Real Time Eye Gaze Tracking with 3D Deformable Eye-Face Model

Kang Wang Qiang Ji
ECSE Department, Rensselaer Polytechnic Institute
110 8th Street, Troy, NY, USA
{wangk10, jiq}@rpi.edu

Abstract

3D model-based gaze estimation methods are widely explored because of their good accuracy and ability to handle free head movement. Traditional methods with complex hardware systems (Eg. infrared lights, 3D sensors, etc.) are restricted to controlled environments, which significantly limit their practical utilities. In this paper, we propose a 3D model-based gaze estimation method with a single web-camera, which enables instant and portable eye gaze tracking. The key idea is to leverage on the proposed 3D eye-face model, from which we can estimate 3D eye gaze from observed 2D facial landmarks. The proposed system includes a 3D deformable eye-face model that is learned offline from multiple training subjects. Given the deformable model, individual 3D eye-face models and personal eye parameters can be recovered through the unified calibration algorithm. Experimental results show that the proposed method outperforms state-of-the-art methods while allowing convenient system setup and free head movement. A real time eye tracking system running at 30 FPS also validates the effectiveness and efficiency of the proposed method.

1. Introduction

Eye gaze tracking is to predict the gaze directions or where human looks in real time. Since eye gaze reflects human's cognitive process [16] (attention or interest), various gaze estimation techniques have been proposed and applied in different fields. In Human Computer Interaction field, eye gaze can replace traditional inputs or serve as additional input to help better interactions with the computer. Besides, more and more games start supporting gaze input from eye tracker to enhance the gaming experience [22]. Furthermore, eye tracking data can also help research and analysis in marketing, advertisement, psychology, etc.

There are two main types of gaze estimation techniques: 2D appearance-based and 3D model-based. The key idea of appearance/feature based methods [5, 6, 17, 13, 4] builds on the assumption that similar eye appearances/features correspond to similar gaze positions/directions, from which different mapping functions can be learned to perform gaze estimation. However, these traditional methods cannot handle free head movement, as eye appearance/features might be similar under different gaze directions and head poses. Several methods [15, 23, 33, 24, 32] have been proposed to compensate head movement. But they typically require a large amount of training data involving different poses or rely on additional data to learn a correction model. Differently, 3D model-based methods perform gaze estimation based on a 3D geometric eye model, which mimics the structure and function of human vision system. They can be further divided into two categories based on their hardware systems. Methods from first category [1, 3, 26, 9, 14, 20, 29, 27] depend on their complex hardware system, including IR lights, stereo vision system or 3D sensors to perform 3D gaze estimation. However, the practical utility of these methods is significantly limited due to complex system setup and sensitivity to environmental settings. Methods from second category [2, 25, 29, 31, 11, 12] perform 3D gaze estimation with a simple web-camera. However, these methods either have strong assumptions, cannot apply in practice or cannot give good gaze estimation accuracy. For more information regarding different gaze estimation techniques, we suggest readers refer to [10].

To build a simple, robust, accurate and real time eye tracking system, we focus on 3D model-based gaze estimation with a single web-camera. To achieve our goals, a new gaze estimation framework based on 3D eye-face model is proposed to effectively perform 3D gaze estimation from 2D facial landmarks. The gaze estimation framework also includes the offline learned DEFM and an efficient calibration algorithm. Given DEFM, individual 3D eye-face models from any new subjects can be effectively reconstructed, without the need to provide subject-dependent 3D data or additional hardware. Through the unified calibration procedure, reconstruction of individual 3D eye-face model as well as estimation of personal eye parameters can be achieved simultaneously.

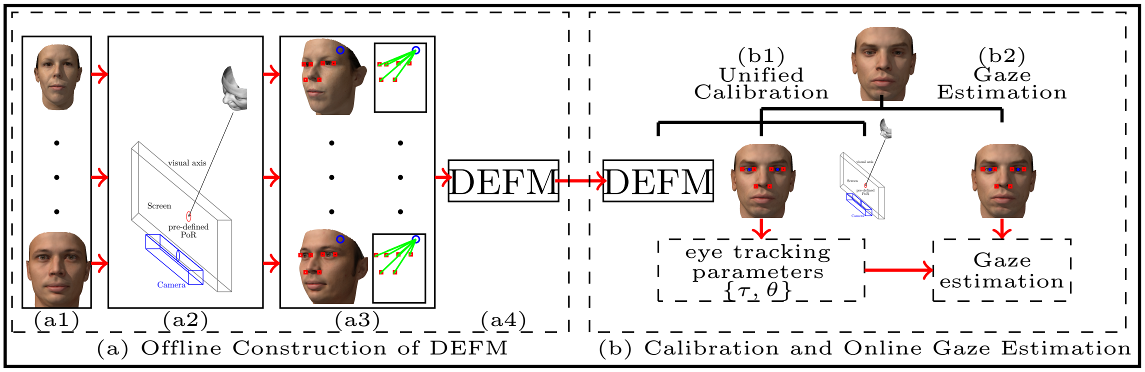


Figure 1. Overview of the proposed gaze estimation framework. (a1) different subjects participate in the offline DEFM construction stage; (a2) 3D facial landmarks are recovered from Kinect sensor, 3D eyeball center is recovered through gaze estimation techniques; (a3) geometric relationship among 3D eyeball center (circle) and 3D rigid facial landmarks (rectangle) are established to construct the 3D eye-face model for each subject; (a4) 3D eye-face models are fused to learn the generic 3D deformable eye-face model (DEFM). In (b1), a unified calibration is performed to simultaneously estimate personal eye parameters $\boldsymbol{\theta}$ and individualize the deformable eye-face model $\boldsymbol{\tau}$; (b2) recovered parameters are used for online eye gaze tracking.

