Iris Tracking and Blink Detection for Human-Computer Interaction using a Low Resolution Webcam

Rupal Khilari*
Department of Information Technology,
S.I.E.S. College of Arts, Science and Commerce
Sion (West), Mumbai
Maharashtra, India
rupal.khilari@gmail.com

ABSTRACT

This paper proposes a technique to develop a non-intrusive interface for human-computer interaction based on visual cues of a human eye, captured using a low resolution USB camera. Such an interface can replace a traditional interface like a mouse thus helping the severely disabled to use the computer just as a normal individual would. It makes use of iris tracking and blink detection for this purpose. Considering fairly constant lighting conditions, a modified Hough Transform is used to track the iris at real time. The *Between-The-Eyes* feature serves as a stable reference point for gaze direction estimation. When mapped to the computer screen, incremental, directional movement of the mouse pointer position provides precise control over its movement. A method based on the projection function is proposed to detect the user's eye blinks, analyze their patterns and duration, interpreting them as appropriate mouse clicks. The system gives an overall accuracy of 87.4 %.

Keywords

Blink Detection, Iris tracking, alternative mouse interface, human-computer - interface, gaze estimation

1. INTRODUCTION

Today, the mouse and keyboard are the main interfaces for communicating information and commands to the computer. The use of these traditional human-computer interfaces demand good manual dexterity and refined motor control. However, there are many people, who due to their physical disabilities such as cerebral palsy or quadriplegia are unable to use the mouse and the keyboard, and therefore are restricted in their attempt to use the computer as a physically normal individual would. Due to their immobility, head movements or gestures as a means to control the mouse may be infeasible. Speech controlled systems may also fail in situations when the individual does not have the ability to speak, or the environment is noisy. In such situations, ocular movement can be presented as an effective, alternate solution.

There have been several advances in recent years that allow appliances to sense the user's pupil movement and expressions. Some assistive technologies that provide an interface for mouse control use expensive, high resolution video cameras and infrared light. Intrusive interfaces such as headgear attached with low resolution webcams provide an accurate means of tracking, but at the same time, are uncomfortable to wear. The need is to develop a non-intrusive interface that allows easy communication with the computer through ocular movement captured using a cost-effective webcam.

Deformable model-based methods and snake models for eye feature description have been presented in the recent years [2]. Though generally accurate and generic, they are computationally demanding, require high contrast images and pose problems in large eye shape variations.

The goal of iris tracking is accomplished by using a modified form of the Weighted Hough transform. This technique uses a circular mask over the probable candidate iris-center points from the resultant accumulator. This increases the accuracy of iris tracking to a large extent. Although small head movements are accounted for using correlation with an online eye template, significant changes in the head position result in automatic reinitialization of the system.

Michael Chau and Margrit Betke presented a system [4] that used USB cameras for issuing mouse clicks using blink detection based on the technique proposed by Grauman et al.[11]. It was observed that drastic changes in the iris position (extremes) caused a fall in the correlational scores below the suggested threshold thus resulting in false positives. Advanced methods of eye state detection based on Gabor Wavelets, neural networks [18] and Active Appearance Models [7] have been proposed, that require large scale offline image training. Rainer Stiefelhagen and Jae Yang proposed a system that performed gaze tracking taking into account the position and orientation of the head [14]. Methods using stereo calibration and regression have also been proposed for this purpose [5].

This paper proposes an alternative solution that uses a regular cost-effective USB camera (webcam) and ordinary light. Here, a computationally inexpensive method of blink detection in constant lighting conditions based on the projection function is introduced that requires no offline train-

