Name: Foo Tun Keat
Degree: Master of Engineering
Dept: Electrical and Computer Engineering
Thesis Title: Eye Gaze based Reading Detection

Abstract

In this thesis, we examine the possibility of determining whether a computer user is engaged in reading material on the computer monitor, or not, using image sequences from an ordinary camera. The challenges in determining whether reading is taking place is in differentiating this from the different activities that the user can perform at the computer, and being tolerant to user differences, different types of text and how they are displayed.

We have proposed an algorithm based on estimating gaze directions, achieving an average accuracy of 84.1% on 10 people. Changes in the gaze directions over a specified interval during reading are modelled using a set of Finite State Machines (FSMs). Gaze directions estimated over a finite time interval (window) are checked with respect to these FSMs to determine whether the interval represents reading. An aggregate of these determinations over successive time windows is finally used to infer whether reading is taking place.

Keywords: Reading detection, gaze tracking, user interest tracking, finite state machines, attentive systems, blink detection.

Eye Gaze based Reading Detection

FOO TUN KEAT

NATIONAL UNIVERSITY OF SINGAPORE

2004

Eye Gaze based Reading Detection

FOO TUN KEAT

(B. Eng. (Hons), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgement

I would like to dedicate this section to all who helped me in this project.

I must thank my main supervisor, A/P. Surendra Ranganath, for his guidance, invaluable time, and encouragement throughout the project. I also have to thank Dr N. Sriram, my co-supervisor, for his insights on the psychological aspect of the project. I would like to express my thanks to Prof Venkatesh who has been tremendous with his inputs and has greatly motivated me with his enthusiasm and energy.

I am grateful for the company of my fellow graduate students especially Meng Howe, Wengang, Feng Wei, Pingkun, Wangyong, Liping and Mr Francis Hoon, the technician of the Vision and Image Processing lab. A special thank to Bhaskar for his help and wonderful company, and also to the rest of the undergraduate students who have helped in the project.

I would like to thank the people who have helped me in one way or another, especially those who have volunteered for recordings.

Table of Contents

List of Figures

List of Tables

Summary

There has been an increasing interest in the fields of Human Computer Interfaces (HCI), wearable computing and affective computing. The emphasis is on building intelligent computers that are not only able to perform traditional computing but also capable of interacting with their users. In order for computers to be more effective in interacting with their users, they (the computers) would need to assess what the users are currently doing, anticipate their needs, gauge how are they feeling – frustrated, bored and so on.

In this thesis, we examine the possibility of determining whether a computer user is engaged in reading material on the computer monitor, or not, using image sequences from an ordinary camera. The challenges in determining whether reading is taking place is in differentiating this from the different activities that the user can perform at the computer, and being tolerant to user differences (e.g. language skills and age), different types of text and how they are displayed.

We have proposed an algorithm to determine whether the computer user is reading, achieving an average accuracy of 84.1% on 10 people. In the experiment, non-reading activities, such as playing computer games and watching video clips, which require different types of attention were included to test our algorithm.

The algorithm is based on estimating gaze directions. In order to obtain gaze direction, firstly, face localization is carried out by detecting skin regions and then detecting faces in them. Next, eye localization is performed on the detected face. Two methods are employed independently. The first method uses colour information of the iris and sclera while the second method relies on blink detection. When blinks are available, the second method is used, otherwise, the first method is employed. The method of using blinks is preferred over using colour information as it provides more accurate localisation of the eyes. Once the eyes are localized, they are tracked. One localized or tracked eye in each frame is then used to determine the person’s gaze direction. The gaze information is then used for reading detection. Changes in the gaze directions over a specified interval during reading are modelled using a set of Finite State Machines (FSMs). Gaze directions estimated over a finite time interval (window) are checked with respect to these FSMs to determine whether the interval represents reading. An aggregate of these determinations over successive time windows is finally used to infer whether reading is taking place.

Chapter 1

Introduction

1.1 Objective

The aim of the project is to differentiate whether the user is involved in reading or other activities. The user will be located in front of a computer monitor.

1.2 Utility of and Challenges in reading detection

As computer technology advances, we are moving on to building computers that are more interactive and personal. In order for computers to be more effective in interacting with their users, they (the computers) would need to assess what the users are currently doing, anticipate their needs, gauge how are they feeling – frustrated, bored and so on. Coupled with cheaper and more powerful video cameras, there has been a growing interest in building such computers or systems as can be seen in the increase of interest in the fields of Human Computer Interfaces (HCI), wearable computing, affective computing and many others. One possible use is in HCI, when creating adaptive peripheral displays, it would be useful to know whether the user is reading [Maglio2000]. When the user is reading, the display would be as quiet and non-disturbing as possible.

Before moving on, perhaps it would be good to establish common grounds. To different people, reading could take on different meaning. So, what is reading? True reading requires integrating the ability to break the code and the ability to understand the meaning intended by the writer. In addition, the reader must maintain interaction with personal background (schema), the context of the text, and the author’s intention.

As we can see, reading is a rather complicated process. It only seems simple because it has become a habitual activity. Indeed, reading detection and reading speed estimation present a few challenges. First and foremost, the direct input, gaze direction, needs to be obtained and the gaze directions need to be analysed to determine whether reading has taken place. This analysis is complicated by the numerous activities that the user can perform while at the computer. Some of the activities include scanning (for information), watching a video clip, playing games and so on. In addition, several factors such as individual differences (e.g. language skills, intelligence, age, etc), text difficulty, differences in display such as font face, font size, have all to be taken into consideration. All these increase the difficulty of reading detection.

1.3 Design Overview

During reading, the eye gaze makes periodic movements from left to right and back. These periodic changes in the gaze direction are modeled and used for reading detection. In order to determine the gaze direction, the eyes have to be detected and localized. Once the eyes are localized, they are tracked in subsequent frames. To reduce search times, frontal faces are first detected and localized.

The algorithm flow is shown in Figure 1.1. The input to the application is a sequence of (near) frontal face images. For the first image, face detection is performed on the whole image to locate the position of a face, assuming that there is only one user. Once the face is located, detection of the eyes is carried out in the upper half of the face. The detected eyes are fit to a geometric model. If eye detection is successful, eye tracking is implemented on subsequent frames. The tracked eyes are also fit to the geometric model to confirm the validity of the tracking. If the tracked eyes fit the model, the gaze direction is estimated. If not, the whole process of face and eyes detection has to be repeated on the current frame.

The gaze direction sequence estimated from each frame is input to the reading detection module, which uses data from a sliding window of 6 seconds to detect whether reading is taking place in that interval. The sliding window is offset by 1 second intervals to determine reading. The results over these 6 sliding windows are aggregated to make a smoothed decision.

The system is based on the assumption that there is only one person or one face in the image. During recording, the subjects were told to make themselves comfortable and read the material on the screen as they normally would. However, the subjects were requested not to rub their eyes or cover their face regions with their hands, or move their heads so much that they are out of the video capture area. Care was taken to ensure that the subjects were not distracted from their reading.

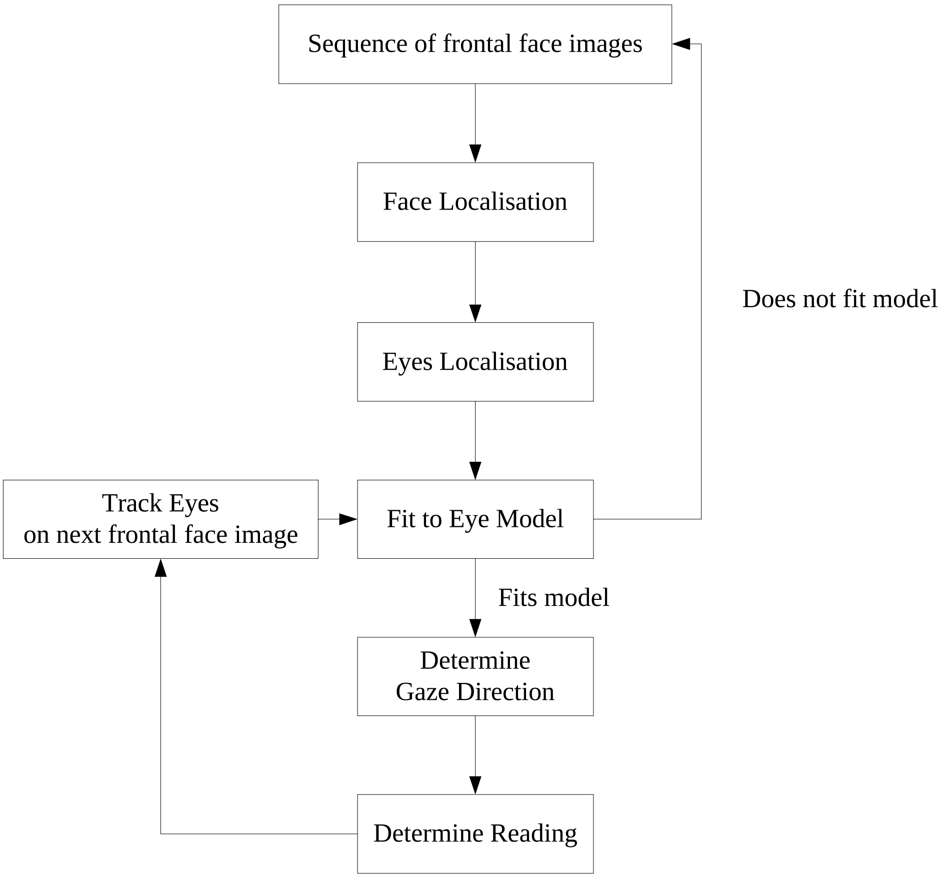


Figure 1.1: System Overview

1.4 Literature survey

Little research has been carried out in the field of Pattern Recognition for reading detection to date. However, observations on a reader's eye movements were made as far back as 1879 by Professor Emile Javal of the University of Paris. In the decades from the 1930s to the late '60s, reading research was dominated by views that put little focus on how the eyes functioned in reading. Most of the research in the field of Psychology in the last few decades (1980s and '90s) focused more on studying eye movements within a set of assumptions arising from that view of the reading process. Summaries of the findings of eye-movement research are now being used to argue for a word-recognition, as opposed to a meaning-construction, view of reading (e.g., [Adams1995]).

Though research on eye movements during reading is prevalent, apparently, the only attempt to detect whether a user is reading or engaged in any other possible user activities at the computer was carried out by Campbell, et. al. [Campbell2001]. They used infra-red cameras to track eye movements. Using pooled data of these movements, they examined changes in them for determining whether reading or other activities such as scanning is taking place. Their experiments showed that their algorithm is robust to noise, individual differences and variations in text difficulty. In addition, they claimed that the pooled evidence algorithm they used has a high (nearly 100%) accuracy rate. However, the experiments were carried out on only 4 or 5 subjects who were constrained to rest their heads on a chin rest.

In our works (Figure 1.1), the first step to reading detection involves face localisation. A common method employed in the process of face localisation is skin colour segmentation. Skin colour segmentation is advantageous in that it is fast and orientation invariant. Examples of systems that make use of skin colour detection are [Menser2000], [Jones1999] and [Yang1997]. In this project, skin colour segmentation is carried out using the method proposed in [Yang1997].

Skin colour segmentation alone is not sufficient for determining the presence of a face. It is helpful in quickly finding possible face regions (where the skin is detected). Several face detectors proposed in the literature could be used to determine face positions in the skin regions. Yang and Huang [Yang1994] presented a hierarchical knowledge-based system for face detection in complex background. However, the rules for discrimination are not necessarily optimal since their structure is fixed beforehand. In [Moghaddam1995], Moghaddam and Pentland employed maximum likelihood estimation on feature vectors obtained from eigenspace decomposition to detect faces. However, principal component analysis does not maximise discrimination. Sung and Poggio [Sung1994] and Rowley et. al. [Rowley1996] reported systems which utilises neural networks which claim very good performance but these systems are extremely computational expensive in both the training and testing procedures. The classifier used for face detection in this project is based on [Colmenarez1997] which reported comparable performance to neural network based face detectors in [Sung1994] and [Rowley1996] but it is much less computationally expensive.

The next step of the design involves eye localisation. Several methods have been proposed, for example in [Fang1994], a novel filter is used for eye localisation. The basic idea is to use the relative high horizontal contrast density determination, facial geometry reasoning, and eye position determination. Another method [Benn1997] proposes to use a gradient decomposed Hough Transform to embody the natural concentricity of the eye region in a peak reinforcement scheme. In this project, eye localisation is performed by combining 2 methods. One uses colour information [Betke2000] and the other uses motion during a blink. Eyes are more accurately localized using involuntary blink information as compared to using colour information which is more varied across different races. However, eye localization using colour information is useful especially in the absence of blinks because it can be carried out any time. The literature contains a few methods that have been devised for blink detection. In [Yano1999], frame differencing is used for blink detection. Al-Qayedi and Clark [Qayedi2000] track features about the eyes and infer blinks through detection of changes in the eye shape. Smith et al. [Smith2000] try to differentiate between occlusion of the eyes (due to rotation of the head) and blinking. The subject's sclera is detected using intensity information to indicate whether the eyes are open or closed (i.e., a blink is occurring). Black et. al. [Black1998] detect blink using optical flow but the system restricts motion of the subject and needs "near frontal" views in order to be effective. Grauman, et al. [Grauman2001] report a success rate of 96.5% in almost real-time. Here, frame differencing is initially performed to obtain motion regions. Then a pre-trained "open-eye" template is selected for the computation of the Mahalanobis distance measure which is used to retain the best pair of blobs as candidates for eyes. The eyes are tracked and correlation scores between the actual eye and the corresponding "closed-eye" template are used to detect blinks. The blink detection method used in this project combines the use of temporal differencing and optical flow computation based on [Black1996]. Temporal differencing allows us to quickly identify motion regions and optical flow computation is used to further ascertain that the motion is due to the closing (downwards) and opening (upwards) of the eyelids during a blink in localised regions. This speeds up the computations.

When eyes are localised, they are tracked. Several trackers are available in the literature, e.g. Kalman Filter [Rosales1998] and CONDENSATION [Isard1996]. In this project, we have chosen to use the Kanade-Lucas-Tomasi (KLT) tracker based on [Tomasi1991], [Lucas1981] and [Birchfield1996] as it is found to serve our needs and easily implemented as source codes are made available in the public domain by Stan Birchfield.

As the eyes are localised or tracked, gaze estimation is performed. In the literature, infra-red cameras are often used to obtain very accurate gaze estimation. However, we are interested in methods that used images captured by ordinary video cameras. Some of the approaches include neural networks [Schiele1995], [Varchmin1997] and [Baluja1993], morphable models [Rikert1998], and self-organizing gray-scale units [Betke1999]. Gee and Cipolla [Gee1994] explore the underlying geometric constraints. In this project, gaze is estimated using neural network based methods as proposed in [Baluja1993] and [Varchmin1997].

Our aim is to create a system to demonstrate reading detection using ordinary video cameras. The system should be as non-intrusive as possible so that the users are not distracted during their activities. In addition, the system should be robust to individual differences as well as the difficulty of contents of text.

1.5 Experimental Setup

Figure 1.2 shows the experimental setup. The camera is placed between the subject and monitor, tilted up at an angle. The monitor is raised to prevent the camera from occluding it.

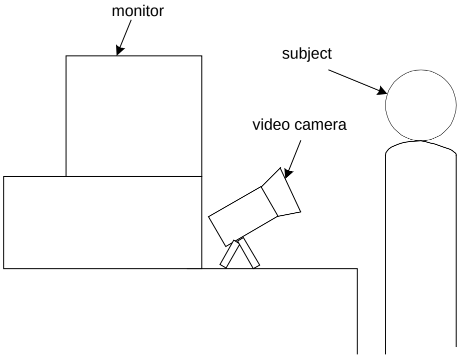


Figure 1.2: Experimental Setup

The sequences were captured using a Panasonic 3 CCD digital video camera at a resolution of 768x576 and at a frame rate of 25 frames/sec. They were transferred to a computer via firewire interface and then converted to portable pixelmap format for processing. The conversion was done using a video editing software, VideoMach. Unless otherwise stated, the system is run on a Pentium 4, 1.7GHz, 256 MB RAM machine.

1.6 Organisation of Thesis

The thesis is organised into 6 chapters. Face localization, eye localization and tracking, gaze direction estimation are discussed in detail, with results for each individual module, in Chapters 2, 3 and 4 respectively. In Chapter 5, the algorithm for reading detection is discussed. Results for the entire system are also presented in the same chapter. Possible future work is discussed and the thesis is concluded in Chapter 6.

Chapter 2

Face Localization

2.1 Background

This chapter discusses how a face in the scene is detected. Automatic detection of the human face forms an essential ingredient in the analysis of human behaviour. Face detection is a challenging problem as there are numerous variations which have to be taken into consideration. They include people with different skin colour, differences in illumination, complexity of the scene background, presence of glasses and so on.

In the next section, we discuss the theory behind the method of classification used for face detection before moving on to how the data is prepared to train the classifier. This chapter will be concluded with discussion on the results obtained.

2.2 Algorithm for Face Localisation

In this project, faces are localized in 2 stages. In the first stage, skin colour segmentation is performed on the whole image to quickly identify possible locations of faces. More details on skin colour segmentation are given in Section 2.2.1. “Skin blobs” are then passed through a classifier built to identify faces. The classifier is discussed in depth in Section 2.2.2.

2.2.1 Skin Colour Segmentation

Skin colour segmentation provides an efficient means of obtaining possible face region in an image. The algorithm used for skin colour segmentation in this project is described in Yang, et. al. [Yang1997]. A normalised chromatic colour space, viz. $\frac{R}{R+G+B}$ and $\frac{G}{R+G+B}$, is used as it is found that skin colours of different people are less variant in this space. Also Yang, et al. assert that in the normalised chromatic colour space, a bivariate normal distribution can be used to characterise skin colour distributions. Thus, pixels with normalised chromatic colours that fall within the bivariate normal distribution will be classified as skin pixel.

2.2.2 Information-Based Maximum Discrimination Classifier for Face Detection

The classifier for face detection maximises the discrimination between positive (faces) and negative (non-faces) examples in a training set [Colmenarez1997]. A Markov process is used to model the face and non-face patterns and estimate their probability distributions using training data. Kullback relative information or Kullback divergence is used to measure the “distance” between the two probability distribution models. The Markov process that optimizes the information-based discrimination between face model and non-face model or the process that achieves the least “distance” between the models is identified. The detection process first computes the likelihood ratio of an observation using the probability models obtained from the learning process. The likelihood is then compared to a fixed threshold for determining whether it is a face.

Let X^n be a random process, and P_{X^n} and M_{X^n} be two probability functions for X^n describing the face and non-face classes, respectively. The divergence of P with respect to M is then defined as:

$$H_{P\|M} = \sum_{X^n} P_{X^n} \ln \frac{P_{X^n}}{M_{X^n}} \qquad (4.1)$$

Let S = {$s_i \in$ [1,n], i = 1, 2, … ,n} be a list of indices such that s_i ¹ s_j for i ¹ j. If X^n can be modelled as a kth order Markov process, then

$$\mathrm{P}(X_{Sn} \mid X_{S1}, \ldots , X_{Sn-1}) = \mathrm{P}(X_{Sn} \mid X_{Sn-k}, \ldots , X_{Sn-1}), \qquad (4.2)$$

and the divergence of P with respect to M is obtained as:

$$H_{P\|M}\left(X^n(S)\right) = \sum_{i=1}^{k} H_{P\|M}\left(X_{S_i} \parallel X_{S_1},\ldots,X_{S_{i-1}}\right) + \sum_{i=k+1}^{n} H_{P\|M}\left(X_{S_i} \parallel X_{S_{i-k}},\ldots,X_{S_{i-1}}\right) \qquad (4.3)$$

We want to find an optimal set of indices S* such that

$$H_{P\|M}\left(X^n(S*)\right) \geq H_{P\|M}\left(X^n(S)\right) \qquad \forall S \tag{4.4}$$

i.e., the set that maximises the discrimination (Kullback divergence) between the face and non-face classes.

