Real-time 3D Face-Eye Performance Capture of a Person Wearing VR Headset

Guoxian Song
Nanyang Technological University
guoxian001@e.ntu.edu.sg

Jianfei Cai
Nanyang Technological University
asjfcai@ntu.edu.sg

Tat-Jen Cham
Nanyang Technological University
astjcham@ntu.edu.sg

Jianmin Zheng
Nanyang Technological University
ASJMZheng@ntu.edu.sg

Juyong Zhang*
University of Science and Technology of China
juyong@ustc.edu.cn

Henry Fuchs
University of North Carolina at Chapel Hill
fuchs@cs.unc.edu

ABSTRACT

Teleconference or telepresence based on virtual reality (VR) head-mount display (HMD) device is a very interesting and promising application since HMD can provide immersive feelings for users. However, in order to facilitate face-to-face communications for HMD users, real-time 3D facial performance capture of a person wearing HMD is needed, which is a very challenging task due to the large occlusion caused by HMD. The existing limited solutions are very complex either in setting or in approach as well as lacking the performance capture of 3D eye gaze movement. In this paper, we propose a convolutional neural network (CNN) based solution for real-time 3D face-eye performance capture of HMD users without complex modification to devices. To address the issue of lacking training data, we generate massive pairs of HMD face-label dataset by data synthesis as well as collecting VR-IR eye dataset from multiple subjects. Then, we train a dense-fitting network for facial region and an eye gaze network to regress 3D eye model parameters. Extensive experimental results demonstrate that our system can efficiently and effectively produce in real time a vivid personalized 3D avatar with the correct identity, pose, expression and eye motion corresponding to the HMD user.

CCS CONCEPTS

• **Computing methodologies** → Motion capture;

KEYWORDS

3D facial reconstruction; gaze estimation; HMDs

ACM Reference Format:
Guoxian Song, Jianfei Cai, Tat-Jen Cham, Jianmin Zheng, Juyong Zhang, and Henry Fuchs. 2018. Real-time 3D Face-Eye Performance Capture of a Person Wearing VR Headset . In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240570

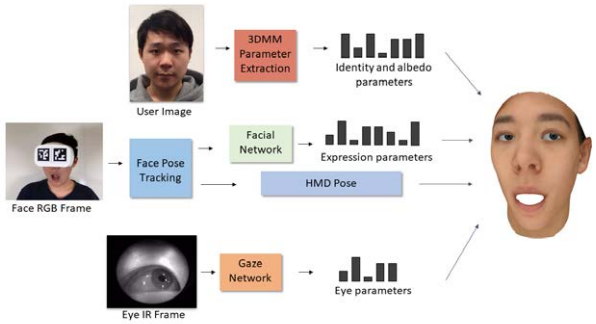

Figure 1: An overview of our system. 3DMM Parameter Extraction module: to extract the personal 3DMM identity and albedo information given one frontal face image, which only needs to be done once and offline. Face Pose Tracking module: track face pose and locate face region based on ArUco Marker detection [9]. Facial Network module: given the cropped face region with HMD, to regress the 3DMM expression parameters. Eye Network module: given an eye IR image, to regress the 3D eye model parameters.

1 INTRODUCTION

This paper considers the problem for only using a commodity RGB camera to do real-time 3D facial performance capture of a person wearing a virtual reality (VR) head-mount display (HMD). Although 3D facial performance capture alone is a well studied problem [5, 8, 16, 27], which aims to track and reconstruct 3D faces from 2D images, performance capture of faces with HMDs is a new problem and it only emerged recently due to the increasing popularity of HMDs such as Oculus [19] and HTC Vive [12] and rapid development of various VR applications. One particular application we consider here is teleconference or telepresence, where two or multiple users wearing VR HMDs can have immersive face-to-face communications in a virtual environment.

However, such a task is extremely challenging, since more than 60 percent face including all upper face landmarks are occluded by HMD (see the input image in Fig. 1). The largely occluded faces

*The Corresponding author.

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240570

in HMD scenarios make the face performance capture problem intractable for almost all the existing landmark based methods [10, 16, 27], which explicitly leverage on the tracked landmarks to locate the face and recover the full facial motion,

Fortunately, there have been are several advances on this challenging problem. For example, Li et al. [16] integrated RGB-D and strain sensors into VR HMD for real-time facial expressions animation, but it requires special hardware setup and tedious calibration for each step, and it cannot automatically digitize a personalized 3D avatar. Thies [28] et al. proposed 2D facial reenactment framework for HMD users using RGB-D camera and carefully optimized parallel computing, which is fairly complex. Unfortunately, none of those efforts included the performance capture of 3D eye gaze movement, which limits their performance in producing compelling digital presence. Note that [28] only classifies eye gaze information to retrieve the corresponding 2D eye image regions in database for producing non-occluded face image by image composition, and it cannot produce a personalized avatar with 3D eye gaze movement. In summary, the existing limited solutions are very complex either in setting or in approach and lack the performance capture of 3D eye gaze movement.

This motivates us to look for a better solution that should be simple in setting, less complex in approach and take into account 3D eye gaze movement. Recently, Guo et al. [10] proposed a convolutional neural network (CNN) based learning framework for real-time dense facial performance capture, based on the well-known 3D morphable model (3DMM) representation, which encodes a 3D face into three sets of parameters: facial identity, expression and albedo. Leveraged on the large-scale training data, [10] can train a powerful network to regress the 3DMM parameters as well as pose and lighting parameters in real time given RGB inputs.

