3D Face Reconstruction and Gaze Tracking in the HMD for Virtual Interaction

Shu-Yu Chen, Yu-Kun Lai, Shihong Xia, Paul L. Rosin and Lin Gao*

***Abstract*—With the rapid development of virtual reality (VR) technology, VR headsets, a.k.a. Head-Mounted Displays (HMDs), are widely available, allowing immersive 3D content to be viewed. A natural need for truly immersive VR is to allow bidirectional communication: the user should be able to interact with the virtual world using facial expressions and eye gaze, in addition to traditional means of interaction. The typical application scenario includes VR virtual conferencing and virtual roaming, where ideally users are able to see other users' expressions and have eye contact with them in the virtual world. In addition, eye gaze also provides a natural means of interaction with virtual objects. Despite significant achievements in recent years for reconstruction of 3D faces from RGB or RGB-D images, it remains a challenge to reliably capture and reconstruct 3D facial expressions including eye gaze when the user is wearing an HMD, because the majority of the face is occluded, especially those areas around the eyes which are essential for recognizing facial expressions and eye gaze. In this paper, we introduce a novel real-time system that is able to capture and reconstruct 3D faces wearing HMDs, and robustly recover eye gaze. We further propose a novel method to map eye gaze directions to the 3D virtual world, which provides a novel and useful interactive mode in VR. We compare our method with state-of-the-art techniques both qualitatively and quantitatively, and demonstrate the effectiveness of our system using live capture.**

***Index Terms*—real-time facial performance capture, eye tracking, head-mounted display, user interaction, communication, virtual reality**

I. Introduction

With the rapid development of virtual reality (VR) technology, both the academic and industry communities have contributed a lot to create a more mature and popular VR technology. One of the contributions is the various types of portable head-mounted displays (HMDs) which could bring immersive 3D VR content to the users. To provide a truly immersive experience, it is essential to allow users to *interact* naturally with the virtual environment. Some works about content creation and exploration in virtual reality are introduced in the survey [1]. Moreover, arguably the most reasonable form of interaction between human subjects is via facial expression and eye contact. As described in [2], [3], emotions are often conveyed through the eyes which are an indispensable part of facial expression. However, the feeling of immersion on current multi-user HMD systems is problematic due to the lack of real-world expression and eye gaze in the virtual world. To realize the interaction, they either require additional input devices (e.g. game consoles), which are non-immersive and distracting, or ask the user to perform slow and unnatural head movements. In contrast, eye gaze is a natural way of showing interest in objects, which is both intuitive and efficient. Therefore, capture and reconstruction of facial expressions and estimation of eye gaze while wearing HDMs *are urgent problems to be solved in VR technology*. Such technology makes it possible for users to see each other's expression and have eye contact, making VR much more realistic and immersive, and user interactions more natural. Moreover, eye gaze tracking has additional benefits for VR, for example, to reduce the scene rendering time by prioritizing rendering of objects close to the user's focus, and to provide a means of interaction with objects, e.g., virtual selection using eye gaze.

Expression animation is essential for a wide range of applications, such as movie production, 3D games, etc. Significant effort has been put into 3D facial expression capture and reconstruction. Most of the work reconstructs 3D facial expressions by using a single RGB camera, including methods that reconstruct 3D expressions offline [4] and real-time reconstruction of 3D expressions [5], [6]. Such approaches, however, have failed to work in the cases where the user is wearing VR glasses, since essential facial features are occluded. Moreover, such work only reconstructs facial expressions without tracking the line of sight, which is also important for interaction. Among various communication mechanisms, facial expressions and eye contact provide essential clues for emotion, attentional focus and future intentions [7]. They are often subtle, but effectively perceived by human viewers. Capturing high quality facial expressions and eye gaze is necessary to improve the realism of 3D animation.

More recent attention has focused on the reconstruction of facial expression and eye gaze. Wang et al. [8] developed the first system that is able to simultaneously perform 3D facial expression reconstruction and eye gaze tracking, using a single RGB camera as input. However, their method only deals with the case where the face is clearly visible without occlusion, and so does not work when HMDs are worn. The methods proposed by Li et al. [9] and Olszewski et al. [10] are able to reconstruct 3D facial expressions while wearing HMDs. However, tracking the eye gaze and building a personalized

* Corresponding Author is Lin Gao (gaolin@ict.ac.cn).

S.-Y Chen, S. Xia and L. Gao are with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. S. Xia and L. Gao are also with the University of Chinese Academy of Sciences, Beijing 100049, China. E-mail:{chenshuyu, xsh, gaolin}@ict.ac.cn.

Y.-K. Lai and P.L. Rosin are with the School of Computer Science & Informatics, Cardiff University, Wales, UK. Email: {LaiY4, RosinPL}@cardiff.ac.uk.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. This material includes a video for real-time recording and a PDF file with appendix.

Manuscript received September XX, 2021; revised September XX, 2021.

facial model is beyond the scope of these works. Moreover, the former equips the HMD with an external depth camera, and uses indirect pressure sensors to estimate the facial expression coefficients of the regions hidden by the HMD. However, the impact of these pressure sensors on the estimated facial expression is understudied since the strain signal may be different for each user, and the strain gauge requires a stable contact to the face. The latter utilizes the separate visible parts (e.g. around the mouth area) of the face along with audio, and heavily relies on machine learning to *infer* 3D facial expressions rather than actually sensing them.

Unlike such works, we propose to directly *capture* occluded facial features within HMDs by using 3 infrared (IR) cameras where two cameras are used for capturing the eyes and the remaining camera captures the unoccluded face (see Fig. 1). Infrared LED lamps are used to illuminate the eyes and the face, projecting invisible light so as to not affect the user experience. An illustration of our hardware setup is shown in Fig. 1(c). To reconstruct 3D facial expressions, we *fuse* the output of the IR cameras, detect feature points for each IR image and use them to drive 3D facial models. We further propose a new eye gaze tracking method based on sampling and correlation of 3D directions with captured 2D IR images of the eyes, which improves accuracy and robustness. Facial expressions are reconstructed based on captured images of facial features, and the 3D facial expressions and eye gaze are simultaneously reconstructed in real time. The outside camera is fixed to the HMDs, along with cell phone sensors to track head rotations. Thus, our system is capable of handling the variation of facial orientation, which means the user can move their head freely and has a more immersive VR experience. Our system also works out the eye gaze focal point in the virtual world, which allows quick and intuitive interaction such as object selection. Fig. 17 shows a typical scenario demonstrating our system in action. Users wearing HMDs participate naturally in a conversation where they are able to see each other's facial expression and have eye contact, even if they are geographically apart. The main contributions of our paper are as follows:

- We propose a novel real-time system to capture and reconstruct both 3D facial expressions and eye gaze while wearing HMDs. Our system captures subtle expression and eye movement, and allows free head movement.
- We develop a new eye gaze tracking technology which identifies eye gaze direction by sampling in the 3D space and maximizing correlation with captured IR images, and eye gaze is used to interact with the VR scenes in an intuitive and flexible manner.

II. Related Work

In the past couple of years, with the emergence of HMDs and the development of the virtual game industry, we have seen significant advances in VR technology and applications. More recent attention has focused on how to improve the immersion and interactivity of the HMDs. We will first begin by reviewing state-of-the-art methods of facial performance capture and eye gaze tracking. It will then go on to the research that specifically addresses these functionalities with HMDs.

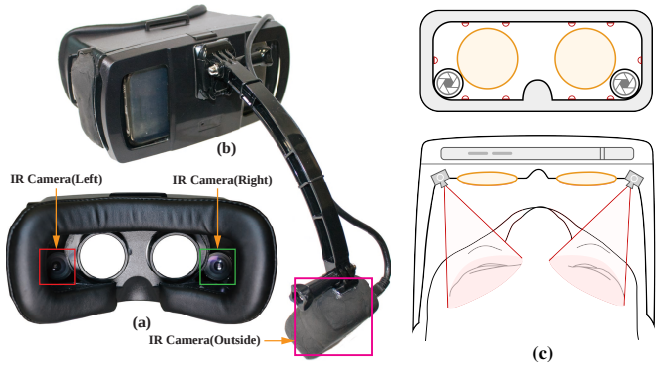


Fig. 1. Our hardware setup: HMDs fitted with three infrared (IR) cameras. (a) the cameras shown in the red and green boxes capture the left and right eye images respectively, (b) the camera shown in the magenta box is used to capture unoccluded facial motion. (c) the camera configuration and the location of the infrared LED lamps.

A. Facial Reconstruction and Gaze Tracking without HMDs

3D facial expression reconstruction is an active topic due to its wide use in the movie and game industry. Most research investigating facial reconstruction has utilized either RGB or RGB-D cameras. Some methods [11], [12], [13], [14], [15] use multiple RGB cameras to capture 3D facial expressions. Although effective, these methods are complex to set up. To address this, methods using a single camera are developed. Zhang et al. [16] propose a semantic volumetric representation and use it to detect 3D landmarks from an image. Song et al. [17] propose a coupled radial basis function network (C-RBF) to recover the mapping between 2D and 3D faces. Cao et al. [18] build a database containing 150 subjects each with 47 facial expressions, and propose a multi-linear fitting method to reconstruct 3D faces from 2D feature points. Cao et al. [5] further propose a method which adds wrinkle details to the reconstructed facial expressions by building relationships between images and wrinkles. This method reconstructs the geometric details of the expression in real time and makes the results more realistic. Kong et al. [19] design a headpose estimation method requiring not only color images but also additional depth information. Image segmentation is also used in facial tracking, such as Hsieh et al. [20] with an RGB-D sensor and Saito et al. [21] with RGB input. Recently, neural networks are also used to reconstruct faces. Tu et al. [22] take a Convolutional Neural Network (CNN) as a regressor to get the 3D Morphable Model (3DMM) parameters from an image, and they also propose a novel self-critical learning-based mechanism to improve the 3D face model. Based on 3DMM, Wen et al. [23] propose a method to predict the mouth movement from audio input and then use it to synthesize video to improve the realism of the generated results. Chai et al. [24] propose a dual-stream framework to decompose the face into geometry and texture streams, and find the corresponding relationship between the 3DMM albedo map and the original face in the texture stream. Unlike using a collected database, Fan et al. [25] propose dual neural networks using 3D point clouds to represent face geometry and utilize Markov random fields to refine it. Chaudhuri et al. [26] design an end-to-

end framework to disentangle the facial geometry and albedo maps, but they need 3D scans to train the model. To address it, Yao et al. [27] propose an framework which could reconstruct the 3D faces with wrinkle from images and also animate the 3D face with realistic wrinkle. The above methods reconstruct the 3D facial expressions without eye gaze, which affects the realism of captured facial animation.

