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

1090

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

A head-mounted display (HMD) could be an important component of augmented reality system. However, as the upper face region is seriously occluded by the device, the user experience could be affected in applications such as telecommunication and multi-player video games. In this paper, we first present a novel experimental setup that consists of two near-infrared (NIR) cameras to point to the eye regions and one visible-light RGB camera to capture the visible face region. The main purpose of this paper is to synthesize realistic face images without occlusions based on the images captured by these cameras. To this end, we propose a novel synthesis framework that contains four modules: 3D head reconstruction, face alignment and tracking, face synthesis, and eye synthesis. In face alignment and tracking, we propose a novel algorithm that can robustly align and track a personalized 3D head model given a face that is severely occluded by the HMD. In eye synthesis, in order to generate accurate eye movements and dynamic wrinkle variations around eye regions, we propose another novel algorithm to colorize the NIR eye images and further remove the "red eye" effects caused by the colorization. Results show that both hardware setup and system framework are robust to synthesize realistic face images in video sequences.

1. Introduction

With the recent surge of interests in virtual reality (VR) and augmented reality (AR) techniques, it is increasingly common to see people wearing head-mounted displays. Often taunted as a new means for social interactions, the form factor of these HMDs, however, severely limit one common form of interactions, that is face-to-face communications, either in the same physical space, or connected via imaging techniques (e.g., video teleconferencing). In the foreseeable future, HMDs that offer an immersive or seamless experience will severely occlude a large portion of the face. As a result, it is difficult or even impossible for other people to identify the user, facial expression, and eye gazes.

We are certainly not the first to identify this problem. There are several recent research papers that aim to address this problem. They can roughly be divided into two categories. The first is to find ways to track the expression, using cameras or other sensing devices embedded inside the helmet such as in [OLSL16, LTO*15, TZS*16b]. The tracked expression is then used to drive an avatar. While very impressive results, in both the tracking accuracy and the realism of the final synthesized face images, have been demonstrated, approaches in this category are limited to providing a talking head experience, body movement and gestures, which are also important for communications are missing. The second category aims to inpaint the occluded facial part, making it possible to present the full picture as if the subject is not wearing the HMD at all. This approach is quite difficult since the occluded part is significant and we are very sensitive to artifacts on the face. We have found only one paper following this direction. In [BAFD*15], Burgos *et al.* first train a regression model of a subject's expressions, then based on the unclouded part (e.g., lower face) to synthesize a complete face image with expressions. Limited results have been

presented and strong artifacts are shown in the reconstructed images.

In this paper, we present a novel framework in the second category to digitally remove the HMD. We use a main stationary camera to capture the subject, as in a typical video conference setup. In addition, we add two small near-IR cameras inside the HMD to track the eye's motion. The goal is to synthesize face part occluded in the main camera image, including pasting the eye images to the correct position and restoring hairline compressed by the straps. To do that, we first build a personalized 3D face model of the subject by using structure-from-motion and morphable model. Using the reconstructed model, we precisely track the subject's head pose and expressions at run-time. The tracked expression information, eye images, as well as images of the subject without wearing the HMD, are used together to fill in the occluded part.

Since we are literately putting different pieces of face parts together, accuracy is of paramount importance. The most significant innovation in this paper is our novel system calibration, tracking, and image warping techniques. In addition we have developed a novel method to colorize eye regions synthesized from NIR cameras and refine them by removing "red eye" effects. Our method is superior than standard red-eye removal method since the eye image is captured under near-IR illumination in which the eye actually appears differently than from regular illumination. As shown in Figure 1, our system has been able to produce photo-realistic results.

The rest of the paper is organized as follows. In Section 2, we discuss related works. Section 3 presents our hardware setup and the method we used to calibrate our system. Our framework and algorithms are described in Section 4, 5, 6, 7, 8. Experimental results, limitation and future works, conclusions are given in Section 9, Section 10 and 11 respectively.

2. Related Work

Analysis and synthesis of face expressions have been studied in the past few decades. There are various algorithms have been proposed. For instance, active appearance models (AAM) and 3D morphable models have been successfully used in many applications to model shape and texture variations of faces [XBMK04,BV99]. Depth sensors are also widely used to reconstruct 3D face models in recent years. Cai *et al.* presented a deformable model fitting algorithm to track 3D face model using a commodity depth camera [CGZZ10]. Ichim *et al.* reconstructed personalized textured avatar from handheld video, which could further be used for tracking and animation in [IBP15]. In [TZN*15], Thies *et al.* proposed an reenactment algorithm to transfer facial expressions from one person to another in real-time assuming no occlusions present. A RGB-D sensor is used to estimate and track face model, head pose, and illumination. Thies *et al.* further extend their reenactment to only use RGB sensor in [TZS*16a]. It is easy to find out that most of these algorithms are not designed for HMDs. Therefore, they often have either different inputs or outputs comparing with our system and algorithms. For example, some of them may not be robust under serious occlusions. Some of them output animations of 3D face models, in which eye gazes are not important.

A few research has been done recently to drive 3D avatars for users with HMD. In [RPZZ14], Romera-Paredes *et al.* adopted an experimental setup that has two visible light cameras which pointing towards eye regions from oblique angles to capture the eye movement. They built a regression framework from the the captured partial face images as input to the blending weights of personalized blendshapes. Multiple machine learning algorithms, such as ridge regression and convolutional neural networks are applied in their framework. In [LTO*15], Li *et al.* develops a novel HMD that uses electronic materials (strain sensor) to measure the surface strain signals and RGB-D camera to track visible face regions. A linear mapping is trained between the blendshape coefficients of the whole face and the vector that concatenates strain signals and the blendshape coefficients of the visible face part. This mapping is then used to animate virtual avatars. In [OLSL16], they propose an approach for 3D avatar control based on RGB data in real-time. The mouth and eyebrow motion is captured separately by an external camera which attached to HMD and two IR cameras mounted inside the HMD. This data is further used to train a regression model that maps the inputs to a set of coefficients of a parametric 3D avatar. All the above approaches are aimed at 3D parametric avatar driven for HMD users.

