An Eye for an Eye: A Single Camera Gaze-Replacement Method

Lior Wolf
The Blavatnik School of Computer Science
Tel-Aviv University

Ziv Freund, Shai Avidan
Department of Electrical Engineering-Systems
Faculty of Engineering
Tel-Aviv University

Abstract

The camera in video conference systems is typically positioned above, or below, the screen, causing the gaze of the users to appear misplaced. We propose an effective solution to this problem that is based on replacing the eyes of the user. This replacement, when done accurately, is enough to achieve a natural looking video. At an initialization stage the user is asked to look straight at the camera. We store these frames,then track the eyes accurately in the video sequence and replace the eyes, taking care of illumination and ghosting artifacts. We have tested the system on a large number of videos demonstrating the effectiveness of the proposed solution.

1. Introduction

Videoconferencing systems hold the promise of allowing a natural interpersonal communication at a range. Recent advances in video quality and the adaptation of large high definition screens are contributing to a more impressive user experience. However, to achieve the desired impact of being in the same room the problem of gaze offset must be addressed.

This gaze problem arises because the user is looking at the screen, while the camera(s) capturing the user are positioned elsewhere. As a result, even when the user is looking straight into the image of his call partner, the gaze, as perceived at the other side, does not meet the partner's eyes. Typically, the camera is located on top of the screen, and the effect is interpreted as looking down.

Our system solves this problem by replacing the eyes of a person in the video with eyes that look straight ahead. We use an example based synthesis that is based on capturing the eyes at an initial training stage. During the videoconferencing session, we find and track the eyes. In every frame, the eyes captured during training are accurately pasted to create an illusion of straight-looking gaze. Somewhat surprisingly, the resulting effect (Figure 1) of replacing the eyes alone looks natural.

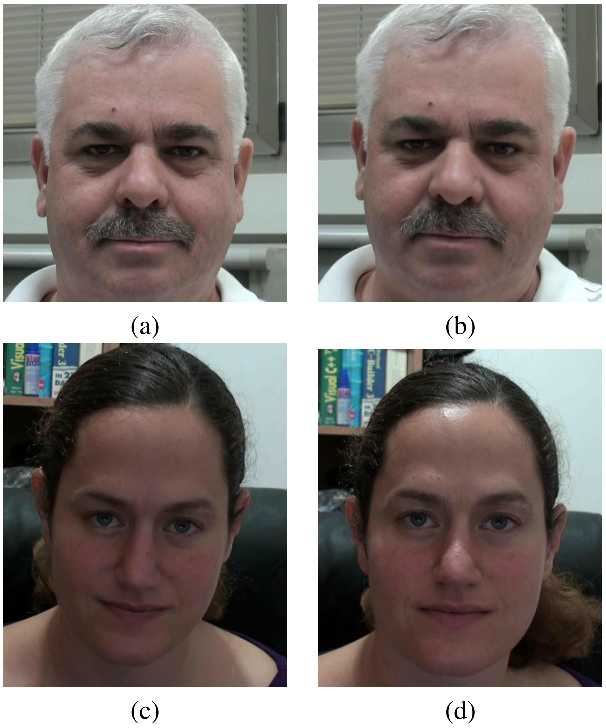


Figure 1. Two of the four images contain artificially replaced eyes to create an effect of a person looking straight forward. These eyes were automatically placed by our system. The other two images were untouched. Can you tell which images were modified?
The answer is in the footnote. [1]

2. Previous work

The importance of natural gaze, and the problem of gaze offset was presented in depth by Gemmell *et al.* [7], where a synthesis framework was presented. The solved problem is somewhat more general than what we aim to solve and includes turning the entire head by means of rendering a 3D head model.

With regards to the eyes, it was suggested that the loca-

[1] Images (a) and (d) are real, images (b) and (c) are modified.

tion of the eyes and the gaze direction would be estimated by a “vision component”. The eyes would then be replaced with a synthetic pair of eyes gazing in the desired direction.

Written in the year 2000, the authors conclude that the main difficulty they face is that the vision component is slow and inaccurate, and suggest using an infrared-based vision system until computer vision “comes up to speed”.

Despite great advances in object detection and tracking in the last decade, the accurate commercial eye trackers that are in existence are still based on infrared images in which the corneal reflections and the center of the pupils are easily detected.

In our work we use an eye model similar to the one proposed by [16]. This model contains parameters such as the center of the eye, the radius of the iris and so on. The authors define an energy function which relates image features to the geometric structure of the eye contours. Localization is performed by the steepest descent method.

In [8] an automatic initialization method is proposed based on the corners of the eye and the computation of the model fitting process is sped up. In [9] a model for blinking is introduced, and a KLT tracker is employed to track the eye model over time.

