Mask-off: Synthesizing Face Images in the Presence of Head-mounted Displays

Yajie Zhao* University of Kentucky
Qingguo Xu† University of Kentucky
Weikai Chen‡ USC Institute for Creative Technologies
Chao Du§ University of Kentucky
Jun Xing¶ USC Institute for Creative Technologies
Xinyu Huang‖ North Carolina Central University
Ruigang Yang** University of Kentucky, Baidu Inc., Beijing, China

Figure 1: Our system automatically reconstruct photo-realistic face videos for users wearing HMD. ***(Left)*** Input IR eye images. ***(Middle)*** Input face image with upper face blocked by HMD device. ***(Right)*** The output of our system.

Abstract

Wearable VR/AR devices provide users with fully immersive experience in a virtual environment, enabling possibilities to reshape the forms of entertainment and telepresence. While the body language is a crucial element in effective communication, wearing a head-mounted display (HMD) could severely hinder the eye contact and block facial expressions. We present a novel headset removal technique that enables high-quality occlusion-free communication in virtual environment. In particular, our solution synthesizes photoreal faces in the occluded region with faithful reconstruction of facial expressions and eye movements. Towards this goal, we develop a novel capture setup that consists of two near-infrared (NIR) cameras inside the HMD for eye capturing and one external RGB camera for recording visible face regions. To enable realistic face synthesis with consistent illuminations, we propose a data-driven approach to fuse the narrow-field-of-view NIR images with the RGB image captured from the external camera. In addition, to generate photorealistic eyes, a dedicated algorithm is proposed to colorize the NIR eye images and further rectify the color distortion caused by the non-linear mapping of IR light sensitivity. Experimental results demonstrate that our framework is capable to synthesize high-fidelity unoccluded facial images with accurate tracking of head motion, facial expression and eye movement.

Index Terms: AR/VR, headset removal, face inpainting, eye synthesis

*e-mail: yajie730@gmail.com
†e-mail: qingguo.xu@gmail.com
‡e-mail: chenwk891@gmail.com
§e-mail: chao.du@uky.edu
¶e-mail: junxnui@gmail.com
‖e-mail: xinyu.huang@nccu.edu
**e-mail: ryang@cs.uky.edu

1 Introduction

The advances of virtual reality (VR) and augmented reality (AR) techniques have unleashed the possibility to explore novel environments in a way that is unprecedentedly immersive. The virtual environment created by VR/AR technology enables users to communicate or collaborate in a shared computer-simulated scene without being physically present as shown in Holoportation [24]. Such immersive experience brings fundamental impact on future forms of communication, training and telepresence. However, eye contact and effective face-to-face communication remain difficult in VR/AR environments as the head-mounted display (HMD) occludes a large portion of the face. Hence many teleconferencing scenarios involving team communication or negotiation tactics using facial cues cannot be realistically simulated.

To resolve this issue, there are several prior works aiming to unveil the facial performance underneath the headset. One proposed solution is to track the facial expressions using cameras or other sensing devices embedded inside the headset. The tracking result is then used to drive a digital avatar [21, 23] or a pre-recorded video [36] which reproduces the facial expressions. While high-quality results have been achieved, approaches in this line cannot perform in-place HMD removal and thus body movements that are out of the view of embedded cameras cannot be replicated. As body language also serves as an important manner of communication, the absence of body motions and gestures could significantly lower the efficiency of communication. The other direction of addressing this issue is to generate an in-place inpainting of the occluded face regions, making it possible to present the full picture of subject with the headset virtually removed. This approach attacks a much more challenging problem as the face is occluded by a significant amount. In addition, as human perception is highly sensitive to facial artifacts, dynamic facial expressions and eye movements need to be faithfully reproduced with temporal coherency. Due to the technical challenge, very few prior works have been proposed towards this end. Burgos-Artizzu *et al.* [5] present a headset removal system by projecting a learned 3D textured model onto the video frame. Though the face inpainting can be achieved in real time, their approach cannot recover eye-gaze and eyeball movement. More recently, Frueh *et al.* [13] improve the realism of the synthesized facial content by incorporating the tracking and reconstruction of eye-gaze of the user.

2019 IEEE Conference on Virtual Reality and
3D User Interfaces
23-27 March, Osaka, Japan
978-1-7281-1377-7/19/$31.00 ©2019 IEEE

However, their technique cannot completely remove the headset as the inferred facial texture is directly projected onto the headset.

In this paper, we present a novel end-to-end framework that provides a complete removal of headset at real-time frame rate. Our approach is capable to synthesize illumination-consistent facial content in the occluded region with high-fidelity details of eye movements(see Figure 1). Compared to Frueh *et al.*'s approach, which requires highly specialized eye-tracking hardware solution and an external RGBD sensor for face tracking, our approach only requires a low-cost hardware setup consisting of two inward NIR cameras inside the headset and one external RGB camera for video recording. The key to our framework is a novel image warping technique which provides a high-quality fusion of the two NIR images of eye regions and the RGB image that captures the visible face part.To faithfully reproduce the eye movement, special care needs to be taken to deal with the gray-scale eye images taken from the NIR cameras. In particular, we propose a novel technique that colorizes NIR images with coherent color tones with the visible part of face. In addition, a refinement algorithm is proposed to rectify the color distortion introduced by the non-linear mapping of NIR light sensitivity (Section 8.2).

2 Related Work

Facial Performance Capture. Our work is based on facial performance capture. Over the past two decades, a number of works have been proposed which significantly advanced this field. In [8, 9], Cao *et al.* propose regression based methods to track and animate 3D faces using a single RGB consumer-level camera. In [7], Cao *et al.* further improve the realism of face tracker by recovering the medium-scale details, e.g. expression wrinkles, through local regressors. In [19], the author propose a solution to high frame rate performance capture on dense depth map. Although the above methods have achieved high-quality performance in the constrained setup, they are still not robust to occlusion and unconstrained sequences. In [16], Hsieh *et al.* present a facial performance capture system using RGB-D sensor, which is robust to occlusion and unknown identities. More recently, by leveraging a pixel-level facial segmentation mask generated by a deep neural network, Saito *et al.* [29] proposed a more robust technique which could faithfully track facial expressions on unconstrained images with large occlusions. However, face performance capture techniques can only reconstruct 3D geometry of faces. Additional care must be taken to recover the facial texture in the occluded regions, which remains a challenging problem.