Recently, In [TZS*16b], Thies *et al.* propose a novel approach for real-time gaze-aware facial reenactment for user's with HMD. They use a RGB-D camera to capture and track the mouth motion and two internal IR cameras to track the eye gazes. They then reconstruct 2D face images by using the tracked expression and pre-recorded videos of the user with eye-gazed corrected. Although they can reconstruct the full face in photo-realistic videos by self-reenactment,their result will lose information in original inputs, like background, head pose, gestures and other information. In [BAFD*15], Burgos *et al.* build a system to reconstruct face with HMD in 2D videos by using a personalized textured 3D model. They use a RGB camera to capture face videos and track the expressions, then the textured model is projected and blended with the remaining mouth part. Their approach do not handle eye gaze and eye movement. In our approach, we will reconstruct HMD faces in photo-realistic videos with ground truth eye movement and the visible information in original input will be faithfully reserved.

In our proposed algorithm, we also integrate various techniques such as landmark detection [CWWS14], colorization [RAGS01, LLW04, SPB*14], and feature extraction [KS04]. Details are provided in corresponding sections.

3. Hardware Setup and Calibration

3.1. Hardware Setup

We have built two prototypes for experiments and validation. Our first system is a fixed setup, as shown in Figure 2 right. It consists of three cameras. The middle one is a color camera with a resolution of 1280×960, 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. To use this setup, the user is expected to put her/his face on the chin reset to maintain the relative transformation between her/his face and all the cameras.

The second system is a mobile one (Figure 2 Left). We use a VR-headset case, one of these types that allow a user to insert a mobile phone to create a low-cost head-mounted display (HMD). We insert two small NIR cameras inside the shell to observe the eye region. We also have two micro IR LEDs besides the two cameras. It should be noted that since our camera/LED set is not small enough, we do not have the phone inserted during all of our experiments. This limitation could certainly be solved by better (and more costly) engineering. In this mobile setup, a user should wear our modified headset as usual, a fixed RGB camera is used to observed the user. This is similar to a regular video conference setup except that the user's face is severely occluded physically and can move freely in any poses.

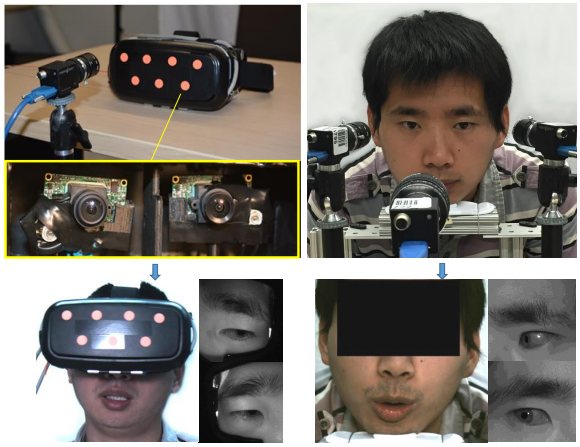

Figure 2: *Two experimental systems we have built.* ***(Left Column)****our mobile setup. Two small cameras are inside the VR-display case.* ***(Right Column)*** *our simulation setup with three cameras, two for eyes and one for the entire face. The occlusion on face is synthesized by applying a mask.*

3.2. Calibration

We first have to geometrically calibrate all cameras. While the fixed setup is easy to deal with, the mobile 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 extract the extrinsic of NIR cameras as long as we know the extrinsic of HMD. We describe our procedure for the mobile system calibration and tracking. We first intrinsically calibrate all the cameras using standard techniques. We then print out a small checkerboard pattern and attached it to the VR-display case so that one half of the patterns are visible to the face camera and the other half is visible to the NIR eye camera. Since the size of the grid is known, we can estimate the pose of these cameras using a Perspective-n-point algorithm (PnP) (e.g., [LMNF09]). Let's denote the points on the checkerboard pattern as $\mathbf{X}_c$ and the relative poses of the face camera and the eye cameras as $M_{c\to f}$, $M_{c\to e^l}$, and $M_{c\to e^r}$ respectively. Furthermore, we put a number of color dots on the front side of the VR-display case. These dots are used for tracking. They are co-planar and their relative positions are measured. These points, denoted as $\mathbf{X}_h$, define their own coordinate space. Using PnP, the face camera's pose $M_{h\to f}$ in the HMD ($\mathbf{X}_h$) space can be estimated. Using the face camera as a bridge, we can now calculate the eye-camera's pose in the space of $\mathbf{X}_h$. For the left eye, it is $M_{c\to e^l}M^{-1}_{c\to f}M_{h\to f}$. Now we can remove the checkerboard pattern (since it will occlude the eye cameras). At run time, the face camera will track the HMD's pose using these color dots and therefore the pose of the eye cameras. The involved coordinate transforms and a photo of our calibration patterns are shown in Figure 3.

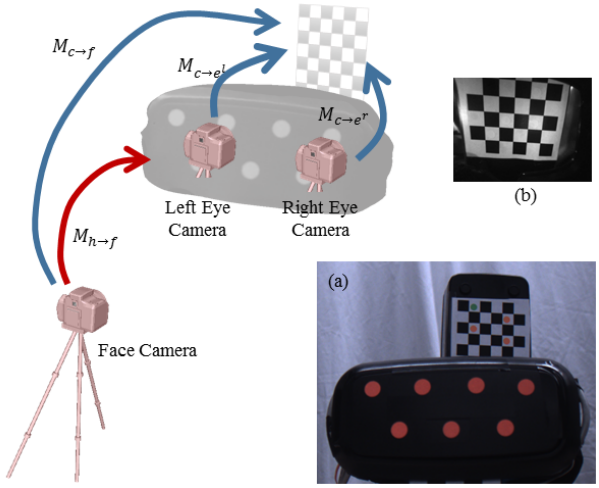

Figure 3: *An illustration of our calibration procedure for the mobile setup. One checkerboard is placed behind the HMD. The transformations between different cameras/coordinate systems are labeled. Inset (a) shows an image captured by the face camera and inset (b) shows an image captured by one of the eye cameras.*

