Real-Time 3D Eye Performance Reconstruction for RGBD Cameras

Quan Wen, Feng Xu, and Jun-Hai Yong

Abstract—This paper proposes a real-time method for 3D eye performance reconstruction using a single RGBD sensor. Combined with facial surface tracking, our method generates more pleasing facial performance with vivid eye motions. In our method, a novel scheme is proposed to estimate eyeball motions by minimizing the differences between a rendered eyeball and the recorded image. Our method considers and handles different appearances of human irises, lighting variations and highlights on images via the proposed eyeball model and the $L0$-based optimization. Robustness and real-time optimization are achieved through the novel 3D Taylor expansion-based linearization. Furthermore, we propose an online bidirectional regression method to handle occlusions and other tracking failures on either of the two eyes from the information of the opposite eye. Experiments demonstrate that our technique achieves robust and accurate eye performance reconstruction for different iris appearances, with various head/face/eye motions, and under different lighting conditions.

Index Terms—Eye reconstruction, facial animation, gaze tracking, multilinear model, RGBD camera

1 Introduction

Gaze information is crucial in many applications in areas ranging from computer vision and graphics to HCI and virtual reality. For example, in immersive virtuality telepresence (IVT), the user experience is directly affected by the gaze of a virtual character that the user is communicating with [1]. In augmented reality (AR), the gaze position of a user can be used to adjust the content of the display to achieve better realism [2]. In computer vision, gaze estimation aims to estimate the on-screen gaze point (or other similar representations) with RGB or RGBD input [3], [4], [5]. In computer graphics, generating realistic humans, particularly human faces, has always been an important and challenging research goal. Because eyes are key organs on the face, which vividly deliver the emotion of a character, the reconstruction of eyes is crucial for face capture and animation. Regarding animation, speech has been used to drive eye motions [6], [7], and eye motions have been used to drive eye blinks [8]. With respect to capture, some prior works have focused on static eye model reconstruction [9], [10], [11], in which high-quality eye models are reconstructed and can be used in photo-realistic facial animation. However, for dynamic facial performance, most works only focus on the skin surface with different levels of details [12], [13], [14]. However, few works focus on eye performance (including user-dependent iris size and motion-dependent eyeball rotation).

In this paper, we propose a novel real-time solution for eye performance reconstruction. The reconstructed eye performance, combined with facial tracking results, dramatically improves the visual realism of the final results. Our system input is a single-view RGBD stream recorded by consumer depth sensors. In addition to reconstructing and tracking facial surfaces from input streams, we propose a method to reconstruct the geometry and texture models of the two eyes of the user and reconstruct their motions in the entire sequence. Our system does not require high-quality facial video input or any pre-training stages, and it is fully automatic in real time. Furthermore, by exploring the correlation between the two eyes, our system is feasible when occlusions occur in eye regions, which greatly improves the robustness of the facial reconstruction system.

Very recently, [15] proposed the first real-time 3D eye gaze capture method that combines eye capture with facial tracking. We achieve the same goal, but our method differs in several aspects. First, [15] proposed a two-step method that first tracks the eye gaze point on 2D images and then estimates the 3D orientation, whereas we propose a unified framework that directly estimates 3D eye motion from 2D images. Second, we propose an online bi-directional regression method to handle occlusions and tracking failures in eye regions, which is mentioned as a limitation in [15]. Third, to manage lighting changes, highlights on images and different appearances on human irises, training samples that cover all these variations are required by [15], which is tedious work for data collection and labeling. In contrast, we propose a time-varying appearance model and an $L0$-based optimization to explicitly handle lighting changes, highlights and iris appearance variations. Note that [15] takes RGB data as input, whereas we still require depth information. This is majorly due to the facial tracking method that we depend on. We may integrate video-based

• *The authors are with the School of Software, Tsinghua University, Beijing 100084, P. R. China, the Key Laboratory for Information System Security, Ministry of Education of China, Beijing 100084, P. R. China, and Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, P. R. China.*
E-mail: {wenq013, xufeng2003}@gmail.com, yongjh@tsinghua.edu.cn.

Manuscript received 27 May 2016; revised 6 Dec. 2016; accepted 12 Dec. 2016. Date of publication 19 Dec. 2016; date of current version 27 Oct. 2017. Recommended for acceptance by Y. Yu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2016.2641442

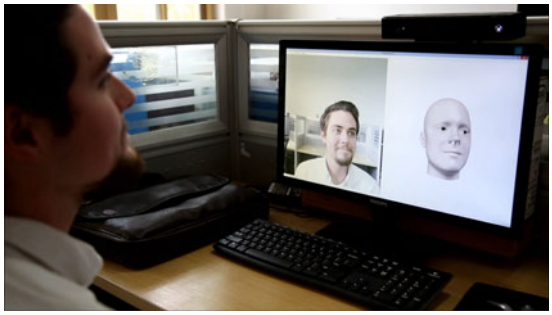


Fig. 1. Our real-time eye performance reconstruction system. A new Kinect sensor records depth and color videos of a user, and our system reconstructs both facial surface and eye motion in real time.

face tracking methods into our system, but more investigations are required to determine whether similar performance can be achieved based on RGB input.

Based on the concept of analysis-by-synthesis, we require an eyeball model to render a consistent appearance with real recorded images and an effective and robust optimization scheme to estimate 3D eyeball motions in real time. In this paper, we propose a simplified geometry and appearance eyeball model (SGAEM) to represent different types of eyeball variations. In terms of geometry, our model has two parameters: the eyeball radius and the iris radius. With respect to appearance, we use two single colors for the iris and sclera. To further model the appearance changes caused by lighting variations, we propose time-varying colors, which are updated online by our method. As real recorded images are of low quality, delicate physical eyeball models are not required, and this simplified model works robustly in practice. To efficiently estimate 3D eyeball motions with this model, we propose a 3D Taylor expansion to linearize the complex optimization and use linear optimization to fit the rendered eyes to the recorded ones. Furthermore, to manage the highlights with various shapes and positions in eye regions, caused by the high specularity of the cornea, we propose an $L0$-based penalty term to remove the influence of the highlights in the optimization, which leads to a more robust eyeball tracking.

For robust eyeball reconstruction, another key challenge is how to handle occlusions and other types of tracking failures. In face tracking, [16] and [17] used a segmentation technique and speech signal to manage occlusions, respectively. To solve this problem for eye performance reconstruction, our key observation is that the motions of the two eyes, including the eyeball rotations and the eyelid regions, are highly correlated. Thus, we propose a bidirectional regression model to explore the correlation. Specifically, we first develop a failure case detector to detect whether occlusion or other tracking failures occur on either of the two eyes on the fly. If neither of the two eyes fail in tracking, both the left-to-right and the right-to-left linear regressors receive a training sample represented by shape parameters of the eye regions plus eyeball orientation parameters. With more training samples involved on the fly, the two regressors are continuously updated. When one of the two eyes is detected as failing in tracking, the corresponding regressor is applied to predict the eyeball orientation and the shape parameters of the eye region from the information of the opposite eye. Because the training and testing samples of the regressors are always of the same user, our method does not need to consider the model variations among users.

Here, we summarize the key contributions of our technique:

- A novel framework that combines real-time facial tracking with eye performance reconstruction. Considering the very recent work [15], our work achieves real-time eye tracking without the requirement of pre-training stages and handles partial eye occlusions and tracking failures.
- A simplified geometry and appearance eyeball model, plus a 3D Taylor expansion-based optimization scheme, achieves efficient and robust eye performance reconstruction. The model and the optimization method handle the tracking with low-quality video input, user- and lighting-dependent appearance variations and highlights in general environments.
- A bidirectional regression model that predicts the motions of eyes when occlusions or tracking failures occur. The regressor is trained online with tracked samples; thus, it does not rely on pre-training stages and always fits the specific user. Moreover, the bidirectional manner handles occlusions and other tracking failures of either of the two eyes.

2 Related Work

Our work aims to reconstruct 3D eye performance from RGBD video input. In the literature, there are two topics that are closely related to our work: gaze estimation and eye center localization, which extract 2D or 3D gaze information from single or multiple videos.

Gaze estimation aims to estimate the on-screen gaze point from facial videos. By correctly detecting the gaze point, human-computer interaction can be achieved without additional equipment such as a mouse or keyboard. Gaze estimation is categorized into model-based methods and appearance-based methods [18]. The model-based methods utilize 3D eyeball models to estimate the gaze point or gaze direction. Some works in this category use corneal reflection to detect eye position on video frames and further estimate the gaze point on-screen or in the scene. Based on the theory presented in [19], a single camera and single IR light source can only achieve gaze estimation with stationary user head poses. To allow free head poses, systems with multiple IR light sources or/and multiple cameras are proposed [20], [21], [22], [23], [24], [25]. Furthermore, some work [26] focuses on simplifying the user-specific calibration step, which is required in this type of technique, and [27] performs the calibration online; thus, it does not rely on an explicit calibration step. However, the requirement of specific hardware (multiple IR light sources and multiple cameras) makes using this technique difficult in many daily applications.

