



PERGAMON

Pattern Recognition 35 (2002) 1389–1401

PATTERN  
RECOGNITION

THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

# A method of detecting and tracking irises and eyelids in video

S. Sirohey<sup>1</sup>, A. Rosenfeld\*, Z. Duric<sup>2</sup>

*Center for Automation Research, Institute for Advanced Computer Studies, Computer Vision Laboratory,  
University of Maryland, College Park, MD 20742-3275, USA*

Received 20 February 2001; received in revised form 12 April 2001; accepted 30 April 2001

## Abstract

We locate the eye corners, eyelids, and irises in every frame of an image sequence, and analyze the movements of the irises and eyelids to determine changes in gaze direction and blinking, respectively. Using simple models for the motions of the head and eyes, we determine the head-independent motions of the irises and eyelids by stabilizing for the head motion. The head-independent motions of the irises can be used to determine behaviors like saccades and smooth pursuit. Tracking the upper eyelid and using the distance between its apex and the center of the iris, we detect instances of eye closure during blinking. In experiments on two short image sequences, in one of which the subject was wearing glasses, we successfully located the irises in every frame in which the eyes were fully or partially open, and successfully located the eyelids 80% of the time. When motion information in the form of normal flow was used, the irises were successfully tracked in every frame in which the eyes were fully or partially open, and the eyelids were successfully located and tracked 90% of the time. © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

*Keywords:* Eye detection; Eyelid detection; Iris detection; Gaze tracking; Blink detection

## 1. Introduction

Analysis of the behavior of the eyes has many useful applications. Some of these applications are in human-computer interaction; for instance, the computer may want to know what the user is looking at on the screen. Other applications can be found in compression techniques like MPEG4, where eye information is part of the communication stream. Still another application can be found in driver behavior analysis. In this

application, attentiveness on the driver's part is directly related to safety. For example, the car may want to know when the driver is tired and keeps closing his/her eyes; if the car finds that the driver is not attentive it may send visual or auditory signals to get the driver's attention.

In Ref. [1] we described a method of detecting the corners of the eyes in color images. The corners provide information relating to the sizes of the eyes and the irises through anthropometric averages. Additionally, localization of the eyes in an image provides us with left and right eye regions. These regions are used in this paper to determine the locations of the irises and the eyelids, as explained in Section 3.

The motions of the eye corners from frame to frame determine the motion of the head, as the eye corners represent a rigid portion of the face. Eye gaze direction is determined by the global (image-based) positions of the

\* Corresponding author. Tel.: +1-301-405-4526; fax: +1-301-314-9115.

E-mail address: ar@cfar.umd.edu (A. Rosenfeld).

<sup>1</sup> Also at General Electric Co, Mt. Prospect, IL 60056, USA.

<sup>2</sup> Also at Department of Computer Science, George Mason University, Fairfax, VA 22030-4444, USA.



Fig. 1. Examples of images used in our experiments.

iris centers over the image sequence. To detect eye movements that are independent of head movements (such as saccades or smooth pursuit), after determining the head motion we use eye-centered coordinate systems to define the head-independent motions of the irises. Detection of blinking is also based on iris-centered coordinate systems. Using the fact that the eyelid does not cover the center of the iris of a seeing eye, we determine the vertical position of the eyelid with respect to the iris center, and thus determine when the eye is in the act of closing for a blink. We describe our frame-to-frame iris and eyelid tracking methods and results in Section 4.

In our experiments, we found that frame-to-frame eye part tracking was not always successful, particularly as regards the eyelids. In Section 5 we show how the tracking can be improved by using normal flow information. We use a flow-based method using head, iris, and eyelid motion models to track the eye corners, irises, and eyelids in a video sequence. The iris and eyelid motions have multiple components, one of which is the head motion. We de-couple the head component of motion from the iris and eyelid components, and perform activity analysis for the irises and eyelids based on the head-independent flow information.

In Section 2 we review previous work on methods of detecting and tracking eye parts in video sequences, and point out limitations of these methods. (Literature on eye part detection by active sensing [2,3] and on face tracking and facial expression detection [4–9] is not reviewed here.) The methods described in this paper can automatically detect and track eye parts under fairly general assumptions.

In the experiments in this paper we used short color image sequences that were collected using a high-quality progressive-scan digital video camera. The resolution of the images was  $640 \times 480$ . A restriction imposed on the subject was that both eyes should be visible in the image. This limited the degree of change in head pose. The subject was asked to assume that she was sitting in front of a computer screen and was allowed to perform natural tasks involving blinking, saccades, and tracking objects

on the screen. Two images from our sequences are shown in Fig. 1; note that in one of the images, the subject is wearing glasses.

Eye corner detection as in Ref. [1] was performed on these sequences frame by frame at half ( $320 \times 240$ ) the original resolution. We used anthropometric averages based on these corners to extract regions containing the left and the right eye.

The images were converted from color to a 256-level grayscale. Edge detection, cleaning, and labeling were performed on these image; for the details see Ref. [10]. The result was a set of edge segments, terminating either at end points or at junctions with other edge segments.

## 2. Literature review

Several methods of detecting and tracking eye parts in video have been described in the literature, but these methods all have limitations.

In Ref. [11] a deformable template, similar to the one in Ref. [12], is used to describe the shape of the eye and thus identify eye behavior. The template is hand-initialized by manually locating the eye region. Its parameters are also similarly initialized. Once this is done the template is allowed to deform in an energy minimization manner. The position of the template in an initial frame is used as a starting point for deformations that are carried out in successive frames. The values of the parameters in each frame are used to identify eye behavior.

Ref. [13] describes a method of tracking the three-dimensional (3D) motion of the iris in a video sequence. The main theme is to determine the torsional (rotational) component of iris movement that takes place when the eyes converge to fuse the pair of images into a single binocular image. A five-parameter scalable and deformable model is developed to relate horizontal and vertical translations, rotation (torsion), uniform scaling due to changes in eye-camera distance, and partial scaling due to expansion and contraction of the pupil. The method requires very high-quality and high-resolution

images; a typical resolution is  $240 \times 180$  pixels showing only the eye. The initial iris position is manually determined and templates are used to track the iris motion. This method is able to track the iris very well as long as the rotational components are relatively small. The authors mention that it failed when the rotation became larger than  $10^\circ$ . This was due to the spherical nature of the eye which the fronto-parallel model was not able to cope with.

