Virtual Gaze Redirection in Face Images

Dorian Weiner and Nahum Kiryati
Department of Electrical Engineering–Systems
Tel Aviv University
Tel Aviv 69978, Israel
nk@eng.tau.ac.il

Abstract

In video-conferencing, the participant watches the screen rather than the camera. This turns the gaze direction away from the other party and makes eye contact impossible. Artificial gaze redirection by iris repositioning was proposed, but the feasibility of crucial algorithmic components, especially the challenging computer vision parts, was not demonstrated. This paper demonstrates a system for gaze redirection in color face images via eye synthesis and replacement. The proposed method can serve as an image editing tool for entertainment and commercial use, and is an important step towards gaze-redirection in video-conferencing.

1. Introduction

In video-conferencing and other visual communication scenarios, the participant watches the display rather than the camera. This turns the gaze direction away from the other party and eliminates eye contact. One way to avoid the problem, followed in the Hydra system [9], is to use a small display/camera unit and place it sufficiently far away from the user, so that gazing at the screen will be indistinguishable from gazing at the camera. Alternatively, by using a projector and a semi-transparent screen, the camera can be positioned directly behind the display [7]. The need for special hardware limits, however, the acceptance of this method.

Zitnick *et al* [15] explored the possibility of modifying the gaze direction using software only, by combining computer vision and graphics techniques. Their study focused on the graphics components and left most of the vision problems for future research. Two general approaches were considered. One is based on detecting and masking the visible parts of the eyeballs and replacing them with synthetic eyes looking in the desired direction. The other calls for rotation of the whole head.

Because of its conceptual simplicity, the first approach, of eye synthesis and replacement, is very appealing. It is most useful in cases where the necessary changes in the gaze direction are small. Since only the eyes are processed, the method can be expected to require less computing resources than the head reorientation approach, and may be more appropriate for real-time implementation.

Gaze redirection using the eye synthesis approach requires to accomplish several computer vision tasks. First, the eyes have to be detected. Second, very accurate segmentation of the eyes must be achieved: the original eyes must be fully masked, but the surroundings must be left intact. Third, in order to fit the colors of the synthetic eye to those of the real eye, the eye has to be internally segmented. These problems were not addressed in [15].

This research evaluates and verifies the overall feasibility of gaze redirection via eye synthesis and replacement. We developed the necessary algorithms, vision and graphics, and integrated them within a demonstration program, inspired by the well known “xeyes” program (believed to be written by Jeremy Huxtable circa 1988). In “xeyes”, the pupils within schematic eyes follow the mouse on the screen. Our system receives a real face image as input, automatically carries out the necessary vision and graphics tasks, and shows the face with the eyes following the mouse in a realistic way.

The main contribution of this research is the successful integration of an automatic system for gaze redirection via eye synthesis and replacement. Its vision components include original algorithms as well as adaptations of methods from the vast literature on face detection and facial feature extraction, e.g. [1, 2, 3, 4, 5, 8, 10, 11, 14, 12, 13]. The graphics part, i.e., the creation of the synthetic eye, follows a path different than that of [15]. The sclera (eye-white) and iris are adapted from parts of real eye images, while in [15] they are created from scratch. The performance of the system has been evaluated using images from the University of Oulu Physics-Based Face Database [6] and images taken locally. All the results were obtained without any change in

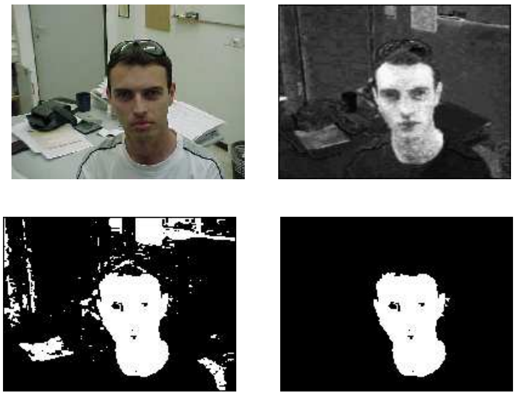

Figure 1. Top-Left: Original image. Top-Right: Similarity to skin color. Bottom-Left: After thresholding. Bottom-Right: The largest connected component.

parameters.

This research focuses on gaze redirection in still images. In addition to being an important step towards gaze-redirection in video conferencing, the method presented is immediately useful as an image editing tool, with potential applications in graphic design, advertising, psychological research, forensic science, visual effects and entertainment.

