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 (e.g., 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. Different-ly, 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.
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].
which we can effectively recover 3D eyeball center without variations like glasses.

The 3D coordinates of landmarks are related with 2D facial landmarks through: 
\[ \lambda \mathbf{x}^e_i = \mathbf{W} \mathbf{x}^{2D}_i \]
where \( \lambda \) is the scale factor and \( \{ \mathbf{R}_e, \mathbf{o}_e \} \) is the rotation and translation. For notation convenience, we denote \( \mathbf{x}^e_i = \mathbf{W} \mathbf{x}^{2D}_i \).

In practice, the distance of facial landmarks to the camera is close to each other, therefore the scale factor \( \lambda_i \) are assumed to be the same. Eyeball center \( \mathbf{o}_e \) and \( \lambda \) can be solved by:
\[
\{ \lambda^*, \mathbf{o}_e^* \} = \arg \min_{\lambda, \mathbf{o}_e} f(\lambda, \mathbf{o}_e)
\]
\[
f(\lambda, \mathbf{o}_e) = \sum_{i=1}^{N} \| (\lambda \mathbf{x}^e_i - \mathbf{o}_e)^T (\lambda \mathbf{x}^e_i - \mathbf{o}_e) - \mathbf{x}^{2D}_i \|^2.
\]

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 \( \{ \mathbf{x}^h_i \}_{i=1}^{N} \) in HCS. The origin of HCS is typically on the face (E.g. nose tip). They first solve the head pose:
\[
\mathbf{R}_h \mathbf{o}_h^* = \arg \min_{\mathbf{R}_h, \mathbf{o}_h} \sum_{i=1}^{N} \| \lambda_i (\mathbf{x}^h_i + \mathbf{o}_h) - \mathbf{W} \mathbf{x}^{2D}_i \|^2
\]
\[s.t. \quad \mathbf{R}_h \mathbf{R}_h^T = \mathbf{I} \quad (2)\]

3D eyeball center \( \mathbf{o}_e \) can then be computed as \( \mathbf{o}_e^* = \mathbf{o}_h^* + \mathbf{R}_h^T \mathbf{o}_{he} \) (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
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 \(o_{nc}\), 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 \(o_c\), we can compute the 3D pupil center \(p\) given detected 2D pupil center \(p^{2D}\). According to camera projection model, \(p\) lies on a line \(l\) passing through camera center \(o_c\). Besides, as in Fig. 2 (b), \(p\) also lies on eyeball sphere with calibrated radius \(r_e\). By solving a line-sphere intersection problem, we can estimate the 3D pupil center \(p\) as:

\[
p^* = \begin{cases} 
  zx \\
  x^Tp - \sqrt{||x^T o_c||^2 - x^Tx(o_c - r_e^2)} 
\end{cases} \text{ if } ||z\text{x} - o_c|| > r_e \\
  x^x \text{ otherwise} 
\]  

(3)

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

Given \(o_c\) and \(p\), optical axis can be computed as: \(n_o = (p - o_c)/||p - o_c||\). As an unit length vector in \(\mathbb{R}^{3\times 1}\), \(n_o\) can also be represented as pitch angle \(\phi\) and yaw angle \(\gamma\):

\[
n_o(\phi, \gamma) = \begin{pmatrix} 
  \cos(\phi) \sin(\gamma) \\
  \sin(\phi) \\
  -\cos(\phi) \cos(\gamma) 
\end{pmatrix} 
\]

From eyeball anatomy, true gaze direction is determined by visual axis \(n_o\), therefore we need to add calibrated \(k = [\alpha, \beta]\) to optical axis: \(n_v = n_o + (\phi + \alpha, \gamma + \beta)\).

2D point of regard (PoR) can be computed by intersecting visual axis with the display plane: \(PoR = o_c + \lambda n_v\). Here we use \(o_c\) as origin of visual axis instead of \(c\). The error caused by this approximation is negligible as \(||o_c - c||\) is only a few millimeters and is much smaller than eye-plane distance. The scalar \(\lambda\) 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:

\[
PoR = f(o_c, p; \theta) 
\]

(4)

where \(\theta = [\alpha, \beta, r_e]\) 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 \(o_{ei}\), 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 remains 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 \(\{p_m, g_m\}_{m=1}^M\), we can solve 3D eyeball center and personal eye parameters as follows:

\[
o_{ei}, \theta^* = \arg\min_{o_{ei}, \theta} \sum_{m=1}^M ||f(o_{ei}, p_m; \theta) - g_m||^2 + \lambda ||d(o_{ei}, p_m) - r_e||^2 
\]

s.t. \(\theta_l \leq \theta \leq \theta_h\)  

(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 \(\lambda\) the weight to balance these two terms. \(\theta_l\) and \(\theta_h\) represents the lower and higher bounds of the eye parameters whose value can be approximated from eyeball anatomy [9]. The optimization problem is solved