Fig. 1 [1] illustrates the overview of the proposed gaze estimation framework. It consists of an offline DEFM construction stage and a calibration and online gaze estimation stage. During the offline stage, we construct 3D eye-face models for each subject and learn a generic DEFM to represent the entire population. During the online stage, personal eye parameters and individual 3D eye-face model $\{\boldsymbol{\tau}, \boldsymbol{\theta}\}$ are first estimated given DEFM through the calibration algorithm. Real time eye gaze tracking can then be performed given the personal eye parameters and individual 3D eye-face model. In summary, the proposed method makes following novel contributions:

- Propose a 3D eye-face model to enable 3D eye gaze estimation with a single web-camera.
- Eliminate the need to estimate the head-eye offset vector online as used by existing 3D deformable face model based approaches ([29, 25]), therefore yielding improved estimation accuracy and robustness.
- Propose a unified calibration algorithm to simultaneously reconstruct individual 3D eye-face model and estimate personal eye parameters.
- Experimental results and a real time system running at 30 fps validate its effectiveness and efficiency for online eye gaze tracking.

2. Related work

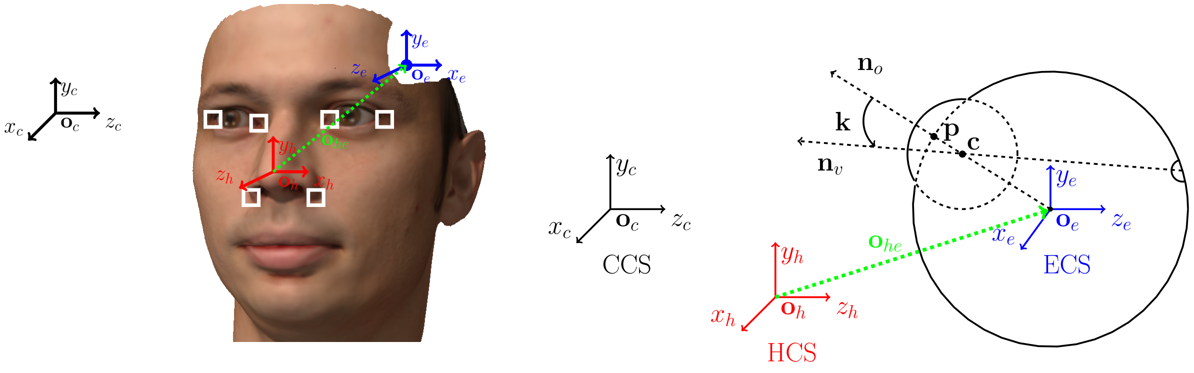
We focus on reviewing 3D model-based eye gaze estimation methods that do not use any infrared lights.

Heinzmann *et al* [11] approximated the 3D gaze direction by the facial normal direction and eye direction relative to facial normal. However, eye direction is sensitive to head pose and eye corner detections, which results in poor accuracy. Ishikawa *et al* [12] proposed to first estimate head pose with AAM model, and then approximated eye gaze with the relative eye directions. However, its use of face scale to estimate 3D information is not robust, and the use of appearance might be affected by illumination and pose variations. Yamazoe *et al* [31] first employed the factorization method [19] to recover the 3D face/eye shape model from image sequences. Then 3D eyeball center and pupil center can be recovered by fitting the model to the 2D image data. However, the structure from motion like algorithm is sensitive to noise and subject's head motion, which accounts for their poor gaze estimation accuracy. Chen *et al* [2] estimated eye gaze using an extended eye model with eye corners. The depth information is estimated from inverse face scale. However, generic face scale factor is sensitive to noise and feature detections, which results in poor accuracy.

In [25, 29], the authors first recover the head rotation and translation from detected 2D facial landmarks. Then they approximated the 3D eyeball center by adding a fixed offset to the head translation and approximated 3D pupil center with geometry constraints. Final eye gaze can be estimated from 3D eyeball center, 3D pupil center, and some personal parameters. However, [25] totally ignored personal eye parameters and only approximated the true gaze directions with the optical axis. And [29] required subject-specific 3D data to estimate the head translation, which is unrealistic in practice.

Recently, Wood *et al* [28] proposed to estimate eye gaze based on a 3D morphable eye region model, which consists of a set of 3D eye shape bases and eye texture bases. Given a testing image, they reconstruct a new image by linear combination of the bases that matches the testing image in

[1] All the portraits are taken from [18].



(a) 3D eyeball center ($\mathbf{o}_e$) estimation given 2D facial landmarks (white rectangle).

(b) 3D eye gaze estimation with 3D geometric eye model.

Figure 2. 3D model-based gaze estimation with facial landmarks. CCS (black), HCS (red) and ECS (blue) represent Camera, Head and Eyeball Coordinate Systems respectively.

both texture and shape, final gaze can be inferred from the learned parameters. Despite their usage of 3D eye model to construct the 3D eye region, their method still relies heavily on eye appearance and is therefore affected by appearance variations like glasses.

In this paper, we propose a 3D eye-face model to directly relate 3D eyeball center with 2D facial landmarks, from which we can effectively recover 3D eyeball center without the need to introduce the head-eye offset. As a result, the proposed 3D eye-face model based gaze estimation enables robust and accurate eye gaze tracking in practical applications.

3. 3D Model-based eye gaze estimation

The proposed method consists of two major steps as illustrated in Fig. 2. We first leverage on the proposed DEFM to estimate 3D eyeball center $\mathbf{o}_e$ from observed 2D facial landmarks (Fig. 2 (a)), then 3D eye gaze can be computed by leveraging on the 3D geometric eye model (Fig. 2 (b)). 3D deformable eye-face model encodes the geometric relationship among facial landmarks and eyeball center across the entire population. For online eye gaze tracking with a particular user, a personal calibration is required to individualize the DEFM to a specific 3D eye-face model for that user. We leave the construction of DEFM in Sec. 4 and the personal calibration in Sec. 5, and focus on estimating 3D eye gaze given the individualized 3D eye-face model.

3.1. 3D Eyeball center estimation

3D eye-face model is defined as the facial landmark coordinates $\{\mathbf{x}_i^e\}_{i=1}^N$ in ECS (Fig. 2). It reflects the relative position information of landmarks w.r.t eyeball center. The 3D eye-face model is related with 2D facial landmarks through: $\lambda_i\mathbf{W}\mathbf{x}_i^{2D} = (\mathbf{R}_e\mathbf{x}_i^e + \mathbf{o}_e)$, where $\mathbf{x}_i^{2D}$ represents the i^{th} 2D landmark in homogeneous form, $\mathbf{W}$ is the inverse camera intrinsic parameters, λ is the scale factor and $\{\mathbf{R}_e, \mathbf{o}_e\}$ is the rotation and translation. For notation convenience, we denote $\mathbf{x}_i^p = \mathbf{W}\mathbf{x}_i^{2D}$.