2. Extracting the Eyes

Given an input face image, it is necessary to detect the eyes and to very accurately segment them, both externally and internally. This section outlines the multi-stage process used to accomplish this task.

2.1. Face detection

Since a color input image is assumed, the face is detected via skin detection. It is well known that regardless of ethnic origin, skin colors tend to cluster in intensity-normalized color spaces (assuming reasonable illumination). In this work, skin is detected using the Cr (chromatic red) and Cb (chromatic blue) components of the $YCrCb$ color space. For each pixel, the similarity of its color to skin color (modeled as a Gaussian probability density function) is evaluated, normalized and thresholded. The largest connected component in the resulting binary image is taken to be the face. See Fig. 1.

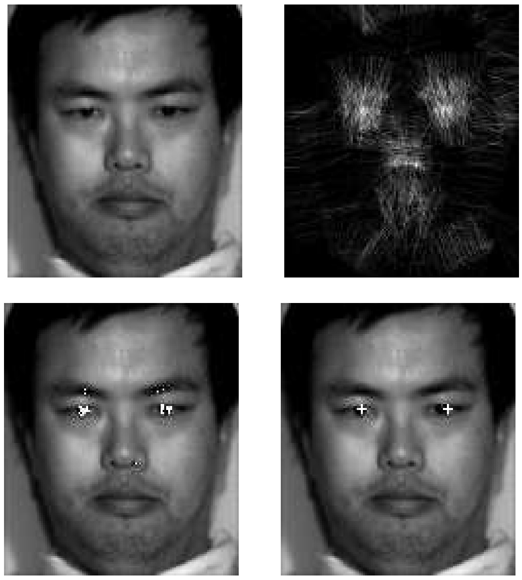

Figure 2. Top left: The red component of a face region (from the Oulu face database). Top right: Accumulator array contents. Bottom left: Candidate iris centers. Bottom right: Coarse eye detection.

2.2. Coarse eye detection

Following face detection and extraction, coarse eye detection is carried out, based on the circularity of the iris. Only the red component (R channel) of the picture is used at this stage. Edge detection is applied, and the results are forwarded to a variation of the Hough Transform for circles with unknown radius using a two dimensional accumulator array. Voting takes place along rays in the direction opposite to the gradient. Fig. 2 shows the red component of a face region (top left) and the resulting accumulator array values (top right). Among the significant peaks we select the pair that is most likely to correspond to the two iris centers, as shown in Fig. 2 (bottom left and bottom right).

2.3. Fine iris detection

Fine determination of the iris center position and the iris radius are performed by applying another version of the Hough transform for circles to a rectangular "bounding-box" containing the eye. Canny edge detection is applied to the red component. Gradients pointing upward are discarded, since the upper part of iris is usually concealed by the eyelid.

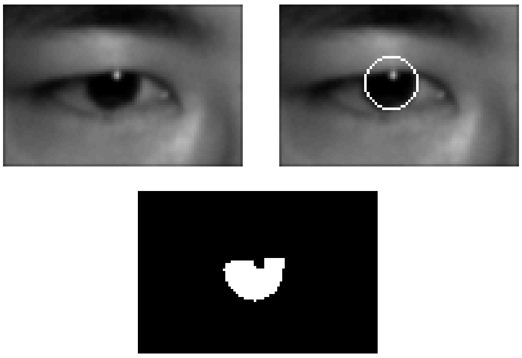

Figure 3. Top-left: Eye window (red component). Top-right: Detected iris. Bottom: Extracted iris.

Detecting circles with unknown radii (that need to be determined) requires a three dimensional accumulator array. The range of radii that have to be considered is however small, since the distance between the (coarse) eye centers provides an initial rough estimate of the iris radius. The accumulated values are normalized by a function of the radius.

The Hough transform relies exclusively on edge data. In our case, the iris is known to be much darker than the sclera. This fact can be used to improve the estimation, by seeking to maximize the contrast between the estimated iris and the sclera. Having found the highest accumulation (and the corresponding iris center coordinates) for each of the possible radius values, we further weight each of them by the difference between the average intensities in the estimated iris and sclera regions associated with each radius value. The operation of the algorithm is exemplified in Fig. 3 (top left and right).

2.4. Iris extraction

The iris center and radius found in the previous step are not sufficient for iris extraction, because in most cases the iris is partially occluded by the eyelids. The iris is extracted using the fact that it is darker than its surroundings. The (gray) intensity image of the eye window is thresholded, using an adaptive threshold designed to be higher than the intensity of iris pixels but lower than the intensity of eye white or skin. Dark pixels located outside the circle defined by the iris center and radius are also excluded. The outcome is shown in Fig. 3 (bottom).

