Abstract— Pupil movement tracking enabled computer interface are highly advanced in processing immobilized human inputs. It reduces hardware cost of external input devices such as keyboards, mouse and touch. Our study uses low resolution images from a built-in or external webcam to capture the positioning of the eye while looking at computer screen. This experiment focusses on the working of a generalized system that can be modified based on user requirements to deliver different purposes. The algorithm used gives faster results in tracking the pupil than other algorithms and techniques. The challenge is to successfully detect the pupil centres and get offsets during pupil tracking. Offsets, further can be helpful in many applications like mouse pointer calibration. We are using information of eye and head positioning that produces the probability estimates of eye positions.

Keywords: Pupil detection, Pupil tracking, Monitoring, Face detection, Low resolution, Webcam

I. INTRODUCTION

The technologies created for eye gaze systems previously used infrared scanner, light emitting diode and head-mounted camera. The generalized system created does not use any electronic device except a webcam and a monitor screen for the user to interact. The webcam records the information about head and eye positioning to estimate the pupil position. Each image frame is processed by the algorithm to give a continuous result of eye gaze. The algorithm is divided into five stages. In the first stage, the input frame from webcam is processed by face detecting algorithm to retrieve the positive part i.e. the face of the user. This frame is divided into multiple regions depending on frame size. In second stage the segregated regions are processed by image classifiers to detect the eyeball i.e. pupil and iris together. The Haar Cascade classifiers uses circular vector positioning to find the centre and boundaries of the eyeball. The third stage uses the centre points for mapping of pupil location and creating mirror image for second eye input. The fourth stage involves monitoring of pupils detected along with gaze track. The final stage involves generation of offsets which can be processed further according to the need of the user, for example – mouse pointer.

II. METHODOLOGY

Determination of special object feature in an image through RGB pixel calibration proved expensive computationally, even slight variation of light could alter the performance and the system needed to be calibrated accordingly. Due, to these reasons it was highly incompetent to use such techniques for recognition. To remove uncertainty and for faster computation Haar-Cascades were used.

A) Face and Eye Detection

Haar-cascades was developed by Voila and Jones and their implementation can be done by using OpenCv, which is an open-source library which deals with real-time object detection and processing. Haar-Cascades mainly concentrates on frontal face detection and feature detection with processing speed of 15 frames per second. This method gives us comparable results even on low-resolution devices such as webcams and other non-infrared video cameras. The proposed algorithm consists of three components based on which the working is explained.
i) **Integral Image** - It is also known as Summed area Table. This is a representation of an image which helps in fast detection of features. Computation of rectangular images on the basis of Haar-like features proved faster. The integral image at a point \((x, y)\) will constitute the sum of pixels above and left of it.

\[
S(x, y) = \sum_{(i,j) \leq (x,y)} I(i,j)
\]

**Figure 1- Graphical representation of integral Image Calculation**

Integral image can calculate rectangular sum in four array references. Therefore, 2 rectangle sum would require 8 array reference, but since the features defined have adjacent rectangular sums, hence 6 array reference would be sufficient.

For example, consider an image *(Figure 2)* divided into four blocks

\[
\begin{align*}
X1 &= P; \\
X2 &= P + R; \\
X3 &= P + Q; \\
X4 &= P + Q + R + S; \\
\text{Therefore } S &= X4 + X1 - (X2 + X3)
\end{align*}
\]

**Figure 2- Integral Image Calculation**

Consider *figure 3* which has image pixels as displayed

\[
\begin{array}{c|c|}
5 & 2 \\
\hline
3 & 6 \\
\end{array}
\]

**Figure 3- Integral Image Calculation**
After summing up the table looks like figure 4-

![Figure 4- Integral Image Calculation](image)

\[ I(x, y) = \sum_{x' \leq x, \ y' \leq y} O(x', y') \]

