Look at Me! Correcting Eye Gaze in Live Video Communication

CHIH-FAN HSU, Department of Electrical Engineering, National Taiwan University
YU-SHUEN WANG, Department of Computer Science, National Chiao Tung University
CHIN-LAUNG LEI, Department of Electrical Engineering, National Taiwan University
KUAN-TA CHEN, Institute of Information Science, Academia Sinica

Although live video communication is widely used, it is generally less engaging than face-to-face communication because of limitations on social, emotional, and haptic feedback. Missing eye contact is one such problem caused by the physical deviation between the screen and camera on a device. Manipulating video frames to correct eye gaze is a solution to this problem. In this paper, we introduce a system to rotate the eyeball of a local participant before the video frame is sent to the remote side. It adopts a warping-based convolutional neural network to relocate pixels in eye regions. To improve visual quality, we minimize the L2 distance between the ground truths and warped eyes. We also present several newly designed loss functions to help network training. These new loss functions are designed to preserve the shape of eye structures and minimize color changes around the periphery of eye regions. To evaluate the presented network and loss functions, we objectively and subjectively compared results generated by our system and the state-of-the-art, DeepWarp, in relation to two datasets. The experimental results demonstrated the effectiveness of our system. In addition, we showed that our system can perform eye gaze correction in real time on a consumer-level laptop. Because of the quality and efficiency of the system, gaze correction by postprocessing through this system is a feasible solution to the problem of missing eye contact in video communication.

CCS Concepts: •**Computing methodologies** →**Computational photography; Image processing; Neural networks;**

Additional Key Words and Phrases: Image processing, Gaze correction, Eye contact, Live video communication, Deep learning

1 INTRODUCTION

Live video communication provides an easy, fast, and cost-effective means of communication between people located far from each other. The increasing number of live video communication users and companies–and the market forecast–indicate that video calls and video conferences will dominate communication in the future [1]. However, despite breaking physical limitations, video communication is inevitably less engaging than face-to-face communication because of limited social, emotional, and haptic feedback [2]. A common but serious drawback of video communication is lack of eye contact (Figure 1). A user may perceive that the other party is not attentive to the conversation. This problem occurs because of the physical deviation between the camera and screen. For example, when using a phone where the camera is above the screen to chat with a remote user, looking at said user's eyes is perceived as looking downward by the remote participant. This common problem is termed "parallax."

DOI: 10.1145/nnnnnnn.nnnnnnn

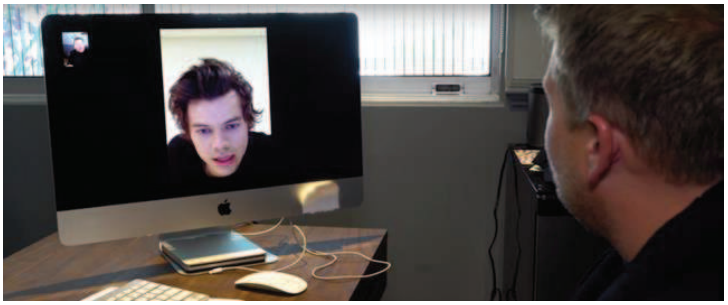


Fig. 1. Eye contact is missing during live video communication because of parallax–the physical deviation between the camera and screen. The image is obtained from YouTube channel "The Late Late Show with James Corden" [3].

Solving the eye contact problem perfectly is challenging if participants communicate with each other by using laptops or mobile phones. Since the remote participant can appear only on a planar screen, the local participant has to be in front of the screen and face the screen during the conversation. Under this circumstance, eye contact can be achieved if the remote participant is looking at the camera. Many of the gaze correction systems were built based on this assumption. They attempted to redirect the participant's gaze from the screen to the camera by video postprocessing. Ideally, the methods have to estimate eyeball rotation according to positions of the camera, screen, and the participants. Then, they can process the video based on the estimated rotations. However, most of the previous methods assumed that the rotation is a constant. Users have to retain their head pose and relative position to the camera when using the previous systems to communicate with remote participants. In addition to the eyeball rotation estimation, processing video frames without producing noticeable visual artifacts while achieving real-time performance is also challenging.

We build the gaze correction system based on the assumption that users are in front of the screen and face the screen during video conferencing. Specifically, the system corrects the gaze by redirecting it from the remote participant's eyes on the screen to the camera before the video frame is sent to the remote side. Based on the parameters manually specified once in advance, the system calculates the positions of the camera, the local participant's eyes in the real world, and the remote participant's eyes on the screen, and then estimates the eyeball rotation angles for gaze correction dynamically. As a result, users are allowed to move with a certain degree of freedom during communication. Note that our system is not presented for cheating. It assumes that the local participant is looking at the remote participant on the screen and corrects the eye gaze by a relative transformation. The eye contact will be missed if this is not the case, which is the same to a face-to-face communication.

To postprocess video frames based on the estimated eyeball rotation angles, we apply a deep neural network to warp the eye regions. Specifically, the eye images are first segmented, warped, and then overlaid onto the original video frames. We extended the network architecture of a state-of-the-art, DeepWarp [4], to compute a pixel flow field for relocating pixel positions. Dense blocks [5] are embedded into the network for better utilizing features in input images. We also introduce four loss functions to improve quality and efficiency. Two of the functions are used to preserve the shape of eyeballs and eyelids by forcing the pixels representing an identical eye component to move similarly. Another two are used to retain colors of the pixels in the boundary

region of the warped eye image. These loss functions relieve the need for fusion when the warped eye images are overlaid onto the original video frame.

To evaluate the performance of our system, we objectively and subjectively compared results generated by DeepWarp and our system. The experimental results revealed that our system outperformed the current state-of-the-art model in terms of quality. Furthermore, our system achieves real-time performance (33 frames per second, FPS) on a laptop with a consumer-level graphics card. The aforementioned advantages render gaze correction through postprocessing a feasible solution to missing eye contact in video communication. We also have released the source codes, the trained model, and the collected gaze dataset for public use (https://github.com/chihfanhsu/gaze_correction).

To summarize our contributions,

- we present a system that dynamically estimates eyeball rotation angles to correct users' gaze during video communication;
- we introduce a warping-based convolutional neural network to relocate pixels for redirecting eye gaze;
- we design four loss functions to preserve eye structures and relieve the need for fusing the gaze redirected images when they are overlaid onto the original video frame; and
- we have implemented a prototype system for gaze correction, released the source codes, the trained model, and the collected gaze dataset, which could be helpful to the future researches in gaze correction.

2 RELATED WORK

Eye contact during live video communication has been a topic of research for many years [6–12]. The presented methods can be classified into hardware- and software-based approaches.

Hardware-based approaches attempt to enable users to look at the screen and camera simultaneously. Smith et al. [13] designed a video tunnel for the camera to capture the reflection of the local participant in a half-silvered mirror positioned directly in front of the screen. The local participant can see the remote participant through this mirror. This design enables the eye contact because the mirror and screen are located at the same x and y coordinates (if z is defined as depth from the user's perspective). Because the half-silvered mirror is not fully transparent, Tapia et al. [14] applied a projector to project the remote participant's face onto a surface embedded in a small camera, thereby achieving eye contact. Jones et al. [15] presented a system to scan the local participant's face in real time and display a three-dimensional (3D) face image on a 3D display on the remote side. Apple [16, 17] and Sony [18] have proposed new types of liquid-crystal displays where the camera is embedded behind the screen. However, such devices remain unavailable to the public.

The goal of software-based approaches is to modify pixels of captured images. Jaklič et al. [11] and Solina and Ravnik [9] have conducted simple rotation of image plane around the x-axis to fix missing eye contact. Their experiments have revealed that the subjective experience of missing eye contact can be reduced. However, to solve the problem, eye gaze redirection is necessary. Several studies have been presented to synthesize face images based on depth maps or 3D facial information. A depth map can be estimated by using multiple cameras [19], stereo vision [6, 7], an RGB-D sensor camera [20, 21], or a monocular RGB camera [10]. Because these methods deform the entire face image, visual artifacts are likely noticeable if the estimated depth map is inaccurate. Moreover, the methods demand an additional step to smooth the boundary around a face when the face is overlaid onto the original video frame.