Recently, [13] proposed a multi-stage process in which the iris, upper eyelid and lower eyelid are detected sequentially. A very accurate model is being detected by Ding and Martinez [4], who observe that classifier based approaches by themselves may be unsuitable for the task due to the large variability in the shape of facial features.

There are many contributions in which the corners of the eyes are detected, however, a detailed model is not recovered. In this work we employ the method of Everingham *et al.* [5] in order to obtain an initial localization for the eyes.

An alternative solution for the problem of gaze manipulation is view synthesis from multiple cameras. In [3] dynamic programming based disparity estimation was used to generate a middle view from two cameras that were positioned on the left and right sides of the screen. The advantage of such a method is that the generated view corresponds to a true view, while our solution generates only a natural looking fake. The disadvantage, of course, is the need to use multiple cameras positioned at locations that are suitable for this purpose.

3. System overview

The core of our system is an accurate eye detector that takes an image of a face and returns an accurate position of the eye. Once we have the eye position we can replace it with an image of an eye with a proper gaze direction.

To achieve this goal we learn a regression function that maps face images to the eye model parameters, using a database of annotated images. This stage is not accurate enough and we follow up with a refinement stage.

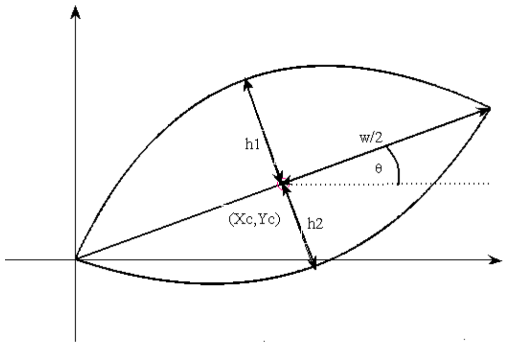


Figure 2. An illustration of the eye model employed in the gaze replacement system

The system consists of three stages. An offline stage where a rough eye pose estimation is learned from a database of annotated images. This database is a general database, unrelated to the any particular videoconferencing session, and we refer to it below as the offline database. A second stage occurs at the beginning of the video conferencing session where the user looks straight at the camera. This allows the system to construct a database of direct gaze images. We refer to this as an online database. Finally, in run-time the system constantly detects the location of the eye in the current frame, finds the most similar image in the database and replaces it.

The eye model we use is a simplified model based on the deformable template of [16, 9]. It is described by six parameters, as depicted in Figure 2. Two coordinates for the center of the eye: x_c and y_c, the width of the eye w, the hight of the eye above the line connecting the corners h_1, and the maximal height below this line h_2. Last, there is the angle θ of the eye from the image scan lines. The coordinates of the center are given relatively to the leftmost corner of the eye.

3.1. Offline Training stage

The direct gaze models are learned based on previous annotated examples. These examples are stored in an offline database that is used to bootstrap the runtime system. It consists of a separate set of 14 individuals for whom videos were captured and manually annotated in accordance with the six parameter model.

3.2. Online Database construction

Given images of a new person looking directly into the camera, the closest eye in the offline database is retrieved and then serves as an initial guess as to the eye parameters in the new image. These parameters are then adapted to provide a more accurate estimation.

3.3. Runtime eye localization

At runtime, the parameters of the eyes in the video are estimated by matching the image to the eyes of the person when looking straight. The illumination is corrected, and a replacement is performed.

4. Training phase

Given a video of a person looking directly into the camera we aim to find the eye parameters. This is done for each frame independently, thereby collecting pairs of straight looking eyes of that person.

The first stage in the processing of each frame is the localization of the corners of the eyes. This is done using the method of [5], which describes the geometric distribution of facial features as a tree-structured mixture of Gaussians [6], and captures appearance by Haar-wavelet like features [14].

After the corners of the eyes are localized, a rectangular region of interest which approximately captures the eye regions is constructed for each eye. Let w be the distance between the two corners of the eye. The rectangle of each eye is a combination of a strip of width $h_1 = .4w$ above the line connecting the two corners, and of a strip of width $h_2 = .3w$ below this line.

Each region of interest (ROI) is scaled to an image of 120 × 80 pixels. Then, SIFT [11] descriptors are computed at 24 evenly spaced points in the stretched image.

From the offline database of manually annotated eyes, we select the left and right eyes with the closest appearance descriptor. This yields an approximate model. In Leave-One-Person-Out experiments conducted on the offline database it was found that their average absolute error is about 3 pixels for the center of the eye, and 6 pixels in the width of the eye.