In many cases, this simple algorithm provides good iris segmentation. However, for face images with overhead lighting, the upper part of the eye is shadowed by the eye socket and the eyebrow. These shadowed areas are dark and close to the iris and will be extracted with the iris. This problem is solved using the eye-white extraction results, as discussed below.

2.5. Eye-white extraction

The white parts of the eye are extracted via color discrimination. Within the rectangular bounding box that contains the eye, the nearly white color of the sclera can be distinguished from skin, iris and eyebrows. We carry this out by measuring and thresholding the distance between the color of a given pixel and the color distribution of eye-white as measured in a sample set. The rgb (normalized RGB) color space is used. In some images, white light reflections appear within the iris, but this is not a problem since the iris is extracted as well and later combined with the sclera anyway to obtain the complete eye mask. Fig. 4 (left) shows an eye area taken from a real image. The color distance from eye-white is visualized on the right. The outcome of thresholding the distance is shown in Fig. 4 (right), demonstrating successful extraction of the white sclera (together with a reflection within the iris).

The precise color of the sclera differs between people and is also dependent on the lighting conditions. Even within a given eye, there is some spatial variability. The threshold is therefore (automatically) adapted to the given image and to its spatial variation.

2.6. Combining the iris and eye-white

The extracted iris can provide valuable information for the eye-white extraction procedure, and vice versa.

The eye-white extraction process may yield some connected components that do not belong to the sclera. To exclude them, only connected components close to the left and right sides of iris are accepted.

In some images, the area above the iris is shadowed, and because of its low intensity it may be incorrectly extracted with the iris. We use the extracted eye-white components to

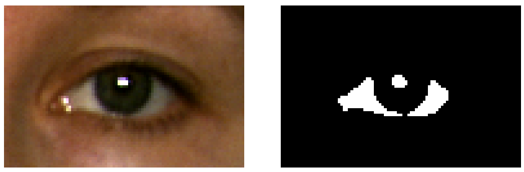

Figure 4. Left: Eye area from original image. Right: Eye-white extracted by thresholding the color distance from eye-white.

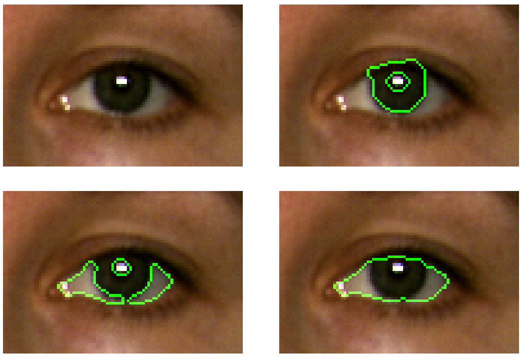

Figure 5. Top-Left: Eye area from an original image. Top-Right: Extracted iris. Bottom-Left: Extracted eye-white. Bottom-Right: Combined iris and eye white. Observe the exclusion of the shadow above the iris.

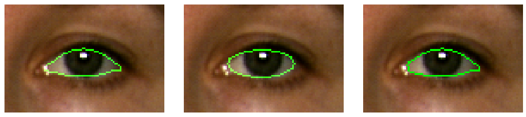

Figure 6. Left: Parabolas fitted to the upper and lower boundaries of the eye element. Middle: Ellipse fitted to the edges of eye element. Right: Segmented eye, obtained by combined use of the fitted parabolas and ellipse.

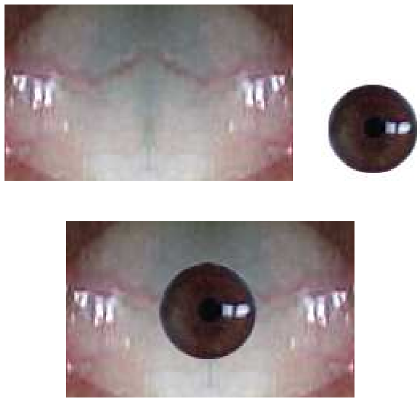

Figure 7. Top-Left: The white part of the synthetic eye. Only small parts of the reflections are eventually visible, near the eye corners. The upper part is intentionally darker, to imitate the effect of shadows caused by the eye socket and brow. Top-Right: Iris model. Bottom: Combined synthetic eye.

remove the shadow above the iris. To accomplish this, the two upper extremity points of the eye-white next to the iris are located. The boundary of the upper eyelid near the iris is approximated by a circle of radius 3ρ that passes through them, where ρ is the radius of the iris. The combination of iris and eye-white is exemplified in Fig. 5.