Several studies establish eye contact by replacing eye images [8, 22–25]. To achieve this, a huge amount of gaze images covering a variety of head poses and illuminations should be collected

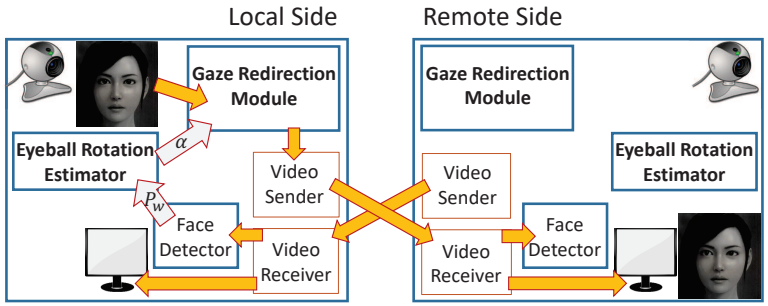


Fig. 2. Framework of our eye gaze correction system. The system corrects the local participant's eye gaze before the video frame is sent to the remote side. The orange arrows denote the flows of the video frame. The white arrows indicate the flows of the estimated eyeball rotation angles (α) and the position of the remote participant's eyes on the screen (P_w). The systems at both sides are identical. Some arrows are omitted on the remote side for conciseness.

in advance. However, because conditions can be unlimited and searching for the optimal eye images can be time consuming, these techniques are not ideal for correcting the eye contact problems during live video communication. In between modifying entire face image and eye images, GazeDirector [26] replaces the upper part of a face to redirect eye gaze. The researchers deformed a 3D template to fit the upper face in a video and then generated the corresponding textures. The 3D upper face plus a 3D synthetic eyeball are then rendered and overlaid onto the original video frames. Although GazeDirector synthesizes good results in real-time as well, after a few seconds of model fitting, however, the method cannot handle the people with eyeglasses. It turns out that a great portion of usage scenarios of video communication are not satisfied.

In addition to replacing the original eyes, DeepWarp [4] was presented to redirect eye gaze by warping. They trained a warping-based convolutional neural network to achieve the goal. Given a demanded rotation, the network can estimate a pixel flow field for warping eye images, which attempts to minimize the L2 distance between the real eye and warped eye images. Although the method approaches high quality and high efficiency, visual artifacts such as the distortion around the pupil and eyelids still frequently appear. In addition, the method assumes that the rotation of eyes is given and does not consider the real scenario of live video communication. To establish eye contact during live video communication, we extend DeepWarp by modifying the network architecture and introduce new loss functions to help network training. This extension improves not only visual quality but also system performance. In addition, we estimate the rotation of eyes to dynamically correct the gaze and establish eye contact when participants pay attention to each other. The experiments demonstrated that the presented method can serve as a practical solution for establishing eye contact in live video communication.

3 GAZE CORRECTION SYSTEM

We introduce a gaze correction system to maintain eye contact between participants during video conferencing. The system is built based on the assumption that two participants are in front of the respective screens and facing the screens during the conversation. The scenario involving more participants will be discussed in Section 5. Since the screen and the camera are not collocated, the goal is to redirect the local participant's eye gaze from the remote participant on the screen to the camera by manipulating each video frame before it is sent to the remote side. Figure 2 illustrates the workflow. Specifically, the system estimates the eyeball rotation angles according to positions

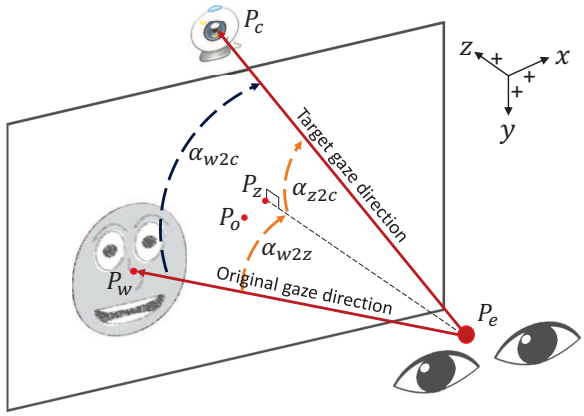


Fig. 3. Our system redirects the gaze of a local participant from the remote participant P_w to the camera P_c. The eyeball rotation angles α_{w2c} are determined by α_{w2z} and α_{z2c}. The arrows located at upper right corner indicate the axes and the corresponding positive directions of the physical coordinate system P.

of the camera and the two participants' eyes. Then, it warps the eye images based on the estimated angles to correct eye gaze.

3.1 Estimation of eyeball rotation angles

We estimate the eyeball rotation angles to redirect the gaze of a local participant from the remote participant on the screen to the camera. Figure 3 illustrates the estimation, where the unit of the physical coordinate system P is centimeter (cm), and the screen is aligned with the $xy-$plane of the coordinate system. Let P_c, P_e, and P_w denote the positions of the camera, center of the local participant's eyes, and center of the remote participant's eyes on the screen, respectively. To establish eye contact during video communication, we modify the local participant's gaze direction from $\overrightarrow{P_eP_w}$ to $\overrightarrow{P_eP_c}$ by rotating the eyeball by α^x_{w2c} and α^y_{w2c} degrees about the $x-$ and $y-$axes, respectively. Let P_z be the intersection of the ray from P_e normal to the $xy-$plane, we further separate the eyeball rotation angles into

$$\begin{cases} \alpha^x_{w2c} = \alpha^x_{w2z} + \alpha^x_{z2c} \\ \alpha^y_{w2c} = \alpha^y_{w2z} + \alpha^y_{z2c} \end{cases}. \tag{1}$$

In equation 1, α_{w2z} and α_{z2c} represent the eyeball rotation angles that can redirect the gaze from $\overrightarrow{P_eP_w}$ to $\overrightarrow{P_eP_z}$ and from $\overrightarrow{P_eP_z}$ to $\overrightarrow{P_eP_c}$, respectively. Accordingly, based on Equation 1, we calculate the separated angles by

$$\begin{cases} \alpha^x_{w2z} = tan^{-1}\left(\dfrac{P^x_w - P^x_e}{P^z_w - P^z_e}\right) \\ \alpha^y_{w2z} = tan^{-1}\left(\dfrac{P^y_w - P^y_e}{P^z_w - P^z_e}\right) \end{cases}, \quad \text{and} \quad \begin{cases} \alpha^x_{z2c} = tan^{-1}\left(\dfrac{P^x_e - P^x_c}{P^z_c - P^z_e}\right) \\ \alpha^y_{z2c} = tan^{-1}\left(\dfrac{P^y_e - P^y_c}{P^z_c - P^z_e}\right) \end{cases}, \tag{2}$$

and then sum the results to obtain the eyeball rotation angles, where the superscript z indicates the $z-$axis of the coordinate system.

To obtain the camera and participants' eye positions (P_c, P_e, and P_w) in the physical coordinate system, we first define the origin of the system P_o at the screen center. For the devices with fixed specifications such as a smartphone or a laptop, the camera position P_c can easily be determined.

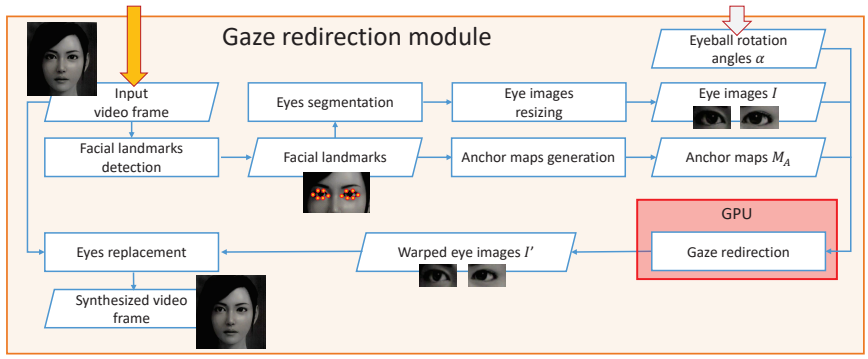


Fig. 4. Framework of our gaze redirection method. We apply the facial landmarks of the eye (anchors) to crop eye images I and generate the corresponding anchor maps M_A. Then, I, M_A, and the estimated eyeball rotation angles α are fed into a warping-based convolutional neural network for gaze redirection. Finally, we overlay the warped eye images onto the original video frame.