This triggers us to propose a CNN based solution with 3DMM representation for 3D face-eye performance capture of HMD users. Particularly, we develop a facial network to predict facial parameters and an eye gaze network to predict the eye motion information. However, there are three challenges we face here. First, there is no public dataset available that contains HMD facial images and their corresponding 3DMM parameter labels. Note that [20] also uses CNN to regress facial parameters, but it needs professional artists to assist on generating labelled training data, which is very costly and time-consuming. Second, given limited observation on face from the input RGB image, it is hard for a network to learn and predict all the 3DMM and pose parameters. Third, there is no training data available for learning a 3D eye gaze network. For the first challenge, we propose to use a data synthesis approach to solve it. Particularly, we synthesize images of face wearing HMD from non-occluded face images. For the second challenge, we adopt a divide-and-conquer strategy. Particularly, for an HMD user, we can obtain its 3DMM identity and albeo parameters in advance from a single full face RGB image, which only needs to be done offline once. During online operation, we use ArUco Marker [9] to track face pose and the facial network only needs to regress the expression parameters from the input HMD image. For the third challenge, we make use of the existing eye tracking techniques, which have been integrated into many commercial HMD products such as [7, 24, 29]. The existing eye tracking-HMDs can provide the captured IR eye images and the corresponding eye gaze information, which will be used to train our eye gaze network to regress the 3D eye model parameters. Fig. 1 gives an overview of the proposed system.

The main contributions of this paper are threefold.

- We construct an HMD face dataset that contains synthetic HMD-occluded facial images with 3DMM expression labels (extracted from full face observations).
- We construct a VR IR eye dataset with various gaze angles. We intend to release both datasets for research purposes. To the best of our knowledge, there is no such dataset currently available in public.
- We develop a face-eye performance capture system that integrates the 3D parametric face-eye models. Extensive experimental results demonstrate that our system can efficiently and effectively produce in real time a vivid personalized 3D avatar with the correct identity, pose, expression and eye motion corresponding to the HMD user.

2 RELATED WORK

In this section we review the existing methods on 3D face and eye reconstruction and the facial performance capturing methods on HMD faces, which are closely related to this research, as well as the developments of HMD devices.

3D face and eye reconstruction. In the context of 3D facial reconstruction, early works focus on the optimization to minimize the difference between the input RGB images and the reconstructed model appearances. 3D morphable model (3DMM) is the most well-known parametric 3D face model, which disentangles a 3D face into three sets of parameters: facial identity, expression and albedo. Oswald et al. [1] proposed a complete framework for face inverse rendering with 3DMM. Recently, with the development of deep learning technology, Guo et al. [10] presented a coarse-to-fine CNN framework for real-time dense textured 3D face reconstruction based on 3DMM with RGB inputs. For unconstrained images (often with occlusions), Saito et al. [22] regressed the 3D face parameters from the non-occluded facial part via semantic segmentation. Yu et al. [33] presented a neural network for dense facial correspondences in highly unconstrained RGB images. There are also some 3D face reconstruction works based on RGB-D inputs with depth sensors like Kinect and Prime Sense, which provide aligned depth and colour streams. For example, Tan et al. [26] introduced a cost-efficient framework to align point clouds from two depth cameras for high quality 3D facial reconstruction in real time. All these 3D face reconstruction methods cannot handle the HMD face images where the face is largely occluded by HMD.

For eye reconstruction, the eye ball is a complex organ comprised of multiple layers of tissues with different transparency and reflectance properties. Wood et at. [32] presented UnityEyes for rapid synthetic eye data generation based on high resolution 3D face scan and complex eyeball materials. Recently, Wood et at. [31] proposed a 3D morphable model for eye region reconstruction from a RGB image.

HMD devices. For over two decades, wearable HMDs are widely used in various entertainment and education applications. Foreseeing the tremendous opportunities, eye tracking techniques have recently been integrated into commercial HMD products such as

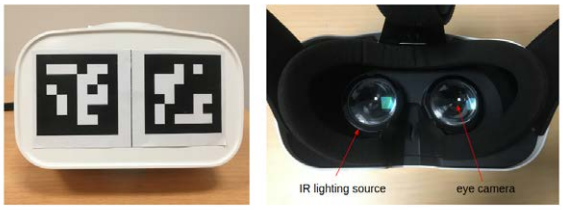

Figure 2: VR HMD device with ArUco markers and eye IR camera.

[7, 24, 29]. For example, a Japanese startup company Fove [7] embedded infrared cameras to track eye movements and released the first commercial eye-tracking VR HMD. Soon after that, Tobii, the largest eye-tracking company collaborated with the well-know VR company HTC Vive [12] and developed eye-tracking VR HMDs Tobii Pro [29]. Meanwhile, on the research side, there is a strong interest in reconstructing the VR rendering environment involved with the users wearing HMD. Very recently, a team of researchers at Google company started a headset removal project, which aims to remove the large occlusion regions caused by the HMD devices, for virtual and mixed reality based on HTC Vive incorporated with SMI [24] eye-tracking technology [4]. Besides, Pupil Labs [15] has developed open source software and accessible eye tracking hardware for existing HMDs, like Oculus [19] and Hololens [18], etc. All together, there efforts make eye tracking HMDs production and distribution much easier and more prevalent than ever. Our system uses Fove 0 HMD (see Fig. 2) to map video recordings of face and eyes to a personalized 3D face-eye model.

Facial performance capture of faces with HMD. There are only a few recent works trying to attack the problem of facial performance capture of HMD faces. Li et al. [16] proposed the first VR HMD system for fully immersive face-to-face telepresence. They attached an external depth sensor to HMD to capture the visible facial geometry while wearing HMD, and they estimated facial expression of the occluded regions by measuring the signals of electronic strain sensors attached to the inside of HMD. The resultant expression coefficients are then used to animate virtual avatars. Later on, Olszewski et al. [20] proposed an approach for an HMD user to real-time control a digital avatar, which is manually created by professional artists. Similarly, they mounted a custom RGB camera for mouth tracking. In addition, a convolutional neural network is used to regress from mouth and eye images to the parameters that control a digital avatar. While the system can provide high-fidelity facial and speech animation, the system is limited to specifically created avatars by professional artists and does not consider the eye gaze movement which is important for vivid presences. Thies [28] proposed a 2D facial reenactment framework for HMD users with RGB-D camera. Their method is an optimization based solution, carefully designed for parallel computing in order to achieve real-time performance, which is highly computational complex.