4. System Overview

Our system consists of four modules as shown in Figure 4. We reconstruct a personalized 3D animatable head model from a video sequence captured off-line in the first module (Section 5). The 2D facial landmarks, 3D sparse point cloud and morphable model fitting are integrated together in our optimization algorithm to obtain an accurate head model. In the second module (Section 6), we propose a novel algorithm to align 3D head model to the face image that has been severely occluded by the HMD. Instead of fitting the head model to the small lower face portion for each image frame, our algorithm first estimates the transformation between the HMD and the head model, which is fixed once the user puts on the HMD. Then, by simply tracking of the HMD pose, we can get the head pose robustly with the estimated transformation described above.

The facial expression coefficients are then computed to capture the expression in each frame. 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 from the image template. The obvious artifacts (e.g., "red eye") in the eye regions also are removed in this module. In order to generate realistic face images without occlusions, in the fourth module (Section 7), we apply a boundary constrained warping algorithm to first align the reference image with the target occluded face image and then compose the complete face from different sources by a mask.

5. 3D Head Reconstruction

In the off-line data acquisition stage, we record a video sequence of a 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 structure from motion (SfM) to estimate a sparse point cloud and projection matrices [HZ03]. A bi-linear face morphable model described in [CWZ*14] is then used to reconstruct a dense 3D model M with $11K$ vertices from the sparse point cloud,

$$M = B \times_2 C_{id} \times_3 C_{exp}, \tag{1}$$

where $B \in \mathcal{R}^{11K\times 50\times 25}$ is the reduced core tensor, $C_{id} \in \mathcal{R}^{50}$ and $C_{exp} \in \mathcal{R}^{25}$ stand for the column vectors of identity weights and expression weights respectively. As we assume the neutral expression during the reconstruction, the expression weights C_{exp} are fixed and only identity weights C_{id} are estimated.

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

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

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^s are the k_{th} pair of 3D vertices in the dense 3D head model and sparse point cloud. In each iteration, N 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 [KS14]. The cost function is defined as,

$$E = \sum_{i=1}^{N}\sum_{j=1}^{K} \|P_i M_j - l_{ij}\|^2 + \lambda \sum_{i=1}^{50} ((C_0^i - C_{id}^i)/\theta)^2 \tag{3}$$

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 3 is a regularization term that makes the estimated head model M close to the head model estimated from equation 2, which is denoted as C_0. This term also prevents the the geometry from degeneration and local minima.

6. Face Alignment and Tracking

As the face is severely occluded by the HMD, the alignment could be highly inaccurate if we align the 3D head model with the face image directly according to remaining visible facial features. In this section, we present a novel approach based on our hardware configuration and calibration.

6.1. Facial Landmark Detection

As we need to use facial landmarks in our alignment, three landmark detectors are trained on occluded face image and eye images separately(left and right eye, lower facial part separately). As illustrated in Figure 5, we use 5 landmarks for eyebrow and 6 landmarks for eye boundary in each eye image, 20 landmarks for mouth, 5 landmarks for nose base and 11 landmarks for lower face boundary in the occluded face image. The cascaded learning framework described in [CWWS14] is applied. In this learning framework, simple pixel-difference features are extracted and two-level boosted regression is applied. In the internal level of the regression, a set of primitive regressors (*e.g.*, ferns) are trained. Few thousands of training eye images are obtained by cropping labeled face images of the LFW data set [HRBLM07]. As the eye is often located in the middle of the captured image without in-plane rotations in our hardware setup and we could provide bounding box of lower facial part by tracking HMD, we achieve accurate eye and lower face landmarks predictions.

6.2. Initial Alignment

After the offline calibration described in section 3.2, it is robust to track the HMD's pose (e.g., $P_{h\to f}$) in real time. The transformations (e.g., $M_{e_l\to h}$ and $M_{e_r\to h}$) between eye cameras and the HMD are also fixed after the calibration. However, the transformation between the head and the HMD is different for different 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 4.

$$E_{init} = E_f + \lambda E_e \tag{4}$$

where λ is the weight to control E_e. 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.

$$E_f = \sum_i d(\mathbf{x}_i, P_{h\to f} R_* T_* \mathbf{X}_i)^2 \tag{5}$$

where $\mathbf{x}_i$ and $\mathbf{X}_i$ are the 2D visible facial landmarks in the ith image