**I** - Integral Image  
**O** - Original Image  
\( S(x, y) \) – Cumulative row sum  
where \( S(x, -1) = 0; S(-1, y) = 0 \) (out of bounds)  
\( S(x, y) = I(x, y) + S(x-1, y) + S(x, y-1) + S(x-1, y-1) \)

Since, adjacent rectangular sums gets involved in 2 or 3 rectangular sums, therefore, it will require 6, 8 and 9 for 2, 3 and 4 rectangular feature respectively. The task of calculating the sum of pixels of a specific rectangle in a comparative larger image can be done in constant time – \( O(1) \).

ii) **Adaboost**- Construction of classifiers are done by using Adaboost. The Adaboost algorithm selects specific important features from a larger set efficiently. There are many weak classifiers which are only constrained to a single feature. Combination of these weak classifiers coupled with boosting up at each stage helps determining the required feature, for scanning a 384 x 288 pixel image it takes 0.7 seconds.

In a human face, the eye region is comparatively darker than the cheek region. Hence, the first classifier selected by Adaboost uses this information to separate out the eyes from the rest of the face. Also, the nose acts like bridge between the two eyes. Therefore, the second classifier by Adaboost separates out the two eyes. The higher stage classifiers have more complex segregation tasks and it goes on till the desired results are achieved.

iii) **Cascading**- Cascading the classifiers will help us detect potential features at a lower computation cost with low False Positive Rate. Many small boosted classifiers are constructed for rapid detection of all positive frames and rejection of negative sub-windows. Simple classifiers help in rejection of majority of sub-windows while complex classifiers are deployed to achieve low False Positive Rates. The detection process is in the form of degenerate tree. Positive results from first classifier is evaluated by second and then third and so on. Only the positive results are evaluated further and negative sub-windows have no part in further processing.
The cascade stages are constructed by adjusting the threshold in Adaboost. Threshold setting should be done in such a way that it generates high detection rate and low False Positive Rates.

For Example, first stage classifier is constructed on two-feature classifier whose threshold can be adjusted to detect 100% of faces with 40% False Positive Rate. The first-cascade tries to reject as many negative sub-windows as possible. At later stages, the focus comes to less of detection and lower False Positive Rate.

The Training of classifiers is also important. Classifiers with more features yields higher detection rates and low False Positive Rates and classifiers with more features require more computation time. Optimization framework could include the determination of-

a) Number of classifier stages
b) Number of features in each stage

c) The threshold of each stage
A target is set for low False Positive Rate and maximum decrease in detection. Until these requirements are met, stages are added. The face detection cascade has 38 stages with over 6000 features. The face detector can process a 384 x 288 pixel image in 0.067 seconds.

B) Determination of Pupil Centres
The pupil is darker than the Sclera and to filter out wrong estimates we use weights, $W_C$. This helps us to figure out the potential centre as dark pixels are more likely than light ones. Analysis of vector field to derive image gradients is required.

Fabian Timm and Erchardt Barth developed a mathematical formula which precisely locates the eye centres even on low resolution images. The dot product of normalized displacement vector from potential centre position and gradient vector will have the same orientation. Let $c$ be potential centre position.
candidate and $g_i$ be gradient vector at $x_i$ pixel and $d_i$ is the normalized displacement vector, then dot product of $d_i$ and and $g_i$ will have same orientation of that of $g_i$.

Evaluating candidates for pupil centre

The most favourable centre point $c'$ of eyes with a particular pixel position $x_i$, $i \in \{1,..N\}$ is given by

$$c' = \arg \max_c \{ \frac{1}{N} \sum_{i=0}^{N} (d^i, g^i)^2 \}; \quad \text{---(i)}$$

$$d_i = \frac{x_i-c}{|x_i-c|_2}, \forall \ i : ||g_i||_2 = 1; \quad \text{---(ii)}$$