In our daily face-to-face interactions, we pay attention not only to each other's facial expressions, but also to their eye motions. Consequently, the research on eye movement is an active topic in recent years. Wang et al. [28], Hansen et al. [29] and Kar et al. [30] summarize the related work of gaze model and estimation. Wood et al. [31] first get the boundary points for the iris region, and then use random sample consensus (RANSAC) to fit ellipses to those boundaries. They map the fitted ellipse into a 3D model to get 3D position and orientation of the eyes. Their method achieves a rate at 12 fps which is still unsuitable for high quality real time performance.

Calibration-free methods are more convenient in gaze tracking, e.g. Tripathi et al. [32] utilize Gaussian Process Regression models to achieve eye gaze tracking, Chen et al. [33] estimate gaze based on gaze probability computed from either a saliency map for a static image-viewing task or a Gaussian distribution for viewing continuous movies of a dynamic environment. Sugano and Bulling [34] capture 20 users' gaze data, analyze the offset errors, and present an automatic calibration method for eye tracking. Wood et al. [35] use high-quality head scans to build a 3D morphable model of the eye region that incorporates a separate eyeball component modeled from anatomical measurements and high-resolution iris photos. Gaze estimation is performed by local optimization using analysis-by-synthesis, which takes several seconds per image that is too slow to use for a real-time system. Sugano et al. [36] use a learning-by-synthesis method to attain calibration-free gaze estimation and build a multi-view gaze database for learning. Morimoto et al. [37] design one of the most successful eye gaze capture approaches. Due to its high level of simplicity and effectiveness, many commercial eye trackers adopt this technique. More methods are proposed, through tracking the iris contour [38] or the pupil contour [39]. For the human face, there is asymmetry in the eye area. By analyzing this, Cheng et al. [40] designed a face-based asymmetric regression-evaluation method to estimate eye gaze.

Lu et al. [41] propose an appearance-based regression method, which focuses on the representation of "uncalibrated gaze pattern" and asigns it to gaze positions. Pfeuffer et al. [42], [43] develop a calibration method which can perform calibration without the user's awareness. Instead of focusing on tracking, they propose a strategy for data selection which is used in the calibration step. This method is based on the user's attention to a moving object, and if the user is distracted this will affect the accuracy. And for each use, the calibration data is different, which may cause interference in the comparison of methods.

Cao et al. [44] use a regression-based method to detect facial 2D landmarks including eyes, nose and chin from an image, but the landmarks are not in 3D space. To address this, Wang et al. [8] develop the first real-time 3D eye gaze capture method with a web-camera. They employ a multi-linear method and use a maximum a posterior (MAP) probabilistic framework to reconstruct facial models with 3D eye gaze. However, this method fails to track eye gaze if there is blinking or facial occlusion, and therefore does not work in VR applications. Several methods [45], [8], [46] are proposed to estimate eye gaze, however these methods are not designed for HMDs and cannot be directly applied to the HMD settings.

B. Facial Reconstruction and Gaze Tracking with HMDs

Some pioneering works are proposed to reconstruct the 3D face while the user wears an HMD. Li et al. [9] are the first to solve the problem of expression capture when having large occlusion with an HMD. They propose to estimate the expression parameters of the occluded area using the electrical signals obtained by the strain sensor and use an external depth camera fixed on an HMD to capture facial geometry. They integrate those two sources of information and apply them to the virtual avatar to produce the expression animation. Luo et al. [47] also use electrical signals, but they need to pre-capture an image without the HMD to build personal blendshapes. Olszewski et al. [10] use a fixed external RGB camera to control a predefined avatar in real-time. Image acquisition is divided into two parts, the RGB data is for the mouth region and the eye image is captured by the integrated camera inside the HMD. They use a convolutional neural network to get the expression parameters from the image, so as to realize the expression performance of avatar. This research work is focused on controlling a digital avatar, rather than for realistic 3D facial reconstruction and does not take eye-gaze tracking into consideration which is necessary to provide a vivid immersive experience. Wei et al. [48] and Lombardi et al. [49] could animate 3D avatar heads through the built-in camera. But they are not general methods, they need to build 3D head and then train the network for each user.

Zhao et al. [50] focus on facial image synthesis. They put two near-infrared (NIR) cameras inside the HMD and one RGB camera outside. Although they reconstruct facial geometry they use the IR images and facial model for RGB image retrieval to synthesize a complete facial image, and eye-gaze is not taken into consideration. Rekimoto et al. [51] propose a concept called "face-through HMD", but they need to pre-scan a detailed 3D model and pre-calibrate the correspondence between model and the image of the eye area for each user. They use IR-image colorization to change the eye region of the model texture, rather than 3D facial reconstruction or eye tracking. Lombardi et al. [52] split the facial area into three parts and design a system to obtain facial geometry and texture through the input of three images. Image-to-model mapping is not directly trained; instead they pre-train a conditional variational autoencoder (CVAE) to reconstruct the mesh and texture. Then, they reconstruct the images using a VAE and learn a transformation from the latent space in the VAE to the latent space in the CVAE. Although this method could drive a detailed 3D model, it is specific to a particular person. Commercial eye gaze tracking systems such as Tobii [53] and Pupil Lab [54] use IR cameras attached in the HMDs.

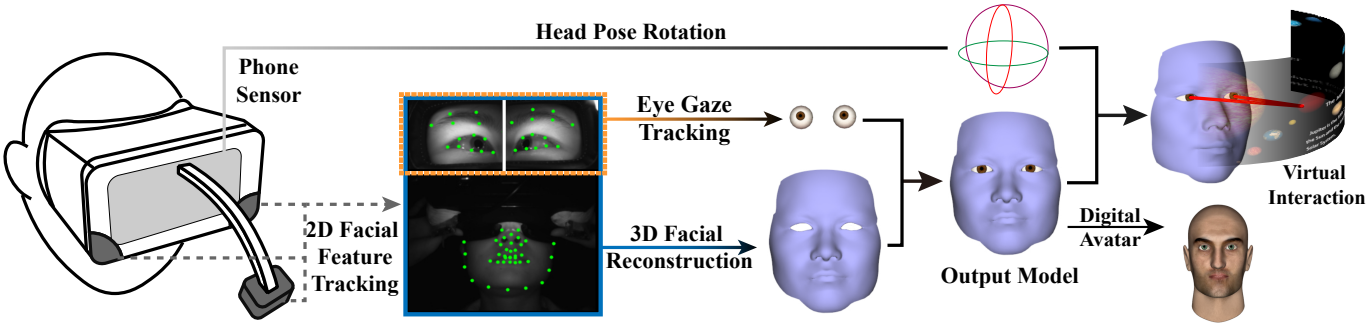
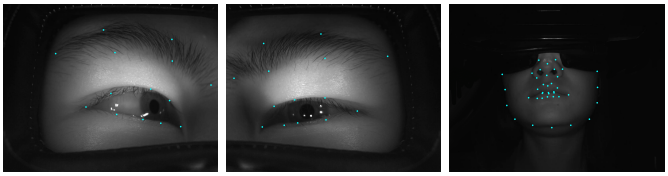


Fig. 2. The pipeline of the proposed method. The input of this system is the images captured by the cameras shown in Fig. 1. The feature points on the face and around the eye will be detected by the specified detector. Then the 3D face and eye gaze will be reconstructed according to the feature points and the image (more details are described in Sec. IV and Sec. V-C). After the gaze state is acquired, eye movements can be used to interact with virtual objects. The reconstructed 3D face can be used to drive the animation of a digital avatar.



(a) (b) (c)

Fig. 3. An example showing the three infrared images captured simultaneously from different views. Green dots are used to indicate feature points. (a) the right eye, (b) the left eye, (c) the face.

However, they either do not publish implementation details or need specific equipment. Thies et al. [55] and Song et al. [17] propose a novel system to reconstruct 3D facial animation and 3D eye gaze jointly. These two methods need to attach the ArUco AR markers on the HMDs, and put the camera in the front of the user to track the HMDs which limits the range of the users' movement. As mentioned in [17], ArUco AR markers may cause some flicking and failure when the user is moving. Compared with these works, our method can reconstruct the 3D face and eye gaze in the HMDs with unrestricted movement. A novel interactive method with the virtual environment is also developed based on the 3D eye gaze.

In this paper, we propose a system to estimate the 3D eye gaze of users wearing HMDs. Our method is robust to occlusion of faces and eye blinking. Moreover, we develop an approach to allow gaze-based interaction/selection which is able to cope with built-in lenses in HMDs, by using a short calibration step.

III. System Overview

To address the occlusion of the face by HMDs, and considering that VR content can be enjoyed when the environment is dark, we use three infrared cameras, two inserted in the HMDs for capturing eye images, and one for unoccluded facial movement, as shown in Fig. 1. Five infrared LED lamps are also fitted internally to each side of the device wall to provide uniform illumination. Camera locations are carefully chosen to avoid affecting the user's views and are embedded in the corners of the wall. Using this headset, the full face image can be divided into three parts. We then use detection models (Sec. IV-A) to get feature points that are used for multi-linear face reconstruction (Sec. IV-B). After that, we focus on eye gaze tracking based on eye images (Sec. V) to get the integrated model. Two interactive applications with VR-scenes are presented in Sec. VII.

In order to obtain the posture of the head, which is needed for animation, we use the sensor of a cell phone fitted to the HMD to get the orientation information. When wearing the device, the relative spatial relation between the phone and the head is fixed, and so the cell phone's orientation sensor provides the rotation information of the head. When the phone is placed into the headset, the coordinate system $\{x_p, y_p, z_p\}$ obtained by the phone sensor has a clear fixed correspondence with the coordinate system $\{x_f, y_f, z_f\}$ of the face model. Details can be found in the supplementary material. We use the OmniVision OV7675 image sensors which produce 640×480 infrared images at 30 fps. The three cameras are synchronized by activating each CCD's VSYNC pin simultaneously. The overall workflow of our method is summarized in Fig. 2.

