Automatic Eyewinks Interpretation System Using Face Orientation Recognition For Human-Machine Interface

Mamatha M. N.*
Assistant professor
Department of Instrumentation
B.M.S. College of Engineering
Bangalore .

Dr. S. Ramachandran.**
S. J. B. Institute of technology
Department of ECE
Bangalore

Abstract
This paper proposes an automatic face orientation interpretation system for human-machine interface to benefit the severely handicapped people. Our system investigates a new method for movement identification and uses a Hybrid scheme consisting of extracting the Eigen values of covariance matrices using Hough Transform and Bresenham's Raster scan Algorithms. The proposed template matching algorithm has been found effective in tracking the movement of the eyes and, using a classifier it is possible to verify the open or closed eyes and convert the eye winks into a sequence of codes (0 or 1) and subsequently translate the code sequence to a certain valid command. The present work differs from the previous eye-gaze tracking methods, in that, the system identifies the open or closed eye and then interprets the eye winking as certain commands for human-machine interface. Our system demonstrates better performance as well as higher accuracy as shown in the results.

Keywords: Hough Transform, Face detection, Eye tracking, Human machine interface.

1. Introduction

Recently, there has been an emerging research interest in the field of machine perception focusing on automatic eye detection and tracking. It can be applied for vision-based human-machine interface (HMI) applications such as monitoring human vigilance [1–6] and assisting the disabled [7, 8]. The eye detection and tracking approaches can be classified into two categories: CCD camera-based approaches [1–11] and active IR-based approaches [12–15]. An eye-wink control interface [7] is proposed to provide the severely disabled with increased flexibility and comfort. The eye tracker [2] makes use of a binary classifier with a dynamic training strategy and an unsupervised clustering stage to efficiently track the pupil (eyeball) in real time. Based on optical flow and color predicates, the eye tracking [4] can robustly track a person's head and facial features. It classifies the rotation of all viewing directions, detects eye blinking and recovers the 3D gaze of the eyes. In Ref. [5], the eye detection operates on the entire image, looking for regions that have the edges with a geometrical configuration similar to the expected one of the iris. It uses the mean absolute error measurement for eye tracking and a neural network for eyes validation. The eye movement collection data can be analyzed to determine the pattern and duration of eye fixations and the sequence of scan path as a user visually moves his eyes. The active approaches [12–15] make use of IR devices for the purposes of pupil tracking based on the special bright
pupil effect. This is a simple and very accurate approach to pupil detection using the differential infrared lighting scheme. It is capable of handling sudden changes between IR and non-IR light conditions, without changing parameters. However, the active methods [12–15] require additional resources in the form of infrared sources and infrared sensors.

In this work, a vision-based eye-wink control interface for helping the severely handicapped people (to manipulate the household devices) is presented. In this treatment, we assume that the possible head poses of severely handicapped people are very limited.

The next section presents the methodology of the work. In Section 3, algorithm developed for face detection and tracking is presented. The proposed methodology and results are presented in Sections 4 and 5 respectively. The last section presents the conclusion and scope for future work.

2. Methodology

Under the front-pose assumption, we may easily locate the eyes, track the eyes, and then identify the open or closed eye. Before eye tracking, we use the skin color information to find the possible face region which is a convex region larger than a certain size. In this work, the Hough Transform is used to isolate features of a particular shape within an image. The techniques guarantee a robust method for object identification and feature extractions in images. After the initial step, in the next frame, the template matching is applied to track the eyes based on the eye template extracted in the previous frame. The eye

Manuscript received May 5, 2009
Manuscript revised May 20, 2009

template is updated every time the eye is successfully tracked. After eye tracking, we apply classifier to verify whether the tracked block is an eye or a non-eye region. If it is an eye region, then it is further classified to identify whether it is open or closed, and then convert the eye winks to a sequence of ones and zeros. Finally, we validate the code sequence and convert it into a certain command. The flow chart of the proposed system is illustrated in Figure 1.

3. Face Detection and Tracking

Our face detection and tracking method consists of three stages: 1. face region detection, 2. face localization and 3. eye tracking. These three steps are illustrated in the following sections.