Since we are only interested in recovering eyeball center $\mathbf{o}_e$, we do not need estimate rotation matrix $\mathbf{R}_e$. Leveraging on $\mathbf{R}_e$ is orthonormal, we can manipulate the equation to have $\mathbf{R}_e$ eliminated, yielding:

$$(\mathbf{R}_e\mathbf{x}_i^e)^T(\mathbf{R}_e\mathbf{x}_i^e) = (\lambda_i\mathbf{x}_i^p - \mathbf{o}_e)^T(\lambda_i\mathbf{x}_i^p - \mathbf{o}_e) = \mathbf{x}_i^{eT}\mathbf{x}_i^e$$

In practice, the distance of facial landmarks to the camera is close to each other, therefore the scale factor λ_i are assumed to be the same. Eyeball center $\mathbf{o}_e$ and λ can be solved by:

$$\{\lambda^*, \mathbf{o}_e^*\} = \arg\min_{\lambda,\mathbf{o}_e} f(\lambda, \mathbf{o}_e) \tag{1}$$

$$f(\lambda, \mathbf{o}_e) = \sum_{i=1}^{N} ||(\lambda\mathbf{x}_i^p - \mathbf{o}_e)^T(\lambda\mathbf{x}_i^p - \mathbf{o}_e) - \mathbf{x}_i^{eT}\mathbf{x}_i^e||^2.$$

To understand the benefits of the proposed 3D eye-face model, we briefly review 3D face model based methods [25, 29] on 3D eyeball center estimation. 3D face model is defined as facial landmarks $\{\mathbf{x}_i^h\}_{i=1}^N$ in HCS. The origin of HCS is typically on the face (Eg. nose tip). They first solve the head pose:

$$\mathbf{R}_h^*, \mathbf{o}_h^* = \arg\min_{\mathbf{R}_h,\mathbf{o}_h} \sum_{i=1}^{N} ||\lambda_i(\mathbf{R}_h\mathbf{x}_i^h + \mathbf{o}_h) - \mathbf{W}\mathbf{x}_i^{2D}||^2$$

$$s.t \quad \mathbf{R}_h\mathbf{R}_h^T = \mathbf{I} \tag{2}$$

3D eyeball center $\mathbf{o}_e$ can then be computed as $\mathbf{o}_e^* = \mathbf{o}_h^* + \mathbf{R}_h^*\mathbf{o}_{he}$ (See Fig. 2 for a geometric illustration), where $\mathbf{o}_{he}$ represents the head-eye offset in HCS. Notice $\mathbf{o}_{he}$ is a personal parameter and needs be estimated.

Compared to 3D face model, the proposed 3D eye-face model brings several benefits. Firstly, eyeball center can

Table 1. Comparison of DFM and DEFM based eye gaze estimation.

Category	3D **D**eformable **F**ace **M**odel	3D **D**eformable **E**ye-**F**ace **M**odel
Offline construction requirements	3D facial landmarks	3D facial landmarks and 3D eyeball center
Offline personal calibration	$\{\boldsymbol{\theta}, \boldsymbol{\tau}, \mathbf{o}_{he}\}$	$\{\boldsymbol{\theta}, \boldsymbol{\tau}\}$
Individualized model	$\{\mathbf{x}_i^h\}_{i=1}^N$ in HCS	$\{\mathbf{x}_i^e\}_{i=1}^N$ in ECS
Online eyeball center estimation	$\mathbf{o}_e^* = \mathbf{o}_h^* + \mathbf{R}_h^* \mathbf{o}_{he}$, $\{\mathbf{R}_h^*, \mathbf{o}_h^*\}$ are from solving Eq. (2).	$\mathbf{o}_e^*$ from solving unconstrained problem in Eq. (1)
Pros	• Easy offline construction, no need of 3D eyeball center.	• More robust against head pose variations and image noise. • Gaze estimation is more accurate. • Online processing is simple and efficient.
Cons	• Needs calibrating $\mathbf{o}_{he}$ for each user. • Calibration is sensitive to initialization and noise. • Not robust against head pose and image noise.	• Offline Construction requires 3D eyeball center

be solved more easily with an unconstrained optimization problem (Eq. (1)) instead of a constrained one (Eq. (2)). More importantly, 3D eye-face model explicitly eliminates the head-eye offset $\mathbf{o}_{he}$, which yields much more accurate and robust eye gaze tracking. Tab. 1 shows a side-by-side comparison of DEFM-based and DFM-based eye gaze tracking systems. The pros and cons of the two methods will be further demonstrated in later analysis and experiments (Sec. 6).

3.2. Gaze estimation with 3D geometric eye model

After solving 3D eyeball center $\mathbf{o}_e$, we can compute the 3D pupil center $\mathbf{p}$ given detected 2D pupil center $\mathbf{p}^{2D}$. According to camera projection model, $\mathbf{p}$ lies on a line l passing through camera center $\mathbf{o}_c$. Besides, as in Fig. 2 (b), $\mathbf{p}$ also lies on eyeball sphere with calibrated radius r_e. By solving a line-sphere intersection problem, we can estimate the 3D pupil center $\mathbf{p}$ as:

$$\mathbf{p}^* = \begin{cases} z\mathbf{x} & \text{if } ||z\mathbf{x} - \mathbf{o}_e|| > r_e \\ \frac{\mathbf{x}^T\mathbf{o}_e - \sqrt{||\mathbf{x}^T\mathbf{o}_e||^2 - \mathbf{x}^T\mathbf{x}(\mathbf{o}_e^T\mathbf{o}_e - r_e^2)}}{\mathbf{x}^T\mathbf{x}}\mathbf{x} & \text{otherwise} \end{cases} \tag{3}$$

where $\mathbf{x} = \mathbf{W}\mathbf{p}^{2D}$, and $z = \frac{\mathbf{x}^T\mathbf{o}_e}{||\mathbf{x}||}$ is the shortest distance from $\mathbf{o}_e$ to line l. If line l does not intersect with eyeball sphere, we choose the closest point $z\mathbf{x}$ as an approximation of pupil center (very rare case in practice).

