic performance of our method, we perform a comparative experiment on the above-mentioned datasets. Table 1 and Table 2 show the comparison between our method and other state-of-the-art methods on the MPIIGaze and EyeDiap datasets, respectively. The 3D degrees error refers to the angular difference between ground truth and prediction. From Table 1, we can see that our method has achieved excellent result on the MPIIGaze evaluation. MPIIGaze dataset covers significant variation in illumination. Fig.8 shows some part of frames and prediction results in MPIIGaze dataset. It can be seen that our method can guarantee high accuracy against various illumination challenges. From Table 2, our method ranks the 2nd on the EyeDiap evaluation, but it still has a significant advantage in network complexity over the ranked 1^{st} method. Network parameters comparison: ranked 1^{st} (about 130 million) vs ournet (about 1.8 million). It indicates that our method needs to be improved to better balance the complexity and accuracy under large head pose environment.

We have selected two state-of-the-art face-based gaze estimation methods for further comparison: (1) AlexSW, a state-of-the-art full-face-based method proposed in [18];(2) Itracker, a fundamental method used in this paper proposed in [17]. For a fair comparison, the image normalization process is the same as used in this paper, and the final output is resized to a resolution of 224×224.

Fig.9a shows the comparison result between our method and other state-of-the-art methods on MPIIGaze dataset. Since it is a 3D gaze estimation, we use the angle error between the prediction value and ground truth to indicate the prediction accuracy. As can be seen from the figure, our method has a significant 11.6% improvement over the state-of-the-art methods on the MPIIGaze dataset. Meanwhile, the overall complexity of our network is much smaller than [18], which is the first bar on the chart showed in Fig.9a. The third bar shows the result of two separate eyes part network structure. It can be seen that the combination of the two eyes does not reduce any accuracy but even give a slight improvement. The last bar is the result of within-person evaluation. The evaluation of this part is mainly for the purpose of demonstrating the potential (upper bound) of our network. The results indicate that our network still has



(a) MPIIGaze

(b) EyeDiap

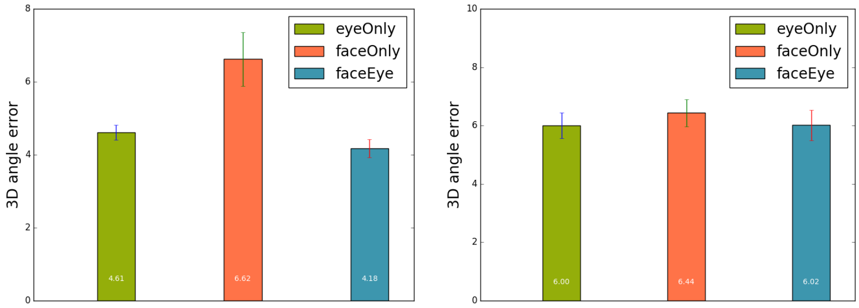
Figure 9: Cross person and person specific evaluation results on MPIIGaze and EyeDiap datasets.

space for improvement in the cross-person evaluation.

Fig.9b shows the comparison result on EyeDiap dataset. Similarly, our method has a small improvement in accuracy compared to the baselines. However, it should be noted that for the method in [18], the result given in our chart are not as good as that mentioned in the original paper. The 3D degree error on EyeDiap in [18] is 6.0, that is to say, our method has similar accuracy compared with [18]. But our method has great advantages in terms of network size. It can be seen from the second and third bars that even better results are obtained after the eye parts have been concatenated. The last bar demonstrates that our method performs poor even in within person evaluation on EyeDiap dataset compared to the result on MPIIGaze dataset, which indicates that our method degrades when encountering large head pose but has robust result responding to various illumination conditions.

3.3. Network Parts Evaluation

In this session, we split the network into two separate branches (eye module and face module) to verify the role of each network module. Fig.10a shows the results of our evaluation on MPIIGaze. It shows that the final prediction accuracy is mainly depending on the eyes branch network and the face part



(a) MPIIGaze

(b) EyeDiap

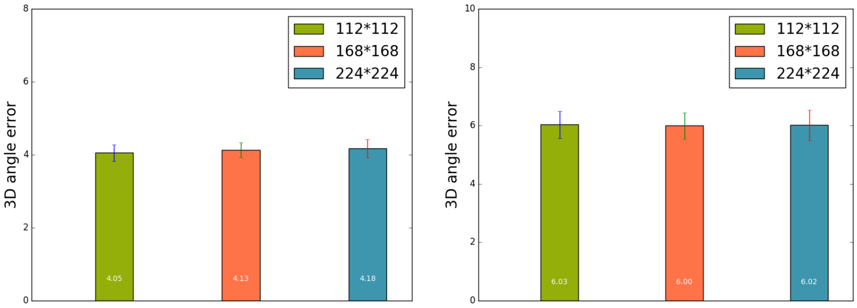
Figure 10: Ablation study on MPIIGaze and EyeDiap datasets.