In model-based methods, some other works use a single camera without IR lighting to perform gaze estimation. These techniques rely on the detection of the pupil center and iris edges on recorded video frames [28], [29], [30]. In these techniques, another important issue is estimating the 3D eyeball position (also called eyeball pose). Then, combined with the 2D location of the pupil center, gaze information can be

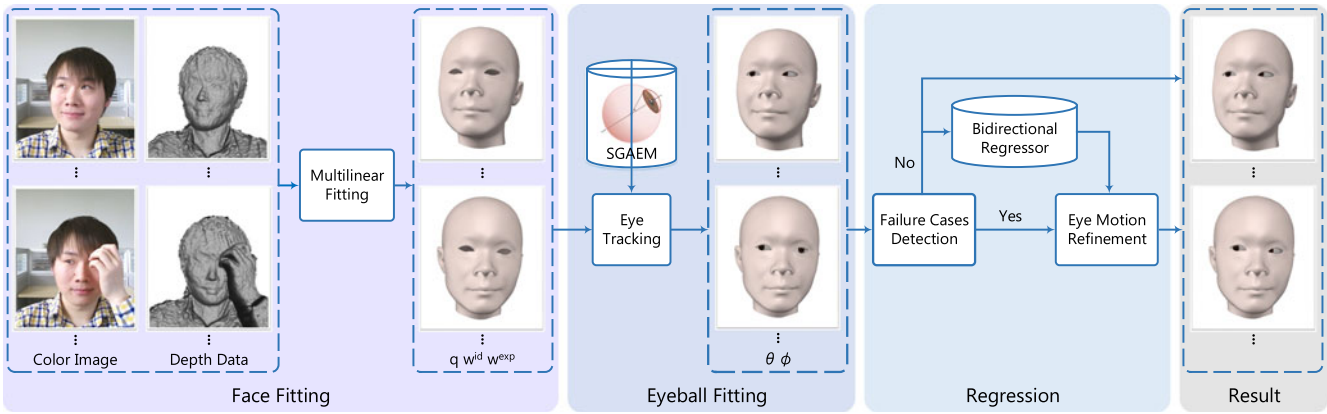


Fig. 2. An overview of our system. Face fitting is first performed to achieve traditional face tracking in our system. Then, we perform our eyeball fitting to obtain the eyeball performance of the user. Since tracking failures may exist on either of the eyes, we propose a regression scheme that is trained and used online to improve the tracking results.

extracted. Ref. [4] estimated 3D head pose to define the eyeball pose. Ref. [31] solved for the eyeball pose by a calibration stage, whereas [32] used online calibration to improve person-specific eye parameters. Ref. [33] combined the pose estimator and the eye locator together to pursue a better result.

Appearance-based methods directly estimate the on-screen gaze points or camera space 2D gaze angles from input images. Early works in this category also assumed fixed head pose of users [34], [35], [36]. Later, with the estimated head poses as input, [37], [38], [39] achieved the handling of arbitrary head poses in gaze estimation. Ref. [40] did not explicitly solve head poses but used a pose bias to model the influence of arbitrary head poses, and this bias was compensated by either a learning-based regression or a geometric-based method. Learning the mapping directly from images to gaze parameters suffers from a generalization problem. To make it robust to handle variations on eye shapes, poses and illuminations, a large amount of training data is required [34], [41], [42]. Recent works have recorded and labeled a large database for this purpose [5] or synthesized virtual data [3], [43]. Most recently, [44] considered a new problem: gaze estimation for a user wearing a head mounted display (HMD).

Gaze estimation can be further used to estimate 3D eyeball rotations with additional 3D position information of the eyeball. Specifically, given the 3D position of the eyeball and the gaze point on a 2D image, the line of sight can be estimated, and the 3D eyeball rotation is determined accordingly.

Pupil center localization aims to locate the pupil center or iris boundary on 2D video frames. A survey of this topic can be found in [45]. It can be used for many applications, such as face identification, gaze estimation, eye performance reconstruction, and so forth. Pupil center localization, combined with head pose estimation, provides an effective approach to achieve gaze estimation. Ref. [46] located the pupil center positions for faces with frontal pose and upright orientation. Combined with a head pose estimation and an image warping technique, faces with various head poses are warped to the frontal pose to perform the pupil center localization. Similarly, [4] used RGBD data to estimate 3D head poses; moreover, a 3D textured face model was reconstructed and a frontal face image was obtained by model re-rendering. Ref. [33] jointly optimized head pose estimation and eyeball localization based on the observation that the solving of either of the two tasks can help the solving of the other. Additionally, [47] and [48] detected iris boundaries, which imply the approximate pupil centers.

There are many other types of ways to achieve pupil center localization. Ref. [49] predicted pupil centers by the intersections of gradient vectors. Ref. [50] used the symmetry on brightness to detect the locations of pupil centers without requiring any prior knowledge. Ref. [51] also used the symmetry but achieved better results by using isophotes to infer the centers of circular patterns. Another type of method extracts various image features and uses training data to learn the correlation between pupil center locations and the feature responses. Ref. [52] used edge features and PCA to learn the distribution of the feature responses on the pupil center regions. Ref. [53] used Harr features and SVM (support vector machine) to learn a classifier from training samples. Ref. [54] also followed this mechanism but used AAM (active appearance model) to further refine the results. [55] extracted facial features to achieve pupil center localization from high-resolution images with frontal faces. Ref. [56] used multi-scale Gabor features to learn the feature distribution for pupil centers. In addition to these techniques, there are also some learning-based methods that do not require explicit feature extraction. Ref. [57] used both positive and negative samples to train two cascaded regressors. Ref. [58] only used positive samples to train one cascaded regressor where features, based on the color difference of pixels, are automatically extracted from the training procedure to maximize the classification ability. In our paper, we use this method as an initialization and compare it with our final result.

3 Overview

The overview of our system is shown in Fig. 2. Our system input contains color and depth video streams recorded by a consumer depth sensor, such as the new Kinect or the Xtion sensor. The two streams are synchronized and calibrated by default. First, we use the multilinear model generated by [59] to perform real-time facial tracking using the method in [17]. With the obtained 3D face shape, our first core stage, the eye performance reconstruction, estimates the parameters of our SGAEM and the motions of the two eyeballs. Specifically, the eyeball radius and iris radius in SGAEM are estimated in the first frame of a sequence because they never change for a particular user. The color

model for the iris and sclera is updated according to each input color image to compensate for lighting changes. The eyeball rotation is then estimated by minimizing the differences between the rendered SGAEM on a 2D image plane and the real captured color image. Because this provides a nonlinear optimization problem, we propose a 3D Taylor expansion scheme to linearize the optimization, which outperforms traditional Taylor expansions on 2D images.

Our second core stage is the bidirectional eye regression that estimates the motion of one eye from the opposite one. A failure case detector is first utilized to detect whether occlusions or other tracking failures occur on the two eye regions. If there are no failure cases, the eye motion is used as a training sample for our bidirectional eye regressor, which contains two linear regressors that map the motion of one eye to the other. With an increasing number of training samples involved, the regressor generates more reliable results in eye motion prediction. If either of the two eyes is detected as occluded or not faithfully reconstructed (caused by fast motion, extreme eye pose, and so forth), the bidirectional eye regressor is applied to refine the eyeball rotation and the eyelid deformation. Since the regressor is trained with online data of the same user, it provides robust results for regular eye motions. Of course, if both of the eyes fail in tracking, we smoothly interpolate the eye motions of the previous frames with the rest eye pose to generate a plausible result.

4 Face Tracking

Before reconstructing eye performance, our system tracks the facial surface with RGBD input in real time. In recent years, many techniques have been proposed to achieve real-time facial surface tracking. Because this task is not the contribution of this paper, we simply implement a tracking method based on a multilinear face model. Note that there are many other existing works [12], [60], [61] that could be used for this task.

4.1 Multilinear Model for Face Tracking

First, we construct a multilinear model based on the method introduced in [59]. The multilinear model represents a 3D face $\mathbf{F}$ as a set of identity weights $\mathbf{w}^{id}$ and a set of expression weights $\mathbf{w}^{exp}$, denoted as:

$$\mathbf{F} = R\left(C_r \times_2 \mathbf{w}^{id} \times_3 \mathbf{w}^{exp}\right) + T, \tag{1}$$

where C_r is called the reduced core tensor, the key component of the multilinear model, which is obtained from the training face meshes with different identities and different expressions. $\mathbf{q} = [R, T]$ contains the global motion parameters. When performing face tracking, we first run identity fitting, where users are asked to be in a neutral expression. Thus, $\mathbf{w}^{exp}$ is fixed to neutral, and 3D head pose $\mathbf{q}$ and $\mathbf{w}^{id}$ are estimated iteratively by minimizing the per-vertex distances between the multilinear model and the input depth. The correspondences are obtained by projective iterative closest point (ICP) [62]. After the identity fitting on the first frame, expression fitting is performed on each of the following frames in real time, i.e., $\mathbf{w}^{id}$ is fixed and 3D head pose $\mathbf{q}_t$ and $\mathbf{w}_t^{exp}$ are estimated iteratively. Note that in the ICP-based fitting stages, we use the automatically detected facial landmarks [58] and their pre-defined mesh vertices to build the initial correspondences.

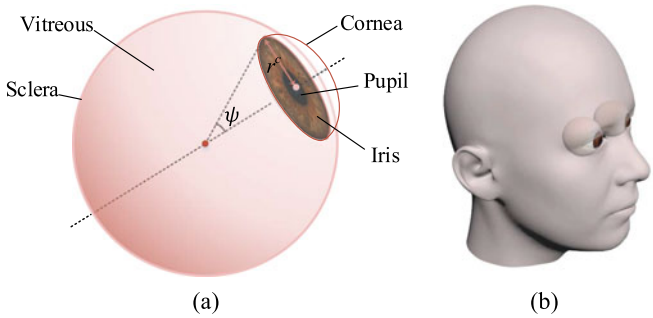


Fig. 3. Eye model used in our system (a) and its relative position and orientation to the head model (b). Note that the appearance of this model is only for indicating the different parts of an eyeball. The exact appearance model used in our method is a different one.

4.2 Occlusion Detection for Face Tracking

To achieve face tracking with partial occlusions, we integrate the method presented in [16] to detect occlusions. In the ICP-based face tracking, a distance threshold $\sigma = 5mm$ is set to reject the vertices located far from their corresponding depth points. We also involve the penetration test to utilize the assumption that the occluding objects are always located in front of the face. After excluding the occlusions, face tracking is robustly performed. Details can be found in [16].