The differences between our approach and the existing methods mainly lie in three aspects: setting, approach, and eye motion capture. In terms of setting, [20] requires the most complex setting, including mounting strain sensor and RGB-D camera on HMD and the tedious calibration process for each subject. [20] also requires mounting RGB camera on HMD. [28] needs an external RGB-D camera to capture the face of an HMD user. In contrast, our system only needs an external RGB webcam without any calibration, which is the simplest one in setting. In terms of approach, both [20] and [28] are optimization based solutions, which require intensive computation, while our solution is based on CNN and it only needs one forward pass during online operation, which is much faster. Although [20] also uses CNN to regress facial parameters, they need professional artists to assist on generating labelled training data. On the contrary, we use data synthesis for training data generation, which does not need any human intervention. In terms of eye motion capture, none of the existing works capture 3D eye gaze movement. Although [28] takes into the eye gaze information, the classified eye gaze information is only used to retrieve the corresponding 2D eye image regions for the subsequent image composition. In contrast, our system produces a personalized avatar with 3D eye gaze movement.

3 OVERVIEW

Figure 1 shows the overall diagram of our proposed 3D facial performance capturing system for a person wearing HMD. The entire system consists of three paths located from top to bottom, respectively, as shown in Figure 1. The top path is to extract the personal 3DMM identity and albedo information given one frontal face image, using the inverse rendering optimization approach [10], which only needs to be done once and offline. During the online operation, at each time instance, the middle path takes in a face RGB frame and regresses the corresponding 3DMM face expression parameters, and the bottom path takes in an eye IR image frame and regresses the corresponding eye model parameters. The final output of our system is a personalized 3D avatar with the corresponding face pose, face expression and eye gaze. Note that the face pose is directly obtained by the Face Pose Tracking module based on ArUco Marker detection [9], which is also used to locate and crop face region so as to align the input face images into the facial network. Overall, the key components of our system are the facial network and the eye network, which respectively regress 3DMM facial expression parameters and 3D eye model parameters in real time. In the next sections, we will describe how to train these two networks in detail.

Particularly, our system is based on the FOVE 0 VR HMD, with integrated eye tracking cameras and custom ArUco markers for face tracking. Each of the infrared (IR) cameras inside HMD is with six 940nm wavelength IR LEDs, by which user's eyes can be observed clearly without the need of ambient illumination. It can record 320x240 images of user's eyes at 60 fps. The user's face is recorded by a commodity RGB WebCam at 30fps with a resolution of 1080x720. Our system is much more convenient than that of Olszewski's [20], which needs external camera and lighting source equipment attached to the HMD.

4 FACIAL REGRESSION NETWORK

To train the facial expression regression network, the main challenge is that there is no public HMD facial dataset available and

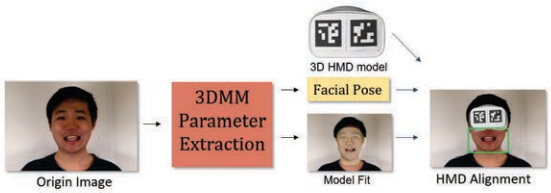

Figure 3: The pipeline for generating facial network training data.

it is extremely difficult to obtain the labels of the ground truth expression parameters with highly occluded faces. [20] requires professional animators to assist on producing expression labels, which is costly and time-consuming. In contrast, we use data synthesis to produce pairs of HMD faces and labels, which does not require any human annotation. In the following, we first introduce the 3D parametric face model we adopt, and then describe our data synthesis and network structure in detail.

4.1 Parametric 3D Face Model

We use 3D Morphable Model (3DMM) [3] as the parametric face model to encode 3D face geometry and albedo. Specifically, 3DMM describes 3D face shape geometry S and color albedo C with PCA (principal component analysis) as

$$S = \bar{S} + A^{id}x^{id} + A^{exp}x^{exp} \tag{1}$$

$$C = \bar{C} + A^{alb}x^{alb} \tag{2}$$

where $\bar{S}$ and $\bar{C}$ denote respectively the 3D shape and albedo of the average face, A^{id} and A^{alb} are the principal axes extracted from a set of textured 3D meshes with a neutral expression, A^{exp} represents the principal axes trained on the offsets between the expression meshes and the neutral meshes of individual persons, and x^{id}, x^{exp} and x^{alb} are the corresponding coefficient vectors that characterize a specific 3D face model. In this research, x^{id} and x^{alb} are of 100 dimensions, with the bases A^{id} and A^{alb} from the Basel Face Model (BFM) [21], while x^{exp} is of 79 dimensions with the bases A^{exp} from FaceWarehouse [6].

4.2 Generating Training Data for Facial Network

As aforementioned, to train the facial network, we need to have the input-output pairs with the input being HMD faces and the output being the corresponding 3DMM facial expression parameters. Our basic idea is to make use of non-occluded faces, for which it is easy to obtain the corresponding expression parameters (treated as ground-truth label), and then add HMD into non-occluded faces by data synthesis to create the corresponding input images. Fig. 3 shows the pipeline for generating facial network training data.

In particular, we record videos of six people (5 male and 1 female) without wearing HMD. To capture subtle variations of a user's expression, each subject firstly performs different expression movements, including opening and closing mouth, smiling and pulling, and then slowly pronounces vowels and a list of 10 sentences from Harvard sentences [23]. For different pose, subjects are allowed to move their head while performing expression.