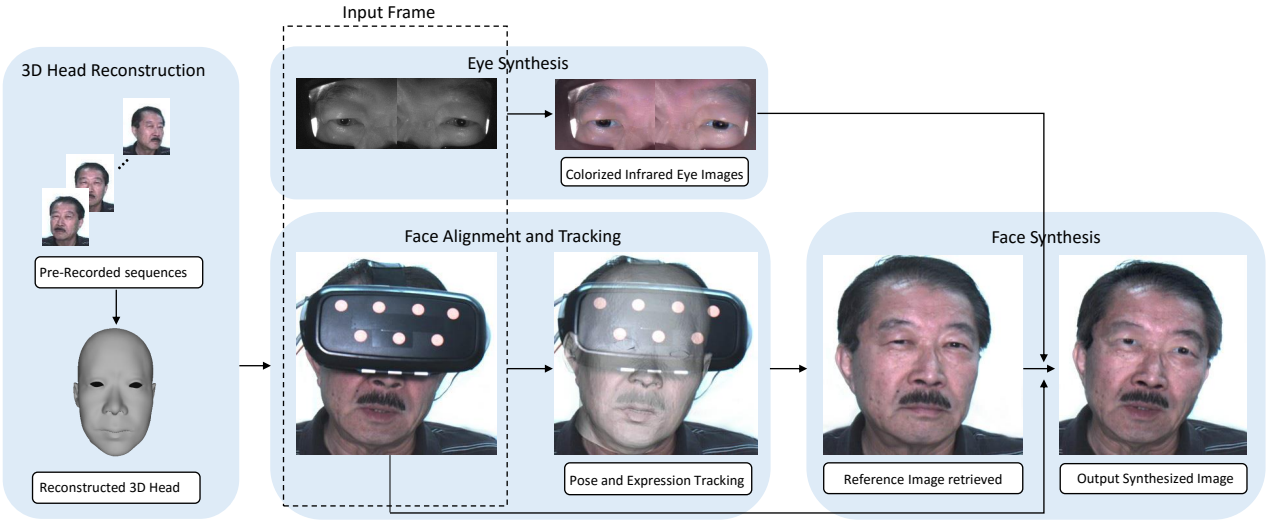

Figure 4: *Conceptual overview of our system. From the first stage to the fourth stage, our goal is to synthesize a photo-realistic face image without occlusion.*

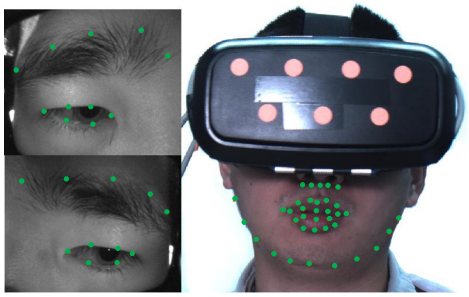

Figure 5: *Illustration of the landmarks set we adopted in our system.*

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

$$E_e = \begin{array}{c} \sum_i d(\mathbf{x}_i, P_{h\to f} M_{e^l\to h} R_* T_* \mathbf{X}_i)^2 + \\ \sum_i d(\mathbf{x}_i, P_{h\to f} M_{e^r\to h} R_* T_* \mathbf{X}_i)^2 \end{array} \quad (6)$$

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 3D point $\mathbf{X}_i$ on the head model is computed by finding the nearest neighbor of the intersection point between the head model and the ray back projected from $\mathbf{x}_i$. The Levenberg-Marquardt iteration method is applied to optimize this objective function.

6.3. 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 estimate the expression weights C_{exp} of the bi-linear face model described in Equation 1. Note that the identity weights C_{id} are fixed in this step. The expression weights are estimated based on the energy function,

$$E_{exp} = E_f + \lambda_1 E_e + \lambda_2 E_t + \lambda_3 E_s, \quad (7)$$

where E_f and E_e are defined in Equation 5 and 6 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,

$$E_t(C_{exp}) = \| C_{exp}^{t-1} - C_{exp}^{t} \|^2, \quad (8)$$

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,

$$E_s = \sum_{i=1}^{N=25} (C_{exp,i}/\theta_i)^2 \quad (9)$$

where θ is the mean vector. 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 energy function E_{exp} is minimized by the least squares method in real time. The weights we used to balance the terms in our setup is $\lambda_1 = 2$, $\lambda_2 = 2$, and $\lambda_3 = 0.7$.

Our alignment and tracking algorithm is summarized in Algorithm 1.

Algorithm 1: Head Pose and Expression Tracking

Data: Image frames captured by three cameras and a personalized 3D head model.

Result: Alignment between the 3D model and image frames, $P_{h\to f}$, R_*, T_*, C_{exp}

Initial Alignment

1. Estimate $P_{h\to f}$ for each frame based on the color markers.
2. Facial landmark detection on both occluded face image and eye images.
3. Initialize R_* to the identity matrix and the translation vector to $[0,0,dz]$.
4. Back project facial landmarks $\mathbf{x}_i$. Corresponding $\mathbf{X}_i$ are computed by finding the nearest neighbor of intersection with the 3D head model.
5. Estimate R_* and T_* by minimizing cost function in Equation 4.
6. Apply transformation to the 3D head model with new R_* and T_* and go to step 4, until converge.

Real-time Alignment

1. Estimate $P_{h\to f}$ for each time frame based on the color dots.
2. Facial landmark detection on both occluded face image and eye images.
3. Estimate C_{exp} for each time frame by minimizing cost function in Equation 7.

7. Face Synthesis

In our face synthesis, we first search for a template image from the data set we have captured off-line, which contains similar head pose as 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 [BAFD*15], the author synthesized the occluded face by rendering of the textured model. This model based method produce face images with strong artifacts especially on the boundary because they don't model hair motion, illumination. Thus 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 10. 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 6.2. 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.

$$D = \| H_q - H_r^i \|^2 + w_1 \| L_r^i - L_{r-1} \|^2 + w_2 \| S_r^i - S_{r-1} \|^2, \quad (10)$$

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, we project 3D head models to generate dense 2D face meshes. The face mesh for the reference image is then warped to the face mesh for the query image. Particularly, as we tracked the expression of the query image, the deformation caused by large expressions reflects to 2D mesh which will enhance the alignment accuracy of the reference image after warping. In order to warp the reference image naturally without obvious distortions, we divide the image to $n \times m$ uniform grid mesh. The energy function is defined as,

$$E = E_d + \alpha E_s + \beta E_b + \gamma E_h \quad (11)$$

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 [ZHG*14]. As we also want to align the silhouette of the warped template image with the query image to avoid artifacts on the face boundary when blending the warped image and the query image. Therefore, we introduce another term E_h to constrain the silhouette. Denote $\hat{P}_s$ and P_s as a pair of 2D correspondence points on silhouette of template image and query image. The template image is divided into $n \times m$ uniform grid mesh $\hat{V}$, the warping problem is to find warped version V of this grid mesh. Then E_h can be formulated as bellow:

$$E_h = \sum_{i=1}^{N} \| w_i V_i - P_i \|^2 \quad (12)$$

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. We define the correspondence pair $\hat{P}$ and P by first finding the silhouette points on the query image, then we use the 3D model as a bridge to find the corresponding 2D points on the template image. Figure 6 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.