*Corresponding author

ICVGIP '10, December 12-15, 2010, Chennai, India

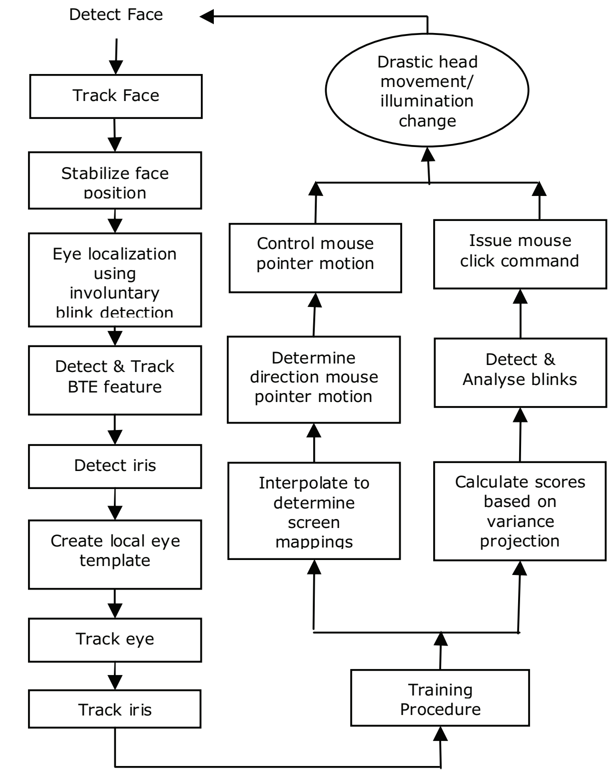


Figure 1: An Overview of the proposed algorithm.

ing. A robust 2D regression based method is used for gaze estimation.

2. PROCESSES

The proposed algorithm is initialized on detecting a face from the USB camera feed, under satisfactory illumination. When the position of the user is sufficiently constant, the system for detecting and analysing blinks and mouse movements is initialized automatically, depending on the involuntary blink of the user. This stage is followed by the extraction of the Between-The-Eyes (BTE) feature and the detection and tracking of the iris of the eye. A local template of the open eye is used for the subsequent tracking of the eye. On performing the training procedure required for each session, scores based on variance projection as well as the relative positions of the iris are analysed and interpreted to perform the various mouse functions accordingly. This analysis is performed at each frame. A block diagram depicting the main stages of the algorithm is shown in Figure 1.

2.1 Face Detection and Tracking

It is assumed that a frontal face is initially detected and tracked using an appropriate procedure. The prototype 'I-Click' uses the face detection procedure built into OpenCV based on Haar-like feature classifiers.

2.2 Eye Localization

Once the face is detected and stabilized, it is necessary to locate the eyes in order to track the iris and analyze blinks. An involuntary eye blink triggers the eye localization process [4, 11]. To accomplish this, the difference image of the head region of consecutive frames is created and then thresholded with a suitable threshold value. This results in a binary image showing the regions of movement between the two frames.

Next, a 3x3 star-shaped kernel is passed over the binary difference image in an opening morphological operation. This helps to eliminate a substantial amount of noise and naturally-occurring jitter that is present around the user in the frame that could occur due to the lighting conditions, the camera resolution, or slight background movement. The opening operation results in fewer connected components in the vicinity of the eyes when a blink occurs.

A recursive labelling procedure is then applied to the upper half of the image (considering the eyes to be present in the upper half of the head region) to recover the number of connected components in that region. Since the users, due to paralysis are relatively still, this procedure yields only a few connected components, with the ideal number being two (the left eye and the right eye). In case other movement in the face region has occurred, producing an unusually large area of the components, the system discards the current binary image and waits to process the next involuntary blink in order to maintain efficiency and accuracy in locating the eyes.

Each labelled component can be thought of as a possible match for the user's left or right eye. The filtering of an unlikely eye is done based on the width to height ratio and the area of each component. For example, if the width to height ratio is very small, or the area is very large, the component is unlikely to be the user's eye. From the components that pass this test the largest component (in effect, the eye that is more clearly visible) is selected. Thus either the right eye region or the left eye region is localized, since tracking the iris of both the eyes would lead to unncessary real-time

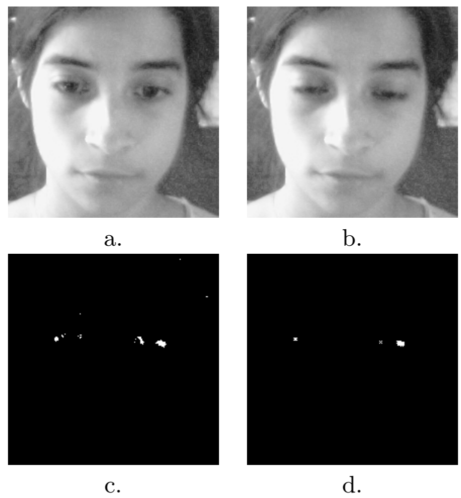


Figure 2: Motion Analysis for eye localization (a) Face at frame f. (b) Face at frame f+1. (c) Difference between the two frames f and f+1 after thresholding operation. (d) Difference image after the opening operation with reduces noise, used to locate the eyes.

Figure 3: Between-The-Eyes feature

processing and redundant information. The appropriate 'R', 'G' or 'B' channel of the eye can be extracted based on the colour of the iris so as to obtain maximum contrast between the iris and the sclera.

