UACEE International Journal of Computer Science and its Applications - Volume 2: Issue 2 [ISSN 2250 - 3765]

Eye Gazing with low resolution web-cam images using Artificial Neural Network

Ritika Dhyawala[1], Jai Dhariwal[2], Dr. Neeta Nain[3]
Department of Information Technology,
Malaviya National Institute of Technology Jaipur,
Jaipur, India.
[1]ritikadhyawala@gmail.com, [2] jaidhariwal@gmail.com, [3]neetanain@yahoo.com

***Abstract*— In this article, we have proposed a low cost technique for eye gazing using commercially available low-cost device such as web-cam. It can be used for human computer interface and is beneficiary for the handicaps and people who are unable to use mouse in order to interact with computers. We have trained Artificial neural network to estimate the position of eye gaze. The user does not need to wear any head mount device. The binary image of the eyes showing iris, moment of the eye image and the ratio of the nose and eye Euclidean distance serve as the features that are fed into the neural network.**

***Keywords*—eye gazing, artificial neural network, image processing, web-cam.**

I. INTRODUCTION

Eye gazing plays an important role in Human computer Interaction .Eye gaze is more natural way to interact with computers than other currently available input devices. Gaze estimation methods can be classified into two categories i.e. intrusive and non intrusive. Intrusive methods involves direct interaction with the user so the user needed to wear a head mounted device or kind of equipment resulting in discomfort. Non Intrusive methods are more comfortable to the user but faces some problems like head orientation, head rotation and movement in all axes and number of faces in front of image acquisition device.

Wen Zhang proposed Gazing Estimation and Correction from Elliptical Features of One Iris using RANSAC algorithm with a average error of 0.84 cm focus point on screen[5]. There has been work done on eye gazing using IR cameras that detect the pupil. More than one cameras are used to generate the 3D model of the eye in order to estimate the orientation of eye for gaze detection. But this requires predefined set up to be arranged to serve the purpose of gazing. For 3D model based systems, gaze directions are estimated as a vector from eyeball center to the iris centers[2][3].

Camera based non intrusive trackers use images of eye to estimate the gaze by using characteristics of eyes to detect. IR based approaches utilizes illumination from infrared light source to enhance the contrast between pupil and iris[1][4]. After grabbing the features of eye image, pupil and glint (a small bright point on the corneal surface) they estimate the gaze by second degree polynomial mapping function based on precalibration of the system on nine points.

Baluja and Pomerleau proposed a neural network method without explicit features [6]. Where they considered each pixel of the image is considered as an input parameter to the neural network. Once the eye is detected, the image of the eyes is cropped and then used as the input of ANN (Artificial Neural Network). In [7], authors proposed remote eye gaze tracker based on eye feature extraction and tracking by combining neural mapping (GRNN) to improve robustness, accuracy and usability under natural conditions.

We are proposing a eye gazing estimation method by using Artificial Neural network with features including the orientation of face and image moments to provide Neural Network with better features. We are capturing images from webcam hence the user do not have to buy or integrate any other hardware as nowadays most of the laptop comes with integrated webcam.

II. APPROACH

A. *Eye Detection.*

The foremost requirement to detect eye gazing is to firstly detect the eye region in a face. This is done using Haar like object detectors which is formely proposed by Paul Viola and later extended by Rainer Lienhart and Jochen Maydt. We can train the detector to classify any desired object with its different view and pose. It uses the haar-like input features which applies threshold over the sum and differences of the rectangular regions over the image. It is a fast approach since it uses integral image technique that rapidly computes the value of the rectangular region. In order to achieve correct position of the two eyes, we have first recognized the face and then placed the region of interest on the two equal vertical halves of the face. We have used the haar-cascades of face recognition, and the right and left eye from the library provided by OpenCV 2.1.. We have got an error of 6.146% along X coordinate and 8.066% along Y coordinate while gazing.

III. ARTIFICIAL NEURAL NETWORK

We have used the Levenberg-Marquardt backpropagation algorithm in order to train the feedforward neural network. It provides high precision and is very efficient for training networks which have up to a few hundred weights. It incorporates standard nonlinear least square optimization algorithm into the backpropagation algorithm.