also contributes a bit. Similarly, Fig.10b shows the results of our evaluation on EyeDiap. It demonstrates that the face branch is less important on the EyeDiap evaluation, which means that we may design a more efficient way to get the prediction that only requires the eyes part as input.

3.4. Resolution Evaluation

Gaze estimation system is normally required to maintain accuracy over a wide range of distances. Although the normalization process can greatly reduce the input variance caused by different distances by rescaling the image to the proper resolution, it still cannot avoid the loss of useful information which would result in a decline in the final estimation. In order to simulate this information loss due to various distances, down-sampling is performed on the input images as follows: (1) Input image 224×224 is downscaled to 168×168 and upscaled to 224×224; (2) Input image 224×224 is downscaled to 112×112 and upscaled to 224×224.

Fig.11a and Fig. 11b show the results of resolution experiments performed by our method on MPIIGaze and EyeDiap, respectively. The results illustrate that our method is very robust to different distances. Even if the image distance is twice as far as the origin, the result of our method still not degrade.



(a) MPIIGaze

(b) EyeDiap

Figure 11: Resolution study on MPIIGaze and EyeDiap datasets.

3.5. Temporal Model Evaluation

In this session, we evaluate the contribution of adding the temporal model to the static model. Since the MPIIGaze dataset is a discontinuous single image format, we only do evaluation on EyeDiap dataset. Fig.12 shows our evaluation results. The first bar is the result of our static model and the second to fifth bars are the results of the four different Recurrent Neural Network (RNN) models combined with the static model, while the sixth and seventh bars are the results of the network model where the left and right eyes are input as separate branches. From the figure, we can see that no matter which basic network used, the evaluation result always be improved when combining with the temporal model. In all the RNN models used in this paper, the bi-LSTM model contributes the most by improving the accuracy of the static model by about 3%. What's more, all bidirectional RNN models have better results than common RNN models. It shows that we can better improve the final estimation accuracy when adding the backpropagation information to the RNN temporal model. Although, we can see that the accuracy improvement brought by this part is subtle. It is mainly caused by the training data. As described in Section 3.1, the video frames after filtration are no longer continuous video sequences, resulting in a serious loss of temporal information.

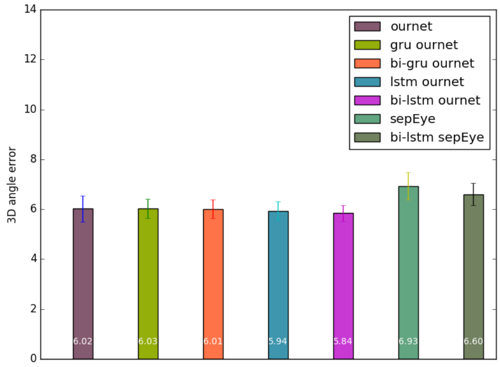


Figure 12: RNN evaluation on EyeDiap dataset.

4. CONCLUSION

In this paper, we have analyzed the relationship between the Itracker model and 3D gaze estimation task. We have effectively modified the Itracker model to not only improve the estimate accuracy, but also effectively reduce the network complexity. We have evaluated the proposed static model at different resolutions. The results have showed that our network is robust to images with different resolutions, which is a great property to the final practical application. Furthermore, we have introduced a bidirectional RNN model (bi-LSTM) to fit the temporal information. To the best of our knowledge, this is the first time that bi-LSTM is used to fit temporal information in the field of gaze estimation to improve the final estimation result. In contrast to the state-of-the-art methods, our method not only significantly improves the accuracy, but also greatly reduces the network size. In the future, we will consider how to effectively improve the performance of our method under a larger head posture application environment.

Acknowledgement

This work was supported in part by National Natural Science Foundation of China (61876168, 61403342, 61325019, 61603341, 61733011, 51575338), Zhejiang

Provincial Natural Science Foundation of China (LY18F030020), and Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering (2017SDSJ03).

References

[1] L. Sun, Z. Liu, M.-T. Sun, Real time gaze estimation with a consumer depth camera, Information Sciences 320 (2015) 346–360.