Table 1 presents the regression results for the left eye, and Table 2 presents the results obtained for the right eye. We compare two representations: the SIFT representation and the vector of gray values in the eye rectangle, resized to 120 pixels times 80 pixels. We also compare two machine learning algorithms: Support Vector Regression and Nearest Neighbor. SIFT obtains preferable performance, especially for the Nearest Neighbor classifier.

This initial model is then refined by performing a local search in the space of the 6 parameters. The search is conducted in a range of twice the average error in each parameter, and in a coarse to fine manner.

For each set of candidate parameter values, the closest eye in the offline database is translated, rotated and stretched in accordance with the difference in parameters. The center is shifted by the different in x_c and y_c, the width is scaled by the ratio of the w values, and the regions above and below the line connecting the two corners are stretched in accordance with h_1 and h_2 respectively. Finally, the database eye image is rotated by the difference in the θ values between the database image and the candidate parameter value.

Table 1. Mean (± standard deviation) error in pixels for each parameter of the eye model for the experiments of the **left** eye. The errors were estimated in a leave-one-person-out fashion on the offline dataset. Tow image representations (gray values and SIFT) and two learning algorithms (Nearest Neighbor and Support Vector Regression) are presented.

Para-	Grayvalues		SIFT	
meter	NN	SVR	NN	SVR
x_c	4.26 ± 3.62	3.08 ± 2.3	3.02 ± 2.47	3.08 ± 2.3
y_c	4.3 ± 2.96	3.84 ± 2.92	2.73 ± 1.8	3.23 ± 2.72
w	7.83 ± 6.16	6.86 ± 6.51	6.95 ± 6.52	6.47 ± 6.79
h_1	3.79 ± 2.25	3.35 ± 2.28	3.58 ± 3.04	3.35 ± 2.28
h_2	3.22 ± 2.73	2.93 ± 2.51	2.45 ± 1.72	2.68 ± 2.56
θ	0.10 ± 0.06	0.08 ± 0.04	0.08 ± 0.06	0.07 ± 0.05

Table 2. Mean (± standard deviation) error in pixels for each parameter of the eye model for the experiments of the **right** eye. See Table 1 for details.

Para-	Grayvalues		SIFT	
meter	NN	SVR	NN	SVR
x_c	3.69 ± 2.99	7.49 ± 4.66	3.76 ± 3.06	5.72 ± 4.42
y_c	3.62 ± 2.89	3.01 ± 2.62	2.84 ± 2.01	2.91 ± 2.54
w	8.03 ± 6.26	6.13 ± 4.7	5.24 ± 4.48	5.81 ± 4.89
h_1	3.28 ± 2.77	2.94 ± 2.4	2.35 ± 1.89	2.89 ± 2.37
h_2	2.4 ± 1.88	2.05 ± 1.71	2.28 ± 2.04	2.05 ± 1.71
θ	0.07 ± 0.05	0.06 ± 0.05	0.076 ± 0.05	0.06 ± 0.05

As a matching score we use the normalized cross correlation measure between the warped database eye and the eye in the new directly looking video. The ROI for this comparison is the region of the eye, slightly enlarged, and not a rectangular frame.

A threshold is used to determine cases in which the method of searching for the eye parameters failed to produce good database to image matches. Typically, the process produces one pair of eyes for 80% of the frames in the training video.

5. Runtime System

The runtime system replaces frames whenever the eye is open. During blinks no replacement is done. Right after the blink, the system is reinitialized and starts similarly to the first frame. Remarkably, the lack of intervention during blink frames does not seem to hurt the quality of the resulting video.

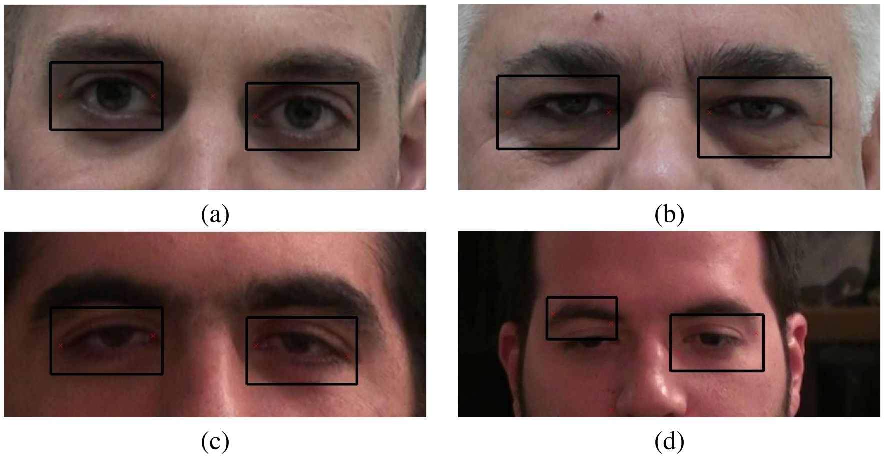


Figure 3. (a-c) Samples of successful facial feature points detection obtained using the method of [5]. The ROI around the eyes are marked by a black rectangle. (d) An example of a failure case in detection.