IV. 3D Face Reconstruction

A. Detection of Feature Points

Traditional methods for facial feature point detection typically detect 2D feature points from the full face area on an image. Since we use three infrared cameras simultaneously capturing the images of different areas of the face, we independently detect feature points in the three images. Training images are required to locate such feature points robustly. Images in the existing facial databases are grayscale or color images with manual annotation of landmarks. IR images have different characteristics, and the IR images of the eyes are also captured from quite an unusual view. Due to the use of IR images and constraint of camera location, there are no such labeled facial and eye image datasets available, so we captured facial infrared images and labeled them manually. We captured IR images for 30 subjects and altogether manually labeled 6000 images of eyes and 3000 images of occluded faces. An example of the captured images along with the feature points are shown in Fig. 3. As Figs. 3(a)(b) show, 14 feature points are landmarked in each eye image, out of which 6 are on the eyebrow, and 8 are around the eye. In Fig. 3(c), 39 feature points are chosen on the face excluding the eyes and eyebrows. Our method thus has a total of 67 feature points.

We employ the method [56] to extract feature points, due to its efficiency, accuracy and reliability. Given the labeled training data, the method takes a set of triplets involving an input image, initial feature point positions, and an update vector from the landmarked datasets and learns a set of cascaded regression trees using gradient tree boosting, to optimize a loss function of sum of squared errors. At runtime, it determines the update vector for feature positions using

$$S^{t+1} = S^t + r_t(I, S^t), \tag{1}$$

where S is a vector containing feature point positions, t is the index of the cascaded regressor being used, S^t and S^{t+1} are feature point positions before and after applying the t-th regressor r_t, which predicts the update vector of S based on the current feature point positions S^t and the input image I.

In the training stage, we build three models for images from the three IR cameras, which detect a set of feature points from each view captured by the camera *cam* (*cam* = 1, 2, 3). Denote by p_k^{cam} the k-th feature point from the image of a given camera *cam*, and n_{cam} is the number of feature points in the view of camera *cam*. To put feature points extracted from individual images into a consistent coordinate system, we calibrate the three cameras using a multi-camera self-calibration method [57], which produces the intrinsic and extrinsic parameters of the cameras. Denote by P^{cam} the camera projection matrix associated with a given camera *cam*, $P^{cam} = K^{cam}[R^{cam}\ T^{cam}]$, where K^{cam} is the intrinsic parameter matrix of the camera and $[R^{cam}\ T^{cam}]$ is the extrinsic parameters (rotation and translation) that describe the transformation between the world coordinate system to the camera coordinate system.

B. 3D Reconstruction using Multilinear Fitting

After detecting the feature points from the three images, we reconstruct the 3D facial expression based on the detected feature points, using a multilinear model [58]. The multilinear model is trained using their large database containing 150 subjects, each with 47 expressions. It utilizes tensor decomposition to obtain the core tensor C_r which separates variation of identity with that of expression. A new 3D face can be synthesized by specifying the identity parameters m_{id} and the expression parameters m_{exp}:

$$M(m_{id}, m_{exp}) = C_r \times_2 m_{id}^T \times_3 m_{exp}^T, \tag{2}$$

where M is the obtained 3D model. We further select the same set of landmarks corresponding to the feature points in different camera views. Denote by M_k^{cam} the 3D point on the reconstructed face model M corresponding to the feature point p_k^{cam}. For each camera, the number of detected landmarks is denoted as n_{cam}. The reconstructed 3D face model may differ by a global rigid transformation (R,T) where R is the rotation matrix and T is the translation vector. We formulate the 3D facial expression reconstruction problem as optimizing model parameters (m_{id}, m_{exp}) and a global rigid transformation (R,T) such that the transformed 3D face model when projected to individual camera views has landmarks as close as possible to the detected feature points, i.e. minimizing the following energy:

$$\begin{aligned} &E_{projection}(m_{id}, m_{exp}, R, T) \\ &= \sum_{cam=1}^{3} \sum_{k=1}^{n_{cam}} \| P^{cam}(RM(m_{id}, m_{exp})_k^{cam} + T) - p_k^{cam} \|^2. \end{aligned} \tag{3}$$

This non-linear least squares problem can be solved efficiently by Google-Ceres [59]. The problem is optimized by iteratively alternating the following steps: 1) optimizing global transformation R,T; 2) optimizing identity parameters m_{id}, and 3) optimizing expression parameters m_{exp}. Since it is obvious that the same subject is being captured for a sequence, in the first few iterations both the identity parameters m_{id} and the expression parameters m_{exp} are solved simultaneously. After that, assuming the identity is correctly identified and kept unchanged, we keep m_{id} fixed, and only optimize m_{exp} for the remaining frames. This helps to further improve the efficiency of our method, achieving real-time performance (see Sec. VI).

To facilitate eye gaze tracking, we also specify the eyeball center for each model in the database, so when we reconstruct the 3D face expression, the eyeball center can also be obtained directly using the multilinear fitting. To achieve this, similar to [8], we fix the eyeball radius $\hat{r}$ to 12.5mm. In the training stage, for each model we calculate the average position of the eyelid vertices and offset the average position by $\hat{r}$ in the z-axis to obtain the initial eyeball center, which is then manually improved.

The 3D reconstruction obtained using the optimization above generally works well. However, it may produce slight jittering which may not be visually attractive. To address this, we further introduce a smoothness constraint for improved temporal coherence. Since the identity weight is fixed after the initial iterations, we only add constraints for the expression weight:

$$E_{smooth}^t = \| m_{exp}^t - m_{exp}^{t-1} \|^2, \tag{4}$$

where m_{exp}^t and m_{exp}^{t-1} are the current and previous identity weights. When optimizing m_{exp}^t, we instead minimize the overall energy:

$$E = E_{projection} + \lambda E_{smooth}^t, \tag{5}$$

where λ balances the importance of both terms, and we set $\lambda = 5$ in our experiments.

V. Eye Gaze Tracking and Focal Point Location

In this section we will first introduce our eye gaze tracking system in detail and then map our 3D eye gaze to image coordinates to get the location of the focal point. Eye gaze tracking aims to calculate the 3D eye gaze direction at each time instance, given two infrared eye images. Denote by V the eye gaze state, which is a triple:

$$V = (c, r_{iris}, d), \tag{6}$$

where $c = (c_x, c_y, c_z)$ expresses the position of the eye ball center, r_{iris} is the radius of the iris and pupil region, and the 3D pupil center indicates the eye gaze direction and is represented as $d = (\phi, \theta)$ using 3D spherical coordinates.

Algorithm 1 Calculating the initial eye state

Require: the position of eyeball center c,
the camera for eye area cam,
the projection matrix P_e of cam,
the captured image I_e,
the feature point p_e.

Ensure: the iris radius r_{iris}, the eye gaze direction d.

1: Locating 2D Iris Center as follows:
Connect p_e to get the eye mask $Mask_e$
Use the pixel classifier on I_e to obtain the probability map Map_e
Cluster the pixel in Map_e to get the center of iris o_{cen}

2: Estimating 3D Iris Radius as follows:
Extract edges from I_e using the Canny operator
Filter to get the filtered edge pixels $p_e \in \mathscr{E}$
Map p_e and o_{cen} to 3D to calculate the iris radius r_{iris}

3: Calculating eye gaze direction as follows:
Calculate the consistency of samples in multisacle sampling
Find the most consistent eye gaze direction d

4: return r_{iris}, d

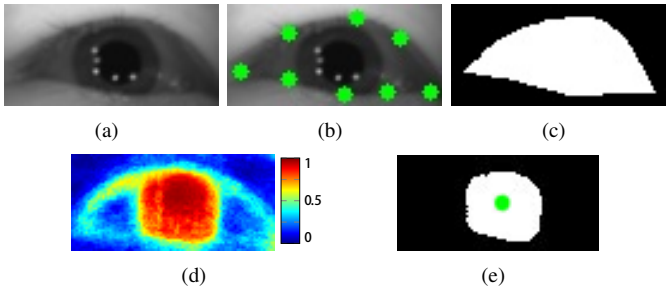
Since the size and location of eyeballs are different for each individual, we need to initialize those before eye gaze tracking. The center of the eyeball c is estimated by reconstructing the 3D facial position (see Sec. IV-B). In the initialization stage (described in Algorithm 1), we first locate the 2D iris center (Sec. V-A), and then estimate the radius r_{iris} of the iris in the 3D space (Sec. V-B). At runtime, we use multi-scale sampling and a robust correlation approach to work out the eye gaze direction d (Sec. V-C).

A. Locating 2D Iris Center

We use a method similar to [8] to locate the 2D iris center in the initialization stage. As shown in Fig. 4(a), the region of interest in the eye image is determined by the bounding box of the detected feature points (highlighted in Fig. 4(b)). We connect detected feature points to form a closed polygon which encloses the mask for the eye region (see Fig. 4(c)). For all the pixels in the eye region, we employ the pupil iris pixel classifier [8] to estimate the probability of a pixel belonging to the iris and pupil region (see Fig. 4(d)). The region is extracted using mean-shift clustering and the 2D center o_{cen} is shown as the green dot (see Fig. 4(e)).

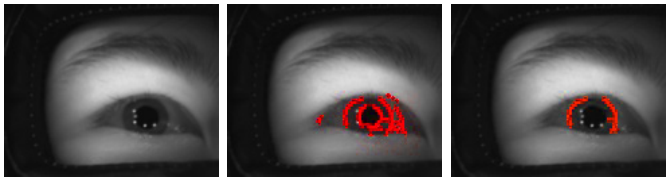
B. Estimating 3D Iris Radius

In the initialization stage, the radius r_{iris} of the iris also needs to be estimated, as it is subject dependent. To achieve this, we extract the edge map from the eye image. We employ the Canny edge detector to extract edges from images. However, simply using the Canny method produces many irrelevant edges. From Fig. 5(a), it can be observed that in the infrared eye image, the edges between iris and pupil are clearly present, which can easily mislead iris radius estimation. Two circles of edges exist (shown in Fig. 5(b)): one is the edge of the iris and the other is the edge of the pupil. However



(a) (b) (c) (d) (e)

Fig. 4. Steps to estimate the center of the iris. (a) the image captured by one IR camera; (b) the detected feature points are highlighted in green; (c) the eye mask that is defined by connecting the feature points; (d) the probability map calculated by the pixel classifier that detects the iris and pupil; (e) the iris and pupil regions where the green dot corresponds to the center of the iris.