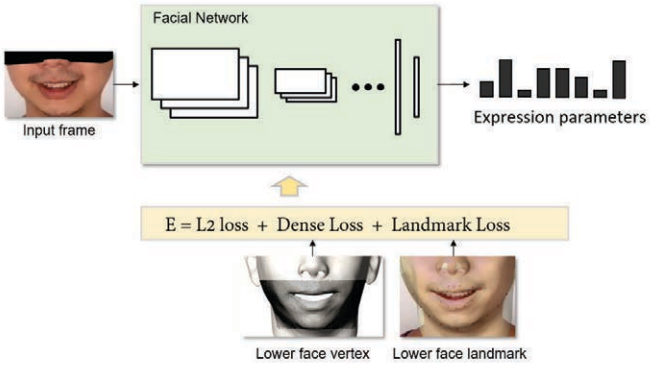

Figure 4: Facial network architecture.

For each video frame, we first use the inverse rendering method [10] to extract the 3DMM coefficients $\{x^{id}, x^{alb}, x^{exp}\}$ and the pose parameters $\{R, t, s\}$, where R, t, s denotes rotation, translation and scale respectively. With the pose parameters, a 3D face mesh S specified by $\{x^{id}, x^{alb}, x^{exp}\}$ can be projected into the image plane according to the weak perspective projection model (see 'model fit' in Fig. 3):

$$p_i = s\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} Rv_i + t,\ v_i \in S \tag{3}$$

where p_i and v_i are the projected 2D point (pixel) in the image plane and the corresponding 3D point on the face mesh S.

To synthesize photo-realistic HMD facial images, we use the 3D scanner Artec Space Spider [25] with 0.05mm accuracy to scan the HMD and get the 3D HMD model and texture. Then, for each frame, we project the scanned 3D HMD into the image according to its face pose (see 'HMD alignment' in Fig. 3). With the synthesized HMD face, finally we apply ArUco Marker detection [9] (same as the 'Face Pose Tracking' module during the online operation in Fig. 1 for training-testing consistency) to locate the visible face part (the green bounding box in Fig. 3), which is used as the input to the facial network to regress the expression parameters. Note that, to eliminate the difference caused by illumination, we mask the HMD part M_{HMD} as black pixels in both training and testing images:

$$M_{HMD} = \{\Pi(Rv + t) | \forall v \in \Omega_{HMD}\} \tag{4}$$

where $\Pi = s\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ and v denotes a point on the HMD mesh Ω_{HMD}.

4.3 Facial Network Architecture

With the obtained cropped face images $\tilde{I}$ and their corresponding labels of the expression parameters x^{exp}, we are now ready to train a facial regression network to learn a mapping: $x_*^{exp} = \psi_F(\tilde{I})$, where x_*^{exp} is the network predicted expression parameter. Fig. 4 shows the overall architecture of the facial network.

Specifically, the input cropped frame is of size 112x224. We modify ResNet-18 structure [11] to fit our frame dimension and also

change the last layer to regress the expression parameter of 79 dimension. We define the loss function for the facial network as

$$E = \left\|\psi_F(\tilde{I}) - x^{exp}\right\|_{L_2} + \omega_d \sum_{v_i^* \in \Omega_v^*, v_i \in \Omega_v} \left\|{v_i}^* - v_i\right\|_{L_2} + \omega_l \sum_{v_i^* \in \Omega_l^*, v_i \in \Omega_l} \left\|\Pi(R{v_i}^* + t) - \Pi(Rv_i + t)\right\|_{L_2} \quad (5)$$

where w_d and w_l are tradeoff parameters, Ω_v^* and Ω_v are the visible vertex set on the reconstruction meshes defined as

$$\Omega_v^* \subset \bar{S} + A^{id}x^{id} + A^{exp}x_F^{\psi}(\tilde{I})$$
$$\Omega_v \subset \bar{S} + A^{id}x^{id} + A^{exp}x^{exp}, \quad (6)$$

and Ω_l^* and Ω_l are the corresponding 3D landmark sets around the mouth region. Eq. (5) essentially consists of three terms: the first term is a common $L2$ loss on the expression parameter; the second term is a dense loss term to enforce the dense consistency on the visible part of the 3D meshes; and the third term is a landmark loss term to emphasize the consistency on the projected 3D landmark points.

5 EYE REGRESSION NETWORK

The purpose of the eye regression network is to regress the 3D eye model parameters, given an IR eye image. In this section, we will first introduce the adopted 3D eye model, and then describe how we prepare the training data and the developed eye regression network architecture.

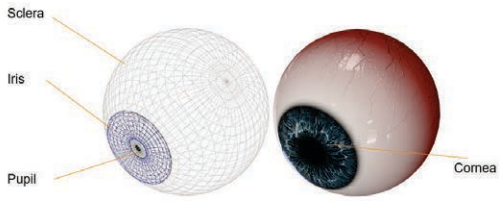

Figure 5: The adopted multi-layer parametric eye model.

Parametric Eye Model. Eye ball is a complex organ comprised of multiple layers of tissues: sclera, cornea, pupil and iris, as shown in Fig. 5 [30]. Cornea is the transparent layer forming the front of the eye with refractive n=1.376. The inner part composed by a pupil (black) and iris (colourful). Sclera is the white outer layer of the eyeball which is connected with the cornea and covered by the veins. We get the basic eye model from the artist store [13], which has 20 different iris colours, and parameterize it to make shape variable in pupillary dilation. With the 3D eye model, to render a 3D gaze, we only need to specify the eyeball gaze direction, position, iris colour and pupil size. Considering the iris color can be specified offline in advance for a user, during the online operation, the eye regression network just needs to regress the 5 parameters: pitch, yaw, pupil size, pupil centre (x, y), where the first two parameters are the angles specifying the gaze direction.