5 Eye Performance Reconstruction

In this section, we discuss how to reconstruct eye performance on top of the facial skin reconstruction. To achieve this goal, we first propose our simplified geometry and appearance eyeball model (SGAEM) that can generate consistent visual appearances with input images recorded by consumer sensors. Because SGAEM relates 3D eyeball with 2D image appearance, it is used to estimate eyeball motions from images. In the following, we first describe the representation of SGAEM, and then we introduce how to use it to estimate eyeball motions.

5.1 SGAEM

The key idea in designing SGAEM is to use a simplified eye model and fewer parameters to generate visual appearances that are consistent with eye images recorded by consumer sensors.In the following, we discuss the representations of both the eyeball geometry and appearance in SGAEM.

In terms of *geometry*, similar to [31], we use one large sphere to represent the vitreous and a small sphere, representing the cornea, to define the iris (shown in Fig. 3a). In SGAEM, it is assumed that the position and size of the eyeball (vitreous and iris) are fixed relative to the head pose and size (shown in Fig. 3b). Specifically, an artist first manually fixes the position of the head model and then iteratively tunes the position of the first eyeball and changes the size of the eyeball 3$\sim$5 times. For the second eyeball, the size is set the same as the first one, and only its position is tuned manually. Because the depth of the eyeball is not easy to determine, the artist achieves this by setting the eyeball as close to the face as possible, while keeping it not intersecting with the eyelid in any pose. Then, after the face tracking described in the previous section, the global scaling and transformation are directly applied to the

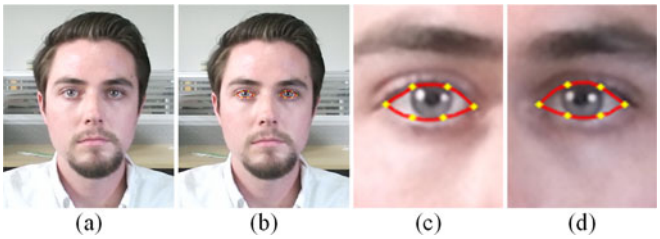


Fig. 4. Eye regions on color image. (a) input image; (b) extracted eye regions denoted by the red contours; (c,d) zooming in on the left and right eye regions.

vitreous. Note that even though this assumption is not exactly satisfied, it works well in all our experiments, as the true variations do not greatly affect the visual appearance. However, the iris size, which determines the size of the dark region of the eyes, affects the visual appearance and considerably varies for different persons. Therefore, SGAEM treats the iris size as a user-dependent parameter r^c, which is estimated in the following section. In practice, as shown in Fig. 3a, r^c is replaced by the angular parameter ψ, which is mathematically more straightforward to be estimated in the following optimization. Biologically, there are many other user-dependent eye features, such as pupil size, cornea shape, and so forth. However, they are difficult to measure from low-quality input videos; thus, they are not considered in our model.

Regarding *appearance*, we use a single intensity i^v to represent the sclera and another i^i to represent the iris, with boundary regions blended by i^v and i^i. We know that the sclera is generally white with red capillaries, whereas the iris is dark, on the intensity map, with wrinkle-like textures. However, for consumer sensors in everyday lighting conditions, these detailed textures cannot be faithfully recorded. Consequently, an appearance model with two intensities fits the recorded image and is more robust for motion estimation from the image. In SGAEM, the values of i^v and i^i vary frame by frame because head motions and lighting changes may cause appearance variations. Ideally, the parameters of eye appearance should be estimated by separating illumination from the input images. Nevertheless, by using this time-varying appearance model, the problem of lighting changes is implicitly considered. Note that we do not have an additional pupil intensity in SGAEM because the pupil is small and thus sometimes difficult to measure from low-quality images, and the iris and sclera intensity model is sufficient to estimate eyeball rotations. One drawback is that the pupil size cannot be reconstructed in our method.

SGAEM is a highly simplified and not accurate eyeball model, but it is only used for the following eyeball motion estimation and generates good results very efficiently in all our experiments. Some recent techniques achieve delicate eyeball model reconstruction [63], [64], which may also be applicable for our system and may generate even better results. At the same time, however, the eyeball reconstruction and the following motion estimation steps may become more complex due to the higher complexity of the eyeball model.

5.2 Optimization by 3D Taylor Expansion

In this section, we discuss how to estimate the parameters $\{\psi, i^v, i^i\}$ of SGAEM and the eyeball rotation, defined by $\{\theta, \phi\}$, for each input color image. In our system, the appearance parameters $\{i^v, i^i\}$ are first determined and then used in the estimation of ψ and $\{\theta, \phi\}$.

5.2.1 Appearance Estimation

For an input image, the eye region R (as shown in Fig. 4) is detected by the facial landmarks [58] used in the facial tracking method. Then, the pixel intensities in the eye regions are used to estimate $\{i^v, i^i\}$ for the two eyes. Because the two eyes follow exactly the same processing steps, there are no extra labels to indicate the left and the right eyes.

Generally, pixels of the sclera are bright, whereas pixels of the iris are dark. Thus, the distribution of the pixel intensities in R should satisfy a Gaussian mixture model with two Gaussians, that is,

$$\sum_{k=1}^{K} \mathcal{N}(\mu_k, \Sigma_k), \tag{2}$$

where $K = 2$ and the two μ_k represent i^v and i^i. However, because the cornea has strong reflectance, there might be highlights in the eye regions, which makes a set of pixels even brighter than the sclera. In addition, for some human races, the iris may be noticeably brighter than the pupil, which also gives a new Gaussian of the pupil in the distribution. In these cases, more Gaussians may fit the data better. Based on these observations, in practice, we use a Gaussian mixture model with $K = 2\sim4$ to fit the intensity distribution in R. An additional Gaussian is involved when it achieves more than κ percent ($\kappa = 15$ in all our experiments) fitting error dropping. To determine which Gaussian represents the sclera or iris, we first classify the pixels into K categories corresponding to the K Gaussians, depending on which Gaussian provides the highest possibility to a pixel. Subsequently, the two "largest" Gaussians with the most pixels are regarded as the sclera and iris because the pupil and highlight regions are typically smaller in size than the sclera and iris. Although there exist some frames that are fitted by a greater number of Gaussians in practice, our system still works well because the values of i^v and i^i are not affected too much by "small" Gaussians. Examples for appearance estimation are shown in Fig. 5.

5.2.2 Geometry and Rotation Estimation

Here, we discuss the estimation of ψ and $\{\theta, \phi\}$ from images. The core idea is to solve an optimization problem that minimizes the differences between the rendered SGAEM from the camera viewpoint and the real recorded image I, which can be illustrated as:

$$E(\psi, \theta, \phi) = \sum_{j \in S} \left\| C_j(\psi) - I_j(\theta, \phi) \right\|_2^2. \tag{3}$$

Here, S represents all visible vertices on SGAEM and C_j is the intensity of vertex j determine by whether it is on the iris. Because $\{i^v, i^i\}$ is determined previously, C_j only relies on ψ, which determines the iris size. I_j is the intensity of the corresponding pixel of vertex j. Because the camera projection matrix is fixed and the position and size of the vitreous are pre-determined, only the eyeball rotation $\{\theta, \phi\}$ influences the labels of the corresponding pixels.

When solving the optimization problem in Eqn. (3), we first focus on the case where ψ is fixed. In this case, with the

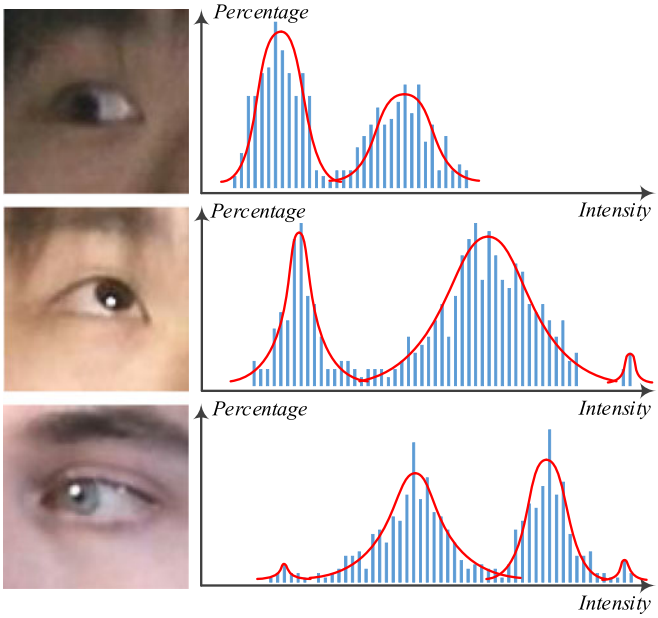


Fig. 5. Appearance estimation using Gaussian mixture model. Fitting intensity distribution with two Gaussians (top), three Gaussians (middle) and four Gaussians (bottom).

full knowledge of SGAEM, the problem becomes fitting a renderable 3D model to an input image. Inspired by [65], we linearize $I_j(\theta,\phi)$ by a first-order Taylor series expansion:

$$E(\theta,\phi)=\sum_{j\in S}\left\|C_j-I_j(\theta,\phi)\right\|_2^2\approx \sum_{j\in S}\left\|C_j-(I_j(\theta^0,\phi^0)+\frac{\partial I_j}{\partial x}\delta_x(\delta_\theta,\delta_\phi)+\frac{\partial I_j}{\partial y}\delta_y(\delta_\theta,\delta_\phi))\right\|_2^2, \tag{4}$$

where θ^0 and ϕ^0 represent the current eyeball rotation. x and y are the two dimensions of the image plane. δ_θ and δ_ϕ are the unknowns to be solved. This linearization converts the minimization of Eqn. (3) into a linear system. Solving the linear system and updating $\{\theta^0,\phi^0\}$ with $\{\delta_\theta,\delta_\phi\}$ are iteratively performed until they converge to the final solution $\{\theta,\phi\}$.