2.3 BTE Feature Detection and Tracking

Some room for head movement should be accounted for without resulting in errors in mouse cursor movement. It is necessary to distinguish head movements from iris movements. Thus tracking of the head to a small extent is necessary. A feature that is unique and robust should be selected as a reference point for tracking to compensate for minor head movements. These facial features could be point features (eye corners), edge features (lip contours) or texture features (skin color). However, the feature should be large and clear enough to be tracked with minimum error. For this, the Between-the-Eyes (BTE) feature is chosen [9] since it is unique on one face, visible under pose variation, and fairly stable despite possible change of facial expressions. It has a relatively bright part at the nose-bridge and relatively dark parts at the eyes like wedges on both sides.

This BTE feature is extracted on the basis of anthropometric measures after locating the eye region. BTE tracking is performed based on the Optical flow technique devised by Kanade and Lucas, proposed by Poelman, as implemented in [15] . This is done by minimizing a residual function $\epsilon(x, y)$ based on the image intensity $I(x, y)$ at a point (x, y)

$$\varepsilon = (I(x, y) - I(x + dx, y + dy))^2$$

A window as large as the allowed head movement is employed to determine the motion of the BTE feature from one frame to the next. This larger window is the neighbourhood region $R(x)$ of the point ϵ. This becomes the problem of minimisation of the residual function ϵ for the entire region R . In discrete form, if the width of the integration window is $(2w_x + 1)$ and its height is $(2w_y + 1)$, the expression for the calculation of the residual function ϵ is:

$$\varepsilon = \sum_{x'=x-w_x}^{x+w_x} \sum_{y'=y-w_y}^{y+w_y} (I(x', y') - I(x' + dx, y' + dy))^2$$

The best match between the features in successive pairs of images is determined by finding the values of dx and dy that minimize the residual function ϵ.

2.4 Iris Detection and Tracking

Although methods using deformable templates and Active Appearance Models have been developed in recent years for iris tracking in visible light, they require high resolution cameras and are very computationally intensive . The Iris can be accurately detected and tracked at real time using a modified form of the Hough Transform. The Circular Hough Transform (CHT) takes advantage of the fact that a human

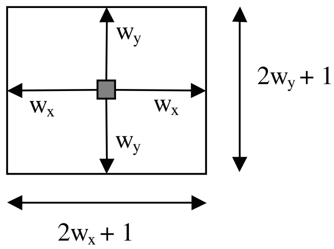


Figure 4: 2D neighbourhood of pixel (x, y) used to minimise residual function ϵ

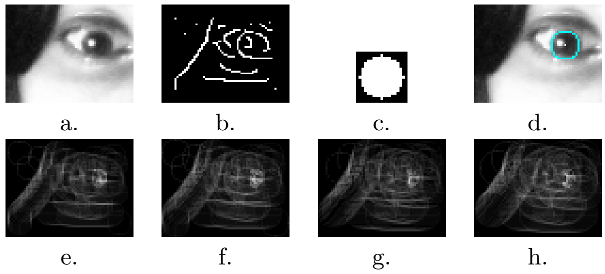
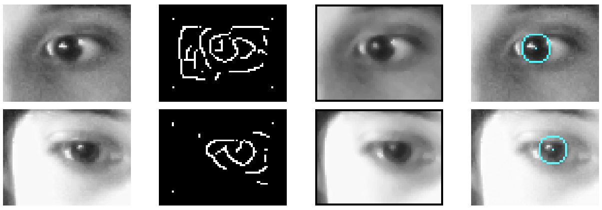


Figure 5: Implementation of Iris detection (a) The eye region. (b) The corresponding edge image after Canny edge detection. (c) A circular mask of various radii is passed over the largest value in accumulator images and the average over the area is computed. (d) The resultant iris detection (blue circle) using the modified weighted CHT. (e,f,g,h) Accumulator images using Bresenham's circle drawing algorithm (intensity uniformly scaled), with radii 6,7,8,9 pixels respectively.

iris is circular in shape [13, 10, 12] . It allows discovering shapes from image edges. The edges in an image can be found using the Canny, Sobel or Morphological operations. On performing unsharp masking, the Canny detector is applied, since it produces a more precise, finer edge with lesser noise.

