Preliminary Investigation of Real-time Monitoring of a Driver in City Traffic

Margrit Betke [*]
Computer Science Department
Boston University
111 Cummington St
Boston, MA 02215

William J. Mullally
Computer Science Department
Boston College
Fulton Hall
Chestnut Hill, MA 02467

Abstract

We present a system that visually monitors the face of a driver and finds the location of the driver's eyes in real time. Eye detection and tracking has mostly been implemented for laboratory or simulator settings and often with special infrared lighting. Our system instead addresses the actual environment within the interior of a moving car and uses a consumer CCD camera mounted on the dashboard in front of the driver. We developed an eye detection method that is based on segmenting the head space from the image and then using edge detection, template matching, for which eye templates are created online, and brightness filters to obtain the location of the eyes within the head space. We tested our real-time system in thousands of frames using various subjects, head and facial motions, and lighting conditions and report our preliminary results.

1 Introduction

There is an increasing interest in being able to locate the position of a driver's eyes while he or she is driving. Reasons for this vary, from obtaining statistical data on where the driver looks while driving to looking for signs of driver fatigue in the hopes of creating an early warning system. Other experiments have been performed with this in mind, but most are done under laboratory conditions without taking into account the adverse environment within the interior of a moving car. Any realistic system must take into account the constant shaking of the camera and a constantly moving background pertaining to the view through the side and rear windows of the car. The system must also take into account constantly changing lighting conditions caused by the reflective properties of the car's windows as the car moves past other light sources. Our research addresses these adverse, but realistic conditions in city driving. Our system performs location, tracking, and analysis tasks in real time using various subjects, head and facial motions, and lighting conditions. Our system provides a tool that can be used to determine how much of the driver's attention is directed towards the city environment in the front, back, and on the sides of the car, and how much attention is directed towards the vehicle's instruments and passengers.

1.1 Previous Work

Various techniques have been used to detect and track people, and their faces and eyes in real time. Facial motions have been analyzed in real time or near real time using parametric flow models [18], models of facial dynamics [5], Hidden Markov Models [12], and stereo systems [9]. Skin color based detection of faces has been explored extensively, e.g., [7, 17]. Gaze estimation in ambient light has been addressed with neural networks [1, 16], morphable models [13], and self-organizing gray-scale units [2]. Gaze estimation in infrared light has been explored extensively, for example, to analyze driver behavior [14]. Video analysis to detect driver fatigue is an important application [6, 11]. A driver's lane-keeping behavior has been modeled and analyzed using simulator and real-world driving experiments [8, 4, 15].

2 Statistical Face Color Model

We use statistical decision theory [10], in particular, Bayes decision rule to estimate from the color of a pixel if it images a face. A training data set of images that are known to contain faces is analyzed to determine a priori probability distributions of skin color.

[*] Email: betke@cs.bu.edu, URL: http://www.cs.bu.edu/fac/betke, Phone: +1-617-353-6412, Fax: +1-617-353-6457. The author acknowledges support by NSF equipment grant 9871219.

Our system minimizes the average loss associated with the classification decision as follows. Let p and q be the respective a priori probabilities that a data vector $\mathbf{v}$ describes/does not describe the color of a particular face region. Let $p(\mathbf{v}|s)$ be the likelihood function for the image data $\mathbf{v}$ given a desired color s. Finally, let $p(\mathbf{v}|0)$ be the probability of the data, given that the desired color s is not present. The *likelihood ratio* $\ell(\mathbf{v})$ is then given by

$$\ell(\mathbf{v}) = \frac{p\,p(\mathbf{v}|s)}{q\,p(\mathbf{v}|0)}. \tag{1}$$

The likelihood ratio $\ell(\mathbf{v})$ is compared to the decision threshold

$$\mathcal{H} = \frac{C_1}{C_2}, \tag{2}$$

where C_1 and C_2 are the respective costs associated with false positive and false negative decisions. The average loss associated with the classification decision is minimized when pixels for which

$$\ell(\mathbf{v}) \geq \mathcal{H} \tag{3}$$

are classified belonging to the desired color, and pixels for which $\ell(\mathbf{v}) < \mathcal{H}$ are classified as not belonging to the desired color.