However, this image-based linearization does not provide good results in real cases. From Eqn. (4), note that each pixel, with nonzero gradients, has its contribution in estimating δ_θ and δ_ϕ. For pixels inside the iris or sclera, their gradients are majorly caused by lighting variations or noise, which are not modeled in SGAEM and thus will not help to estimate eyeball rotations. However, for pixels near the boundary of the iris and sclera, their gradients are robust to lighting variations and noise because the iris and sclera have significant appearance differences. Consequently, these gradients are useful in estimating eyeball rotations. As both types of pixels are involved, the image-based linearization is not that robust to lighting variations and noise in practice.

Based on this observation, rather than using all pixels with nonzero gradients in the 2D expansion-based method, we propose a novel method based on 3D expansion, which involves only the pixels near the boundary of the iris and sclera to estimate eyeball rotations. The 3D expansion-based method also minimizes the intensity differences between the rendered SGAEM and the recorded image. Differently,

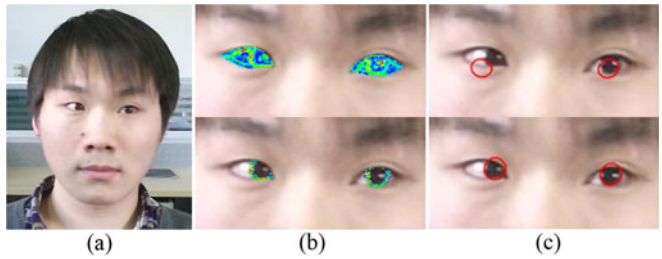


Fig. 6. 2D (top) and 3D (bottom) Taylor expansion methods. (a) input image; (b) pixels used in the optimization (color indicates the intensity of image gradient); (c) final tracked iris boundary.

it counts pixels on the 2D image rather than vertices on the 3D mesh, that is:

$$E(\theta,\phi)=\sum_{k\in T}\|C_k(\theta,\phi)-I_k\|_2^2\approx \sum_{k\in T}\left\|\left(C_k(\theta^0,\phi^0)+\frac{\partial C_k}{\partial\theta}\delta_\theta+\frac{\partial C_k}{\partial\phi}\delta_\phi\right)-I_k\right\|_2^2. \tag{5}$$

Here, T contains all pixels in the eye regions of image I. I_k is the intensity of pixel k. C_k is the intensity of the corresponding vertex of pixel k, which is determined by the motion of the eyeball $\{\theta,\phi\}$. In this formulation, the vertex intensity C_k is related to the unknowns $\{\theta,\phi\}$; thus, the Taylor expansion should be performed on the eye surface in the 3D domain. Note that the sampling patterns of the 2D and 3D expansion-based methods are different, which are chosen for convenience but do not affect the final result. Switching to the opposite sampling pattern generates a similar result.

For vertices inside the iris or sclera, because their surrounding vertices share the same intensity given by SGAEM, their intensity gradients along θ and ϕ should be zero. According to Eqn. (5), these vertices have no contributions in estimating δ_θ and δ_ϕ. Consequently, only vertices around the boundary of the iris and sclera are considered in the estimation. Correspondingly, only pixels near the boundary are used in the optimization because the initial eyeball pose is not too far from the ground truth. Compared with the 2D expansion-based method, this expansion form removes the pixels without robust gradients, thereby helping the iterative method converge to the global optimum rapidly and accurately. Fig. 6 shows the pixels used in the optimizations and the final results of both the 2D and 3D Taylor expansion-based schemes. As shown, the latter method does not use the pixels with noisy gradients and achieves a better result. Details about solving Eqn. (5) are described in the Appendix.

Now we return to the original energy where ψ is in the unknowns. Following Eqn. (5), its formulation is:

$$E(\psi,\theta,\phi)=\sum_{k\in T}\|C_k(\psi,\theta,\phi)-I_k\|_2^2. \tag{6}$$

To minimize this energy, we propose a two-step iteration scheme. In the first step, θ and ϕ are assumed to be fixed. In this case, the energy and its linearization are as follows:

$$E(\psi)=\sum_{k\in T}\|C_k(\psi)-I_k\|_2^2\approx \sum_{k\in T}\left\|\left(C_k(\psi^0)+\frac{\partial C_k}{\partial\psi}\delta_\psi\right)-I_k\right\|_2^2. \tag{7}$$

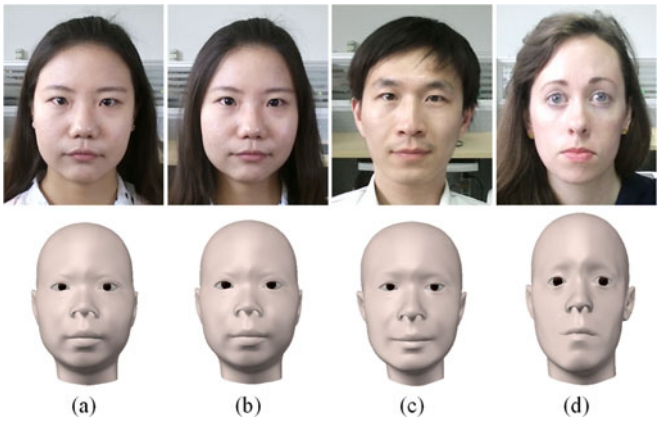


Fig. 7. Iris size estimation. (a,b) the same character without ($\psi = 0.33$) and with ($\psi = 0.39$) contact lens; (c,d) two characters with noticeably different iris sizes (left: $\psi = 0.33$; right: $\psi = 0.39$). Top: input images. Bottom: results of iris size estimation.

The full linearization and solving of Eqn. (7) are shown in the Appendix, and the results of this step are presented in Fig. 7. We observe a difference in iris size for the same character with and without contact lens and for different characters with noticeably different iris sizes. In the second step, we assume that ψ is fixed and only update $\{\theta, \phi\}$ by minimizing the energy in Eqn. (5).

In practice, Eqn. (6) is only solved in the first frame of an input sequence. For the following frames, because the iris size does not change with time, the estimated ψ in the first frame is directly used, and only the minimization of Eqn. (5) is performed. Our iterative optimization requires initial values of the unknowns. ψ is set to 0.31 in radians, which is manually given by the artist. For the eyeball rotation $\{\theta, \phi\}$, we use the result of the previous frames and the information from pupil landmarks [58] for the first frame. Specifically, a pupil landmark defines a line in the 3D world space. θ and ϕ of the corresponding eye are calculated from the intersection point of this line and the eyeball.

5.2.3 Highlight Handling

As formulated in Section 5.2.1, the cornea has strong reflectance. Therefore, images may contain highlights in the eye regions (shown in Fig. 8), which is not modeled by SGAEM and thus may cause problems in the aforementioned optimization. To solve this problem, we observe that highlights generally appear in a small region compared with the full eye region. In this case, it causes sparse but large energy penalties in the optimization. Based on this observation, an $L0$-based regularization term is involved in the optimization to remove these sparse errors. Specifically, the energy in Eqn. (6) is modified as follows:

$$E(\psi, \theta, \phi, H) = \sum_{k \in T} \|I_k - C_k(\psi, \theta, \phi) - H_k\|_2^2 + \lambda \|H\|_0. \quad (8)$$

Here, H is a vector whose dimension is equal to the number of pixels in T and H_k is the kth element corresponding to vertex k. H represents the sparse errors caused by the highlight. λ is set to 0.22 after normalizing the intensity values to $[0, 1]$. The first term in Eqn. (8) measures the fitting error, where the large errors caused by the highlight are eliminated by H, whereas the second term controls the sparsity of H to avoid degeneration.

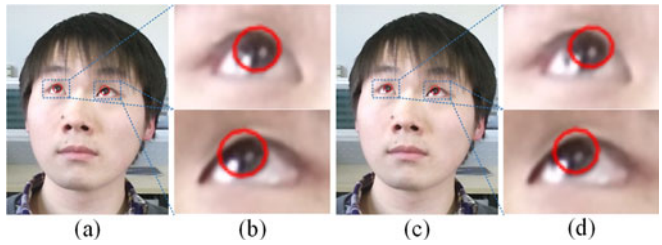


Fig. 8. Reconstructed iris boundary on input image. Left: with highlight handling. Right: without highlight handling.

Inspired by the method in [66], a two-step iteration is used to minimize Eqn. (8). In the first step, we fix $\{\psi, \theta, \phi\}$ to calculate H with a closed-form solution:

$$H_k = \begin{cases} 0 & \text{if } I_k - C_k(\psi, \theta, \phi) < \sqrt{\lambda} \\ I_k - C_k(\psi, \theta, \phi) & \text{if } I_k - C_k(\psi, \theta, \phi) \geq \sqrt{\lambda} \end{cases} \quad (9)$$

Because highlight pixels generally have very high intensities while true iris has very low intensity, we are able to set an appropriate λ to detect highlights appearing on the iris. Meanwhile, because the intensity of the sclera is typically lower than that of the highlight, the desired penalties of iris mismatching to sclera pixels are still preserved.

In the second step, $\{\psi, \theta, \phi\}$ is estimated while H is fixed. This returns to the minimization of Eqn. (6) with the only difference that T does not contain pixels with nonzero H_k. Note that for the first frame in a sequence, this two-step iteration constructs the outer iteration, while the iteration solving for ψ and $\{\theta, \phi\}$, described in Section 5.2.2, is the inner iteration. For the following frames, only the outer iteration is performed because ψ is fixed. The effectiveness of the highlight handling is demonstrated in Fig. 8. With this $L0$-based regularization, our method achieves more accurate eyeball tracking when a highlight appears in the input image.