Face Reconstruction and Appearance Synthesis. There have been many works over the years which have demonstrated strong capability to reconstruct facial geometry and appearance from RGB images. The pioneering active appearance models (AAM) and 3D morphable models have been extensively applied in image-based reconstruction of 3D shape and texture of faces [4, 37]. Ichim *et al.* [17] aim to reconstruct personalized textured avatar from hand-held video, which could further be used for tracking and animation. With the advancement of 3D sensing technology, depth sensors are also widely employed to obtain 3D clues for more robust reconstruction of geometrical features. In particular, Cai *et al.* [6] present a deformable model fitting algorithm to track 3D face model using a consumer-level depth camera. In [12], the author proposes a seamless texture fusion to reduce ghosting effects in multiview RGB-D capture system, which are caused by discontinuities in depth, occlusion and critical time constraints. In [34], Thies *et al.* proposed a face reenactment technique that transfers facial expressions from source subject to a target person at an interactive frame rate. A RGB-D sensor is required in their framework to estimate and track facial expressions and head poses. More recently, in their follow-up work, Thies *et al.* [35] further extend their framework to a low-cost capturing setup which only requires RGB cameras. However, the above approaches cannot scale well to inputs with large occlusions, limiting their application in headset removal scenarios, where the subject's face is severely occluded.

Headset removal. Back to the early 2000s, Takemura *et al.* [33] pioneered in removing the HMD in a partner's view for recovering the eye-contact. In their follow-up work in [32], they proposed "MR face" that restores the eye regions by overlaying a virtual face on the real one using mix-reality techniques. As directly removing headset from a video footage is highly challenging, recent research efforts focus on revealing the occluded face by transferring the user's facial expressions to a third-party medium, e.g. a digital avatar or a target video clip. In [28], Romera-Paredes *et al.* track the facial expression underneath the headset, which is visualized by animating a digital avatar. Li *et al.* [21] further develop a novel HMD that uses electronic material to measure the surface strain signals and a RGB-D camera to track visible face regions. In the follow-up work [23], Olszewski *et al.* improved the tracking accuracy by proposing a specialized HMD with additional sensors to collect signals from occluded regions. A deep learning algorithm is also incorporated to extract high-fidelity motions from video in real time. Similarly, In [15, 31], the author use special designed hardware to recognize and map expressions to cartoon avatars. Instead of driving a 3D avatar, FaceVR [36] proposes to animate a target video with tracked facial expressions from a VR user wearing a headset.

The above methods fail to replicate body languages, e.g. the head pose and body motions, presented in the original inputs. Therefore, another line of research strives to achieve in-place headset removal which could create an illusion of revealing the user's face. In [5], Burgos-Artizzu *et al.* propose to inpaint faces for HMD-occluded videos. They employ a RGB camera to capture and track the facial expressions of subject and then project the reconstructed textured facial model on top of the occluded regions. However, their approach cannot handle eye gaze and eyeball movement and suffers from inconsistent color tones in the synthesized facial content. Frueh *et al.* [13] later incorporates an eye-tracking solution and is capable to generate in-place headset removal with accurate tracking of eye movement. However, their approach fails to completely remove the headset as the recovered facial texture is directly projected onto the surface of headset. In recent work of [27], Rekimoto *e*t al. propose another headset removal prototype. They leverage a pre-scaned high-quality 3D face model and synthesize the result by merging the captured texture with the face images reflected from two embeded IR-cut filters. However, they only propose the concept of face-through HMD and fail to demonstrate any reconstruction results on wearers with dynamic motions. In addition, strong artifacts can be observed in the eye regions of the presented static results. More recently, Zhao *et al.* [38], propose a deep learning based approach that restores largely occluded facial features in the presence of VR headset. However, their approach tends to generate blurry results as the resolution and quality of synthesized results are limited by the network capability and the training data. Our approach, on the other hand, provides an end-to-end solution to completely remove headset in the input video sequence and synthesize high-fidelity faces while maintaining accurate tracking of facial expressions and eye movement.

3 Hardware Setup and Calibration

3.1 Hardware Setup

We have built two prototypes for experiments and validation. Our first system is a simulation setup, as shown in Figure 2 right. It consists of three cameras. The middle one is a color camera with a resolution of 640×480, it is used to capture the entire face. The other two cameras are near-infrared (NIR) VGA (640×480) cameras capturing the two eye regions using narrow field-of-view lenses. IR LEDs are used to provide sufficient illuminations for the NIR cameras. All cameras are synchronized. The color image can capture the full face of the user without any occlusion. We then simulate

the occluded face image by synthesizing masks to block the upper face. In this setup, the three cameras are used to simulate the case in which all three cameras are rigidly attached to the HMD display, so that head pose doesn't need to be tracked and always be frontal. This setup allows us to capture ground truth images for evaluation purpose. The second system is for run-time capturing (Figure 2 Left). We install two small NIR cameras along with two micro IR LEDs inside the headset to observe the eye region. The camera/LED set, in current form, could partially occlude the display (around 10% occlusion). However, the occlusion does not greatly affect the user experience, this limitation can be easily resolved by using more advanced IR camera or LED with more compact size.

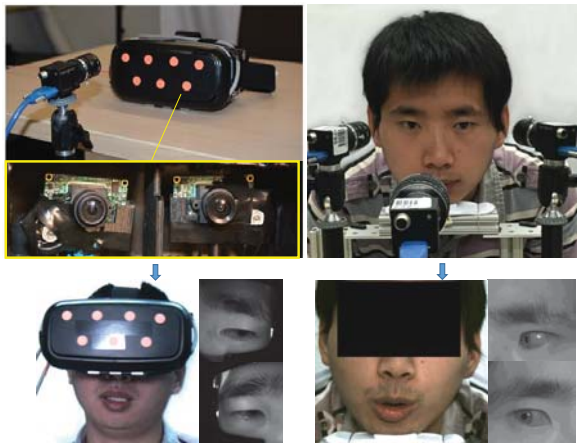

Figure 2: Two experimental systems we have built. ***(Left Column)***our real setup. ***(Right Column)*** our simulation setup with three cameras, two for eyes and one for the entire face.