In system calibration, we obtained the 3D transformation between eyes cameras and the HMD device. After the initial alignment, we aligned the head model to the HMD to obtain the projection matrix from the head model to eye images. Therefore, we can also easily warp the NIR eye images to the query image. 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

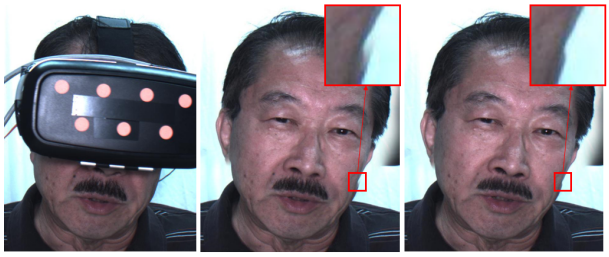

Figure 6: *Illustration of the effect of silhouette constraints. (**Left**) is the input target frame needs to be reconstructed. (**Middle**) is the blending result by warping the template without term E_h. (**Right**)is the blended result by warping the template with term E_h. The result shows that this term forces the warped template image and target image to align on the boundary to eliminate the artifacts.*

Laplacian blending approach [AAB*84]. Figure 7 shows the formation of the final result. Note that in the rightmost image of Figure 7, we further apply the background replacement to remove the HMD region and wires that are far from the head and not covered by the mask image.

Figure 7: *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 process 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., “red eye") during the colorization.

8.1. Colorization

We use the Lab color space as it is close to human visual conception 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 [RAGS01, SPB*14] are applied and evaluated to transfer from I to C_L. The first algorithm [RAGS01] is a straightforward histogram transfer based on the standard deviations and mean values of I and M_L. In the second algorithm [SPB*14], the images I and M are aligned based on the landmarks and the SIFT flow [LYT11]. 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 [LLW04]. The color in the channel a is computed by minimizing the following energy function

$$E(a) = \sum_p ((a(p) - \sum_{q \in N(p)} w_{pq} a(q))^2 + \alpha \sum (a(p_m) - P_m)^2 \quad (13)$$

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 [LLW04]). 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

$$error = \frac{|I_p - I_c|}{I_c} \quad (14)$$

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.*, “red eye" effects) as shown in Figure 8(b). One possible reason is that we treat the NIR image as the grayscale image. We found that, 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 “red eye" 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 [Dau04].

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 "red eye" 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.

$$E_d = \sum_{p_m} (C''_L(p_m) - C'_L(p_m))^2 \tag{15}$$

$$E_s = \sum_{p} ((C''_L(p) - \sum_{q \varepsilon N(p)} w_{pq} C''_L(q))^2 \tag{16}$$

$$E_b = \sum_{p} (|N_p| C''_L(p) - \sum_{q \in \Omega} C''_L(q) - \sum_{q \notin \Omega} C_L(q) - \sum_{q \varepsilon N_p} V_{pq})^2 \tag{17}$$

$$E = E_d + \alpha_1 E_s + \alpha_2 E_b \tag{18}$$

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 (the seed colors are selected using the criteria described in Equation 14), $C'_L(p_m)$ 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

$$w_{pq} = 1 + \frac{1}{\sigma_p^2}(C_L(p) - \mu_p)(C_L(q) - \mu_p) \tag{19}$$

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 that makes the color are smoothly transformed among its neighborhood. The last term E_b is the boundary term that is inspired by the gradient image editing [PGB03]. This term is equivalent to the Poisson equation with Dirichlet boundary conditions. The refinement algorithm is summarized below.

Figure 8 shows one group of eye images after refinement. In this Figure, we can find that image (b) contains "red eye" effects, image (c) has strong artifacts around boundaries of segmentation, and image (e) is the result using all three terms and "red eye" effects are removed.

9. Experiments

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

Algorithm 2: Refinement of eye regions.

Data: The eye image C that is the color image after colorization and contains "red eye effects".
Result: Refined eye image C''.

1. Detection of iris and pupil boundaries based on the intergrodifferential operator in [Dau04].
2. Segmentation of pupil, iris, and sclera using eye landmarks and boundaries of iris and pupil.
3. Histogram transformation for each category and each color channel. C' is denoted as the output after transformation.
4. Selection of seed pixels p_m based on Equation 14.
5. Minimization of Equation 18.

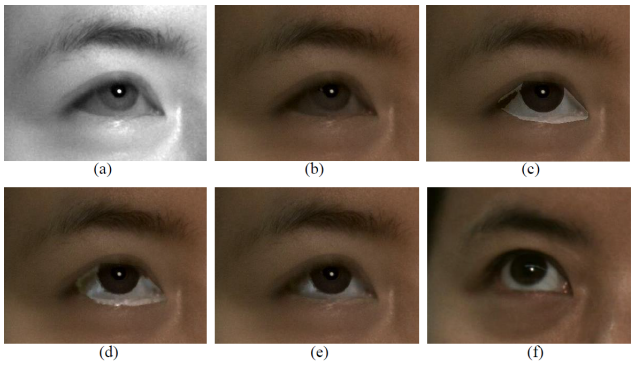

Figure 8: *Results of eye refinement. (a) near-infrared image. (b) result after colorization (with "red eye" effects). (c) result only using the date term E_d. (d) result using data and smoothness terms (E_d and E_s). (e) result using all three terms (E_d, E_s, and E_b). (f) reference eye color images.*

9.1. Runtime

Our implementation on CPU takes around 568 ms to process one frame on Intel Core i7-4710 CPU(3.4GHz) with the color image in resolution of 1280×960 and two NIR images in resolution of 640×480. Table 1 shows the runtime of major components in our system. The face synthesis component consists of reference image retrieval, face warping and final blending, which is the most time consuming module. We believe that, by using the parallel processing power of GPUs and reducing the image resolution, we can achieve real-time performance.