Generating training data for eye network. To train the eye regression network, we need the input-output pairs with the input being eye IR images and the output being the corresponding five eye-model parameters. Since there is no such labeled eye IR image dataset available, we construct the dataset from the scratch. In particular, we record sequences of 7 users' eyes (5 male and 2 female) from the embed IR cameras of HMD while the users are watching a black dot moving in the virtual scene. The black dot moves in a circle and also moves forward and backward in 8 directions to create gaze variations and pupil size variations.

For obtaining the gaze orientation label, we directly use the gaze vector provided by the FOVE HMD device. Particularly, we first calibrate eye gaze for each subject using the default calibration program from the device, and then record each IR frame with the gaze vector returned from the device.

For obtaining the labels of pupil size and position, we use the conventional image processing techniques to segment pupil. Fig. 6 shows the pipeline for pupil segmentation. Specifically, given an eye Image, we use the method [34] to calculate an accumulated value of a fixed-size area 5x5 for each pixel to find the darkest point, which is deemed to be in the pupil region. Then, we set the darkest point as a seed, and use the classical watershed segmentation method to segment the pupil. Considering there are 6 IR lighting points that often cause holes or discontinuity for segmentation, we further refine the result using hole filling and morphology techniques [17]. After that, we use the conventional ellipse fitting method [2] to find an optimal ellipse to match the segmented contour. Finally, we label the ellipse centre as the pupil centre, and use the major axis length as the pupil size.

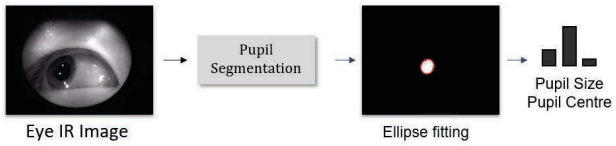

Figure 6: Image processing pipeline for pupil segmentation.

Eye Network Architecture. Fig. 7 shows the architecture of the developed eye gaze network. In particular, an eye IR image is first rescaled to 87x135 before input into the eye network, and then we apply a chain of CNN operations: 11x11 and 5x5 convolutional layer with reLU, 2x2 max pooling, followed by 5x5 convolutional layer and 2x2 max pooling, and a final fully connected layer with 5 parameters output. We use L2 loss to regress the five parameters: pitch, yaw, pupil size, and pupil centre (x, y).

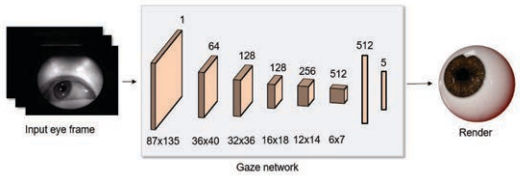

Figure 7: Architecture for the developed eye gaze network.

6 EXPERIMENTAL RESULTS

6.1 Implementation

For the facial network, we choose 8608 recorded images with various facial expressions. For each image, as described in Section 4.2, we generate an HMD mask on the image and crop the lower face region of size 120x230 according to the face pose. For data augmentation, we further randomly crop 10 times for each facial 120x230 image into 112x224. In this way, we generate 86080 112x224 images for training. The learning rate is initialized as 0.001 and decays with a base of 0.9 at every epoch for total 70 epochs. The tradeoff parameters w_l andw_d are set as 1 and $1 \cdot 10^{-6}$.

For the eye gaze network, we collect 18806 frames for 7 subjects from IR cameras and record the gaze labels. The pupil size and centre labels are generated using the image processing pipeline described in Section 5. We calibrate each eye IR camera and undistort the image. For data augmentation, we mirror left eye image to right image and get in total 14 groups. And we further randomly crop 10 times of each eye image into a size of 87x135. In this way, we obtain totally 376120 images for the eye gaze network. The learning rate is initialized as 0.0001 and decays with a base of 0.96 at every half epoch for total 70 epochs.

The facial and eye networks are trained with a batch size of 64 and 256, respectively, on a PC with 2 GPUs GTX1080. The system is tested with a variety of subjects under different circumstances. The system is very convenient to use for different people since it does not require any user-dependent calibration.

6.2 Results on Facial Expression Capture

Visual results of facial expression capture. Fig. 8 shows the visual results of facial expression capture using our facial network on a variety of users. We can see that even with faces being largely occluded by HMD, our system can still regress the expression parameter and reconstruct the face shape properly. Our system is quite robust to different people, different expressions and different poses. For the real-time facial performance capture, please see our supplementary video.

Quantitative evaluation of facial expression capture. In order to quantitatively evaluate the proposed HMD facial expression capture, we use synthesized data for testing. Specifically, we simulate 500 test images of one subject, while using the synthesized images of the remaining subjects for training. We use the mean 2D landmark error (L_2 distance) as the evaluation metric. For each synthesized HMD face image, since we have the original non-occluded face image, we apply the facial landmark detector, dlib [14], on the non-occluded face image, and choose 29 detected landmarks around the lower face region as the 'ground-truth' landmarks for evaluation. The predicted 2D landmarks are obtained by projecting the corresponding 3D landmark vertices of the reconstructed 3D face mesh into the image plane. Note that none of the existing facial landmark detection methods can work on HMD face images.

Table 1 lists the mean landmark error results. We compare our method with a simplified version, i.e. only use the L_2 loss (the first term in (5)) for training. We can see that, compared with the baseline, our method achieves much lower landmark error by introducing the dense loss and the landmark loss in (5). This is because the L_2 loss alone cannot distinguish the importance of different expression parameters, and during our experiment, we find out that using combined loss can make network training converge faster.

