Active Detection of Eye Scleras in Real Time

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

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

Abstract

We introduce a method to detect the white visible portion of the eyeball, the "sclera." Our method is embedded in a real-time vision system that actively controls the camera's pan, tilt, and zoom. We designed the system to automatically detect the head of a moving person and zoom towards the face until the eyes are imaged with sufficient resolution. The scleras are then detected using color-based Bayes decision thresholds. We tested our system for various subjects, head and facial motions, and lighting conditions.

1 Introduction

In the near future, standard desktop computers will be equipped with cameras that can capture the computer user's head orientation, facial expression, lip movement, and gaze direction. A computer vision system that interprets this information reliably has the potential to become a new communication tool and augment traditional human-computer interfaces such as keyboard and mouse. Such a system will also have an important impact on people who cannot use the keyboard or mouse due to severe disabilities. Our research is motivated by the goal to provide a communication tool to non-speaking children with cerebral palsy and traumatic brain injuries at Boston College's campus school. Currently two systems are used as mouse replacements and important means of communication by several children at the campus school. The older system, called "EagleEyes," is based on measuring the user's electro-oculographic potential [11, 6]. Our new system, the "Camera Mouse" is based on facial feature tracking using a video camera [10]. The children use the systems to spell out words or messages and play games. This paper describes our steps towards substituting EagleEyes with an inexpensive, unobtrusive computer vision system that not only tracks features but also detects them. Our ultimate goal is to automatically determine the gaze direction of a computer user. Our current system addresses the following two tasks:

- Detecting a face, zooming towards it, and tracking it.
- Detecting the scleras, the white visible portion of the eyeballs.

Our system performs these tasks in real time using various subjects, head and facial motions, and lighting conditions. We developed a color-based technique to detect eye scleras. The color of eye scleras, a yellowish white, does not vary much between different subjects, while iris color, eye shape, and surrounding skin color generally vary substantially.

1.1 Previous Work

Various techniques have been used to detect and track people, and their faces and eyes in real time. Temporal differencing is often used to segment moving a region of interest from a stable background [5, 19]. Facial motions have been analyzed in real time or near real time, using normalized color histograms [3, 18], parametric flow models [21], models of facial dynamics [4, 7], Hidden Markov Models [14], and stereo systems [12]. We are unaware of any previous work that addressed sclera detection. Skin color based detection of faces, however, has been explored extensively, e.g., [8, 16, 20, 22].

Gaze estimation has proven to be a challenging problem. Previous approaches include systems based on neural networks [1, 17], morphable models [15], and self-organizing gray-scale units [2]. Gee and Cipolla [9] explore the underlying geometric constraints.

*The author acknowledges support by NSF equipment grant 9871219.
Email: betke@cs.bu.edu, http://www.cs.bu.edu/fac/betke.

2 Statistical Sclera and Skin Color Models

We use statistical decision theory [13], in particular, Bayes decision rule to estimate from the color of a pixel if it images a face or the sclera of an eye. A training data set of images that are known to contain faces is analyzed to determine a priori probability distributions of skin and sclera color. 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.

2.1 Sclera Color

Our training data consists of images taken from 8 subjects under two different lighting conditions. First, only the neon ceiling lights in our laboratory were used, and a set of images was obtained that shows dark shadows around the subject's eyes. Then an additional desk lamp was placed to brighten the subject's face. We segmented the white of the eyes in each training image by hand.

Figure 1 shows a sample set of our training images, where the subject's sclera is segmented by hand and displayed in pure white. Figure 2 plots the distributions for the sclera and non-sclera pixels for data points $v_1 = red - green$ and $v_2 = green - blue$. The non-sclera pixels are also segmented by hand to include all pixels that make up the eye region except the sclera, i.e., the iris, pupil, eye brows, lashes, and lids. The distributions were computed using 7550 training pixels for the sclera color, and 275,547 for the non-sclera color, and taken under the same desk and ceiling light conditions.

Figure 1: Training images for sclera color taken with ceiling lights (top row), ceiling and desk lights (bottom row). The black region in the right bottom image is used as training data for non-sclera eye color.