Every edge point in the new image is processed by drawing circles with the desired radii (within a predetermined range) and incrementing the circumference values in the accumulator (as shown in Figure: 5). The initial range of radii is fixed based on the size of the detected face. Once the initial radius of the iris is determined, this range is narrowed down to use radii values comprising of the initially found value ± 2 pixels. The accumulator image is created based on a weighted voting system favouring the pixels with lower intensity value that are likely to be a part of the iris border. To optimize computation for real-time processing, the Bresenham's circle drawing algorithm is adopted that uses the concept of mirroring in all octants to draw a complete circle. When every edge point and every desired radius is used, the accumulator images are analyzed. The accumulator will now contain numbers corresponding to the number of circles passing through the individual coordinates. Thus the highest numbers correspond to the centre of the most probable circle in the image.

Now the appropriate circle feature that would represent the iris has to be chosen. The use of a ring shaped gra-



a. b. c. d.

Figure 6: Iris Detection (a) The eye region. (b) The corresponding edge image after Canny edge detection. (c) The 3x3 median filtered eye region. (d) The resultant iris detection (blue circle) using the modified weighted CHT.

a. b.

Figure 7: (a) Iris detection using the standard weighted Hough transform. Notice the influence of the upper eyelid in determining the centre and radius of the iris. (b) Iris detection using the modified form of weighted Hough transform

dient mask to extract pupil features from a face image is proposed in [17]. As a real-time variation, the eye template is filtered using a 3x3 median filter to reduce the contribution of the corneal reflection that could cause an error in detecting the iris. A circular mask of the same size as the radius is used to compute the average intensity of the pixels within the circular area. The lower the intensity, the higher is the probability that the iris is located. To avoid shrinking towards the smallest radius feature that could be the pupil, the largest radius feature that lies within a fixed range of the lowest average intensity is selected as the iris. Applying this process on each frame results in accurate iris tracking independent of the previous frame.

When standard weighted Circular Hough Transform is used to detect the iris, the eyelids and eyelashes often affect the final result. Thus the Hough transform is modified to extract the iris centre and radius, which can reduce the effect of the eyelids and eyelashes.

2.5 The Local eye template

The eye region located is fairly larger than the actual eye. Thus the user should be allowed limited head motion as long as the eye remains within this region. When the user is required to look at the centre of the screen as a part of the training procedure, an eye template is extracted based on the iris position and size. The details and need of this training procedure will be outlined later. This eye template is typically smaller than the eye region and allows the user some freedom to move around slightly. It serves to restrict the region of the eye image that is searched in order to detect the iris or evaluate a blink, thus reducing errors and unnecessary processing.

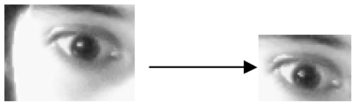


Figure 8: Eye region when the user looks at the centre of the screen and the reduced local eye template.

2.6 Local eye tracking using normalized cross correlation

After creating the local eye template, an eye tracking procedure maintains exact knowledge about the eye's appearance. A simple tracking algorithm suffices to update the region of interest centered around the eye region.

In order to track the template, the system utilizes the normalized correlation coefficient R proposed by Grauman et.al [11] as follows:

$$R(x,y) = \frac{\sum_{y'=0}^{h}\sum_{x'=0}^{w} \mathrm{T}(x',y')\mathrm{I}(x+x',y+y')}{\sqrt{\sum_{y'=0}^{h}\sum_{x'=0}^{w} \mathrm{T}(x',y')^2 \sum_{y'=0}^{h}\sum_{x=0}^{w} \mathrm{I}(x+x',y+y')^2}}$$

where T(x',y')=$T(x',y')-\bar{T}$, I(x+x',y+y')=$I(x+x',y+y')-\bar{I}(x,y)$ and $T(x,y)$ and $I(x,y)$are the brightness of the pixels at (x,y) in the template and source image, respectively, and $\bar{T}$ is the average value of the pixels in the template raster and $\bar{I}(x,y)$ is the average value of the pixels in the current search window of the image. The coefficient $R(x,y)$ is a measure of the match between the open eye template and all points within the small search region surrounding the location of the eye given from the previous frame. R is maximized to find the closest match to the open eye template. In this way, the current eye position is updated nearly twenty times per second.

2.7 Blink Detection

Motion analysis alone cannot provide accurate information about an eye blink and its duration. Relying on motion would make the system extremely intolerant of extra motion due to facial expressions or head movement.

Although blink detection is possible through optical flow computation [3], it is less tolerant to vertical movements of the iris and the head. As in the system presented by Grauman et al. [11] the correlation scores obtained in the previous step may also be used to determine the occurrence of a blink. Usually $0.55 < R < 0.85$ is taken as the range that confirms a blink. However it has been observed that these thresholds vary depending on the size of the eye template as compared to the region. Also at times, the scores fall within this range when the iris moves to the eye extremes though the eyes are opened. Hence a very high threshold of 0.93 is used to only confirm an open eye and to avoid actual processing for blink detection.