6 Bidirectional Eyeball Regression

The method in Section 5.2 provides promising results when both eyes are clearly observed by video cameras. However, for ordinary face videos, occlusions and tracking failures may always occur. Hsieh et al. [16] utilized a segmentation technique to achieve robust facial tracking with partial occlusions, but the true motion of the occluded region cannot be reconstructed. Meanwhile, the motions of the two eyes are highly correlated, including the eyeball rotations and the surface deformations around the eye regions, e.g., the eyelids. Based on this observation, we propose a bidirectional online regression method that estimates the motion of one eye, when it is occluded or fails in tracking, from the opposite one. To avoid the pre-training step, we propose an online training strategy that integrates user's tracked data to train a regressor on the fly, and then we apply it for later input with occlusions or tracking failures. Because the regressor is trained and used for one specific user, user-dependent adaption is not required. In the following, we discuss our failure case detector, which detects both occlusions and tracking failures. Then, we introduce the online training scheme that trains the regressor with the tracked samples and the online regression scheme that refines the

samples with failure of one eye. Note that our system cannot refine samples with failures of both eyes.

6.1 Failure Case Detection for Eyeball

After the robust face tracking under occlusion, we use the model-to-depth distance to detect the occluded region on the face. That is, we label the vertices that are more than σ away from their closest depth as occluded. For tracking failures caused by other reasons, such as fast motions or extreme eye poses, we use the average per-vertex energy E_v, calculated by dividing the energy in Eqn. (8) by the number of pixels in each eye region, to detect the tracking failures on either of the two eyes. As the user is required to remain in a neutral expression and without occlusion in the first frame, we calculate E_v for this frame, denoted as E_v^0, and use $1.3 * E_v^0$ as the threshold for failure detection. Note that our system is not sensitive to this threshold for two reasons. First, E_v in failure cases are typically extremely larger than in success cases. Second, the threshold can be set to a relatively small value because incorrectly classifying a success case as a failure case only triggers our prediction step (introduced in the following section), which will also provide a plausible result.

6.2 Online Training

If neither of the two eye regions fail in the tracking in one frame, the reconstructed face and eyeball motion are used in the online training. Here, we train linear regressors that map the motion parameters from one eye to the opposite one. In this case, the motions of the two eyes should be represented separately. The eyeball rotations of the two eyes are denoted as $\left[\theta^L, \phi^L\right]^T$ and $\left[\theta^R, \phi^R\right]^T$, where $*^L$ and $*^R$ indicate the left and right eyes, respectively. For the surface deformations of the two eye regions, as our multilinear model on the expression domain is reconstructed by basis with motions on local regions, we extract expression parameters that represent the local motions on the two eye regions, denoted as $\left[w_1^L, \ldots, w_F^L\right]^T$ and $\left[w_1^R, \ldots, w_F^R\right]^T$. $F = 8$ is because there are 8 types of motions for each eye region in our multilinear model. With these representations, the linear regressors are trained as follows:

$$\begin{aligned} M_1^L * \left[\theta^L, \phi^L, 1\right]^T &= \left[\theta^R, \phi^R, 1\right]^T, \\ M_2^L * \left[w_1^L, \ldots, w_F^L, 1\right]^T &= \left[w_1^R, \ldots, w_F^R, 1\right]^T. \end{aligned} \tag{10}$$

M_1^L and M_2^L are the regression matrices for the eyeball rotations and surface deformations of eye regions, respectively. With more training samples involved in the recording, M_1^L and M_2^L can be estimated increasingly more robustly. Note that we use the same method to train matrices M_1^R and M_2^R to map motions from the right eye to the left eye. This bidirectional training guarantees that the regressor can be used when either of the two eyes is occluded or fails in the tracking.

To avoid data explosion in the training stage, we do not always add new samples into the training. After involving the first $P = 150$ training samples to estimate the regression matrices, we reconstruct PCA spaces for the training data $\left[\theta^L, \phi^L, \theta^R, \phi^R\right]^T$ and $\left[w_1^L, \ldots, w_F^L, w_1^R, \ldots, w_F^R\right]^T$, respectively, and choose eigen-components that represent 95 percent of all the training samples. Then, if a new training sample can be represented by the PCA space with errors no greater than 5 percent, this sample is considered to be redundant and ignored. Otherwise, the new sample is involved, the regression matrices are updated, and the PCA spaces are rebuilt for testing future samples.

6.3 Eye Motion Prediction

After obtaining the regression matrices, if one eye is detected as failed while the opposite one is correct, we use the regressors to predict the motion of the failed eye. Our failure case detector can discover which eye fails in tracking; thus, we choose $\left\{M_1^L, M_2^L\right\}$ or $\left\{M_1^R, M_2^R\right\}$ accordingly. Then, the predicted motion parameters replace the original ones to generate the final results.

7 Experiments

In this section, we demonstrate the effectiveness of our method. First, because previous works rarely focus on eye performance reconstruction, we compare our method with several state-of-the-art methods of pupil center localization, which only estimate the 2D locations of pupils on the image domain. Second, we evaluate some key components of our method, showing their unique contributions to the entire system. Third, we demonstrate the final results of our technique for different users, different eye motions with different facial expressions, partial occlusions and varying lighting conditions. Finally, we discuss the limitations of our technique. Note that for all our experiments, we use a multilinear model to estimate a personalized face, but the position and size of the eyeball are simply decided by the rules introduced in Section 5.1.

Performance. Our system is implemented on a computer with a 3.60 GHz eight-core CPU, 16 GB RAM and an NVIDIA Geforce GTX 980 graphics card. For each input frame, our system takes 25.82 ms for facial expression tracking and 3.25 ms for eye tracking. If an occlusion or tracking failure is detected, we need an additional 0.11 ms for the regression-based eye motion refinement. Otherwise, the training stage requires approximately 5.7 ms on another thread. Our system is primarily implemented on a CPU, except the multilinear model-based facial expression estimation, which is implemented on a GPU. On average, our system runs within 30 ms per frame.

7.1 Comparisons

With estimated 3D head poses, traditional pupil center localization techniques can be naively extended to achieve 3D eye performance reconstruction by aligning the 3D line of sight with the 2D pupil location on the image domain. Our technique (rely on RGBD data) is compared with several naive solutions (rely on RGB data) based on the state-of-the-art pupil center localization techniques, which are used as benchmarks in previous publications [15], [33], [48]. To perform quantitative comparisons, the inter-pupil distance normalized errors [58] and corresponding standard deviations are calculated and compared in Fig. 9 (left). The errors are calculated on three videos (denoted as videos 1, 2 and 3 in the accompanying video) with different pupil appearances and different lighting conditions. Each of the three videos contains 300 frames. The ground-truth pupil center is labeled by an artist who is asked to click the pupil

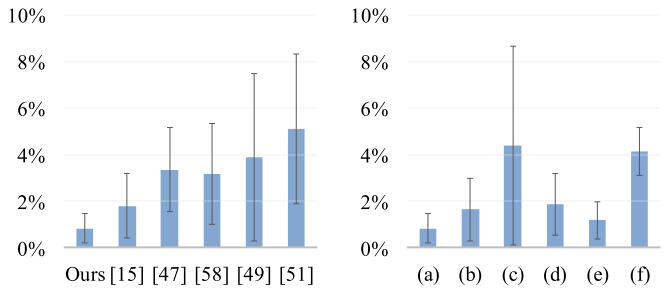


Fig. 9. Eye tracking errors normalized by inter-pupil distances and corresponding standard deviations. Left: comparisons with different methods. Right: evaluations of different components in our method. (a) our full method; (b) without appearance estimation; (c) without 3D Taylor expansion; (d) without L0 term; (e) bidirectional regression; (f) naive regression.

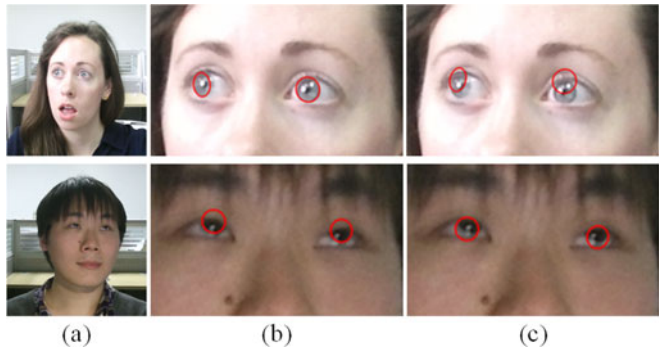


Fig. 11. Appearance estimation for SGAEM. (a) input images; (b) results with appearance estimation; (c) results with predefined appearance.

center positions on images for two rounds, in which the second round is to correct the center position if necessary. Note that the three videos are also used in the following evaluations.

Moreover, to visually compare these methods, we project the iris boundaries to the recorded images in Fig. 10. In the selected frames (Video1), our method outperforms the previous methods in terms of tracking accuracy. Furthermore, our technique reconstructs both the geometry and appearance model for the specific user, which is not considered by these previous methods. Finally, the facial animation results are compared in the accompanying video. Noticeable temporal jittering and tracking errors can be observed in the results of previous methods.

7.2 Evaluations

We first evaluate our appearance estimation of SGAEM. If the appearance of SGAEM is consistent with the input image, eyeball motion can be correctly estimated by minimizing the differences between the rendered eyeball and the input image. Otherwise, the result will become inaccurate. This is more apparent in two cases. First, different human races may have different iris colors. If an appearance model with a darker iris color is applied in the eye tracking of a character with a lighter iris color, the incorrect appearance causes errors as shown in the first row of Fig. 11. Note that our appearance estimation correctly extracts the iris color from the input image and thus obtains accurate results. Second, when the lighting condition changes during the recording, a fixed model is not able to represent the changes in appearance, thus leading to tracking errors as shown in the second row of Fig. 11. Because our method estimates appearance for each input frame, this dynamic appearance model correctly handles the lighting change problem. The tracking errors with and without appearance estimation are presented in Fig. 9 (right).