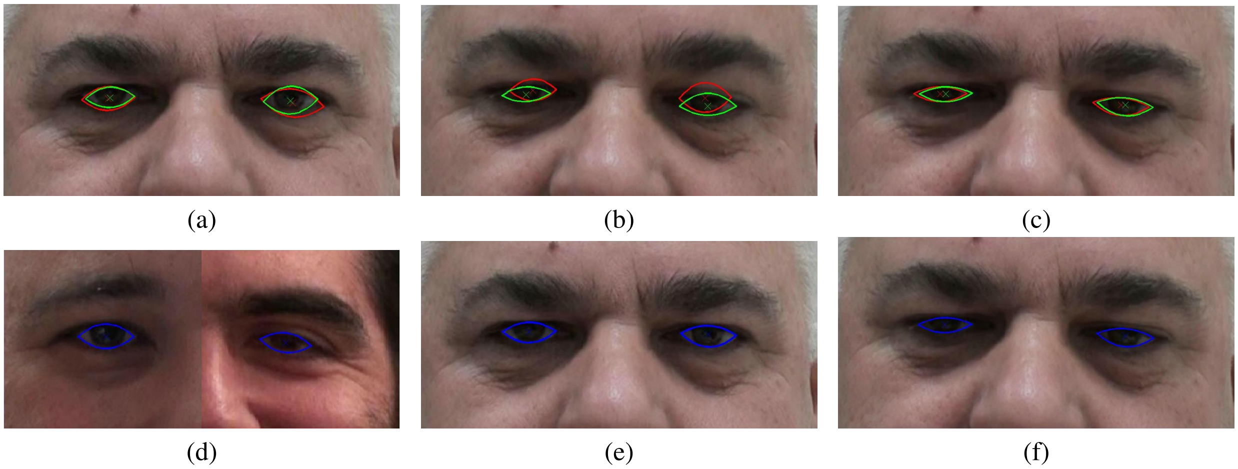


Figure 4. Initially fitted model (red) and refined model obtained by the search procedure (green). (a) During the online database construction, the initial guess is the nearest neighbor (for each eye separately) in the SIFT feature space from among the training examples of the offline database. (b) In the first frame of the videoconferencing session, the initial guess for each eye is the highest correlated example in the online database. (c) In the tracking phase, the initial guess is the eye-model of the previous frame. (d) depicts the two eyes (from two different individuals) from the offline database that were used as an initial guess for the eye model of (a). (e) the initial models for (b) are taken from this frame of the online database. (f) the initial model for (c) is the taken from the previous video frame shown here.

Table 3. Mean (± standard deviation) error in pixels for each parameter of the eye model after the refinement stage. The errors were estimated in a leave-one-out fashion on the offline dataset

Parameter	Left eye	Right eye
x_c	2.04 ±1.80	1.75±1.52
y_c	1.86 ±1.72	1.60±2.02
w	6.66 ± 5.03	4.29 ± 3.70
h_1	3.68 ±3.42	2.54±1.93
h_2	2.22 ±1.95	2.20±1.83
θ	0.08 ±0.06	0.06±0.04

5.1. Eye detection

In the first frame, the best matching set of eyes are searched for. This is done by using the normalized correlation measurements to compare the learned eye models to an eye shaped ROI situated between the corners of the eyes and at a width of $0.7w$. Notice that we do not use SIFT descriptors here, since our goal is to find eyes that are similar in appearance. Such eyes are more likely to produce minimal artifacts during replacement.

After the closest direct looking pair of eyes is found, they are morphed in order to better fit the new frame. This is done by a search process over the six eye parameters, similar to the search done during training. Here, unlike the training phase, the normalized cross correlation matching score is used.

Figure 4(b) shows the model eyes and how they were fit to the new frame before and after the search stage.

5.2. Eye tracking

The eye parameters must be estimated for every frame. Given a new frame, the eye parameters of the previous frame serve as an initial guess, and a search (i.e. refinement) process is once again conducted. The matching score is composed of two components: one considers the sum of squared differences (SSD) between the eye in the current frame and the eye in the previous frame, where the latter is warped to the new frame in accordance with the difference in the eye parameters; The second considers the SSD between the current eye and the warped eye from the first frame.

SSD is used since illumination changes between consecutive frames are expected to be small and since it is convenient to combine multiple SSD scores. The combined cost term minimizes drift over time. Without it, small tracking errors would accumulate. A noticeable example would be for the eye to gradually shrink, see Figure 5.