Table 1: *Runtime of Different Modules*

Tracking	Eye Colorization	Face Synthesis
8ms	160ms	400ms

9.2. Evaluation of 3D reconstruction

We first scan 3D head models with a high resolution structured light 3D scanning system which has an average reconstructing error less than 2mm. More details about this scanner could be found

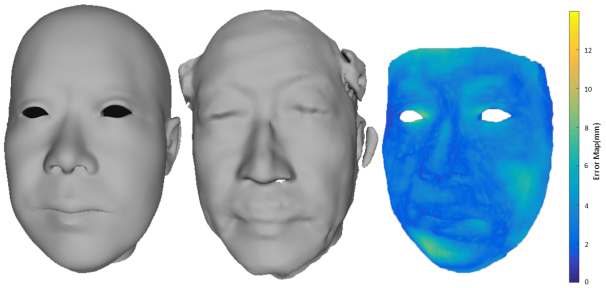

Figure 9: *Evaluation of 3D reconstruction.* ***from left to right*** *(1) The head model reconstructed by our algorithm. (2) The model scanned by a structured light 3D scanning system. This model serves as ground truth. (3) The error map between our model and the ground truth.*

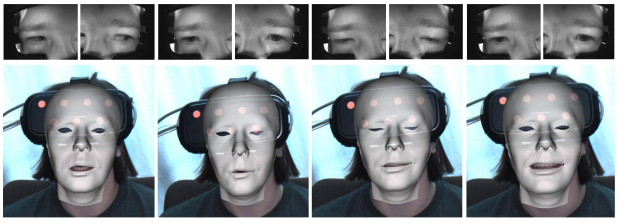

Figure 10: *Evaluation of tracking with HMD. The 3D model is overlaid on the original input frame. Facial expression, especially the eye blinks, are tracked robustly by our algorithm.*

in [Zha15]. These 3D models serve as ground truth in our evaluation. To measure the difference between the reconstructed model and the ground truth, we first roughly align them by computing a transformation between them using 3D facial landmark correspondences and then the ICP algorithm [BM92] is used to refine the alignment. In order to evaluate the surface distance, we define each 3D point on the reconstructed model p and its corresponding point $\widetilde{p}$ as a pair, in which $\widetilde{p}$ is the first intersection on the ground truth mesh along the normal direction of p. The Euclidean distances are then calculated for all the pairs. The mean error distance between the reconstructed model and the ground truth is 2.926 mm. Figure 9 shows one example of our reconstructed model, the corresponding ground truth, and the error map.

9.3. Evaluation of Face Tracking

Our algorithm described in section 6 can robustly track 3D face models in real-time. As shown in Figure 10, these models could contain various facial expressions, such as eye blinks and mouth movements. In order to further evaluate the tracking performance, we project a virtual pattern to each input image frame. As shown in Figure 11, we can find that the virtual pattern is deformed smoothly and consistently with various mouth movements. **Video of Tracking results can be found in supplementary material.**

Figure 11: *Evaluation of tracking performance with a projected virtual pattern. The virtual pattern is deformed smoothly and consistently with the mouth movements.*

9.4. Evaluation of Eye Synthesis

It is essential to generate accurate eye movements in the final synthesized face image. Figure 12 demonstrates our results of eye synthesis for three different users. The first column are the reference images retrieved from the pre-recorded image frames. The second column contains the input NIR images. The third column contains images after colorization which is the first step of our eye synthesis. The color appearances have been adjusted to be very similar to the reference images. However, the “red eye" effects are also quite obvious in these images. After the second step of our eye synthesis algorithm, the “red eye" effects are removed and more realistic eye images are generated (the fourth column).

9.5. Evaluation of Face Warping

Instead of using one template frontal face image for all the frames, we retrieve the data set for the best matched reference image for each frame based on head pose similarity and time space consistence as described in Section 7.1. Figure 13 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 shapes. The reason is that we only have control points inside face region during warping and we need to keep the background unchanged.

In our system, after the calibration in Section 3.2 and initial alignment in Section 6.2, 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, large mouth motions will deform the 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 14 shows the comparison of blending results between tracking with and without expressions. It is clearly that the middle image has thinner cheek than the right one and the left one(ground truth) as we warped the reference image guided by a neutral 3D model. Reflected in blending results, the absence of expression will lead to artifacts on face boundary.