Table 1 lists sample means and variances for the training data distributions. Figure 2 shows Gaussian approximations based on these statistics. We use them to define the likelihood functions

$$p(v_1|sclera) = \frac{1}{\sqrt{2\pi}\sigma_1}\exp(-\frac{(v_1 - m_1)^2}{2\sigma_1^2}) \tag{4}$$

for $v_1 = red - green$, $m_1 = 29$, and $\sigma_1^2 = 121$, and

$$p(v_1|eye) = \frac{1}{\sqrt{2\pi}\sigma_2}\exp(-\frac{(v_1 - m_2)^2}{2\sigma_2^2}) \tag{5}$$

for $v_1 = red - green$, $m_2 = 42$, and $\sigma_2^2 = 231$. The likelihood functions for $v_2 = green - blue$ are defined similarly. We assume that sclera or non-sclera colors are equally likely to occur within the eye region. Then the prior probabilities p and q can be set to 1/2 and the likelihood ratio is

$$\ell(\mathbf{v}) = \frac{p(v_1|sclera)p(v_2|sclera)}{p(v_1|eye)p(v_2|eye)}. \tag{6}$$

If the same costs are associated to false positive and false negative decisions, i.e., $\mathcal{H} = C_1/C_2 = 1$, pixels with $v_1 = red - green \leq 36.4$ and $v_2 = green - blue \leq 8.3$ are classified to have sclera color. These thresholds on v_1 and v_2 are given by the intersections of $p(\mathbf{v}|sclera)$ and $p(\mathbf{v}|eye)$ in Fig. 2. However, for our application it is worse to miss a true sclera pixel rather misclassify a non-sclera pixel. We

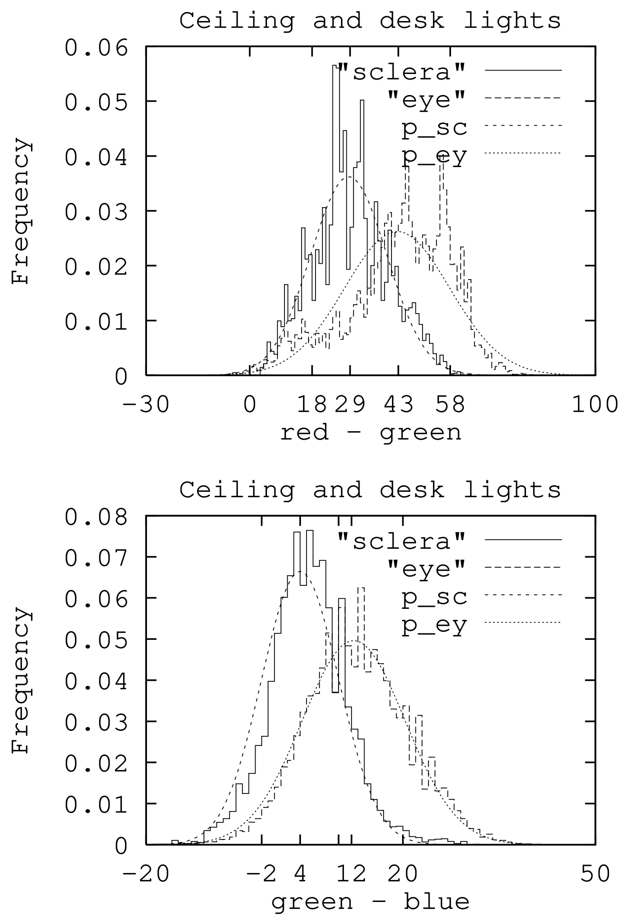

Figure 2: Distributions of sclera and surrounding eye color of training data and respective likelihood functions $p(v_1|sclera), p(v_1|eye), p(v_2|sclera)$, and $p(v_1|eye)$. On top $v_1 = red - green$, at bottom $v_2 = green - blue$.

Table 1

Statistics of Training Data				
	Sclera	Eye	Sclera	Eye
Sample	*red - green*		*green - blue*	
Mean	29	42	4	12
Variance	121	231	36	64

therefore use a cost ratio of $C_1/C_2 = 2/3$ as the decision threshold $\mathcal{H}$. This results in classifying pixels with $v_1 \leq 40$ and $v_2 \leq 12$ as sclera colored.

Figure 3 shows the sclera classification results for a typical scene. Sclera-colored pixels are shown in white. Notice that glare at the border of the face and clusters of background pixels are also identified as sclera colored. To avoid such misclassifications, our system first identifies the face and then only searches the face area for sclera-colored pixels. Skin-colored pixels are used to help identify the face. We use the same techniques as described above to determine decision thresholds for skin-color classification. Misclassification of skin color is addressed by a model-based approach as described in Section 4.

Figure 3: Pixels that match the sclera color are shown in white; other pixels are shown in black.

3 System Overview

Our system actively acquires and processes live video input of a person and outputs an online description of location and size of the person's face and eyes. The vision system contains five main components: camera initialization, process coordination, face detection, face tracking, and eye detection. Figure 4 provides a system flow chart.