3D Taylor expansion is also a key component of our system. As illustrated in Fig. 6, 3D Taylor expansion successfully excludes the noisy gradients in the iris and sclera regions, thus achieving better results of eyeball tracking. In the 3D expansion-based optimization, we also propose an $L0$ term to exclude the influence of the highlight in our method. The improvement is demonstrated in Fig. 8, where the highlight on the bottom of the iris pushes the estimated iris upward in the result without the $L0$ term. The tracking errors in these two evaluations are also listed in Fig. 9 (right). More visual results of these evaluations are presented in the accompanying video.

Our online bidirectional regression makes our system more robust in eye motion tracking. It provides benefits in two situations, as shown in Fig. 12. First, when one of the

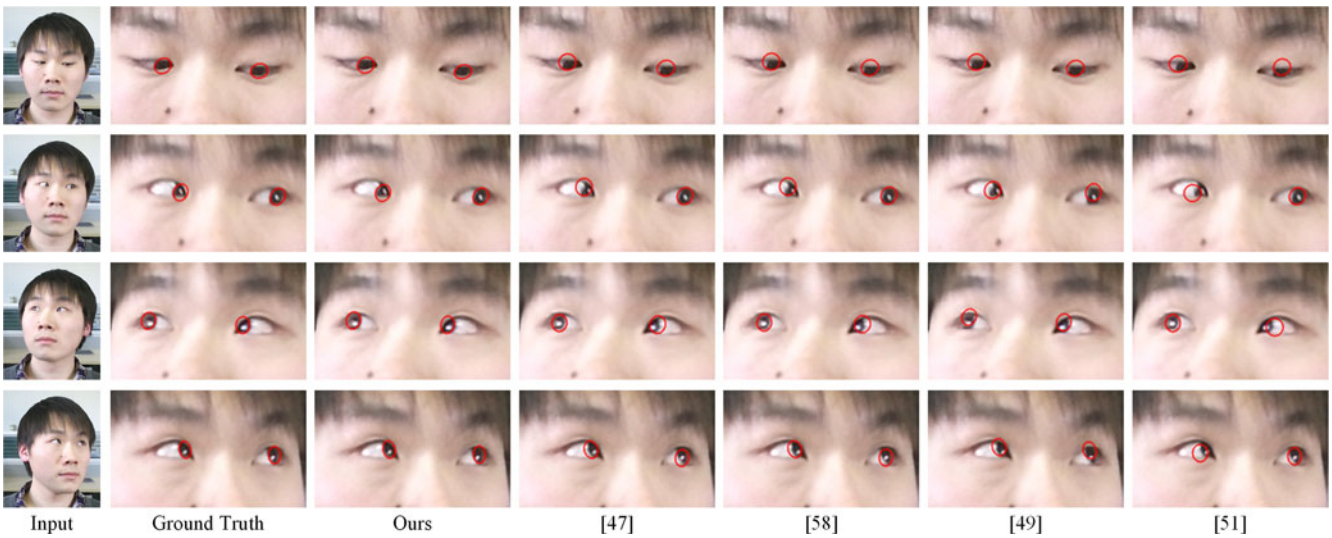


Fig. 10. Tracked iris overlapped on input images.

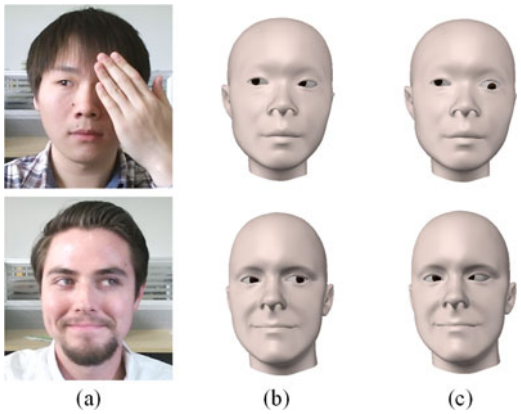


Fig. 12. Online bidirectional regression for occlusion and tracking failure handling. (a) input images; (b) results with the regression; (c) results without the regression.

two eye regions is occluded, the regression estimates the motion of the occluded eye (both the eyeball and the eyelid) from the observed one. Second, when a tracking failure is detected in one eye region, the regression also corrects this failure using the opposite eye. Our bidirectional regressor is compared with a naive regressor, which simply copies the performance of one eye to the other. We calculate the motions of the right eye in Video1 from the left eye by the two regressors such that the accuracy of these generated results can be estimated by the labeled ground truth, which is also demonstrated in Fig. 9 (right). Both the quantitative and visual results (Fig. 13 and accompanying video) illustrate that our bidirectional regressor generates more accurate and natural eye performances than the naive one. In addition, we compare our method with [15] in the situation where one of the two eyes is occluded. In this case, [15] fails to track both of the eyes, whereas our method reconstructs reasonable results, as shown in Fig. 14.

Moreover, we perform a user study with 32 participants to evaluate our bidirectional regression. Each participant is shown a video including four side-by-side parts. One of them is the input sequence, whereas the other three are generated visual results (our full method, bidirectional regression and naive regression). Then, participants are asked to evaluate how realistic and convincing the reconstructed eye performances appear in the three results on a scale from 5 (the highest score) to 0 (the lowest score). Consistent with the numeric measurements, the tracking with our full method generates the best results (average score of 4.34). Our bidirectional regression is ranked as the second best (average score of 4.16) and clearly outperforms the naive regression (average score of 2.75). This result is statistically significant with a one-way ANOVA p-value < 0.01.

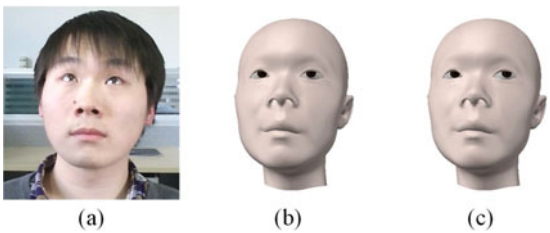


Fig. 13. Comparison of different regressions. (a) input image; (b) result with bidirectional regression; (c) result with naive regression.

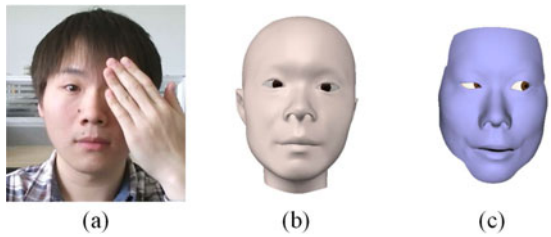


Fig. 14. Comparison of our method and [15] in the situation of occlusion. (a) input image; (b) result of our method; (c) result of [15].

7.3 Results

Fig. 15 displays the final results of our method on different users with different head poses, expressions and eye motions. Different iris sizes, iris appearances, highlights and occlusions are all demonstrated in these results. In the accompanying video, we also show sequences with varying lighting conditions and other results to demonstrate the accuracy and robustness of our technique.

7.4 Limitations

Our system still relies on an RGBD camera rather than an RGB camera, which is more available for end users. Note that depth information is only used in 3D facial surface tracking and occlusion detection, whereas our proposed eye performance reconstruction does not require depth information. Because techniques for video-based facial tracking and occlusion detection [12], [67] have also recently been proposed in the literature, we can integrate these techniques into our system to remove the requirement of depth information. However, robust eyeball tracking may not be easily achieved by this direct implementation, and more investigations are still required. As this is not the contribution of this technique, we leave this to future work.

Although the eyeball motion is correctly reconstructed by our technique, the eyelid shape and motion are still not accurate, which dramatically downgrades the quality of the final result in the eye region. As shown in Fig. 16, the eyeball motion is correct (shown in Fig. 16b by projecting the iris onto the input image). However, because the eyelid is not correctly tracked, the overall result does not match the input very well.

Our scheme utilizes pixels in eye regions to estimate the eye performance. Therefore, its accuracy decreases when there are considerably fewer pixels in eye regions (e.g., looking downward, extreme head pose). In this case, motion priors may be involved to achieve better robustness.

Our eye tracking method requires facial surface tracking to define the position and size of the eyeball by some simple rules, which may not be correct compared with the real case. In addition, our system shares common failure cases with current face tracking techniques, such as extreme head poses and lighting conditions. Moreover, our method requires a pre-fixed eyeball size and position, and an initial value of iris size, which is set once by the artist.