Otherwise, we ask users to manually specify the camera position based on the given coordinate system. To determine the center of the remote participant's eyes P_w, we track his or her eyes on the screen and then transfer the tracked position to the physical coordinate system. Finally, to determine the position of the local participant's eyes P_e, which is dynamic, we use the focal length of the camera, f, to estimate the depth from the screen to the participant. The focal lengths of popular devices are available online. Alternatively, we ask the participant to place their head d cm away from the camera. Our system counts the number of pixels in the interpupillary space in the captured image to estimate the focal length based on the perspective projection [27]:

$$f = \frac{R_{IPD} \times d}{P_{IPD}}, \tag{3}$$

where P_{IPD} and R_{IPD} denote the user's interpupillary distances in the physical (cm) and screen (pixel) coordinate systems, respectively. We set $P_{IPD} = 6.3$ cm by default based on the statistic in [28]; this value can be modified for personalization.

Once the focal length is available, the distance from the participant's eye to the screen P_e^z and then the center of the eyes P_e can be dynamically estimated from the captured video frame by

$$P_e^z = -\frac{f \times P_{IPD}}{R_{IPD}} \tag{4}$$

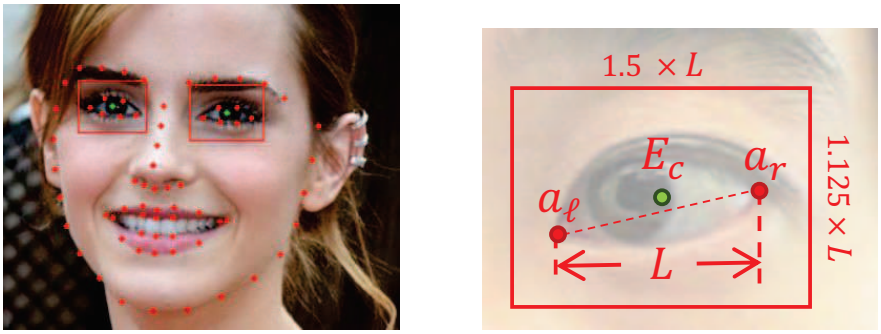
and

$$P_e^{x,y} = \frac{R_{le} + R_{re}}{2} \left(\frac{|P_e^z|}{f} \right) + P_c^{x,y}, \tag{5}$$

where R_{le} and R_{re} are the positions of the left and right eyes in the screen coordinate system R (pixel), respectively. Once the aforementioned three positions (P_c, P_e, and P_w) are estimated, the eyeball rotation angles α_{w2c}^x can be calculated to correct the local participant's gaze direction.

3.2 Image warping based on the estimated eyeball rotation angles

We warp eye images based on the estimated eyeball rotation angles to achieve gaze correction. Specifically, the system segments the left and right eyes from each video frame, modifies the eye pixels by a warping-based convolutional neural network, and then overlays the warped results onto the original video frame. Figure 4 shows the framework of the gaze redirection method.



(a) (b)

Fig. 5. (a) Sixty-eight facial landmarks detected by Dlib. (b) We crop the eye images according to the detected landmarks. The leftmost a_ℓ and rightmost a_r landmarks that tightly surround an eye are used to define the eye region. We set the center of the eye region E_c slightly above the center of a_ℓ and a_r because the upper eyelid is more stretchable than the lower eyelid.

3.2.1 Preprocessing. To segment the eye region I, we employ the Dlib library [29, 30] to detect 68 facial landmarks, as illustrated in Figure 5(b). Let the six landmarks that tightly surround the eye term "anchors", and the leftmost and rightmost anchors of an eye be a_ℓ and a_r, respectively. To cover the whole region of an eye, we empirically set the eye region's width to $1.5 \times L$, where L is the horizontal distance between a_ℓ and a_r, and set the region's aspect ratio to 4:3. Because the upper eyelid is more stretchable than the lower eyelid, we set the eye center E_c, to a position slightly above the center point of a_ℓ and a_r. Namely,

$$\begin{cases} E_c^x = \frac{1}{2}(a_\ell^x + a_r^x) \\ E_c^y = 1.5L \times \frac{3}{4} \times \frac{1}{12} + \frac{1}{2}(a_\ell^y + a_r^y) \end{cases}, \tag{6}$$

where $1.5L \times \frac{3}{4}$ is the height of the eye region. In addition, we generate twelve anchor maps M_A as same as DeepWarp to embed eye's structural information to the neural network. These maps are computed from the six anchors. Specifically, let A be the anchor of an eye. For each anchor, $a_i = (a_i^x, a_i^y), a_i \in A$ is used to generate two maps by

$$\begin{cases} M_{a_i^x}(x, y) = a_i^x - x \\ M_{a_i^y}(x, y) = a_i^y - y \end{cases}. \tag{7}$$

3.2.2 Warping-based convolutional neural network. The core of the gaze redirection is a warping-based convolutional neural network. The network takes three inputs, namely a segmented eye image I, the corresponding anchor maps M_A, and the eyeball rotation angles α_{w2c}. Subsequently, the network generates a pixel flow field F_θ and a brightness field B_θ to warp eye images I'_θ. In other words, the goal of the network is to minimize the objective function

$$min_\theta D(I'_\theta, I^t), \tag{8}$$

where I^t is the ground truth, $D(\cdot)$ is a distance function that measures the similarity of I'_θ to I^t, and θ denotes the hyperparameters in the network. Note here that θ is omitted in the following passages.

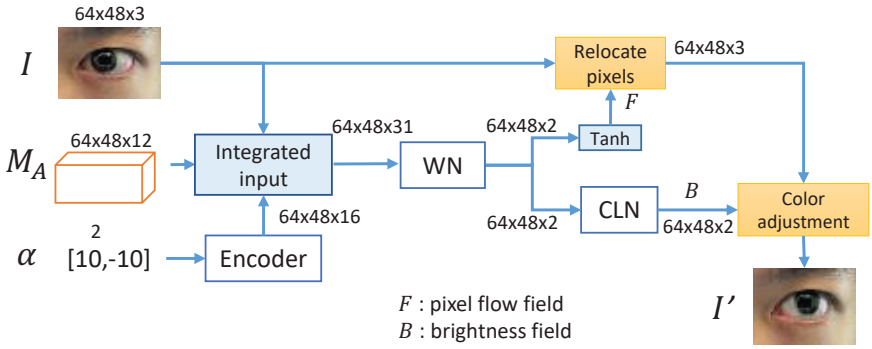


Fig. 6. Overview of our warping-based convolutional neural network, which contains three components: an encoder, a warping network (WN), and a color network (CLN). The sizes of each input and output are shown to help readers to interpret the structure.

We construct the neural network based on the concept of DenseNet [5] which improves the utility of features in hidden layers and prevents the vanishing gradient problem. The detailed structure is shown in Figures 6 and 7. As illustrated, the network structure is composed of three components: an encoder, a warping network (WN), and a color network (CLN). The encoder transforms the eyeball rotation angles α_{w2c} from a two-dimensional (2D) vector to a 16-dimensional vector (16D) and duplicates the 16D vector to form a feature map. The feature map is concatenated with I and M_A and then the integrated input is fed into the WN. The WN takes the input to infer a pixel flow field for image warping and a feature map for color correction. Eye gaze redirection through simple warping may create visual artifacts; specifically, artifacts appear when the eye anchors deviate from the boundary of eyelids and sclera or when the iris is occluded or dis-occluded by eyelids. Because artifacts mainly appear at the sclera, we alter the brightness of the warped eye image by adding the CLN at the back end of the WN in order to generate a brightness field that can linearly interpolate the white and the original pixel colors. Notably, each position in the field is a 2D weight vector. Because the weight should be among [0,1] and the L1-norm of each weight vector should be 1, we add a spatial Softmax layer at the end of the CLN.

3.2.3 Loss functions. We train the neural network to minimize the L2 distance between I' and I^t (Equation 8) because this strategy can strongly penalize outliers that are visually noticeable to humans. However, minimizing only the L2 distance is insufficient for generating high-quality warped results because the distance only considers color differences between images. While the shape and structure of eyes are oblivious, ghost effects frequently appear at regions close to pupil, iris, sclera, and eyelids. We thus present new loss functions to help training the network.