In Ref. [14] the outline of the head is found by using snakes; the approximate positions of the eyes are then found by anthropometric averages; finally, the corners of the eyes are found using edge information. The corners are detected as regions that have high edge curvature values. Four corners are located for each eye: the left and right corners of the eye as well as the corners formed by the iris and the upper eyelid. After locating these corners an energy minimization approach is used to fit a model of the eye to the image. This model consists of parabolas for the eyelids and a partial circle for the iris. A speedup is obtained over the method used in Ref. [12] by using the positions of the corners of the eye. This method requires the presence of four corners on the eye, which occur only if the iris is partially occluded by the upper eyelid. When the eyes are wide open, the method fails because it cannot find corners that do not exist.

In Ref. [15] an iris tracking scheme based on Kalman filtering is described. The eye region is determined using gray level histogram based thresholding and binary search. The irises and the eyelids are modeled by circles and parabolas that have pre-determined parameters. The eye region is divided into seven parts: the whole eye region, two regions representing the sclera, the whole iris, the unoccluded portion of the iris, and the occluded parts of the iris. Under the assumption that the head motion is negligible (this assumption is relaxed in later work [16] using cascaded Kalman filters), the eye region and its parts are tracked using model-based Kalman filtering. The authors performed experiments on two sequences; in one the irises moved to the right, while in the other they moved to the left. In Ref. [16] the authors used the center of the eye to identify the head motion. Their experiments used two cascaded Kalman filters; the first determined the iris motion as explained above, and the second used the center of the eye to determine the head motion. They mention that they translated the image to simulate head motion and used it to determine a stabilized iris motion. Such a scheme would not be able to determine the 3D motion of the head and would work only when the head motion is very small.

### 3. Eye part detection

In this section we describe our methods of iris and eyelid detection.

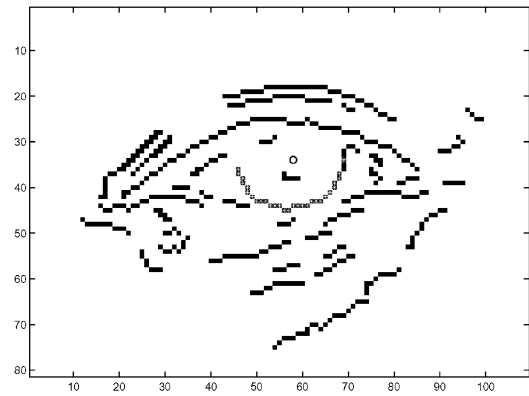


Fig. 2. Edges in an eye region (black squares), and the selected iris center and iris edge (hollow circles).

#### 3.1. Iris detection

We use a method of iris detection which does not require hand initialization and does not restrict the position of the iris in any way. The possible extreme cases that we deal with are: iris rolled to either corner of the eye, eyes wide open (no occlusion of the iris by the upper eyelid), and eyes only partially open (up to half of the iris occluded by the upper eyelid, as in the cases of squinting, blink in progress, or opening after a blink).

In an intensity image the iris is always darker than the sclera no matter what color it is, so that the edge of the iris is relatively easy to detect as a circular arc. Anthropometric averages give a very tight bound on the size of the iris relative to the size of the eye, and iris movements have limits imposed on them by the size of the eye. For a person to be able to see, the upper eyelid should not cover the pupil, so that a majority of the iris must be visible. This to detect the iris, we can look for the lower half of the iris boundary.

The diameter of the iris can be approximated to be one-third of the length of the eye. Let  $r$  represent the iris radius. To allow for variations in the size of the iris we used a tolerance  $\delta r$  of one pixel. We used a simple method to search for the edges of the iris. A semicircular template centered at the center of the iris was applied to the region containing the eye. Since  $\delta r = 1$ , we used a semicircular annulus three pixels thick to select edge pixels located in the annulus region.

We imposed the constraint that the directions of the edge gradient and the normal to the annulus should differ by at most  $\pi/6$ . We chose as the location of the iris the position for which the annulus contains the largest number of edge pixels that satisfy this constraint. Fig. 2 shows an example of iris detection using this method; the edges detected in an eye region are shown as black squares, and the iris center and edge are shown

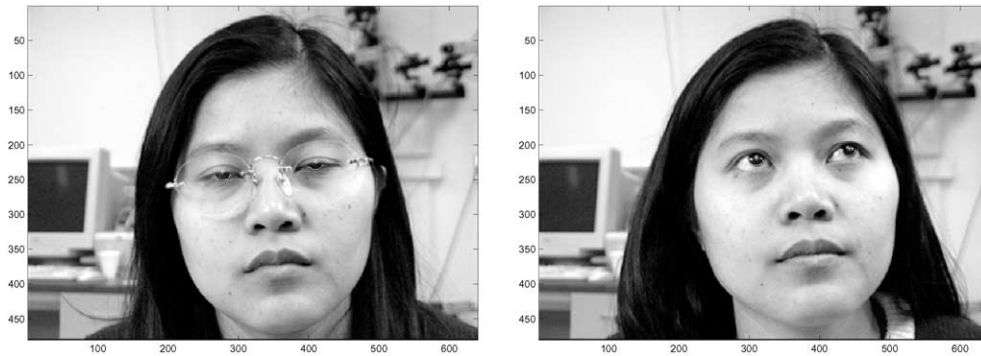


Fig. 3. Iris detection results on the images in Fig. 1.

as hollow circles. Fig. 3 shows detected iris centers superimposed on the images in Fig. 1.

### 3.2. Eyelid detection

The duration of a blink in a video sequence can range from 5 to 15 frames. Feature detection algorithms that rely on the eye being open to locate the eye will not work when the eye is closed. It is therefore important that blinks be detected. Knowledge of the start and the end of a blink in a video stream can then be supplied to feature detection algorithms, thus improving their reliability.

The upper eyelid has great freedom of motion, ranging from wide open to closed, whereas the lower eyelid moves much less freely. To describe normal eye behavior, it is sufficient to determine the movement of the upper eyelid. We used the upper eyelid to detect blinking. The edge of the upper eyelid is deformable; it intersects the sclera and (possibly) the iris. In a closed or near-closed position this edge is almost like a horizontal line, whereas in a wide-open position it is a concave arc. Due to its deformable nature we cannot use a simple template-based method to detect it, as we did for the iris. However, iris detection provides us with additional information that we can use for detecting the eyelid.

For eyelid edge detection we use information about the eye corners and the iris center. The upper eyelid edge in Fig. 2 forms an arc that bounds the eye above the iris center. However, as the eye opens widely, the upper eyelid recedes into the fold on top of the eye, so that the eyelid edge becomes fragmentary.

Depending on the degree of eye opening, the number and shape of the edges representing the eyelid vary. However, if we look for edges along a vertical line going upward from the iris center, the eyelid edge should be the first one that we encounter, followed by the edges formed by the eyelid fold.

We used the following facts in developing an algorithm for detection of eyelid edges:

- The eyelid edge is above the center of the iris when the eye is able to see.
- The eyelid edge is an arc joining the two corners of the eye.
- The more open the eye, the greater the curvature of this arc.