When the search process performed during tracking fails, a blink is declared, and the system enters a blink mode. While in this mode, the eye parameters do not adapt, and no eye replacement takes place. During every blink frame, tracking based on the last detected eye model is attempted. Once this tracking is successful for both eyes for at least 2 consecutive frames, the blink state is terminated.

The first frame after the blink mode is treated as the first frame of the video sequence, in order to allow the system to move to a more suitable set of eyes. In case the eye corner detector fails, the last model of an open eye is used to initialize the eye model search process. Although the pair of replacement eyes used in between blinks does not change, this effect is unnoticeable.

5.3. Eye replacement

The eye replacement is done by pasting the warped set of model eyes onto the eye location in the image. The warping is done in order to adjust the parameters of the model eyes to the parameters of the actual eyes in the video frame.

In order to eliminate artifacts caused by this pasting operator several measures are taken. First, the eyes are enlarged by a factor of 10% in the vertical direction. This compensates for underestimation of the height of the eye due to the change in gaze between the model eyes an the actual eyes, and, in addition, makes sure that there are no residue pixels from the original eye.

A second measure done in order to ensure smooth integration is the adjustment of image illumination. To this end, a low pass quotient image [12, 10, 15] is applied, similar to what is done in [2] for swapping entire faces in images.

For both the original pixels to be replaced and the new eye to be pasted, we estimate the illumination by fitting a third order polynomial to the image data [1].

Denote the Red value of pixel (x, y) in the original video frame as $I_R^{(1)}(x, y)$, and the values in the pasted eye image as $I_R^{(2)}(x, y)$. Let $\hat{I}_R^{(1)}(x, y)$ and $\hat{I}_R^{(2)}(x, y)$ be the corresponding values of the fitted low-order polynomial:

$$\hat{I}_R^{(1)}(x, y) = \sum_{i=0}^{3}\sum_{j=0}^{3-i} \beta_{R,ij}^{(1)} x^i y^j$$

$$\hat{I}_R^{(2)}(x, y) = \sum_{i=0}^{3}\sum_{j=0}^{3-i} \beta_{R,ij}^{(2)} x^i y^j$$

The fitting is done to each of the three channels R,G,B separately using a least square system of equations (10 unknown βs per image per channel).

Using quotient image reillumination, the new image values $\widehat{I}_R^{(1)}(x, y)$ are given by:

$$I_R^{(1)}(x, y) = I_R^{(2)}(x, y) \frac{\hat{I}_R^{(1)}(x, y)}{\hat{I}_R^{(2)}(x, y)}$$

To further ensure unnoticeable pasting, we use a simple feathering technique in which on a strip of 10 pixels around the replaced region, blending with linear weights is performed between the old image and the pasted eye.

6. Results

We have tested our system on a number of sequences. All sequences are captured at 1280×1024 pixels at 25 frames per second. The offline database consists of 200 images of 14 individuals with manually specified eye models. At the beginning of the videoconferencing session, we asked each user to look straight at the camera for a couple of seconds. These frames became the online database that was used for the remaining of the session of each individual.

Fig. 7 shows frames from some of the video sequences we processed. We properly detect the eye and seamlessly replace it with a direct gaze eye image. In addition, the system can automatically detect blinks. During blinks the input image remains unaltered. Also demonstrated are some of the limitations of the system. For example, in the third row, right image, we observe that the person is looking sideways, but not down. Still, our system replaces the eye with a direct gaze image. Also, in the bottom right image, the person is twitching his left eye. We correctly replace the eye, but also affect some of the surrounding skin.

The system also maintains a very high level of temporal consistency with no flickering (i.e., the eye does not change