Shape-based loss functions. To retain the shape and structure of an eye, we expect pixels on eyeballs and eyelids to move similarly when the eye image is warped. An intuitive idea to achieve this is to segment the sclera, pupil, and iris, and then constrain their movements. However, this additional segmentation process can be imperfect and time consuming. Observing that the sclera is almost white, whereas the pupil and iris are usually darker than the sclera, the shape of each region can be preserved according to the brightness of a pixel. In other words, dark pixels–which represent the iris and pupil–are forced to have similar motion, whereas bright pixels–which represent the sclera–have greater freedom of movement. Accordingly, we define the loss function

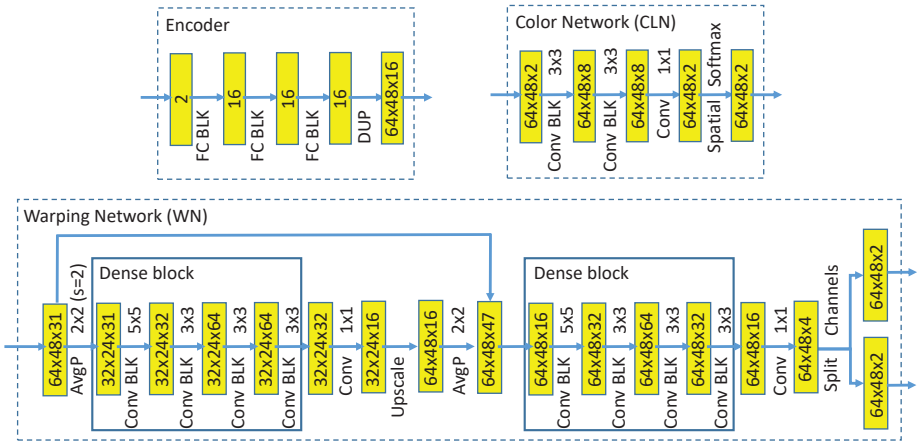


Fig. 7. Detailed network structure of each component. The shape in each box indicates the tensor size of the hidden layer. In the encoder, each fully connected block (FC BLK) sequentially contains a fully connected layer and ReLU activation function [31]. In the WN and CLN, the convolutional block (Conv BLK) sequentially contains a convolutional layer, ReLU activation function, and batch normalization layer [32]. AvgP denotes the average pooling layer, and s in parentheses represents the stride of the layer. The default value of the stride is set to 1 and is not included in the figure.

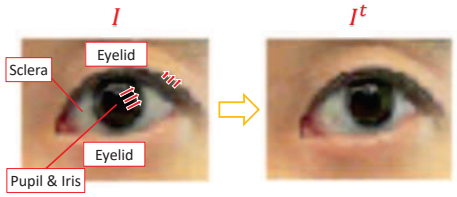


Fig. 8. We warp each eye image I to as similar as I^t to redirect the gaze. To retain the eyeball structure, pixels on pupil and iris should move similarly. We also expect pixels on eyelids to move similarly because of the low degree of freedom of eyelids.

as follows:

$$loss_{eb} = \sum_{p \in I} R(p) \cdot (1 - L(p)) \cdot \left(\left| \frac{\partial F(p)}{\partial x} \right| + \left| \frac{\partial F(p)}{\partial y} \right| \right), \tag{9}$$

where

$$R(p) = \begin{cases} 1 \text{ if } p \text{ is a pixel representing the eyeball,} \\ 0 \text{ otherwise.} \end{cases} \tag{10}$$

The subscript eb indicates the eyeball, $L(p)$ is the brightness of a pixel p, and $F(\cdot)$ is pixel flow field inferred by the neural network. Regarding the eyelids el, we similarly preserve their shapes as follows:

$$loss_{el} = \sum_{p \in I} (1 - R(p)) \cdot \left(\left| \frac{\partial F(p)}{\partial x} \right| + \left| \frac{\partial F(p)}{\partial y} \right| \right). \tag{11}$$

Color-based loss functions. We apply the CLN to reduce visual artifacts when an iris is occluded or dis-occluded by the eyelid. However, when training of an end-to-end neural network, the CLN may markedly change pixel colors to minimize the L2 distance. For example, the CLN can

transform the color of a pixel from black to white to represent sclera instead of moving a sclera pixel to the correct position. To prevent this phenomenon, we add color-based loss functions to the objective function. We expect the brightness field at each pixel $B(p)$ to be small and introduce the term

$$loss_p = \sum_{p \in I} C(p) \cdot B(p), \tag{12}$$

where $C(\cdot)$ is a predefined penalty map whose values increase from the eye center E_c (Figure 5(b)) to the boundary of the eye region. Specifically,

$$C(p) = \beta \times \sqrt[2]{(p^x - E_c^x)^8 + (p^y - E_c^y)^8} + \gamma, \tag{13}$$

where β and γ are the arbitrary parameters used to control the curve and base of the penalty map $C(\cdot)$, respectively, and we set $\beta = 3$ and $\gamma = 0.05$ during training. In addition to reducing the color adjustment, we expect the adjustment to be smooth, particularly in the areas close to the boundary of the eye region. Otherwise, unnatural color discontinuity may appear. Therefore, we minimize the first derivative of the brightness field. To implement this idea, we present the following term:

$$loss_s = \sum_{p \in I} C(p) \cdot \left(\left| \frac{\partial B(p)}{\partial x} \right| + \left| \frac{\partial B(p)}{\partial y} \right| \right). \tag{14}$$

By integrating the aforementioned loss functions,

$$loss_{tot} = loss_{L2} + loss_{eb} + loss_{el} + loss_p + loss_s, \tag{15}$$

into the training procedure, we found out that the loss function not only improves the quality of warped results but also relieves the burden of postprocessing for stitching or blending when the results are overlaid onto the original image because relocating pixels in the boundary region will be strongly penalized by the loss function, therefore, the boundary pixels are nearly untouched. Notably, these loss functions are differentiable and the integrated loss function can be a weighted sum of these loss functions.

4 SYSTEM IMPLEMENTATION AND NETWORK TRAINING

4.1 Implementation details

We have implemented the presented gaze correction system using Python 3.5.3. The video sender and receiver are implemented by the TCP socket API and the proposed neural network is implemented using TensorFlow [33]. To accelerate the face detection of the local participant, we downsample the input video frame from VGA (640 × 480 pixels) to QVGA (320 × 240 pixels) to reduce the searching cost because the position of faces can tolerate distortions to a certain degree. Then, to crop the eyes of the local participant and estimate the participant's eye position P_e (Equation 5), the 68 facial landmarks are detected in the VGA-size video frame based on the corresponding regions of faces. Regarding the position of remote participant's eyes P_w, we simply estimate P_w by the face position for acceleration. The simplification is reasonable because the remote participants' eyes are only used for eyeball rotation estimation. Namely, P_w can be estimated by

$$\begin{cases} P_w^x = \dfrac{P_s^W}{R_s^W} \left(\dfrac{R_s^W}{2} - R_w^x \right) \\ P_w^y = \dfrac{P_s^H}{R_s^H} \left(\dfrac{R_s^H}{2} - R_w^y \right), \end{cases} \tag{16}$$

where R_w denotes the position of the face in the screen coordinate system, P_s denotes the physical size of the screen, R_s denotes the screen resolution, and superscripts W and H represent the screen

width and height, respectively. Considering that detecting the remote participant's face in the screenshot can be time consuming and interfere by the desktop wallpaper, we estimate R_w based on the position of the application window and the center of the face in the application window for further acceleration. Once the segmented eye images I, corresponding anchor maps M_A, and eyeball rotation angles α_{w2c} are ready, they are fed into two pre-loaded networks to correct the eye gaze.