The displacement vector and the gradient vector both are scaled up to unit length in order to avoid noise and variations due to lighting and to have an equal weight for all pixels positions. Gradients outside the iris boundary are eliminated. Introduction of weight $w_c$, in the formulae will filter out non potential centre candidates more efficiently.

$$\arg \max_c \{ \frac{1}{N} \sum_{i=0}^{N} w_c(d^i, g^i)^2 \}$$

**Figure 7- Pupil centre detected with keypoints displayed.**

**Figure 8- Pupil centre detected with glasses on.**

Where $w_c = I^* (c_x, c_y)$ is the grey value at $(c_x, c_y)$ of smoothed and inverted input image $I^*$. Smoothing is necessary to avoid problems that occur due to bright lighting and reflections. The
proposed method yields accurate pupil estimation even with the presence of glasses or in low contrast environment. Incorrect estimation are given where eyes are closed or very strong reflections on the glasses occur. The gradient orientation is disturbed by such noise and hence incorrect results are observed.

C) Combined probability of two eyes
The pupils of the eye move together while gazing, hence, combining probability estimates gave more reliability. Proposed by Luke Allen and Adam Jensen, if the head position is steady then the vector from the left to right eye’s pupil is almost constant. Therefore, if the left eye is shifted by a vector so will the right eye. A slow- moving average filter is used of filter out noise in estimation.

i) Reference Point of Eye Detection
In order to find gaze direction pupils co-ordinates should be defined in terms of an offset from reference point on the face. A reliable reference point should be stable and as proposed by Luke and Adam, the facial reference point does not needs to be based on any specific facial feature. An arbitrary ‘virtual reference point’, preferably the average of the two eye bounding box is taken into consideration. Image key points, are located using SURF algorithm, which are only inside the face and outside the eye window to avoid pupil and other parts of the eyes to be chosen as reference point. For each new keypoint, distance is measured from that to the virtual reference point. In subsequent frames, if the same keypoint appears again, then its ‘vote’ is computed for the position of the virtual reference point by taking the keypoint’s instantaneous position along with its saved immutable vector. The vote for all virtual reference point are combined in weighted average. Keypoints tend to appear and disappear, therefore, consideration of keypoint’s position along the immutable vector is viable.

III. LITERATURE SURVEY
Our project comprises of five parts that are face Detection, eye region Detection, pupil Detection, Pupil Centre Detection and obtaining offsets of pupil while tracking. Haar-Cascades is the prime component in early stages as it segregates face and eyes from the rest of the image. This was proposed by Voila and Jones [1]. Haar- Cascades is a part of a larger framework which includes the usage of Integral Images [5], Adaboost and cascade classifiers. These parts are explained in detail based on the research carried on further. The next part includes detection of eye centre. Eye centre are located through means of gradient which was proposed by Fabian Timm and Erchardt Barth [3]. This method suggests to calculate the dot product of normalized displacement vector from potential centre position and gradient vector. This gives a near accurate estimation of eye centres. The weights are used to filter out the noisy data and non-potential candidates. For pupil tracking, we took joint probability of the two eye centres into consideration and for this reference points on the image were recorded in concordance with virtual reference point. Image key points are located through SURF algorithm [4]. An interesting observation was made by Luke Allen and Adam Jensen, that the facial reference point is not based on any specific facial feature. By setting a virtual point we get the offsets of pupil during tracking. On combining all the above, we can track eyes on low-contrast environment.

IV. CONCLUSION
High end devices that use infra-red radiations have an easy access to the grayscale image of an eye and thus the detection becomes easier and also those devices are costly. On the other hand, low resolution devices such as webcam or video cameras do not have IR sensors thus detection becomes complicated. Thus, we have combined different methods and algorithms to bring the most out of the low contrast images captured by low resolution devices. The method proposed successfully detects and tracks pupils with some restrictions being strong reflection on glasses, very low light environment.
or closed eyes. This method yields results even when the subject is wearing glasses or is present in low contrast environment.

REFERENCES