3.1 Hough Transform

In this work, the Hough Transform is used for face region identification and subsequently localization of the eye. The steps involve parametric analysis of HT, statistical analysis based on Eigen values and geometrical properties of objects to identify the geometric primitives. A Line in Cartesian

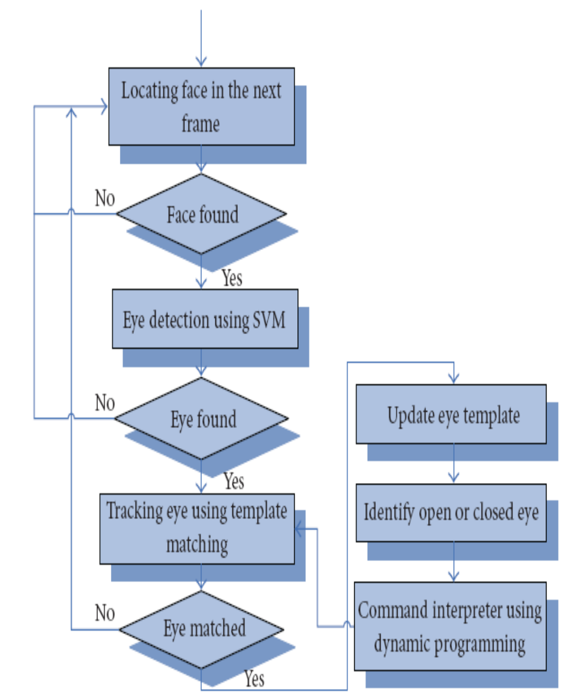


Figure 1 Flowchart of the eye-winks interpretation system

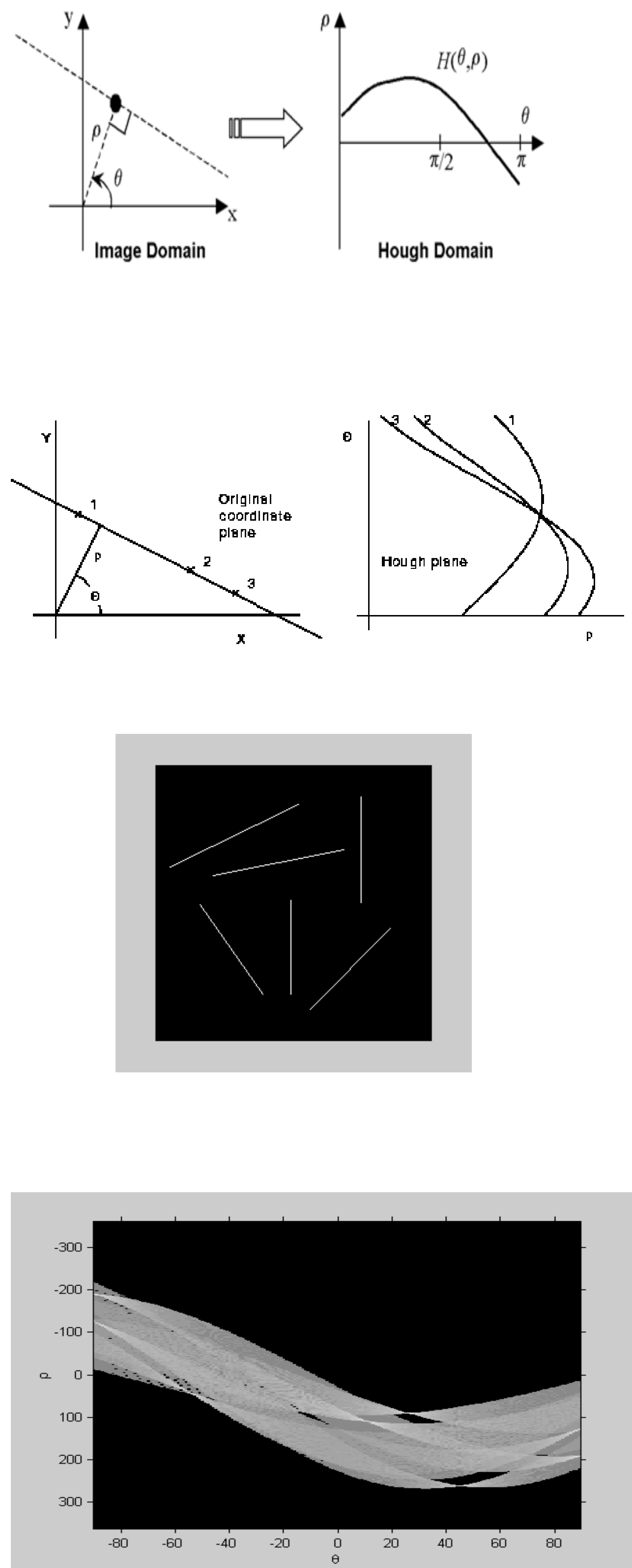


Figure 2 Illustration of the Hough transform

co-ordinates is represented by y = mx+c. In Polar co-ordinates, it is represented as ρ = xcosθ+ysinθ. If we want to draw the same line in (θ, ρ) space, each point of the line generates a sinusoid. This (θ, ρ) space is called Hough space (Fig. 2). For example, the small eigen values for a line segment in the continuous domain will be zero. Similarly, the large eigen values and small eigen values are equal if the object is a full circle. Alternatively, if the object is an ellipse, the large eigen values and small eigen values corresponds to the major and minor axial lengths of an ellipse. Therefore, eigen values of covariance matrix can be used to extract the primitives in images.

3.2 Face Region Detection

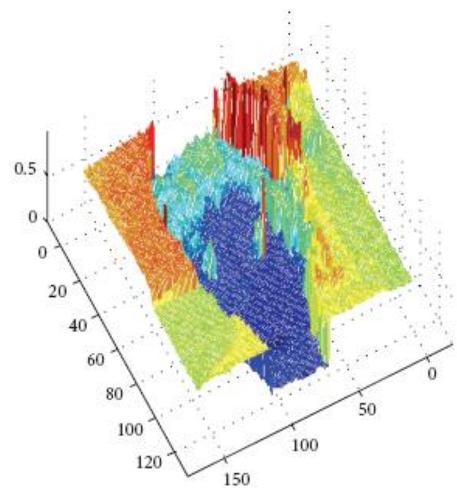