We segmented the skin and non-skin pixels in our training data by hand. Non-skin pixels include pixels such as hair, iris, pupil, teeth, nostrils, etc. If we assume that skin or non-skin colors are equally likely to occur within the face region, then the prior probabilities p and q can be set to 1/2. Assuming a Gaussian approximation of the likelihood function, the likelihood ratio is

$$\ell(\mathbf{v}) = \frac{\prod_i p(v_i|skin)}{\prod_i p(v_i|non\text{--}skin)} \tag{4}$$

for $v_1 = \text{red}, v_2 = \text{green}$, and $v_3 = \text{blue}$. More details on our statistical skin model can be found in Ref. [3].

3 System Overview

Our system processes image frames that are continuously fed from a video camera to the computer. It outputs an online description of location and size of the person's face and eyes. The vision system contains two main components, the face detector and the eye detector. Every frame past the third is compared to previous frames to determine motion values for pixels within skin-tone range. An estimate of the face location is made from these "skin-tone motion" images. The system then detects the eyes within the estimated face location. With the exception of the motion detection, each frame is processed individually.

4 Face Detection

The task of the face detector is to segment a person's head from the background. It provides a boundary within which the eye detector searches for the eyes. The face detector outputs estimates for the center and the top, bottom, left, and right borders of the face.

The face detector identifies pixels with skin tone in the entire image frame. To classify the color of a pixel, Bayes decision rule is applied as described in Section 2. The face detector then creates a "skin-tone motion image" that describes where the significant changes in skin color from one frame to the next occur. As can be seen in Figure 1, these changes appear strong within the face and at its border, where skin-tone motion is obtained by subtracting face pixels from non-face background pixels. The face detector can therefore identify the face outline by searching the skin-tone motion image for strong edges.

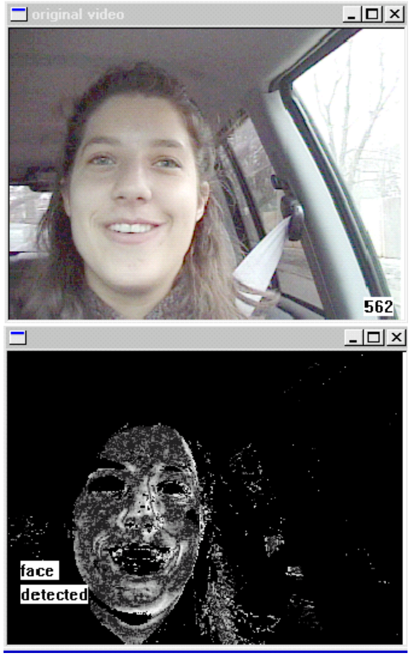

Figure 1: An original image and its skin-tone motion. Bright pixels indicate significant temporal changes in skin tone, black pixels indicate no change.

Strong horizontal and vertical lines in the $n \times m$ skin-tone motion image are identified as follows. Its pixel values are projected horizontally and vertically onto vectors $\mathbf{h}$ and $\mathbf{v}$, respectively. The horizontal projection vector $\mathbf{h}$ is an m-dimensional column vector, and the vertical projection vector $\mathbf{v}$ is an n-dimensional row vector. A large vector component h_i

indicates that the ith image row contains a substantial number of pixels with skin-tone motion that are due to up or down head motion. Similarly, a large component v_j indicates that the jth image column contains a substantial number of skin-tone motion pixels that are dues to left or right head motion. Local peaks in $\mathbf{h}$ and $\mathbf{v}$ can therefore be used to estimate the top-, bottom-, left-, and rightmost coordinates of the face. The search for these coordinates starts at the top, bottom, left, and right border of the image and moves towards the image center.

Our method to detect the face outline only assumes that the face will be a "recognizable blob" in the skin-tone motion image. It does not assume the coherence of same-color pixels and therefore does not waste computational resources trying to find crisp, contiguous edges that mark the border of the face.

The computed coordinates of the face outline are then used as search boundaries for eye detection. The set of coordinates can be used to define an ellipsoidal model on the face, as described in Ref. [3]. However, in practice, the ellipsoidal model does not appear to increase accuracy over a rectangular region defined by the same coordinates. To keep the computational costs low, our system therefore does not use an ellipsoidal face model, but uses a rectangular "skin-tone motion blob" as the face outline.