4.2 Network training

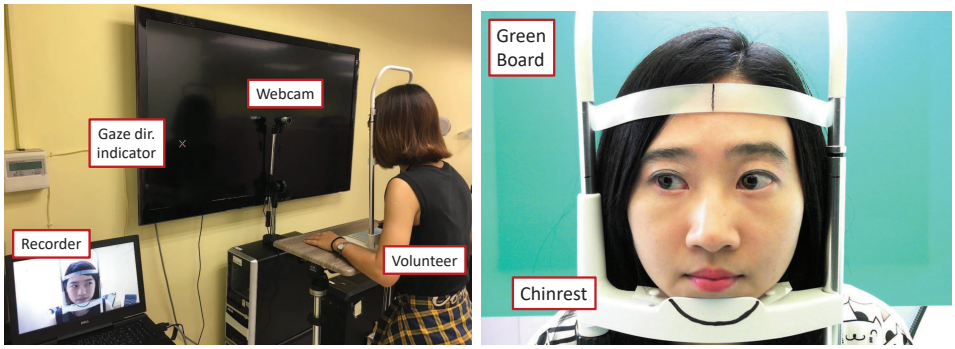
We trained the presented network on a desktop with an Intel® Core™ i7-6850K and two NVIDIA GeForce GTX 1080 graphics cards. The collected gaze dataset was partitioned into 35 and 2 volunteers for training and validation (3,450 and 198 gaze images), respectively. Adam optimizer [34] was used to minimize the total loss, the batch size was set to 256 for each GPU, and the initial learning rate η was set to 10^{-3}. To fully utilize GPU resources, in each step, two different training batches were fed into the networks on the two graphics cards. Because the networks shared the same weight, the gradients computed from various image batches were first averaged and then used to update the shared weights for synchronization. During the training, we calculated the validation losses once every half of the training set had been fed to the network. Empirically, we found that the adaptive learning rate algorithm could further improve the quality of the gaze redirected image. If the validation loss did not decrease five times in a row, the learning rate was decreased by $0.9 \times \eta$. To avoid the network overfitting, the training process was stopped if the validation loss did not decrease 16 times in a row.

4.3 Dataset collection

We developed our own training dataset because no gaze datasets containing precise anchors are publicly available. The dataset was developed in a carefully designed laboratory environment under consistent lighting conditions. Specifically, a "×" mark was shown on the screen to indicate the direction where the volunteer was asked to gaze at. In addition, a chinrest was used to fix the position of volunteer's head to reduce the direction error; and a green board was placed behind the volunteer to prevent unstable autofocus of the camera. Thirty-seven Asian volunteers participated in the dataset collection; each of them was asked to fix his or her gaze at 100 different gaze directions. Considering the physical limitations of eye movement, we recorded the volunteers' gaze directions from -40° to 40° horizontally and -30° to 30° vertically. Among the 100 recorded gazes, sixty-three were of fixed directions and the remaining 37 were randomly sampled. In the fixed directions, the angle difference between two adjacent directions was set to 10° horizontally or vertically. To maintain data quality, we manually removed images that the volunteer's eyes were closed. Figure 9 shows the environmental setup and an example of a collected gaze image. Overall, our collected dataset contains 3,648 gaze images (359,810 training pairs).

5 RESULTS AND DISCUSSIONS

We have implemented the presented eye gaze correction system and tested the program on a consumer-level laptop with Intel® Core™ i7-4720HQ @ 2.60GHz, 16 GB RAM, and a NVIDIA GeForce GTX 960M GPU. The input video was captured by a built-in webcam with VGA resolution. We tested our system in a one-on-one communication scenario (a local participant chatted with a remote participant). Table 1 shows the timing statistics, namely the averages measured in 1 minute. The table also shows the performance of DeepWarp measured under the same condition. Our system is slightly faster than DeepWarp although it contains more hyperparameters than DeepWarp (249,328 vs. 58,656). The reason is that DeepWarp adopts a two-level warping



(a) Environmental setup (b) A collected image

Fig. 9. Setup of our collected dataset. We showed a "×" mark on a screen to indicate the position where the volunteer was asked to gaze at. A camera was placed in front of his or her head to collect the gaze image. To maintain quality when collecting the dataset, we used a chinrest to restrict the participant's head pose and a green board to prevent unstable autofocus of the camera.

Table 1. Timing statistic of each major step of our system, which was tested using a consumer level laptop. Values in parentheses are standard errors, which were calculated by $\sigma/\sqrt{n}$, where σ is the standard deviation and n is the number of samples.

	Face detection	Eyes segmentation	Gaze redirection	Total
Our system	14.99 (0.02) ms	4.51 (0.05) ms	9.14 (0.03) ms	30.4 (0.04) ms
DeepWarp	14.99 (0.02) ms	4.57 (0.05) ms	9.95 (0.03) ms	31.2 (0.05) ms

framework that needs to relocate pixels and perform bilinear interpolation twice, whereas our system adopts a one-level framework, thus the computation cost can be cut in half. We mention here that the total computation time comprises not only gaze redirection but also the processing time of face detection, eye image segmentation, and other minor steps. Overall, our system can perform in real time.

To evaluate whether our system is sufficient to solve eye contact problem in video conferencing, we extend the system to a video communication platform. Figures 10, 11, and 12 show screenshots with and without gaze correction. As can be seen in Figure 10, the local participant was allowed to move while communicating with the remote participant because the eyeball rotation angle for correction was dynamically estimated based on the relative positions of the camera and the two participants. We point out that, although the network was trained on the dataset that contains only frontal faces, it is still able to correct the eye gaze of a local participant whose head is slightly rotated. It it because eye movements in a video would not be strongly affected if the head is close to the frontal pose. Since our system corrects the eye gaze by warping rather than synthesis, it allows users to move when communicating with remote participants. In addition, our system can correct the eye gaze when the participant wears eyeglasses (Figure 11). Thanks to the image warping, visual artifacts can be unnoticeable if the eyeglasses frame does not occlude eyeballs. Finally, Figure 12 shows the result that contains complex background and dynamic illumination. While the work of [29, 30] is quite robust to detect facial landmarks, our system inherits the advantages and is sufficient to crop the eye regions and then correct the gaze in such a challenging environment. The figure also shows that our method can handle multiple people in the local side, with the

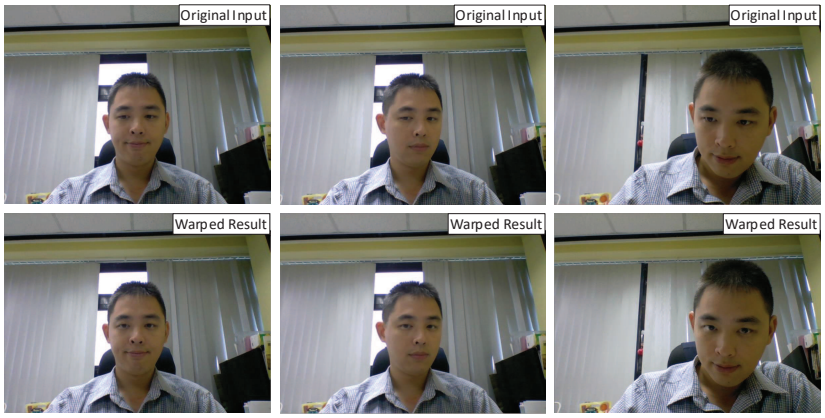


Fig. 10. Our system can dynamically estimate eyeball rotation angles to correct eye gaze. The participant is allowed to move his body and head poses during video conferencing.

Fig. 11. Our system can correct eye gaze even when a participant's head is partially occluded by his or her hand or accessories such as eyeglasses.

price of multiple computation. We encourage readers to view our supplementary video of system demonstration (https://youtu.be/9nAHINph5a4) because dynamic motions are difficult to observe in still images. Note that all the experimental results shown in the paper and the supplementary video were generated with the same system parameters.

Fig. 12. Our system can correct eye gaze in various sites with various background. In addition, the system can correct multiple local participants' gaze based on the estimated eyeball rotations.

5.1 Redirecting Gaze on Heterogeneous Datasets

We trained the network on our collected dataset, which comprises images of only Asian people. To evaluate its robustness against individuals of a range of ethnicities, we tested the network on two heterogeneous datasets: 1) the validation set of our collected gaze dataset (DIRL) and 2) the Columbia gaze dataset (CAVE) [35]. Figure 13 shows the results of gaze being redirected vertically and horizontally, ranging from -15° to 15°. The left and right columns show the images selected from DIRL and CAVE, respectively. As indicated, although our system was trained on DIRL dataset, it works well on the CAVE dataset. The result is not surprising because we redirect eye gaze by warping. The color of pupil and iris will be kept by our system. We also show the gaze redirected faces in Figure 14, which are not considered when the network was trained.