Table 1: Quantitative results of facial expression capture.

	Mean landmark error (pixel)
L_2 Loss	3.04
Our method	0.94

Table 2: Quantitative results of eye gaze capture.

	Mean gaze error	Runtime/frame
Optimization method [34]	26.6^o	136.9 ms
Our method	11.6^o	2.18 ms

6.3 Results on Eye Motion Capture

Visual result of gaze estimation. Fig. 9 visualizes our 3D gaze estimation results using our eye gaze network on different users. It can be seen that our system tracks the eye gaze well and is robust to different gaze directions, pupil locations and different users. We also put a sequence for gaze estimation in the supplementary video.

Fig. 10 shows the results on the cases with subtle changes of users pupil size. As we know, different lighting environment or rending object at different distances for HMD might cause physiological pupillary response that varies the size of the pupil. This pupil's dilation or constriction can be well captured by our eye gaze network via regressing the pupil size, as demonstrated in Fig. 10.

Quantitative results of eye gaze capture. Table 2 shows the quantitative results of eye gaze capture in terms of mean gaze error and runtime. We compare the estimated gaze vectors with the ground truth gaze vectors, which are directly obtained from the HMD device. We choose one subject's left eye as the test sample and keep the rest 13 groups for training. We compare our CNN based solution with an optimization based baseline method, which uses the traditional pupil tracking method [34] to fit the projected 3D pupil contour to the 2D pupil contour. Particularly, similar to our image processing pipeline shown in Fig. 6, the baseline first segments the pupil region, finds the optimal ellipse contour [2], and then computes the ratio of the ellipse's major and minor lengths, from which it can obtain the pitch and yaw of the 3D pupil contour [34].

From Table 2, we can see that our method is much faster than the optimization method, since the latter needs to go through a series of image processing while the former only needs to go through one forward pass on the eye gaze network. Moreover, the mean gaze error of our method is much smaller than the baseline.

6.4 Other Results and Discussions

Integration results. Fig. 11 shows the visual results of the integrated face-eye performance capture on various users with different pose. For front view, please see our video. It can be seen that, given a RGB frame of a user wearing HMD and an eye IR frame, our system can produce in real time a personalized avatar with the corresponding identity, pose, expression and eye motion. The entire

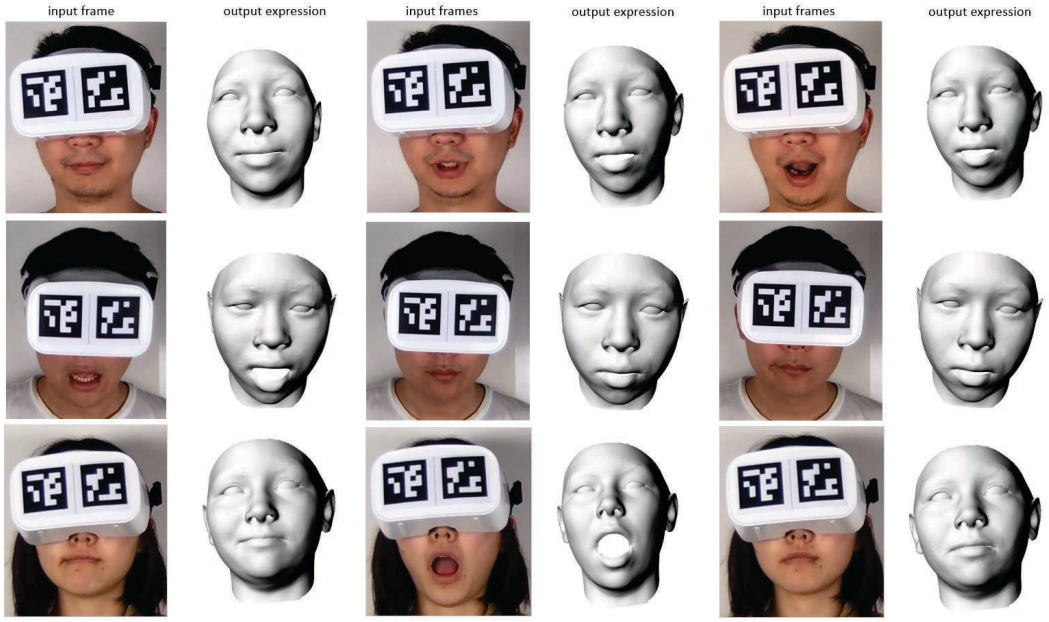

Figure 8: Expression capture results obtained with our facial network on a variety of users. For the real-time facial performance capture, please see our supplementary video.

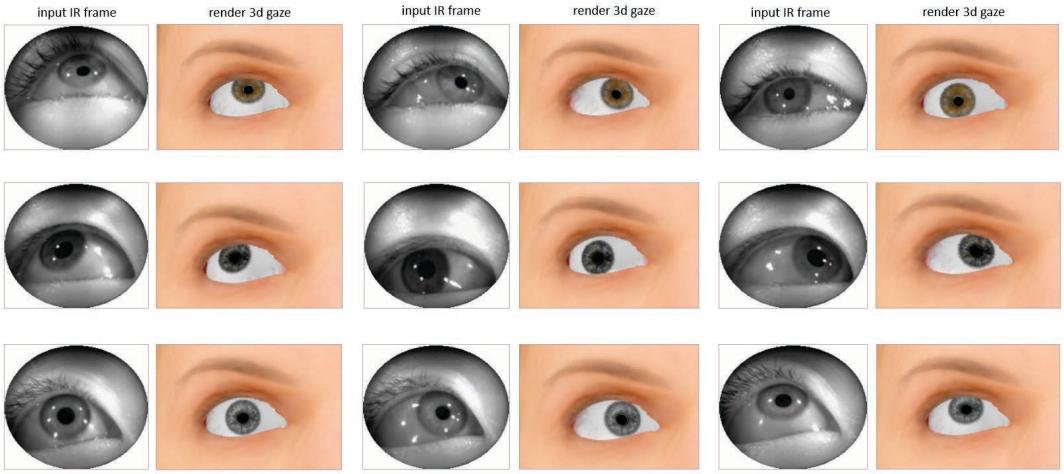

Figure 9: Gaze estimation results obtained by our eye gaze network on different users.