Once S* is found, the likelihood ratio given in equation (4.5) is the one that optimizes the correct-answer-false-alarm trade off of the training set and may be expected to work well with other data sets.

$$L(X^n(S^*)) = \frac{P\left(X^n(S^*)\right)}{M\left(X^n(S^*)\right)} \tag{4.5}$$

In order to find S* in equation (4.4), we only consider a 1st order Markov process, and the greedy algorithm of Kruskal [Cormen1990] is used to obtain sub-optimal but good results. In simplifying equation (4.3) for a 1st order Markov process, we need to find the divergence of the two probability density functions for individual pixels and pairs of pixels. For each pair of pixels X_i and X_j, the divergence of the two probability density functions is computed as follows:

$$H(X_i \parallel X_j) = \sum_{\|A\|^2} P\left(X_i, X_j\right) \ln \frac{P\left(X_i \mid X_j\right)}{M\left(X_i \mid X_j\right)} \tag{4.6}$$

where ||A|| is the number of possible values that the pixels can take (the images are pre-processed to limit the possible gray values to a small number).

For each pixel, H(X$_i$) is computed as follows:

$$H(X_i) = \sum_{\|A\|^2} P(X_i)\ln\frac{P(X_i)}{M(X_i)} \tag{4.7}$$

Finally, using equations (4.6) and (4.7), the indices S* (S$_1$, S$_2$, ..., S$_n$) which maximise the total divergence of the Markov process is found using equation (4.8).

$$H(X^n(S)) = H(X_{S_1}) + \sum_{i=2}^{n} H(X_{s_i} \parallel X_{s_{i-1}}) \tag{4.8}$$

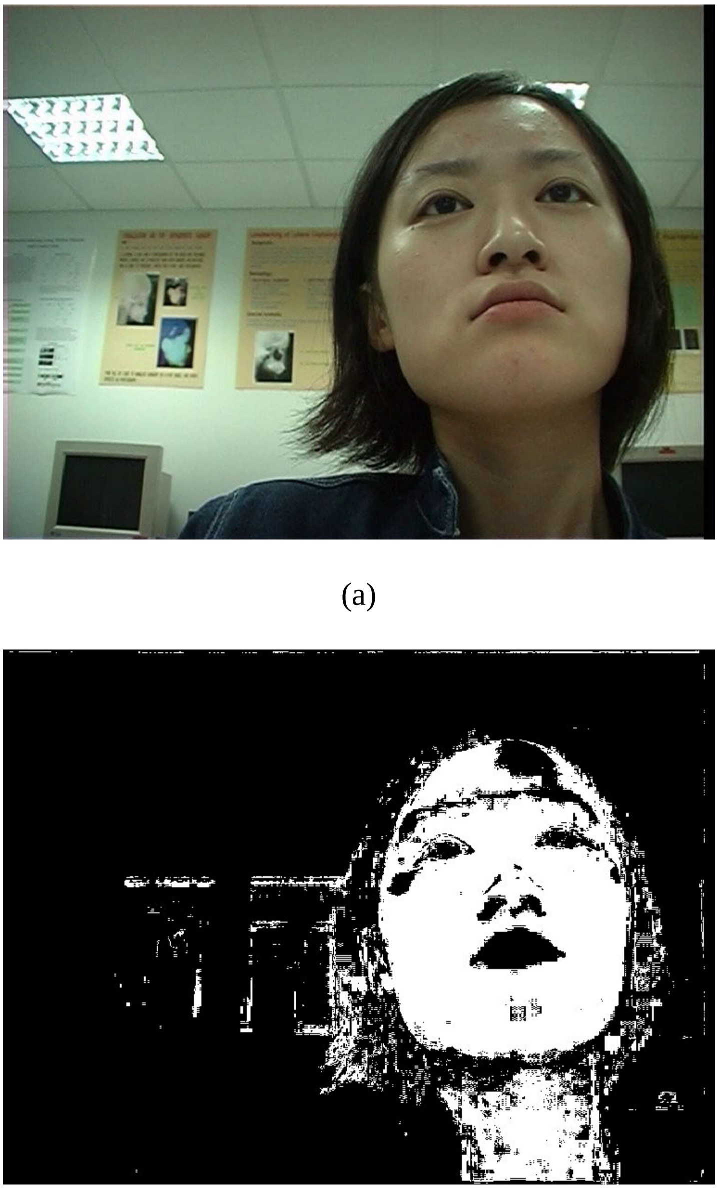
For each pair of pixels in the ensemble of images in the training set, a joint histogram is used to estimate the probability density function for both classes, faces and non-faces.

2.3 Data Preparation

For skin colour segmentation, skin colour pixels from training images are used for the computation of the parameters, viz. mean and covariance matrix, of the bivariate normal distribution. An example of detected skin colour is shown in Figure 2.1.

For training the face detector based on the information-based maximum discrimination classifier, faces are extracted manually from images captured using a digital camera.

The outer eye corners are used as normalizing references, to resample the extracted faces to a chosen window size, 11x11.



(a)

(b)

Figure 2.1: Skin colour detection. (a) Original image, (b) Detected skin in white.

In order to increase the size of the training set, each of these images (before being down-sampled to 11x11) is rotated in-plane with angles of ±6°, as shown in Figure 2.2

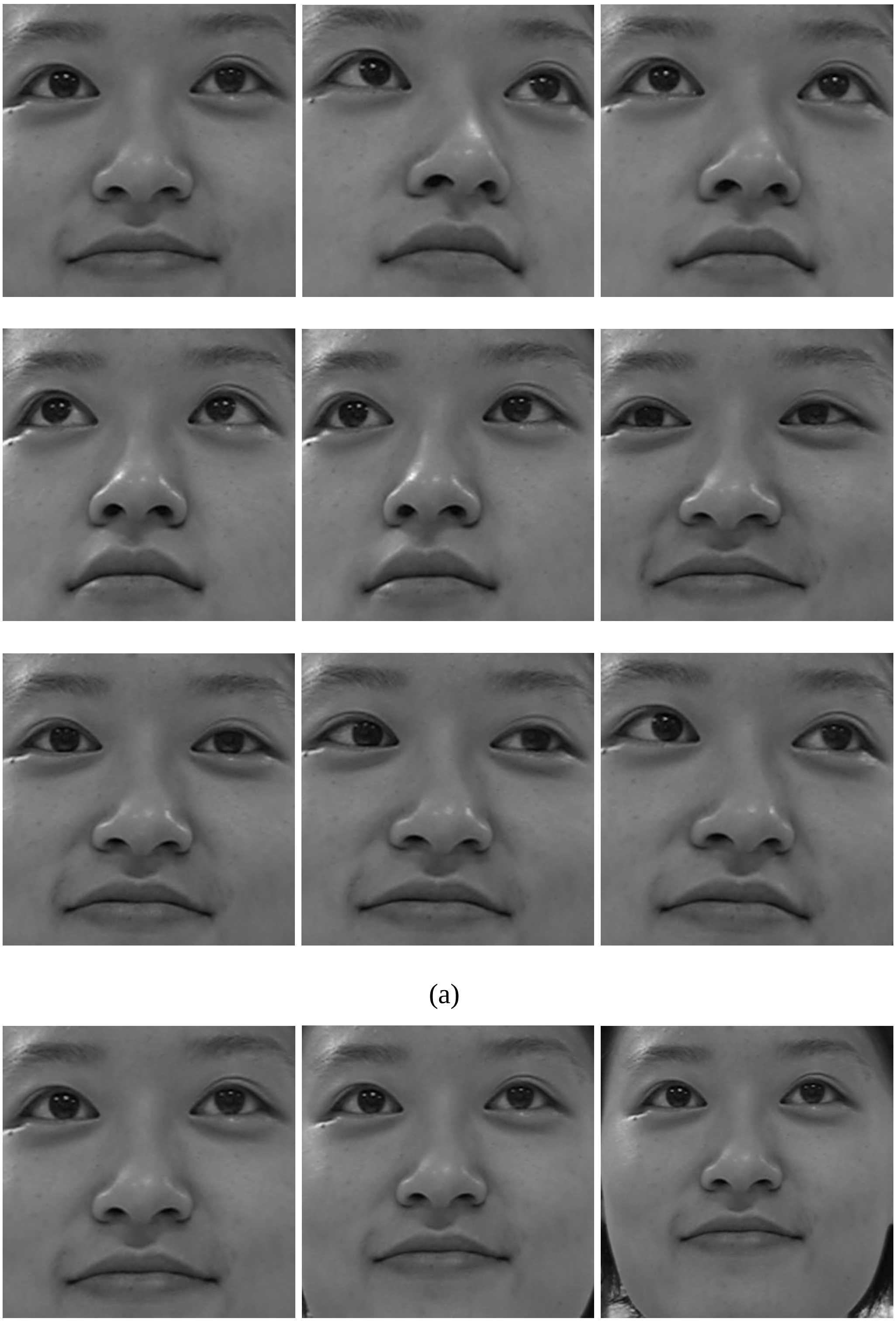
(a), and rescaled using scales of 0.9 and 0.81, as shown in Figure 2.2 (b), to generate another 9 training images. Histogram equalization is performed on each training image for lighting correction and the image is quantised into 4 levels as shown in Figure 2.3. Four levels of pixel quantisation are chosen because it gives reasonably good performance while not being too computationally expensive.

Similar processing is also performed on non-face images. Non-face images are extracted from natural scenes such as satellite images. Some examples are shown in Figure 2.4. In order to increase the discriminatory power of the classifier, images are also extracted which contain part of the face.

2.4 Implementation

Skin colour segmentation is first performed on the acquired image of size 768x576 to obtain skin blobs. Assuming that there is only one face in the image, the biggest blob is passed to the classifier for face detection.

To detect faces of different sizes, a multi-scale search has to be carried out. At each scale, a window of size 11x11 is translated over the whole image and a likelihood value is computed (see equation (4.5)) for each window location and compared with a threshold to determine whether a face is present. In our application context, observed face sizes of faces of interest range from 140x135 to 340x340, and thus, 4 scales within this range are used to speed up the detection process.



(a)

(b)

Figure 2.2: Some samples of face training images. (a) Images are taken from a person looking at 9 different spots: centre, 4 corners and the 4 centres of the sides of the monitor, (b) Scales of 1.0, 0.9 and 0.81 are used to get different face sizes.

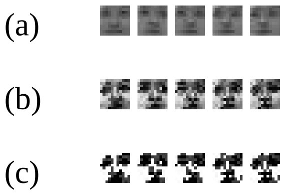


Figure 2.3: Samples of pre-processed training images. (a) Original 11x11 images, (b) Corresponding images, histogram equalized, (c) Images in (b) quantised to 4 levels.

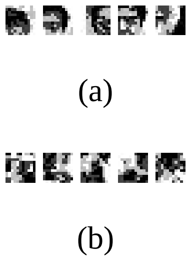


Figure 2.4: Samples of negative training images, histogram equalised and quantised to 4 levels. (a) Parts of face, (b) Other images, such as satellite images.

2.4.1 Selecting the threshold to determine presence of a face

The likelihood values in equation (4.5) are obtained for all the samples, with positive samples differentiated from negative ones. The statistics of the likelihood values are given in Table 2.1. Graphs of false alarm rate and missed detection rate are shown in Figure 2.5. From Figure 2.5, a threshold of 3.3 is selected to have a low false alarm rate and a low missed detection rate.

Table 2.1 Statistics of the likelihood values

	Mean	Standard Deviation
Face	3.87	0.72
Non-face	2.06	1.02

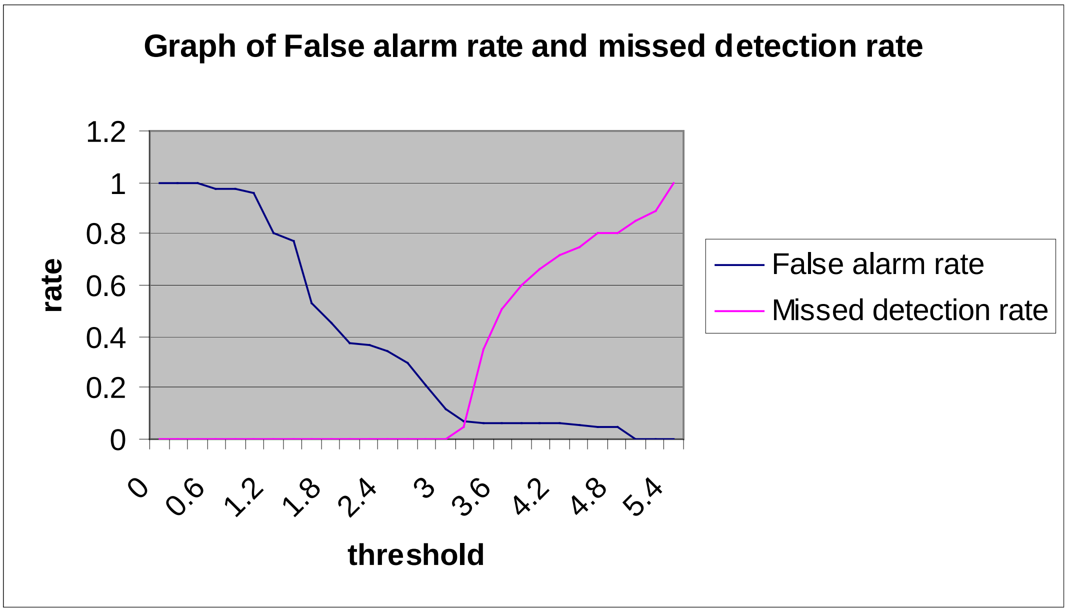


Figure 2.5: Graphs of false alarm rate and missed detection rate.

2.4.2 Bootstrapping

For classifying faces from non-faces, it is difficult to get a good representation of the non-face class as it would include all other images that are not faces. Hence, bootstrapping is used during the training phase, where false positives obtained are added to the negative example set at the end of the training round and the classifier is re-trained. This process is repeated a few times to increase the discriminatory power of the classifier.

2.4.3 Determining the final location of the face

As a multi-scale search is performed to detect faces of different sizes at different positions, it is highly possible that we will get multiple detections around a face, from which the final position of the face needs to be determined. To resolve this, the mean location of all the centres of the face candidates is first obtained. Face candidates whose centres are beyond a certain acceptable distance from the mean centre location will be rejected. The remaining face candidates is then used to mark the final location, having top left-hand corner coordinates (x_{UL},y_{UL}) and bottom right-hand corner (x_{BR}, y_{BR}), of the detected face as follows:

$$x_{UL} = \min(x_i),\ i\in \text{ remaining face candidates}$$
$$y_{UL} = \min(y_i),\ i\in \text{ remaining face candidates}$$
$$x_{BR} = \max(x_i),\ i\in \text{ remaining face candidates}$$
$$y_{BR} = \max(y_i),\ i\in \text{ remaining face candidates}$$

2.5 Results and Discussion

We computed the likelihood ratio of the training images using the probability models obtained from the learning process, and compared it to a fixed threshold to make the decision. The threshold is set at 4 in order to have a high detection rate while keeping false alarm rate reasonably low.

A candidate obtained from the multi-resolution search is considered a correct detection if it falls in the scale range considered, and the error in the position of the face (with respect to the ground truth) is less than 10 percent of the size of the face. All other candidates are marked as false positives. The classifier was tested on 15 people, with 9 poses each. In this experiment, seven scales were used to test the robustness of the classifier. The results are tabulated in Table 2.2. A high detection rate of 97.8% is achieved with a corresponding low false alarm rate of 0.28%. In the implementation of the system, to speed up the detection process, only 4 different scales are used as the face sizes are expected to range from about 140x135 to 340x340. Figure 2.6 shows an example of how the face candidates are combined to give a final localization of the face. Skin detection module takes about 0.7 seconds to process a 768x576 image. Using 4 scales, the amount of time required to process a 768x576 image is about 11.4 seconds.

A false alarm rate of 0.28% is higher than that reported in the literature, e.g. [Colmenarez1997] which reported about 0.048% on the database it tested. However, the reported false alarm rate is reasonably low and does not pose a major problem for detecting faces in our project.

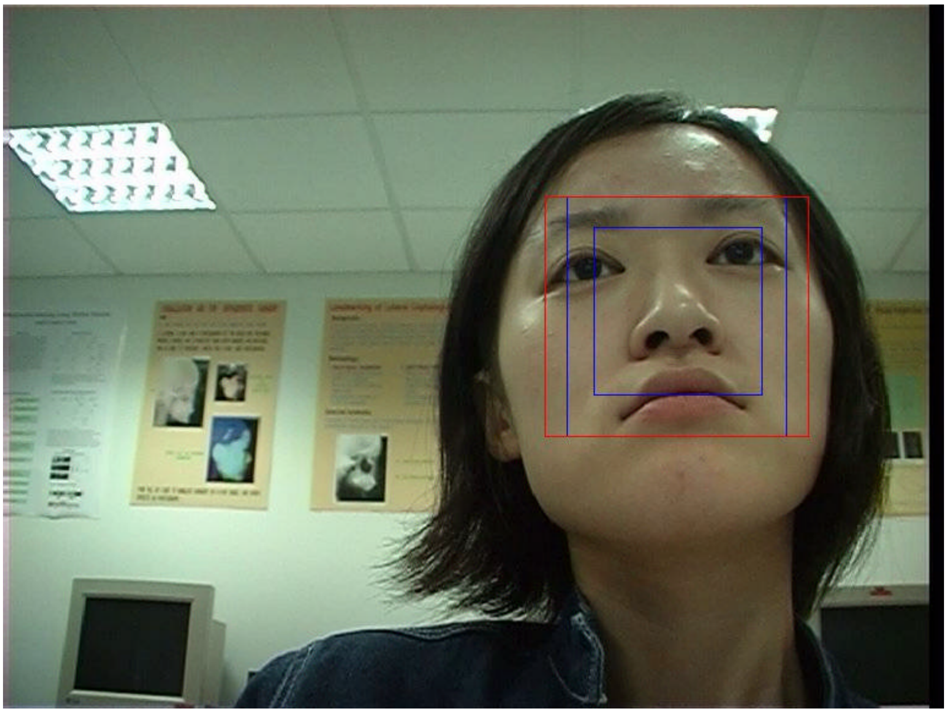


Figure 2.6: Combining face candidates for face localization. The 3 initial face candidates are bounded by blue boxes. The final face localization is bounded using red box as shown.

Table 2.2 Classification results for face detection.

Detected Faces	**Detection Rate**	**False Faces/Total Windows Tested**	**False Alarm Rate**
132/135	97.8%	596/216126	0.28%

2.6 Conclusion

In conclusion, skin colour detection gives reasonably smaller search areas for face detection. Skin detection takes about 0.7 seconds to process a 768x576 image. The visual learning technique, Information-Based Maximum Discrimination, employed for face detection is able to optimize the discrimination between faces and non-faces. To further speed up the detection process, only 4 different scales are used as the face sizes are expected to range from about 140x135 to 340x340. The amount of time required to process a 768x576 image is about 11.4 seconds. The face detector has a detection rate of 97.8% and a false positive alarm rate of 0.28%.