[2] Z. Zhu, Q. Ji, Eye gaze tracking under natural head movements, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, IEEE, 2005, pp. 918–923.

[3] F. Lu, Y. Sugano, T. Okabe, Y. Sato, Inferring human gaze from appearance via adaptive linear regression, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 153–160.

[4] Y. Sugano, Y. Matsushita, Y. Sato, H. Koike, Appearance-based gaze estimation with online calibration from mouse operations, IEEE Transactions on Human-Machine Systems 45 (6) (2015) 750–760.

[5] X. Zhou, H. Cai, Y. Li, H. Liu, Two-eye model-based gaze estimation from a kinect sensor, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 1646–1653.

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

[7] Z. Zhu, Q. Ji, Novel eye gaze tracking techniques under natural head movement, IEEE Transactions on Biomedical Engineering 54 (12) (2007) 2246.

[8] K. Wang, Q. Ji, Real time eye gaze tracking with kinect, in: International Conference on Pattern Recognition (ICPR), IEEE, 2016, pp. 2752–2757.

[9] X. Zhou, H. Cai, Z. Shao, H. Yu, H. Liu, 3d eye model-based gaze estimation from a depth sensor, in: IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2016, pp. 369–374.

[10] H. Cai, H. Yu, X. Zhou, H. Liu, Robust gaze estimation via normalized iris center-eye corner vector, in: International Conference on Intelligent Robotics and Applications (ICIRA), Springer, 2016, pp. 300–309.

[11] A. Kacete, R. Séguier, M. Collobert, J. Royan, Unconstrained gaze estimation using random forest regression voting, in: Asian Conference on Computer Vision (ACCV), Springer, 2016, pp. 419–432.

[12] Y. Wang, T. Zhao, X. Ding, J. Peng, J. Bian, X. Fu, Learning a gaze estimator with neighbor selection from large-scale synthetic eye images, Knowledge-Based Systems 139 (2018) 41–49.

[13] X. Zhang, Y. Sugano, M. Fritz, A. Bulling, Appearance-based gaze estimation in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4511–4520.

[14] X. Zhang, Y. Sugano, M. Fritz, A. Bulling, Mpiigaze: Real-world dataset and deep appearance-based gaze estimation, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (1) (2019) 162–175.

[15] R. Ranjan, S. De Mello, J. Kautz, Light-weight head pose invariant gaze tracking, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018, pp. 2156–2164.

[16] H. Deng, W. Zhu, Monocular free-head 3d gaze tracking with deep learning and geometry constraints, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 3162–3171.

[17] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, A. Torralba, Eye tracking for everyone, in: IEEE Conference on Computer Vision and Ppattern Recognition (CVPR), 2016, pp. 2176–2184.

[18] X. Zhang, Y. Sugano, M. Fritz, A. Bulling, It's written all over your face: Full-face appearance-based gaze estimation, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2017, pp. 2299–2308.

[19] K. A. Funes Mora, F. Monay, J.-M. Odobez, Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras, in: ACM Symposium on Eye Tracking Research & Applications (ETRA), ACM, 2014, pp. 255–258.

[20] X. Zhang, Y. Sugano, A. Bulling, Revisiting data normalization for appearance-based gaze estimation, in: ACM Symposium on Eye Tracking Research & Applications (ETRA), ACM, 2018, p. 12.

[21] T. Baltrušaitis, P. Robinson, L.-P. Morency, Openface: an open source facial behavior analysis toolkit, in: IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2016, pp. 1–10.

[22] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, A. Bulling, Learning an appearance-based gaze estimator from one million synthesised images, in: ACM Symposium on Eye Tracking Research & Applications (ETRA), ACM, 2016, pp. 131–138.

[23] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, R. Webb, Learning from simulated and unsupervised images through adversarial training., in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, 2017, p. 5.

[24] S. Nie, M. Zheng, Q. Ji, The deep regression bayesian network and its applications: Probabilistic deep learning for computer vision, IEEE Signal Processing Magazine 35 (1) (2018) 101–111.

[25] S. Park, X. Zhang, A. Bulling, O. Hilliges, Learning to find eye region landmarks for remote gaze estimation in unconstrained settings, arXiv preprint arXiv:1805.04771.

[26] C. Palmero, J. Selva, M. A. Bagheri, S. Escalera, Recurrent cnn for 3d gaze estimation using appearance and shape cues, arXiv preprint arXiv:1805.03064.