We modeled the eyelid edge as a polynomial with horizontal support bounded by the horizontal positions of the left and right corners of the eye. In Ref. [3] the eyelid was modeled by a second-degree polynomial, but we found that this was not adequate. Due to fragmentation we sometimes had to combine two or three edge segments together to form a representation of the eyelid edge. In doing so, we required that when two segments are combined, their fitted polynomials should have average curvatures (in the support defined by the segments' horizontal extents) of the same sign. When we used a second-degree polynomial, many candidate segments were eliminated by this criterion. We therefore modeled the eyelid as a third-degree polynomial.

We used an over-determined system of equations to fit a third-degree polynomial to the edge pixels in each edge segment. This eliminated all edge segments that were less than four pixels long.

We defined goodness of fit by the root mean square difference, where the mean is taken over the support region defined by the horizontal extent of the segment. A segment for which this measure was less than a threshold was selected for further consideration. If the horizontal extent of the segment was more than 1/3 the horizontal length of the eye, the segment was selected as a possible eyelid edge; otherwise, it was put on a candidate list for possible combination with other segments.

A two-segment pairing was considered only if the two segments had the same sign of curvature and were non-overlapping along the horizontal axis. We then

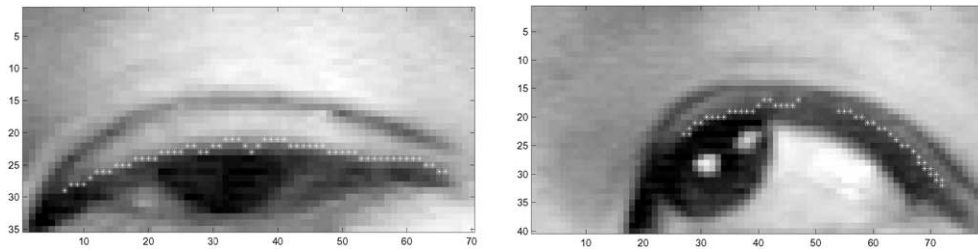


Fig. 4. Examples of eyelid edge detections.

looked for third segments that could form a triple with each selected pair. Here again, we considered only non-overlapping segments taken from the candidate list. We could similarly proceed to consider quadruples, quintuples, etc., but in our experiments we found that triples were sufficient.

The above procedure gave us three lists of possible candidates for the eyelid edge. The first one contained all selected single segments, the second contained all selected pairs, and the third contained all selected triples. We will now describe our method of selecting the best candidate from these lists.

We mentioned earlier that the eyelid edge should be the first edge segment that we encounter when we search upward from the iris center. Sometimes, however, this segment is a small edge segment and is not the best choice for the eyelid edge. To get the best choice, we examined all the edge segments whose distances above the iris center were close to the minimum and chose the one that had the greatest number of pixels. We also required that these segments must be curving downward and must not have large slopes. Fig. 4 shows two examples of eyelid edges that were detected using these rules.

#### 4. Frame-to-frame tracking

Detection of eye corners, iris edge segments and centers, and eyelid edge segments was performed independently for every frame of two 120-frame sequences in one of which the subject was wearing glasses. We describe frame-to-frame iris tracking in Section 4.1, and frame-to-frame eyelid tracking in Section 4.2. Experimental results on two image sequences are described in Section 4.3.

##### 4.1. Iris motion

Fig. 5 shows tracks of the eye corners and iris centers over a 120-frame sequence, overlaid on the first frame of the sequence. (This sequence had two instances of a blink. Since eye corners and iris centers cannot be found during an eye closure, we linearly interpolated between



Fig. 5. Eye corner and iris center motions in a 120-frame sequence.

the positions found just before the eyes closed and those found just after the eyes reopened.) The iris tracks in Fig. 5 clearly show that the irises have a compound motion. Their centers move as a result of both the head motion, shown by the tracks of the eye corners, and the independent iris motion, which can be determined by subtracting the eye corner motion from the compound iris motion. We see that there is a predominant upward motion of the eyes over the sequence, and that the eyes move to the right and then come back to the left.

Fig. 6 plots the independent motion of the irises in the subsequence beginning with the reopening of the eye after the first blink and ending with its closing in the second blink. This subsequence starts at frame 30 and lasts until frame 97. The plots show horizontal displacements of the left and right irises in their respective eye-centered coordinate systems. (The vertical axis indicates the frame number.) We see that at the start of the subsequence the irises move to the right (frames 30–50); they stay in this position (relative to the eye corners) from frame 50 to 85, and then move to the left in frames 85–97. We can qualitatively describe this motion as a rightward saccade motion in frames 30–50, a smooth pursuit in frames 50–85, and a leftward saccade in frames 85–90. At the

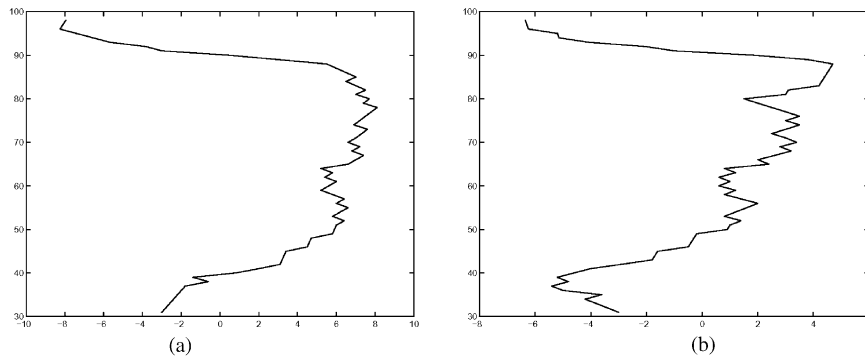


Fig. 6. Horizontal motions of the left (a) and right (b) irises in their respective eye-centered coordinate systems.

beginning of the subsequence the eyes move ahead of the head motion; the head then follows the eye motion; and finally the eyes move back to the other side.

#### 4.2. Eyelid motion

The eyelid deforms as it opens and closes. If we wanted to do a complete quantitative analysis of eyelid deformation, we would have to establish a one-to-one correspondence between the eyelid edge pixels in successive frames. Fortunately, we are interested in observing the eyelids only to aid us in detecting blinking; analysis of eyelid deformation is therefore not necessary. All we need to know is the vertical displacement of the eyelids over our sequence. Evidently, some eyelid edge pixels have larger vertical displacements than others, but for our purposes it is sufficient to study the vertical displacement of the point representing the apex of the eyelid. We used the polynomials fitted to the eyelid edges to find these apexes.

The eyelid heights, defined as the vertical distances between the left and right eyelid apexes and their respective iris centers, are plotted for an entire 120-frame sequence in Fig. 7. The spikes in the plots are due to mis-detections of the eyelid.