Chapter 3

Eye Localization and Tracking

3.1 Introduction

Eye localization is useful in applications such as normalization of the face, gaze-based human-computer interface, and security systems using the iris for identification. Eye localization is implemented by combining 2 independent methods. The first uses colour information of the irises and sclera of the eyes while the second uses motion associated with eye blinking. Eye localization is discussed in Section 3.2. Once the eyes are localised, they are tracked using a tracking mechanism discussed in Section 3.3. The tracker is initialised on the areas of the eyes, i.e., feature points around the eyes are chosen and then subsequently tracked. If more than 20% of these points are lost during tracking, the tracker is disabled and a check is made to determine whether the points are lost due to sudden eye-ball movement or other movements. First, motion analysis, which includes frame differencing, thresholding and connected component analysis, is performed to obtain motion blobs. Unsuitable blobs are removed (refer to Section 3.4 for suitability of blobs for eyes) and optical flow computation is carried out. From optical flow analysis, if the motion is due to eye-ball movement, the tracker is reinitialised and tracking resumes. If it is due to other movements, the whole process of eye localization is repeated. As a confirmatory step, the localised and tracked eyes are fit to a geometric model to ascertain the correctness of the eye localization. This is discussed in Section 3.4. The entire process of eye localization is shown in Figure 3.1.

3.2 Eye Localization

The eyes are localized by a combination of 2 methods. The first uses color information of human iris and sclera. The second uses motion information during blinks. From our experiments, we found that eyes are more accurately localized using blink information as compared to using color information of human iris and sclera. Thus, when blinks are available, they are used to mark the positions of the eyes. Otherwise, eye localization is done using colour information because it can be carried out any time. When this method is used, skin and face detection is required. Once the eyes are localized, they are tracked in subsequent frames. Details are found in Section 3.2.1. The second method checks for downward and upward motions of the eyelids that occur during blinks. If this is found, then blinking is deemed to have taken place and the appropriate motion regions are marked as eye regions. Details are given in Section 3.2.2.

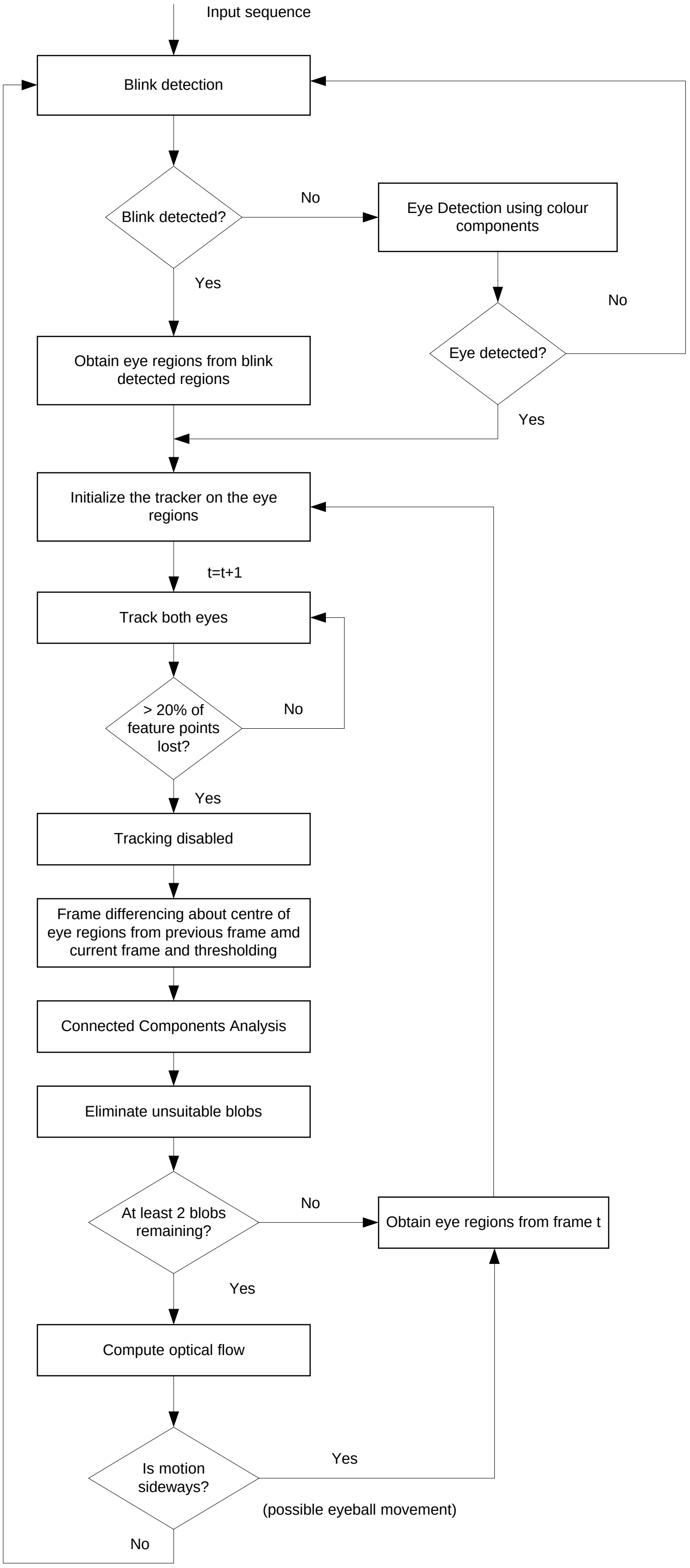

Input sequence
Blink detection
Blink detected?
No
Eye Detection using colour components
Yes
No
Obtain eye regions from blink detected regions
Eye detected?
Yes
Initialize the tracker on the eye regions
t=t+1
Track both eyes
> 20% of feature points lost?
No
Yes
Tracking disabled
Frame differencing about centre of eye regions from previous frame amd current frame and thresholding
Connected Components Analysis
Eliminate unsuitable blobs
At least 2 blobs remaining?
No
Obtain eye regions from frame t
Yes
Compute optical flow
Is motion sideways?
Yes
(possible eyeball movement)
No

Figure 3.1: Flowchart depicting how eyes are localised and tracked.

3.2.1 Eye Localization Using Colour Information

When a face is detected, only the upper half is searched for eyes. The eyes are found by identifying “sclera” and “iris” pixels. The identification is carried out in 2 steps. In the first step, the R-G and G-B components of each pixel are considered based on [Metke2000]. Some statistics on the 2 components, collected from 20 people, is given in Table 3.1. Histograms for R-G and G-B are shown in Figures 3.2 and 3.3, respectively. Based on these statistics, the thresholds are found to discriminate between iris/sclera and other pixels. Using the gaussian fit on these statistics, the thresholds for R-G and G-B are selected. Here, a pixel is considered as iris/sclera after this first test if its R-G value is less than 8 and its G-B value is less than 19.

After this test, most of the non-eye pixels would have been eliminated. However, sometimes, a few non-eye pixels remain and they hinder eye localization as shown in Figure 3.4 (b). Thus, another test is performed to remove these pixels that hinder eye localization.

In the second step, an analysis of the colour components of pixels that remain is carried out. It is found that using the components of Cr and Cb in the YCrCb colour space gives the best separation of iris and sclera pixels from the non-eye ones. The distribution of Cr and Cb of those pixels that are not removed in the first test is shown in Figure 3.5.

From the distribution, 2 linear equations (as shown in Figure 3.5b) are defined to identify 2 regions for iris/sclera and non-eye. The equations are:

1) Cr-Cb = -3, and

2) Cr + 4*Cb = 22.

After the second test, the eye is correctly localized as shown in Figure 3.4 (c).

Table 3.1 Statistics of` iris and sclera's colour components.

	Iris/Sclera		**Non-eye**	
	Mean	**Variance**	**Mean**	**Variance**
R-G	-0.488	48.34	10.54	66.02
G-B	14.11	20.06	21.22	56.43
Total Number of pixels used	0.189 x 10^6		5.77 x 10^6	

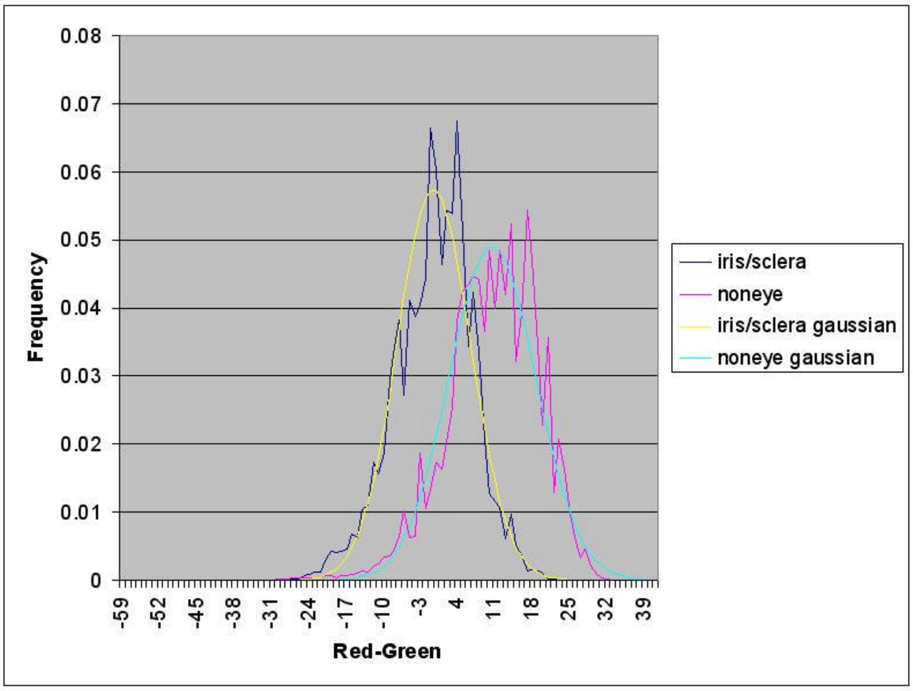


Figure 3.2: Graph of frequency vs R-G. Gaussian refers to the gaussian fit to the distribution.

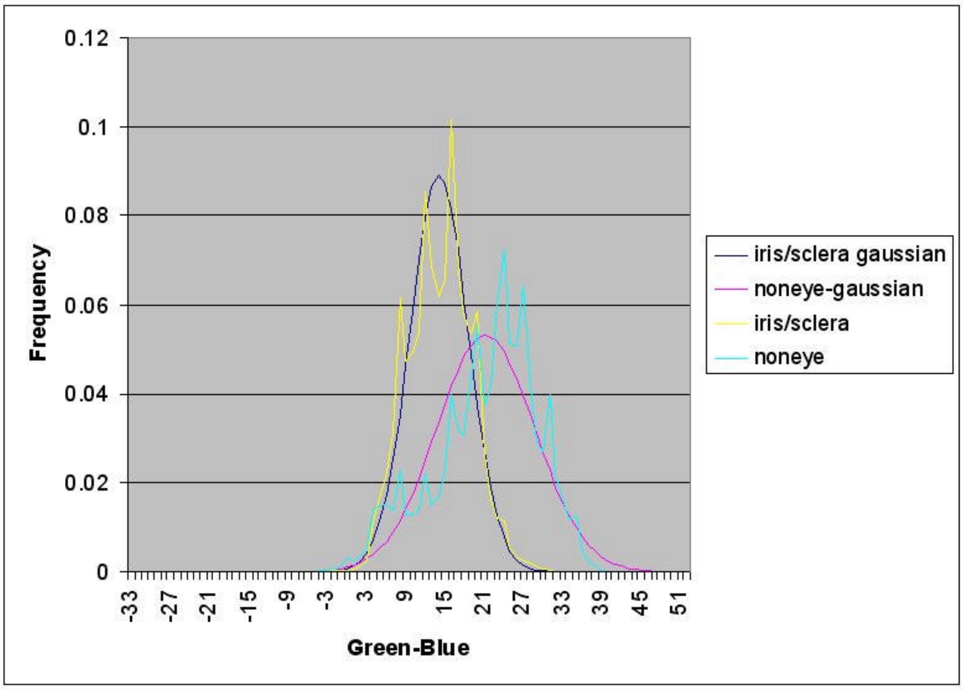
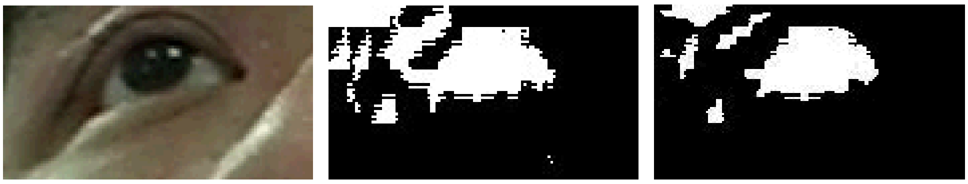
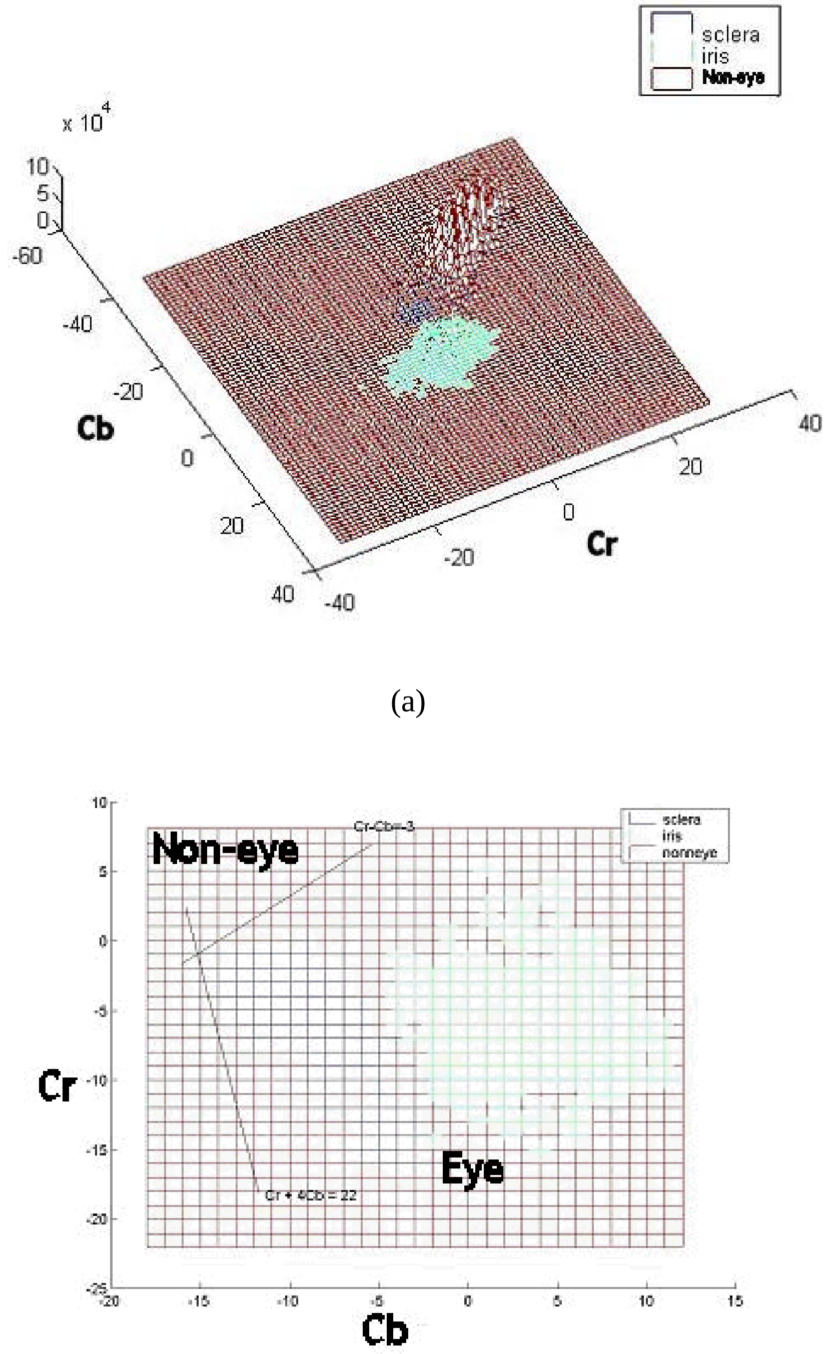


Figure 3.3: Graph of frequency vs G-B. Gaussian refers to the gaussian fit to the distribution.



(a) *(b)* *(c)*

Figure 3.4: A sample where the eye is not properly segmented after the first test and can be properly localised after the second test. (a) Original image, (b) Binary image after the first test; white pixels indicate "eye" pixels, (c) Binary image after the second test, the eye can now be properly localised.



(a)

(b)

Figure 3.5: Histogram of Cr and Cb of remaining pixels after the first test. (a) Three-dimensional view, (b) Two-dimensional view.

3.2.2 Blink Detection for Eye Localization

Detecting blinks enables us to determine eye regions. The process of blink detection involves (a) temporal frame differencing, and (b) computation of optical flow. The flowchart is shown in Figure 3.6. Optical flow field is computed using the algorithm in [Black1996]. The steps in the algorithm for eye localization using blink detection are given below:

Step (i) Obtain locations of possible motion using frame difference between successive frames.

Step (ii) Threshold the frame difference at 15 and obtain blobs using morphological operations and connected components analysis.

Step (iii) Remove unsuitable blobs i.e. remove blobs which are either too big or too small or have incorrect width to height ratios to be considered as eye candidates.

Step (iv) If there are no blobs remaining, repeat (i) to (iii) on subsequent frames until at least 2 blobs remain. Mark the positions of these blobs.

Step (v) Compute optical flow field around the vicinity of the remaining blobs.

Step (vi) Ascertain dominant direction of motion. If the dominant motion is downwards and its magnitude is greater than 3, then the position of the blob(s) is noted. This denotes eye closure during a blink. If the motion is not downwards then repeat (i) to (vi). Steps (vii) onwards are used for detecting subsequent eye opening phase of the blink.

Step (vii) Repeat steps (i) to (iii) to find motion blobs.

Step (viii) Discard blobs that are not situated near the location of the blobs which had downward motion.

Step (ix) Compute optical flow to ascertain if the dominant motion is upwards if there are at least 2 blobs remaining. Otherwise, increment count and go to Step (vii). If count is greater than 3, it means that no corresponding upward motion is detected and thus no blink is deemed to have occurred. Thus, jump to Step (i) to restart the process of blink detection. The threshold for count is 3 because from our observations, involuntary blinks in general do not last more than 5 frames.

Step (x) If the dominant motion is upwards and its magnitude is greater than 3, then classify the frames beginning from the frame where downward motion was detected to the frame where upward motion is detected as blink frames. Go to Step (xi). Otherwise, increment the count and go to Step (vii).

Step (xi) Mark the positions of the bounding boxes of eye regions.

In Step (ii), a threshold of 15 is used because it was found to give good segmentation of motion regions from non-motion ones. The criteria for pairs of suitable blobs are given in Section 3.4. In Steps (vi) and (ix), based on Table 3.2, a threshold of 3 for the flow magnitude is used. Table 3.2 shows the statistics for the magnitude of the optical flow field for different types of movements. Both the magnitude and especially, the direction of the flow vectors from the computation of optical flow are used to differentiate vertical eyelid movements during blinks from movement of the eyeball (predominantly horizontal) and horizontal head movements (Figure 3.9). As for vertical head movements, they result in blobs that can be eliminated based on size, size ratio (please refer to Section 3.4) and the average magnitude of the velocity vectors.