<table>
<thead>
<tr>
<th>Category</th>
<th>3D Deformable Face Model</th>
<th>3D Deformable Eye-Face Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline construction requirements</td>
<td>3D facial landmarks</td>
<td>3D facial landmarks and 3D eyeball center</td>
</tr>
<tr>
<td>Offline personal calibration</td>
<td>([\theta, \tau, o_{nc}])</td>
<td>([\theta, \tau])</td>
</tr>
<tr>
<td>Individualized model</td>
<td>([x_i^{nc}]_{i=1}^N) in ECS</td>
<td>([x_i]_{i=1}^N) in ECS</td>
</tr>
<tr>
<td>Online eyeball center estimation</td>
<td>(o_i = o_i + R_i^e o_{nc}, [R_i^e, o_i]) are from solving Eq. (2).</td>
<td>(o_i^e) from solving unconstrained problem in Eq. (1)</td>
</tr>
<tr>
<td><strong>Pros</strong></td>
<td>Easy offline construction, no need of 3D eyeball center.</td>
<td>More robust against head pose variations and image noise.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gaze estimation is more accurate.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Online processing is simple and efficient.</td>
</tr>
<tr>
<td><strong>Cons</strong></td>
<td>Needs calibrating (o_{nc}) for each user.</td>
<td>Offline construction requires 3D eyeball center</td>
</tr>
<tr>
<td></td>
<td>Calibration is sensitive to initialization and noise.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not robust against head pose and image noise.</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Comparison of DFM and DEFM based eye gaze estimation.
by alternating between $o_e$ and $\theta$ with an iterative algorithm. Given estimated $o_e^*$ and $\{x_i^c\}_{i=1}^N$ in CCS, 3D eye-face model can be computed as $x_i^e = x_i^c - o_e^* \ \forall i$ 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 $x^e \in \mathbb{R}^{3N \times 1}$, then we stack all individual subject-specific 3D eye-face models from different subjects to form a large matrix $A \in \mathbb{R}^{3N \times L}$, where each column of $A$ represents one 3D eye-face model. Since different subjects have roughly similar skull anatomy and face shape, hence we believe the column space of $A$ 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 $A$. 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 $A$ to produce $q \ll L$ basis vectors $b_k \in \mathbb{R}^{3N \times 1}$. 3D eye-face model for a new subject can, therefore, be reconstructed as:

$$x^e = \sum_{k=1}^q b_k \tau_k + \bar{x} = B \tau + \bar{x} \quad (6)$$

with $B = [b_1, ..., b_q]$ the matrix of concatenated bases, $\tau = [\tau_1, ..., \tau_q]^T$ the coefficients, and $\bar{x}$ the average model (column average of $A$). Since any 3D eye-face models can be reconstructed from the bases and average model, we denote $\{B, \bar{x}\}$ 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.

Calibrating $[\tau, \theta]$ 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 $\tau$ and $\theta$, 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 $L(\lambda, o_e, \tau)$.

### Algorithm 1: Calibration Algorithm

1. **Input**:
   - 2D landmarks and pupil: $x^2D_{m}, p^2D_{m}$
   - Groundtruth PoR: $g_m$
   - Deformable eye-face model: $B, \bar{x}$

2. **Output**: $[\tau^*, \theta^*]$.

3. **Initialization**: $\tau^0 = 0; \theta^0 = \text{human average}$

4. **Sub-problem 1**: Solve $\tau$:

   $$L(\lambda, o_e, \tau) = \sum_{i=1}^N \left( \|\lambda x_i^e - o_e\|^2 - \|B(\tau + \bar{x}) + \gamma \|\tau\|^2 \right)$$

   where $B, \tau$ and $\bar{x}$ represent the corresponding rows of the $i^{th}$ facial landmark.

   **while not converge**

   **do**

   \[ \left\{ \lambda, o_e \right\}^k = \arg \min_{\lambda, o_e} L(\lambda, o_e, \tau^{k-1}) \]

   \[ \tau^k = \arg \min_{\tau} L(\lambda, o_e, \tau) + \gamma \|\tau\|^2 \]

   **end**

5. **Sub-problem 2**: Solve $\theta$:

   $$\theta^* = \arg \min_{\theta} \sum_{m=1}^M \left( \| f(o_e^*, p_m^e, \theta) - g_m \|^2 \right)$$

   \[ s.t. \quad \theta_l < \theta < \theta_u \]

   where $f(\cdot)$ is defined in Eq. (4), $o_e^*$ can be computed using Eq. (1), $p_m^e$ can be computed using Eq. (3).

Comparison with DEFM: Calibration of head-eye offset


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 poses 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 4. Geometric illustration of calibration in DFM and DEFM.](image)

\( \mathbf{o}_{hc} \) 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.

Calibration \( \mathbf{o}_{hc} \) is sensitive to initializations. Gaze estimation with DFM requires calibration of additional offset \( \mathbf{o}_{hc} \) for each user. However, the objective function is highly nonlinear w.r.t \( \mathbf{o}_{hc} \) 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 \( 40 \times \) more time.