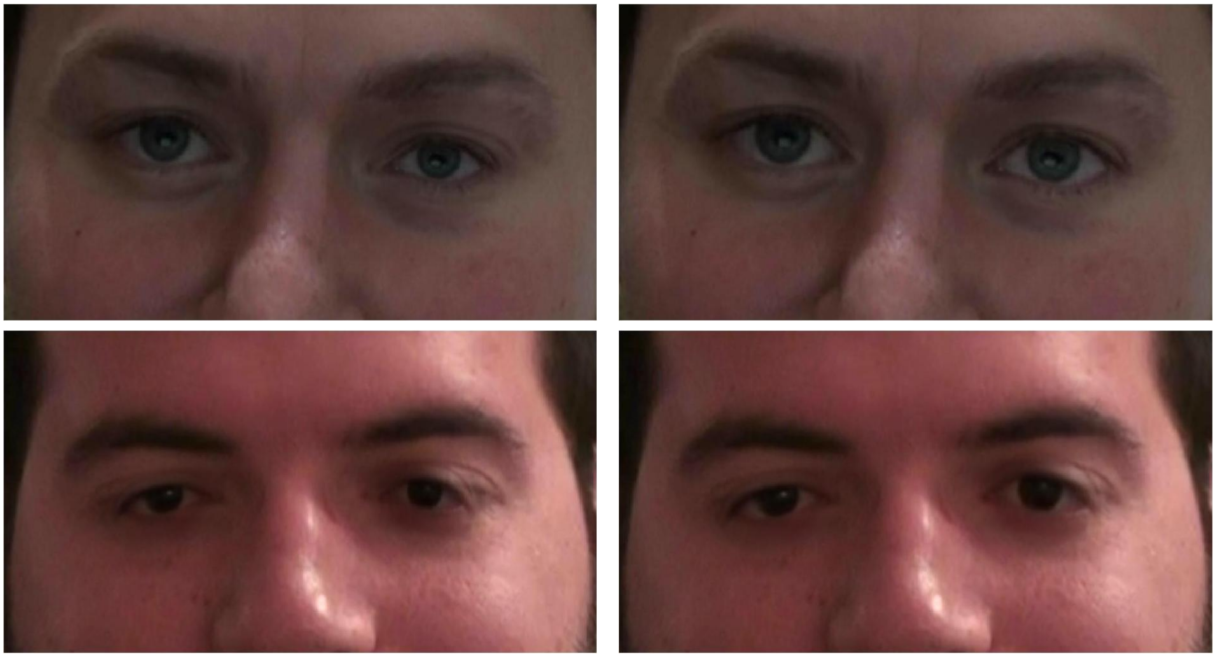


Figure 5. The need for a combined tracking cost function that takes into account both the first frame and the previous frame. Left column: Result of eye replacement after tracking the eye model while only considering the previous frame. Right column: Result of eye replacement after tracking the eye model while considering both the previous frame and the first frame. Notice how the eye shrinks if we do not use the first frame as well.

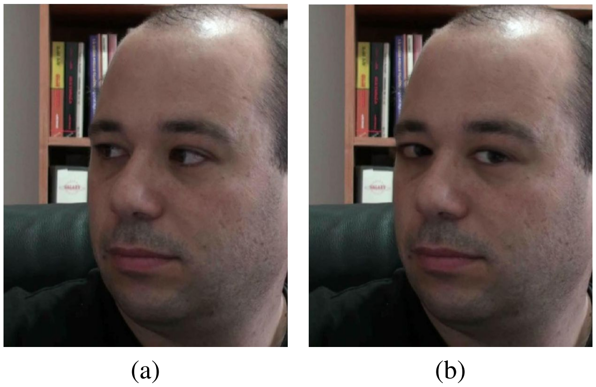
its size from frame to frame in an irritating manner). Please refer to the supplemental video for additional results.

7. Limitations and Future work

The method works very well but there is still room for improvement. First, the system is limited by the quality of the tracker. Currently, the system cannot handle large head motions. However, adding a head tracker should solve this problem. We also plan to allow a better control of the direction of the gaze. For example, in a multi-party conference call, we can render the gaze differently for every viewer to reflect the information of who is looking at who. As can be seen in Figure 8 example-based gaze replacement works even for large head rotations and in replacing left to right gaze directions. We plan to address these issues and build a complete end-to-end video conferencing solution.

8. Conclusions

We proposed a practical solution for gaze correction in videoconferencing systems. The method accurately tracks the eyes of the user and replaces them with images of eyes with correct gaze direction. To achieve this we introduce a short initialization phase in which the user is looking straight into the camera. The images collected in this stage are then used to replace the eyes in the rest of the video conference. We tested our system on a number of video clips and found that it produces very realistic results.