To reduce the eye search region, it is first required to locate the possible face region. In the captured human face images, it can be reasonably assumed that the color distribution of the human face is somehow different from that of the image background.

3.3 Algorithm

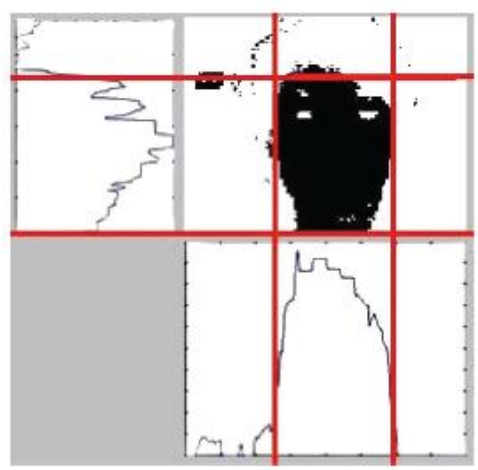
Pixels belonging to face region exhibit similar chrominance values within and across people of different ethnic groups [16]. However, the color of face region may be affected by different illuminations in the indoor environment. For skin color detection, the color of the pixels in HSI color space is analyzed to decrease the effect of illumination changes, and then classify the pixels into face color or non-face color based on their hue component. From 100 training close-up face images, the probability density function of hue value H is obtained, which can be either face color and non-face color, i.e., p(*H*—*face*) and p(*H*—*non-face*). Based on hue statistics, we use the Bayesian approach to determine the face color region. Each pixel is assigned to the *face* or *non-face* class that gives the minimal cost when considering cost weightings on the classification decisions. The classification is performed by using the Bayesian decision rule which can be expressed as: if p(*H*—*face*)/p(*H*—*non-face*) > τ, then the pixel (with *H* hue value) belongs to a face region; otherwise it is inside a non-face region, where τ = p(*non-face*)/p(*face*).