#### 4.3. Experimental results

Detailed experimental results on two 120-frame sequences, in one of which the subject was wearing glasses, can be found in Ref. [10] together with further details about the algorithms for iris center and eyelid edge detection. In the sequence without glasses, if we exclude the subsequences (frames 16–26 and 98–105) in which the eyes were closed, there was over 95% correct detection of the eye corners and iris centers. The eyelids were not always detected even when the eyes were open and their corners were detected, but they were correctly detected about 80% of the time. The results were essen-

tially the same in the sequence with glasses, if we exclude the subsequence (frames 43–55) in which the eyes were closed.

### 5. Flow-based tracking

As we have just seen, frame-to-frame eye part tracking was not always successful, especially in tracking the eyelids. In this section we describe a flow-based method of tracking the eye corners, irises, and eyelids using simple head, iris, and eyelid motion models. The iris and eyelid motions have multiple components, one of which is the head motion. We de-couple the head component of motion from the iris and eyelid components, and analyze the motions of the irises and eyelids based on the head-independent flow information.

We use an approach similar to that used in Ref. [4]; but in that paper, each eye was treated as a whole, and a statistical analysis of the motion vectors in the eye was used to determine blinking and other expressions. Evidently, we cannot rely on this approach to determine iris or eyelid behavior. Furthermore, we require very precise motion information to track the iris and eyelid between successive frames. We will derive this motion information from the optical flow associated with the edge segments of the irises and eyelids.

We use normal flow (e.g., Ref. [17]) to compute the apparent displacement vectors of edge pixels. We identify the edge segments associated with the silhouette of the head, which we regard as an essentially rigid part of the head, and we use the normal flow associated with these segments to drive an affine head motion model. Using the parameters obtained from this model for each frame, we “stabilize” the motions of the irises and eyelids by warping the next frame onto the current frame. Then we compute the normal flow for the edge segments representing the irises and eyelids based on the stabilized version of the next frame; this yields flow vectors that represent the independent motions of the irises and

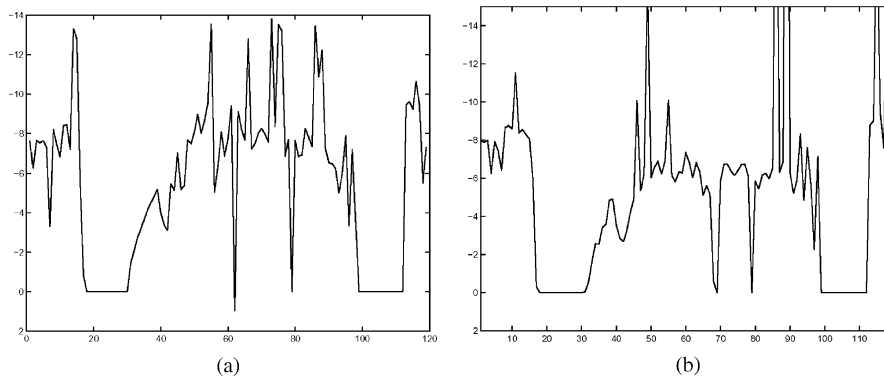


Fig. 7. Heights of the eyelid apices above their respective iris centers for the (a) left and (b) right eyes. The spikes in the plots indicate mis-detections.

eyelids. These flow vectors are then used to identify the iris and eyelid motions based on their models.

In Section 5.1 we describe our method of normal flow measurement. Section 5.2 describes how we identify regions of interest (the head outline and interior) and describes our head and eye motion models. In Section 5.3 we show how flow information is used in conjunction with the head motion model to estimate the head motion. In Section 5.4 we describe how to decouple the motions of the irises and eyelids from the head motion. Finally, in Sections 5.5 and 5.6 we describe how we track the irises and eyelids, and show results on our two image sequences.

### 5.1. Normal flow measurement

To measure normal flow we proceed as follows: For each edge element in a frame, say at  $\vec{r}$ , we resample the image locally to obtain a small window with its rows parallel to the image gradient direction  $\vec{n}_r = \nabla I / \|\nabla I\|$ . For the next frame we create a larger window, typically twice as large as the maximum expected value of the magnitude of the normal flow. We then slide the first (smaller) window across the second (larger) window in the gradient direction and compute the difference between the image intensities. Let the zero of the resulting function be at distance  $u_n$  from the origin of the second window; note that the gradient of the function at the positions close to  $u_n$  must be positive. Our estimate of the normal displacement field at  $\vec{r}$  is then  $-u_n$ , and we call it the *normal flow*.

The normal flow associated with the edge pixels found in one of the frames in Fig. 1 is displayed in Fig. 8 as vectors emanating from the edge pixels. The lengths of these vectors have been exaggerated by a factor of 10 for display purposes.

Fig. 9 zooms in on the left (a) and right (b) eye regions and displays the flow vectors in these regions. We see that these vectors show motions different from those shown



Fig. 8. Normal flow displayed as vectors emanating from the edges found in one of the frames shown in Fig. 1.

by the vectors on the head boundary. Specifically, the flow vectors on the iris edges indicate that the irises are moving upward and to the right, and the flow vectors on the edges of the upper eyelids indicate that these eyelids have an upward (opening) motion.

### 5.2. Regions of interest and motion models

In Ref. [1] we described a method of locating the face in an image based on the flesh-tone region, and of locating the eyes in the face image. We define an interior region of the face, bounded horizontally by the outer eye corners; the vertical extent of this region is determined by anthropometric averages. The head boundary flow is the flow associated with the edge segments that lie in the face region, but outside the interior region. This flow is regarded as arising from the head outline, which we regard as a “rigid” portion of the head. As we see in Fig. 8 the flow associated with the edge segments in the

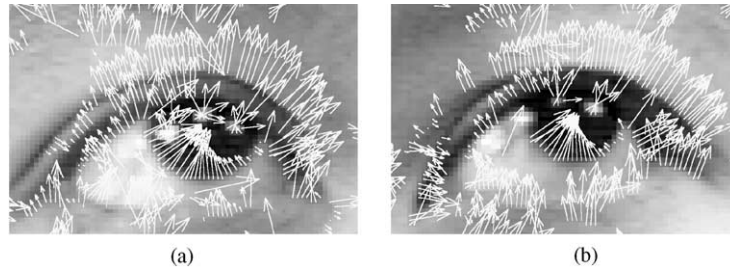


Fig. 9. Normal flow in the left (a) and right (b) eye regions in Fig. 8.

interior region arises from the facial features, including the eyes and mouth.