8 Conclusion

In this paper, we achieve real-time eye performance reconstruction using a single RGBD sensor. Our technique is in a unified framework that directly estimates 3D eyeball motion

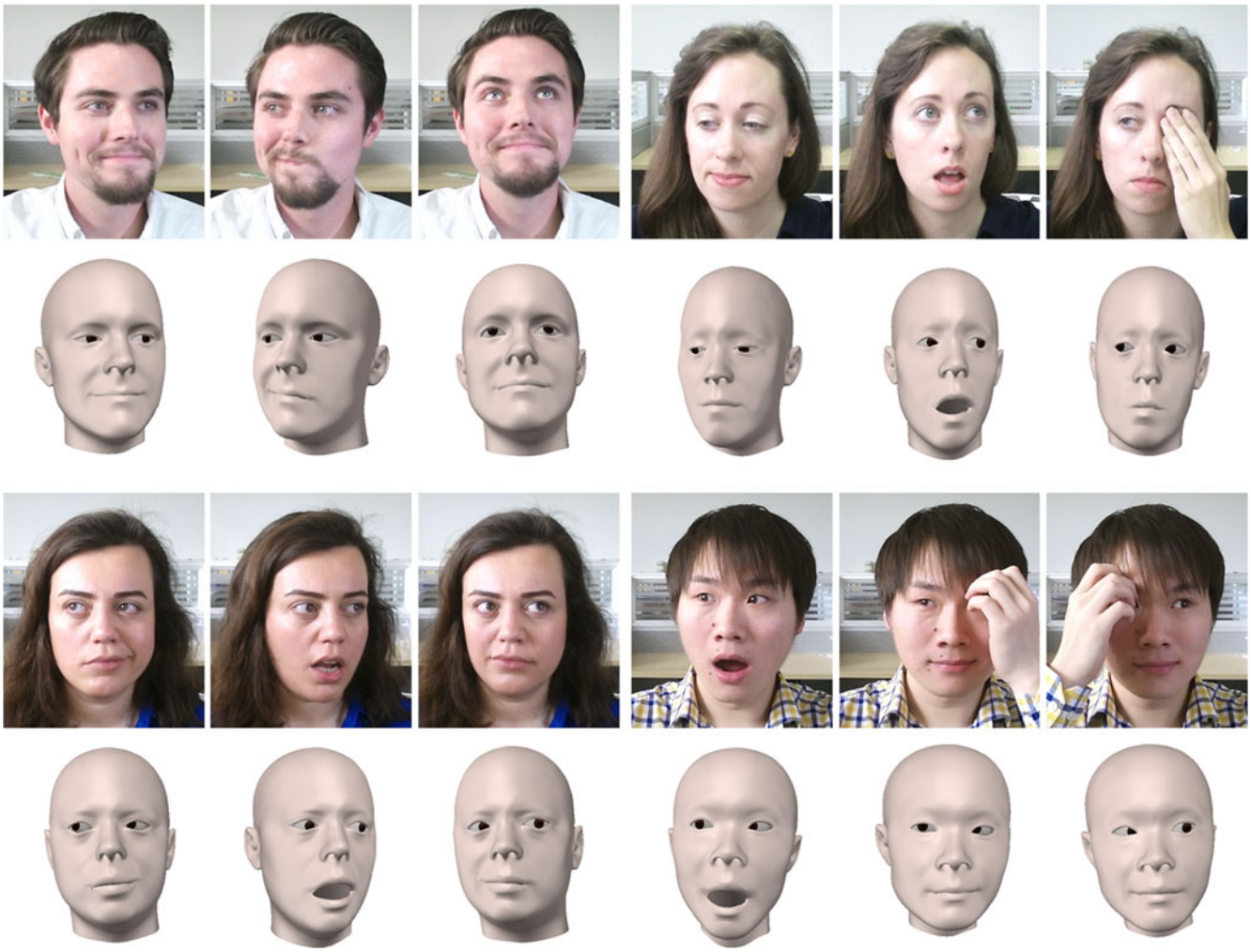


Fig. 15. Final results of our technique with 4 characters and 3 different eye poses for each. The input image is on the top, and our result is on the bottom.

from input images, without requiring 2D pupil center localization. Our system does not require any pre-training stages but rather proposes a novel method to explicitly handle different iris sizes and appearances, lighting variations and highlights on images, which will involve a considerable amount of labeled data for training-based solutions. Furthermore, we propose an online bidirectional regression method that succeeds in handling occlusions and tracking failures in either of the two eyes. As we take online samples of the specific user and train the regression model on the fly, we do not require any user adaption or pre-training steps.

APPENDIX

As described in Section 5.2.2, Eqn. (5) is solved to estimate the eyeball poses. Specifically, $C_k(\theta^0, \phi^0)$ and I_k are known values from the current eyeball rotation and the input image. Then, the two partial derivatives are calculated as follows:

$$\begin{aligned}\frac{\partial C_k(\theta^0,\phi^0)}{\partial \theta} &= C^0(\theta_k+\bar{\theta},\phi_k) - C^0(\theta_k-\bar{\theta},\phi_k),\\ \frac{\partial C_k(\theta^0,\phi^0)}{\partial \phi} &= C^0(\theta_k,\phi_k+\bar{\phi}) - C^0(\theta_k,\phi_k-\bar{\phi}).\end{aligned} \tag{11}$$

Here, θ_k and ϕ_k represent the world polar coordinates of vertex k. $\bar{\theta}$ and $\bar{\phi}$ (in practice, 0.06 in radian for both) are the offsets along the θ and ϕ directions, respectively. $C^0(\theta_k, \phi_k)$ provides the intensity of vertex k under the current eyeball rotation (θ^0, ϕ^0). As in Eqn. (11), the gradients of vertex k are estimated by the intensity differences between the two points with opposite offsets on a certain direction. Therefore, Eqn. (5) can be solved from the following equations:

$$\begin{aligned}\frac{\partial E}{\partial \delta_\theta} &= \sum_{k\in T} 2\frac{\partial C_k}{\partial \theta}\left(\left(C_k(\theta^0,\phi^0) + \frac{\partial C_k}{\partial \theta}\delta_\theta + \frac{\partial C_k}{\partial \phi}\delta_\phi\right) - I_k\right) = 0,\\ \frac{\partial E}{\partial \delta_\phi} &= \sum_{k\in T} 2\frac{\partial C_k}{\partial \phi}\left(\left(C_k(\theta^0,\phi^0) + \frac{\partial C_k}{\partial \theta}\delta_\theta + \frac{\partial C_k}{\partial \phi}\delta_\phi\right) - I_k\right) = 0.\end{aligned} \tag{12}$$

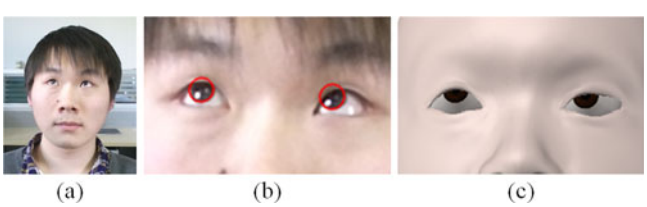


Fig. 16. Negative effect of inaccurate eyelid shape on the final visual result. (a) input image; (b) tracked iris overlapped on input image; (c) final visual result.

Similarly, Eqn. (7) is solved to estimate the iris size ψ. The partial derivative in this equation is defined as:

$$\frac{\partial C_k}{\partial \psi} = C^0(\psi_k + \bar{\psi}) - C^0(\psi_k - \bar{\psi}). \tag{13}$$

ψ_k is the polar angle of vertex k in the eyeball coordinates (the polar axis goes through the iris center). $\bar{\psi}$ denotes an offset (0.06 in radian) as $\bar{\theta}$ and $\bar{\phi}$. $C^0(\psi_k)$ is the intensity of vertex k. Consequently, δ_ψ can be estimated by the following equation:

$$\frac{\partial E}{\partial \delta_\psi} = \sum_{k \in T} 2 \frac{\partial C_k}{\partial \psi} \left(\left(C_k(\psi^0) + \frac{\partial C_k}{\partial \psi} \delta_\psi \right) - I_k \right) = 0. \tag{14}$$

ACKNOWLEDGMENTS

This work was supported by the NSFC (No. 61671268, 61672307) and the National Key Technologies R&D Program of China (No. 2015BAF23B03). Feng Xu is the corresponding author.

REFERENCES

[1] D. J. Roberts, J. Rae, T. W. Duckworth, C. M. Moore, and R. Aspin, "Estimating the gaze of a virtuality human," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 4, pp. 681–690, 2013.

[2] J. Orlosky, T. Toyama, K. Kiyokawa, and D. Sonntag, "Modular: Eye-controlled vision augmentations for head mounted displays," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 11, pp. 1259–1268, 2015.

[3] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-synthesis for appearance-based 3d gaze estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1821–1828.

[4] K. A. F. Mora and J.-M. Odobez, "Geometric generative gaze estimation (g3e) for remote RGB-D cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1773–1780.

[5] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4511–4520.

[6] A. Weissenfeld, K. Liu, and J. Ostermann, "Video-realistic image-based eye animation via statistically driven state machines," *Vis. Comput.*, vol. 26, no. 9, pp. 1201–1216, 2010.

[7] B. H. Le, X. Ma, and Z. Deng, "Live speech driven head-and-eye motion generators," *IEEE Trans. Visual. Comput. Graph.*, vol. 18, no. 11, pp. 1902–1914, Nov. 2012.

[8] Z. Deng, J. P. Lewis, and U. Neumann, "Automated eye motion using texture synthesis," *IEEE Comput. Graph. Appl.*, vol. 25, no. 2, pp. 24–30, Mar.-Apr. 2005.

[9] G. Francois, P. Gautron, G. Breton, and K. Bouatouch, "Image-based modeling of the human eye," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 5, pp. 815–827, Sep.-Oct. 2009.

[10] P. Bérard, D. Bradley, M. Nitti, T. Beeler, and M. H. Gross, "High-quality capture of eyes," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 223–221, 2014.

[11] A. Bermano, T. Beeler, Y. Kozlov, D. Bradley, B. Bickel, and M. Gross, "Detailed spatio-temporal reconstruction of eyelids," *ACM Transactions on Graphics*, vol. 34, no. 4, 2015, Art. no. 44.

[12] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Trans. Graph.*, vol. 33, no. 4, 2014, Art. no. 43.

[13] F. Shi, H.-T. Wu, X. Tong, and J. Chai, "Automatic acquisition of high-fidelity facial performances using monocular videos," *ACM Trans. Graph.*, vol. 33, no. 6, 2014, Art. no. 222.

[14] C. Cao, D. Bradley, K. Zhou, and T. Beeler, "Real-time high-fidelity facial performance capture," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 46.