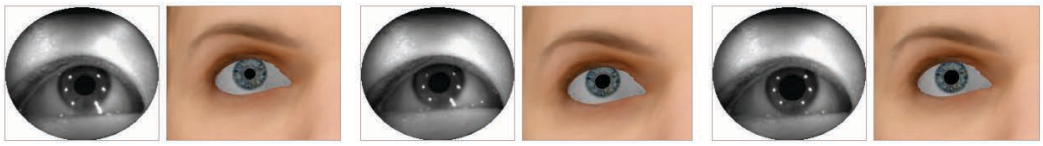

Figure 10: Gaze estimation results on the cases with subtle changes of users pupil size.

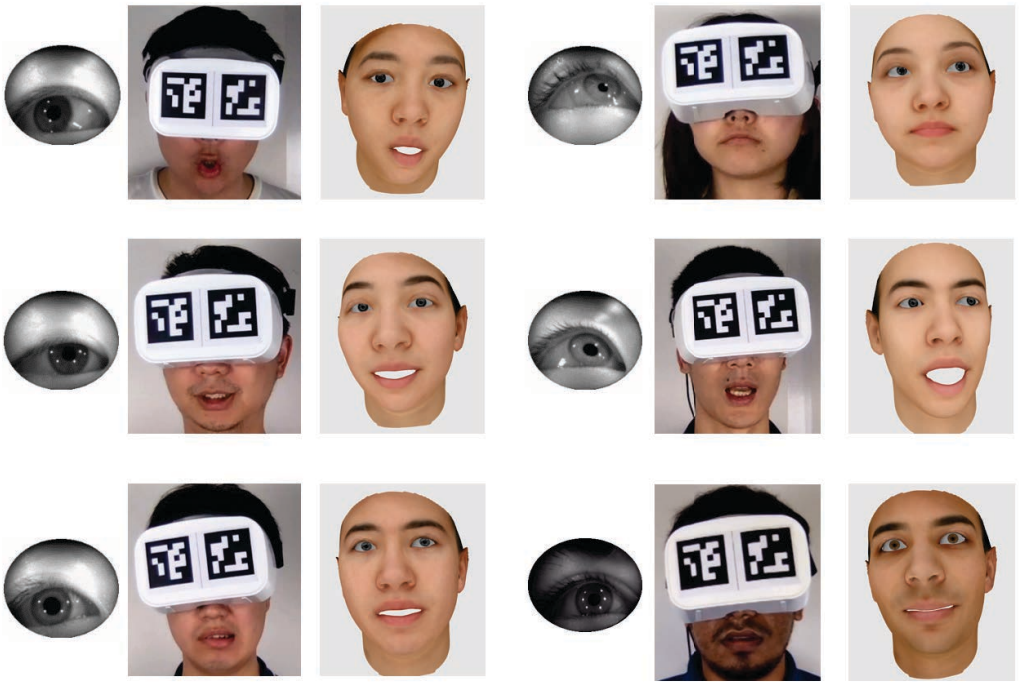

Figure 11: Integrated face-eye performance capture results for various users.

Figure 12: Retarget the face-eye motion to different avatars.

system spends 31.5 ms on the facial network and 2.18 ms on the eye gaze network, and overall it can achieve 30 fps.

Face-eye motion retarget. With the regressed expression and eye parameters, we can easily retarget the face-eye motion to different avatars by using different mesh identity and albedo parameters, as illustrated in Fig. 12.

Limitations. There are some limitations in our system. First, for ArUco Marker detection we used to detect the position of HMD, occasionally it might not detect the markers correctly, which will lead to the system failure or flicking. Second, as we only collected data from a limited number of subjects under normal lighting situation, the system may not be robust to users whose appearances are significantly different from the training data. We are planing to collect more data from a wider group of subjects to improve our network robustness. Third, our system does not consider the eyelid movement. It will not be able to regress the gaze when the user is about to close his or her eyes.

We would like to point out that, although the recent studies [16, 20, 28] attack similar problems, we are not able to give comparisons. This is mainly because of the high complexity of their systems, no available codes, and different system setups. For example, [16] needs the input from strain sensors. Both [16] and [28] requires RGB-D inputs.

7 CONCLUSIONS

In this paper, we have presented a CNN-based 3D face-eye capture system for HMD users. Our system integrates a 3D parametric gaze model into 3D morphable face model, and it can easily produce a digital personalized avatar from the input of a RGB HMD face image and an IR eye image, with no calibration step. Moreover, to train the facial and eye gaze networks, we collect face and VR IR eye data from multiple subjects, and synthesize pairs of HMD face data with expression labels. Extensive results show that our system is robust in capturing facial performance and 3D eye motion. It provides a promising direction for VR HMD based telepresence.

8 ACKNOWLEDGMENTS

We thank all reviewers for their valuable comments. We would like to thank Yudong Guo for 3DMM extraction module and also thank Chuanxia Zhen, Deng Teng, Yujun Cai, Eri Ishikawa, Chenqiu Zhao and Ayan Kumar Bhunia for data collection. This research is supported by the BeingTogether Centre, a collaboration between Nanyang Technological University (NTU) Singapore and University of North Carolina (UNC) at Chapel Hill. The BeingTogether Centre is supported by the National Research Foundation, Prime Minister Office, Singapore under its International Research Centres in Singapore Funding Initiative. Also, this research is partially supported by SCALE@NTU.