A novel approach to blink detection has been proposed based on an idea implemented by Unuzova [16] .

First, the horizontal integral projection H of the local template of the eye is computed. Thus, for each line y of the image the following sum is calculated :

$$H(y) = \frac{1}{w}\sum_{i=1}^{w} I(x_i, y)$$

Where w is the width of the image, $I(x_i, y)$is the intensity value in the red channel of (x_i, y)pixel. The difference can be detected by a variance projection function. It is defined by:

$$\sigma_H^2(y) = \frac{1}{w} \sum_{i=1}^{w} [I(x_i, y) - H(y)]^2$$

Where H(y) is the mean value for a row.

In order to eliminate the noisy projections, a one dimensional median filter, with mask size of T=5 is used to smooth the variance projection. It can be found using:

$$y(t) = median\left(x\left(t - \frac{T}{2}\right), .., x(t), x(t+1), .., x\left(t + \frac{T}{2}\right)\right)$$

Where y is the median filtered image and x is the original image.

When the eye is opened, the variance scores correspond to the borders between eyelid-sclera and sclera-iris. In some cases, a high variance is also caused due to the corneal reflection produced in the eye due to light. The falling and rising values of the variance projection function can be found as minimums and maximums (extremes) of the first derivative. The absolute sum of the extreme values of the first derivative (B) is very large when the eye is open. However, when the eye closes these scores fall very low since a major portion of the area in question has uniform intensity of that of the skin. Since the variance scores vary to a large extent depending on the contrast exhibited by the webcam feed or the lighting conditions, a threshold is computed for each session taking into account the current conditions.

$$B = \left|\frac{dy}{dx_{\max}}\right| + \left|\frac{dy}{dx_{\min}}\right|$$

A training procedure for each session is used by the prototype 'I-Click' where several parameters are evaluated based on the current conditions. The user is required to look at five vital points on the screen (i.e centre, right, left, top and bottom) for a duration of 2 seconds. The value of B as well as the iris-BTE vectors are computed for each of the five points. The threshold is taken to be a fraction (1/10) lower than the minimum computed value.

2.8 Mapping the vectors to the screen for gaze estimation

The training procedure to find the iris-BTE vectors is necessary since the deflection of the iris to gaze at the extremes of the computer screen varies on each run depending on the position of the user, the distance of the user from the screen and the zoom factor of the webcam. The distance of the iris centre from the BTE feature is captured and mapped to fixed positions on the screen.

As noted in [8], the accuracy of gaze tracking greatly depends upon the resolution of the eye images captured by the webcam. Consider the cross section of a human eye as shown below :

Suppose the size of an eye image is 30×20, which is a situation when the person is from 60 cm from the screen, the range of the iris movement from its pupil is about 16 pixels. In the figure, suppose $|l| \in [0, 8]$ pixels, the radius of the eyeball r = 20 pixels. The smallest level unit of l is $|\Delta l|$ = 1 pixel. Then,

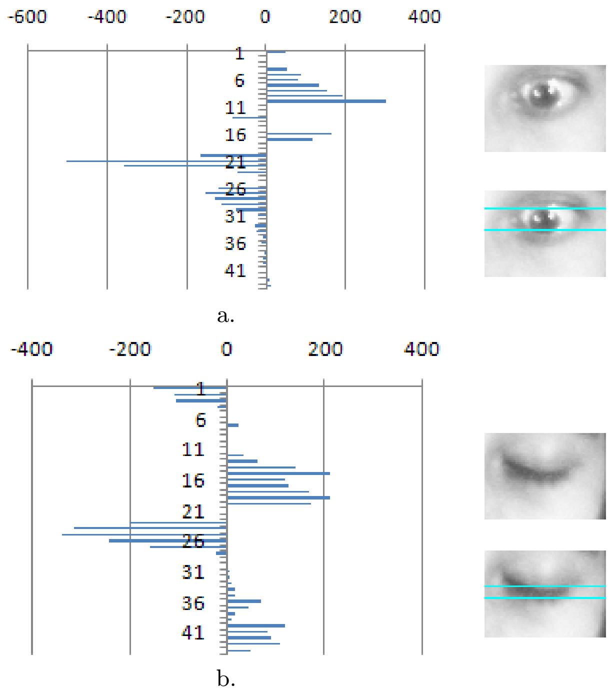


Figure 9: High scores are produced when the eye is opened, in this case B=807.61 (b) Scores lower than the threshold B_{thresh}=635.79 produced when a blink occurs, in this case B = 522.62