[15] C. Wang, F. Shi, S. Xia, and J. Chai, "Realtime 3D eye gaze animation using a single rgb camera," *ACM Trans. Graph. (TOG)*, vol. 35, no. 4, p. 118, 2016.

[16] P.-L. Hsieh, C. Ma, J. Yu, and H. Li, "Unconstrained realtime facial performance capture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1675–1683.

[17] Y. Liu, F. Xu, J. Chai, X. Tong, L. Wang, and Q. Huo, "Video-audio driven real-time facial animation," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 182.

[18] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 478–500, Mar. 2010.

[19] E. D. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 1124–1133, Jun. 2006.

[20] C. H. Morimoto, A. Amir, and M. Flickner, "Detecting eye position and gaze from a single camera and 2 light sources," in *Proc. 16th Int. Conf. Pattern Recognit.*, 2002, vol. 4, pp. 314–317.

[21] S.-W. Shih and J. Liu, "A novel approach to 3-d gaze tracking using stereo cameras," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 34, no. 1, pp. 234–245, Feb. 2004.

[22] D. H. Yoo and M. J. Chung, "A novel non-intrusive eye gaze estimation using cross-ratio under large head motion," *Comput. Vis. Image Understanding*, vol. 98, no. 1, pp. 25–51, 2005.

[23] Z. Zhu, Q. Ji, and K. P. Bennett, "Nonlinear eye gaze mapping function estimation via support vector regression," in *Proc. 18th Int. Conf. Pattern Recognit.*, 2006, vol. 1, pp. 1132–1135.

[24] C. Hennessey, B. Noureddin, and P. Lawrence, "A single camera eye-gaze tracking system with free head motion," in *Proc. Symp. Eye Tracking Res. Appl.*, 2006, pp. 87–94.

[25] A. Nakazawa and C. Nitschke, "Point of gaze estimation through corneal surface reflection in an active illumination environment," in *Proc. Comput. Vis.–ECCV*, 2012, pp. 159–172.

[26] Z. Zhu and Q. Ji, "Eye gaze tracking under natural head movements," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 918–923.

[27] J. Chen and Q. Ji, "Probabilistic gaze estimation without active personal calibration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 609–616.

[28] T. Ishikawa, S. Baker, I. Matthews, and T. Kanade, "Passive driver gaze tracking with active appearance models," in *Proc. 11th World Congr. Intell. Transportation Syst.*, Oct. 2004.

[29] J. Chen and Q. Ji, "3d gaze estimation with a single camera without ir illumination," in *Proc. 19th IEEE Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.

[30] 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 *Proc. Symp. Eye Tracking Res. Appl.*, 2008, pp. 245–250.

[31] L. Jianfeng and L. Shigang, "Eye-model-based gaze estimation by RGB-D camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 592–596.

[32] L. Sun, M. Song, Z. Liu, and M.-T. Sun, "Real-time gaze estimation with online calibration," *IEEE MultiMedia*, vol. 21, no. 4, pp. 28–37, Oct.-Dec. 2014.

[33] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 802–815, Feb. 2012.

[34] S. Baluja and D. Pomerleau, "Non-intrusive gaze tracking using artificial neural networks," DTIC Document, Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep. 864994, 1994.

[35] K. Liang, Y. Chahir, M. Molina, C. Tijus, and F. Jouen, "Appearance-based gaze tracking with spectral clustering and semi-supervised gaussian process regression," in *Proc. Conf. Eye Tracking South Africa*, 2013, pp. 17–23.

[36] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Adaptive linear regression for appearance-based gaze estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2033–2046, Oct. 2014.

[37] K. A. F. Mora and J.-M. Odobez, "Gaze estimation from multimodal kinect data," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 25–30.

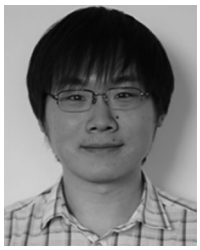
[38] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Head pose-free appearance-based gaze sensing via eye image synthesis," in *Proc. 21st Int. Conf. Pattern Recognit.*, 2012, pp. 1008–1011.

[39] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, "A head pose-free approach for appearance-based gaze estimation," in *Proc. 22nd British Mach. Vis. Conf.*, 2011, pp. 1–11.

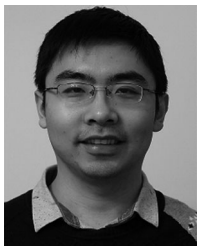
[40] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, "Learning gaze biases with head motion for head pose-free gaze estimation," *Image Vis. Comput.*, vol. 32, no. 3, pp. 169–179, 2014.

[41] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, "An incremental learning method for unconstrained gaze estimation," in *Proc. Comput. Vis.–ECCV*, 2008, pp. 656–667.

[42] O. Williams, A. Blake, and R. Cipolla, "Sparse and semi-supervised visual mapping with the s^ 3gp," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 1, pp. 230–237.
[43] E. Wood, T. Baltrusaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling, "Rendering of eyes for eye-shape registration and gaze estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV) IEEE*, 2015, pp. 3756–3764.
[44] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Facevr: Real-time facial reenactment and eye gaze control in virtual reality," arXiv:1610.03151, 2016.
[45] D. W. Hansen and J. P. Hansen, "Eye typing with common cameras," in *Proc. Symp. Eye Tracking Res. Appl.*, 2006, pp. 55–55.
[46] M. Türkan, M. Pardas, and A. E. Cetin, "Human eye localization using edge projections," in *Proc. VISAPP*, 2007, pp. 410–415.
[47] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: An open source facial behavior analysis toolkit," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2016, pp. 1–10.
[48] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers," in *Proc. Symp. Eye Tracking Res. Appl.*, 2014, pp. 207–210.
[49] F. Timm and E. Barth, "Accurate eye centre localisation by means of gradients," *VISAPP*, vol. 11, pp. 125–130, 2011.
[50] L. Bai, L. Shen, and Y. Wang, "A novel eye location algorithm based on radial symmetry transform," in *Proc. 18th Int. Conf. Pattern Recognit.*, 2006, vol. 3, pp. 511–514.
[51] R. Valenti and T. Gevers, "Accurate eye center location and tracking using isophote curvature," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[52] S. Asteriadis, N. Nikolaidis, A. Hajdu, and I. Pitas, "An eye detection algorithm using pixel to edge information," in *Proc. Int. Symp. Control, Commun. Sign.*, 2006.
[53] P. Campadelli, R. Lanzarotti, and G. Lipori, "Precise eye localization through a general-to-specific model definition," in *Proc. British Mach. Vis. Conf.*, 2006, pp. 187–196.
[54] D. Cristinacce, T. F. Cootes, and I. M. Scott, "A multi-stage approach to facial feature detection." in *Proc. British Mach. Vis. Conf.*, 2004, pp. 1–10.
[55] M. Hamouz, J. Kittler, J.-K. Kamarainen, P. Paalanen, H. Kälviäinen, and J. Matas, "Feature-based affine-invariant localization of faces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1490–1495, Sep. 2005.
[56] S. Kim, S.-T. Chung, S. Jung, D. Oh, J. Kim, and S. Cho, "Multi-scale gabor feature based eye localization," *World Academy Sci. Eng. Technol.*, vol. 21, pp. 483–487, 2007.
[57] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao, "2d cascaded adaboost for eye localization," in *Proc. 18th Int. Conf. Pattern Recognit.*, 2006, vol. 2, pp. 1216–1219.
[58] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 FPS via regressing local binary features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1685–1692.
[59] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "Facewarehouse: A 3d facial expression database for visual computing," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 3, pp. 413–425, Mar. 2014.
[60] H. Li, J. Yu, Y. Ye, and C. Bregler, "Realtime facial animation with on-the-fly correctives," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 42.
[61] S. Bouaziz, Y. Wang, and M. Pauly, "Online modeling for realtime facial animation," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 40.
[62] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Dig. Imaging Model.*, 2001, pp. 145–152.
[63] P. Bérard, D. Bradley, M. Gross, and T. Beeler, "Lightweight eye capture using a parametric model," *ACM Trans. Graph.*, vol. 35, no. 4, 2016, Art. no. 117.
[64] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "A 3d morphable eye region model for gaze estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 297–313.
[65] C. Wu, K. Varanasi, and C. Theobalt, "Full body performance capture under uncontrolled and varying illumination: A shading-based approach," in *Proc. Comput. Vis.–ECCV*, 2012, pp. 757–770.
[66] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l 0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6., 2011, Art. no. 174.
[67] S. Saito, T. Li, and H. Li, "Real-time facial segmentation and performance capture from RGB input," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016.



Quan Wen received the BS degree in School of Software from Tsinghua University, China, in 2014. He is currently working toward the PhD degree in School of Software at Tsinghua University.



Feng Xu received the BS degree in physics from Tsinghua University, Beijing, China, in 2007, and the PhD degree in automation from Tsinghua University, Beijing, China, in 2012. He is currently an assistant professor in School of Software, Tsinghua University. His research interests include face animation, performance capture and 3D reconstruction.

Jun-Hai Yong received the BS and PhD degrees in computer science from the Tsinghua University, China, in 1996 and 2001, respectively. He is currently a professor in School of Software, Tsinghua University. He held a visiting researcher position in the Department of Computer Science, Hong Kong University of Science & Technology, in 2000. He was a post doctoral fellow in the Department of Computer Science, University of Kentucky from 2000 to 2002. He obtained a lot of awards such as the National Excellent Doctoral Dissertation Award, the National Science Fund for Distinguished Young Scholars, the Best Paper Award of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation, the Outstanding Service Award as Associate Editor of the Computers & Graphics Journal by Elsevier, and several National Excellent Textbook Awards. His main research interests include computer-aided design and computer graphics.