(a) (b) (c)

Fig. 5. Steps to extract an iris edge map. (a) the original image captured by an IR camera, (b) the edge pixels extracted using the Canny operator, (c) the edge map after removing outliers both outside and inside the iris boundary.

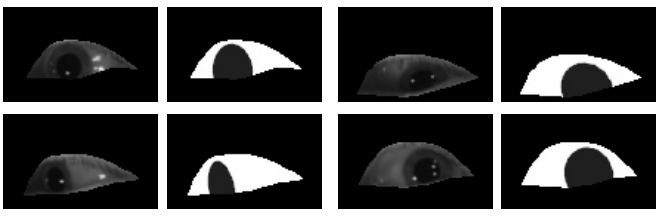
the pupil size changes as the light and focus change, making it difficult to track, whereas the size of the iris is fixed. To address this, we use the following two criteria to filter out irrelevant edges and only preserve edges corresponding to the iris boundary. First, we calculate the Euclidean distance between edge pixels p_e and the center of the iris o_{cen}, and only keep those edge pixels that satisfy:

$$t_1 H \leq ||p_e - o_{cen}||_2 \leq t_2 H, \tag{7}$$

where H is the height of the eye region, t_1 and t_2 are thresholds, set to 0.1 and 0.6 respectively in our experiments. Second, we further require edge pixels to be consistently oriented. Denote by ∇e the gradient direction at edge pixel p_e, then it needs to satisfy:

$$(p_e - o_{cen}) \cdot \nabla e \geq 0, \tag{8}$$

which means the angle between the gradient direction and the vector connecting the edge pixel to the center of the iris is no more than 90°. The filtered edges are shown in Fig. 5(c),



(a) (b) (c) (d)

Fig. 6. Projected eye images with high correlation to the captured IR images. (a)(c) the captured IR images, (b)(d) corresponding projected images with the eye region rendered in white and the iris region rendered in dark gray.

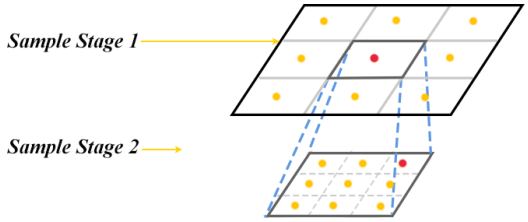


Fig. 7. Illustration of two-stage sampling for efficient eye gaze direction estimation. Firstly, at the coarse stage, the solution space of the eye rotation will be uniformly sampled with the 10×10 grid indicated at 'Sample Stage 1'. Then for the fine stage, the solution with the largest score will be re-sampled by a 3×3 sub-grid to refine the accuracy of the solution.

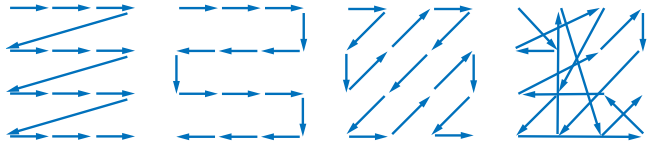


Fig. 8. Four different patterns of calibration paths (The fourth pattern is the random location). Taking the 4×4 grid as an example, the arrows indicate the display order of the vertices from previous one to next.

and the majority of irrelevant edge pixels have clearly been removed.

Once we obtain a set of filtered edge pixels $p_e \in \mathscr{E}$, we map them as well as the 2D iris center o_{cen} to 3D using the intrinsic parameters K^{cam} of the camera. The iris radius is calculated as the average distance between the edge pixels to the iris center in 3D space:

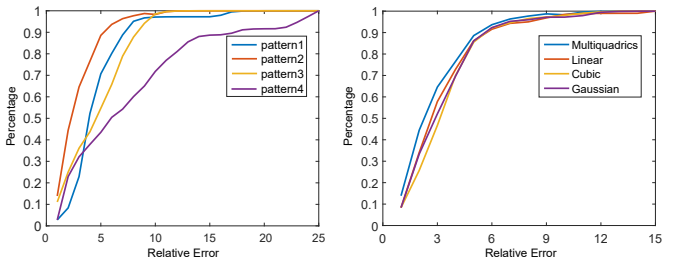
$$r_{iris} = \frac{1}{|\mathscr{E}|} \sum_{p_e \in \mathscr{E}} \| P_{c,\hat{r}} \left((K^{cam})^{-1} \tilde{p}_e \right) - P_{c,\hat{r}} \left((K^{cam})^{-1} \tilde{o}_{cen} \right) \|, \tag{9}$$

where $\tilde{p}_e$ and $\tilde{o}_{cen}$ are the homogeneous coordinates of p_e and o_{cen}, respectively. Given the eyeball with center c and radius $\hat{r}$, the intersection point of the eyeball and the line that connects the eyeball center and the viewpoint x can be calculated by the function $P_{c,\hat{r}}(x)$ as follows:

$$P_{c,\hat{r}}(x) = c + \hat{r} \frac{x - c}{\|x - c\|}. \tag{10}$$

C. Eye Gaze Tracking

In the previous section, to obtain the 3D iris radius, we use Eq. 10 to calculate the 3D location of the iris center in the eye ball and use it to get the initial 3D eye gaze direction. Since eye movement is mostly continuous, to work out the current eye gaze direction d_t, we uniformly sample directions $d \in \mathscr{D}$ around the previous direction d_{t-1}. For each sampled direction d, we project the 3D eyeball including the iris region (rendered using the calculated center position c and iris radius r_{iris}) to the camera's image space, and intersect the image with the eye region. Fig. 6 shows the projected images (the right image of each image pair), where the eye region is rendered in white, and the iris region is rendered in dark gray. Denote by I_d the pixels of the rendered image in the eye region with eye gaze direction d, and I_e the captured eye image, both considered



(a) (b)

Pattern	1	**2**	3	4
Error (%)	4.4435	**2.6048**	4.4979	7.4044

(c)

Methods	**Multiquadrics**	Bilinear	Bicubic	Gaussian
Error (%)	**2.6048**	2.7932	3.0812	2.9683

(d)

Fig. 9. Different relative error distributions calculated by Eq. 14. (a) The error distribution of focal point location using four patterns in Fig. 8(a), (b) the error distribution of four approximation functions with the same pattern(Pattern 2). (c) The average relative error for different patterns. (d) The average relative error for different approximation functions.

as long vectors. Then we calculate their consistency ρ using the Pearson correlation coefficient due to its robustness and efficiency

$$\rho(I_d, I_e) = \frac{cov(I_d, I_e)}{\sigma_{I_d} \sigma_{I_e}} = \frac{E[(I_d - \mu_{I_e})(I_e - \mu_{I_d})]}{\sigma_{I_d} \sigma_{I_e}}, \tag{11}$$

where μ and σ are the mean and standard deviation, cov is the covariance, and $E(\cdot)$ is the expectation. The eye gaze direction d_t is chosen as the direction that maximizes ρ:

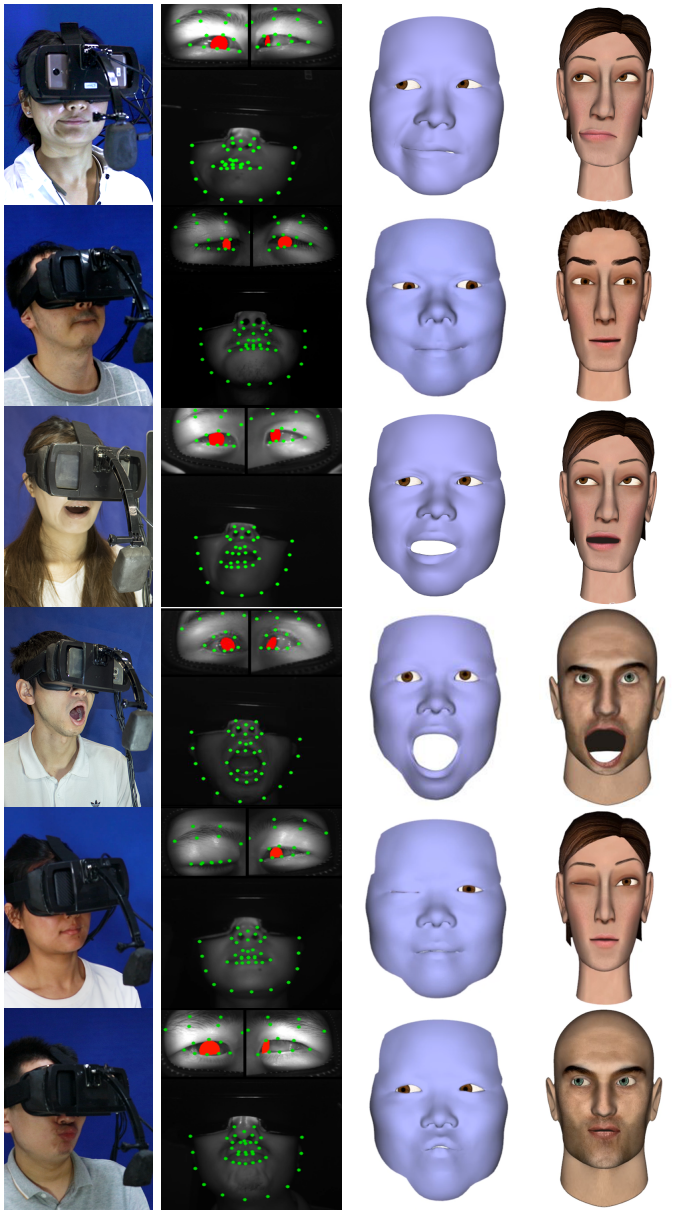
$$d_t = \arg\max_{d \in \mathscr{D}} \rho(I_d, I_e). \tag{12}$$

The projected image with the highest correlation ρ is selected as the tracking state. Some examples are shown in Fig. 6 where (a)(c) are the captured IR images of eyes and (b)(d) are projected images of the matched eye state.

Multiscale sampling. To further improve efficiency, we use a hierarchical sampling strategy for the eye gaze direction sampling. As illustrated in Fig. 7, the first layer is sampled sparsely and in the second layer dense samples are made around the first layer samples which have high correlation values. In our system, in coarse sample stage 1, the solution space is uniformly divided into a 10×10 grid. The center of each grid cell is rendered as an eye image and we use Eq. 11 to calculate its consistency. Then in fine sample stage 2, a fixed number of cells (we choose to use 5 cells) with the largest values will each be divided into a 3×3 sub-grid. Similar to stage 1, we calculate the scores and find the solution with the highest score. With this acceleration, it takes 4$\sim$8ms to detect eye gaze direction for a single frame.