5 Eye Detection

The task of the eye detector is to find the eyes in a face by first locating the left and then the right eye. The eye detector is started as soon as the face region is identified.

Several observations about eyes help us locate the eyes within the reduced scope of the face. The area around the eyes has numerous edges and detecting these edges narrows the search field. The eyes also correspond to a darker region of the face. By finding the pixels in the face that correspond both to strong edges and to dark regions, we significantly reduce the area of the face that could be the eye. Taking out all pixels that do not correspond to both of these criteria leaves an image with pixels corresponding to face features such as lips, eyes, and eye brows, and shadows that the nose and chin cast and possibly hair that falls at the edge of the face boundary.

The eye detector first identifies all pixels in the face region with strong horizontal edges. To compute these edges, finite-difference approximations of the derivatives in the y-direction of the three color bands red, green, and blue are thresholded. If all three bands have a finite difference of at least 20 (out of the 256 possible), the pixel is considered to belong to an edge. The eye detector then determines the dark regions within the face by thresholding the color components of the image. In particular, if any two of the three color bands red, green, or blue have brightness values below 65 (out of the 256 possible levels), the pixel is considered to be dark.

The eye detector creates a "composite image" that classifies the face pixels into three classes that represent the likelihood that a pixel belongs to an eye. The highest concentration of likely eye pixels are expected to occur around the eyes, where large clusters of dark pixels, and horizontal edges due to eye brows, eyelashes and iris borders are the most prominent features. An example of a composite image is shown in Fig.2.

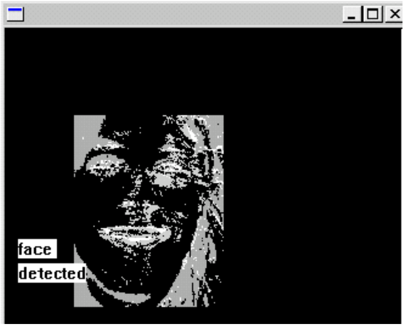

Figure 2: Composite image: Most likely eye pixels are shown in white. Pixels that are either part of a strong horizontal edge, or belong to a dark region are shown in gray. Pixels that do not fulfill either criterion are shown in black.

The correlation coefficient r is computed for each likely eye pixel in the composite image. The correlation coefficient r measures how accurate a subimage I_T containing the red component of the original color image frame can be approximated by the template T shown on the left in Fig. 3. The correlation coefficient r is defined as

$$\begin{aligned} r = \quad & (p_T \textstyle\sum_{x,y} I_T(x,y)T(x,y) - \Big(\sum_{x,y} I_T(x,y)\Big) \cdot \\ & \Big(\textstyle\sum_{x,y} T(x,y)\Big))/\sigma_{I_T}\sigma_T \end{aligned}$$

where
$\sigma_{I_T} = (p_T \sum_{x,y} I_T(x,y)^2 - \Big(\sum_{x,y} I_T(x,y)\Big)^2)^{1/2}$, $\sigma_T = (p_T \sum_{x,y} T(x,y)^2 - \Big(\sum_{x,y} T(x,y)\Big)^2)^{1/2}$ and p_T

is the number of pixels in the template image T with nonzero brightness values. Image I_T and template T are perfectly correlated if $r = 1$. Figure 3 shows a "correlation image" that illustrates the pixels with a correlation coefficient of at least 0.4 in white and the other pixels in black.

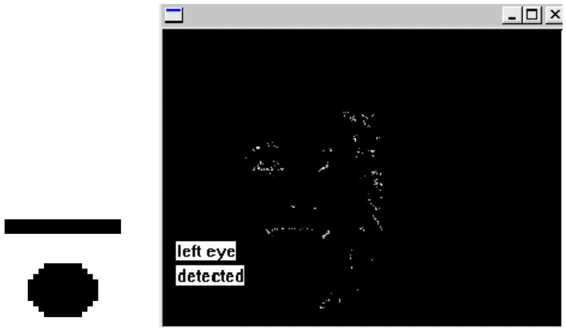

Figure 3: Correlation image: pixels with a correlation coefficient of at least 0.4 are white, other pixels are black.