2.7. Curve fitting

The edges of the eye element found in the previous steps are usually ragged. To smooth them, parabolas are fitted to the upper and lower parts of the eye via least squares. Additional processing is needed for removing the tear gland from the eye. An ellipse is fitted to the upper and lower edge points of the eye element, combined. Let c_L and c_R denote the left and right focal points of the ellipse. For the right eye (from the observer's point of view), pixels outside the ellipse and left of c_L are removed from the eye element. Similarly, for the left eye, pixels outside the ellipse and right of c_R are removed. This process is demonstrated in Fig. 6. The resulting eye elements are used as masks for defining the alpha channel (transparency) in the eye replacement method.

3. Eye Synthesis

In reference [15], the starting point for eye synthesis was the drawing of two circles, for the iris and the pupil, on white canvas. To give the eye a more realistic look, a dark circle was drawn around the edge of the iris (limbus), and highlights or reflections were added. Additional refinements were suggested in [15] but not implemented. Our approach is different, using parts of a real eye image as the basis for eye synthesis.

We created the white part by cropping a region from a real eye image, stretching and duplicating it until a sufficiently large area has been obtained. Since the upper part of the eye is usually shadowed by the eye socket and eyebrow, the intensity in that part was reduced. The outcome is shown in Fig. 7 (top-left). Note that only small parts of the highlights at the far left and right regions are usually visible, at the eye corners.

The starting point for the synthetic iris is an an iris from a real eye image, see Fig. 7 (top right). It is positioned at the center of the white area as shown in Fig. 7 (bottom). In order to approximate the true iris color, it is sampled in the input image and used to set the iris color in the synthetic eye.

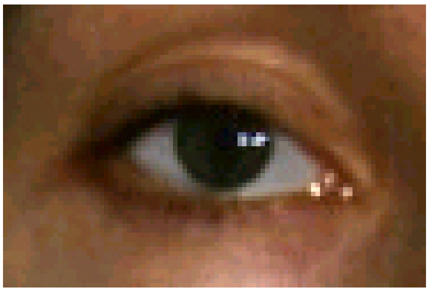

Figure 8. Synthetic eye displayed instead of a real eye.

We use the transparency attribute (alpha channel) for displaying the synthetic eye instead of the real eye. In order to soften the transition between the input image and the synthetic eye, the eye masks obtained from the segmentation process are smoothed with a Gaussian filter.

The human eye is nearly spherical in shape. When gazing in the left or right directions, the iris appears elliptical. To render this effect, the horizontal size of the iris is slightly reduced as a function of its horizontal displacement. Finally, the synthetic eye is resized according to the size of the iris in the input image. An example of a synthetic eye, positioned in place, is shown in Fig. 8.

By moving each eye in a slightly different, well calculated way, the gaze can be directed at a desired point in three dimensional space, see [15]. In the experimental results shown in this paper, both eyes are moved by the same amount with respect to the original iris positions. Fig. 9 shows the overall eye replacement outcome. Establishment of virtual eye contact using gaze redirection is demonstrated in Fig. 10.

Figure 9. Left: Original image (from the Oulu face image database). Right: Image with synthetic eyes, looking to the side.

Figure 10. Left: Original image. Right: Establishment of virtual eye-contact via gaze redirection.

4. Testing

The eye detection and extraction procedure has been tested using the ‘aa’ sub-library of the University of Oulu Physics-Based Face Database [6]. It contains 125 color face images, 428×569, taken with incandescent illumination using a 3-CCD Sony DXC-755P camera. All subjects in this series of images were not wearing eyeglasses. Five images were removed from the test set: three in which the subjects had their eyes completely closed, one in which an eye was occluded by hair and one which was significantly out of focus. Successful gaze redirection via eye synthesis and replacement requires extremely accurate segmentation of the eyes. Thorough examination of the results revealed that good segmentation has been obtained for 108 images out of 120.

For images that were correctly segmented, no difficulties were encountered in the eye replacement stage. Note however, that since the eyelid positions are not modified, excessive vertical iris displacements may lead to unnatural facial expressions.