In order to determine the face region, the following steps are performed: (i) the vertical and the horizontal projections on the classified face pixels (ii) locate the right and left region boundaries where the projecting value exceeds a certain threshold. With 100 test images of different subjects, background complexities and lighting conditions, the correct face detection rate is found to be 88% in this work. It is worth mentioning that the proposed system does not

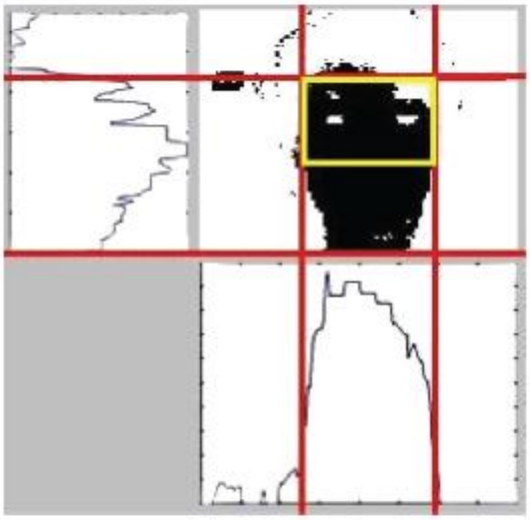
a



b



c



d

Figure 3: Results of face detection: (a) The original image (b) Hue distribution of the original image (c) Skin color projection and (d) Possible eye region

know whether the identified face region is accurate or not. However, if the following eye detection cannot locate an eye region, then the detected face region is not a false alarm. After the face region detection, there may be more than one face-like region in the block image. We select the maximum region as the face region. We assume that eyes should be located in the upper half face area. Once the face region is found, it may be assumed that the possible eye region is the upper portion of the face region. These eyes are searched within the yellow rectangle area only. The above steps are illustrated in Fig. 3 for a sample image.

3.4 Eye localization using eigen value extraction

The algorithm for eye localization using eigen value extraction is as follows:
Given a set of points B = {pi | pi = (xi, yi) ∈ Z^2, i = 1, 2, 3,…, n}, then compute

$$Variance(x) = c_{11} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$Variance(y) = c_{22} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

$$Co-Variance(x,y) = c_{12} = c_{21} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

Using these, construct a Matrix $C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$

3.4.1 Eigen Values computation

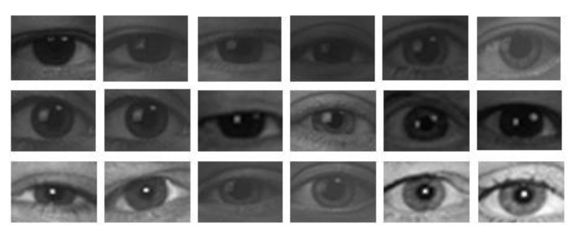
Eigen values computations involve the following steps:
(i) Solve for λ in | C - λI | = 0
(ii) Compute Large Eigen values
(iii) Compute Small Eigen values

$$\lambda_L = \frac{1}{2}\left[c_{11} + c_{22} + \sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2}\right]$$

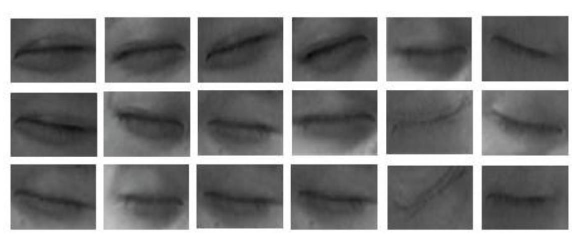
$$\lambda_S = \frac{1}{2}\left[c_{11} + c_{22} - \sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2}\right]$$

(iv) Solve for eigen vector V in CV = λV

An eye edge image is represented by a feature vector consisting of the edged pixel values. We manually select the two classes: positive set (eye) and negative set (non-eye). The eye images are processed by using histogram equalization and their image sizes are normalized to 20×10 pixels. Fig. 4 shows the samples consisting of open eye images, closed eye images, and non-eye images. The eye detection algorithm will search every candidate image block inside the possible eye region to locate the eyes. Each image block is processed by Sobel edge detector.