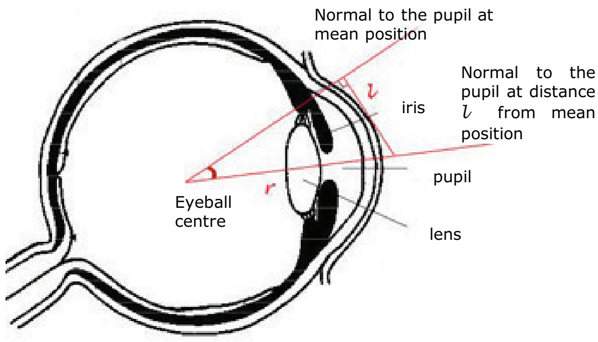


Figure 10: Cross section of the human eye for error analysis in gaze estimation

$$\theta = \arcsin\left(\frac{l}{r}\right)$$

$$\Delta\theta \approx \frac{\partial\theta}{\partial l}\Delta l + \frac{\partial\theta}{\partial r}\Delta r$$

$$= \frac{1}{r\sqrt{1-\left(\frac{l}{r}\right)^2}}\Delta l - \frac{1}{r^2\sqrt{1-\left(\frac{l}{r}\right)^2}}\Delta r$$

Suppose $\Delta r = 0$, then for $|l| \in [0, 8]$ pixel, $|\Delta\theta| = [2.87^\circ, 3.12^\circ]$. Thus, the smallest unit of θ, i.e. the resolution of gaze direction is about 3.0°. If the iris detection has a small error of 1 pixel, it will generate a tracking error of about 3.0° of gaze direction. Therefore, it is hard to determine the exact point of gaze on the screen accurately with such low resolution, however the gaze region can be determined. Interpolation provides a relatively simple, robust and accurate method to do so. Here, the BTE feature is used not only as a means of tracking movements of the head but also as a stable reference point to track the relative movement of the iris thus enabling gaze estimation.

A 2-D linear mapping is performed from the iris-BTE vector to the gaze angle. Gaze directions in successive frames are calculated by interpolation. Suppose the x and y components of the gaze angle and the iris-BTE vector for fixed calibration points P1 and P2 respectively are ((α_1,β_1),(x_1 ,y_1)) and ((α_2 ,β_2),(x_2 ,y_2)), then if the instantaneous BTE feature centre-iris vector is (x,y) the corresponding gaze angles are calculated as follows :

$$\alpha = \alpha_1 + \frac{x - x_1}{x_2 - x_1}(\alpha_2 - \alpha_1)$$

$$\beta = \beta_1 + \frac{y - y_1}{y_2 - y_1}(\beta_2 - \beta_1)$$

This method is fast, easy, and adaptive. Here the gaze angles can conveniently map to coordinates on the screen to control the position of the mouse pointer.

When the values of α and β are directly mapped to the screen as shown, the mouse pointer position appears to have a large variation for each consecutive sample, since the iris movement is small and limited in low resolution images. In case of an error of 1 pixel, the gaze angle error is magnified to a great extent when mapped to the screen. Hence accurate and precise mouse control through direct mapping would not be possible. This problem has been possibly overcome through use of incremental mouse movements in a particular direction as a result of gaze region estimation. For example, when α and β directly map to a limited region of the screen, (say 1/3rd of the left portion of the screen excluding the top and bottom 1/3rd) the mouse is moved in small increments of 20 pixels towards the left. This helps to achieve high accuracy and control even though a low resolution webcam is used to read ocular information.

3. EXPERIMENTS AND RESULTS

The prototype ‘I-Click’ was developed and tested on a Windows XP PC with an Intel Pentium IV, 3.2 GHz processor and 1 GB RAM. Video was captured with a HP 2.0 Megapixel VGA Webcam at 30 frames per second. All video