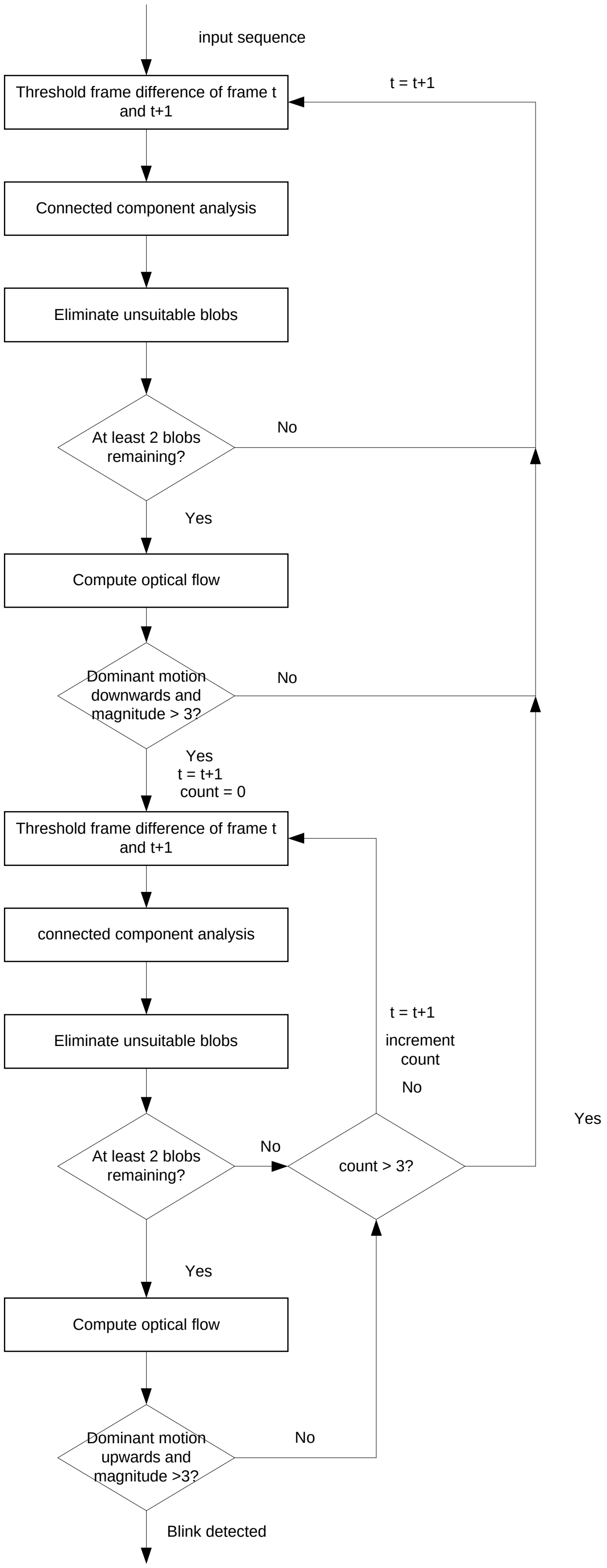
From Table 3.2, in the case of head motion, its average magnitude is 2.88 compared to 3.05 and 4.80 for upward and downward eyelid movements respectively. Figure 3.10 shows an example of vertical head movement and its optical flow field.

Table 3.2 Mean and variance of magnitude of optical flow vectors based on 50 blinks from 10 different people.

	Mean	**Variance**
Blinking (closing)	4.80	2.60
Blinking (opening)	-3.05	0.14
Vertical head movement (absolute)	2.88	0.32
Horizontal head movement (absolute)	0.19	0.01
Eyeball movement	9.74	9.14

3.3 Eye Tracking

After having located the eyes, we track them using the Kanade-Lucas-Tomasi (KLT) tracker [Tomasi1991], [Lucas1981], [Birchfield1996]. The combined process of localizing and tracking of the eyes is shown in Figure 3.1. The KLT tracker classifies a tracked feature as good or bad according to the residual of the match between a window around the feature in the previous and current frame; if the residual exceeds a fixed threshold, the feature is considered lost. For feature selection, the selection criterion is based directly on the definition of the tracking algorithm, which expresses how well a feature (or rather the window around the feature) can be tracked. Details can be found in refer to Section 4 of [Tomasi1991].



input sequence
Threshold frame difference of frame t and t+1
t = t+1
Connected component analysis
Eliminate unsuitable blobs
At least 2 blobs remaining?
No
Yes
Compute optical flow
Dominant motion downwards and magnitude > 3?
No
Yes
t = t+1
count = 0
Threshold frame difference of frame t and t+1
connected component analysis
Eliminate unsuitable blobs
t = t+1
increment count
No
Yes
At least 2 blobs remaining?
No
count > 3?
Yes
Compute optical flow
Dominant motion upwards and magnitude >3?
No
Blink detected

Figure 3.6: Flowchart depicting blink detection process.

The tracker is initialized on the eye regions on the frame immediately after the blink is detected. We found that twenty feature points selected in the eye regions give the best bounding boxes for the eyes during tracking.

Occasionally, when there is relatively larger motion, a few feature points could be lost. When there is drastic motion in the eyes (which can be caused by a blink or motion of the eye ball), more feature points will be lost. When the accumulated number of feature points lost is more than 20% of the total feature points, then the tracker is disabled. There have, however, been cases when less than 20% of the feature points were lost even during a blink. Thus, additional checks are performed based on a geometrical model of eye regions to ensure that the tracker will be disabled. After this, eye localization is performed to reinitialize the tracker.

3.4 Geometric model for eyes

A geometric model is built for the localised/tracked eyes to check for the validity of the eye region. The model consists of the following measures:

(a) Width to Height ratio (WHR) of the bounding box of the eyes.

(b) Ratio of the left eye's WHR to right eye's WHR.

(c) Distance between the centres of the 2 bounding boxes.

From our data set, the acceptable range for (a) is from 1.7 to 5.1 and for (b), the acceptable values are from 0.6 to 1.6. As for (c), the acceptable range in the y direction is from 0 to 60 pixels while for the x direction, the range is from 100 to 220 pixels.

3.5 Results and Discussion

3.5.1 Results for eye localization using colour components

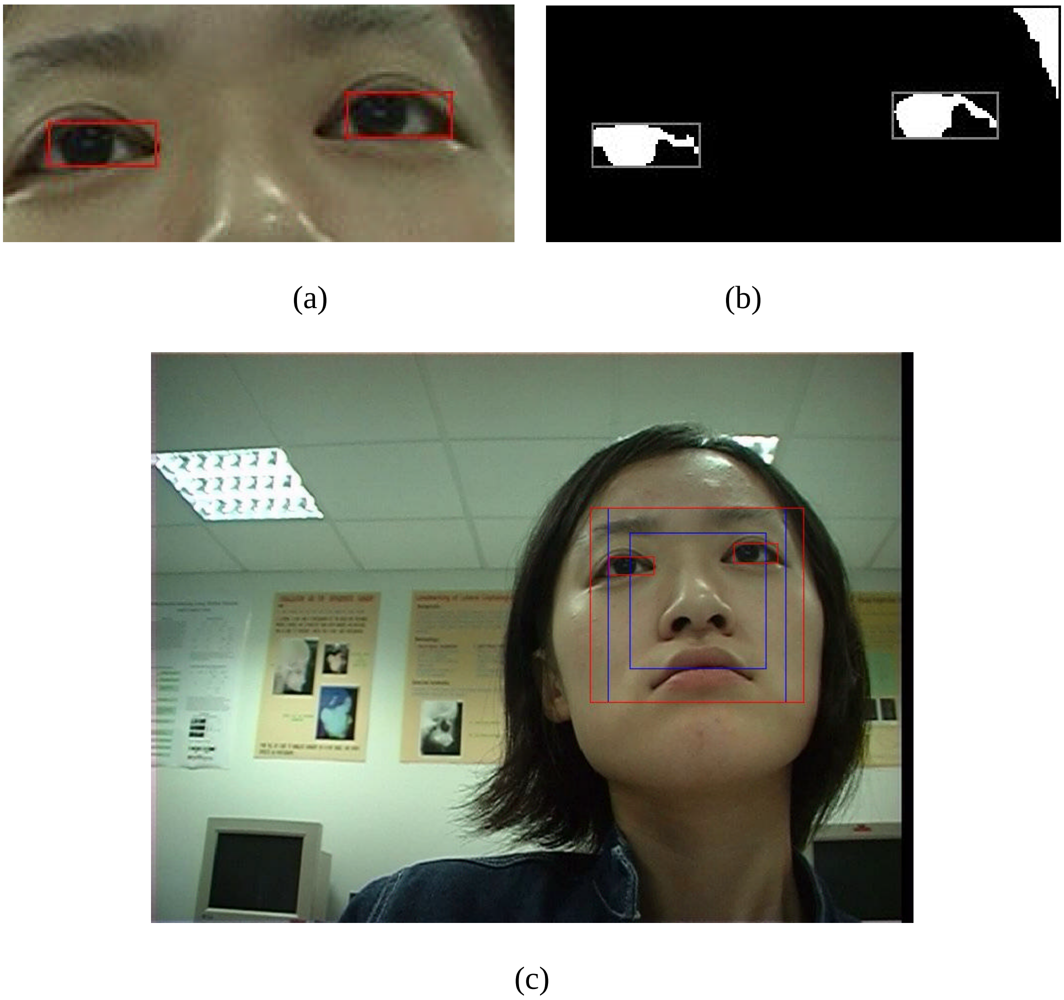
The method was tested on 15 people in 9 poses and the results of eye localization using color components are tabulated in Table 3.3. An eye is considered correctly localized if the error in the localization (with respect the ground truth) is less than 10 percent of the size of the eye. Nine poses were chosen as the camera was placed, tilted upwards (please refer to Section 1.5), and thus, lighting from the ceiling lights appears differently for the various poses. An example of eye localization using color information is shown in Figure 3.7.

Table 3.3 Results of eye localization using colour components.

Total number of eyes	***Number of eyes correctly localized***	***Percentage Accuracy***
270	263	97.4

3.5.2 Results on eye localization using blinks

Through our experiments, we found that during a blink there are only 2 blobs which will satisfy the requirements of the correct ratio, size and dominant downward motion in the frame where the blink starts (Figure 3.8a), followed by 2 blobs that have the correct ratio and size but with dominant upward motion (Figure 3.8b) as the blink ends a few frames later.



(a) (b)

(c)

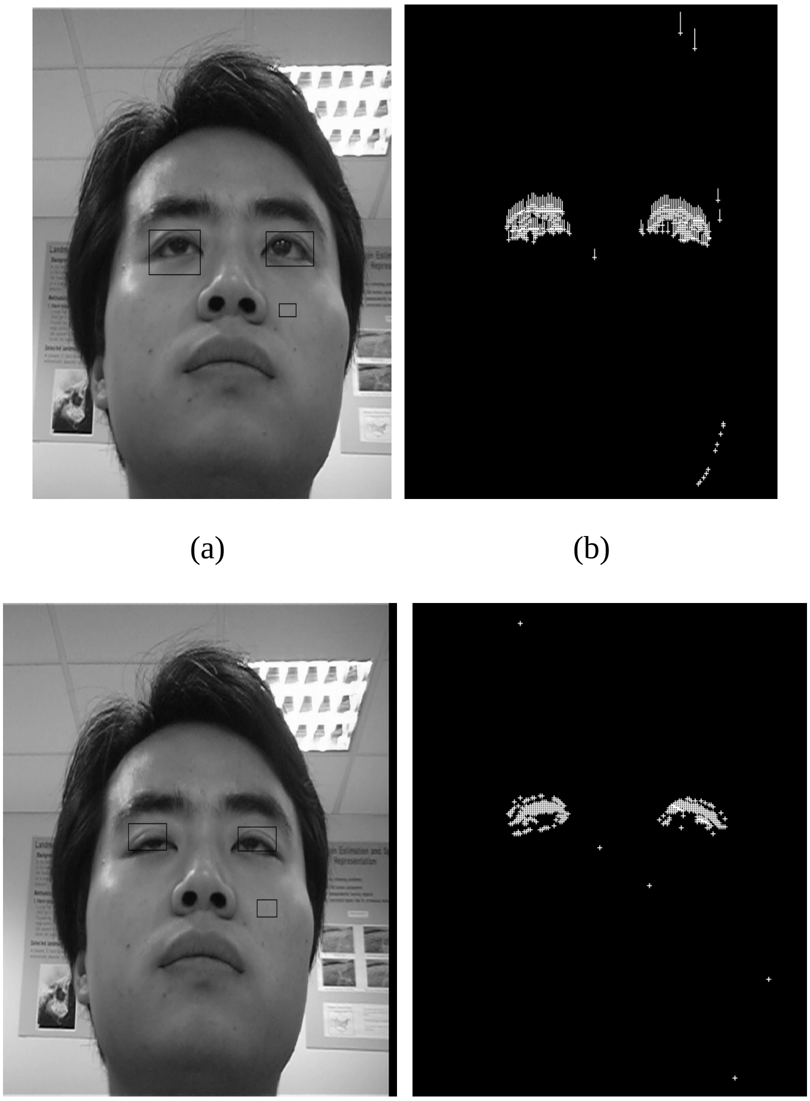
Figure 3.7: Localization of eyes using colour components. (a) Upper-half of detected face with red bounding boxes around the eyes. (b) Binary image after iris/sclera identification and morphological operations. White regions indicate possible iris/sclera regions. Bounding boxes are drawn around the blobs to localize the eyes. (c) Localized eyes in the original image. Bigger red bounding box indicates localized face while blue bounding boxes indicate face candidates (See Figure 2.6).

From Figures 3.9 and 3.10, we see that performing only frame-differencing can lead to the wrong regions being identified as eyes. Frame differencing fails to distinguish between motion of the eyeball and a blink as seen in Figure 3.9. Thus optical flow is incorporated to ensure the accuracy of the localization of the eyes. Comparing the optical flow diagrams in Figures 3.8 (b) and 3.10 (b), the magnitude of the velocity vectors in the case of the blink is much larger (as indicated by the longer arrows) than in the case of vertical head movement. Thus the magnitude of the flow can be used to differentiate downward blink movement from vertical head movements for detecting the start of a blink.

The algorithm was tested on 13 sequences of people, engaged in reading or playing computer games, which consisted of about 700 frames (resolution of the frames is 768 x 576, captured at 25 frames per second) each. There are 5 blinks in each sequence on average. It takes about 20 seconds for optical flow computations employed in blink detection per sequence. Of the 65 blinks in the sequences, two were missed and there were no false detections, giving an accuracy rate of 97.0%.

3.5.3 Results on eye tracking

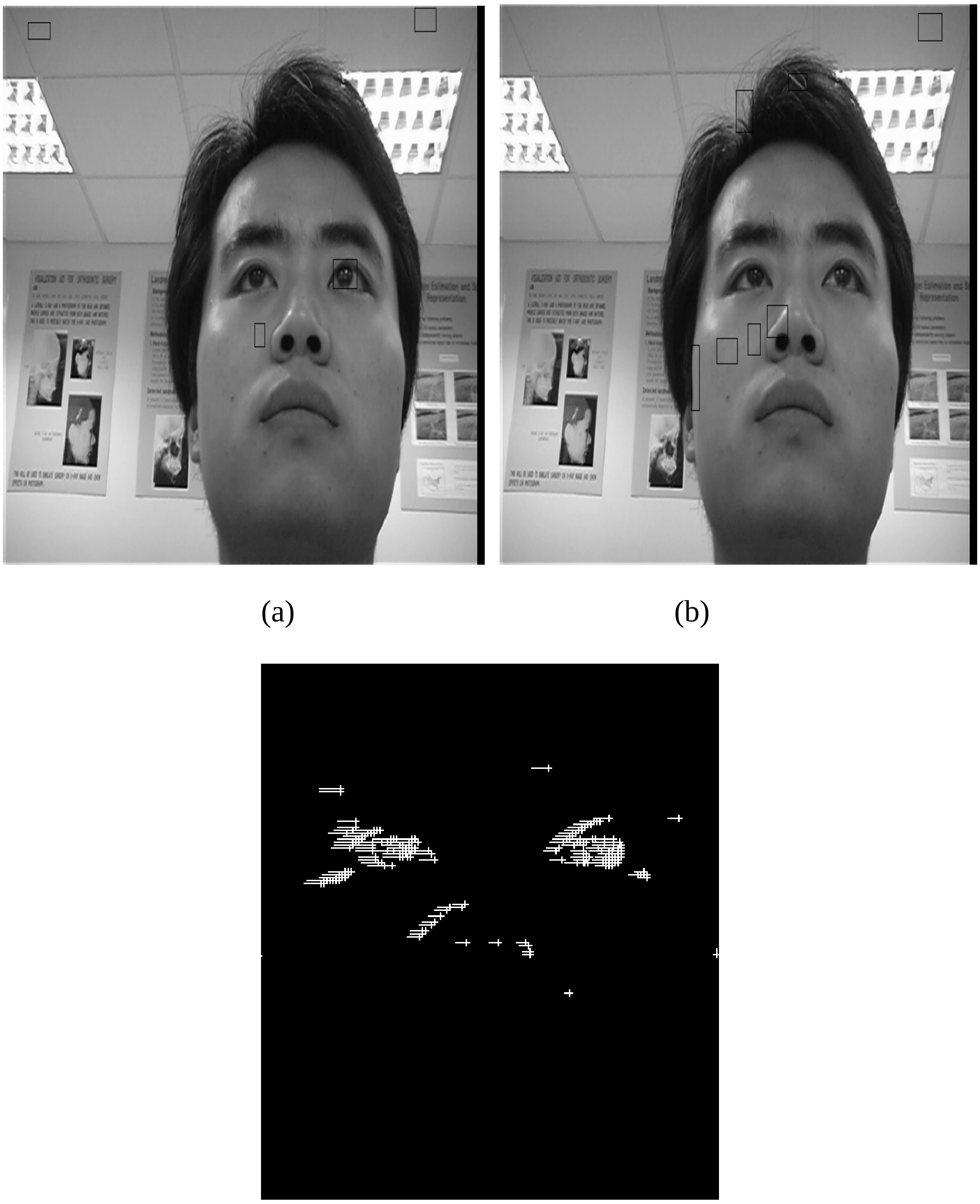
It was observed that the KLT tracker could track the eyes in the interval between blinks for the same 13 sequences used in blink detection. The tracker takes a mere 10 seconds to track eyes for a sequence of 700 frames (28 seconds long at 25 frames per second). Thus, eye localization and tracking could run in near real-time.



(a) (b)

(c) (d)

Figure 3.8: Results of motion analysis and optical flow computation.
(a) Frame differencing, thresholding and connected components on the first 2 frames of the blink to obtain motion regions. Motion regions found are bounded by black boxes.
(b) Optical flow computations on the whole image of (a). Majority of flow vectors are pointing downwards.
(c) Same motion as (a) on the frames where the eye first reopens. Motion regions are bounded by black boxes.
(d) Optical flow computations on whole image of (c). Majority of flow vectors are pointing upwards.



(a)

(b)

(c)

Figure 3.9: Results of motion analysis and optical flow computations where there is horizontal motion of the head as well as motion of the eye ball. The motion of the eye ball lasts for 2 frames.

(a) Frame differencing, thresholding and connected component analysis on first 2 frames when the movement starts to obtain motion regions. Motion regions are bounded by black boxes.

(b) Same motion analysis as (a) on the last 2 frames when the movement ends. Motion regions are bounded by black boxes.

(c) Optical flow computations on whole image of (a). Majority of the flow vectors are pointing sideways.