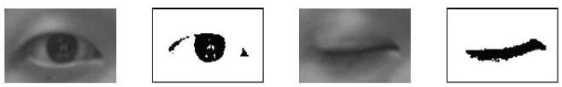


a



b

c



d

Figure 4 (a) Original open eye (b) Original closed eye (c) Binarized open eye and (d) Binarized closed eye

3.4.2 Face position tracking

Face orientation tracking is applied to find the face position in each frame by using template matching. Given the detected face positions in the previous frame, the positions in subsequent frames can be tracked frame by frame. Once the position is correctly localized, we update the facial templates (gray level image) for face position tracking in the

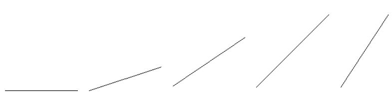


Figure 5 Different eye orientations captured

Table 1 Slope versus Small Eigen values

Slope angle (θ)	**Small Eigen values** λ_s		
	W=3x3	**W=5x5**	**W=7x7**
0^0	**0.00000**	**0.00000**	**0.00000**
10^0	**0.09456**	**0.09538**	**0.09745**
15^0	**0.10372**	**0.10486**	**0.10549**
30^0	**0.11645**	**0.11823**	**0.11942**
45^0	**0.00000**	**0.00000**	**0.00000**

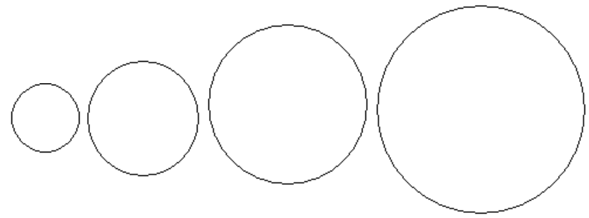


Figure 6 Facial Images with different radii (r=30, r=50, r=70 and r=90 pixels respectively)

Table 2 Small and large eigen values

Radius (pixels)	**Region of support w=5x5**		**Region of support w=7x7**		**Region of support w=9x9**	
	λs	**λL**	**λs**	**λL**	**λs**	**λL**
30	0.249	0.249	1.397	1.398	2.542	2.544
50	0.089	0.090	0.465	0.465	0.840	0.841
70	0.042	0.042	0.231	0.231	0.419	0.422
90	0.023	0.023	0.140	0.140	0.256	0.257

next frame. The different orientations of the face for a sample image are illustrated in figure 5. For each of the orientations, the obtained small eigen values are presented in Table 1. Similarly, for face with different radii (Fig. 6), the obtained small and large eigen values are presented in Table 2.

It is observed that, in the closed eye image, the centroid is located at the center of eyelashes instead of the center of bounding box. Based on the centroids of the binarized images, the eye tracking will be faster and more accurate.

4. Proposed Methodology

The methodology for face orientation detection is given as follows:

Step 1: For the given grayscale image, find the edge image using suitable edge detection operators.
Step 2: Obtain the eigen values of covariance matrix for edge image.
Step 3: Perform the Hough Transform (HT) using sparse matrix technique.

Step 4: Find the meaningful set of distinct Hough peaks using the following steps (Neighborhood suppression scheme):

i) Find the accumulator cell containing the highest value and record its location.

ii) Set to zero the accumulator cells in the immediate neighborhood of the maximum found. Select the window size based on the accuracy needed.

iii) Repeat the above steps until the desired peaks have been found.

Step 5: Once the candidate peaks and their locations are identified, find the coordinates with respect to a particular primitive using Bresenham's Raster Scan algorithm.

Step 6: Construct a full matrix for the nonzero pixels obtained from step 3 for a corresponding primitive.