In the initialization phase, the camera is positioned at zero pan and tilt angles and widest field of view. Frame acquisition is then started with the face detector searching the full frame. Once the face detector recognizes a face, the process coordinator creates a tracking process. While the face is being tracked, the system decreases the camera's field of view and zooms towards the detected face until it appears large enough to employ the eye detector.

The process coordinator uses the detection and tracking history to decide whether a face estimate is reliable. If the process coordinator concludes that the face disappeared or is tracked incorrectly, it switches control from the face tracker back to the full-frame face detector. The dynamic use of either face detector and tracker reduces the amount of computational

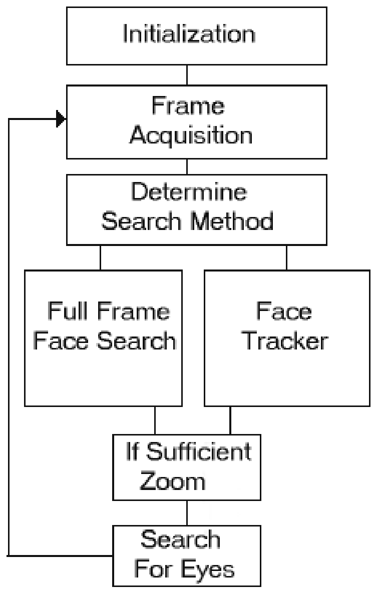

Figure 4: System overview

resources needed and allows real-time detection and tracking.

4 Face Detection

The task of the face detector is to identify a person's head and rotate the camera so that the face is in the center of the image. It 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 5, 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.

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 due 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.

Figure 5: A skin-tone motion image. Bright pixels indicate significant temporal changes in skin tone, black pixels indicate no change.

Once the face detector identifies a set of coordinates $(x_t, y_t), (x_b, y_b), (x_l, y_l)$ and (x_r, y_r) that potentially describe the borders of a face, it counts skin-tone pixels (x, y) within the ellipse

$$\frac{(x - (x_r + x_l)/2)^2}{((x_r - x_l)/2)^2} + \frac{(y - (y_b + y_t)/2)^2}{((y_b - y_t)/2)^2} \leq 1$$

that is defined by these coordinates (see Fig. 6). If the skin-tone portion of the ellipse is large enough to provide evidence that it indeed models a face, the camera is repositioned to center the face in the image. At this point, the process coordinator switches control to the face tracker. It also disables the face detector for the next few frames to increase the speed of the system.

Figure 6: The face in the left image is detected and modeled by the ellipse shown in the right image.

5 Face Tracking

The task of the face tracker is to

- follow the face by rotating the camera and
- obtain or maintain a resolution that allows reliable eye detection by changing the camera zoom.

To determine the face outline in the current image frame, the face tracker uses as inputs either its own or the face detector's estimates of face border coordinates in the previous frame.

The face tracker and detector determine the face outline in a similar manner. The main differences lie in the search size, data, and direction. Instead of searching the entire frame, the face tracker only searches an image region slightly larger than the region covered by the detected face in the previous frame. Instead of searching for strong edges in the skin-tone motion image alone, the face tracker includes in its search both skin-tone pixels and skin-tone motion pixels. This enables the tracker to identify the face even if there is no or only small motion. The face tracker searches the projection vectors from the center of the search region towards its borders. Once it identifies a set of coordinates that possibly describe the face outline, the face tracker checks whether the skin-tone portion of the ellipse defined by these coordinates is large compared to the size of the ellipse. If this portion is large, the coordinates are considered to describe a face.

Once the face tracker has identified a face, it checks the size of the elliptic face model and determines if the camera's field of view should be decreased to increase the size of the face within the image. After the potential change of the camera's zoom, system control moves on to the eye detector.

6 Eye Detection

The task of the eye detector is to find the eyes in a face by locating the left and right eye scleras. The eye detector is started by the process coordinator as soon as the identified face region contains at least two-thirds of the size of the image. This threshold is needed to ensure a resolution of the eyes in the image that makes sclera detection feasible. The threshold was not chosen arbitrarily, but in fact tested extensively.

The eye detector identifies all sclera-colored pixels within the face using Bayes decision rule as described in Section 2.1. To improve sclera color classification, the pixels in the face region are also analyzed for skin tone and vertical edges and weights are assigned to combine color and edge information as follows. A weight of zero is assigned for a skin-colored pixel, a weight of one for a pixel matching the non-skin color or indicating the presence of a vertical line, and a weight of three for a sclera-colored pixel. The highest concentration of weights can be expected to occur around the eyes, where large pixel clusters of sclera-white, clusters of non-skin color due to pupils and irises, and vertical edges due to eyelashes and iris borders are the most prominent features.