The eye detector then searches the upper left half of the correlation image for a peak in white pixel concentrations. The largest concentration is generally located at the left eye and therefore can be used to estimate the position of the left eye. Note the numerous high coefficient values in the driver's hair on the right side of the correlation image in Fig. 3. Because of stronger light on this side of the drivers face, the peak in upper right corner of the face in the correlation image actually corresponds to the edge of the face, not the eye. The eye detector therefore searches the upper right half of the *composite* image for a peak in white pixel concentrations to find the right eye.

Once the left eye has been detected and tracked without large positional changes in three consecutive frames, a 20×25 eye template image is cropped from the current frame. As long as this template correlates with subimages in subsequent frames with a coefficient of at least 0.4, the left eye is tracked locally. If for some subsequent frame the template does not correlate well, eye detection in the composite image is resumed until the local tracking conditions are again satisfied.

6 Hardware

Our system uses a Sony DCR-TRV510 color video CCD camera. The camera has autofocus and adjusts to lighting changes. Its NTSC video output is send to a Matrox Meteor II image capture board on a 450 MHz dual processor PC with 384MB RAM. Our system processes images of size 320×240 pixels, which is half the resolution that the camera provides.

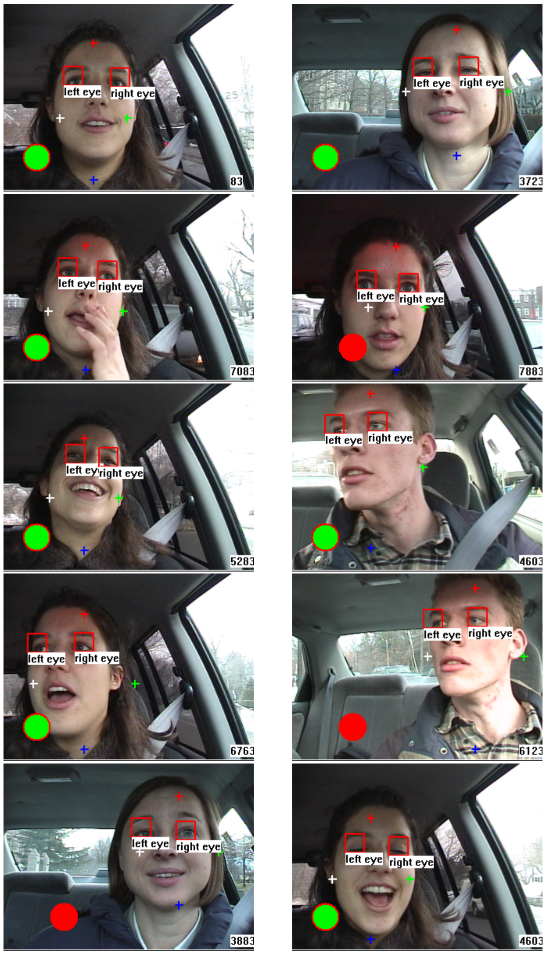

Figure 4: Eye detection.

7 Experiments and Discussion

We recorded people while they were driving around Boston. The drivers were asked to ignore the camera and drive as they normally would in order to create as close to a real driving situation as possible. We then tested our system using the data obtained in this fashion.

Our system processes about 11 frames per second. To give a quantitative analysis of our system's performance, we added the option to save processed images that are annotated with information about face borders, eye locations, and movement directions. Saving images for later analysis slows down the system to about 6.2 frames per second. We chose a uniform sampling rate and recorded every 21th frame, which resulted in 614 stored images over a period of about

12,894 frames or 35 minutes of video. We observed comparable results when we tested the system on an additional 20 minutes of video without taking the option to save images. Of the 614 frames, 521 frames contained relevant results. Our system correctly identifies one eye in 83% of the stored images and correctly identified both eyes in 56% of the stored images. The left eye is located correctly 76% of the time and the right is correctly located 68% of the time. Because the detection of the right eye relies on the detection of the left eye, the lower percentage of right eyes detected is expected. Since the sensitivity of our system is computed over a random subset of our data, it also includes our results for difficult lighting situations. In such situations, the images display blooming effects, strong shadows across the driver's face, and scenes in a dark parking garage, as shown in Fig. 5. Test runs with few drastic lighting changes show much better performance, e.g. in 95% of the frames at least one eye is identified correctly.