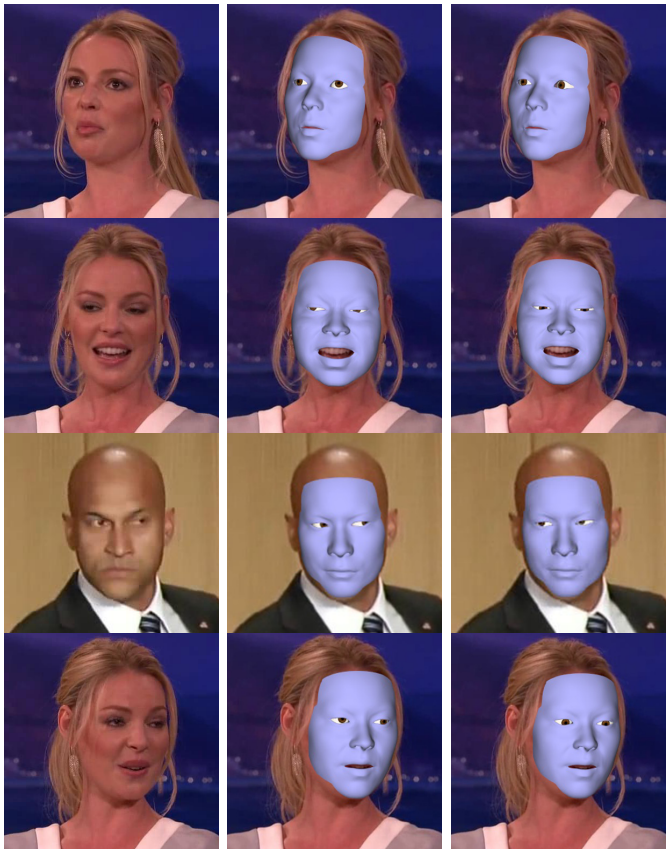
Failure detection and recovery. Sometimes eye gaze cannot be detected, typically because the eye is closed or blinks. We use a simple strategy to detect such failure cases. The eye gaze direction is rejected if $\rho(I_{d_t}, I_e) \leq 0$. A lower threshold is set here as the aim is to find failure cases such as closed eyes. To recover from such a failure state, in the follow-up frames, we increase the sampling area at the coarse level. This ensures eye gaze tracking recovers quickly from the failure situation.

Compared with state-of-the-art method [8], our eye gaze tracking method is simpler, and more robust. This is because



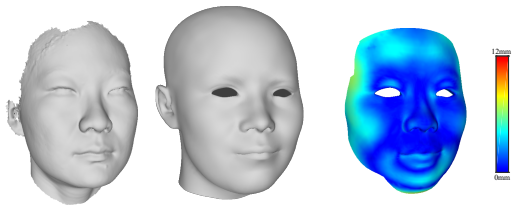
(a) (b) (c) (d)

Fig. 10. 3D facial expression reconstruction and eye gaze tracking. (a) the picture captured by an RGB camera, (b) the three captured IR images, (c) the reconstructed 3D face and eye gaze, (d) an avatar driven by the captured 3D face.



(a) (b) (c)

Fig. 11. Comparison with [8] for eye gaze tracking with complete face images as input. The same facial points are used in those two methods. (a) input images, (b) our results, (c) [8].



(a) (b) (c)

Fig. 12. Comparison of the 3D reconstruction and scan model. (a) is the model scanned by Artec EVA. (b) is the reconstruction result with our equipment. (c) visualizes reconstruction errors in the facial area. Mean and standard deviations of the geometry fitting errors are 1.7mm / 1.5mm.

we build a facial model for each user, use region based correlation and avoid using edge maps which are sensitive to noise. Wang et al. [8] also use the previous frames to heavily constrain the location of the next frame in a probabilistic framework. This approach will help to improve robustness, but causes visible delays when eye gaze moves rapidly and slows down recovery from tracking failure.

D. Focal Point Location

In addition to supporting eye contact with other users, eye gaze also provides a natural way of interacting with virtual objects. Existing VR systems use this idea, but restrict the "control point" to be the center of the view, and is therefore solely controlled by head movement rather than true eye gaze. Benefitting from the 3D eye tracking, we develop a technique to map 3D eye gaze to pixel location in the image, which can easily be mapped to the 3D position in the virtual world with the projection matrix for virtual scene rendering. Assuming the display in the HMDs is planar and without any additional lens in the line of sight, mapping from 3D gaze direction to the pixel location can be achieved by a projection transformation. However, this is not typically the case: many HMD products have a built-in lens for improved display quality and eyesight correction (e.g. for short sightedness), and the cell phone screen is often covered by protection glass which causes distortions. To cope with such diverse situations, we introduce a simple and effective calibration process adapted from stereo

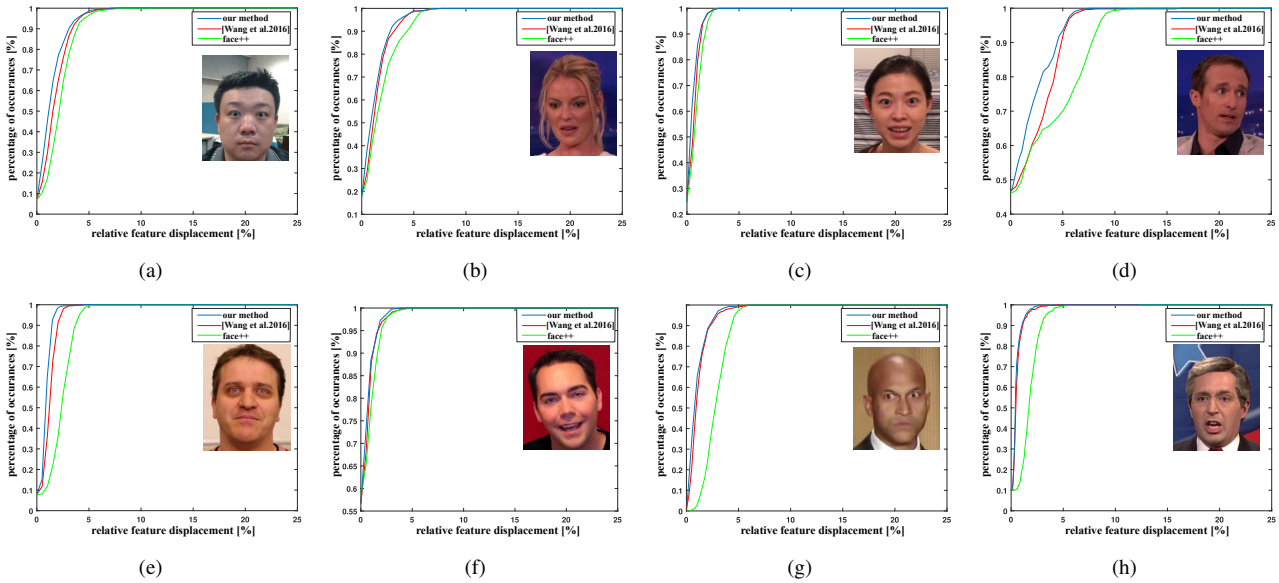


Fig. 13. Quantitative comparison with existing methods Wang et. al. [8] and Face++ tracker [60]. We take 8 videos for test and use Eq. 14 to calculate the relative error. The blue line: our method, the red line: Wang et. al. [8], the green line: Face++ tracker [60]. The statistics of these results are shown in Table I.

calibration which typically takes several seconds and is only needed for improved focal point location. The task of focal point location is to map a 3D eye gaze direction $d = (\phi, \theta)$ to a 2D pixel location $l = (x, y)$. For a simpler description, we focus on mapping the eye gaze of one eye when describing the method. In practice, we apply the same method to both eyes and use a simple strategy to fuse the responses from both eyes since they are normally consistent.

We now describe the calibration process. First, we render a small 3D ball which moves along a path to cover the 2D viewing space to get the 2D location of the ball center. Then the user is asked to look at the ball and follow it. At each time step, we obtain the pair of eye gaze direction $d_t(\phi_t, \theta_t)$ and the 2D location $l_t(x_t, y_t)$.

We then treat focal point location as fitting a non-linear function F that maps d_t to l_t with minimum error $E(F)$:

$$E(F) = \sum_t \|l_t - F(d_t)\|^2. \tag{13}$$

We performed user studies involving 10 participants to determine the suitable path along which the ball should move as well as the form of the approximating function F. The first user study aims to determine the suitable path of movement. We tried three typical patterns (Figs. 8(1∼3)) as well as random movement (as the 4th pattern Fig. 8(4)). To validate subjective preference, users were asked to score each pattern with a value in the range of 0 and 10 to show their comfort level where 0 is worst and 10 is best. Giving fine-grained scores in isolation can be difficult for the users. However, it is not a problem in our scenario, because users are essentially ranking 4 different patterns, and scoring gives them flexibility to specify relative differences between patterns. The average scores for those four patterns are 6.1, 8.8, 6.3, 4.0. It shows that Pattern 2 is clearly superior over the other patterns in terms of comfort.

To measure the accuracy of each method, we further work out the location error distribution of each pattern. After calibration, users are asked to look at randomly located 3D balls. The relative error is then defined as $e_l = \|l - \tilde{l}\|/L$ where l and $\tilde{l}$ are the ground truth and estimated locations, and L is the number of pixels in the longer side of the axis. We fix the approximation function to piecewise linear interpolation to compare different patterns. Ten different people participated in the testing in order to evaluate the accuracy of the proposed eye-gaze calibration method. For each person, the testing was carried out three times. Each time, once the person was wearing the HMDs, 25 uniformly sampled calibration points were displayed on the screen using the pattern 2 calibration path for the person to track and calibrate, and the 16 randomly sampled points were displayed one by one for the evaluation.

Figure 9(a) shows the curve with the proportion (percentage) of test 3D balls (y-axis) whose error is within a given threshold (x-axis, relative error in percentage). The table in Figure 9(c) shows the average relative errors. Pattern 2 performs best whereas Pattern 4 (random locations) performs worst, and therefore pattern 2 is chosen in our system.