3.2 Calibration

One challenging task is how to geometrically calibrate all cameras. While the simulation setup is easy to deal with, the real one is more difficult since the HMD with two internal IR cameras can move freely. However, we notice that the geometries among the three cameras are fixed. We can obtain the extrinsics of NIR cameras as long as the extrinsic of HMD is known. Due to the page limit, we postpone the details of calibration to the *supplemental materials.*

4 System Overview

Our system consists of four modules as shown in Figure 3. We reconstruct a personalized 3D animatable head model from a video sequence captured offline in the first module (Section 5). In the second module (Section 6), we propose a novel algorithm to align 3D head model to the face image with severe occlusion. In the third module (Section 8), we propose another novel algorithm to process the warped near-infrared eye images. The eye images are first colorized based on the color information of reference image. The obvious artifacts (e.g., color distortion caused by IR light sensitivity) in the eye regions are further removed in this module. In order to generate realistic face images without occlusions, in the fourth module (Section 7), we first retrieve for the reference image and then compose the complete face from different sources.

5 3D Head Reconstruction

In the offline data acquisition stage, we record a video sequence of a novel user with neutral expression under various head poses(These data will also be used in Section 7). The image frames are used to reconstruct a personalized 3D head model for the user. We first apply the approach in [14] to estimate a sparse point cloud. A bi-linear face morphable model described in [10] is then used to reconstruct a dense 3D model M with $11K$ vertices from the sparse point cloud with identity weights C_{id} and expression weights C_{exp}. As we assume neutral expression during the reconstruction, only identity weights C_{id} need to be estimated in this stage.

Denote the reconstructed sparse 3D point cloud as M^s, our fitting energy function is defined as,

$$E = \sum_{k=1}^{V} \|sRM_k + t - M_k^p\|^2, \tag{1}$$

where the 3D rigid transformation between the sparse point cloud and the bi-linear face model consists of a scale factor s, a 3D rotation matrix R and a translation vector t. M_k and M_k^p are the k_{th} pair of 3D vertices in the dense 3D head model and sparse point cloud. In each iteration, V vertices are selected from the spare point cloud and the corresponding nearest vertices in the dense head model are updated. The initial transformation is computed by using seven 3D facial landmarks reconstructed in 3D point cloud.

We further improve the reconstruction accuracy by using 2D facial landmarks in images that are detected based on the real-time algorithm proposed in [18]. The cost function is defined as,

$$E = \sum_{i=1}^{N}\sum_{j=1}^{K} \|P_i M_j - l_{ij}\|^2 + \lambda \sum_{i=1}^{50}((C_0^i - C_{id}^i)/\theta)^2 \tag{2}$$

where N image frames and K facial landmarks in each frame are used. M_j is the jth 3D facial landmark in the dense head model, l_{ij} is the jth facial landmark in the ith image frame, and P_i is the projection matrix for the ith image frame. The second term in the Equation 2 is a regularization term that makes the estimated head model M close to the head model estimated from Equation 1, which is denoted as C_0. θ is the eigenvalue of the shape covariance matrix. This term also prevents the the geometry from degeneration and local minima.

6 Face Alignment and Tracking

In this section, we present the details on face tracking based on our hardware configuration, calibration and landmarks. *The details of facial landmarks can be found in the supplementary material.*

6.1 Initial Alignment

We first extract the facial landmarks from both the color and IR images (more details on landmark extraction can be found in the supplemental materials). After the offline calibration, it is robust to track the HMD's pose in real time. The transformations between eye cameras and the HMD are fixed after the calibration. However, the transformation between the head and the HMD is different for individual users. Even for the same user, the transformation could also be different every time when they wear the HMD. Therefore, it is necessary to estimate the transformation (represented as rotation R_* and translation T_*) between the 3D head model and the HMD after a user puts on the device.

In our system, we conduct an initial alignment right after a user puts on the HMD. The user is instructed to change head pose with a neutral expression. The alignment is formulated as a non-linear minimization problem. The cost function E_{init} consists of two terms as shown in Equation 3.

$$E_{init} = E_f + \lambda E_e \tag{3}$$

where λ is the weight to control E_e, we set $\lambda = 2$ for all the cases. The first term E_f is the projection error between visible facial features and corresponding 3D points of the head model projected to images. This term is defined in the following equation:

$$E_f = \sum_i d(\mathbf{x}_i, P_{h\to f}[R_* T_*]\mathbf{X}_i)^2 \tag{4}$$

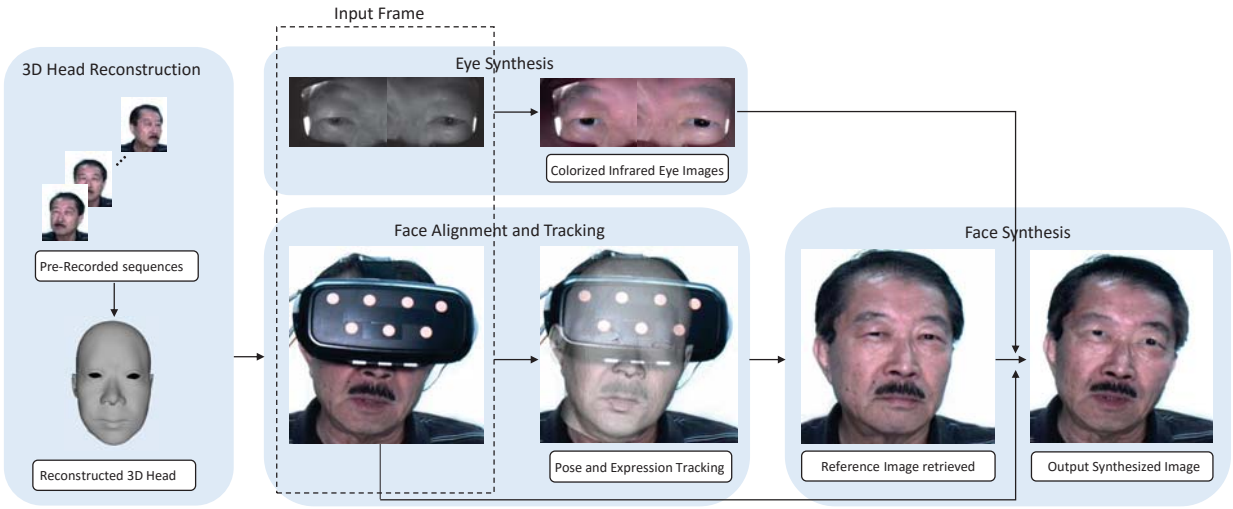

Figure 3: Conceptual overview of our system. From the first stage to the fourth stage, our goal is to synthesize a photo-realistic face image without occlusion.