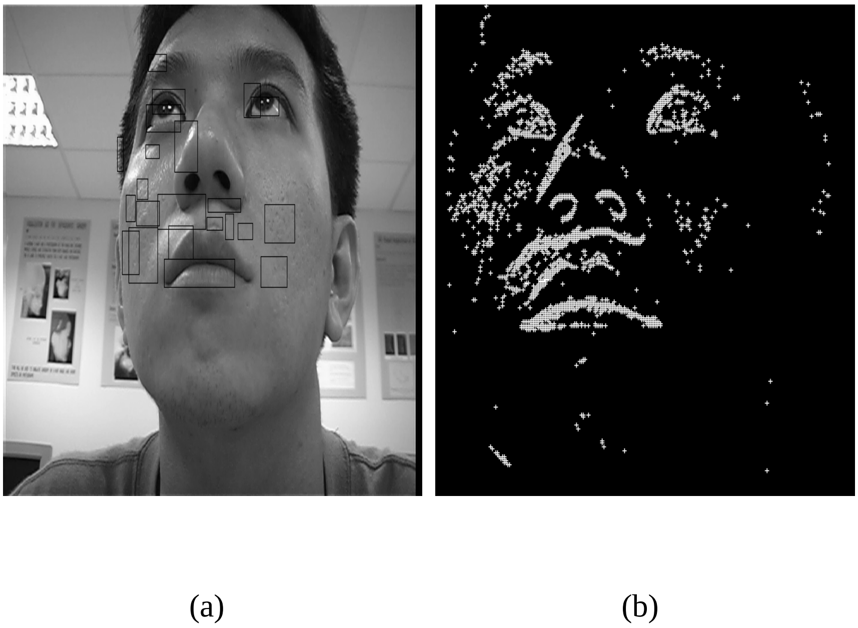
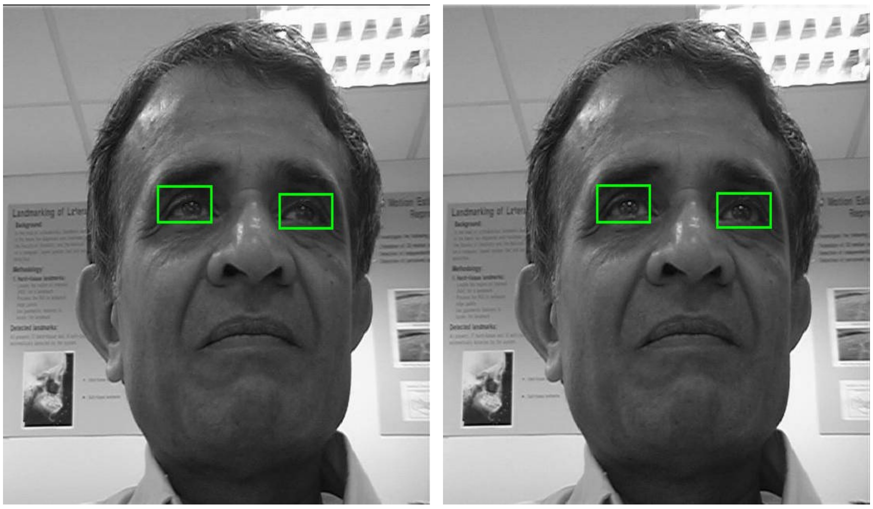


Figure 3.10: Results of motion analysis and optical flow computations where there is a vertical head movement.
(a) Frame differencing, thresholding and connected component analysis are performed to obtain motion regions. Motion regions are bounded by black boxes.
(b) Optical flow computations on whole image of (a). The majority of the flow vectors are pointing upwards and their magnitude is smaller than that seen in Figure 3.8 (for a blink where the eyelid motion is faster).

In our experiments, we found that for eyes of size about 50x30, whenever there was a drastic movement (about 9 pixels) in the subject's eyeball, the tracker was disabled. This is undesirable as the tracker should only be disabled by the occurrence of a blink. To avoid this, we use the fact that the dominant direction of the optical flow for eye-ball movements is horizontal. If the direction of the flow vectors is sideways in the motion regions, then eye-ball movement is deemed to have taken place and the tracker is reinitialized in the next frame using eye regions of the previous frame.

At every blink, to prevent drifting, the tracker needs to be reinitialized. Figure 3.11 shows the eyes being tracked in the interval between 2 blinks.



(a) *(b)*

Figure 3.11: Tracking of the eyes. (a) Immediately after being initialised, (b) Just before a blink.

3.6 Conclusion

Temporal differencing is very fast and gives an indication of possible motion regions in the image. However, this alone is not sufficient to differentiate blinking from other possible motions like head and eyeball movements. Thus, when frame differencing indicates motion, computation of the optical flow magnitude and direction can be used to differentiate blinking from the other motions. The KLT tracker is used to track the eyes between blinks. Blink detection is extremely useful in providing accurate eye localization. However, a limitation of this method of eye localization is that it only works when blinks are encountered. Thus, localizing eyes using colour information is employed in the absence of blinks. We have achieved a success rate of 97.0% for blink detection while the accuracy for localizing eyes using colour information was 97.4%.

Chapter 4

Eye Gaze Direction Determination

4.1 Introduction

In this chapter, we will discuss how the eye-gaze direction is estimated in a non-intrusive manner as input to the next module, reading detection. In the next section, the theory behind the algorithm used is first discussed followed by data preparation and implementation. Results and discussion are given before the chapter is concluded.

The classification of eye gaze direction is carried out using feature vectors obtained from projections onto eigenspace of the eyes as used in [Baluja1993] and [Varchmin1997]. A set of eigen-vectors, or eigen-eyes, of size 40x20 is built from a training set of eyes. The detected eye from the eye detection/tracking module is normalized to the pre-determined size of 40x20 and projection weights of this normalized eye to the eigen-eyes are obtained. These projections are used as input features to a Radial Basis Function neural network (RBFNN) for determining the direction of the gaze. The structure and learning strategy of the RBFNN are discussed in Section 4.2 and Section 4.3, respectively. The centres of the RBFNN are obtained by feeding the same training projections to Self Organizing Maps (SOMs) and details are given in Section 4.4.

4.2 Radial Basis Functions Neural Networks

The RBFNN, whose structure is shown in Figure 4.1, involves 3 layers, the input layer, the hidden layer and the output layer. The input layer has nodes connected to its environment, in this case, the features that we specified. The hidden layer applies a nonlinear transformation from the input space to the hidden space. The weights, ω_i, of the nodes in the output layer are just a linear sum of the outputs in the hidden layer and basically, the output layer supplies the response of the RBFNN to the activation pattern applied to the input layer.

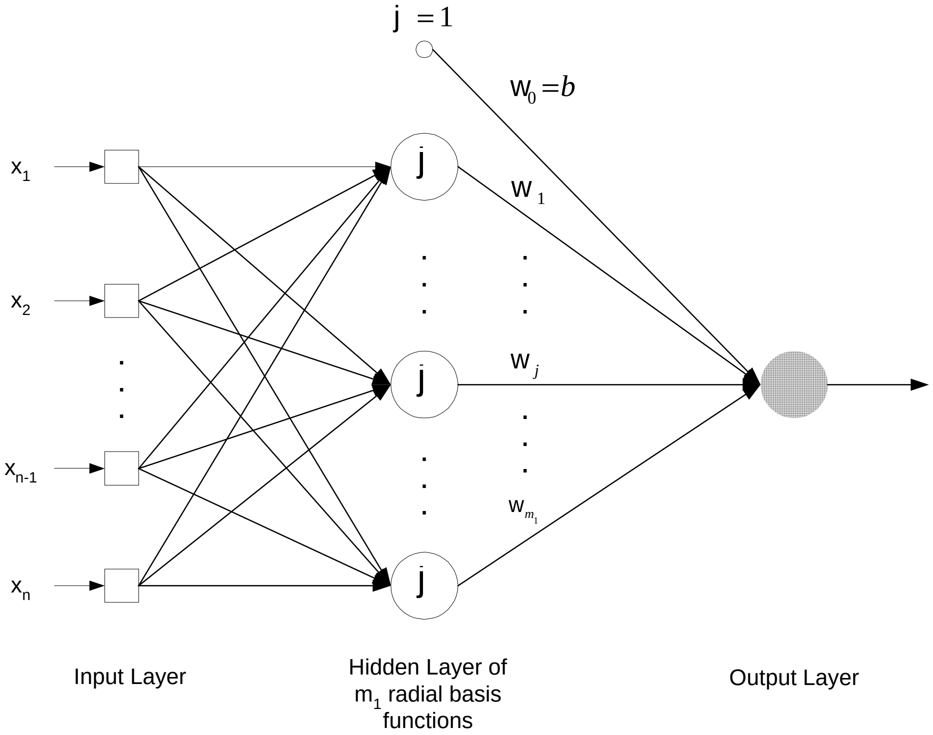


Figure 4.1: Structure of Radial Basis Function Neural Network. For simplicity, only one output node is shown in the figure.

4.3 Training the RBFNN

The learning strategy used is found in [Haykin1999]. The centres of the radial basis functions are selected through a self-organising process. Self Organizing Maps (SOMs) are chosen over K-means clustering method because we can see how the different classes cluster in the maps. How the centres are obtained is discussed in Section 4.4.

Gaussian functions are used as activation functions for the hidden nodes in the hidden layer. A gaussian activation function centred at t_i is defined as

$$G\left(\|x-t_i\|^2\right)=\exp\left(-\frac{m_1}{a^2 d_{\max}^2}\|x-t_i\|^2\right), \qquad i= 1, 2, ..., m_1 \qquad (4.1)$$

where m_1 is the number of centres and $d_{\max}$ is the maximum distance between the chosen centres. The standard deviations of the Gaussian radial basis functions are varied to find the best gaussian fit for the training data. As the value of σ is varied, given in equation (4.2), the standard deviations of all the Gaussian radial basis functions take on the same value of σ.

$$s = a\frac{d_{\max}}{\sqrt{2m_1}} \qquad (4.2)$$

where α = {0.8, 0.85,…,1.2}.

Once t_i and α are determined by some means, the only parameters that would need to be learned are the linear weights, **w,** in the output layer of the network. The pseudoinverse method is employed.

$$\mathbf{w} = \mathbf{G}^{+}\mathbf{d} \tag{4.3}$$

where **d** is the desired response vector in the training set. The matrix $\mathbf{G}^{+}$ is the pseudoinverse of the matrix **G**, which itself is defined as

$$\mathbf{G} = \{g_{ji}\} \tag{4.4}$$

where

$$g_{ji} = \exp\left(- a \frac{m_1}{d_{\max}^2} \left\| \mathbf{x}_j - \mathbf{t}_i \right\|^2 \right), \text{ j = 1, 2, ..., N; i = 1, 2, ..., m}_1 \tag{4.5}$$

where x_j is the *jth* input training vector and N is the number of training vectors.

4.4 Obtaining Centres for RBFNN using Self Organizing Maps (SOMs)

Self-Organizing Maps (SOMs) are a special class of artificial neural networks based on competitive learning; the output neurons of the network compete among themselves to get activated. Normally, only one output neuron gets activated and it is known as the winning neuron. In a SOM, neurons are placed at the nodes of a lattice, in this case, a 2-D map of size 20x20. The size of 20x20 was chosen as it gave good grouping with reasonable training times as compared to sizes such as 10x10 and 30x30. The neurons get selectively tuned to the input patterns in the course of a competitive learning process. A SOM is characterized by the formation of a topological map of the input patterns in which the spatial locations of the neurons in the lattice are indicative of intrinsic statistical features contained in the input patterns, hence its name. If the features are properly selected and the SOMs have learned well, the SOMs should show grouping in clear proper clusters. The weights of the neurons in the SOM are then used as centres for the RBFNN.

4.4.1 Algorithm for SOM

The algorithm for the SOM is given below.

1) The weights of the neurons in a map of size say 20x20 are first randomly initialised.

2) Learning: An input pattern of a fixed dimension is presented to the map and the winning neuron is determined. The neighbouring neurons around the winning neuron learn at the same rate.

3) If maximum number of iterations is reached, goto 4, otherwise goto 2.

4) Mapping: We tried 2 types of mapping to see any difference in the formation of clusters .

    i) Present the training input patterns, one at a time, to the map. The neuron whose weight is closest to this input will claim this input. A histogram of the number of times that each label was won by the neuron is maintained for every neuron. At the end of the pattern presentation process, the histogram of each neuron is examined. The neuron is assigned the label which it won the most number of times.

ii) Present the input patterns from all the labels to each neuron in each turn. The neuron will win the pattern that is closest to it. A histogram of the number of times that each label was won by the neuron is maintained for every neuron. At the end of the pattern presentation process, the label of the maximum number of inputs claimed by the neuron will be assigned to this neuron.

When method (i) is employed, some of the neurons may not claim any pattern and thus their labels are not determined. In this project, we have selected method (ii) mainly because the clusters obtained are much more defined than method (i). The SOMs obtained using the 2 methods are shown in Figure 4.4.

4.4.2 Parameters

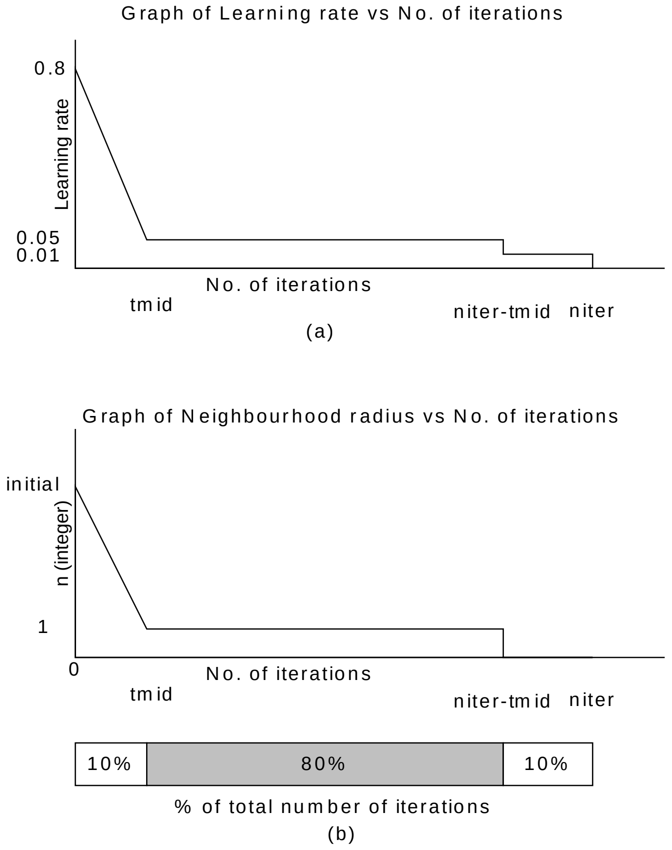
Some considerations involved in SOMs are:

1) Initialisation of the weights. These are randomly initialised.
2) learning rate
3) neighbourhood size
4) Terminating criterion: we terminate the iterations based on a maximum number of iterations.

The learning rate refers to how much the winning neuron and its neighbours learn from the input patterns. The neurons learn by having their weights adjusted according to equation (4.6). The learning rates were changed as shown in Figure 4.2 (a). Figure 4.2 (b) shows the change of the neighbourhood radius with respect to the number of iterations.

$$\mathbf{w}_{n,a} = \alpha(\mathbf{x} - \mathbf{w}_{n,b}) \tag{4.6}$$

where $\mathbf{w}_{n,b}$ is the weights of the neuron before learning, $\mathbf{w}_{n,a}$ is the adjusted weights to be added to the neuron, $\mathbf{x}$ is the input pattern, and α is the learning rate.



(a)

(b)

Figure 4.2: Learning rate and neighourhood radius. (niter = Total no. of samples *100, tmid = Total no. of samples *10)

The initial radius is set such that the number of neighbourhood neurons covers approximately half the map size. This is to allow a quick global organisation of the SOM. As time passes, the neighbourhood radius shrinks and eventually, only the winning neuron learns. This allows the SOM to pick up local characteristics of the patterns. After the SOM has learnt, the weights of the neurons are used as centres for RBFNN.

4.5 Implementation of Gaze Direction Determination

The subjects are requested to look at the mouse pointer as it is moved and sequences of the subjects are captured. The positions of the pointer are logged at the same frequency at which the frames are captured, i.e., 25 frames per second. Thus, we know where the subject is looking at in every frame and this will help us in labeling our data.

The mouse pointer is moved horizontally, from left to right, and from top to bottom, resembling how the gaze will move during reading. While following the mouse pointer, if a blink occurs, the subjects’ eyes will be closed or half closed. Thus, images of such nature are not used for training or testing for eye gaze estimation. In addition, during the transition in which the mouse pointer is moved from right to left, as the mouse pointer is moving rapidly, the eyes may not be able to follow as accurately. Thus, samples during this transition are also discarded.

The remaining eye gaze samples are labeled according to where the subjects are looking on the monitor screen (from the logged position of the pointer). The screen is divided into 4 vertical strips as shown in Figure 4.3.

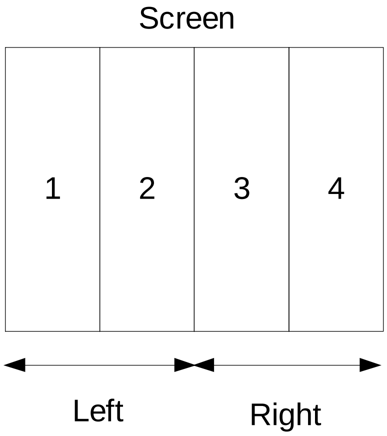


Figure 4.3: Regions defined on the monitor.

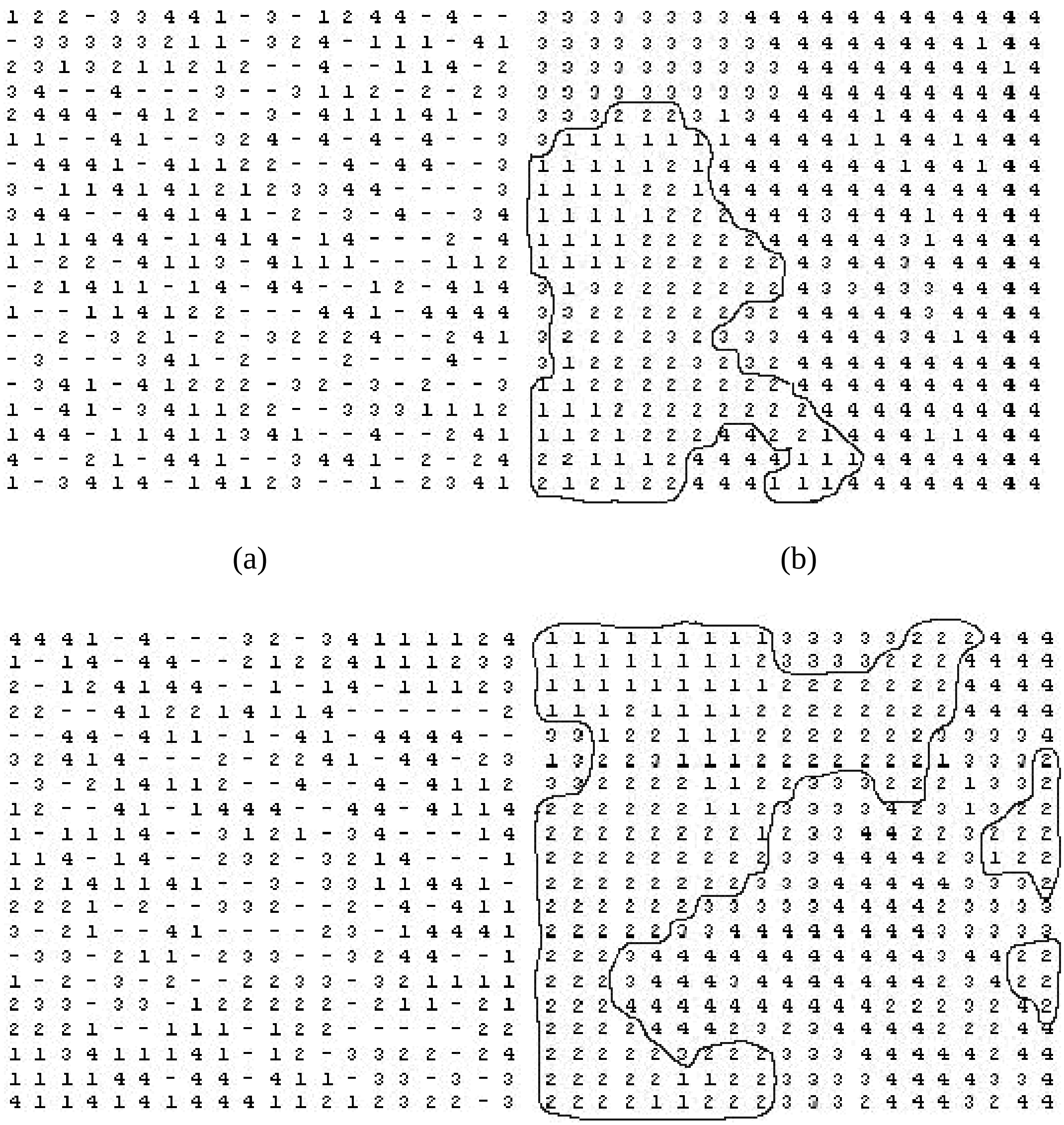
Once these labeled images are obtained, eye localization and tracking (as in Chapter 3) are performed to extract the left eye of the person. The left eyes are then normalized, or resized, to a pre-determined size of 40x20 using bi-cubic interpolation. After this, PCA of the eye for each person is computed from these 40x20 training images. On average, 1800 images are used for training. The number of eigen-eyes is chosen for 95% representation accuracy. Numbers of eigen-eyes for 95% accuracy ranged from 38 to 55. Projections of each training eye sample to these eigen-eyes are obtained. These projections are used as input vectors to the SOM to obtain centres for RBFNN (see Section 4.4). A map size of 20x20 is used and the weights of 400 neurons in the map are used as centres for RBFNN. Using these centres and the projections of training eye sample to the eigen-eyes, the RBFNN is trained as discussed in Section

4.3. Four output nodes are used in the output layer. The 4 classes obtained from the RBFNN are then quantised to 2 (left and right) as shown in Figure 4.3.