Since the restrictions on our subject limited the extent of her head motion, and since head motion from frame to frame is small, we used an affine motion model to represent the motion of the head

$$\begin{pmatrix} \delta x \\ \delta y \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} x(m) \\ y(m) \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix}. \quad (1)$$

Let  $\vec{r}(k)$  be an edge pixel location, and let  $\vec{n}_r(k)$ , be the gradient direction associated with  $\vec{r}(k)$ . Let  $n_f(k) = \langle \delta \vec{r}(k), \vec{n}_r(k) \rangle$  be the normal flow, where  $\langle \cdot, \cdot \rangle$  indicates inner product. Then

$$n_f(k) = \delta x(k)n_x(k) + \delta y(k)n_y(k). \quad (2)$$

Using Eqs. (1) and (2) we get

$$n_f(k) = An_x(k)x(k) + Bn_x(k)y(k) + Cn_y(k)x(k) + Dn_y(k)y(k) + T_x n_x(k) + T_y n_y(k). \quad (3)$$

Let

$$\mathcal{N}_f = \begin{pmatrix} n_f(1) & n_f(2) & \dots & n_f(m) \end{pmatrix}^t$$

$$\mathcal{A} = \begin{pmatrix} A & B & C & D & T_x & T_y \end{pmatrix}^t$$

$$\mathcal{X} = \begin{pmatrix} n_x(1)x(1) & n_x(2)x(2) & \dots & n_x(m)x(m) \\ n_x(1)y(1) & n_x(2)y(2) & \dots & n_x(m)y(m) \\ n_y(1)x(1) & n_y(2)x(2) & \dots & n_y(m)x(m) \\ n_y(1)y(1) & n_y(2)y(2) & \dots & n_y(m)y(m) \\ n_x(1) & n_x(2) & \dots & n_x(m) \\ n_y(1) & n_y(2) & \dots & n_y(m) \end{pmatrix}^t. \quad (4)$$

Then

$$\mathcal{N}_f = \mathcal{X}\mathcal{A}. \quad (5)$$

Eq. (5) is solved for  $\mathcal{A}$  using least-square approximation applied to an over-determined system of equations.

Since we are considering an essentially frontal face image, the motions of the iris and eyelid are very simple. We assume that the iris motion is approximately a

translation:

$$\begin{pmatrix} \delta x \\ \delta y \end{pmatrix} = \begin{pmatrix} T_{ix} \\ T_{iy} \end{pmatrix}. \quad (6)$$

Using the expression for the normal flow magnitude gives

$$n_f = T_{ix}n_x + T_{iy}n_y \quad (7)$$

which we can solve for the translation parameters to determine the motion of the iris. Similarly, we assume that the apex of the eyelid moves only vertically, so we can immediately solve for its vertical translational parameter  $T_{ey}$ .

### 5.3. Head motion

A majority of the edge segments outside the inner face region represent the head boundary, but some segments are erroneous. We will now describe a method of selecting edge segments that more accurately represent the head boundary, so that we can more accurately determine the parameters of the head motion model.

We use flesh-tone color segmentation (see Ref. [1]) to obtain a connected component (“face blob”) representing the face region. This blob often has interior holes associated with the eyes, nostrils, and lips, or with high-reflectance areas. We perform morphological operations on the face blob to get a five-pixel thick boundary of the face region. We then select edge segments that lie inside this boundary zone. The boundary zone corresponding to the face in Fig. 8, and the flow vectors associated with the edge segments that lie in this zone, are shown in Fig. 10.

The normal flow associated with these segments is used to solve Eq. (5). At frame  $m$  the computed rotation and translation parameters indicate the head motion between frame  $m$  and frame  $m + 1$ . These instantaneous motion parameters could be used to track the head over our sequence, but head tracking is not our primary goal. We use the motion parameters only to determine the frame-to-frame head motion, which is then subtracted from the motions of the irises and the eyelids.

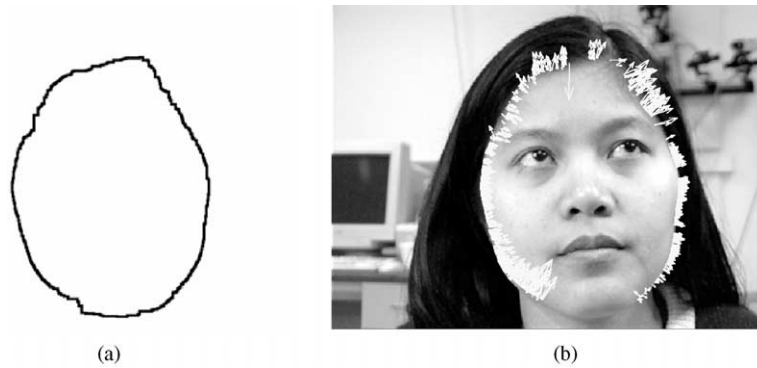


Fig. 10. (a) Head boundary after applying morphological operations to the face blob. (b) Flow vectors of Fig. 8 that emanate from the head boundary.

Once the head motion components of the iris and eyelid motions are removed, we are left with their independent motions.

#### 5.4. De-coupling motion components by stabilization

Fig. 9 showed an example of the normal flow in the two eye regions. These flow vectors represent the composite motions of the irises and the eyelids. Two methods can be used to remove the head motion from these composite motions. The first method is simple subtraction of the head motion component (computed, at each pixel position, from the motion model parameters) from every flow vector in the eye regions. The second method involves image stabilization based on the head motion parameters. We used the second method because the irises have the potential of moving much faster than the head. Since the normal flow computation assumes a maximum threshold on the flow vector magnitudes, it might be possible to miss rapid motions of the irises in situations where the head and irises are moving in the same direction. Therefore, to maximize flow detection on the irises, we stabilize the image sequence based on the head motion estimates—i.e., we warp each frame to zero out the head motion estimated in the preceding frame. The flow computed for the eyelid and iris segments between a given frame and the warped version of the next frame then indicates their independent motions.

Let  $I_m(x, y)$  and  $I_{m+1}(x, y)$  be the image intensities in the frames at times  $m$  and  $m+1$ . The stabilized version of  $I_{m+1}(\hat{x}, \hat{y})$  is determined by warping its intensity values as follows:

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} A+1 & B \\ C & D+1 \end{pmatrix}^{-1} \left[ \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} T_x \\ T_y \end{pmatrix} \right]. \quad (8)$$

Eq. (8) is a backward warping of frame  $m+1$  on to frame  $m$ . We now recompute the normal flow between  $I_m(x, y)$  and  $I_{m+1}(\hat{x}, \hat{y})$ . This stabilized flow, associated with the

iris and eyelid segments, approximates their independent motions.

Fig. 11 shows an example of the result of the stabilization process. The head motion (predominantly upward) in this example is significant; the stabilized and unstabilized flow vectors are noticeably different.