The peak concentration is found by filtering the weight map of the face. The filter is defined by a computational mask of 5×5 that averages the local weights that are associated with the pixels. The eye detector first searches for a peak filter output that represents the center of the left eye. The search includes only pixels located within the left half of the elliptic face model. Once the left eye is found in this manner, the eye detector searches the right half of the ellipse for the right eye.

7 Hardware

Our system uses a Sony EVI-D30 color video CCD camera. Its NTSC video output is processed by a Matrox Meteor II image capture board on a 450 MHz dual processor PC with 384MB RAM. Our system controls the camera's pan, tilt, and zoom mechanisms via the PC's serial port. The camera's pan and tilt angles are $\pm 100^o$ and $\pm 25^o$, respectively. Horizontally the camera's field of view can change from 48.8^o to 4.3^o; vertically it can change from 37.6^o to 3.2^o. The camera has autofocus. Our system processes images of size 320×240 pixels, which is half the resolution that the camera provides.

8 Real-time Performance

Our system processes between 5 and 14 frames per second. The wide range of possible frame rates is due to the active nature of our system. During detection and tracking phases that do not require any adjustments of the camera's controls, the system runs about 12 to 14 frames per second. When our system decides that repositioning of the camera becomes necessary and activates the camera's control mechanisms, image acquisition cannot occur and the frame rate drops.

At the beginning of the experiments, the camera is initialized to use its widest field of view. When a face is detected at this wide angle, the camera takes several seconds to zoom towards the face until a sufficient resolution of the eyes is obtained and the scleras can be detected. Due to frequent camera readjustments, the frame rate is only about 5 frames per second during this time.

9 Experiments and Discussion

Since our system *actively* acquires and processes video input, it only works in *live* experiments. This complicates the analysis of our system's performance. We cannot work with a database of image sequences and exploit the advantage that stored test sequences provide, namely, repeatability of experiments. When a test person becomes available, both image acquisition and processing, and analysis of results must all be done in a live session. In these live experiments, we find that the system locates and zooms towards faces well and detects eyes reasonably well. A simpler version of our system that manually determines the field of view and automatically detects and tracks eye scleras has been tested extensively in several public demonstrations that included about 100 test subjects. *All* subject eyes were detected and tracked successfully, independent of a subject's age, race, sex, facial hair, glasses, etc.

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 and eye locations. Saving images for later analysis slows down the system to about 4 to 6 frames per second, and therefore causes a significant impact on its detection performance.

We first tested for long-term tracking performance. We recorded 190 images over the period of 11.5 minutes, choosing a uniform sampling rate. The outline of the head was identified correctly in 75% of the stored images. An eye matched correctly in 63% of the sample images.

We then tested how well our system can track a face if the subject moves around significantly. Within 2500 frames, the subject made 14 drastic movements so that only half of the subject's face was imaged in the frame that immediately followed the move. The system correctly repositioned the camera 71% of the time. The system repositioned the camera within the span of 30 to 100 frames. It failed if the subject moved too fast out of the camera's field of view before the camera could reposition itself.

We also tested our system's performance on eight different subjects in 18 live tests, each lasting 33 seconds. For analysis purposes, 48 images were stored per test. They included 13 images with eye localization. The initial zooming process was 95% successful, taking between 86 and 254 frames until an optimal field of view was obtained. The system localized at least one eye in 89% of the cases. Figure 7 shows how the vision system actively changes the field of view once it detects a face. The camera zooms in until the face is imaged large enough for the eyes to be detected. The person in the sequence on the right moved out of the field of view, which delayed the zooming process.

Figure 8 illustrates successful eye detection. Figures 9 and 10 show cases where only one eye is detected or eye detection failed. Mismatches are due to closed eyes, misidentification of the face outline, and problems with the autofocus.