![Figure 5. Objective contour around groundtruth offset \( \mathbf{o}_{hc} \).](image)

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.

<table>
<thead>
<tr>
<th>True ( \theta )</th>
<th>Method</th>
<th>Est ( \theta )</th>
<th>e1</th>
<th>e2</th>
<th>e3</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0</td>
<td>DFM</td>
<td>5.4</td>
<td>2.7</td>
<td>3.7</td>
<td>3.6</td>
</tr>
<tr>
<td></td>
<td>DFM</td>
<td>4.0</td>
<td>2.8</td>
<td>3.3</td>
<td>17.7</td>
</tr>
<tr>
<td>1.5</td>
<td>DFM</td>
<td>1.7</td>
<td>2.3</td>
<td>2.9</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>DFM</td>
<td>4.8</td>
<td>2.3</td>
<td>2.9</td>
<td>14.6</td>
</tr>
<tr>
<td>-1.0</td>
<td>DFM</td>
<td>0.6</td>
<td>3.6</td>
<td>4.6</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>DFM</td>
<td>2.9</td>
<td>3.1</td>
<td>3.8</td>
<td>21.3</td>
</tr>
</tbody>
</table>

![Figure 6. Gaze estimation with different head poses relative to calibration head pose.](image)

Robustness against head pose variations. Tab. 2 shows the estimated \( \theta \) and the gaze estimation performance on different poses. For DFM, the estimated \( \theta \) 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 $\theta$, it cannot give good results on a different head pose. On the contrary, DEFM gives consistent errors across head poses, as the estimated $\theta$ 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)

<table>
<thead>
<tr>
<th>Type</th>
<th>mean</th>
<th>$\pm$ std</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-translation</td>
<td>-30.0</td>
<td>$\pm$ 32.0</td>
</tr>
<tr>
<td>y-translation</td>
<td>-11.0</td>
<td>$\pm$ 27.0</td>
</tr>
<tr>
<td>z-translation</td>
<td>525.0</td>
<td>$\pm$ 39.0</td>
</tr>
<tr>
<td>yaw (left or right)</td>
<td>3.0</td>
<td>$\pm$ 19.5</td>
</tr>
<tr>
<td>pitch (up or down)</td>
<td>2.8</td>
<td>$\pm$ 12.7</td>
</tr>
<tr>
<td>roll (in plane rotation)</td>
<td>-82.0</td>
<td>$\pm$ 7.1</td>
</tr>
</tbody>
</table>

**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 $9 \times 5$ heatmap is resized to screen size ($1920 \times 1080$) 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.

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

<table>
<thead>
<tr>
<th></th>
<th>Manual</th>
<th>Method in [29]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error / degree</td>
<td>2.5</td>
<td>3.1</td>
<td>3.5</td>
</tr>
<tr>
<td>Time / ms</td>
<td>4000.0</td>
<td>30.0</td>
<td>2.3</td>
</tr>
</tbody>
</table>

**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 preserving good accuracy.

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

<table>
<thead>
<tr>
<th></th>
<th>landmark detection/tracking</th>
<th>pupil detection</th>
<th>gaze estimation</th>
<th>misc</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time / ms</td>
<td>25.7</td>
<td>2.3</td>
<td>0.3</td>
<td>5.0</td>
<td>33.3</td>
</tr>
</tbody>
</table>

**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,
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 (E.g. {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.

<table>
<thead>
<tr>
<th>Index</th>
<th>Configurations</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1, 2, 7−11}</td>
<td>4.1</td>
</tr>
<tr>
<td>2</td>
<td>{1, 4−7, 11}</td>
<td>3.5</td>
</tr>
<tr>
<td>3</td>
<td>{1, 2, 5, 7−11}</td>
<td>3.7</td>
</tr>
<tr>
<td>4</td>
<td>{1−11}</td>
<td>3.9</td>
</tr>
<tr>
<td>5</td>
<td>{3−11}</td>
<td>4.8</td>
</tr>
</tbody>
</table>

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 DFM 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 (τ = 0) gives larger error but is most suitable for applications without personal calibration.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>[2]</th>
<th>[29]</th>
<th>[25]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error/degree</td>
<td>7.3</td>
<td>5.7</td>
<td>7.2</td>
<td>3.5</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>[28]</th>
<th>[29]</th>
<th>[25]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>ColumbiaGaze</td>
<td>8.9</td>
<td>12.3</td>
<td>9.7</td>
<td>10.2</td>
</tr>
<tr>
<td>EyeDiap (V, S)</td>
<td>21.5</td>
<td>32.3</td>
<td>21.3</td>
<td>22.2</td>
</tr>
<tr>
<td>EyeDiap (H, M)</td>
<td>22.2</td>
<td>35.7</td>
<td>25.2</td>
<td>28.3</td>
</tr>
</tbody>
</table>

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 (~ 13 × 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

[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
[31] X. Xiong and F. D. la Torre. Supervised descent method and its application to face alignment. CVPR, 2013. 6, 7
[34] 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