#### 5.5. Iris motion

We are interested in two kinds of analyses relating to the irises' motions. The first involves global tracking of the irises to determine changes in gaze direction. The other is analysis of the head-independent motions of the irises.

We use the flow vectors associated with the iris segments to solve Eq. (7), for the left and right irises, to give us  $T_{ix}$  and  $T_{iy}$  pairs in every frame. Let  $\delta\hat{x}_l(m) = T_{ix}$  and  $\delta\hat{y}_l(m) = T_{iy}$  for the left iris in frame  $m$ . The right iris similarly gives  $\delta\hat{x}_r(m)$  and  $\delta\hat{y}_r(m)$ . When we solve Eq. (7) for every frame we get the instantaneous translations of the centers of the irises in the  $x$  and  $y$  directions. In this computation, we can use either the unstabilized or stabilized flow vectors. If we use the former we obtain the global (i.e., total) motions of the irises; if we use the latter we obtain their independent motions.

If we compare the unstabilized and stabilized flow vectors (Fig. 11), we indeed see that the former tend to extend further in the upward direction than the latter. The crossover points are also somewhat further to the left in the unstabilized frames than in the corresponding stabilized frames; this indicates that the head is also moving slightly to the left.

Plots of the iris motion in frames 70–96 are shown in Fig. 12. To compute these plots, we computed  $\delta\hat{x}_l(m)$ ,  $\delta\hat{x}_r(m)$ ,  $\delta\hat{y}_l(m)$  and  $\delta\hat{y}_r(m)$  for each frame  $m$  in the sequence using Eq. (7). Let  $\delta\vec{r}_l(m) = \vec{i}\delta\hat{x}_l(m) + \vec{j}\delta\hat{y}_l(m)$  be the instantaneous (stabilized or unstabilized) motion vector of the left iris in frame  $m$ , and let  $\delta\vec{r}_r(m)$

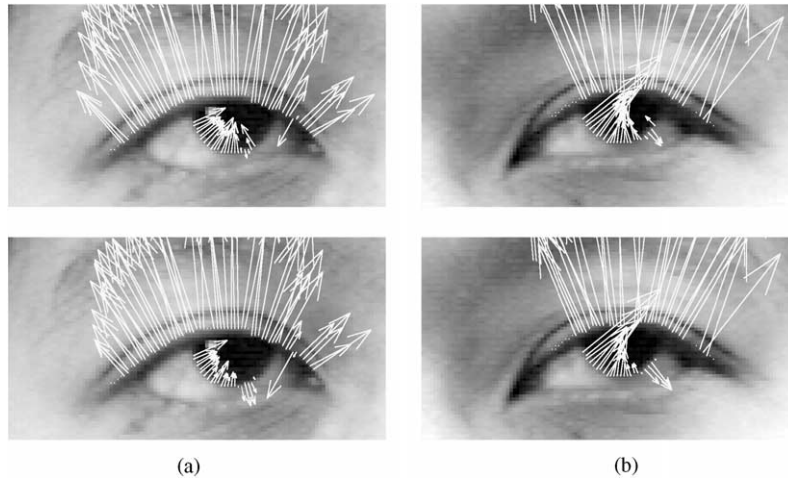


Fig. 11. Top row: normal flow in (a) left and (b) right eye. Bottom row: normal flow after stabilization.

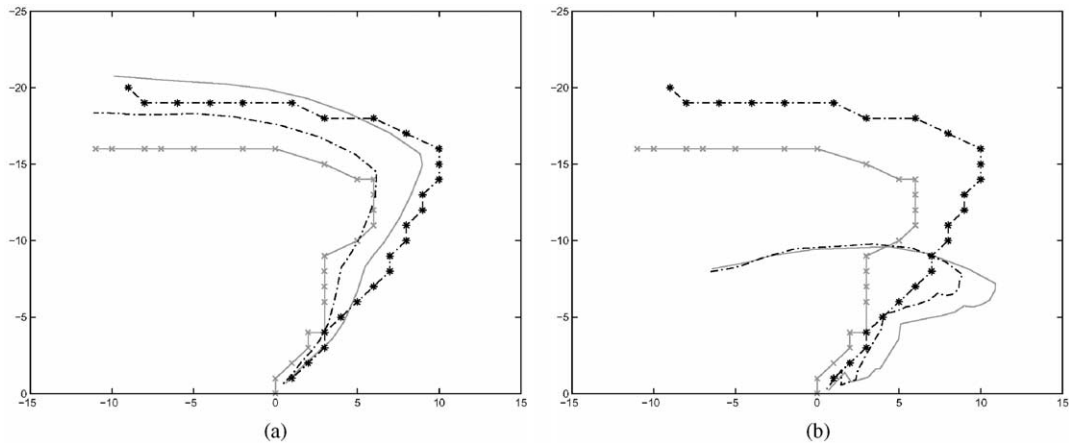


Fig. 12. Tracks of iris motion. The solid line and dot-dash lines are the left and right iris motions from the flow computations. The dash-x and dot-dash-star lines are the displacements of the left and right irises obtained from the frame-to-frame tracking of the iris center positions. (a) shows the tracks for the unstabilized flow and (b) for the stabilized flow.

be defined similarly for the right iris. The tracks of the irises in the sequence were obtained by concatenating the individual vectors in a head-to-tail fashion. These tracks are plotted in Fig. 12(a) and (b) for the unstabilized and stabilized motions, respectively. The solid line in each figure is the track of the left iris and the dot-dash line is that of the right iris. As Fig. 12 shows, the irises are moving to the left and upward. As we see by comparing Fig. 12(a) to (b), almost none of the leftward component of this motion is due to head movement; but a significant fraction of its upward component is due to head movement. We see that the irises move upward and rightward; after they reach their rightmost points they move back leftward, but still slightly upward. For

comparison, the frame-to-frame tracks of the iris center position are shown on the same plots, as dash-x (left) and dot-dash-star (right) curves. The tracks of the unstabilized motion in Fig. 12(a) closely resemble the frame-to-frame tracks, but the tracks of the stabilized motion in Fig. 12(b) are markedly different. This shows the effective de-coupling of the head motion from the iris motion using stabilization.

### 5.6. Eyelid motion

The method of eyelid detection in a single frame described in Section 3.2 was not always reliable. In this section we show that the eyelid can be detected more

reliably using flow-based tracking. Here too we assume that the iris is at least partially visible.