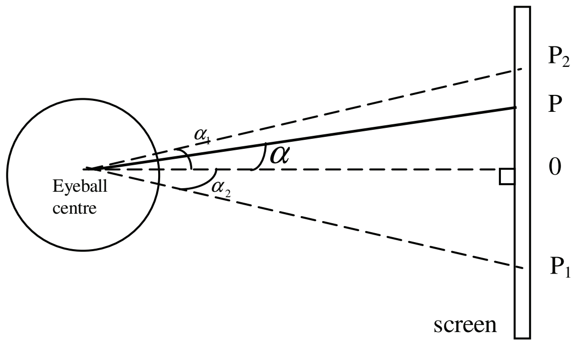


Figure 11: Mapping of gaze angle α to a point P on the screen

was processed as grayscale images of 480 x 360 pixels using Java, Java Advanced Imaging and the Intel OpenCV libraries. The system works at approximately 20 frames per second under almost constant and sufficiently bright lighting conditions. The zoom factor of the camera should be adjusted so that the user’s face comprises of about 30% to 90% of the input image. This assumption helps to fix the initial range of radii need to select the optimum radius of the iris.

Tests were conducted to gauge the accuracy and usability of the low resolution camera as an input device. In order to test the effectiveness of the modified circular Hough transform as a means to perform iris tracking, the users were asked to gaze in different directions, the tracker being displayed on each frame. All the processed frames were saved and examined individually as runs of approximately 300 frames at a time. It was found that out of 1538 frames, 18 frames showed a deviation from the iris while tracking resulting in an accuracy of 98.82%

While analyzing the blink patterns for mouse clicks, the system was tested by targeting one type of mouse click (eg. single left click) at a time, with a random number of involuntary blinks interspersed. A number of runs of a predefined number of blinks were allocated to each blink type.

Tests involving the voluntary blink length parameter were also conducted, with values ranging from 0.7 seconds to 4.5 seconds. Test runs were also conducted for mouse movement as a result of gaze estimation using predetermined random test patterns.

Nearly all of the misses and false positives in the experimental system were caused by an error in the user’s judgment of the duration of the voluntary blink to obtain the required type of mouse click. Hence it is necessary to produce test results taking into account the experimental system and overall system for blink detection. The experimental blink system measures include the experiments involving the adjustments in the voluntary blink length parameter, while the overall blink system measures disregard these outliers.

The system considers a blink of duration 0.7 to 1.5 sec as a left click, 1.5 to 3 sec as right click and 3 to 4.5 sec as a double click (upper limit inclusive).

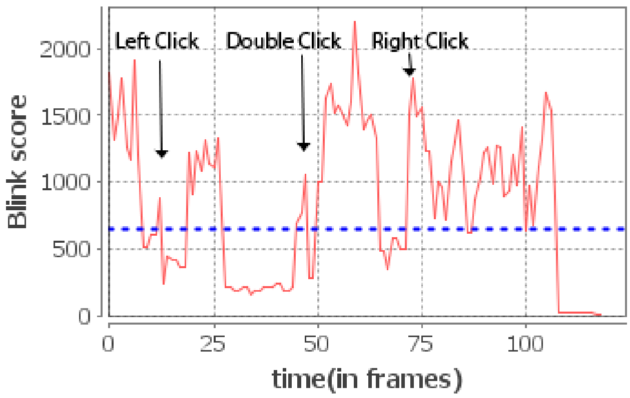


Figure 12: Shows the scores computed for blink detection based on the projection functon. The blue dotted line shows the evaluated threshold (B_{thresh}=652.96 for this session). An appropriate click is issued based on the blink duration.

The results can be summarized as follows:
For Mouse Clicks

Total blinks analyzed : 350
Overall blink system measures
Total missed blinks: 18 = 5.1%
Total false positives: 13 =3.7%
Blink Detector accuracy: 91.4%

Experimental blink system measures
Total missed blinks : 18+26 = 44 = 12.6%
Total false positives : 13+12 = 25 = 7.1%
Blink Detector accuracy : 80.3%

For Mouse motion

Total gaze directions analysed : 120
Efficiency of mouse pointer response to gaze directionis found to be as follows :
Right: 80.0%
Left: 73.3 %
Top: 86.7 %
Bottom: 93.3 %

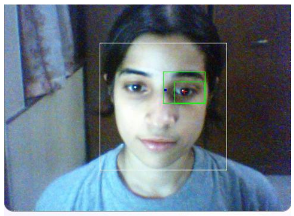
Overall efficiency of mouse pointer response to gaze estimation : 83.33 %

The system gives an overall accuracy of 87.4 % when considering the overall blink system measures.

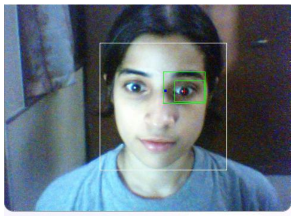
In addition, other experiments were also conducted to determine the fitness of the system under varying circumstances, such as alternative camera placements, lighting conditions, distance to the camera and people in the vicinity. The system showed good results under bright artificial light and sun light.