5.2 Ablation study

To evaluate the performance of the presented loss functions, we trained the network with three loss combinations; the results are shown in Figure 15(a) and 15(b). Specifically, the combinations were L2 loss only (L2), L2 loss plus shape-based loss (L2S), and L2 loss plus shape-based loss and color-based loss (L2SC). In the experiment, the L2 distance between the warped eye image I' and ground truth I^t was computed for evaluation. We evaluated only gaze ranging from $-15°$ to $15°$ horizontally and $-10°$ to $10°$ vertically because of the range limitation of CAVE. No gaze images outside of this range were included.

Figure 15(a) shows the averaged L2 distances of our system. The error bar on the top shows the 95% confidence interval, which can be calculated by $\pm 1.96\times$ standard error. As indicated, the shape-based loss functions yielded significant improvements in both datasets, whereas the color-based loss functions did not. We applied our loss functions to DeepWarp (Figure 15(b)) in the experiment and found similar results. However, the color-based loss functions remain important

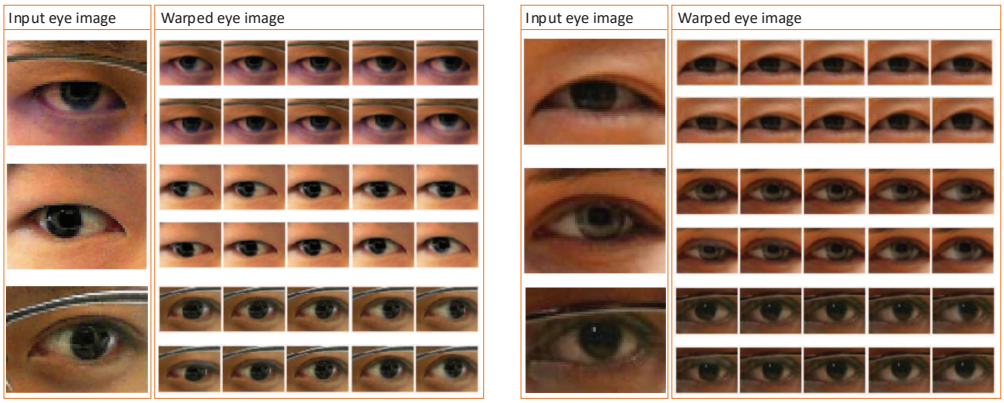


Fig. 13. We redirect the gaze of eye images, which are from our dataset (left) and the CAVE dataset (right), by using our system. The range is from -15° to 15°, both vertically and horizontally. Although the presented network was trained on DIRL dataset, which comprises images of only Asian people, it can handle individuals of a range of ethnicities in the CAVE dataset.

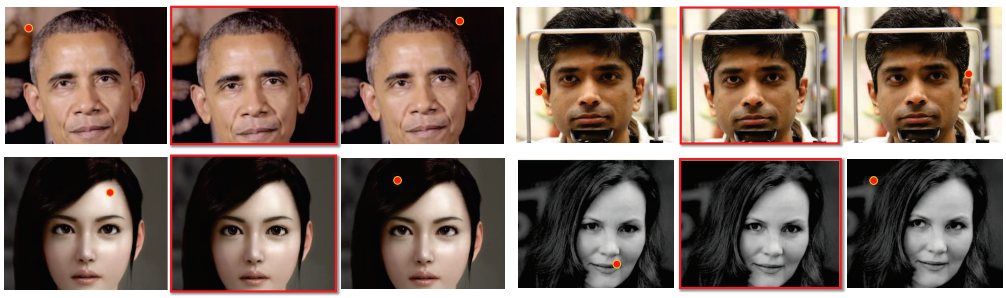


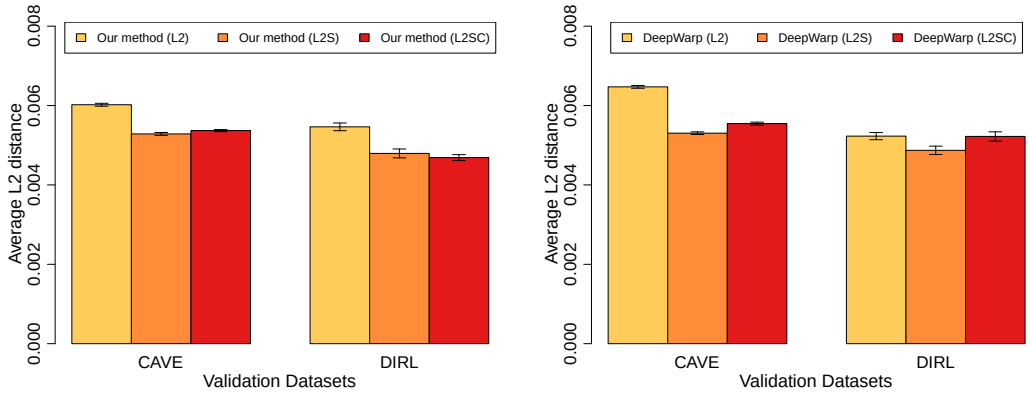
Fig. 14. Original images (highlighted with a red border) and the gaze redirected results (on the left and right sides of each set). The dot indicates the direction in which the person in the image is looking. Notice that these images were not considering in network training.

because they can retain the colors of pixels on image boundaries. Accordingly, postprocessing such as stitching and blending are not required when warped results are overlaid onto the original video frame. Figure 16 shows an example of this. The left and right images were warped by our network trained with L2 and L2SC, respectively. In the left image (trained by L2 only), shape artifacts on the eyeball and discontinuity artifacts on the boundary of the eye image are easily noticeable.

5.3 Image quality evaluations

We compared our gaze redirection network with the state-of-the-art method, DeepWarp [4], objectively and subjectively to evaluate the effectiveness of our network. The two models were trained on the DIRL dataset and the training processes was halted before overfitting began.

5.3.1 Objective evaluation. The eye landmarks demanded by DeepWarp and our method are different. Both require six landmarks to describe the eye's structure. However, DeepWarp demands an additional landmark to indicate the pupil. Therefore, we compared our method to two



(a) Our system (b) DeepWarp

Fig. 15. Average L2 distance to ground truth under various loss combinations. Shape-based loss functions significantly improve the quality. Although the color-based loss functions dose not, it restricts the color changing in the boundary of the eye image. Hence, an additional fusion process for our system is not required.

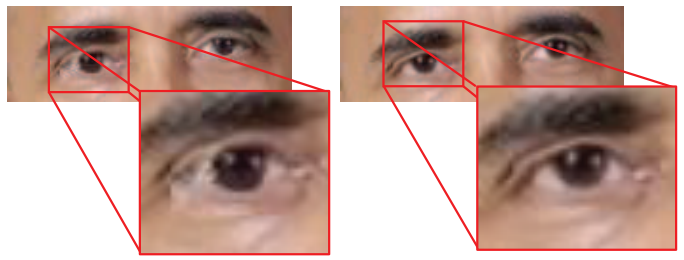
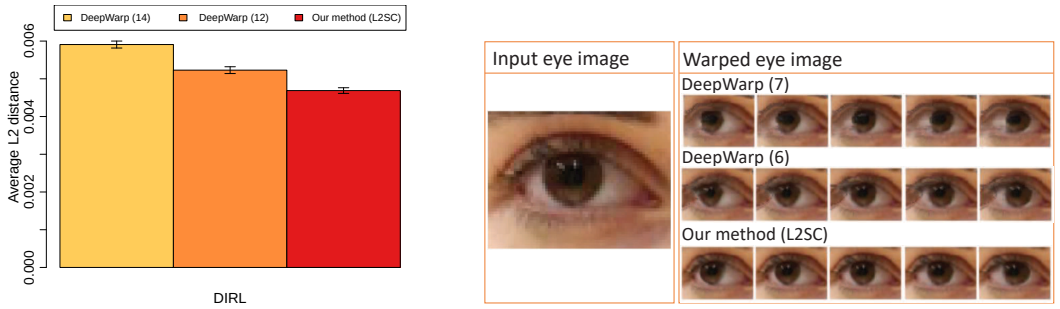


Fig. 16. Results generated by our networks trained by L2 (left) and L2SC (right) loss combinations. Shape artifacts on the eyeball and discontinuity artifacts on the eye image boundary are easily noticeable in the left image.