To track the eyelid using flow we proceed as follows: In the first frame of the sequence we locate the eyelid using the method described in Section 3.2. The edge pixels of the eyelid that have flow vectors associated with them are used to find edge pixels in the next frame. This is done by examining a  $2 \times 2$  window in the next frame centered at the position indicated by the flow vector at each pixel in the current frame. All edge pixels found in this way in the next frame are labeled as possibly belonging to eyelid segments. We fit a third-order polynomial to these edge pixels. All the edge pixels that lie in the neighborhood of this fitted polynomial are labeled as the eyelid segment of the next frame. Tracking is achieved by repeatedly using these flow-identified eyelid segments to find eyelid segments in successive frames.

When the iris is visible and we have located its center, opening and closing of the eyes can be completely described by the position of the eyelid apex relative to the iris center. Eye closing is indicated by a decrease in this distance, and eye opening by an increase. Even when the center of the iris is not visible, as long as some of the iris is visible, eye opening is indicated by an increase in the amount of visible iris, and eye closing is indicated by a decrease in this amount. This allows us to handle cases of eye closing which occur when the iris rolls up into the upper eyelid, the upper eyelid moves down over the iris, or both. Note that in this situation the iris center is at a negative distance below the eyelid apex. Eye opening is then indicated by a decrease in the absolute value of this negative distance, and eye closing by an increase. An important consequence of using these relative measures is that we do not have to take head motion into account, since it has no effect on the relative positions of the eyelid apex and the iris center.

Figs. 13(a) and (b) show plots of the vertical distance between the eyelid apex and the iris center for an entire 120-frame sequence. (Results of both iris tracking and eyelid tracking for another sequence, in which the subject was wearing glasses, are shown in Ref. [10].) The eyelid edges are found using the tracking method described above. The eyes are closed in two subsequences (approximately frames 20–30 and 100–110); for these frames no distance is plotted, but the plots show negative distance in two cases where the iris was detected just before the eyelid closed, and its (invisible) center was above the eyelid. In the right eye plot (Fig. 13(b)) the distances are incorrect between frames 60 and 70, because track was lost; but the correct distance was found again after frame 70 when the tracking was re-initialized.

When the iris is entirely invisible, there are two possible situations: (1) the eyelids are entirely closed; (2) the eyelids are partly open, but the iris is rolled up or down and is entirely hidden by an eyelid. In the first case, the edges of the eyelids will not be visible, making tracking impossible. In the second case, the eyelid can be tracked, but its motion upward or downward may be due to head movement rather than to closing or opening of the eye. Thus in this case, eye closing or opening is indicated by the stabilized eyelid motion being downward or upward. In our image sequences, cases in which the iris is entirely invisible but the eyelids are not closed (and the head is moving) did not occur, so we cannot show an example of eyelid tracking when the irises are invisible.

Detailed results of the experiments on the two 120-frame sequences can be found in Ref. [10]. The irises were successfully tracked in every frame in which the eyes were even partially open, and the eyelids were successfully located and tracked 90% of the time.

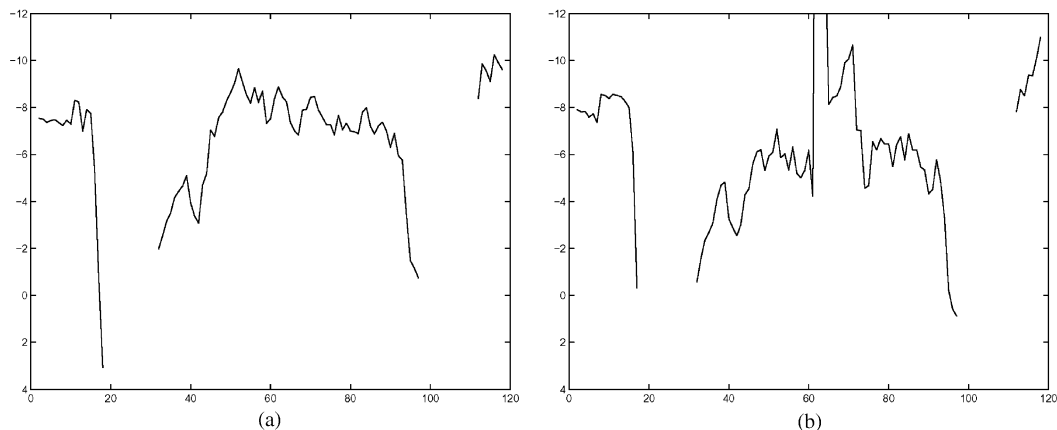


Fig. 13. Plots of the vertical distance between the eyelid apex, found by tracking using flow, and the iris center. (a) Left eye, (b) right eye.

## 6. Conclusions

We implemented methods of tracking the motion of the head and the independent motions of the irises and upper eyelids in an image sequence. We made use of the detected eye corners [1] to define eye regions for further analysis. Our methods required no manual initialization and did not depend on assumptions about the relative positions of the irises and eyelids, unlike the methods reviewed in Section 2.

We used a vote accumulation method to locate the edge of the iris. Since the upper part of the iris may not be visible, we accumulated votes by summing edge pixels in a U-shaped annular region whose radius approximates the radius of the iris disk. The annulus center that received the most votes was selected as the iris center.

To detect the edge of the upper eyelid, we examined all edge segments in the eye region. We fitted a third-degree polynomial to each segment, as well as to pairs and triples of segments, and applied the following criteria: An eyelid segment must have edge pixels above the iris center; the polynomial fitted to it must not have a large slope, and must be concave downward, above the iris center; it should ideally be the segment closest to the iris center, but if several segments are almost equally close above the iris center, we chose the one with the largest number of pixels.

We fitted an affine head motion model to the eye corners in successive frames. To determine the independent motions of the irises, we used the head motion model to correct the frame-to-frame motions of the iris centers. The unstabilized iris center motion can be used to determine changes in gaze direction, including saccades and smooth pursuits. Changes in the vertical distance between the upper eyelid apex and the iris center can be used to detect closing of the eyes.

In experiments on two 120-frame sequences, the eye corners were detected in about 95% of the frames in which the eyes were not closed. The iris centers were correctly detected in every case where the eye corners were detected, and the upper eyelids were correctly detected in about 80% of these cases. These results were obtained whether or not the subject was wearing glasses.

We also tracked the head and eye parts by using normal flow to compute the apparent displacement vectors of edge pixels. We used the normal flow associated with edge segments of the head silhouette to drive an affine motion model. The parameters obtained from this model were used to stabilize the motions of the irises and eyelids by warping each frame onto the preceding frame before computing the flow. The stabilized flow vectors thus obtained were used to identify the independent iris and eyelid motions. By using this flow-based tracking approach we were able to improve eyelid detection perfor-

mance to about 90% and to obtain better estimates of iris motion.