Given $\mathbf{o}_e$ and $\mathbf{p}$, optical axis can be computed as: $\mathbf{n}_o = (\mathbf{p} - \mathbf{o}_e)/||\mathbf{p} - \mathbf{o}_e||$. As an unit length vector in $\mathcal{N}^{3\times1}$, $\mathbf{n}_o$ can also be represented as pitch angle ϕ and yaw angle γ:

$$\mathbf{n}_o(\phi, \gamma) = \begin{pmatrix} \cos(\phi)\sin(\gamma) \\ \sin(\phi) \\ -\cos(\phi)\cos(\gamma) \end{pmatrix}$$

From eyeball anatomy, true gaze direction is determined by visual axis $\mathbf{n}_v$, therefore we need to add calibrated $\mathbf{k} = [\alpha, \beta]$ to optical axis: $\mathbf{n}_v = \mathbf{n}_o(\phi + \alpha, \gamma + \beta)$.

2D point of regard (PoR) can be computed by intersecting visual axis with the display plane: PoR $= \mathbf{o}_e + \lambda\mathbf{n}_v$. Here we use $\mathbf{o}_e$ as origin of visual axis instead of $\mathbf{c}$. The error caused by this approximation is negligible as $||\mathbf{o}_e - \mathbf{c}||$ is only a few millimeters and is much smaller than eye-plane distance. The scalar λ can be computed given the rotation and translation of the display plane relative to CCS. Based on above equations, we denote 2D PoR estimation as:

$$\text{PoR} = f(\mathbf{o}_e, \mathbf{p}; \boldsymbol{\theta}) \tag{4}$$

where $\boldsymbol{\theta} = [\alpha, \beta, r_e]$ represents all the person-dependent eye parameters.

4. 3D Deformable eye-face model

The construction of DEFM or 3D eye-face model requires additional 3D eyeball center position $\mathbf{o}_e$, which is inside head and invisible from the sensors. Therefore we propose to leverage on gaze estimation techniques to estimate 3D eyeball center during offline construction.

4.1. Construction of individual 3D eye-face model

In this paper, we propose to use a Kinect sensor to retrieve 3D facial landmark positions. The 3D eyeball center can be estimated with gaze estimation techniques. As shown in Fig. 1 (a2), subject is asked to perform a 9-points calibration with a chin rest. The chin rest is to ensure eyeball center $\mathbf{o}_e$ remain unchanged over the time. 2D facial landmarks can be detected and their 3D positions can be recovered. Given M pairs of 3D pupil center and 2D PoR $\{\mathbf{p}_m, \mathbf{g}_m\}_{m=1}^M$, we can solve 3D eyeball center and personal eye parameters as follows:

$$\begin{aligned} \mathbf{o}_e^*, \boldsymbol{\theta}^* = \arg\min_{\mathbf{o}_e, \boldsymbol{\theta}} \sum_{m=1}^{M} ||f(\mathbf{o}_e, \mathbf{p}_m; \boldsymbol{\theta}) - \mathbf{g}_m||^2 + \\ \lambda\, ||d(\mathbf{o}_e, \mathbf{p}_m) - r_e||^2 \\ s.t \quad \boldsymbol{\theta}_l \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_h \end{aligned} \tag{5}$$

The first term represents the prediction error and $f(\cdot)$ is defined in Eq. (4). The second regularization term forces the estimation to give a consistent and reasonable eyeball radius with $d(\cdot)$ the function to compute the Euclidean distance between two vectors, and λ the weight to balance these two terms. $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_h$ represents the lower and higher bounds of the eye parameters whose value can be approximated from eyeball anatomy [9]. The optimization problem is solved

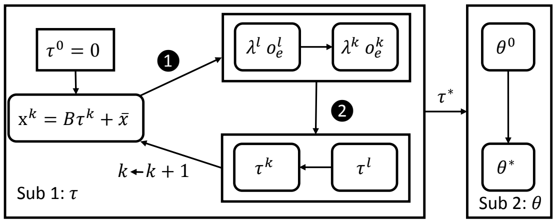


Figure 3. Overview of calibration flow. In the first sub-problem, we solve $\{\lambda, \mathbf{o}_e\}$ and $\boldsymbol{\tau}$ alternately leveraging on Eq. (1). Linear approximated solutions $\{\lambda^l, \mathbf{o}_e^l, \boldsymbol{\tau}^l\}$ are used as initializations. The estimated $\boldsymbol{\tau}^*$ are used to construct the 3D eye-face model and solve the personal eye parameters $\boldsymbol{\theta}$.

by alternating between $\mathbf{o}_e$ and $\boldsymbol{\theta}$ with an iterative algorithm. Given estimated $\mathbf{o}_e^*$ and $\{\mathbf{x}_i^c\}_{i=1}^N$ in CCS, 3D eye-face model can be computed as $\mathbf{x}_i^e = \mathbf{x}_i^c - \mathbf{o}_e^* \quad \forall i$ for each training subject.

4.2. Construction of generic 3D deformable eye-face model

To generalize to a new subject during eye gaze tracking, we propose to construct a generic 3D deformable eye-face model. We first stack one 3D eye-face model as a column vector $\mathbf{x}^e \in \mathcal{R}^{3N\times 1}$, then we stack all individual subject-specific 3D eye-face models from different subjects to form a large matrix $\mathbf{A} \in \mathcal{R}^{3N\times L}$, where each column of $\mathbf{A}$ represents one 3D eye-face model. Since different subjects have roughly similar skull anatomy and face shape, hence we believe the column space of $\mathbf{A}$ is able to cover a large range of subjects. The 3D eye-face model for a new subject can be reconstructed by the linear combination of columns of $\mathbf{A}$. However, without reduction, the linear combination coefficients contain L variables, which is difficult to estimate. Therefore for compact representation, we perform a Principal Component Analysis on $\mathbf{A}$ to produce $q \ll L$ basis vectors $\mathbf{b}_k \in \mathbf{R}^{3N\times 1}$. 3D eye-face model for a new subject can, therefore, be reconstructed as:

$$\mathbf{x}^e = \sum_{k=1}^{q} \mathbf{b}_k \tau_k + \bar{\mathbf{x}} = \mathbf{B}\boldsymbol{\tau} + \bar{\mathbf{x}} \tag{6}$$

with $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_q]$ the matrix of concatenated bases, $\boldsymbol{\tau} = [\tau_1, ..., \tau_q]^T$ the coefficients, and $\bar{\mathbf{x}}$ the average model (column average of $\mathbf{A}$). Since any 3D eye-face models can be reconstructed from the bases and average model, we denote $\{\mathbf{B}, \bar{\mathbf{x}}\}$ as generic 3D **deformable eye-face model** (DEFM).