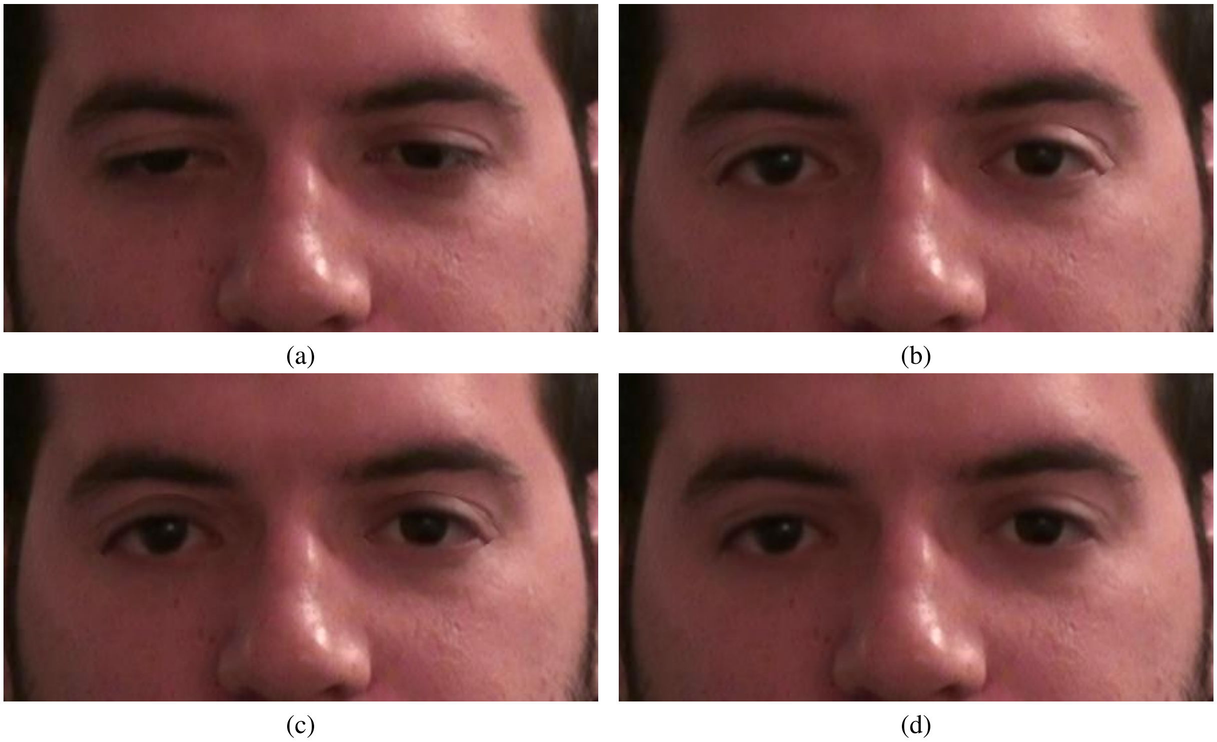


(a) (b)

Figure 8. Gaze replacement works even for large head rotations and for replacing left to right gaze direction. Since we do not have the appropriate trackers yet, the eye parameters are specified here manually. (a) original image. (b) after eye replacement.

Acknowledgments

Lior Wolf is supported by the Israel Science Foundation (grant No. 1440/06), and The Ministry of Science and Technology Russia-Israel Scientific Research Cooperation.



(a)

(b)

(c)

(d)

Figure 6. The importance of various component in the eye replacement procedure.(a) The original image of the eye to be replaced. (b) Eye replacement without the quotient image reillumination stage. Observe that the highlight on the left eye lid of the subject is not consistent with the input image. (c) Eye replacement with the quotient image and without feathering. The highlight is removed, but additional edges are created around the right eye. (d) A complete eye replacement example. The illumination is corrected and the spurious edges are suppressed by feathering.

References

[1] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *PAMI*, 25(2):218–233, Feb 2003. 5

[2] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar. Face swapping: automatically replacing faces in photographs. In *SIGGRAPH*, 2008. 5

[3] A. Criminisi, J. Shotton, A. Blake, and P. H. S. Torr. Gaze manipulation for one-to-one teleconferencing. In *ICCV*, 2003. 2

[4] L. Ding and A. Martinez. Precise detailed detection of faces and facial features. In *CVPR*, 2008. 2

[5] M. Everingham, J. Sivic, and A. Zisserman. “Hello! My name is... Buffy” – automatic naming of characters in TV video. In *BMVC*, 2006. 2, 3, 4

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 2005. 3

[7] J. Gemmell, K. Toyama, C. L. Zitnick, T. Kang, and S. Seitz. Gaze awareness for video-conferencing: A software approach. *IEEE MultiMedia*, 2000. 1

[8] K.-M. Lam and H. Yan. Locating and extracting the eye in human face images. *Pattern Recog.*, 1996. 2

[9] Y. li Tian, T. Kanade, and J. Cohn. Dual-state parametric eye tracking. In *Automatic Face and Gesture Recognition*, pages 110–115, 2000. 2

[10] Z. Liu, Y. Shan, and Z. Zhang. Expressive expression mapping with ratio images. In *SIGGRAPH*, 2001. 5

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 3

[12] T. Riklin-Raviv and A. Shashua. The quotient image: Class based recognition and synthesis under varying illumination conditions. In *CVPR*, 1999. 5

[13] V. Vezhnevets and A. Degtiareva. Robust and accurate eye contour extraction. In *Graphicon*, 2003. 2