4.6 Results and Discussion

Two sequences are captured for each subject, one for training and one for testing. For each subject, a new SOM and RBFNN is built. Figure 4.4 shows examples of SOMs obtained. Figure 4.4 (a) and (b) show SOMs for the same person by using different methods of mapping mentioned in Section 4.4. Figure 4.4 (a) is obtained using method (i) while Figure 4.4 (b) is obtained using method (ii). Figure 4.4 (c) and (d) are SOMs obtained for another person. Similarly, Figure 4.4 (c) is obtained using method (i) and Figure 4.4 (d) is obtained using method (ii). From Figure 4.4, SOMs obtained using method (i) have gaps, represented by "-". In addition, the SOMs obtained using method (ii) are more organized compared to method (i). Comparing Figure 4.4 (b) and (d), the gaze patterns for person A form better clusters, values 1 and 2 (representing Left) are grouped "together" in the middle left hand side of the map and values 3 and 4 (representing Right) are grouped on the rest of the map.

The average results 10 people for the gaze direction estimation module are tabulated in Table 4.1. In obtaining the standard deviation using equation (4.2), we have used α = 1. On average, 1800 training and 450 testing samples are used.



```
1 2 2 - 3 3 4 4 1 - 3 - 1 2 4 4 - 4 - -
- 3 3 3 3 3 2 1 1 - 3 2 4 - 1 1 1 - 4 1
2 3 1 3 2 1 1 2 1 2 - - 4 - - 1 1 4 - 2
3 4 - - 4 - - - 3 - - 3 1 1 2 - 2 - 2 3
2 4 4 4 - 4 1 2 - - 3 - 4 1 1 1 4 1 - 3
1 1 - - 4 1 - - 3 2 4 - 4 - 4 - 4 - - 3
- 4 4 4 1 - 4 1 1 2 2 - - 4 - 4 4 - - 3
3 - 1 1 4 1 4 1 2 1 2 3 3 4 4 - - - - 3
3 4 4 - - 4 4 1 4 1 - 2 - 3 - 4 - - 3 4
1 1 1 4 4 4 - 1 4 1 4 - 1 4 - - - 2 - 4
1 - 2 2 - 4 1 1 3 - 4 1 1 1 - - - 1 1 2
- 2 1 4 1 1 - 1 4 - 4 4 - - 1 2 - 4 1 4
1 - - 1 1 4 1 2 2 - - - 4 4 1 - 4 4 4 4
- - 2 - 3 2 1 - 2 - 3 2 2 2 4 - - 2 4 1
- 3 - - - 3 4 1 - 2 - - - 2 - - - 4 - -
- 3 4 1 - 4 1 2 2 2 - 3 2 - 3 - 2 - - 3
1 - 4 1 - 3 4 1 1 2 2 - - 3 3 3 1 1 1 2
1 4 4 - 1 1 4 1 1 3 4 1 - - 4 - - 2 4 1
4 - - 2 1 - 4 4 1 - - 3 4 4 1 - 2 - 2 4
1 - 3 4 1 4 - 1 4 1 2 3 - - 1 - 2 3 4 1
```

(a)

```
3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4
3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 1 4 4
3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 1 4
3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4
3 3 3 2 2 2 3 1 3 4 4 4 4 1 4 4 4 4 4 4
3 1 1 1 1 1 1 1 4 4 4 4 1 1 4 4 1 4 4 4
1 1 1 1 1 2 1 4 4 4 4 4 4 4 1 4 4 1 4 4
1 1 1 1 2 2 1 4 4 4 4 4 4 4 4 4 4 4 4 4
1 1 1 1 1 2 2 2 4 4 4 3 4 4 4 1 4 4 4 4
1 1 1 1 2 2 2 2 2 4 4 4 4 4 3 1 4 4 4 4
1 1 1 1 2 2 2 2 2 2 4 3 4 4 3 4 4 4 4 4
3 1 3 2 2 2 2 2 2 2 4 3 3 4 3 3 4 4 4 4
3 3 2 2 2 2 2 2 3 2 4 4 4 4 4 3 4 4 4 4
3 2 2 2 2 3 2 3 3 3 4 4 4 4 3 4 1 4 4 4
3 1 2 2 2 2 3 2 3 2 4 4 4 4 4 4 4 4 4 4
1 1 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4 4 4 4
1 1 1 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4 4 4
1 1 2 1 2 2 2 4 4 2 2 1 4 4 4 1 1 4 4 4
2 2 1 1 1 2 4 4 4 4 1 1 1 4 4 4 4 4 4 4
2 1 2 1 2 2 4 4 4 1 1 1 4 4 4 4 4 4 4 4
```

(b)

```
4 4 4 1 - 4 - - - 3 2 - 3 4 1 1 1 1 2 4
1 - 1 4 - 4 4 - - 2 1 2 2 4 1 1 1 2 3 3
2 - 1 2 4 1 4 4 - - 1 - 1 4 - 1 1 1 2 3
2 2 - - 4 1 2 2 1 4 1 1 4 - - - - - - 2
- - 4 4 - 4 1 1 - 1 - 4 1 - 4 4 4 4 - -
3 2 4 1 4 - - - 2 - 2 2 4 1 - 4 4 - 2 3
- 3 - 2 1 4 1 1 2 - - 4 - - 4 - 4 1 1 2
1 2 - - 4 1 - 1 4 4 4 - - 4 4 - 4 1 1 4
1 - 1 1 1 4 - - 3 1 2 1 - 3 4 - - - 1 4
1 1 4 - 1 4 - - 2 3 2 - 3 2 1 4 - - - 1
1 2 1 4 1 1 4 1 - - 3 - 3 3 1 1 4 4 1 -
2 2 2 1 - 2 - - 3 3 2 - - 2 - 4 - 4 1 1
3 - 2 1 - - 4 1 - - - - 2 3 - 1 4 4 4 1
- 3 3 - 2 1 1 - 2 3 3 - - 3 2 4 4 - - 1
1 - 2 - 3 - 2 - - 2 2 3 3 - 3 2 1 1 1 1
2 3 3 - 3 3 - 1 2 2 2 2 2 - 2 1 1 - 2 1
2 2 2 1 - - 1 1 1 - 1 2 2 - - - - - 2 2
1 1 3 4 1 1 1 4 1 - 1 2 - 3 3 2 2 - 2 4
1 1 1 1 4 4 - 4 4 - 4 1 1 - 3 3 - 3 - 3
4 1 1 4 1 4 1 4 4 4 1 1 2 1 2 3 2 2 - 3
```

(c)

```
1 1 1 1 1 1 1 1 1 3 3 3 3 3 2 2 2 4 4 4
1 1 1 1 1 1 1 1 2 3 3 3 3 2 2 2 4 4 4 4
1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 4 4 4 4
1 1 1 2 1 1 1 1 2 2 2 2 2 2 2 2 4 4 4 4
3 3 1 2 2 1 1 1 2 2 2 2 2 2 2 3 3 3 3 4
1 3 2 2 3 1 1 1 2 2 2 2 2 2 2 1 3 3 3 2
3 3 2 2 2 2 1 1 2 2 3 3 3 2 2 2 1 3 3 2
2 2 2 2 2 2 1 1 2 3 3 3 3 4 2 3 1 3 2 2
2 2 2 2 2 2 2 2 1 2 3 3 4 4 2 2 3 2 2 2
2 2 2 2 2 2 2 2 2 3 3 4 4 4 4 2 3 1 2 2
2 2 2 2 2 2 2 2 3 3 3 4 4 4 4 4 3 3 3 2
2 2 2 2 2 2 3 3 3 3 3 4 4 4 4 2 3 3 3 3
2 2 2 2 2 3 3 4 4 4 4 4 4 4 4 3 3 3 3 3
2 2 2 3 4 4 4 4 4 4 4 4 4 4 4 3 4 4 2 2
2 2 2 3 4 4 4 3 4 4 4 4 4 4 4 2 3 4 2 2
2 2 2 4 4 4 4 4 4 4 4 4 4 2 2 2 3 2 4 2
2 2 2 2 4 4 4 2 3 2 3 4 4 4 4 2 2 2 4 4
2 2 2 2 2 3 2 2 2 3 3 3 4 4 4 4 4 2 4 4
2 2 2 2 2 1 1 2 2 3 3 3 3 4 4 4 4 3 3 4
2 2 2 2 1 1 2 2 2 3 3 3 2 4 4 4 3 2 4 4
```

(d)

Figure 4.4: Self-Organising Maps using the algorithm discussed in Section 4.4.1.
(a) SOM obtained for person A using mapping method (i),
(b) SOM obtained for person A using mapping method (ii), The cluster drawn represents a homogenous group of vectors representing the left side of the screen.
(c) SOM obtained for person B using mapping method (i),
(d) SOM obtained for person B using mapping method (ii) as above.

Table 4.1 Classification results for gaze direction estimation using RBFNN.

	% Error in classification					
	Column 1	**Column 2**	**Column 3**	**Column 4**	**Left**	**Right**
Training Samples	4.1	6.5	7.1	4.8	3.1	4.1
Testing Samples	7.5	10.8	12.3	9.1	8.9	10.2

For example, under “Column 1” for training samples, 4.1% of the samples belonging to this column are wrongly classified; and under “Left” for test samples, 8.9% of the samples that belong to columns 1 and 2 are wrongly classified.

From Table 4.1, the percentage errors for columns 2 and 3 are significantly higher than that for column 1 and 4. Depending on how closely the subjects follow the mouse pointer, there could be some misassociation of frames and mouse position log. Thus, some samples could be wrongly labelled, especially those at the boundaries of the columns. Chances of misassociation are higher for columns 2 and 3, as compared to columns 1 and 4 as the former pair has more boundaries.

In [Baluja1993], the system is reported to run at 15Hz on a Sun SPARC 10 machine, and is able to achieve a 1.5 degree accuracy. Individual users have to be customized to use the system. For [Varchmin1997], their system is reported to run at 1 frame per second on a common workstation. Varchmin et. al. reported an average error of 1.5 degrees for the gaze pan angle and 2.5 degrees for the tilt angle. In their implementation, a lamp is used to produce a specular highlight on the user’s eye to help center the region of interest. For the gaze module of this work, on a Pentium 4 1.7 GHz machine, once the eyes are localized, it takes on average 40ms to determine gaze direction of a person for each frame in sequences captured at 25 frames per second.

4.7 Conclusion

In conclusion, a RBFNN classifier whose centres are obtained using SOM is implemented. The classifier built is used to differentiate which side (left/right) of the monitor the eyes are looking at. The classifier has an accuracy of 90% classification for unseen data and it runs at 25 frames per second.

Chapter 5

Reading Detector

5.1 Introduction

The proposed reading detection method uses gaze direction as input. The modules discussed in Chapters 2 to 4 provide a means to obtain the required gaze directions. This chapter discusses the algorithm for reading detection and how gaze directions are used for this purpose. Theory for reading detection is presented in the next section, followed by preparation of data to test the algorithm in Section 5.3. The results of the system are presented and discussed in Section 5.4 before the chapter is concluded.

During reading, eyes make periodic movements from left to right and back. It has been generally accepted that there are 2 types of eye movements, saccades and fixations, during reading. Saccades are jerky motions as the eyes make a scan. Fixations refer to the times when the eyes dwell longer on certain words.
Different people fixate on different words while reading. As we are interested in capturing the characteristic periodic movements of the gaze during reading, the gaze directions on the monitor are quantised into 2 categories, “LEFT” or “RIGHT”. These two gaze directions are obtained from the gaze direction module (Chapter 4).

Quantisation of gaze directions into 2 levels smoothes the variation of fixation on different words.

Finite State Machines (FSMs) are used to model eye gaze transitions. These are described in Section 5.2.1 and the algorithm for reading detection is discussed in detail in Section 5.2.2.

5.2 Theory of Reading Detection

5.2.1 Finite State Machines

A finite-state machine (FSM) is an abstract model of a system (physical, biological, mechanical, electronic, or software) whose key components are

- a finite number of states which represent the internal "memory" of the system by implicitly storing information about the past.
- state transitions which represent the "response" of the system to its environment. Transitions depend upon the current state of the machine as well as the current input and often result in a change of state.

Consider, for example, the use of an FSM to model an old-time soda machine that dispenses soda for 30 cents as shown in Figure 5.1. The possible inputs to the machine are n - nickel, d - dime, q - quarter, s - select soda.

The states of the machine are designated by circles, each labeled with the amount of money that has been deposited so far. State 00 is designated as the *start* or *initial state* by the incoming arrow. States which represent a total input of 30 or more cents are considered *final states* and are designated by double circles. The transitions from state to state are shown as arcs (lines with arrows). Each transition is labeled with the input that caused it.

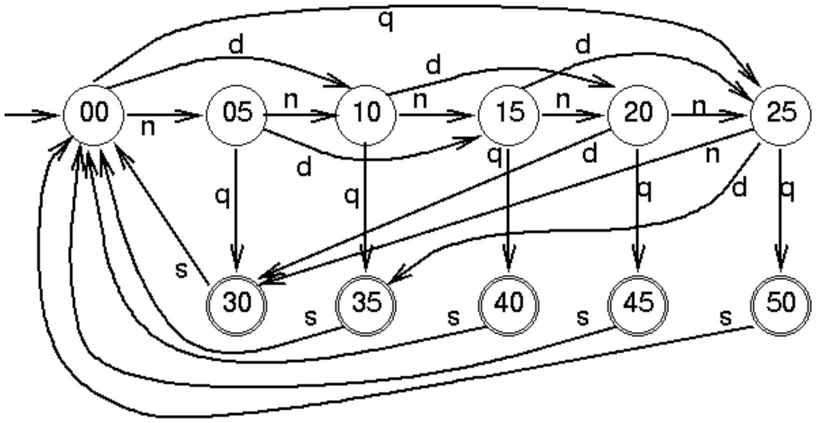


Figure 5.1: Finite State Machine to model an old-time soda machine.

If a person puts a nickel into the machine followed by a dime followed by a quarter, the FSM would transition from state 00 to state 05 to state 15 to final state 40. At that point, he or she could select a soda.

In addition to the FSM state, there may be variables, external to the system, that remember other details. The designer has to use judgement to decide what to model with a FSM state and what to leave as a variable.

5.2.2 Reading Detection Algorithm

Hong, et. al. [Hong2000] used the FSM for gesture recognition. We use a similar technique for reading detection. FSM is chosen because it is well-suited to model the required information. In addition, usually, one can easily achieve real-time speed with FSM as it is computationally inexpensive. The inputs to the reading detection algorithm from the gaze direction module are either "LEFT" or "RIGHT" for every frame, indicating that the eye gaze is on the left or right side of the monitor respectively. Two pieces of information are required to be captured from the periodic movements of the gaze during reading. The first is the type of transition, i.e., from left to right or from right to left. The second is the minimum and maximum dwell times of the gaze on one side of the monitor before it makes a transition to the other side. The states of the FSM are "LEFT" and "RIGHT". An example of a FSM that can be used to model reading is shown in Figure 5.2. Two variables, Tmin and Tmax, are used to store the minimum and maximum dwell times encountered in each state of the machine. The dwell times are in units of number of frames. An additional variable, TCounter, is used to keep track of the amount of time the FSM stays in each state. The FSM shown has 4 states. For convenience, the length of a FSM is used to denote the number of states the FSM has.

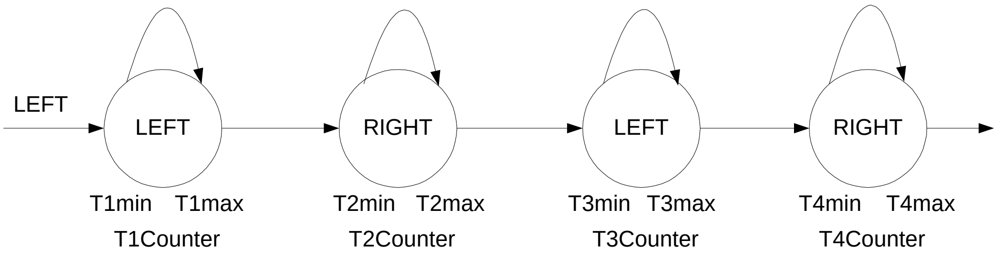
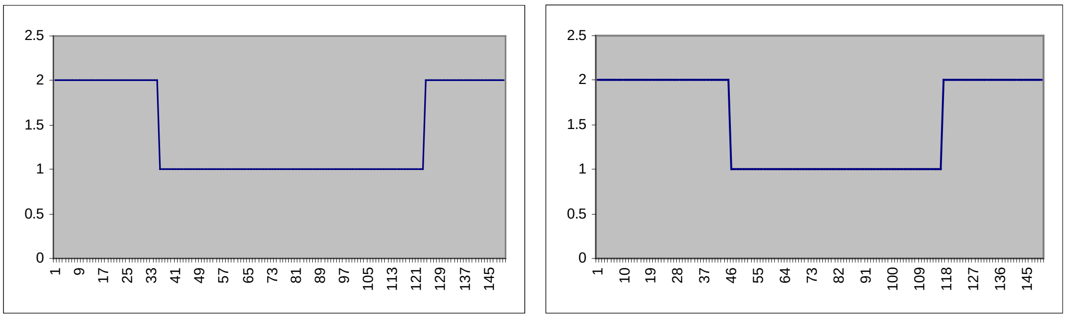


Figure 5.2: An example of a FSM.

The FSM makes 2 types of transitions, either self-transition or transition to the next state. The FSM makes a self-transition if the next gaze direction input is the same as the current one. In this case, the counter, e.g. for the first state, T1Counter is increased by 1. In order to make a transition to another state, 2 criteria have to be satisfied. The next gaze input must be different from the current state and the current state's counter must be between the current state's Tmin and Tmax. Otherwise, the particular FSM is disabled and we cannot infer from it that the current gaze pattern inputs correspond to that of reading. Transition into the first state occurs when the gaze input is appropriate. For example, in Figure 5.2, the gaze input has to be "LEFT" in order to enter the first state of the FSM. Another term "order" is used to denote the ordering of the FSM's states. For the FSM in Figure 5.2, the order of the FSM is [L R L R]. This FSM is different from another FSM which may have the same length but has a different order of states, i.e., [R L R L].