where $\mathbf{x}_i$ and $\mathbf{X}_i$ are the 2D visible facial landmarks in the ith image frame of the face camera and corresponding 3D points of the head model, $d(\cdot)$ represents the Euclidean distance between two 2D image points, and $P_{h\to f}$ is the projection matrix from the HMD device to the face camera.

The second term E_e is defined as

$$E_e = \begin{array}{l} \sum_i d(\mathbf{x}_i, P_{h\to f} M_{e^l\to h}[R_* T_*]\mathbf{X}_i)^2 + \\ \sum_i d(\mathbf{x}_i, P_{h\to f} M_{e^r\to h}[R_* T_*]\mathbf{X}_i)^2 \end{array} \quad (5)$$

where $\mathbf{x}_i$ and $\mathbf{X}_i$ are the 2D visible landmarks in the ith image frame of the NIR eye camera and corresponding 3D points of the head model, and $M_{e^l\to h}$ and $M_{e^r\to h}$ are the transformation matrices from eye cameras to the HMD device.

The initial guess of R_* is set to the identity matrix as the rotation between the HMD device and the head model is often very small, and the initial guess of the translation vector in T_* is set to $[0,0,dz]$, where dz is the rough distance between the eye region and the corresponding near infrared camera. The Levenberg-Marquardt iteration method is applied to optimize this objective function.

6.2 Real-time Alignment

With the initial alignment, we can easily track the head pose in real time by estimating the projection matrix $P_{h\to f}$ for each image frame. In this step, we further update the expression weights C_{exp} in each frame with identity weights fixed. The expression weights are estimated based on the energy function,

$$E_{exp} = E_f + \lambda_1 E_e + \lambda_2 E_t + \lambda_3 E_s, \quad (6)$$

where E_f and E_e are defined in Equation 4 and 5 with $\mathbf{X}_i$ replaced by the bi-linear model and the transformation matrices (R_* and T_*) fixed. E_t is the constraint imposed by the expression weights from the previous image frame. E_t is defined by,

$$E_t(C_{exp}) = \| C_{exp}^{t-1} - C_{exp}^{t} \|^2, \quad (7)$$

where C_{exp}^{t} and C_{exp}^{t-1} are the expression weights for current and previous image frame respectively. E_s is the regularization term that forces the expression weights to be close to the statistical center which avoids of degeneration. E_s is defined as,

$$E_s = \sum_{i=1}^{N=25} (C_{exp,i}/\theta_i)^2 \quad (8)$$

where θ eigenvalue of the expression covariance matrix. E_s can also be defined as a Tikhonov regularization energy term $C_{exp}^T D C_{exp}$ with $D = diag(1/\theta^2)$. The weights we used to balance the terms in our setup is $\lambda_1 = 2$, $\lambda_2 = 2$, and $\lambda_3 = 0.7$. *Detailed algorithm can be found in the supplementary material.*

7 Face Synthesis

In this section, we first search for a reference image from the data set we have captured off-line, which contains similar head pose to the query image. Then we apply a two-step warping to warp both the template image and the NIR eye images. This method mainly fills the blocked face region with visible region unchanged. Note that in [5], the author synthesized the occluded face by rendering of the textured model, which has strong artifacts especially on the boundary as the hair motion and illumination are explicitly modeled. Therefore we choose to complete the face using real images.

7.1 Retrieval of Reference Image

The similarity between ith image in the data set and the query image is measured based on three distances as shown in Equation 9. The first term is the distance between head poses of the query image (H_q) and the reference image candidate (H_r^i). The head pose is measured by pitch, yaw, and roll angles based on the transformation $P_{h\to f} R_* T_*$ that is described in Section 6.1. The second term is the distance between 2D facial landmarks of the selected reference image in previous time frame (L_{r-1}) and current reference image candidate (L_r^i). This term removes large 2D translation between two consecutive image frames even they have similar poses. The third term is defined so that current image candidate (S_r^i) and previous reference image (S_{r-1}^i) have close time stamps. This term could further make the selected reference images continuous.

$$D = \| H_q - H_r^i \|^2 + w_1 \| L_r^i - L_{r-1} \|^2 + w_2 \| S_r^i - S_{r-1} \|^2, \quad (9)$$

where w_1 and w_2 are the weights for the second and third term respectively.

7.2 Face and Eye Image Warping

As the 3D head model have been estimated and aligned with both the reference image and the query image separately, we use 3D model as a bridge to warp reference image to align with query image. The

energy function is defined as,

$$E = E_d + \alpha E_s + \beta E_b + \gamma E_h \quad (10)$$

where E_d is the data term that assumes bilinear interpolation coefficients remain unchanged after warping, E_s is similarity transformation term based on two sets of mesh points, E_b is the term to reduce the transformation outside the face region. Details of these three terms can be found in [39]. As we also want to align the silhouette of the warped template image with the query image to avoid artifacts on the face boundary for blending purpose. Therefore, we introduce another term E_h to constrain the silhouette. We set $\alpha = 0.5$, $\beta = 1$ and $\gamma = 6$. Denote $\hat{P}_s$ and P_s as a pair of 2D correspondence points on silhouette of template image and query image. To accelerate the warping process using GPU, instead of solving a large sparse matrix with unknown number as the pixel number, we divided the input into 30×20 uniform grid mesh $\hat{V}$, the warping problem is to find warped version V of this grid mesh. Then the GPU-based interpolation of each pixel is applied using the grid mesh. The interpolation weights is pre-computed and unchanged for all the frames as long as they share the same grid structure. Then E_h can be formulated as bellow:

$$E_h = \sum_{i=1}^{N} \|w_i V_i - P_i\|^2 \quad (11)$$

in which N stands for the number of corresponding pairs on silhouette, each of the $\hat{P}_i$ can be represented as the bilinear interpolation of mesh grids which contains $\hat{P}_i$, $\hat{P}_i = w_i \hat{V}_i$, in which w_i remains unchanged after warping. Figure 4 shows the effect of this term. Note that the artifacts in red rectangle caused by misalignment of silhouette is resolved by adding the term E_h, which forces the face boundary of the warped template to align with the query image.