## References

- [1] S. Sirohey, A. Rosenfeld, Eye detection in a face image using linear and nonlinear filters, *Pattern Recognition* 34 (2001) 1367–1391.
- [2] C. Morimoto, D. Koons, A. Amir, M. Flickner, Pupil detection and tracking using multiple light sources, Technical Report, IBM Almaden Research Center, 1998.
- [3] R.P. Wildes, J.C. Asmuth, G.L. Green, S.C. Hsu, R.J. Kolczynsky, J.R. Matey, S.E. McBride, A machine-vision system for iris recognition, *Mach. Vision Appl.* 9 (1996) 1–8.
- [4] M. Black, Y. Yacoob, Recognizing facial expressions in image sequences using local parameterized models of image motion, *Int. J. Comput. Vision* 25 (1997) 23–48.
- [5] D. DeCarlo, D. Metaxas, The integration of optical flow and deformable models with applications to human face shape and motion estimation, *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 1996, pp. 231–238.
- [6] D. DeCarlo, D. Metaxas, Combining information using hard constraints, *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 1999, pp. 132–138.
- [7] T. Horprasert, Y. Yacoob, L.S. Davis, Computing 3D head orientation from a monocular image sequence, *Proceedings, Second International Conference on Automatic Face and Gesture Recognition*, 1996, pp. 242–247.
- [8] H. Tao, T.S. Huang, Explanation-based facial motion tracking using a piecewise Bézier volume deformation model, *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 1999, pp. 592–597.
- [9] Y. Yacoob, L.S. Davis, Recognizing human facial expressions from long image sequences using optical flow, *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (1996) 636–642.
- [10] S. Sirohey, A. Rosenfeld, Z. Duric, Eye tracking, Technical Report CAR-TR-922, Center for Automation Research, University of Maryland, College Park, 1999.
- [11] J.Y. Deng, F. Lai, Region-based template deformation and masking for eye-feature extraction and description, *Pattern Recognition* 30 (1997) 403–419.
- [12] A. Yuille, D. Cohen, P. Hallinan, Feature extraction from faces using deformable templates, *Proceedings, IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1989, pp. 104–109.
- [13] J.P. Iwins, J. Porrill, A deformable model of the human iris for measuring small three-dimensional eye movements, *Mach. Vision Appl.* 11 (1998) 42–51.
- [14] K.-M. Lam, H. Yan, An improved method for locating and extracting the eye in human face images, *Proceedings, International Conference on Pattern Recognition*, 1996, pp. 411–415.
- [15] X. Xie, R. Sudhakar, H. Zhuang, Real-time eye feature tracking from a video image sequence using Kalman filter, *IEEE Trans. Systems Man Cybernet.* 25 (1995) 1568–1577.

- [16] X. Xie, R. Sudhakar, H. Zhuang, A cascade scheme for eye tracking and head movement compensation, *IEEE Trans. Systems Man Cybernet. A* 28 (1998) 487–490.
- [17] Z. Duric, A. Rosenfeld, Image sequence stabilization in real time, *Real-Time Imaging* 31 (1996) 271–284.

**About the Author**—SAAD A. SIROHEY received his B.Sc. with highest honors in Electrical Engineering from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in 1990, and his MS and Ph.D. degrees from the University of Maryland, College Park, also in electrical engineering. He is currently working on medical imaging at General Electric Medical Systems as an Imaging Architect. His research interests are signal/image processing, compression and image understanding. He has done research on automated face detection/recognition and tracking, with emphasis on eye gaze tracking. He is currently working on bio-medical imaging.

**About the Author**—AZRIEL ROSENFELD is a tenured Research Professor, a Distinguished University Professor, and Director of the Center for Automation Research at the University of Maryland in College Park. He also holds affiliate professorships in the Departments of Computer Science, Electrical Engineering, and Psychology. He holds a Ph.D. in mathematics from Columbia University (1957), rabbinic ordination (1952) and a Doctor of Hebrew Literature degree (1955) from Yeshiva University, and honorary Doctor of Technology degrees from Linköping University, Sweden (1980) and Oulu University, Finland (1994) and an honorary Doctor of Humane Letters degree from Yeshiva University (2000).

Dr. Rosenfeld is widely regarded as the leading researcher in the world in the field of computer image analysis. Over a period of 35 years he has made many fundamental and pioneering contributions to nearly every area of that field. He wrote the first textbook in the field (1969); was founding editor of its first journal (1972); and was co-chairman of its first international conference (1987). He has published over 30 books and over 600 book chapters and journal articles, and has directed over 50 Ph.D. dissertations. In 1985 he served as chairman of a panel appointed by the National Research Council to brief the President's Science Advisor on the subject of computer vision; he has also served (1985–1988) as a member of the Vision Committee of the National Research Council. In honor of his 65th birthday, a book entitled "Advances in Image Understanding—A Festschrift for Azriel Rosenfeld", edited by Kevin Bowyer and Narendra Ahuja, was published by IEEE Computer Society Press in 1996.

He is a Fellow of the Institute of Electrical and Electronics Engineers (1971), won its Emanuel Piore Award in 1985, and received its Third Millennium Medal in 2000; he is a founding Fellow of the American Association for Artificial Intelligence (1990) and of the Association for Computing Machinery (1993); he is a Fellow of the Washington Academy of Sciences (1988), and won its Mathematics and Computer Science Award in 1988; he was a founding Director of the Machine Vision Association of the Society of Manufacturing Engineers (1985–1988), won its President's Award in 1987 and is a certified Manufacturing Engineer (1988); he was a founding member of the IEEE Computer Society's Technical Committee on Pattern Analysis and Machine Intelligence (1965), served as its Chairman (1985–1987), and received the Society's Meritorious Service Award in 1986, its Harry Goode Memorial Award in 1995, and became a Golden Core member of the Society in 1996; he received the IEEE Systems, Man, and Cybernetics Society's Norbert Wiener Award in 1995; he received an IEEE Standards Medallion in 1990, and the Electronic Imaging International Imager of the Year Award in 1991; he was a founding member of the Governing Board of the International Association for Pattern Recognition (1978–1985), served as its President (1980–1982), won its first K.S. Fu Award in 1988, and became one of its founding Fellows in 1994; he received the Information Science Award from the Association for Intelligent Machinery in 1998; he was a Foreign Member of the Academy of Science of the German Democratic Republic (1988–1992), and is a Corresponding Member of the National Academy of Engineering of Mexico (1982).

**About the Author**—ZORAN DURIC received a Ph.D. in Computer Science from the University of Maryland at College Park in 1995. From 1995 to 1997 he was an Assistant Research Scientist at the Machine Learning and Inference Laboratory at George Mason University and at the Center for Automation Research at the University of Maryland. From 1996 to 1997 he was also a Visiting Assistant Professor at the Computer Science Department of George Mason University. He joined the faculty of George Mason University in the Fall of 1997 as an Assistant Professor of Computer Science.