5.2.2.1 Building a Database of FSMs

Using labeled training data of a person engaged in reading, a set of FSMs are built to represent the (possibly different) types of reading patterns, and stored in a FSM database.



(a) (b)

State	Tmin	Tmax
Right	35	35
Left	88	88
Right	27	27

(c)

State	TCounter
Right	45
Left	71
Right	44

(d)

State	Tmin	Tmax
Right	35	45
Left	71	88
Right	27	44

(e)

Figure 5.3: Example illustrating how the database of FSMs is constructed. Value of 1 in the y-axis indicates "Left" and 2 indicates "Right". The x-axis refers to frame numbers. The FSMs are of length 3 and have state order [R L R].
(a) Gaze pattern with transitions at 35-36, 123-124.
(b) Gaze pattern with transitions at 45-46 and 116-117.
(c) A new FSM is constructed for (a), assuming that this FSM is not in the database.
(d) A FSM representing the pattern in (b) exists. Hence the dwell time parameters of the existing FSM are updated.
(e) Dwell time parameters, Tmin and Tmax of FSM shown in (c) in database are updated when the FSM in (d) is presented to the database.

The Tmin and Tmax parameters of each state of a FSM are learnt from the training data. During training, a trial FSM, similar to the one shown in Figure 5.2, is constructed for each reading sample, when only the dwell time in each state is noted. If the trial FSM has the same order and number of states as one in the existing database, the dwell time parameters (Tmin and Tmax) of the exisiting FSM are merely updated and the trial FSM is deleted. Otherwise the FSM is added to the database. If Tmin for a particular state of an existing FSM in the database is greater than the corresponding value for the trial FSM, Tmin is set to trial FSM's TCounter value. Similarly, if the value of Tmax of a state of an existing FSM in the database is less than that of the corresponding value of the trial FSM, Tmax is set to trial FSM's Tcounter value. Figure 5.3 shows how the parameters for dwell times are obtained.

5.2.2.2 Noise filtering

Filtering is required to remove noise in the output of the gaze direction module arising from classification errors, or from the subject back-tracking during reading, at the center of the monitor (the boundary of LEFT and RIGHT). Figure 5.4(a) shows an output pattern from the gaze detection module. Figure 5.4(b) shows the results after filtering. Here, filtering involves smoothing isolated peaks/valleys as well as groups of peaks/valleys.

Two parameters are used for filtering, called stable width and group width (width of groups of peaks/valleys), which are defined as follows:

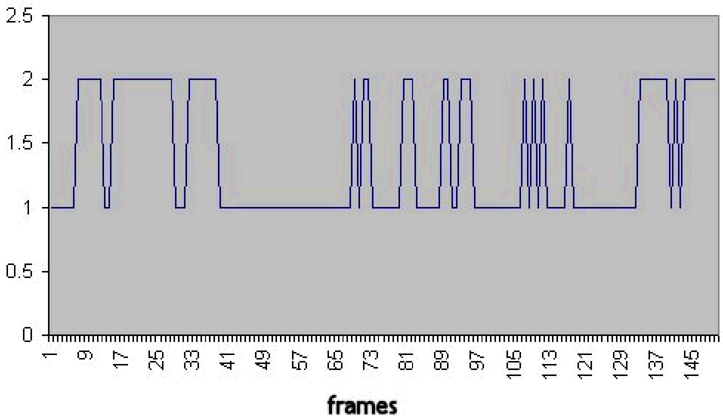
Stable width: The amount of time units during which the data is constant.

Group width: The time width where there is a “frequent” change in the data points.

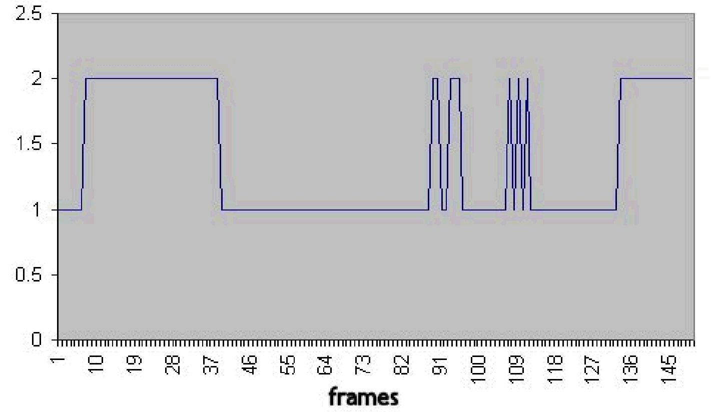
Isolated peaks/valleys are filtered out. Peaks/valleys are considered isolated if they fulfill 2 criteria: 1) if the peak/valley widths are less than a specified value, and 2) the peaks/valleys have stable widths on their left and right that are not less than a specified value. In this project, the specified values are the same for stable widths on the left and right of peaks/valleys.

Groups of “high frequency” peaks/valleys are smoothed if their widths are less than a specified group width as they are most likely noise. High frequency refers to continuous transitions with the widths of peaks/valleys less than “stable width”. When the group width of “high frequency” peaks/valleys exceeds a specified value, they are left unchanged. This is because it could be due to the eye making rapid left to right or right to left transitions.

In Figure 5.4(a), isolated peaks/valleys are found at 13-14, 29-31, 80-82, 117 and these are removed as shown in Figure 5.4(b). Groups of peaks/valleys are found at 68-73, 88-96, 106-112, 139-143. If the group width is set to 6, then groups at 68-73 and 139-143 are filtered as shown in Figure 5.4(b).



(a)



(b)

Figure 5.4: Input pattern from the gaze direction module. The y axis refers to gaze direction, value of 1 for LEFT while value of 2 indicates RIGHT. (a) Before filtering, (b) After filtering.

5.2.2.3 Determination of Reading

After the database of FSMs (a description of FSMs in the database is given in Section 5.4.1.) representing reading has been built from training data, they are used to detect reading in 2 steps. The first step involves comparing the FSM of gaze direction over an interval, say 6 seconds, with those in the database. If no existing FSM in the database has the same number of states (FSM length) and order of states, the order is [R L R], no reading is deemed to have taken place in that interval. If the database has a FSM that has same length and state order as the FSM presented to the database, the dwell time of each state of the sample FSM is compared with that of the FSM in the database. If each of these dwell times falls within Tmin and Tmax (inclusive of Tmin and Tmax) of every state, then reading is deemed to have taken place over the interval. Otherwise, no reading is deemed to have taken place. The first step of reading detection is performed, say at intervals of one second. For example, for a detection interval of 6 seconds, at t =10, gaze direction information from t = 4 to t = 10 is used. For the next detection, at t = 11, gaze direction information from t = 5 to t = 11 is used.

In the second step, reading detections over a set of 6 consecutive intervals is considered to determine whether reading is taking place. The result of the majority of these 6 detections is used to determine whether reading is taking place. For this project, 6 consecutive windows are used with a reading detection window of 6 seconds.

5.3 Data Preparation

Ten subjects are asked to choose an article they would like to read from a list of 30 articles. Each subject is requested to read the articles for about 10 minutes, seated about 0.5 metres away from the monitor. These articles are informative in nature. Bulleted text and pictures are common in these articles as shown in Figure 5.5. The articles are prepared to minimize any variation due to formatting such as font type and font size. The font face used is Arial with the size of the main text set at 18 inches. The articles are displayed on a 17” monitor with a resolution of 1024x768. On average, 6 minutes of the sequence of each subject is extracted for training and testing purposes.

Besides reading, the subject is also asked to undertake various common computer-related activities. In each of these activities, different types of attention are emphasized. There are 3 types of activities, namely playing computer games, watching video clips and deciphering hidden patterns in complicated images. Two examples in each activity were used.

5.3.1 Computer games

The first computer game that the subject is asked to play is called picture error, depicted in Figure 5.6. In this game, the subject has to identify 5 differences between 2 images. In this case, one would expect the gaze of the subject to move quickly from one image to another. This left-right transition is almost the same as that for reading except that it is performed at a faster speed.

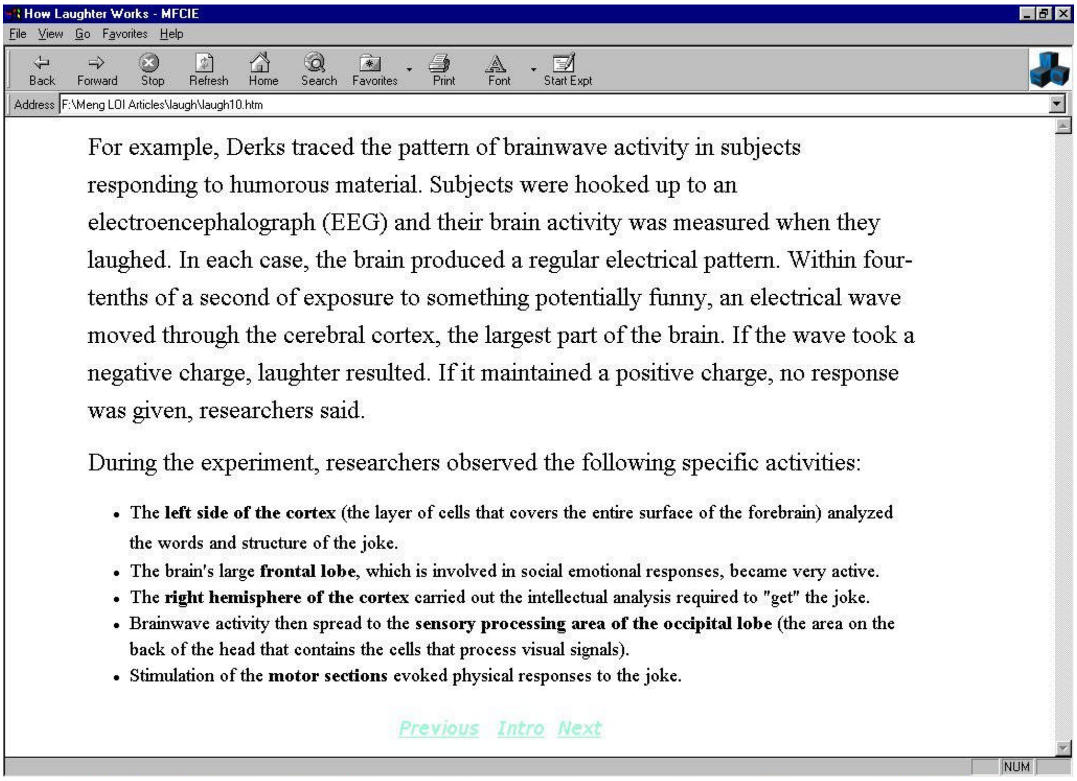
The second computer game is called zball, a variant of the well-known brick game shown in Figure 5.7. In this game, a ball is moving around, knocking off bricks and the subject will be controlling a vertical slider to prevent the ball from going out of bounds to the right hand side. Again, one would expect left-right transitions of gaze as the eyes follow the ball. However, in this case, the speed at which the transitions are made is expected to be lower than that of the first game, closer to the speed of reading, due to the speed at which the ball is moving.

5.3.2 Complicated images with hidden patterns

In Figure 5.8, the subject is asked to locate 11 faces embedded in the image. As the subject searches for the hidden pattern, i.e. faces, they would have to focus or concentrate on certain portions of the image. Hence, one would expect their gazes to be fixated on a particular spot for a relatively long period of time before their gaze shifts quickly to another region.

5.3.3 Video Clips

The users were shown 2 video clips on wild life documentaries, e.g. Wild Sanctuaries, showing a hunting scene. One would expect eye movements with moderate speed but for the majority of the time, the eyes are focused on the centre of the scene.

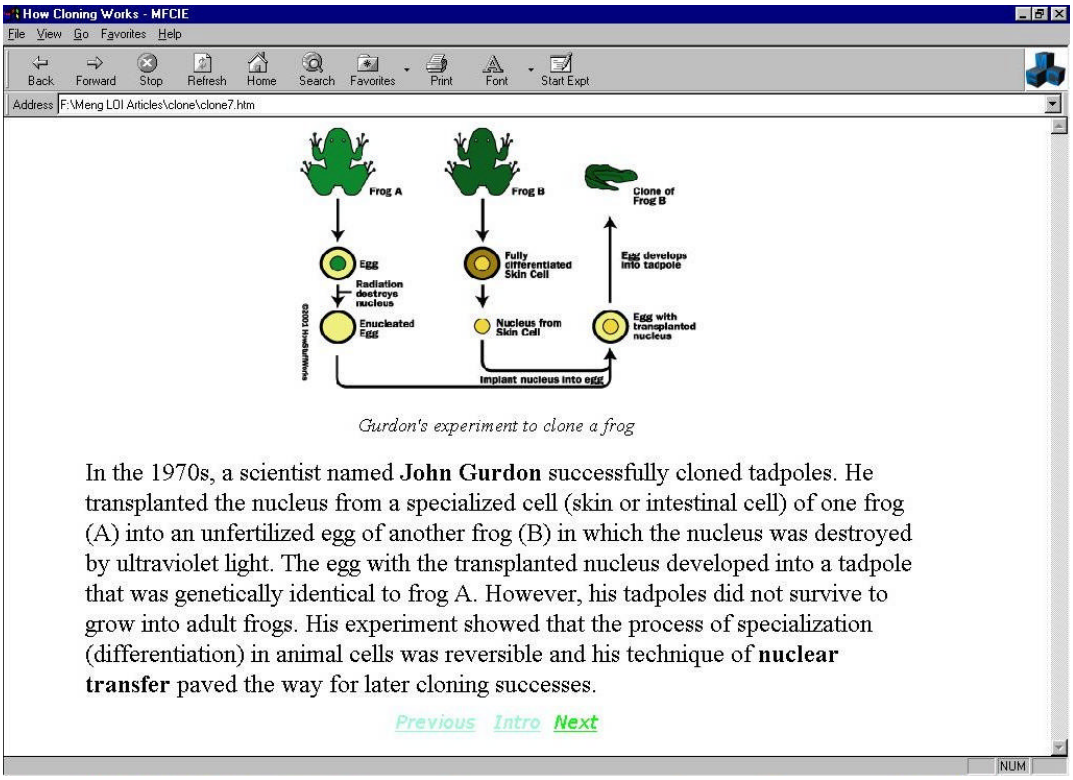

How Laughter Works - MFCIE
File View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Start Expt
Address F:\Meng LOI Articles\laugh\laugh10.htm

For example, Derks traced the pattern of brainwave activity in subjects responding to humorous material. Subjects were hooked up to an electroencephalograph (EEG) and their brain activity was measured when they laughed. In each case, the brain produced a regular electrical pattern. Within four-tenths of a second of exposure to something potentially funny, an electrical wave moved through the cerebral cortex, the largest part of the brain. If the wave took a negative charge, laughter resulted. If it maintained a positive charge, no response was given, researchers said.

During the experiment, researchers observed the following specific activities:

- The **left side of the cortex** (the layer of cells that covers the entire surface of the forebrain) analyzed the words and structure of the joke.
- The brain's large **frontal lobe**, which is involved in social emotional responses, became very active.
- The **right hemisphere of the cortex** carried out the intellectual analysis required to "get" the joke.
- Brainwave activity then spread to the **sensory processing area of the occipital lobe** (the area on the back of the head that contains the cells that process visual signals).
- Stimulation of the **motor sections** evoked physical responses to the joke.

(a)


How Cloning Works - MFCIE
File View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Start Expt
Address F:\Meng LOI Articles\clone\clone7.htm
Frog A
Frog B
Clone of Frog B
Egg
Radiation destroys nucleus
Enucleated Egg
Fully differentiated Skin Cell
Nucleus from Skin Cell
Egg develops into tadpole
Egg with transplanted nucleus
Implant nucleus into egg
©2001 HowStuffWorks

Gurdon's experiment to clone a frog

In the 1970s, a scientist named **John Gurdon** successfully cloned tadpoles. He transplanted the nucleus from a specialized cell (skin or intestinal cell) of one frog (A) into an unfertilized egg of another frog (B) in which the nucleus was destroyed by ultraviolet light. The egg with the transplanted nucleus developed into a tadpole that was genetically identical to frog A. However, his tadpoles did not survive to grow into adult frogs. His experiment showed that the process of specialization (differentiation) in animal cells was reversible and his technique of **nuclear transfer** paved the way for later cloning successes.

(b)

Figure 5.5: Sample articles. (a) Contains bulleted text. (b) Contains images.

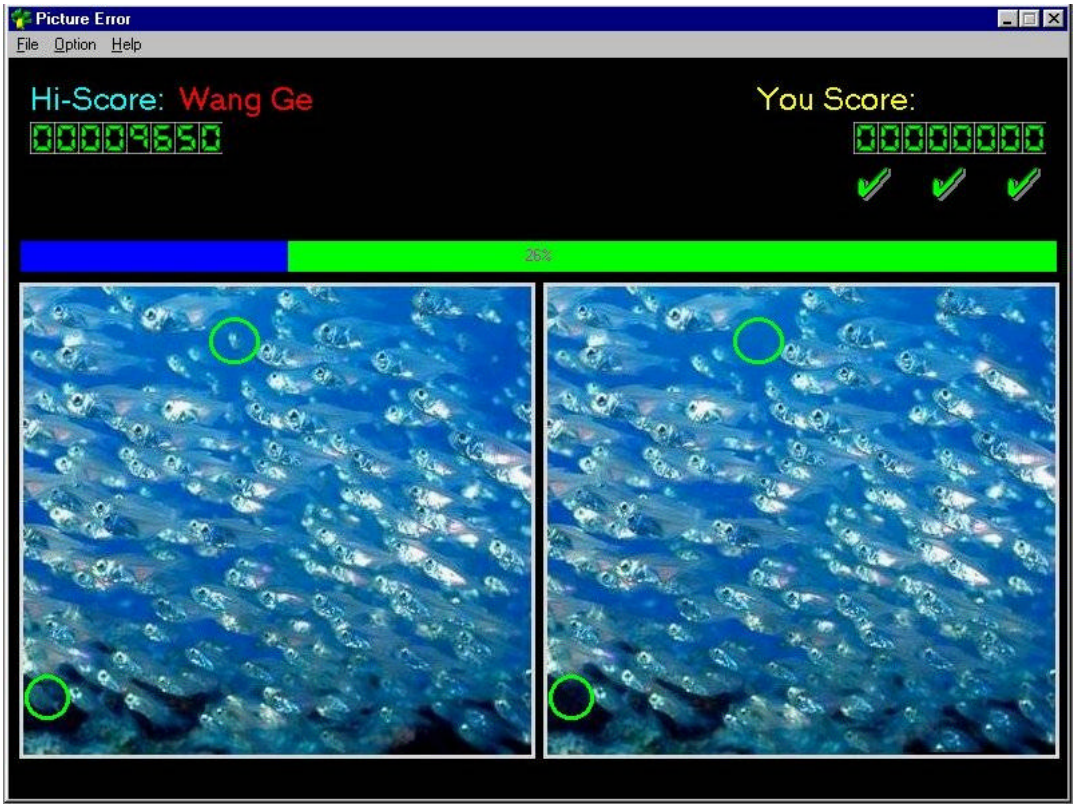


Figure 5.6: Computer game: Picture Error.