We want to note that construction of DEFM is a one-time offline process that only applies to training subjects. It is not necessary for any new testing subjects.

5. A Unified Calibration Algorithm

To start eye gaze tracking for a new subject, a one-time personal calibration procedure is required. Once all parameters are calibrated, we can follow Eqs. (6), (1), (3) and (4) to compute 2D PoR on the screen or 3D visual axis.

```
Algorithm 1: Calibration Algorithm
1. input : { 2D landmarks and pupil: $\mathbf{x}_{mi}^{2D}, \mathbf{p}_m^{2D}$
           { Groundtruth PoRs: $\mathbf{g}_m$
           { Deformable eye-face model: $\mathbf{B}, \bar{\mathbf{x}}$
2. output: $[\boldsymbol{\tau}^*, \boldsymbol{\theta}^*]$
3. Initialization: $\boldsymbol{\tau}^0 = \mathbf{0}; \boldsymbol{\theta}^0 =$ human average[9]
4. Sub-problem 1, Solve $\boldsymbol{\tau}$:
   $L(\lambda, \mathbf{o}_e, \boldsymbol{\tau}) = \sum_{i=1}^N ||(\lambda\mathbf{x}_i^p - \mathbf{o}_e)^T(\lambda\mathbf{x}_i^p - \mathbf{o}_e) - (\mathbf{B}_i\boldsymbol{\tau} + \bar{\mathbf{x}}_i)^T(\mathbf{B}_i\boldsymbol{\tau} + \bar{\mathbf{x}}_i)||^2$, where $\mathbf{B}_i$ and $\bar{\mathbf{x}}_i$ represent the corresponding rows of the $i^{th}$ facial landmark.
   while not converge do
       $\{\lambda, \mathbf{o}_e\}^k = \arg\min_{\lambda, \mathbf{o}_e} L(\lambda, \mathbf{o}_e, \boldsymbol{\tau}^{k-1})$
       $\boldsymbol{\tau}^k = \arg\min_{\boldsymbol{\tau}} L(\lambda^k, \mathbf{o}_e^k, \boldsymbol{\tau}) + \gamma||\boldsymbol{\tau}||^2$     (7)
   end
5. Sub-problem 2, Solve $\boldsymbol{\theta}$:
   $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{m=1}^{M} ||f(\mathbf{o}_e^*, \mathbf{p}_m^*; \boldsymbol{\theta}) - \mathbf{g}_m||^2$
   $s.t \quad \boldsymbol{\theta}_l < \boldsymbol{\theta} < \boldsymbol{\theta}_h$     (8)
   where $f(\cdot)$ is defined in Eq. (4), $\mathbf{o}_e^*$ can be computed using Eq. (1), $\mathbf{p}_m^*$ can be computed using Eq. (3).
```

Calibrating $[\boldsymbol{\tau}, \boldsymbol{\theta}]$ simultaneously is challenging as it requires solving a complex non-convex optimization problem. We, therefore, decouple the original problem to two sub-problems with respect to $\boldsymbol{\tau}$ and $\boldsymbol{\theta}$, where each subproblem can be effectively solved with proper regularizations. Fig. 3 illustrates the overall calibration flow, and the detailed calibration algorithm is summarized in Alg. 1.

In sub-problem 1, we define a loss function $L(\lambda, \mathbf{o}_e, \boldsymbol{\tau})$. It is an extension of $f(\lambda, \mathbf{o}_e)$ in Eq. (1) where the 3D eye-face model $\mathbf{x}_e$ is now a function of the deformable coefficients $\boldsymbol{\tau}$. To accelerate convergence, linear solutions $\{\lambda^l, \mathbf{o}_e^l, \boldsymbol{\tau}^l\}$ are used for initializations. Linear solutions are obtained by ignoring orthogonal constraints and assume weak perspective projection. Notice in Eq. (7), a regularization term $||\boldsymbol{\tau}||^2$ is imposed to penalize coefficients that yield large distance to average model, which help prevent over-fitting and unrealistic face shapes. In practice, the weight $\gamma \in [0.3, 0.5]$ can give good results. The estimated $\boldsymbol{\tau}^*$ is used to construct the 3D eye-face model (Eq. (6)), from which we can solve $\boldsymbol{\theta}$ by minimizing PoR prediction error as in Eq. (8).

Comparison with DFM. Calibration of head-eye offset

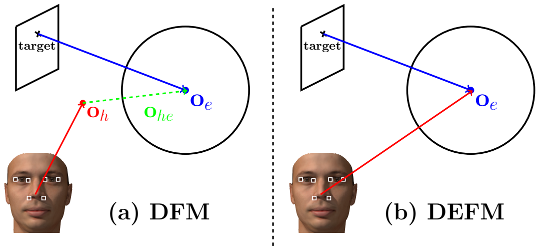


Figure 4. Geometric illustration of calibration in DFM and DEFM.

$\mathbf{o}_{he}$ causes issues for DFM-based methods. As shown in Fig. 4(a), the basic idea is to estimate head center $\mathbf{o}_h$ from 2D landmarks and eyeball center $\mathbf{o}_e$ from known calibration marker/target. Offset vector is simply the subtraction of $\mathbf{o}_h$ from $\mathbf{o}_e$. However, the estimation of both $\mathbf{o}_h$ and $\mathbf{o}_e$ is affected by noise and optimization algorithms. The offset vector introduces more freedom and are prone to over-fitting. As a consequence, the estimated parameters cannot generalize to different poses and are sensitive to image noise. On the contrary as in Fig. 4(b), we explicitly eliminate the offset vector. Both facial landmarks and screen target are directly connected to the intermediate eyeball center $\mathbf{o}_e$. This helps regularize the problem and yields better parameters for eye gaze estimation.