For determining the suitable approximation function F, we evaluated 4 typical approximation functions: piecewise bilinear and bicubic interpolation based on the grid structure of the calibration points, as well as multiquadric Radial Basis Functions (RBFs) [61] and Gaussian RBFs. We performed similar user testing, and the curve of error distribution is shown in Fig. 9(b). Comparing the average relative errors listed in Fig. 9(d), multiquadric RBFs are most accurate and are selected for our system.

VI. Experimental Results

In this section, we evaluate our system in detail in terms of qualitative and quantitative evaluations, and compare our

TABLE I
STATISTICS FOR EXAMPLES IN FIG. 11. WE SHOW THE NUMBER OF FRAMES n_f AND AREA-UNDER-CURVE (AUC) VALUES FOR ALL THE METHODS. THE HIGHEST SCORES ARE IN BOLD.

Ex.	n_f	Face++	[Wang et al.]	Our method
(a)	911	0.907	0.926	**0.940**
(b)	398	0.927	0.943	**0.950**
(c)	438	0.966	0.971	**0.976**
(d)	730	0.870	0.911	**0.932**
(e)	845	0.903	0.948	**0.963**
(f)	600	0.970	0.975	**0.978**
(g)	626	0.876	0.944	**0.954**
(h)	528	0.924	0.973	**0.976**

method with the state-of-the-art method. More results, especially live demos, are provided in the accompanying video. This system runs on a computer with an Intel i7-6700 CPU and 24GB memory. The time cost of every frame is around 32.3ms, and achieves real-time performance with 30 FPS.

A. Qualitative Evaluation

As shown in Fig. 10, we evaluate our method in real-time performance. In addition, we project the iris area of the 3D eyeball onto captured images (Fig. 10(a)), as shown in Fig. 10(b), and also demonstrate the reconstructed 3D face (Fig. 10(c)) and driven avatar (Fig. 10(d)). Note that due to space restriction in the interior of HMDs, the infrared images for the eyes are taken from views at a non-frontal angle (see also the hardware setup in Fig. 1). Our method effectively recovers plausible 3D facial expressions and eye gaze for these examples with diverse expression and eye movement. We also evaluate our method in real scenes with applications such as interacting with virtual objects and virtual conferencing. The results are shown in Fig. 17 and Fig. 19 and more details are presented in the accompanying video.

Our eye-gaze tracking method is not only applicable to HMDs, but also can be used for RGB input. We compare with the state-of-the-art method [8] where complete images are used as input for both methods. The comparison results are shown in Fig. 11. Fig. 11(a) gives the original images, Fig. 11(b) is the result of our method and (c) is the result of [8]. The results have shown that our eye gaze tracking method is more accurate and robust, due to the use of region correlation rather than edge maps. Moreover, the edge maps are unreliable when the eye is nearly closed, as the edge and eye shape can confuse the system. Our method is robust in all of these situations, and effectively tracks fast eye movement.

B. Quantitative Evaluation

For 3D facial expression reconstruction, it is difficult to perform quantitative evaluation as the ground truth is unavailable. To address it, we collect the same facial expressions with and without HMDs. The groundtruth of the facial geometry is captured without HMDs by a 3D scanner: Artec EVA. The comparison results are shown in Fig. 12.

We compare our method with the state-of-the-art method [8] for eye gaze tracking. Similar to the qualitative evaluation, full face images are used to allow [8] to work. Since it is difficult to obtain the ground truth eye gaze direction in the 3D space, to visually show the comparison, we project the tracked 3D iris centers to 2D images, and compare them with manually labeled ground truth locations.

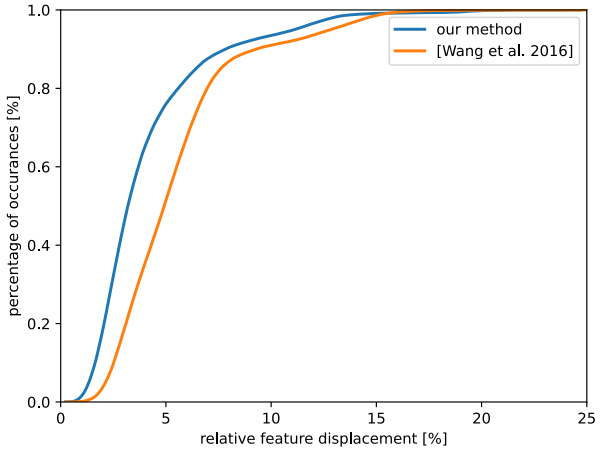


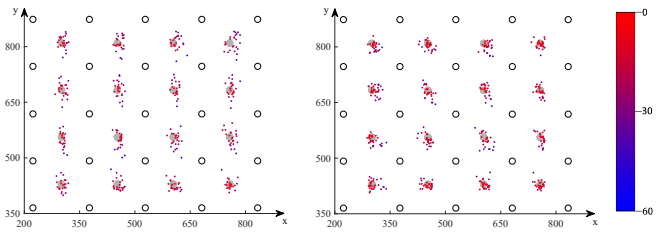
Fig. 14. Quantitative comparison with Wang et al. [8]. We take the test video in EVE dataset and use Eq. 14 to calculate the relative error. The blue line is our method and the orange one is Wang et al. [8].

Denote by $\tilde{c}_l$ and $\tilde{c}_r$ the ground truth left and right iris centers, and by c_l and c_r the tracked iris centers of the left and right eyes. The relative feature displacement $\tilde{e}$ is defined as:

$$\tilde{e} = \frac{\max(\|\tilde{c}_l - c_l\|, \|\tilde{c}_r - c_r\|)}{\|c_l - c_r\|}. \quad (14)$$

We take 8 videos, and compare our method with the state-of-the-art methods (Wang et al. [8] and Face++ tracker [60]). We record the percentage of cases with relative feature displacement e below a certain value, which forms a curve for each method and show the results in Fig. 13. Our eye-gaze tracking results are consistently better than Face++ and [8] for all the test videos. The statistics of these results are shown in Table I, including the number of frames for each video and the Area-Under-Curve (AUC) values as a summary. At the same time, we also compare with Wang et al. [8] on the EVE dataset [62]. In EVE dataset, face images are taken from an upward perspective and the eye tracking data is obtained by the commercial equipment Tobii Pro Spectrum. As stated in this database, synchronization between camera and eye-tracking data is only reliable on "basler" videos, so we compare on test data corresponding to these videos. We also use Eq. 14 to calculate the relative feature displacement. The curves for the percentage of cases with relative feature displacement e blow a certain value are shown in Fig. 14. Our method performs better than Wang et al. [8].

We also carry out comparison with a traditional pupil tracking method [63] which determines 3D eye-gaze state through fitting the 2D pupil contour. As it is hard to get the ground-truth of 3D eye gaze, we compare the eye gaze accuracy based on the estimated location in pixel of the image space. We create a grid of 5×5 points for calibration (black circles in Fig. 15) and a grid of 4×4 points for estimation (gray points in Fig. 15). In this experiment, the

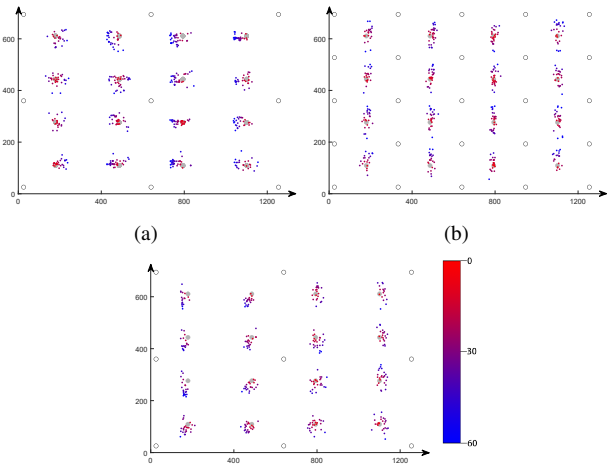


(a) (b)

	Mean Value	Standard Deviation
Method [63]	21.2520	12.1390
Our method	**13.1098**	**7.3411**

(c)

Fig. 15. The accuracy comparison of the traditional optimization method [63]. The calibration and test results in the comparison are shown in (a) and (b). (a) is the traditional optimization method [63]. (b) is our method. The black circles represent the calibration points, and the gray points are the test points. The remaining points represent the predicted positions, and the red to blue transition indicates the error from small to large. (c) statistics of of mean and standard deviation of errors in the estimated distance.



(a) (b) (c)

	Mean Value	Standard Deviation
WebGazer [64]	30.3799	16.8450
Ours in (3×3)	29.8597	15.6411
Our in (5×5)	**28.8672**	**15.7477**

(d)

Fig. 16. The accuracy comparison of our method and WebGazer [64]. The calibration and test results in the comparison are shown in (a), (b) and (c). (a) is the traditional optimization method [64]. (b) is our method in 3×3 calibration and (c) is in 5×5. The black circles represent the calibration points, and the gray points are the test points. The remaining points represent the predicted positions, and we use the red to blue transition to indicate the error from small to large. (d) statistics of the mean and standard deviation of errors in the estimated distance.

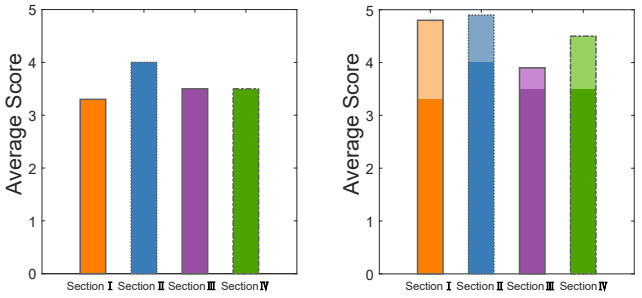
scene was displayed on an Android phone with a 5.1-inch screen of 1920×1080 resolution. Fig. 15 shows the estimation of focal point location denoted by different colors according to the Euclidean distance. Fig. 15(a) is the result of traditional optimization method [63] and (b) is our result. The mean and standard deviation of the Euclidean distance errors (in pixels) are presented in the table in Fig.15(c). We also compare with WebGazer [64], which is designed to predict the focal point on the screen when capturing the whole face in an RGB camera. As it is difficult to apply in our device, for comparison, we extend the gaze tracking method to the whole face. In this experiment, we compare with WebGazer [64] on a screen with a resolution of 1280×720 and place the camera in front. In the calibration stage of WebGazer, although they use 9 calibration points, for each point, the user has to look at and record 5 times. So a total of $9 \times 5 = 45$ samples have to be recorded during the calibration phase. In our method, we calibrate on 5×5 points, once for each point. In the experiment shown on Fig. 16 (a) (b) (c), We run WebGazer and calibrate our method on two calibrations, then perform the error calculation on the unified test point (4×4 gray points). (a) is the results of WebGazer [64], (b) is our method calibrated on 5×5 points and (c) is our method calibrated on 3×3 points (recording 5 times as WebGazer for each point), as shown by black circles. The comparison with WebGazer [64] and the mean and standard deviation of the Euclidean distance errors (in pixels) are presented in Fig. 16 (d). The calibration method we use has a larger coverage area, and it can be seen from the results that the accuracy rate is higher than WebGazer [64]. Even with the same calibration samples, our method works worse than our setting, but still better than WebGazer, demonstrating the advantage of our method.