[14] P. Viola and M. Jones. Robust real-time face detection. In *CVPR*, volume 2, page 747, 2001. 3

[15] Y. Wang et al. Face relighting from a single image under arbitrary unknown lighting conditions. *PAMI*, 31(11):1968–1984, 2009. 5

[16] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *IJCV*, August 1992. 2

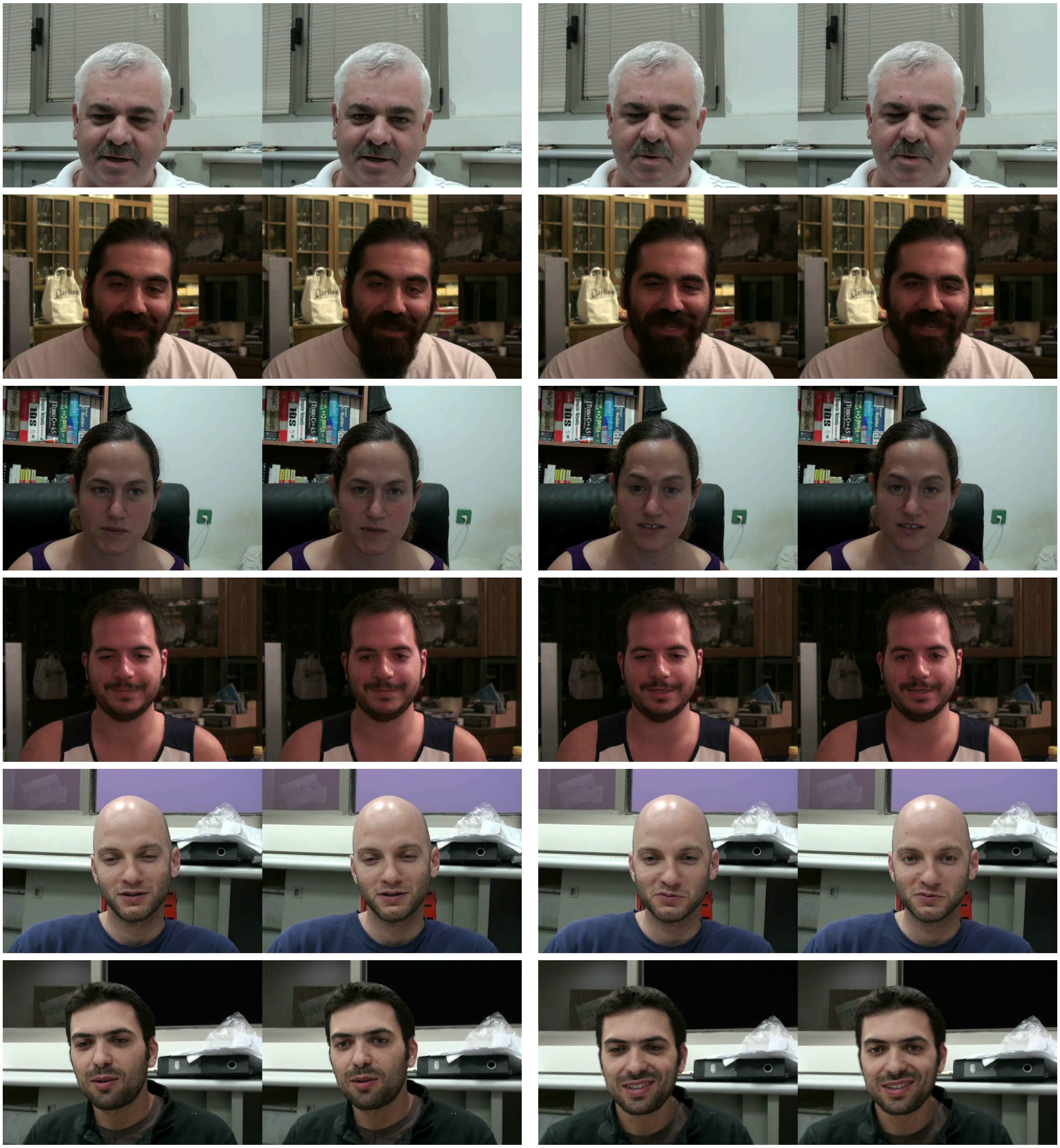


Figure 7. Results on a number of sequences. In each image pair, the left image is the original image and the right image is the modified image. The top right and bottom left images show cases in which the system automatically detects a blink and does not modify the image at all. There are also several failure cases. The left image on the third row demonstrates one of the limitations of our system. They eyes are looking sideways and not down, but our system still replaces them. The right image in the bottom raw shows a case where the eye replacement slightly modifies the eye edge as well.