6. Experiments and Analysis

6.1. Experimental Settings

We use a Kinect sensor for offline DEFM construction and a Logitech webcam C310 for online gaze estimation. Both cameras are placed under the 21.5 inch LCD monitor (1920 × 1080) around the central region. The frame resolution is set to 640 × 480 for both cameras. 2D facial landmarks and pupil center can be effectively detected with the algorithms in [30] and [8]. We evaluate the proposed method on following three types of data.

Simulated data: from the learned 3D deformable eye-face model, we can simulate different subjects by adjusting the coefficients τ in Eq. (6). Then we can generate 3D facial landmarks by providing head rotation and translation. Finally, 2D landmarks (observation) can be obtained by projecting the 3D facial landmarks onto 2D image plane. Different noise can be added to simulate detection error.

Real data from 10 subjects: seven male and three female subjects participate in the experiments. The distance between subjects and camera ranges from 450 mm to 650 mm, and subjects typically move within 200 mm in vertical direction and 400 mm in horizontal direction. Calibration data is collected with a 5-points pattern, while evaluation data is collected densely with a 45-points (9 × 5) pattern. We collect the evaluation data at 5 different positions for better head pose coverage.

Benchmark datasets: We select two datasets which provide full-face images. Columbia Gaze dataset [21] contains 56 subjects, each with 21 gaze angles and 5 head pose angles. We perform cross-validation on each subject with 15 images for calibration and 90 images for testing. EyeDiap dataset [7] contains 16 subjects with different sessions. We select 20 images for calibration and test on 500 images from each subject.

6.2. Experiments on simulated data

For all experiments on simulated data, we use 9 rigid landmarks (Configuration 2 in Fig. 9). And except the study on sensitivity against image noise, we add ~ 1 pixel noise on 2D landmarks and pupil centers.

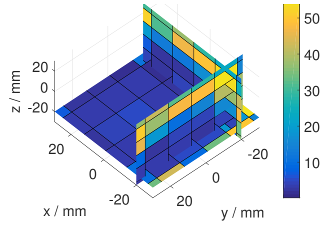


Figure 5. Objective contour around groundtruth offset $\mathbf{o}_{he}$.

Calibration $\mathbf{o}_{he}$ is sensitive to initializations. Gaze estimation with DFM requires calibration of additional offset $\mathbf{o}_{he}$ for each user. However, the objective function is highly nonlinear w.r.t $\mathbf{o}_{he}$ as can be seen in Fig. 5. We cannot obtain a good solution with one single initialization. To achieve a similar calibration/training error of DFM compared to DEFM, we need to randomly initialize multiple times, which may cost 40× more time.

Table 2. Head pose study with 2D landmark and pupil noise at 1 pixel. e1, e2 and e3 represent calibration error, testing error under calibration pose and testing error under a different pose.

True θ			Method	Est θ			e1	e2	e3
5.0	1.2	13.0	DEFM	5.4	1.7	12.4	2.7	3.7	3.6
			DFM	4.0	−0.6	13.5	2.8	3.3	17.7
1.5	−3.7	15.0	DEFM	1.7	−3.5	14.7	2.3	2.9	2.9
			DFM	4.8	−2.4	15.7	2.3	2.9	14.6
−1.0	3.4	11.0	DEFM	0.6	3.3	10.5	3.6	4.6	4.5
			DFM	2.9	1.8	11.3	3.1	3.8	21.3

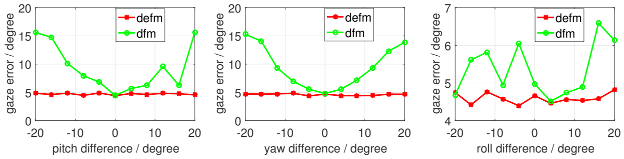


Figure 6. Gaze estimation with different head poses relative to calibration head pose.

Robustness against head pose variations. Tab. 2 shows the estimated $\boldsymbol{\theta}$ and the gaze estimation performance on different poses. For DFM, the estimated $\boldsymbol{\theta}$ is far away from groundtruth, though the inaccurate parameters can still give small testing error on the same head pose. But due to the inaccurate $\boldsymbol{\theta}$, it cannot give good results on a different head pose. On the contrary, DEFM gives consistent errors across head poses, as the estimated $\boldsymbol{\theta}$ is close to groundtruth. We also systematically study how head rotation (pitch, yaw, roll) affect the performance as shown in Fig. 6. DFM degrades significantly with pitch and yaw changes. The influence of roll angle is less significant as it is in-plane rotation. The proposed DEFM performs consistently well across different head poses and thus is more suited for robust and accurate eye gaze tracking in practice.

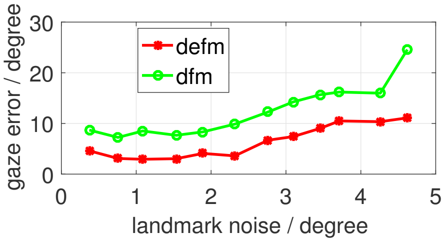


Figure 7. Gaze estimation error with different level of 2D landmark noise, 2D pupil noise is set to 1 pixel.

Accuracy w.r.t 2D landmark noise. As pupil noise causes similar effects on both DEFM and DFM, we set it to ~ 1 pixel. As shown in Fig. 7, with increased 2D landmark noise, both methods perform worse, but DEFM is consistently better than DFM with a margin of 5 degree. The figure comes from average results from different testing poses around calibration pose. Therefore we believe DEFM can give much accurate results in challenging settings.