Figure 4: Illustration of the effect of silhouette constraints. (***Left***) input. (***Middle***) blending result without term E_h. (***Right***) blending result with term E_h. The result shows that this term forces the warped template image and target image to align on the boundary to eliminate the artifacts.

Through system calibration, we obtained the 3D transformation between eyes cameras and the HMD device. As this transformation is fixed. therefore, we can easily warp the NIR eye images to the query image through 3D head geometry. The eye images contain correct eye gazes and the dynamic wrinkles around eye regions that are necessary for the face synthesis. As these images have no color information, we propose a novel eye synthesis algorithm that is described in details in Section 8.

In the final step, we blend the query image which is partially blocked by HMD, warped reference image, and two colorized eye images together. As the lower face in the target image is often darker than faces captured under the same illumination in the data set due to the shade of the HMD, we first conduct a histogram transformation to adjust the reference and two eye images to match the color of the query face image. We then blend them by using the Laplacian blending approach in [2]. Figure 5 shows the formation of the final result. Note that in the rightmost image of Figure 5, we further apply the background replacement to remove the unwanted HMD residue region that are far from the head and not covered by the mask image.

Figure 5: Illustration of the final image formation. (*from left to right*) (1) The mask used to blend images from different sources. The green region represents the background we would like to keep in the final result. The red region represent the head region that is extracted from the warped reference image. The purple region is the region corresponding to the NIR eye images. Note that there is a transformation region between the red and green region. This region feather the boundary so that different image sources could be transformed smoothly from one to another. (2) The query image with the face occluded by the HMD. (3) The blending result of the query image, warped reference image, and colorized eye images. (4) The final blending result after background replacement.

8 Eye Synthesis

In this section, we colorize the warped NIR eye images in two steps. Firstly, we colorize the eye images based on the color information from the reference image. Secondly, we further refine the eye regions by removing obvious artifacts (e.g., IR color distortion) during the colorization.

8.1 Colorization

We use the L*a*b* color space as it is close to human visual perception and separates the illuminance channel from color channels. We denote the input NIR image as I, the reference image as M, and output color image as C. M is decomposed to three channels M_L, M_a, and M_b. I is assumed as the grayscale image for C. The colorization consists of two steps. We first transfer I to C_L based on the M_L. Then we transfer M_a and M_b to C_a and C_b.

Two existing algorithms [26, 30] are applied and evaluated to transfer from I to C_L. The first algorithm [26] is a straightforward histogram transfer based on the standard deviations and mean values of I and M_L. In the second algorithm [30], the images I and M are aligned based on the landmarks and the SIFT flow [22]. Then two images are decomposed into multiscale Laplacian stacks. These stacks are updated by the gain maps and are aggregated to generate C_L. In our problem, the performance of the second algorithm could slightly better than the first algorithm. However, it is more time-consuming due to the alignment based on the SIFT flows.

In the second step, we estimate C_a and C_b using the algorithm in [20]. The color in the channel a is computed by minimizing the following energy function

$$E(a) = \sum_p ((a(p) - \sum_{q \varepsilon N(p)} w_{pq} a(q))^2 + \alpha \sum (a(p_m) - P_m)^2 \quad (12)$$

where $a(p)$ is pixel p on channel a, $N(p)$ is the neighbor pixel of p. p_m and P_m are the pre-defined seed pixels (*i.e.*, 'micro scribble' defined in [20]). This equation minimizes the difference between the color at pixel p and its weighted averages of the neighboring pixels. The weight w_{pq} is computed based on C_L and statistics of the local patch around p. The color in the channel b is also computed in the same way. However, we need to be careful to select seed pixels.If we uniformly sample from image M, colors of some seed pixels

could be wrong on the image I, such as moles and highlights. These colors propagate to following image frames gradually and generate obvious artifacts. To avoid this, we use a voting scheme to remove the unreliable seed pixels. We first run adaptive k-means clustering to segment the image M at gray scale level. Then we only select seed pixels with high confidence that is measured by

$$error = \frac{|I_p - I_c|}{I_c} \tag{13}$$

where I_p is the intensity value of pth pixel in I and I_c is the intensity value of the center of each segment. We only select the seed colors with $error < 0.06$.

8.2 Refinement of Eye Regions

The eye region after colorization often contains very strong artifacts (*i.e.*, IR color distortion effects) as shown in Figure 6(b). One possible reason is that we treat the NIR image as the grayscale image. The contrast in a NIR eye image is often weaker, especially the contrast between sclera and skin and the contrast between iris and sclera. As a result, skin colors could be transferred to the regions like sclera and iris, which easily generates IR color distortion effects.