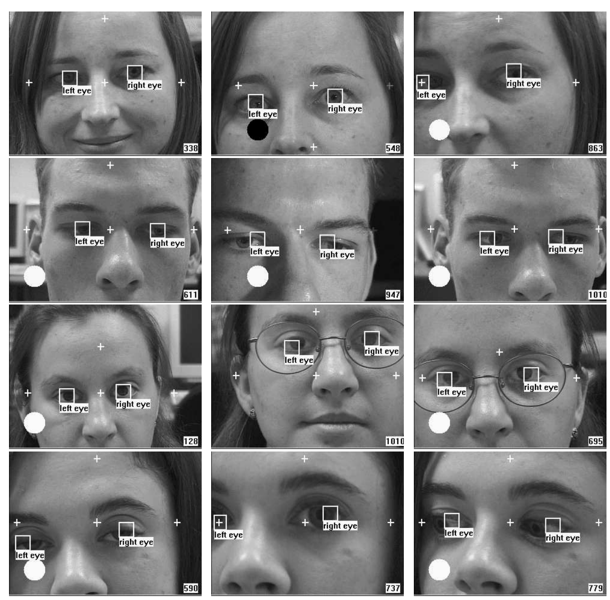

Figure 8: Eye detection and iris localization.

Figure 7: Once a face is detected, the system rotates to center the face within the image frame and widens the field of view. The faces detected in frame 23 have a width of about 55 pixels. Eye detection starts once the width of the imaged face is 231 pixels.

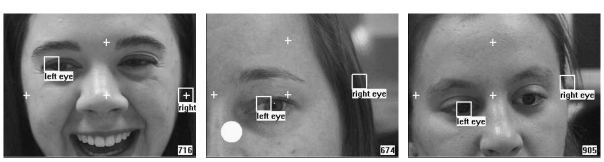

Figure 9: Localization of one iris and false match with hair or background.

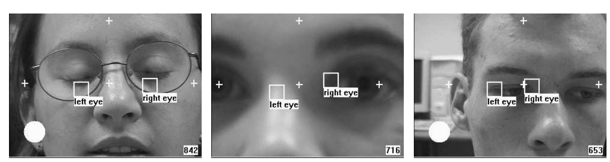

Figure 10: Incorrect localization of eyes due to closed eyes, failure of camera's autofocus, and incorrect face localization.

10 Future Work

We have presented a system that detects, tracks, and zooms in on faces, and locates eyes. We developed a statistical model based on Bayes decision rule to detect the color of the sclera of an eye. Our plan for the future is to add geometric constraints to our face and eye models that improve eye detection without substantially reducing the real-time performance of our system. To make eye detection reliable over long time periods, we will also add an eye tracker to our system.

We strongly believe that reliable sclera detection is an important tool for estimating gaze direction, which is our ultimate goal.

Acknowledgements

We would like to thank Ismail Haritaoglu for sharing his expertise on controlling the EVI-D30 camera.

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] J. L. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In *Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition*, pages 640–645, Puerto Rico, June 1997.

[4] 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.

[5] J. Davis and A. Bobick. The representation and recognition of action using temporal templates. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 928–934, Puerto Rico, June 1997.

[6] The EagleEyes Project at Boston College. http://www.cs.bc.edu/~eagleeye.

[7] I. A. Essa and A. Pentland. Tracking facial motion. In *Proceedings of the IEEE Workshop on Motion of Nonrigid and Articulated Objects*, pages 36–42, 1994.

[8] 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.

[9] A. Gee and R. Cipolla. Determining the gaze of faces in images. *Image and Vision Computing*, 12(18):639–647, 1994.

[10] J. Gips, M. Betke, and P. Fleming. The Camera Mouse: Preliminary investigation of automated visual tracking for computer access. In *Proceedings of the RESNA 2000 Annual Conference*, Orlando, FL, July 2000.

[11] J. Gips, P. DiMattia, F. X. Curran, and P. Olivieri. Using EagleEyes – an electrodes based device for controlling the computer with your eyes – to help people with special needs. In J. Klaus, E. Auff, W. Kremser, and W. Zagler, editors, *Interdisciplinary Aspects on Computers Helping People with Special Needs*. R. Oldenbourg, Vienna, 1996.

[12] 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.

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

[14] 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.

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

[16] D. Saxe and R. Foulds. Toward robust skin identification in video images. In *Second International Conference on Automatic Face and Gesture Recognition*, Killington, VT, 1996.

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

[18] K. Schwerdt and J. L. Crowley. Robust face tracking using color. In *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, March 2000.

[19] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. PFINDER: Real-time tracking of the human body. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 19(7):780–785, 1997.

[20] H. Wu, Q. Chen, and M. Yachida. Face detection from color images using a fuzzy pattern matching method. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 21(6), June 1999.

[21] 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.

[22] B. D. Zarit, B. J. Super, and F. K. H. Quek. Comparison of five color models in skin pixel classification. In *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 58–63, Kerkyra, Greece, September 1999. IEEE.