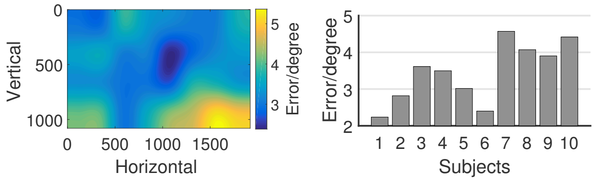
6.3. Experiments on real data from 10 subjects

Table 3. Statistics of head translation (mm) and rotation (degree)

Type	mean	±	std
x-translation	−30.0	±	32.0
y-translation	−11.0	±	27.0
z-translation	525.0	±	39.0
yaw (left or right)	3.0	±	19.5
pitch (up or down)	2.8	±	12.7
roll (in plane rotation)	−82.0	±	7.1

Head pose analysis. The head pose statistics for all subjects are listed in Tab. 3. The head pose coverage is sufficient to naturally cover the screen region. 3D model-based methods theoretically can handle large head motion, as head rotation and translation are explicitly encoded in the 3D model. The results in Fig. 6 with synthetic data validates the point. But in practice, large head motion causes poor landmark detections, which is the main reason for performance drop. Besides using improved landmark detectors, several practical tricks, such as using right eye when left eye is occluded, can be applied to improve the accuracy.

Overall performance for 10 subjects. Fig. 8 shows the average error for each subject as well as the overall error heatmap normalized in degree from all subjects. The 9×5 heatmap is resized to screen size (1920×1080) for better illustration. The average error for all subjects is 3.5 degree. The poor performance (5.5 degree) on the corner/boundary regions is mainly caused by poor feature detections when subjects look at extreme directions or with large head motions. However, for most of the central regions, the error is relatively small around 3.0 degree.



(a) Error heatmap for all subjects (b) Error for each subject

Figure 8. Overall error heatmap and average error for each subject. The results are based on configuration 2 of rigid facial landmarks as shown in Fig. 9.

Table 4. Gaze estimation performance with different pupil detection algorithms.

	Manual	Method in [29]	Ours
Error / degree	2.5	3.1	3.5
Time / ms	4000.0	30.0	2.3

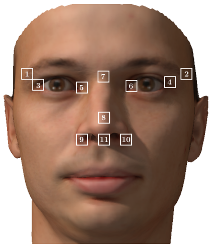
Comparison of pupil detection algorithms. As listed in Tab. 4, the first algorithm manually annotated the 2D pupil positions, the second one is the starburst algorithm used in [29], and the last one is ours from [30]. Manually annotated pupil centers yield the best accuracy, but it takes huge amount of time. Nevertheless, it demonstrates that with better feature detections, the proposed DEFM can give much better accuracy. The one used in [29] gives better performance but it takes 30 ms for a single image, which might not suffice the need for real time eye gaze tracking. We find a compromise between accuracy and efficiency, allowing real time eye gaze tracking while reserving good accuracy.

Table 5. Running time of eye tracking system in milliseconds.

landmark detection/tracking	pupil detection	gaze estimation	misc	total
25.7	2.3	0.3	5.0	33.3

Real time performance The eye tracking system runs on a windows system with Inter Core i7-4770 CPU (3.40 GHz) and 16 GB memory. The code is written in Matlab. We perform online eye gaze tracking for 5 minutes and collected around 9000 frames. The average time for major components during eye gaze tracking are listed in Tab. 5, which allows real time eye gaze tracking at 30 fps.

Evaluation on landmark configurations. The robustness of the system relies heavily on facial landmarks. By analyzing how facial landmarks vary with pose/expression variations, we identify 11 candidate rigid facial landmarks as shown in Fig. 9, including 2 eyebrow corners, 4 eye corners,



Index	Configurations	Error
1	$\{1, 2, 7 - 11\}$	4.1
2	$\{1 - 4, 7 - 11\}$	3.5
3	$\{1, 2, 5, 6, 7 - 11\}$	3.7
4	$\{1 - 11\}$	3.9
5	$\{3 - 11\}$	4.8

Figure 9. Candidate landmarks configurations and their results.

2 points on nose bridge and 3 points under nose. We plan to use symmetric points and eliminate points that are too close to each other (Eg. $\{7 - 11\}$). We end up with 5 configurations and gaze estimation error with each configuration is listed in the right part of Fig. 9.

We have two major observations: 1) Eye corners tend to move toward the direction the eyeball rotates. For example, when we fix our head position and look up, the eye corners also move up. This is not caused by the detection but the real eye appearance changes because of eyeball rotation. 2) Eye brows tend to be occluded under large head motions. We find that configurations without eye corners (Config. 1) and configurations without eyebrows (Config. 5) cannot give accurate results. But their combinations (Config. 2, 3, and 4) can alleviate the issues of eye corner motion and eyebrow occlusion and give much better results.

Table 6. Gaze estimation error with different 3D shape model.

	Original	Calibrated	Average
DEFM	3.2	3.5	4.5
DFM	5.1	5.7	6.5

Evaluation of different 3D (eye) face model. In practice, we use the individualized 3D eye-face model for a new subject. We are also interested in the performance of original model learned from 3D data and the average model. We also compare with DFM-based method (an extension of [29]). For the 10 training subjects, we use their offline learned 3D shape model to perform gaze estimation, and obtain an error of 3.2 and 5.1 for DEFM and DFM respectively. The individualized model obtain similar results with 3.5 and 5.7 respectively, which demonstrates the calibration algorithm can individualize to a new subject. The average model ($\tau = 0$) gives larger error but is most suitable for applications without personal calibration.

Table 7. Comparison with state-of-the-art method

Method	[2]	[29]	[25]	**Ours**
Error/degree	7.3	5.7	7.2	3.5

Comparison with state-of-the-art landmark-based methods. We compare with other landmark-based methods in [2, 29, 25]. As they did not release their data or code, we try our best to implement their methods. Because of usage of a generic face scale factor, which is sensitive to noise and poor detections, [2] gives an error of 7.3 degree. [29] relies on original 3D face model for each subject. It cannot adapt for a new subject, and it suffers the issues of DFM-based approaches and cannot give good accuracy. As for the method presented in [25], they estimate 3D pupil based on a generic 2D-3D mesh correspondence, and approximate 3D gaze by the optical axis. Their method, therefore, cannot generalize to different subjects and give good accuracy.

6.4. Experiments on benchmark datasets

Table 8. Comparison with state-of-the-art on benchmark datasets.

Dataset	[28]	[2]	[29]	[25]	**Ours**
ColumbiaGaze	8.9	12.3	9.7	10.2	7.1
EyeDiap (V, S)	21.5	32.3	21.3	22.2	17.3
EyeDiap (H, M)	22.2	35.7	25.2	28.3	16.5

On ColumbiaGaze [21], we outperform the three landmark-based methods [3, 29, 25] with a big margin. Compared to [28] which only evaluated on selected 680 images without glasses, we achieve 7.1 degree error for all 3570 images without glasses, and 8.2 degree for all 5880 images. [28] cannot handle glasses as they significantly change the eye appearance, while our model-based method is more robust to appearance variations and can still give good results.