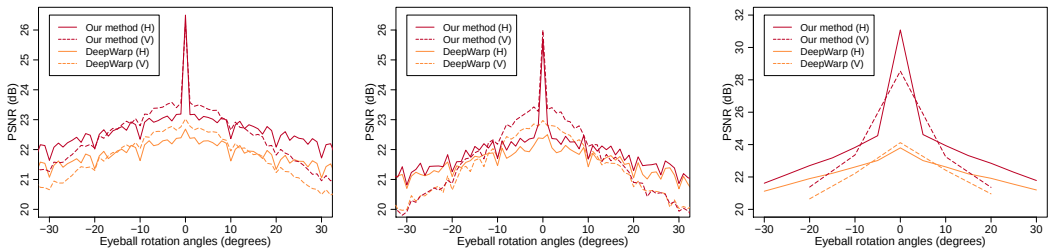
versions of DeepWarp, which takes seven and six landmarks as inputs in this objective evaluation. Figure 17(a) shows the averaged L2 distances measured in the DIRL dataset. Unsurprisingly, our system performs the best because of the shape- and color-based loss functions. However, unexpectedly, DeepWarp with seven landmarks performs worst. We suspect the reason for this is that the network is undermined by ambiguous pupil positions labelled by humans. In addition to statistics, we show examples generated by all three networks. As can be seen, visual artifacts such as structural distortions and ghost effects appear in the results generated by DeepWarp.

In addition to L2 distances, we compared the warped results and ground truths by calculating the peak signal-to-noise ratios (PSNRs). The results generated by our system and DeepWarp were evaluated. Of the DeepWarp versions, only that with six landmarks was included in the comparison because it performs better than that with seven landmarks. Figure 18(a) shows the averaged PSNRs of the training set for vertical and horizontal eyeball rotations. As indicated by the lines, our system outperforms DeepWarp in both vertical and horizontal directions. In addition, we evaluated the two methods in the validation set; Figure 18(b) and 18(c) shows the results. Again, our system outperforms DeepWarp in our collected validation set and the CAVE dataset. One notable



(a) Image quality comparison (b) Warped result

Fig. 17. Our system outperforms two versions of DeepWarp, which utilizes six and seven (including pupil position) landmarks, respectively, in objective evaluation (a) and visual quality comparison (b).



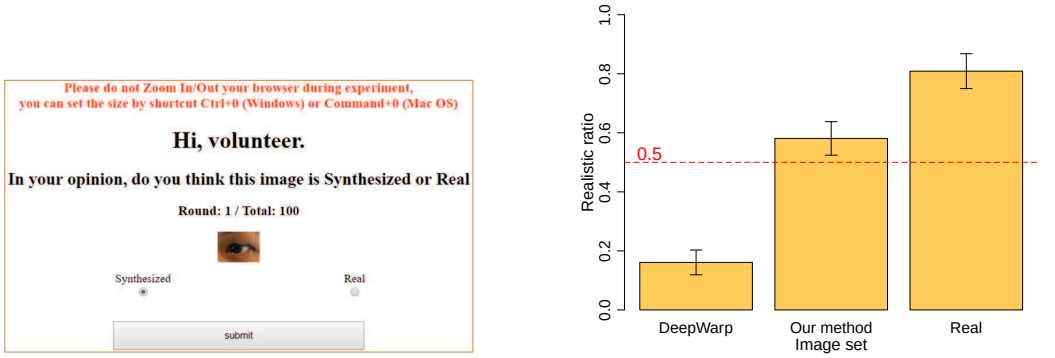
(a) Training dataset (b) Our validation dataset (c) CAVE

Fig. 18. Objective evaluations of our system and Deepwarp on our collected dataset (DIRL) and the Columbia gaze dataset (CAVE). The PSNR values indicate that our system outperforms Deepwarp.

finding is that although the average PSNR decreased as the eyeball rotation angle increased, the quality of gaze redirected horizontally is superior to the quality of gaze redirected vertically. We reason that redirecting gaze vertically is more difficult because pupils and eyelids interact when humans shift their gaze upward or downward.

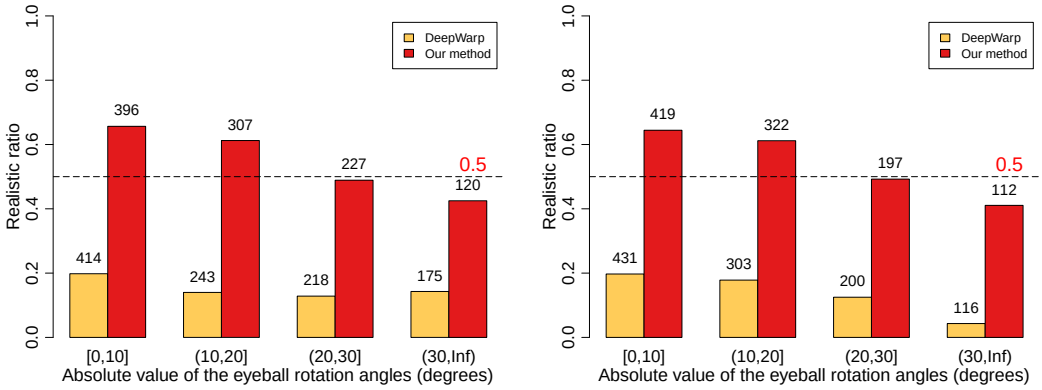
5.3.2 Subjective evaluation. Because users are the most important aspect of the system, we conducted a user study to evaluate the degree of realism of the warped eye images. The participants were shown a test eye image on the screen, and then asked to judge whether the image was real. We randomly selected 30, 35, and 35 images from the real dataset and results generated by our system and DeepWarp; and the test images were shown in a random order to avoid bias. During the study, the participants were able to observe the test images with no time restrictions. However, once they had made a decision, they were not allowed to change their answer. In addition, to obtain reliable results, the participants were not informed of the setting of the test images, and we disregarded studies that were incomplete or exhibited periodical patterns throughout the questionnaire. Figure 19(a) shows a page of the questionnaire.

Figure 19(b) shows the results contributed by 30 participants (18 males and 12 females). The height of each bar chart indicates the rate of the test images in the corresponding category that were considered real (realistic ratio). The images selected from the real dataset have the highest rate (mean = 0.81, standard error = 0.03), those generated by our system and DeepWarp have the



(a) Questionnaire (b) Result of user study

Fig. 19. Questionnaire and the results of the subjective evaluation.



(a) Horizontal (b) Vertical

Fig. 20. The larger rotation angles result in the lower realism. The value above each bar chart indicates the number of samples in the interval of rotation angles.

second highest rate (mean = 0.58, standard error = 0.03) and lowest rate (mean = 0.16, standard error = 0.02), respectively. The participants can easily recognize the results generated by DeepWarp because of the shape distortions of the pupils and eyelids, as shown in Figure 17(b). Figure 20 shows the impacts of eyeball rotation angles on the realistic ratio. Apparently, the larger rotation angles result in the lower realism.

5.3.3 Correlation between subjective and objective evaluations. Considering that no quality assessment is subjectively or objectively perfect, we compared the evaluation results to check for matching. Specifically, we computed the Pearson correlation coefficients of the PSNRs and the degrees of realism (i.e., realistic ratios) based on the participants' assessments. The values of correlation coefficients were 0.57 (p-value < 3.0×10^{-5}) and 0.72 (p-value < 5.0×10^{-8}) in the horizontal and vertical directions, respectively, indicating that the PSNRs and degrees of realism were correlated.

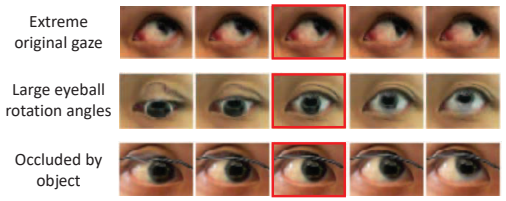


Fig. 21. Visual artifacts may occur when 1) the original eye gaze is far from center, 2) an eye gaze is redirected considerably, and 3) eyeglasses or hairs are present in the eye image. The images with a red border are the input images; those on the left and right of the input images are warped by our system.