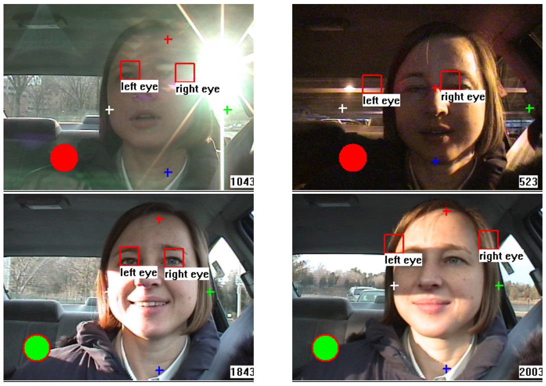

Figure 5: Results for difficult lighting situations.

Figure 4 illustrates successful eye detection. Figures 6 and 7 show cases where only one eye is detected or eye detection failed. Mismatches are due to self-occlusion, closed eyes, and misidentification of the face outline. Figure 8 reports the driver's up-and-down eye motion over time. This motion is due to the true head movements that we observed, as well as the camera's shaking motion during the drive.

A simpler version of our system that manually determines the field of view and automatically detects and tracks the eyes has been tested extensively indoors in several public demonstrations that included about 100 test subjects. *All* subject eyes were detected and tracked successfully.

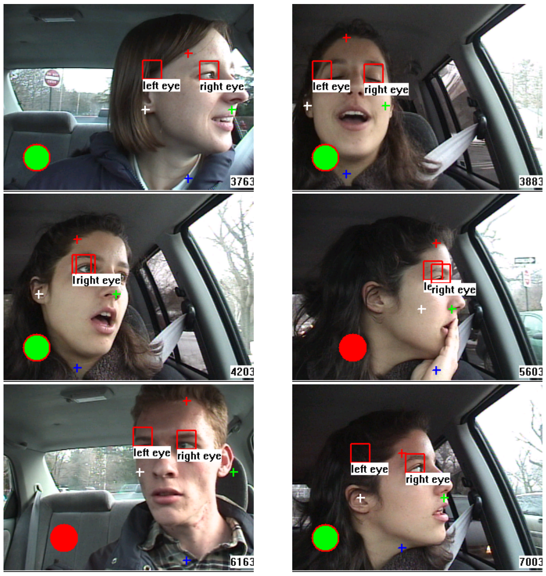

Figure 6: Localization of one eye.

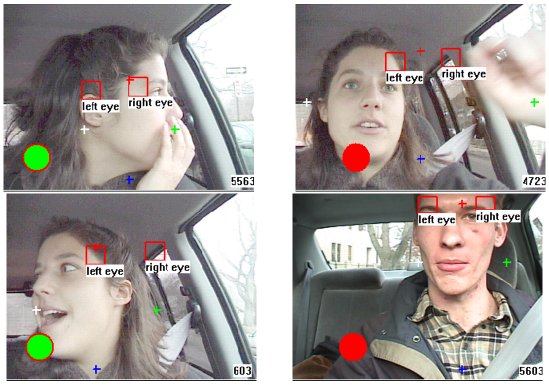

Figure 7: Incorrect localization of eyes due to occlusion, fast motion, and incorrect face localization.

8 Future Work

Our testing did not take into account persons with large amounts of facial hair or persons wearing glasses. We plan to do so in the future. To make eye detection reliable over long time periods, we will also add a tracking filter that exploits the tracking history. We would also like our system to be able to realize when only one eye is visible as in the cases of the driver turned 90 degrees to the right or left.

Currently we use one camera and do not adjust camera zoom, pan, or tilt. The system could be improved by adding an additional camera that zooms in on the face. This would avoid the tradeoff between being able to adequately find the driver's face in a wide field of view, and zooming in on the eyes to track them better.