On EyeDiap[7] with VGA camera (640 × 480) and static head motion (V, S), the proposed method also outperforms all the 4 competing methods. The large error is due to the extremely small eye images ($\sim 13 \times 6$ pixel). With head movement and HD camera (1920×1080) (H, M), ours shows more robust and accurate results while [25, 29] cannot handle head movement well.

To summarize, the proposed method is more robust against head movement compared to model-based methods, and is less sensitive to appearance variations compared to appearance-based methods.

7. Conclusion

In this paper, we propose a new 3D model-based gaze estimation framework, which enables simple, accurate and real time eye gaze tracking with a single web-camera. With the proposed 3D eye-face model, 3D eye gaze can be effectively estimated from 2D facial landmarks during online gaze tracking. A 3D deformable eye-face model learned offline also facilitates efficient online calibration for a new subject, without the need to provide extra hardware or person specific 3D data. Compared to state-of-the-art methods, the proposed method not only gives better gaze estimation accuracy but also allows natural head movement and real time eye gaze tracking.

Acknowledgements: This work was supported in part by a National Science Foundation grant (IIS 1539012) and in part by RPI-IBM Cognitive Immersive Systems Laboratory (CISL).

References

[1] D. Beymer and M. Flickner. Eye gaze tracking using an active stereo head. In *Computer Vision and Pattern Recognition*, 2003. 1

[2] J. Chen and Q. Ji. 3d gaze estimation with a single camera without ir illumination. In *Pattern Recognition, 19th International Conference on*, 2008. 1, 2, 8

[3] J. Chen, Y. Tong, W. Gary, and Q. Ji. A robust 3d eye gaze tracking system using noise reduction. In *Proceedings of the 2008 symposium on Eye tracking research and applications*, 2008. 1, 8

[4] E. Demjen, V. Abosi, and Z. Tomori. Eye tracking using artificial neural networks for human computer interaction. 2011. 1

[5] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Inferring human gaze from appearance via adaptive linear regression. *In Proc. International Conference on Computer Vision*, 2011. 1

[6] K. Funes Mora and J. Odobez. Geometric generative gaze estimation (g3e) for remote rgb-d cameras. *Computer Vision and Pattern Recognition*, 2014. 1

[7] K. A. Funes 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 *ETRA*, 2014. 6, 8

[8] C. Gou, Y. Wu, K. Wang, K. Wang, F.-Y. Wang, and Q. Ji. A joint cascaded framework for simultaneous eye detection and eye state estimation. *Pattern Recognition*, 67:23–31, 2017. 6

[9] E. D. Guestrin and E. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *Biomedical Engineering, IEEE Transactions on*, 2006. 1, 4, 5

[10] D. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *TPAMI*, 2010. 1

[11] J. Heinzmann. 3-D facial pose and gaze point estimation using a robust real-time tracking paradigm. *Face and Gesture Recognition,*, 1998. 1, 2

[12] T. Ishikawa, S. Baker, I. Matthews, and T. Kanade. Passive Driver Gaze Tracking with Active Appearance Models. *Proc. 11th World Congress Intelligent Transportation Systems*, 2004. 1, 2

[13] K. -H. Tan, D.Kriegman, and N. Ahuja. Appearance-based eye gaze estimation. *In Proc. 6th IEEE Workshop on Applications of Computer Vision*, 2002. 1

[14] J. Li and S. Li. Eye-model-based gaze estimation by rgb-d camera. *Computer Vision and Pattern Recognition Workshops*, 2014. 1

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

[16] M. Mason, B. Hood, and C.Macrae. Look into my eyes : Gaze direction and person memory. *Memory*, 2004. 1

[17] O. Williams, A. Blake, and R. Cipolla. Sparse and semi-supervised visual mapping with the s3gp. *IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 1

[18] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. *AVSS*, 2009. 2

[19] C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. 1992. 2

[20] S. -W. Shih and J. Liu. A novel approach to 3-d gaze tracking using stereo cameras. *IEEE Transactions on Systems, Man and Cybernetics, PartB*, 2004. 1

[21] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar. Gaze Locking: Passive Eye Contact Detection for Human-Object Interaction. *ACM Symposium on User Interface Software and Technology*, 2013. 6, 8

[22] Steelseries. https://steelseries.com/gaming-controllers/sentry-gaming-eye-tracker. 1

[23] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike. An incremental learning method for unconstrained gaze estimation. In *ECCV*, 2008. 1

[24] R. Valenti, N. Sebe, and T. Gevers. Combining head pose and eye location information for gaze estimation. *Image Processing, IEEE Transactions on*, 2012. 1

[25] F. Vicente, H. Zehua, X. Xuehan, F. De la Torre, Z. Wende, and D. Levi. Driver gaze tracking and eyes off the road detection system. *Intelligent Transportation Systems, IEEE Transactions on*, 2015. 1, 2, 3, 8

[26] K. Wang and Q. Ji. Real time eye gaze tracking with kinect. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016. 1

[27] K. Wang, S. Wang, and Q. Ji. Deep eye fixation map learning for calibration-free eye gaze tracking. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, 2016. 1

[28] E. Wood, T. Baltrusaitis, L. P. Morency, P. Robinson, and A. Bulling. A 3D morphable eye region model for gaze estimation. *ECCV*, 2016. 2, 8

[29] X. Xiong, Q. Cai, Z. Liu, and Z. Zhang. Eye gaze tracking using an rgbd camera: A comparison with a rgb solution. *UBICOMP*, 2014. 1, 2, 3, 7, 8

[30] X. Xiong and F. D. la Torre. Supervised descent method and its application to face alignment. *CVPR*, 2013. 6, 7

[31] 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 and applications*, 2008. 1, 2

[32] C. Yiu-Ming and P. Qinmu. Eye Gaze Tracking With a Web Camera in a Desktop Environment. *Human-Machine Systems, IEEE Transactions on*, 2015. 1

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