REFERENCES

[1] Oswald Aldrian and William AP Smith. 2013. Inverse rendering of faces with a 3D morphable model. *IEEE transactions on pattern analysis and machine intelligence* 35, 5 (2013), 1080–1093.

[2] Robert B. Fisher Andrew W. Fitzgibbon, Maurizio Pilu. 2017. Direct Least Squares Fitting of Ellipses. (2017). http://cseweb.ucsd.edu/~mdailey/Face-Coord/ellipse-specific-fitting.pdf

[3] Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 187–194.

[4] Google Research Blog. 2017. (2017). https://research.googleblog.com/2017/02/headset-removal-for-virtual-and-mixed.html

[5] Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. 2015. Real-time High-fidelity Facial Performance Capture. *ACM Trans. Graph.* 34, 4, Article 46 (July 2015), 9 pages. https://doi.org/10.1145/2766943

[6] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. 2014. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics* 20, 3 (2014), 413–425.

[7] Fove. 2017. Fove. (2017). https://www.getfove.com/

[8] Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2016. Reconstruction of Personalized 3D Face Rigs from Monocular Video. *ACM Trans. Graph.* 35, 3, Article 28 (May 2016), 15 pages. https://doi.org/10.1145/2890493

[9] S. Garrido-Jurado, R. Muñoz Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. 2014. Automatic Generation and Detection of Highly Reliable Fiducial Markers Under Occlusion. *Pattern Recogn.* 47, 6 (June 2014), 2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005

[10] Yudong Guo, Juyong Zhang, Jianfei Cai, Boyi Jiang, and Jianmin Zheng. 2018. 3DFaceNet: Real-time Dense Face Reconstruction via Synthesizing Photo-realistic Face Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* abs/1708.00980 (2018). arXiv:1708.00980 http://arxiv.org/abs/1708.00980

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

[12] HTC. 2017. HTC Vive. https://https://www.vive.com/sg/. (2017).

[13] Carel Jordaan. 2016. (2016). https://www.highend3d.com/downloads/3d-textures/c/eye-iris-textures

[14] Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10 (2009), 1755–1758.

[15] Pupil Labs. 2017. Pupil Labs. https://www.pupil-labs.com/. (2017). https://www.pupil-labs.com/

[16] Hao Li, Laura Trutoiu, Kyle Olszewski, Lingyu Wei, Tristan Trutna, Pei-Lun Hsieh, Aaron Nicholls, and Chongyang Ma. 2015. Facial Performance Sensing Head-Mounted Display. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2015)* 34, 4 (July 2015).

[17] Matlab. 2017. Image Segmentation. (2017). https://www.mathworks.com/help/images/image-segmentation-using-the-image-segmenter-app.html

[18] Microsoft. 2017. (2017). https://www.microsoft.com/en-us/hololens

[19] Oculus. 2017. Oculus. (2017). https://https://www.oculus.com/

[20] Kyle Olszewski, Joseph J. Lim, Shunsuke Saito, and Hao Li. 2016. High-Fidelity Facial and Speech Animation for VR HMDs. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2016)* 35, 6 (December 2016).

[21] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. 2009. A 3D face model for pose and illumination invariant face recognition. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*. Ieee, 296–301.

[22] Shunsuke Saito, Tianye Li, and Hao Li. 2016. Real-Time Facial Segmentation and Performance Capture from RGB Input. *CoRR* abs/1604.02647 (2016). arXiv:1604.02647 http://arxiv.org/abs/1604.02647

[23] Harvard Sentences. 2017. (2017). https://en.wikipedia.org/wiki/Harvard_sentences

[24] SMI. 2015. SMI. (2015). https://www.smivision.com/

[25] Artec Space Spider. 2017. (2017). https://www.artec3d.com/portable-3d-scanners/artec-spider

[26] Fuwen Tan, Chi-Wing Fu, Teng Deng, Jianfei Cai, and Tat-Jen Cham. 2017. FaceCollage: A Rapidly Deployable System for Real-time Head Reconstruction for On-The-Go 3D Telepresence. In *Proceedings of the 2017 ACM on Multimedia Conference (MM '17)*. ACM, New York, NY, USA, 64–72. https://doi.org/10.1145/3123266.3123281

[27] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. 2016. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.

[28] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. FaceVR: Real-Time Facial Reenactment and Eye Gaze Control in Virtual Reality. *CoRR* abs/1610.03151 (2016). arXiv:1610.03151 http://arxiv.org/abs/1610.03151

[29] https://www.tobiipro.com/ Tobii. 2017. (2017). https://https://www.tobiipro.com/

[30] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. 2015. Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. *CoRR* abs/1505.05916 (2015). arXiv:1505.05916 http://arxiv.org/abs/1505.05916

[31] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. 2016. A 3D Morphable Model of the Eye Region. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Posters (EG '16)*. Eurographics Association, Goslar Germany, Germany, 35–36. https://doi.org/10.2312/egp.20161054

[32] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. 2016. Learning an Appearance-based Gaze Estimator from One Million Synthesised Images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. ACM, New York, NY, USA, 131–138. https://doi.org/10.1145/2857491.2857492

[33] Ronald Yu, Shunsuke Saito, Haoxiang Li, Duygu Ceylan, and Hao Li. 2017. Learning Dense Facial Correspondences in Unconstrained Images. *CoRR* abs/1709.00536 (2017). arXiv:1709.00536 http://arxiv.org/abs/1709.00536

[34] YutaItoh. 2017. 3D pupil tracking. (2017). https://github.com/YutaItoh/3D-Eye-Tracker