In this section, we propose an algorithm to refine the eye regions. We first detect iris and pupil boundaries in both near-infrared and color images using the intergrodifferential operator in [11]. Combining with the eye landmarks, we segment the eye regions into three categories, pupil, iris, and sclera. We then apply histogram transformation separately in the regions of these categories. We denote the image after histogram transformation as C'. This result partially removes color distortion effects. However, it introduces strong artifacts around boundaries of these categories and makes the result unnatural. In order to remove the artifacts, we formulate a minimization based on a cost function with three terms.

$$E_d = \sum_{p_m} (C''_L(p_m) - C'_L(p_m))^2 \tag{14}$$

$$E_s = \sum_{p} ((C''_L(p) - \sum_{q \varepsilon N(p)} w_{pq} C''_L(q))^2 \tag{15}$$

$$E_b = \sum_{p} (|N_p| C''_L(p) - \sum_{q \in \Omega} C''_L(q) - \sum_{q \notin \Omega} C_L(q) - \sum_{q \varepsilon N_p} V_{pq})^2 \tag{16}$$

$$E = E_d + \alpha_1 E_s + \alpha_2 E_b \tag{17}$$

where C''_L is the L channel of the output eye image C'' (the same procedure is also applied to a and b channels), p_m is a seed pixel, $C'_L(p_m)$ and $C''_L(p_m)$ are values of the L channel on pixel p_m for input image C'_L and output image C''_L respectively, N_p is neighboring pixels of pixel p, $|N_p|$ is the number of N_p, Ω is the mask that includes only the eye region, and $V_{pq} = C_L(p) - C_L(q)$ is the gradient value of this two pixels. α_1 and α_2 are tuned based on our experiments. The weight w_{pq} is proportional to the normalized correlation between two values of the L channels. w_{pq} is given by

$$w_{pq} = 1 + \frac{1}{\sigma_p^2} (C_L(p) - \mu_p)(C_L(q) - \mu_p) \tag{18}$$

where μ_p and σ_p are the mean and standard deviation of pixel values in an image patch around p.

The first term E_d is the data term that color an unknown pixel same as the seed pixel in the input image. E_s is the smoothness term and the last term E_b is the Poisson equation with Dirichlet boundary conditions inspired by the gradient image editing [25]. *The refinement algorithm can be found in supplementary material.* Figure 6 demonstrates the effect of eye color refinement. In this Figure, we can find that image (b) contains "IR color distortion" effects, which is removed after applying our refinement process.

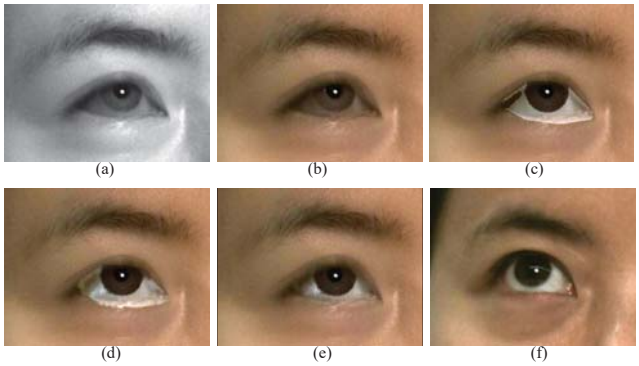

Figure 6: Results of eye refinement. (a) near-infrared image. (b) result after colorization (with color distortion). (c) result only using the date term E_d. (d) result using data and smoothness terms (E_d and E_s). (e) result using all three terms (E_d, E_s, and E_b). (f) reference eye color images.

9 Experiments

Our system framework is tested on both simulation and real setups. For the simulation setup, the goal is to validate and quantify the accuracy of our system.

9.1 Runtime

Our implementation on GPU with CUDA runs at 17 fps on a standard desktop computer (Intel i7-6850K 3.6 GHz with NVidia Gtx1080) on a image of size 640×480, and can achieve real-time performance (27 fps) on size of 320×240. In particular, the size here refers to the face area instead of the whole input frame. In real application scenarios, such face area could be quite small (e.g. Figure 11).

Table 1 shows the run time of each individual major component on 640×480 image size. In particular, the face synthesis component consists of reference image retrieval, face warping and final blending, which is the most time consuming module.

Table 1: Runtime on a face area of 320×240 (per frame)

Tracking	Colorization(w/o,w/) refinement	Face Synthesis
8ms	15/37ms	35ms

9.2 Evaluation of 3D reconstruction

We first scan head models with a high resolution structured light 3D scanning system [1] which has an average reconstructing error less than 2mm. The obtained 3D models will serve as ground truth in our evaluation. To measure the difference between the reconstructed model and the ground truth, we first align them by computing a transformation matrix using 3D facial landmark correspondences. The alignment is further refined iteratively in an ICP manner [3]. In order to evaluate the surface distance, we define each 3D vertices on the reconstructed model as p and its corresponding point as $\widetilde{p}$, in which $\widetilde{p}$ is the first intersection on the ground truth mesh along the normal direction of p. Figure 7 demonstrates our reconstructed face model in comparison with the ground truth geometry. As indicated in the error map, our approach has achieved high reconstruction accuracy. The statistics shows that the mean distance between the reconstructed model and the ground truth is 2.926 mm.

9.3 Evaluation of Face Tracking

Our algorithm described in Section 6 can robustly track 3D face models with large occlusion in real-time. As shown in Figure 8, our face tracking solution scales well to large variations of facial

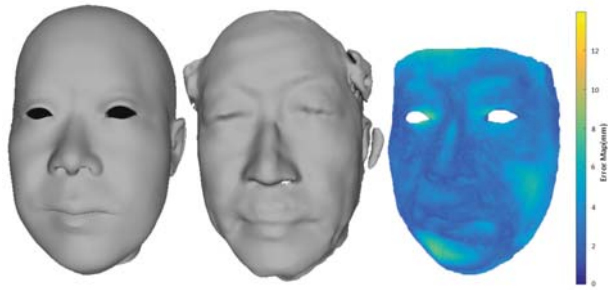

Figure 7: Evaluation of 3D reconstruction. ***from left to right*** (1) The head model reconstructed by our algorithm. (2) Ground truth. (3) The color coded error map of our reconstruction result.

Figure 8: Evaluation of tracking with HMD. The 3D model is overlaid on the original input frame. Facial expression, especially the eye blinks, are tracked robustly by our algorithm.

Figure 9: Results of eye synthesis. (***1st column***) Reference images retrieved from pre-recorded image frames. (***2nd column***) Input of NIR images. (***3rd column***) Results after colorization. (***4th column***) Results after refinement of eye regions.

expressions including eye blinks and mouth movements. *Video of Tracking results can be found in supplementary material.*

9.4 Evaluation of Eye Synthesis

We evaluate the performance of eye synthesis using inputs with different identities. Figure 9 demonstrates our results of eye synthesis from three different users. Note that the color appearances have been adjusted to be highly similar to the reference images after applying our eye colorization technique. However, the IR color distortion are also quite obvious in iris and sclera regions. The proposed eye refinement algorithm is able to significantly rectify the color distortion and achieve natural and photo-real eye rendering as shown in the fourth column of Figure 9.