5.3.4 People with eyeglasses. Considering that a great portion of system users may wear eyeglasses, we conducted an experiment to quantify the effectiveness of our system. Eighty-seven facial images with eyeglasses were selected from the Internet (eyeballs are not occluded by the frame). We designated gaze directions to 40 radial segments along the cone surface of 30° cone angle. Namely, forty images with different gaze directions were generated from each facial image (3,480 images in total) by our system. Eight volunteers (five males and three females, age ranged from 25 to 43) in Academia Sinica participated in the experiment to determine whether warped facial image is natural or not. The result shows that 84.5% (standard deviation = 5.9%) of the warped facial images are treated as real images, and the eyeglasses can be moderately handled by our system.

5.4 Limitations

Our system assumes that there is only one remote participant on the screen when it corrects the local participant's eye gaze. If this is not the case, the system cannot distinguish which remote participant is gazed during communication, and corrects the gaze from the firstly detected participant's face to the camera. We plan to solve the problem by applying the eye tracker in future. Regarding the image processing part, distortions may appear if the eyeballs are occluded by eyeglasses' frame or hairs, because such external objects are not considered. Figure 21 shows several examples of failure cases. Finally, the system inevitably produces visual artifacts when the original gaze is far off center or redirected to an extreme degree. Solving this problem by using the warping algorithm can be difficult because of occlusion of an eyeball by the eyelids. Fortunately, the problem is not serious because eyeball rotations for gaze correction are usually small, unless the distance between the camera and remote participant's face on the screen is long or the local participants are very close to their screens.

6 CONCLUSION AND FUTURE RESEARCH

This paper presents a real-time gaze correction system to solve the problem of missing eye contact in live video communication. First, we estimate eyeball rotation angles based on the positions of the camera, and local and remote participants' eyes. Then, the neural network is used to warp eye images to correct the gaze direction. To fulfill quality and efficiency requirements, we extend the current state-of-the-art model, DeepWarp, by modifying its network architecture and presenting shape- and color-based loss functions for training. Objective and subjective evaluations reveal the effectiveness of our system. Since the system achieves real-time performance (33 FPS) on a consumer-level laptop, it can serve as a practical solution to the problem of missing eye contact in video conferencing. Currently, the system is presented to solve eye contact problems for two-participant scenarios because we assumed that the local participant should look at the remote

participant on the screen if he or she pays attention to the conversation. In future, we will consider using eye tracker for gaze detection and solve the eye contact problem thoroughly.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Chin-Fu Nien of Academia Sinica for assisting the implementation of the TCP socket and the mathematical derivation of the eyeball rotation angles and Yu-Cheng Chen of Academia Sinica for assisting the gaze dataset collection.

REFERENCES

[1] Global webinar and webcast market overview, "https://www.researchnester.com/reports/webinar-and-webcast-market-global-demand-growth-analysis-opportunity-outlook-2023/237".
[2] P. S. N. Lee, L. Leung, V. Lo, C. Xiong, and T. Wu. Internet communication versus face-to-face interaction in quality of life. *Social Indicators Research*, 100(3):375–389, Feb 2011.
[3] The Late Late Show with James Corden. Harry styles video chats with james corden (https://www.youtube.com/watch?v=H7ZjRna4ZK4). YouTube, 2017.
[4] Y. Ganin, D. Kononenko, D. Sungatullina, and V. Lempitsky. *DeepWarp: Photorealistic Image Resynthesis for Gaze Manipulation*, pages 311–326. Springer International Publishing, 2016.
[5] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.
[6] R. Yang and Z. Zhang. Eye gaze correction with stereovision for video-teleconferencing. Technical report, Microsoft, 2001.
[7] A. Criminisi, J. Shotton, A. Blake, and P. H. S. Torr. Gaze manipulation for one-to-one teleconferencing. In *Proceedings Ninth IEEE International Conference on Computer Vision*, volume 1, pages 191–198, 2003.
[8] L. Wolf, Z. Freund, and S. Avidan. An eye for an eye: A single camera gaze-replacement method. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 817–824, 2010.
[9] F. Solina and R. Ravnik. Fixing missing eye-contact in video conferencing systems. In *Proceedings of the ITI 2011, 33rd International Conference on Information Technology Interfaces*, pages 233–236, 2011.
[10] D. Giger, J. C. Bazin, C. Kuster, T. Popa, and M. Gross. Gaze correction with a single webcam. *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2014.
[11] A. Jaklič, F. Solina, and L. Šajn. User interface for a better eye contact in videoconferencing. *Displays*, 46:25 – 36, 2017.
[12] L. S. Bohannon, A. M. Herbert, J. B. Pelz, and E. M. Rantanen. Eye contact and video-mediated communication: A review. *Displays*, 34(2):177 – 185, 2013.
[13] G. Doherty-Sneddon, A. Anderson, C. O'Malley, S. Langton, S. Garrod, and V. Bruce. Face-to-face and video-mediated communication: A comparison of dialogue structure and task performance. *Journal of Experimental Psychology: Applied*, 3(2):105–125, 1997.
[14] E. M. Tapia, S. S. Intille, J. R. Rebula, and S. Stoddard. Concept and partial prototype video : Ubiquitous video communication with the perception of eye contact. 2003.
[15] A. Jones, M. Lang, G. Fyffe, X. Yu, J. Busch, I. McDowall, M. Bolas, and P. Debevec. Achieving eye contact in a one-to-many 3d video teleconferencing system. *ACM Trans. Graph.*, 28(3):64:1–64:8, 2009.
[16] Z. Spear. Apple files may-2016 for camera hidden behind displa, "https://mobilesyrup.com/2018/03/09/apple-patent-display-hidden-camera/". 2018.

[17] Apple invents a wild new display that could conceal a camera, strobe flash and/or fingerprint scanner until needed, "http://www.patentlyapple.com/patently-apple/2013/05/apple-invents-a-wild-new-display-that-could-conceal-a-camera-strobe-flash-andor-fingerprint-scanner-until-needed.html". 2013.

[18] Sony patents technology to put camera and sensors behind smartphone display, "https://www.redorbit.com/news/technology/1112501121/sony-patents-technology-to-put-camera-and-sensors-behind-smartphone-display/". 2012.

[19] M. Dumont, S. Rogmans, S. Maesen, and P. Bekaert. Optimized two-party video chat with restored eye contact using graphics hardware. In Joaquim Filipe and Mohammad S. Obaidat, editors, *e-Business and Telecommunications*, pages 358–372, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[20] C. Kuster, T. Popa, J. C. Bazin, C. Gotsman, and M. Gross. Gaze correction for home video conferencing. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH ASIA)*, 31(6):174:1–174:6, 2012.

[21] E. T. Baek and Y. S. Ho. Gaze correction using feature-based view morphing and performance evaluation. *Signal, Image and Video Processing*, 11(1):187–194, 2017.

[22] J. Gemmell, K. Toyama, C. L. Zitnick, T. Kang, and S. Seitz. Gaze awareness for video-conferencing: a software approach. *IEEE MultiMedia*, 7(4):26–35, 2000.

[23] D. Weiner and N. Kiryati. Virtual gaze redirection in face images. In *12th International Conference on Image Analysis and Processing*, pages 76–81, 2003.

[24] Y. Qin, K. C. Lien, M. Turk, and T. Höllerer. *Eye Gaze Correction with a Single Webcam Based on Eye-Replacement*, pages 599–609. Springer International Publishing, Cham, 2015.

[25] Z. Shu, E. Shechtman, D. Samaras, and S. Hadap. Eyeopener: Editing eyes in the wild. *ACM Transactions on Graphics*, 36(1), 2016.

[26] E. Wood, T. Baltrušaitis, L. P. Morency, P. Robinson, and A. Bulling. Gazedirector: Fully articulated eye gaze redirection in video. *EUROGRAPHICS*, 37(2):217–225, 2018.

[27] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[28] N. A. Dodgson. Variation and extrema of human interpupillary distance. In Andrew J. Woods, John O. Merritt, Stephen A. Benton, and Mark T. Bolas, editors, *Stereoscopic Displays and Virtual Reality Systems XI*, volume 5291, pages 19–22. SPIE, 2004.

[29] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

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

[31] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. *ICML Deep Learning Workshop*, pages 06–11, 2015.

[32] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning*, volume 37, pages 448–456, 2015.

[33] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[34] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. 2015.
[35] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar. Gaze locking: Passive eye contact detection for human–object interaction. *ACM Symposium on User Interface Software and Technology (UIST)*, pages 271–280, 2013.