4. CONCLUSION

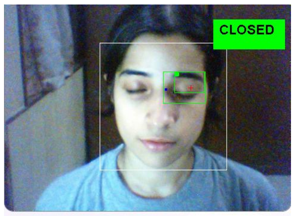
Through the implemented system an alternative assistive technology is developed that is non-intrusive, does not require specialized hardware or lighting and uses a cost-effective, low resolution webcam. It also makes an attempt



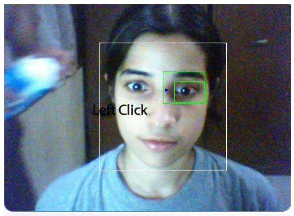
a. Frame 137



b. Frame 144



c. Frame 151



d. Frame 160

Figure 13: Frames captured over time. (a) Localized face (Outer white rectangle), localized eye region (Outer green rectangle), Local eye template (Inner green rectangle), Red cross (Iris tracker), Blue dot (BTE tracker). (b) Eye in an open state (c) Detection of eye in the closed state. (d) 'Left Click' event occurs as blink is completed.

to suggest novel ideas for accurate cognitive computer control, which can be developed further to produce an ideal state-of-the-art interface. The use of such an assistive technology by differently abled people would definitely help them to communicate, play games, browse the internet, learn and live a better life, without being completely dependent on someone else.

It can further be improved by enhancing the current techniques to include illumination invariant blink detection which is also tolerant to strong reflections due to spectacles. Procedures can be incorporated to allow considerable head movement or changes in depth. With a view to providing a complete mouse interface solution, future work aims to provide an easy means to perform the scrolling function, selection of text and drag-drop operations. In addition, further optimization, tuning and enhancement of the approaches are necessary to improve the overall accuracy of operations.

5. ACKNOWLEDGMENT

Sincere thanks to Prof. Satishkumar Chavan for his guidance and suggestions.

6. REFERENCES

[1] Report on new approaches to eye tracking. COGAIN Communication by Gaze Interaction, 2006.

[2] D. S. C. Alan L. Yuille and P. W. Hallinan. Feature extraction from faces using deformable templates. 1991.

[3] S. R. . V. Bhaskar, Foo Tun Keat. Blink detection and eye tracking for eye localization. National University of Singapore, Singapore, Indian Institute of Science, Bangalore, 2004.

[4] M. Chau and M. Betke. Real time eye tracking and blink detection with usb cameras. Boston University Boston, MA 02215, USA.

[5] S. M. Djamel Merad, Stephanie Metz. Eye gaze estimation from a video. February 2009.

[6] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. IEEE, January 2009.

[7] M. I. P. C. Ioana Bacivarov. Statistical models of appearance for eye tracking and eye-blink detection and measurement. August 2008.

[8] J. Y. Jie Zhu. Subpixel eye gaze tracking. Interactive Systems Laboratory, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

[9] S. Kawato and N. Tetsutani. Detection and tracking of eyes for gaze-camera control. 2004.

[10] M. A. Klaus D. Toennies, Frank Behrens. Feasibility of hough-transform-based iris localisation for real-time-application. University of Magdeburg, Germany, 2001.

[11] J. G. G. R. B. Kristen Grauman, Margrit Betke. Communication via eye blinks - detection and duration analysis in real time. 2001.

[12] Y. Nawaz and S. Sircar. Real time eye tracking and blink detection using low resolution web cam. 2005.

[13] S. J. K. Pedersen. Circular hough transform. Aalborg University, Vision, Graphics, and Interactive Systems, November 1991.

[14] R. Stiefelhagen and J. Yang. Gaze tracking for multimodal human-computer interaction. University of Karlsruhe - Germany, Carnegie Mellon University - USA.

[15] F. Z. T. Morris, P.Blenkhorn. Blink detection for real-time eye tracking. 2002.

[16] A. B. Violeta Ivanova Uzunova. An eyelids and eye corners detection and tracking method for rapid iris tracking. University of Magdeburg, ISG, August 2005.

[17] Z. Xu and P. Shi. A robust and accurate method for pupil features extraction. Institute of image processing and pattern recognition, Shanghai Jiao Tong University, Shanghai 200030, P.R.China.

[18] T. K. Ying-li Tian and J. F. Cohn. Eye-state action unit detection by gabor wavelets. Carnegie Mellon University, Pittsburgh, PA 15213,, 2000.