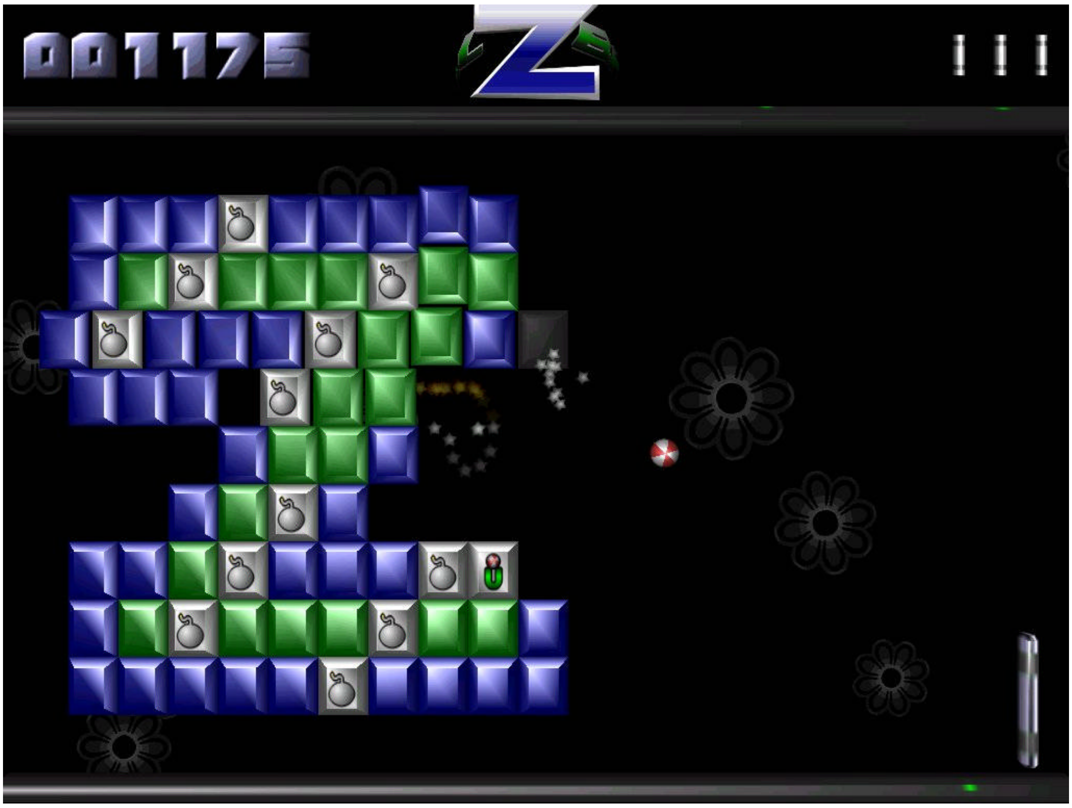


Figure 5.7: Computer game: zball.

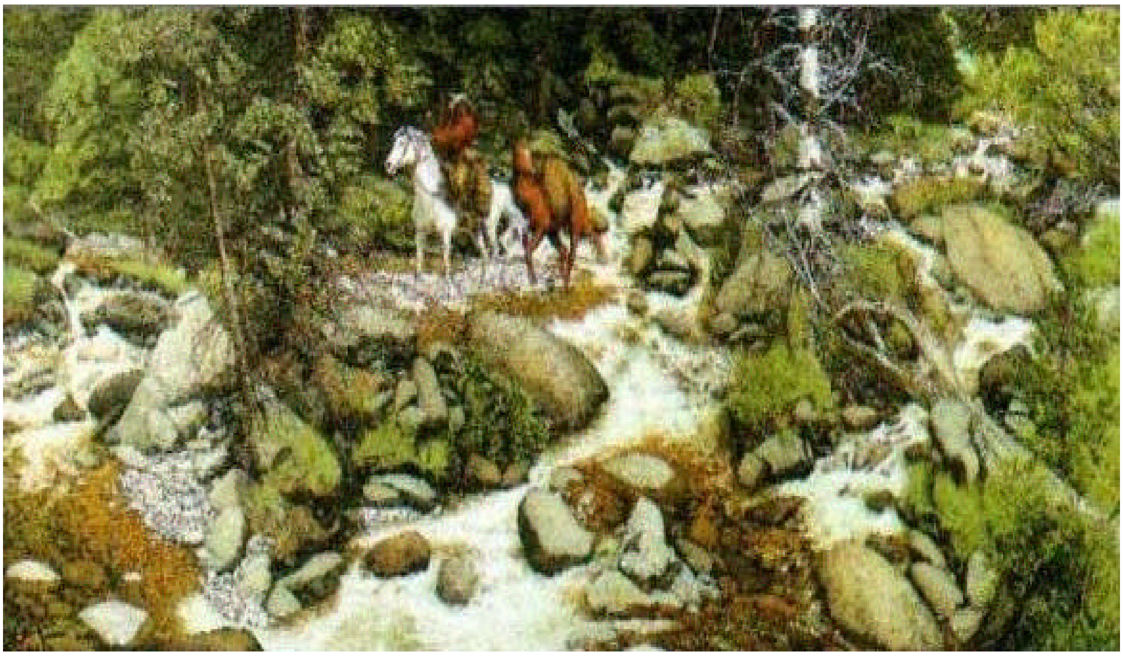


Figure 5.8: Image with embedded patterns. Eleven human faces are embedded.

5.4 Results and Discussion

A total of 10 people took part in the experiment. Detection intervals of 4, 6 and 8 seconds were tried. For noise filtering, values of 3 to 6 frames were used for “stable width” and values of 4 to 9 frames for “group widths”.

The sequences obtained were captured at 25 frames per second. One reading sequence of about 6 minutes is obtained for each person. Two sequences, on average 2 minutes long, for each of the 3 types of “non-reading” activities are acquired for each person. The sequences were processed using the modules presented in the earlier chapters to obtain the required gaze inputs for the reading module. Please refer to Figure 1.1.

5.4.1 Training the database of FSMs for reading

The database of FSMs for reading is constructed during training by using only reading sequences. For training purposes, gaze samples from the sequences are obtained by shifting a window, having a size of 4, 6 or 8 seconds, every 25 frames, which corresponds to a 1 second shift. The “leave-one-out” strategy is used. In this strategy, samples from 9 people are used for training leaving one person out exclusively for testing. This procedure is repeated in a round-robin manner, in this case 10 times. Only 70% of the samples are collected from each of the 9 people for training. An example of a FSM for reading is shown in Figure 5.9.

State	Min Dwell Time	Max Dwell Time
2	1	20
1	1	15
2	3	33
1	1	28
2	4	21
1	1	34
2	1	27
1	4	35
2	1	42
1	5	34
2	3	23
1	1	31

Figure 5.9: A FSM for reading of length 12. Window size of 6 seconds and a shift of 1 second are used.

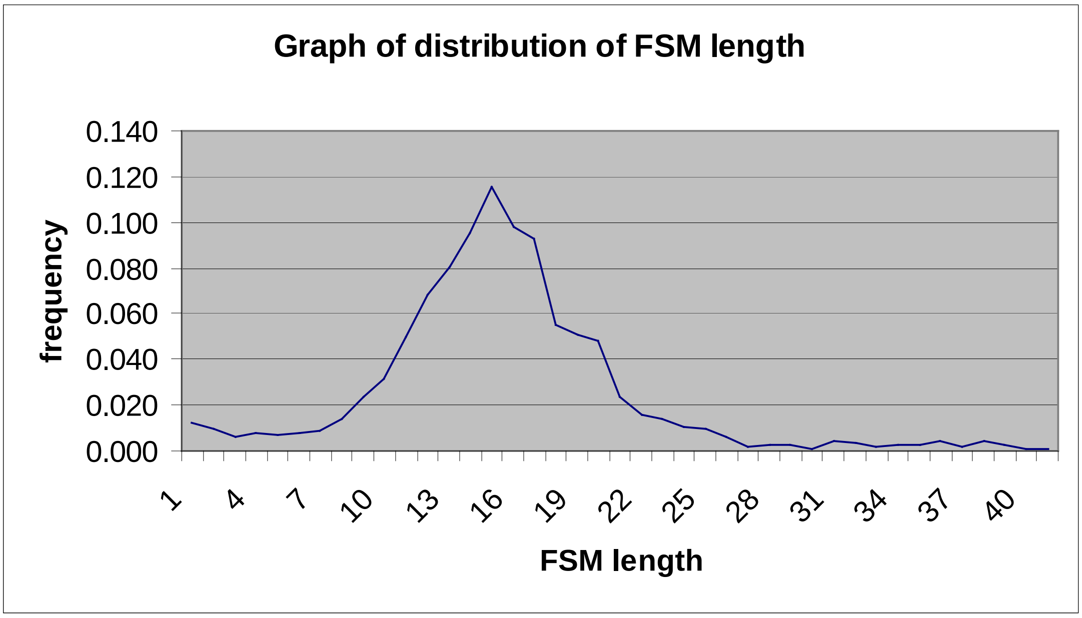


Figure 5.10: Graph of distribution of FSM length. Window of size 6 seconds and a shift of 1 second are used.

Figure 5.10 shows the distribution of the lengths of the FSMs in the database. There are about 80 distinct FSMs in the database (for each length, there exists 2 FSMs of different order). From Figure 5.10, we see that the majority of the lengths lie in the range of 12 to 20. This would correspond to reading 6 to 10 lines in 6 seconds. The number of FSMs having lengths that are less than 7 and greater than 28 are relatively few. FSMs that have lengths greater than 28, are most likely obtained from noisy gaze inputs.

As for the dwell time parameters, for FSMs having lengths less than or equal to 20, the maximum amount of dwell times does not change drastically from one state to the other of the same FSM. In fact, the largest difference between the longest and shortest dwell times is not more than 3 times. As the subjects read at their “steady state” pace, we would expect the amount of time spent in each state to be approximately the same except during situations when they are engaged in “non-reading” activities such as studying an embedded image in the text.

5.4.2 Tests

Three types of tests are performed for each cycle of the “leave one out” strategy. In test 1, the remaining 30% of reading samples of the 9 people are used for testing. In test 2, reading sequences from the “left-out” person are used for testing. In test 3, the non-reading sequences of all the 10 people. It was found that the results for detection interval of 6 seconds, stable width and group width both of 5 frames gave the best results and are shown in Table 7.1.

Table 7.1 Results of reading determination for detection interval of 6 seconds, stable width and group width of 5 with reading detection performed periodically every second. Results presented are the average of the 10 round-robin rotations of the “leave one out” strategy. Result for non-reading accuracy is applicable to all the 10 people.

Average accuracy of reading, or missed detections, for the remaining 30% of the reading samples of the 9 people used in training (test 1)	88.3%
Average accuracy of reading, or missed detections, for reading samples of subjects that are left out in each round, i.e., not used in training (test 2)	82.4%
Average accuracy of non-reading sequence from all 10 subjects in each round, or false alarms (test 3)	85.8%
Average total accuracy for people used in training (combining both reading and non-reading sequences)	87.0%
Average total accuracy for people not used in training (combining both reading and non-reading sequences)	84.1%

If “stable width” is too low and “group width” too high, a great deal of smoothing will be performed. The non-reading gaze patterns become more similar to the reading gaze patterns. Although this helps to bridge the differences in the reading habits of different individuals, unfortunately, this increases false alarms, i.e., non-reading data are wrongly detected as reading.

When the detection interval is increased, more data is considered during detection. Compared to a shorter detection interval, the gaze patterns are more varied and thus more different from one another. Differences in individual reading style are accentuated and thus the accuracy for reading drops. On the other hand, when the interval is decreased, less data is considered and thus, the possibility of variation decreases. The gaze patterns become less different from each other. As a result, false alarms will increase.

Campbell, et. al. [Campbell2001] reported a high (nearly 100%) accuracy rate using the pooled evidence algorithm in 2 experiments conducted on 4 and 5 people separately. In their experiments, a chin rest is used to stabilize the participant’s head. As compared to ordinary camera used in this project, they used infra-red camera which allowed to determine gaze directions more accurately.

5.5 Conclusion

A novel method of using FSMs for reading determination has been proposed. By considering only 2 possible gaze directions, i.e. looking at the left or right side of the monitor, the number of transitions and amount of dwell time on each direction are modeled using FSM. The method has been successfully tried, with an overall accuracy of 84.1%, on 10 different people engaged in reading and other common activities, e.g. playing computer games, watching video clips, at the computer.

Chapter 6

Conclusion and Future Work

In this work, we proposed an algorithm to differentiate reading activities from other non-reading ones, such as playing computer games and watching video clips, of a computer user and we have achieved an average accuracy of 84.1% on 10 people using the "leave one out" strategy.

In the proposed algorithm, firstly, face localization is carried out combining skin detection and face detection. Face detection has an accuracy of 97.8% and a low false alarm rate 0.002%. Using the upper-half of the detected face, eye localization is performed. Two methods are employed independently. The first method uses colour information of the iris and sclera and it gives an accuracy of 97.4%. The second method relies on blink detection, which has 97.0% accuracy, for eye localization. Once the eyes are localized, they are tracked using KLT tracker. One localized or tracked eye of each frame is then used to determine the person's gaze direction, either looking at the left or right half of the monitor. Determining the gaze directions has an accuracy of 89.8%. The gaze directions are then used as inputs to reading detection. FSMs are used to model transitions in these gaze directions over an interval using only reading samples. Intervals of gaze directions are checked against the trained FSMs and an aggregate of the results of this detection is considered to determine whether reading is taking place.

Some future work includes making the gaze module person independent, allowing the users more freedom of movement and making the system robust to these movements and lastly, making the system cope with instances when the users look out of the monitor momentarily.

References

[Adams1995] Adams, M.A., & Bruck, M. Resolving the great debate. American Educator, 19(2), 7, 10-20, 1995.

[Baluja1993] S. Baluja and D. Pomerleau, , "Non-Intrusive Gaze Tracking Using Artificial Neural Networks," Working Notes: AAAI Fall Symposium Series, Machine Learning in Computer Vision: What, Why and How?, 1993.

[Benn1997] D. E. Benn, M. S. Nixon, and J. N. Carter, “Robust eye centre extraction using the Hough Transform,” in Proc Proceedings of 1st Int. Conf. On Audio-and-Video-Based Biometric Person Authentication (AVBPA97), p3-9, 1997.

[Betke1999] M. Betke and J. Kawai, "Gaze detection via self-organizing gray-scale units", in Proc. of the Int. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pg 70-76, Kerkyra, Greece, Sep 1999, IEEE.

[Betke2000] M.Betke, W. J. Mullally and J. J. Magee, “Active Detection of Eye Scleras in Real Time”, IEEE CVPR Workshop on Human Modeling, Analysis and Synthesis, HMAS 2000, Hilton Head Island, SC, June 2000.

[Birchfield1996] Birchfield, S. Derivation of Kanade-Lucas-Tomasi Tracking Equation unpublished, May 1996. Available on the website: http://vision.stanford.edu/~birch.

[Black1996] M. J. Black and P. Anandan. The Robust Estimation of Multiple Smooth Motions: Parametric and Piecewise – Smooth Flow Fields Computer Vision and Image Understanding Jan 1996 pg: 75-104.

[Black1998] M.J. Black, D.J. Fleet, Y. Yacoob. A framework for modeling appearance change in Image Sequences Computer Vision 1998, pg: 660-667.

[Cormen1990] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, "Introduction to algorithms", McGraw-Hill, 1990.

[Campbell2001] Campbell, C. S. & Maglio, P. P. “A robust algorithm for reading detection”. In Proceedings of the ACM Conference on Perceptive User Interfaces. 2001.

[Colmenarez1997] A. J. Colmenarez and T. S. Huang, “Face Detection With Information-Based Maximum Discrimination”, Computer Vision and Pattern Recognition, p 782-787, 1997. Proceedings., 1997 IEEE Computer Society Conference on, 1997.

[Fang1994] M. Fang, A. Singh, and M.Y. Chiu, “A fast method for eye localisation”, Siemens Corporate Res., Tech. Rep. SCR-94-TR-488, 1994.

[Gee1994] A. Gee and R. Cipolla, "Determining the gaze of faces in images", Image and Vision Computing, 12(18):639-647, 1994.

[Grauman2001] K. Grauman , M. Betke, J. Gips and G. Bradski Communication via Eye Blinks- Detection and Duration Analysis in Real Time CVPR 2001, Vol 1 pg:1010-1017.

[Haykin1999] Haykin, S, “Neural Networks: A Comprehensive Foundation”, 2nd edition, pg 299, Prentice-Hall Inc, 1999.

[Hong2000] P. Hong, M. Turk, and T. Huang, "Gesture modeling and recognition using finite state machines," Proc. IEEE International Conference on Face and Gesture Recognition, Grenoble, France, March 2000.

[Isard1996] M. Isard and A. Blake, “Contour Tracking by Stochastic Propagation of Conditional Density”, Proc. European Conf. Computer Vision, pp. 343-356, 1996.

[Jones1999] M. Jones, and J. Rehg, “Statistical Color Models with Application to Skin Detection”, Computer Vision and Pattern Recognition, Vol. 1, pp. 274-280, 1999.

[Lucas1981] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. International Joint Conference on Artificial Intelligence, pages 674-679, 1981.

[Maglio2000] Maglio, P. P., & Campbell, C. S. (1999). Tradeoffs in displaying peripheral information. In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2000). New York: ACM Press.

[Menser2000] B. Menser, and M. Wien, “Segmentation and Tracking of Facial Regions in Color Image Sequences”, SPIE VCIF 2000 Vol 2, pp. 731 – 740, 2000.

[Moghaddam1995] Moghaddam and A. Pentland, "Maximum Likelihood Detection of Faces and Hands"," Int. Workshop on Automatic Face and Gesture Recognition, pp122-128, Zurich, 1995.

[Qayedi2000] Al-Qayedi A.M. and Clark A.F. Constant-rate eye tracking and animation for model-based-coded video, ICASSP ’00 pg: 2353-2356.

[Rikert1998] T. Rikert and M. Jones, "Gaze estimation using morphable models", in Int. Conf. on Automatic Face and Gesture Recognition, 1998.

[Rosales1998] R. Rosales, and S. Sclaroff, “Improved Tracking of Multiple Humans with Trajectory Prediction and Occlusion Modeling”, IEEE Conf. on Computer Vision and Pattern Recognition, Workshop on Interpretation of Visual Motion, 1998.

[Rowley1996] H. Rowley, S. Baluja and T Kanade, "Neural Network-based Face Detection", CVPR, 1996.

[Schiele1995] B. Schiele and A. Waibel, "Gaze tracking based on face colour", in Int. Workshop on Automatic Face and Gesture Recognition, 1995.

[Smith2000] P.Smith, M.Shah, N. daVitoria. LoboMonitoring Head-eye motion for driver alertness with one camera Pattern Recognition, 2000 pg: 636-642.

[Sung1994] K. Sung and T. Poggio, "Example-Based Learning for View-Based Human Face Detection", A.I. Momo 1521, CBCL Paper 112, MIT, December 1994.

[Tomasi1991] Tomasi, Carlo and Takeo Kanade. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[Varchmin1997] A. C. Varchmin, R. Rae and H. Ritter, “Image based Recognition of gaze direction using adaptive methods”, In I. Wachsmuth, editor, Proc. Int. Gesture Workshop, pages 245-257. Springer, 1997.

[Yang1994] G. Yang and T.S. Huang, "Human Face Detection in a Complex Background", Pattern Recognition, Vol 27, No 1, pp53-63, 1994.

[Yang1997] J. Yang, W. Lu and A. Waibel, “Skin-Color Modeling and Adaptation”, In: Computer Vision - ACCV'98, Vol. 2, Berlin 1997. S. 687-694.

[Yano1999] K.Yano , K.Ishihara, M. Makikawa, H. Kusuoka Detection of eye blinking from video camera with dynamic ROI fixation IEEE SMC ’99 Vol 6 pg:335-339.