9.5 Evaluation of Face Warping

Instead of using one frontal face image as template for all the frames, we retrieve the data set for the best matched reference image for each frame based on head pose similarity and temporal coherence described in Section 7.1. Figure 12 demonstrates the effectiveness of using database and retrieval algorithm compared to one template. Similar poses will result in more natural warping results especially on the face boundaries, hair styles and ear shape.

In our system, after the calibration in Section 3.2 and initial alignment in Section 6.1, we can directly get the head pose for each frame. It seems that the estimation of expression weights is unnecessary as we will not change the lower face part. However, very large mouth motions will deform the upper face shape which reflects as face silhouette changes in 2D image. As the 3D mesh works as warping guidance during the warping of the reference image, we need the target 3D mesh to be as accurate as possible. Figure 13 shows the comparison of blending results between tracking with and without expressions. Results show that the cheek of the middle image is thinner than the ground truth, which leads to artifacts on the blending boundary. However, when tracking with expressions as shown in the right image, the upper face is naturally warped to align with the mouth motion.

Figure 10: Results of real system.

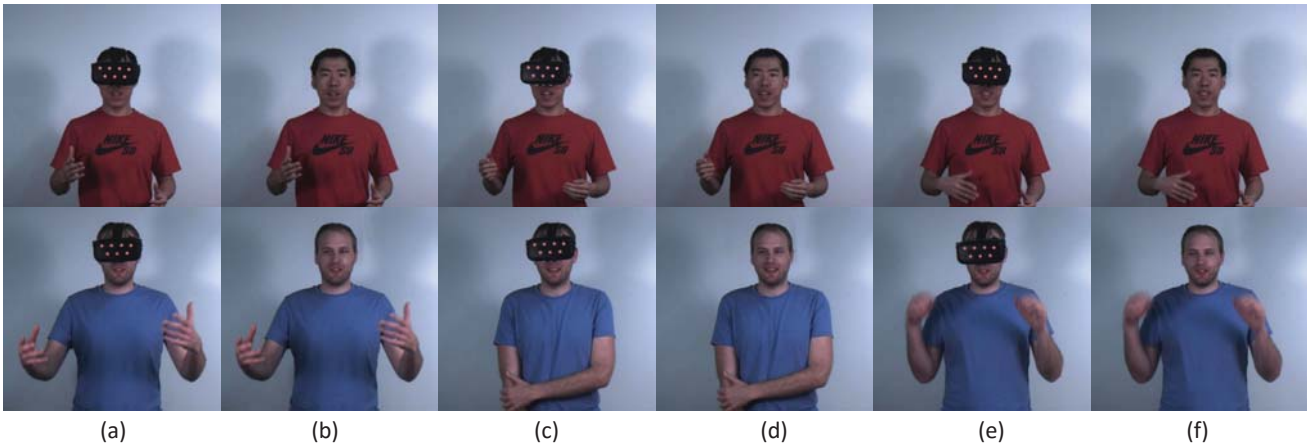

Figure 11: Results of real system(far range). (a),(c),(e) are captured face images with HMD. (b),(d),(f) are the results of our system.

Figure 12: Comparison of warping results by using retrieved image and by using one template. ***from left to right.*** (1) Target face image with HMD; (2) Warped version of (3), aligned with (1); (3) Retrieved reference image in dataset, which has similar head pose to (1); (4) Warped version of (5), aligned with (1); (5) One frontal template image.

Figure 13: Comparison of blending results with and without expression tracking. ***The left*** is the target image with HMD. ***The middle*** is the blending result generated without expression tracking. ***The right*** is the blending result generated with expression tracking.

9.6 Face Synthesis Results

As the ground truth is available in the simulation setup, we can evaluate our expression tracking and eye colorization algorithm by computing the error map between our synthesized image and the ground truth. Figure 14 shows results for different users. The average intensity error is around 5.6 in the mask region based on intensity range from $0 \sim 255$, which indicates that our eye colorization algorithm can produce accurate results. In the simulation setup, the head pose is fixed. We ask users to perform as much expression as they can in an off-line expression database. We calculate the expression weights by using facial landmarks for all the frames in the database. For the query frame with upper face occluded, we also calculate the expression parameters by using our algorithm, then retrieving for the best matched expression in the database. This retrieved image is used to fill the missing part of the query image. Note that in Figure 14, the facial details are reconstructed by using the best matched expression template in the database, this demonstrates the effectiveness of our expression tracking algorithm.

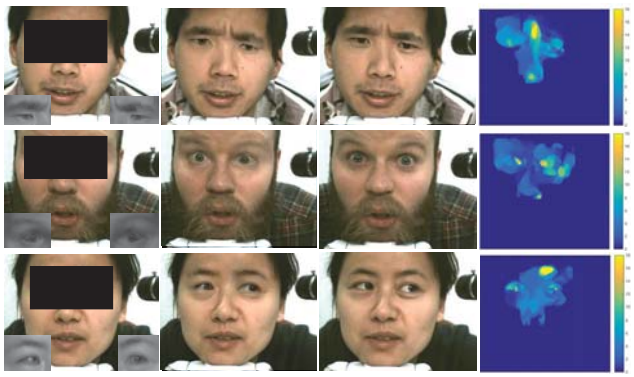

Figure 14: Simulation Results. (***1st column***) are the input images with eye images shown at the bottom of each face image. (***2nd column***) are the synthesized face images by our system. (***3rd column***) are the ground truth images. (***4th column***) are the error maps between ground truth and synthesized image.