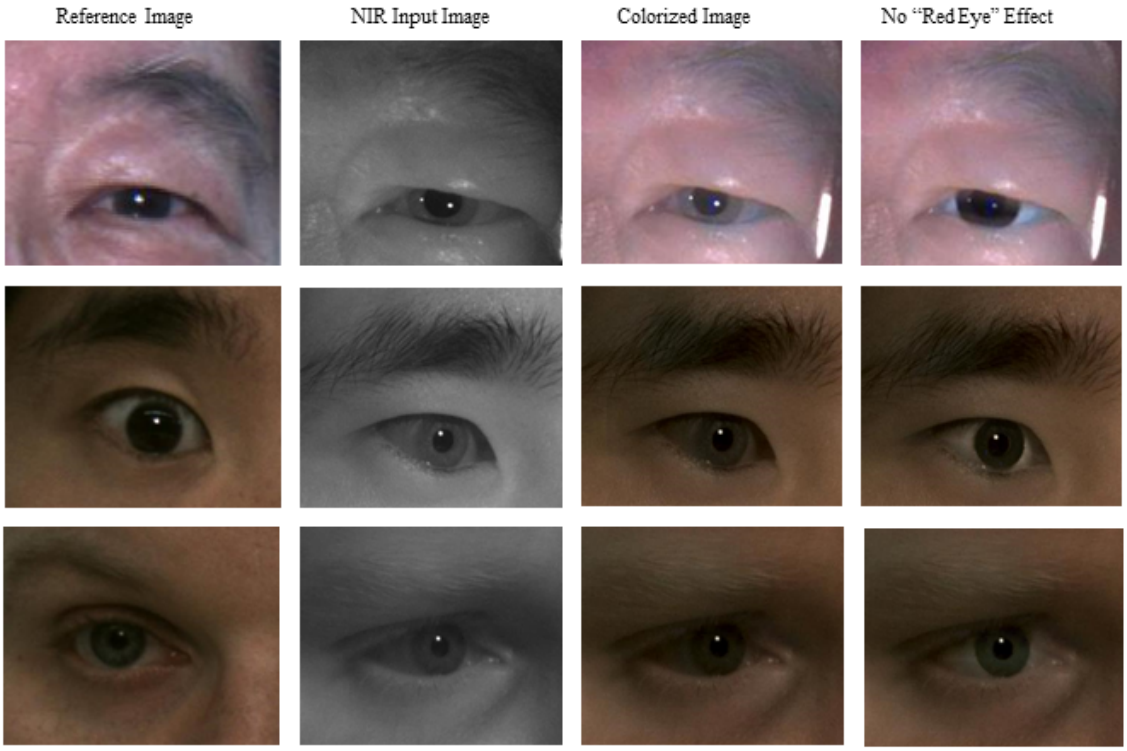

Figure 12: *Results of eye synthesis. (**1st column**) Reference images retrieved from pre-recorded image frames. (**2nd column**) Input of NIR images. (**3rd column**) Results after colorization. (**4th column**) Results after refinement of eye regions.*

Figure 13: *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 14: *Comporison of blending results with and without expression tracking. **The left** is the target image with HMD. **The middle** is the blending result generated from the reference warped from 3D model without expression. **The right** is the blending result generated from the reference warped from 3D model with expression.*

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 15 shows results for different users. The average intensity error is around 5.6 under the area of mask 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 15, 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 16 shows results for our mobile setup. We have tested our system on different users with various facial expressions including eye and month movements. Our results demonstrate the effective-

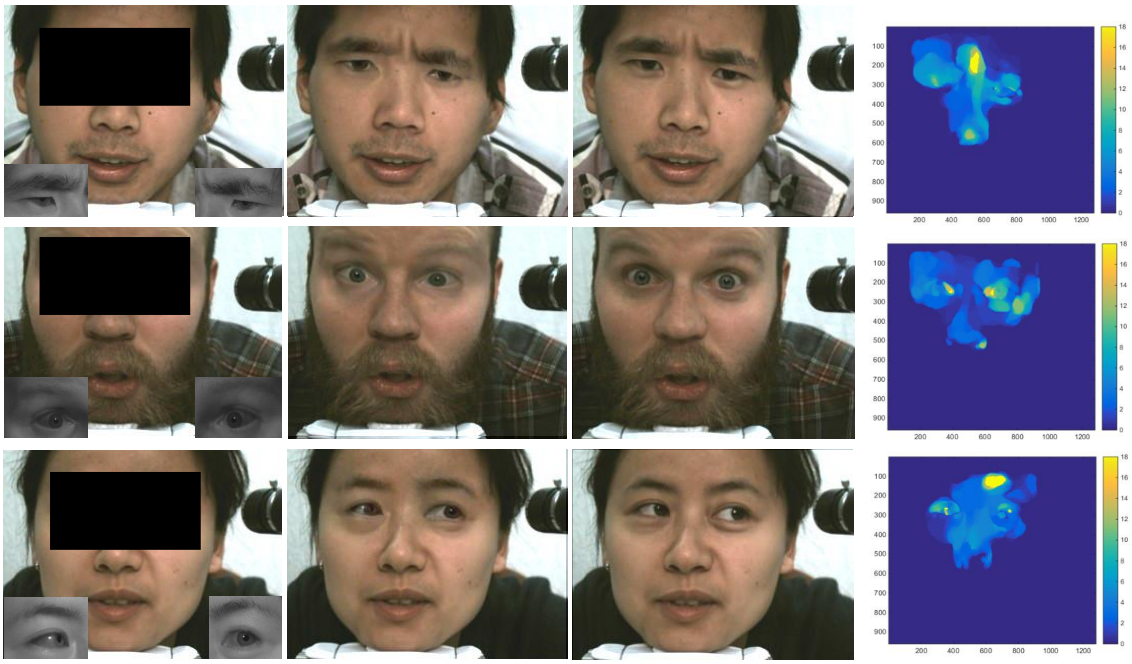

Figure 15: *Simulation Results. (**1st column**) are the input images with eye images shown at the bottom of each face image. (**2nd column**) are the synthesized face images by our system. (**3rd column**) are the ground truth images. (**4th column**) are the error maps between ground truth and synthesized image.*

ness and robustness of our system. **More results including videos can be found in our supplementary material**.

10. Limitation and Future Works

Although Mask-off is able to solve a challenge problem in VR, it is one of the first stepping stones in a new area thus limited in several aspects.

Our current system requires a personalized database each time before the user puts on the HMD to ensure the illumination environment unchanged between reference image and target image. We also apply histogram adjustment on reference image to match the color tone since the lower face will be darker under the shadow of HMD. However, we don't really deal with the shadow and illumination changes. One of the future work is to model the albedo as well as 3D geometry, so that we could handle the lighting environment variation. For each user, the database only needs to be captured once and can be applied to different lighting conditions by using relighting techniques.

Although the current system produces convincing results, there are still artifacts on the face boundaries, eye regions after blending. To reduce artifacts, precise processing are required in every steps: reconstruction, tracking, warping and blending. We believe that the introduction of deep neural network can provide guidance for robust warping and blending. Face synthesis in our context can be regarded as a face inpainting problem. Li *et al.* explored the possibility in [LLYY17] to complete the face image with occlusion. We believe that with an initial low resolution complete face image provided by CNN network, the artifacts on the high resolution results of our approach caused by boundary misalignment and illumination variance will be reduced.