The computer vision part of the algorithm (face and eye detection, eye segmentation) was implemented in Matlab (version 5.3). The graphics part (creating the synthetic eye, eye replacement, gaze control) was written using Java 2. On a personal computer (Celeron 333MHz with 128MB RAM), processing a typical image from the Oulu face database takes about 14 seconds, of which 10 seconds are spent in the Matlab environment and the rest in Java (including image loading and display). Much faster operation can obviously be achieved by re-implementing the programs in a compiled programming language such as C. Since image resizing is employed in early stages of the algorithm, the dependence of processing time on the overall image size is moderate. Note, however, that the full resolution is maintained in the eye-areas. Roughly, processing time grows linearly with iris radius.

5. Conclusions

We presented an automatic system for gaze redirection in color face images based on eye synthesis and replacement, and demonstrated its performance. These results prove the overall feasibility of the system hypothesized by Zitnick *et al* [15]. In fact, combining the graphics module of [15] with our vision module (section 2) would allow to compare our method of eye synthesis (section 3) with that of [15].

Gaze redirection in video is the obvious next step. Much can be gained from the inter-frame redundancy within video sequences. Since the changes between consecutive frames are usually small, parameters such as the locations of the face and the eyes, the radius of the iris and its color need not be determined independently and from scratch in each frame. This can lead to improved reliability and dramatic reductions in the computational load. Additional work will be needed to deal with large head movements. Another cause for concern is the lower spatial resolution of video systems. But note that the inter-frame redundancy and the small size of the regions of interest create an ideal setting for dynamic super-resolution algorithms, that can bring resolution in the eye areas up to the level necessary for the segmentation algorithms used in this research. Overall, we expect that by progressing from stills to video, less computational resources will be needed per-frame and robustness can be increased.

Acknowledgments

We are grateful to Antti Ylä-Yääski (Espoo, Finland) for directing our attention to the gaze correction problem. This research was supported by the Internal Research Fund of Tel Aviv University.

References

[1] T.C. Chang and T.S. Huang. Facial feature extraction from color images. *Proc. ICPR*, 1:39-43, 1994.

[2] G. Chow and X. Li. Towards a system for automatic facial features detection. *Pattern Recognition*, 26:1739-1755, 1993.

[3] A. Haro, M. Flickner and I. Essa. Detecting and tracking eyes by using their physiological properties. *Proc. CVPR*, 163-168, 2000.

[4] R. Kothari and J.L. Mitchell. Detection of eye locations in unconstrained visual images. *Proc. ICIP*, 19A8, 1996.

[5] K. Lam and H. Yan. Locating and extracting the eye in human face images. *Pattern Recognition*, 29:771-779, 1996.

[6] E. Marszalec, B. Martinkauppi, M. Soriano and M. Pietikäinen. A physics-based face database for color research. *Journal of Electronic Imaging*, 9:32-38, 2000.

[7] K.I. Okada, F. Maeda, Y. Ichikawa and Y. Matsushita. Multiparty videoconferencing at virtual social distance: MAJIC Design. *Proc. CSCW*, 385-395, 1994.

[8] E. Saber and A.M. Tekalp. Frontal-view face detection and facial features extraction using color, shape and symmetry based cost functions. *Pattern Recognition Letters*, 19:669-680, 1998.

[9] A.J. Sellen. Remote conversations: The effects of mediating talk with technology. *Human-Computer Interaction*, 10:401-444, 1995.

[10] S.A. Sirohey and A. Rosenfeld. Eye detection in a face image using linear and nonlinear filters. *Pattern Recognition*, 34:1367-1391, 2001.

[11] Y. Tian, T. Kanade, and J.F. Cohn. Dual-state parametric eye tracking. *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, 110-115, 2000.

[12] M.H. Yang and N. Ahuja. *Face Detection and Gesture Recognition for Human-Computer Interaction*, Kluwer, Boston, 2001.

[13] A.L. Yuille, P.W. Hallinan and D.S. Cohen. Feature extraction from faces using deformable templates. *Int. J. of Comp. Vision*, 8:99-111, 1992.

[14] R.P. Wildes, J.C. Asmuth, G.L. Green, S.C Hsu, R.J. Kolczynski, J.R. Matey and S.E. McBride. A machine vision system for iris recognition. *Machine Vision and Applications*, 9:1-8, 1996.

[15] C.L. Zitnick, J. Gemmell and K. Toyama. Manipulation of video eye gaze and head orientation for video teleconferencing. *Microsoft Research Technical Report MSR-TR-99-46*, June 1999.