4.1 The Command Interpreter Using Dynamic Programming

After face tracking, we distinguish between the open eye and the closed eye. If the eye opens and exceeds a fixed duration, then it represents a binary digit "1". Similarly, the closed eye represents a "0". So we can convert the sequence of eye winks to a sequence of 0s and 1s. The command interpreter validates the sequence of codes and issues the corresponding output command. Each command is represented by the corresponding sequence of codes. Starting from the base state, the user issues a command by a sequence of eye winks. The base state is defined as an open eye for a long time without intentionally closing the eye. The input sequence of codes is then matched with the predefined sequence of codes by the command interpreter. To avoid an unintentional or very short eye wink, we require that the duration of a valid open or closed eye should exceed a duration threshold θtl . If the time interval of the continuously open or closed eye is longer than θtl , then it can be converted to a valid code, that is, "1" or "0". However, we may allow two contiguous "1"s or "0"s, so we define another threshold $\theta th \approx 2\theta tl$. If the time interval of the continuously open or closed eye is longer than θth, then we may consider it as code "00" or "11". The threshold θtl is user dependent; user may select the best suitable threshold for his specific eye blinking condition. Here, we may predefine some valid code sequences, and each one corresponds to a specific command. Once the code sequence has been issued, we need to validate the code sequence. To find a valid code sequence, we need to calculate the similarity (or alignment) score between the issued code sequence and the predefined code sequences. Because the code lengths are different, we need to align the two code sequences to maximize the similarity by using the dynamic programming [19]. We assume the predefined codes as shown in Table 3.

Table 3 Predefined Code lengths

Code length	1	3	4	5
—	0	010	0010	00100
—	—	—	0100	00110
—	—	—	0110	01010
—	—	—	—	01100

5. Experimental Results

A Logitech Quick Cam Pro 3000 camera was used to capture the video sequence of the disabled at a picture resolution of 320×240 pixels. The system was implemented on a PC with Athlon 3.0 GHz CPU

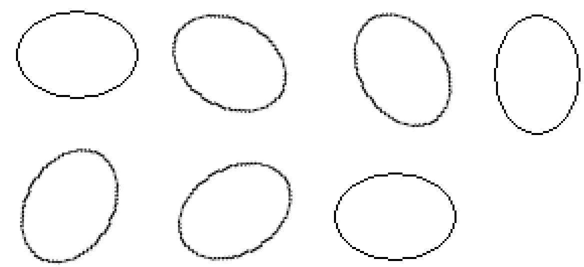
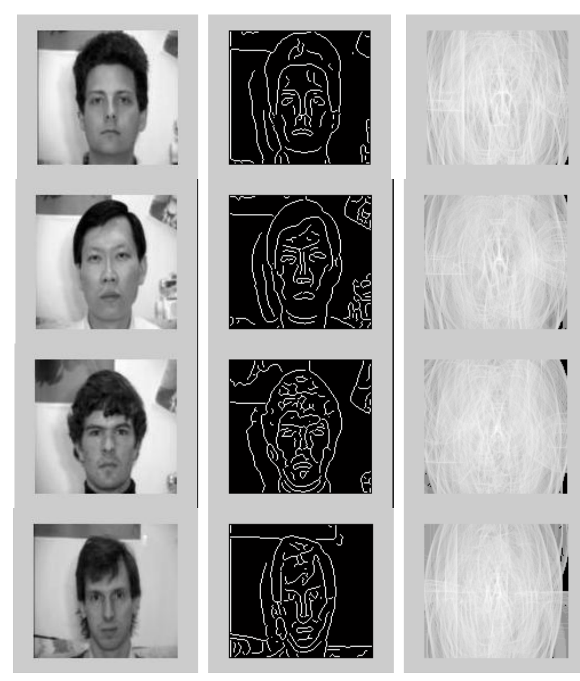


Figure 7 Faces with different orientations



a b c

Figure 8 Edge images and HT transforms obtained
a Original Image b Edge image c HT

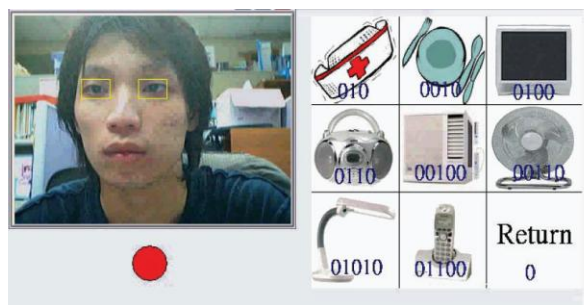
a b c

Figure 9 Segmented faces from the ellipse images
a Ellipse detected b Ellipse marked c Segmented Face