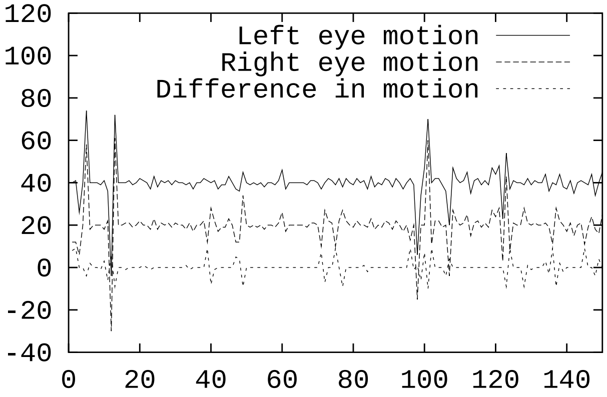

Figure 8: Monitoring head motion. The graphs show the up-and-down motion of the left eye, right eye, and the differences between left and right eye motion over 150 frames, respectively. For display purposes, the top two graphs are shifted up by 20 and 40 pixels.

References

[1] S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. In *Advances in Neural Information Processing Systmes (NIPS)*, volume 6, 1994.

[2] M. Betke and J. Kawai. Gaze detection via self-organizing gray-scale units. In *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 70–76, Kerkyra, Greece, September 1999. IEEE.

[3] M. Betke, W. J. Mullally, and J. Magee. Active detection of eye scleras in real time. In *Proceedings of the IEEE Workshop on Human Modeling, Analysis and Synthesis*, June 2000.

[4] E. R. Boer, E. C. Hilredth, and M. A. Goodrich. Drivers in pursuit of perceptual and virtual targets. In *Proceedings of the International Conference on Intelligent Vehicles*, pages 291–296, Stuttgart, Germany, October 1998. IEEE Industrial Electronics Society.

[5] T. Darrell, I. A. Essa, and A. Pentland. Task-specific gesture analysis in real time using interpolated views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1236–1242, 1996.

[6] M. Eriksson and N. Papanikolopoulos. Eye-tracking for detection of driver fatigue. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, Boston, MA, November 1997.

[7] M. Fleck, D. Forsyth, and C. Bregler. Finding naked people. In *Lecture Notes in Computer Science. Vol. 1065: Proceedings of the 4th European Conference on Computer Vision*, volume II, pages 592–602. Springer-Verlag, Berlin, April 1996.

[8] T. Fujioka and K. Muramatsu. Drivers' behavior in ITS environment investigated by a driving simulator. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 295–299, Tokyo, Japan, September 1996. IEEE Industrial Electronics Society.

[9] Y. Matsumoto and A. Zelinsky. Real-time stereo face tracking system for visual human interfaces. In *Proceedings of the International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, pages 77–82, Kerkyra, Greece, September 1999. IEEE.

[10] D. Middleton. *Introduction to Statistical Communication Theory.* Peninsular Publishing, Los Altos, CA, 1987.

[11] T. Nakano, K. Sugiyama, M. Mizuno, and S. Yamamoto. Blink measurement by image processing and application to warning of driver's drowsiness in automobiles. In *Proceedings of the International Conference on Intelligent Vehicles*, pages 285–290, Stuttgart, Germany, October 1998. IEEE Industrial Electronics Society.

[12] N. Oliver, A. Pentland, and F. Bérard. LAFTER: Lips and face real-time tracker with facial expression recognition. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, Puerto Rico, 1997.

[13] T. Rikert and M. Jones. Gaze estimation using morphable models. In *International Conference on Automatic Face- and Gesture- Recognition*, 1998.

[14] Y. Sakaguchi, T. Nakano, and S. Yamamoto. Development of non-contact gaze detecting system and its applications to gaze duration measurement of on-board display. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 289–294, Tokyo, Japan, September 1996. IEEE Industrial Electronics Society.

[15] A. Saroldi, E. Ansaldi, C. Sidoti, and M. Miglietta. Describing human driving behavior. In *Proceedings of the International Conference on Intelligent Vehicles*, pages 302–304, Stuttgart, Germany, October 1998. IEEE Industrial Electronics Society.

[16] B. Schiele and A. Waibel. Gaze tracking based on face color. In *International Workshop on Automatic Face- and Gesture-Recognition*, 1995.

[17] L. Sigal, S. Sclaroff, and V. Athitsos. Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159, June 2000.

[18] Y. Yacoob and L. S. Davis. Recognizing human facial expressions from long image sequences using optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):636–642, 1996.