VII. Applications

Our real-time reconstruction system can benefit many VR applications. Here, we apply this system to two major applications to demonstrate its effectiveness: a VR meeting and interaction with the virtual environment by eye gaze tracking.

For the VR conference meeting, it is important for the speaker to catch the reaction of the listener. In this application, both users can be equipped with our equipment and could be in different locations as shown in Fig. 17. When the speaker finds the listener getting puzzled by watching his or her reconstructed face and eye gaze, the speaker can change his or her speech pace based on the reaction of the listener's reconstructed face and eye gaze so as to make the listener better understand the intended meaning. A user study is proposed to evaluate the effectiveness of our system for the VR conference meeting. This user study is similar to the form of listening test in an English exam in which the speaker presents the instructions according to the reaction of the listener, while the listener is required to only listen to these instructions without talking with the speaker. Four sections of instructions are prepared for the test. For each listener, two of the sections will be selected randomly to be presented using our system, the other two sections will be presented by the traditional HMDs with only voice information. Each statement of the instructions takes about 2 minutes. After attending to the instructions from the speaker, each listener will be asked five questions, with each correct answer scoring one point. The contents of the instructions and questions are shown in the appendix. One speaker and five listeners participated in this user study. As shown in Fig. 18, with the help of our system, the listener achieved much better scores. We also calculate the

Fig. 17. A picture showing our system in action. Two users wearing HMDs are able to have immersive VR conversations even if they are at geographically distant locations. Their 3D face and eye gaze are captured and reconstructed in real time, and are then used to drive the animation of a 3D avatar shown on the VR display of the other subject in the conversation.



(a) (b)

ANOVA	Sec. I	Sec. II	Sec. III	Sec. IV
MSR	12.8	5.04167	1.5	6.01136
MSE	0.2611	0.44508	0.25	0.23636
F	49.0213	11.33	6	25.43
p-value	0.0000015	0.0028	0.02273	0.00006

(c)

Fig. 18. The average score of each section of questions described in the Appendix. (a) is with traditional HMDs and (b) is with the help of our system. It is obvious that the users achieve higher scores on all the sections compared with wearing traditional HMDs. (c) is the value of ANOVA.

value of Analysis of Variance (ANOVA) to compare our VR system with the tradition HMDs with only voice information, see Fig. 18(c). All the p-values are below 0.05, indicating that the differences between the two methods are statistically significant. This demonstrates that this system is much more helpful than traditional HMDs for exchanging information during VR conference meetings.

Another application is the interactivity with the virtual environments by the tracked eye gaze. Users wearing the HMD could move the focus point to select and pick up objects in the virtual environments by moving their eyes, as shown in Fig. 19. This example demonstrates an education scenario where users are able to find information about planets in the solar system. When the user looks at one of the planets, it is automatically chosen and highlighted in translucent red rendering. Additional information is then displayed in the bottom right corner. This provides a very natural and efficient way of interacting with the virtual environment.

VIII. Limitation

The scope of our system is limited in terms of the following issue. First, we use a camera exterior to the HMD that may make the equipment hard to place. Second, our method is based on a multi-linear model which cannot express high-quality facial details. One reason is that feature point detection is relatively sparse and cannot depict facial movement in detail. Another reason may be that the model in the database has a smoother surface and lacks details. Third, as the phone sensor was used to capture the head movement, the stability of sensor affects the display, and may cause some jitter in video.

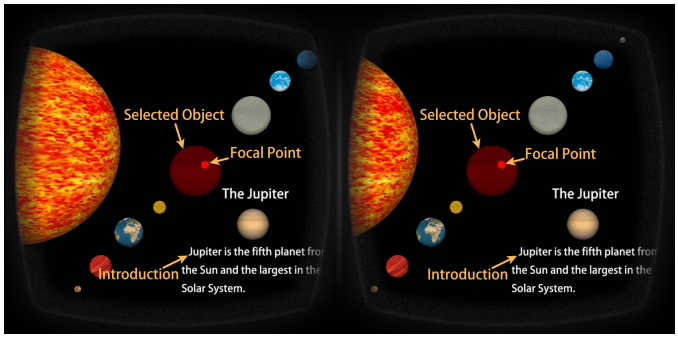


Fig. 19. Focal point location and its use for interaction with the VR scene. We show the images for the left and right eyes in a demo where users are able to obtain detailed information shown in the bottom right corner of the display using eye gaze. The focal point is shown as the solid red dot and the selected ball is rendered in translucent red.

IX. Conclusions

In this paper, we introduce a novel method that can robustly reconstruct 3D facial expressions and eye gaze in the HMDs in real time. This proposed system is easily assembled by off the shelf products including HMD headsets, infrared LED lights and cameras. Our device also captures the entire eyes and eyebrows, and uses all the information for accurate 3D face reconstruction. As shown in the experimental results sections, this algorithm performs well for new users who are not in the training dataset. With this new developed system, some potential applications in virtual reality would be benefited. Such as using eye gaze direction for improving the efficiency of scene rendering, as well as more general human-human interaction and human computer interaction in the VR settings e.g. by exploiting affective computing. We will investigate these in the future.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 62102403, No. 61872440 and No. 62061136007), the Beijing Municipal Natural Science Foundation for Distinguished Young Scholars (No. JQ21013), the Science and Technology Service Network Initiative, Chinese Academy of Sciences (No. KFJ-STS-QYZD-2021-11-001), the Youth Innovation Promotion Association CAS and Royal Society Newton Advanced Fellowship (No. NAF\R2\192151).

References

[1] M. Wang, X.-Q. Lyu, Y.-J. Li, and F.-L. Zhang, "VR content creation and exploration with deep learning: A survey," *Computational Visual Media*, vol. 6, pp. 3–28, 2020.

[2] S. Jörg, A. Duchowski, K. Krejtz, and A. Niedzielska, "Perceptual adjustment of eyeball rotation and pupil size jitter for virtual characters," *ACM Trans. Appl. Percept.*, vol. 15, no. 4, pp. 24:1–24:13, Oct. 2018. [Online]. Available: http://doi.acm.org/10.1145/3238302

[3] K. Ruhland, S. Andrist, J. B. Badler, C. E. Peters, N. I. Badler, M. Gleicher, B. Mutlu, and R. McDonnell, "Look me in the Eyes: A Survey of Eye and Gaze Animation for Virtual Agents and Artificial Systems," in *Eurographics 2014 - State of the Art Reports*, S. Lefebvre and M. Spagnuolo, Eds. The Eurographics Association, 2014.

[4] P. Garrido, L. Valgaert, C. Wu, and C. Theobalt, "Reconstructing Detailed Dynamic Face Geometry from Monocular Video," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 158:1–158:10, Nov. 2013. [Online]. Available: http://doi.acm.org/10.1145/2508363.2508380

[5] C. Cao, D. Bradley, K. Zhou, and T. Beeler, "Real-time High-fidelity Facial Performance Capture," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 46:1–46:9, Jul. 2015. [Online]. Available: http://doi.acm.org/10.1145/2766943

[6] 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, pp. 43:1–43:10, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2601097.2601204

[7] W. Simpson and S. Crandall, *The perception of smiles*. Psychonomic Society, US, 1972.

[8] C. Wang, F. Shi, S. Xia, and J. Chai, "Realtime 3D eye gaze animation using a single RGB camera," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 118:1–118:14, Jul. 2016. [Online]. Available: http://doi.acm.org/10.1145/2897824.2925947

[9] H. Li, L. Trutoiu, K. Olszewski, L. Wei, T. Trutna, P.-L. Hsieh, A. Nicholls, and C. Ma, "Facial Performance Sensing Head-mounted Display," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 47:1–47:9, Jul. 2015. [Online]. Available: http://doi.acm.org/10.1145/2766939

[10] K. Olszewski, J. J. Lim, S. Saito, and H. Li, "High-fidelity Facial and Speech Animation for VR HMDs," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 221:1–221:14, Nov. 2016. [Online]. Available: http://doi.acm.org/10.1145/2980179.2980252

[11] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, "High Resolution Passive Facial Performance Capture," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 41:1–41:10, Jul. 2010. [Online]. Available: http://doi.acm.org/10.1145/1778765.1778778

[12] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross, "High-quality Single-shot Capture of Facial Geometry," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 40:1–40:9, Jul. 2010. [Online]. Available: http://doi.acm.org/10.1145/1778765.1778777

[13] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross, "High-quality Passive Facial Performance Capture Using Anchor Frames," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 75:1–75:10, Jul. 2011. [Online]. Available: http://doi.acm.org/10.1145/2010324.1964970

[14] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt, "Lightweight Binocular Facial Performance Capture Under Uncontrolled Lighting," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 187:1–187:11, Nov. 2012. [Online]. Available: http://doi.acm.org/10.1145/2366145.2366206

[15] C. Wu, T. Shiratori, and Y. Sheikh, "Deep incremental learning for efficient high-fidelity face tracking," in *SIGGRAPH Asia 2018 Technical Papers*, ser. SIGGRAPH Asia '18. New York, NY, USA: ACM, 2018, pp. 234:1–234:12. [Online]. Available: http://doi.acm.org/10.1145/3272127.3275101

[16] H. Zhang, Q. Li, and Z. Sun, "Adversarial learning semantic volume for 2d/3d face shape regression in the wild," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4526–4540, 2019.

[17] G. Song, J. Cai, T.-J. Cham, J. Zheng, J. Zhang, and H. Fuchs, "Real-time 3d face-eye performance capture of a person wearing vr headset," in *Proceedings of the 26th ACM International Conference on Multimedia*, ser. MM '18. ACM, 2018, pp. 923–931.