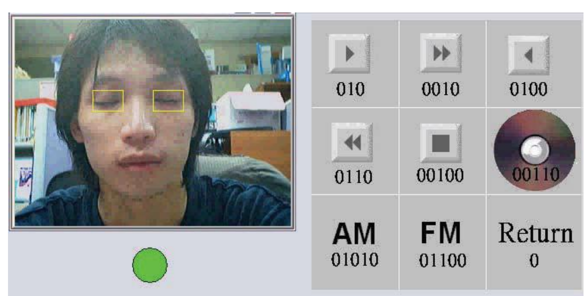
Figure 10 Faces with different orientations
a Tilted Face b Ellipse detected c Ellipse marked

using the Microsoft Windows XP operating system. Faces with different orientations as shown in Fig. 7 were captured and given as input to the algorithm described earlier. The small and large eigen values obtained for each of these cases is given in Appendix 1. The original images, their edge detected images and the corresponding Hough transforms obtained are given in Fig. 8. Similarly, Fig. 9 gives the segmented faces from the marked ellipse images. Fig. 10 shows the results for faces with different orientations.

Fig. 11 shows the system control interface. The red solid circle indicates that the eyes are open. Similarly, the green solid circle indicates that the eyes are closed. There are nine blocks at the right portion. In each block, there is a binary digit number representing the specific command code. In the base mode, we design eight categories: medical treatments, a diet, a TV, a radio, an air conditioner, a fan, a lamp and a telephone. In this system, there are two layers in the command mode. Therefore, we can create at the most 9x9, i.e., 81 commands. However, we can effectively use only 8x8+1, i.e., 65 commands since we have a "Return" command in each layer. In Fig. 11, we illustrate layer 1 and layer 2 commands.



a



b

Figure 11 Program interface
a Layer 1 commands b Layer 2 commands for audio

6. Conclusion and Futurework

In this work, an effective algorithm for face orientation interpretation for human-machine interface is proposed. The scheme is an efficient and accurate method for primitive identification. Conventional schemes such as HT variants take more memory and computational time, whereas the proposed method takes less memory (sparse matrices) and has less error rate and computational time. It is highly suitable to extract specified segment. Experimental results have illustrated improved performance of the proposed methods in terms of both accuracy and speed. Future direction of research could be to extend the algorithms for eye wink detection and the scope of HMI.

References

[1] P.W.Hallinan, "Recognizing human eyes," in *GeometricMethods in Computer Vision*, vol. 1570 of *Proceedings of SPIE*, pp. 214–226, San Diego, Calif, USA, July 1991.

[2] S. Amarnag, R. S. Kumaran, and J. N. Gowdy, "Real time eye tracking for human computer interfaces," in *Proceedings of the International Conference on Multimedia and Expo (ICME '03)*,
vol. 3, pp. 557–560, Baltimore, Md, USA, July 2003.

[3] Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-Time Imaging*, vol. 8, no. 5, pp. 357–377, 2002.

[4] P. Smith, M. Shah, and N. da Vitoria Lobo, "Monitoring head/eye motion for driver alertness with one camera," in *Proceedings of the 15th International Conference on Pattern Recognition (ICPR '00)*, vol. 4, pp. 636–642, Barcelona, Spain, September 2000.

[5] T. D'Orazio, M. Leo, P. Spagnolo, and C. Guaragnella, "A neural system for eye detection in a driver vigilance application," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITS '04)*, pp. 320–325, Washington, DC, USA, October 2004.

[6] K. F. Van Orden, T. P. Jung, and S. Makeig, "Combined eye activity measures accurately estimate changes in sustained visual task performance," *Biological Psychology*, vol. 52, no. 3, pp. 221–240, 2000.

[7] R. Shaw, E. Crisman, A. Loomis, and Z. Laszewski, "The eye wink control interface: using the computer to provide the severely disabled with increased flexibility and comfort," in *Proceedings of the 3rd Annual IEEE Symposium on Computer- Based Medical Systems (CBMS '90)*, pp. 105–111, Chapel Hill, NC, USA, June 1990.

[8] L. Gan, B. Cui, and W. Wang, "Driver fatigue detection based on eye tracking," in *Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA '06)*, vol. 2, pp. 5341–5344, Dalian, China, June 2006.