In this proposed system, we use color markers to track the HMD. These markers are assumed in the same plane. This method requires the frontal panel of HMD to be a planar and the head pose should not be too large, which limits the application of our method. Inspired by [TZS*16b, BAFD*15], we will replace the markers with QR pattern, which is more robust to large poses.

One important limitation of our system is the run-time, we cannot achieve real-time performance required by proposed scenarios. we would like to explore different alternates to speed up the framework and achieve real-time performance with modern GPU.

11. Conclusions

In this paper, we tackle the face synthesis problem in which the upper face region is severely occluded by the HMD. We design a novel system framework 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. In order to synthesize realistic face images, we present two novel algorithms in our framework. Firstly, we 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. Secondly, a novel approach to synthesize eye regions is presented. In this approach, we colorize the NIR eye images and further remove the "red eye" effects.

Figure 16: *Results of our mobile system.*

References

[AAB*84] ADELSON E. H., ANDERSON C. H., BERGEN J. R., BURT P. J., OGDEN J. M.: Pyramid methods in image processing. *RCA engineer 29*, 6 (1984), 33–41. 7

[BAFD*15] BURGOS-ARTIZZU X. P., FLEUREAU J., DUMAS O., TAPIE T., LECLERC F., MOLLET N.: Real-time expression-sensitive hmd face reconstruction. In *SIGGRAPH Asia 2015 Technical Briefs* (2015), ACM, p. 9. 1, 2, 6, 11

[BM92] BESL P. J., MCKAY N. D.: Method for registration of 3-d shapes. In *Robotics-DL tentative* (1992), International Society for Optics and Photonics, pp. 586–606. 9

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 187–194. 2

[CGZZ10] CAI Q., GALLUP D., ZHANG C., ZHANG Z.: 3d deformable face tracking with a commodity depth camera. In *Computer Vision–ECCV 2010*. Springer, 2010, pp. 229–242. 2

[CWWS14] CAO X., WEI Y., WEN F., SUN J.: Face alignment by explicit shape regression. *International Journal of Computer Vision 107*, 2 (2014), 177–190. 2, 4

[CWZ*14] CAO C., WENG Y., ZHOU S., TONG Y., ZHOU K.: Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (2014), 413–425. 4

[Dau04] DAUGMAN J.: How iris recognition works. *Circuits and Systems for Video Technology, IEEE Transactions on 14*, 1 (2004), 21–30. 7, 8

[HRBLM07] HUANG G. B., RAMESH M., BERG T., LEARNED-MILLER E.: *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007. 4

[HZ03] HARTLEY R., ZISSERMAN A.: *Multiple view geometry in computer vision*. Cambridge university press, 2003. 4

[IBP15] ICHIM A. E., BOUAZIZ S., PAULY M.: Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG) 34*, 4 (2015), 45. 2

[KS04] KE Y., SUKTHANKAR R.: Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (2004), vol. 2, IEEE, pp. II–506. 2

[KS14] KAZEMI V., SULLIVAN J.: One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1867–1874. 4

[LLW04] LEVIN A., LISCHINSKI D., WEISS Y.: Colorization using optimization. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 689–694. 2, 7

[LLYY17] LI Y., LIU S., YANG J., YANG M.-H.: Generative face completion. *arXiv preprint arXiv:1704.05838* (2017). 11

[LMNF09] LEPETIT V., MORENO-NOGUER F., FUA P.: Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision 81*, 2 (2009), 155–166. 3

[LTO*15] LI H., TRUTOIU L., OLSZEWSKI K., WEI L., TRUTNA T., HSIEH P.-L., NICHOLLS A., MA C.: Facial performance sensing head-mounted display. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 47. 1, 2

[LYT11] LIU C., YUEN J., TORRALBA A.: Sift flow: Dense correspondence across scenes and its applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33*, 5 (2011), 978–994. 7

[OLSL16] OLSZEWSKI K., LIM J. J., SAITO S., LI H.: High-fidelity facial and speech animation for vr hmds. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2016) 35*, 6 (December 2016). 1, 2

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. In *ACM Transactions on Graphics (TOG)* (2003), vol. 22, ACM, pp. 313–318. 8

[RAGS01] REINHARD E., ASHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Computer graphics and applications*, 5 (2001), 34–41. 2, 7

[RPZZ14] ROMERA-PAREDES B., ZHANG C., ZHANG Z.: Facial expression tracking from head-mounted, partially observing cameras. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on* (2014), IEEE, pp. 1–6. 2

[SPB*14] SHIH Y., PARIS S., BARNES C., FREEMAN W. T., DURAND F.: Style transfer for headshot portraits. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 148. 2, 7

[TZN*15] THIES J., ZOLLHÖFER M., NIESSNER M., VALGAERTS L., STAMMINGER M., THEOBALT C.: Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 183. 2

[TZS*16a] THIES J., ZOLLHÖFER M., STAMMINGER M., THEOBALT C., NIESSNER M.: Face2face: Real-time face capture and reenactment of rgb videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* (2016). 2

[TZS*16b] THIES J., ZOLLÖFER M., STAMMINGER M., THEOBALT C., NIESSNER M.: FaceVR: Real-Time Facial Reenactment and Eye Gaze Control in Virtual Reality. *arXiv preprint arXiv:1610.03151* (2016). 1, 2, 11

[XBMK04] XIAO J., BAKER S., MATTHEWS I., KANADE T.: Real-time combined 2d+ 3d active appearance models. In *CVPR (2)* (2004), pp. 535–542. 2

[Zha15] ZHAO Y.: Full body scanner, 2015. http://cs.uky.edu/~yzh272/FullBodyScanandMeasurement.html. 9

[ZHG*14] ZHAO Y., HUANG X., GAO J., TOKUTA A., ZHANG C., YANG R.: Video face beautification. In *ICME* (2014), pp. 1–6. 6