[18] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "FaceWarehouse: A 3D Facial Expression Database for Visual Computing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 413–425, Mar. 2014. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2013.249

[19] S. G. Kong and R. O. Mbouna, "Head pose estimation from a 2d face image using 3d face morphing with depth parameters," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1801–1808, 2015.

[20] P.-L. Hsieh, C. Ma, J. Yu, and H. Li, "Unconstrained realtime facial performance capture," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1675–1683.

[21] S. Saito, T. Li, and H. Li, "Real-time facial segmentation and performance capture from RGB input," in *European Conference on Computer Vision*, 2016, pp. 244–261.

[22] X. Tu, J. Zhao, M. Xie, Z. Jiang, A. Balamurugan, Y. Luo, Y. Zhao, L. He, Z. Ma, and J. Feng, "3D face reconstruction from a single image assisted by 2D face images in the wild," *IEEE Transactions on Multimedia*, vol. 23, pp. 1160–1172, 2021.

[23] X. Wen, M. Wang, C. Richardt, Z.-Y. Chen, and S.-M. Hu, "Photorealistic audio-driven video portraits," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 12, pp. 3457–3466, 2020.

[24] X. Chai, J. Chen, C. Liang, D. Xu, and C.-W. Lin, "Expression-aware face reconstruction via a dual-stream network," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.

[25] X. Fan, S. Cheng, K. Huyan, M. Hou, R. Liu, and Z. Luo, "Dual neural networks coupling data regression with explicit priors for monocular 3D face reconstruction," *IEEE Transactions on Multimedia*, pp. 1–1, 2020.

[26] B. Chaudhuri, N. Vesdapunt, L. Shapiro, and B. Wang, "Personalized face modeling for improved face reconstruction and motion retargeting," in *IEEE European Conference on Computer Vision (ECCV)*, 2020.

[27] Y. Feng, H. Feng, M. J. Black, and T. Bolkart, "Learning an animatable detailed 3D face model from in-the-wild images," vol. 40, no. 8, 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459936

[28] X. Wang, K. Liu, and X. Qian, "A survey on gaze estimation," in *International Conference on Intelligent Systems and Knowledge Engineering*, 2015, pp. 260–267.

[29] 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. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2009.30

[30] A. Kar and P. Corcoran, "A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms," *CoRR*, vol. abs/1708.01817, 2017. [Online]. Available: http://arxiv.org/abs/1708.01817

[31] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '14. New York, NY, USA: ACM, 2014, pp. 207–210. [Online]. Available: http://doi.acm.org/10.1145/2578153.2578185

[32] S. Tripathi and B. Guenter, "A Statistical Approach to Continuous Self-Calibrating Eye Gaze Tracking for Head-Mounted Virtual Reality Systems," in *IEEE Winter Conference on Applications of Computer Vision*, 2017, pp. 862–870.

[33] J. Chen and Q. Ji, "A probabilistic approach to online eye gaze tracking without explicit personal calibration," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1076–1086, 2015.

[34] Y. Sugano and A. Bulling, "Self-Calibrating Head-Mounted Eye Trackers Using Egocentric Visual Saliency," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 2015, pp. 363–372. [Online]. Available: http://doi.acm.org/10.1145/2807442.2807445

[35] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "A 3D Morphable Eye Region Model for Gaze Estimation," in *European Conference on Computer Vision*, 2016, pp. 297–313.

[36] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-Synthesis for Appearance-Based 3D Gaze Estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1821–1828. [Online]. Available: https://doi.org/10.1109/CVPR.2014.235

[37] C. H. Morimoto and M. Flickner, "Real-time multiple face detection using active illumination," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 8–13.

[38] H. Wu, Y. Kitagawa, T. Wada, T. Kato, and Q. Chen, "Tracking iris contour with a 3D eye-model for gaze estimation," in *Proceedings of the 8th Asian Conference on Computer Vision*, 2007, pp. 688–697.

[39] K. Dierkes, M. Kassner, and A. Bulling, "A novel approach to single camera, glint-free 3d eye model fitting including corneal refraction," in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & amp; Applications*, ser. ETRA '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3204493.3204525

[40] Y. Cheng, X. Zhang, F. Lu, and Y. Sato, "Gaze estimation by exploring two-eye asymmetry," *IEEE Transactions on Image Processing*, vol. 29, pp. 5259–5272, 2020.

[41] F. Lu, X. Chen, and Y. Sato, "Appearance-based gaze estimation via uncalibrated gaze pattern recovery," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1543–1553, 2017.

[42] K. Pfeuffer, M. Vidal, J. Turner, A. Bulling, and H. Gellersen, "Pursuit calibration: Making gaze calibration less tedious and more

flexible," in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '13. New York, NY, USA: ACM, 2013, pp. 261–270. [Online]. Available: http://doi.acm.org/10.1145/2501988.2501998
[43] M. Vidal, A. Bulling, and H. Gellersen, "Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '13. New York, NY, USA: ACM, 2013, pp. 439–448. [Online]. Available: http://doi.acm.org/10.1145/2493432.2493477
[44] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face Alignment by Explicit Shape Regression," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2887–2894.
[45] C. Hennessey, "Point-of-gaze estimation in three dimensions," Ph.D. dissertation, University of British Columbia, 2008. [Online]. Available: https://open.library.ubc.ca/cIRcle/collections/24/items/1.0066824
[46] E. Gutierrez Mlot, H. Bahmani, S. Wahl, and E. Kasneci, "3D Gaze Estimation Using Eye Vergence," in *Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies*, 2016, pp. 125–131. [Online]. Available: https://doi.org/10.5220/0005821201250131
[47] J. Lou, Y. Wang, C. Nduka, M. Hamedi, I. Mavridou, F.-Y. Wang, and H. Yu, "Realistic facial expression reconstruction for VR HMD users," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 730–743, 2020.
[48] S.-E. Wei, J. Saragih, T. Simon, A. W. Harley, S. Lombardi, M. Perdoch, A. Hypes, D. Wang, H. Badino, and Y. Sheikh, "Vr facial animation via multiview image translation," *ACM Trans. Graph.*, vol. 38, no. 4, Jul. 2019.
[49] S. Lombardi, J. Saragih, T. Simon, and Y. Sheikh, "Deep appearance models for face rendering," *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201401
[50] Y. Zhao, Q. Xu, X. Huang, and R. Yang, "Mask-off: Synthesizing face images in the presence of head-mounted displays," *CoRR*, vol. abs/1610.08481, 2016. [Online]. Available: http://arxiv.org/abs/1610.08481
[51] J. Rekimoto, K. Uragaki, and K. Yamada, "Behind-the-mask: A face-through head-mounted display," in *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, ser. AVI '18. New York, NY, USA: ACM, 2018, pp. 32:1–32:5. [Online]. Available: http://doi.acm.org/10.1145/3206505.3206544
[52] S. Lombardi, J. Saragih, T. Simon, and Y. Sheikh, "Deep appearance models for face rendering," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 68:1–68:13, Jul. 2018. [Online]. Available: http://doi.acm.org/10.1145/3197517.3201401
[53] "Tobii eyecore," https://www.tobiipro.com/, 2017.
[54] "Pupil labs," https://pupil-labs.com/, 2017.
[55] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "FaceVR: Real-time gaze-aware facial reenactment in virtual reality," *ACM Trans. Graph.*, vol. 37, no. 2, pp. 25:1–25:15, Jun. 2018. [Online]. Available: http://doi.acm.org/10.1145/3182644
[56] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874. [Online]. Available: https://doi.org/10.1109/CVPR.2014.241
[57] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multicamera self-calibration for virtual environments," *Presence: Teleoper. Virtual Environ.*, vol. 14, no. 4, pp. 407–422, Aug. 2005. [Online]. Available: http://dx.doi.org/10.1162/105474605774785325
[58] C. Cao, Y. Weng, S. Lin, and K. Zhou, "3D Shape Regression for Real-time Facial Animation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 41:1–41:10, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2461912.2462012
[59] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org, 2017.
[60] M. Technology, "Face++," 2015. [Online]. Available: http://www.faceplusplus.com.cn
[61] R. L. Hardy, "Multiquadric equations of topography and other irregular surfaces," *Journal of Geophysical Research*, vol. 76, no. 8, pp. 1905–1915, Mar. 1971.
[62] S. Park, E. Aksan, X. Zhang, and O. Hilliges, "Towards end-to-end video-based eye-tracking," in *European Conference on Computer Vision (ECCV)*, 2020.
[63] "3D pupil tracking," https://github.com/YutaItoh/3D-Eye-Tracker, 2017.
[64] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays, "WebGazer: Scalable webcam eye tracking using user interactions," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16)*. AAAI, 2016, pp. 3839–3845.

Shu-Yu Chen received the PHD degree in computer science and technology from University of Chinese Academy of Sciences. She is currently working as a research associate in Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include computer graphics.

Yu-Kun Lai received his bachelor's and Ph.D. degrees in computer science from Tsinghua University, China, in 2003 and 2008, respectively. He is currently a Professor at the School of Computer Science & Informatics, Cardiff University. His research interests include Computer Graphics, Computer Vision, Geometry Processing and Image Processing. He is on the editorial boards of *Computer Graphics Forum* and *The Visual Computer*.

Shihong Xia is a professor associated with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences. He received the BS degree in Mathematics from the Sichuan Normal University, China in 1996 and the PhD degree in Computer Software and Theory from University of Chinese Academy of Sciences in 2002. His research interests include computer graphics, virtual reality and artificial intelligence.

Paul L. Rosin is a Professor at the School of Computer Science & Informatics, Cardiff University. Previous posts were at Brunel University, Joint Research Centre (Italy), and Curtin University of Technology (Australia). His research interests include the representation, segmentation, and grouping of curves, knowledge-based vision systems, early image representations, low level image processing, machine vision approaches to remote sensing, methods for evaluation of approximations, algorithms, etc., medical and biological image analysis, mesh processing, and the analysis of shape in art and architecture.

Lin Gao received his PhD degree in computer science from Tsinghua University. He is currently an Associate Professor at the Institute of Computing Technology, Chinese Academy of Sciences. He has been awarded the Newton Advanced Fellowship from the Royal Society and the AG young researcher award. His research interests include computer graphics and geometric processing.