[9] A. Haro, M. Flickner, and I. Essa, "Detecting and tracking eyes by using their physiological properties, dynamics, and appearance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, vol. 1, pp. 163–168, Hilton Head Island, SC, USA, June 2000.

[10] A. Iijima, M. Haida, N. Ishikawa, H. Minamitani, and Y. Shinohara, "Head mounted goggle system with liquid crystal display for evaluation of eye tracking functions on neurological
disease patients," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '03)*, vol. 4, pp. 3225–3228, Cancun, Mexico, September 2003.

[11] I. Fasel, B. Fortenberry, and J. Movellan, "A generative framework for real time object detection and classification," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 182– 210, 2005.

[12] Z. Zhu and Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 124–154, 2005.

[13] C. H. Morimoto and M. Flickner, "Real-time multiple face detection using active illumination," in *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition (FG '00)*, pp. 8–13, Grenoble, France, March 2000.

[14] X. Liu, F. Xu, and K. Fujimura, "Real-time eye detection and tracking for driver observation under various light conditions," in *Proceedings of the IEEE Intelligent Vehicle Symposium (IV '02)*, vol. 2, pp. 344–351, Versailles, France, June 2002.

[15] D. W. Hansen and A. E. C. Pece, "Eye tracking in the wild," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 155–181, 2005.

[16] D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 551– 564, 1999.

[17] D. Chai, S. L. Phung, and A. Bouzerdoum, "Skin color detection for face localization in human-machine communications," in *Proceedings of the 6th International, Symposium on Signal Processing and Its Applications (ISSPA '01)*, vol. 1, pp. 343–346, Kuala Lumpur, Malaysia, August 2001.

[18] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[19] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

Mamatha. M. N. received her M.E. degree in Electronics from University of Bangalore in 1999. She received her B.E. degree in Instrumentation from Mysore University in 1993. Presently, she is working as an assistant professor in B. M. S. College of engineering, Visvesvaraya Technological University. She is presently doing a Ph. D. Research in Vinayaka Missions University, Salem,Tamilnadu. Her areas of interest are biomedical instrumentation and transducers. She has presented papers in national and International Conferences.

Dr. S. Ramachandran has wide academic as well as industrial experience for over 30 years, having worked as Professor in various engineering colleges as well as design engineer in industries. Prior to this, he has been with the Indian Institute of Technology, Madras. He has industrial and teaching experience, having worked both in India and USA, designing systems and teaching/guiding students and practicing engineers based on FPGAs and Microprocessors. His research interests include developing algorithms, architectures and implementations on FPGAs/ASICs for Video Processing, DSP applications, reconfigurable computing, open loop control systems, etc. He has a number of papers in International Journals and Conferences. He is the recipient of the Best Design Award at VLSI Design 2000, International Conference held at Calcutta, India and the Best Paper Award of the Session at WMSCI 2006, Orlando, Florida, USA. He has completed a video course on Digital VLSI System Design at the Indian Institute of Technology Madras, India for broadcast on TV by National Programme on Technology on Enhanced Learning (NPTEL). He has also written a book on Digital VLSI Systems Design, published by Springer Verlag, Netherlands (www.springer.com).

Appendix 1
The eigen values for different orientations and regions

Orientation	**Region of support**					
	W = 5x5			**W = 7x7**		
0^0	**1.0921**	**0.7432**	**1.4694**	**1.0897**	**0.7398**	**1.4729**
30^0	**1.0919**	**0.7429**	**1.4697**	**1.0893**	**0.7396**	**1.4728**
60^0	**1.0917**	**0.7426**	**1.4701**	**1.0890**	**0.7393**	**1.4730**
90^0	**1.0914**	**0.7422**	**1.4704**	**1.0887**	**0.7391**	**1.4730**
120^0	**1.0916**	**0.7425**	**1.4701**	**1.0889**	**0.7392**	**1.4730**
150^0	**1.0919**	**0.7428**	**1.4699**	**1.0894**	**0.7395**	**1.4732**
180^0	**1.0922**	**0.7433**	**1.4693**	**1.0898**	**0.7399**	**1.4729**