Figure 10 shows results for our real setup. We have tested our system on different users with various facial expressions including eye and month movements. Our results demonstrate the effectiveness and robustness of our system. In Figure 11, we also test our system in far range, results show that our approach can faithfully preserve body movements such as gestures, which plays important role in teleconference. *Video results can be found in our supplemental material.*

10 Limitations And Future Work

Our current system cannot deal with strong shadow such as the cast shadow under the HMD shown in Figure 15 (a). One solution is to model albedo as well as 3D geometry in the future to use relighting techniques totally remove the shadow. Our system will fail when the lower face part is also occluded as shown in Figure 15 (b) due to the failure of landmarks detection. In this proposed system, the marker-based tracking will fail due to large head pose and motion(Figure 15(c) and (d)), which will cause the failure of circle detection. Inspired by [5,36], we will replace the markers with QR pattern, which is more robust and works well on large poses.

Figure 15: Failure cases

11 Conclusions

In this paper, we present a novel headset removal technique that is able to synthesize the upper facial region which is occluded by the HMD.We design a hardware system that consists of two NIR cameras capturing the eye regions and one visible-light camera to capture the face image with only lower part visible. We first propose a novel algorithm to align and track 3D head model based on the input image with a large portion of face occluded by the HMD. A novel eye colorization algorithm is then proposed to fuse the gray-scale NIR images with the color image to achieve realistic result. In addition, our approach is capable to faithfully colorize the NIR eye image and remove the color distortion resulted from the non-linear mapping of IR light sensitivity. We have shown a variety of results which demonstrate the efficacy of our proposed framework with photo-real quality.

Acknowledgments

This work is partially supported by the US NFS (IIS-1231545, IIP-1543172), NSFC (No. 61332017, 61305011), National High Technology Research and Development Program of China (No. 2015AA015905).

References

[1] Artec3d, 2013. https://www.artec3d.com/.

[2] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984.

[3] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pp. 586–606. International Society for Optics and Photonics, 1992.

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

[5] X. P. Burgos-Artizzu, J. Fleureau, O. Dumas, T. Tapie, F. LeClerc, and N. Mollet. Real-time expression-sensitive hmd face reconstruction. In *SIGGRAPH Asia 2015 Technical Briefs*, p. 9. ACM, 2015.

[6] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang. 3d deformable face tracking with a commodity depth camera. In *Computer Vision–ECCV 2010*, pp. 229–242. Springer, 2010.

[7] C. Cao, D. Bradley, K. Zhou, and T. Beeler. Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics (ToG)*, 34(4):46, 2015.

[8] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)*, 33(4):43, 2014.

[9] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):41, 2013.

[10] 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*, 20(3):413–425, 2014.

[11] J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, 2004.

[12] R. Du, M. Chuang, W. Chang, H. Hoppe, and A. Varshney. Montage4d: interactive seamless fusion of multiview video textures. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, p. 5. ACM, 2018.

[13] C. Frueh, A. Sud, and V. Kwatra. Headset removal for virtual and mixed reality. In *ACM SIGGRAPH 2017 Talks*, p. 80. ACM, 2017.

[14] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[15] S. Hickson, N. Dufour, A. Sud, V. Kwatra, and I. Essa. Eyemotion: Classifying facial expressions in vr using eye-tracking cameras. *arXiv preprint arXiv:1707.07204*, 2017.

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

[17] A. E. Ichim, S. Bouaziz, and M. Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)*, 34(4):45, 2015.

[18] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874, 2014.

[19] A. Kowdle, C. Rhemann, S. Fanello, A. Tagliasacchi, J. Taylor, P. Davidson, M. Dou, K. Guo, C. Keskin, S. Khamis, et al. The need 4 speed in real-time dense visual tracking. In *SIGGRAPH Asia 2018 Technical Papers*, p. 220. ACM, 2018.

[20] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 689–694. ACM, 2004.

[21] 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 Transactions on Graphics (TOG)*, 34(4):47, 2015.

[22] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):978–994, 2011.

[23] K. Olszewski, J. J. Lim, S. Saito, and H. Li. High-fidelity facial and speech animation for vr hmds. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2016)*, 35(6), December 2016.

[24] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, et al. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pp. 741–754. ACM, 2016.

[25] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, vol. 22, pp. 313–318. ACM, 2003.

[26] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer graphics and applications*, (5):34–41, 2001.

[27] 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*, p. 32. ACM, 2018.

[28] B. Romera-Paredes, C. Zhang, and Z. Zhang. Facial expression tracking from head-mounted, partially observing cameras. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2014.

[29] S. Saito, T. Li, and H. Li. Real-time facial segmentation and performance capture from rgb input. In *European Conference on Computer Vision*, pp. 244–261. Springer, 2016.

[30] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand. Style transfer for headshot portraits. *ACM Transactions on Graphics (TOG)*, 33(4):148, 2014.

[31] K. Suzuki, F. Nakamura, J. Otsuka, K. Masai, Y. Itoh, Y. Sugiura, and M. Sugimoto. Recognition and mapping of facial expressions to avatar by embedded photo reflective sensors in head mounted display. In *2017 IEEE Virtual Reality (VR)*, pp. 177–185. IEEE, 2017.

[32] M. Takemura, I. Kitahara, and Y. Ohta. Photometric inconsistency on a mixed-reality face. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 129–138. IEEE Computer Society, 2006.

[33] M. Takemura and Y. Ohta. Diminishing head-mounted display for shared mixed reality. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, p. 149. IEEE Computer Society,

2002.
[34] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics (TOG)*, 34(6):183, 2015.
[35] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.
[36] J. Thies, M. Zollöfer, M. Stamminger, C. Theobalt, and M. Nießner. FaceVR: Real-Time Facial Reenactment and Eye Gaze Control in Virtual Reality. *arXiv preprint arXiv:1610.03151*, 2016.
[37] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+ 3d active appearance models. In *CVPR (2)*, pp. 535–542, 2004.
[38] Y. Zhao, W. Chen, J. Xing, X. Li, Z. Bessinger, F. Liu, W. Zuo, and R. Yang. Identity preserving face completion for large ocular region occlusion. *The British Machine Vision Conference (BMVC)*, 2018.
[39] Y. Zhao, X. Huang, J. Gao, A. Tokuta, C. Zhang, and R. Yang. Video face beautification. In *ICME*, pp. 1–6, 2014.