Identifying correct features for training the artificial neural network plays an essential role for fast and efficient results. The features should be non-redundant, and contented to provide sufficient information to analyze the output. For this we have extracted various features from the image which were fed to the ANN for training and testing.

B. Image Processing for feature

- After detecting the two eyes from the face, we place region of interest on both eyes one by one.
- The image obtained contains some part of eyebrows in it. Since it has dark shade, so it might act as a hindrance in detecting the iris. So, we had to crop the image from vertical direction. Normally, cropping it 15% from both ends vertically lead us good results. [8]
- The segment obtained after applying haar-feature classifier over face to detect the eyes do not provide uniform size of the rectangle. So, after cropping the image vertically, we resize the image into a constant size for further processing. Here have used a fixed size of 50x50 pixel.
- Each frame of eye consists of three channels i.e. RGB components. We will split the image into the respective channels.
- Each channel passes through a process of histogram equalization that yields equalized intensity distribution histogram. The resultant image has a better contrast.
- When the resultant image of the three channels are merged again, it becomes easier for us to implement thresholding over its grayscale value.
- The image is thresholded with minimum value of 60, since the skin color and sclera would come under this threshold level, leaving behind the dark iris and the eye lashes.
- Image moments are calculated of the resultant image.

C. Image Moments

The n[th] order moment about the point c is defined as:

$$\mu_n = \int_{-\infty}^{+\infty} (x-c)^n f(x)\, dx \tag{1}$$

The moment for two dimensional image can be determined by:

$$\mu_{m,n} = \int \int_{-\infty}^{+\infty} (x-c_x)^m (y-c_y)^n f(x,y)\, dy\, dx \tag{2}$$

Where, f(x,y) is the image. Since image pixel have discrete values, so the integral becomes summation.

$$\mu_{m,n} = \sum_{x=0}^{\infty} \sum_{y=0}^{\infty} (x-c_x)^m (y-c_y)^n f(x,y) \tag{3}$$

The order of the moment is m+n. We have calculated the moment about (0, 0).

The area of binary image can be calculated by:

$$\mu_{0,0} = \sum_{x=0}^{w} \sum_{y=0}^{h} x^0 y^0 f(x,y) \tag{4}$$

The centroid of binary image can be calculated by:

$$centroid = \left(\frac{\mu_{1,0}}{\mu_{0,0}}, \frac{\mu_{0,1}}{\mu_{0,0}}\right) \tag{5}$$

The moment along x coordinate:

$$sum_x = \sum \sum x f(x,y) \tag{6}$$

The moment along y coordinate:

$$sum_y = \sum \sum y f(x,y) \tag{7}$$

In order to get the average, we compute the centroid as:

$$\mu_{1,0} = \frac{sum_x}{\mu_{0,0}}, \quad \mu_{0,1} = \frac{sum_y}{\mu_{0,0}} \tag{8}$$

D. Face alignment feature

With respect to the change in Iris position in eyes, the change in orientation of face plays a major role in detecting the gaze. The feature included in it are:

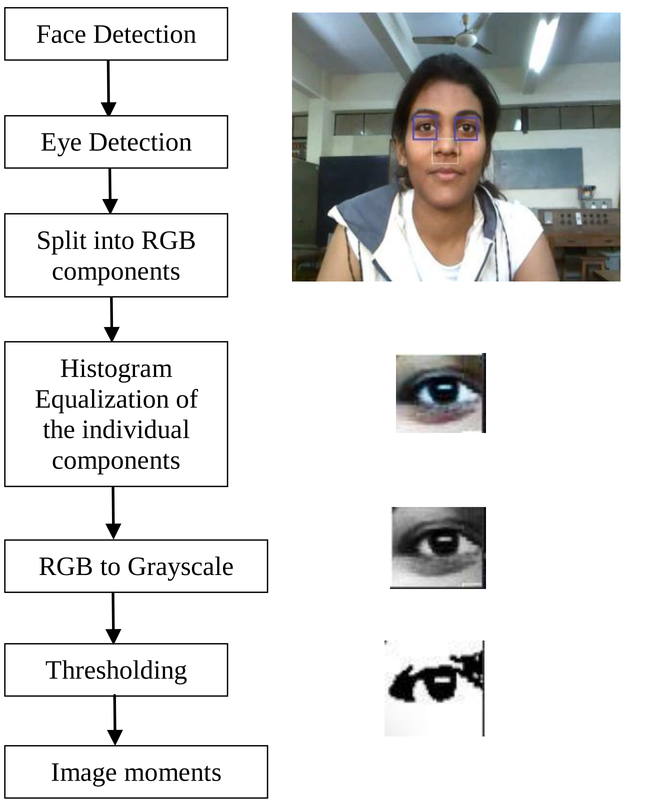

Figure 1. Processing on the image of Eye

- The Euclidean distance between the center of the region of the two eyes detected.
- The Euclidean distance between the center of two eyes and the center of the nose. The nose can be detected by using the haar based classifier over the face as region of interest. The haar-cascade of nose is availabl e in the library of OpenCV.
- The ratio of the above two calculated distances.

E. Features used in training ANN

There were a total of 5013 features which were used to train the ANN. Those were:

- The pixel values of both the eye image after thresholding is done. This is a binary image where blob of Iris is present along with eye lashes. There are 50x50 pixel values for one eyes and a total of 5000 pixel value feature.
- The centroid of the thresholded image.
- The area of the thresholded image.
- The three spatial moments calculated i.e. $\mu_{0,1}$, $\mu_{1,0}$ and $\mu_{0,0}$.
- The distance between the centers of two eyes, say d1.
- The distance between the center of eye and the nose, say d2.
- The ratio of d1 and d2.

F. X11 library for mouse pointer

The X11 library provides a low-level C library for writing X-clients. The communication between x display and the client uses a serial communication channel. Using this library, we can determine the position of the mouse pointer and can control the action of the mouse through eye gazing.

G. Data Set for training ANN

The data set which is used to train the Artificial Neural Network is made using the webcam available on the laptop. The user sits in front of the laptop and gaze the movement of the mouse pointer. The position of the mouse pointer at that instant is moved to the file along with the features of the eye image of that particular frame. The data set can be assumed as reliable since the user continuously gazes at the screen to the position where he moves the mouse pointer. We have used 15% of the data set as validation set and 15% of the data set as testing set.

IV. RESULT

We have used Intel® Core™ i3 Processor to train and test our ANN. The image processing is done using open source C library OpenCV using linux platform. Our project is tested under all lightening conditions. It is user friendly since it does not require user to wear anything on his head nor he has to fix his head stable. We have tested the system and we found out that this system is slow in respect to the currently available control technology such as mouse. By this system we can randomly access any position on the screen without continuously moving the mouse pointer through a specific path. We have got a very reliable data set which can also be extended or changed. We used the features which are not specific to the user hence the trained network can be used for any person. Problems like head rotation do affect the performance of the system and besides that, the factors causing low precision are that the inadequate veracity of algorithm and the low resolving power of camera used in the system.

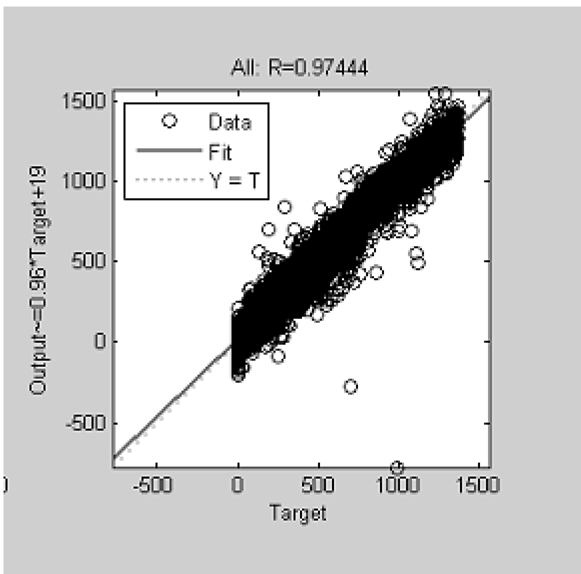

Figure 2. Regression Plot obtained between output and target of ANN

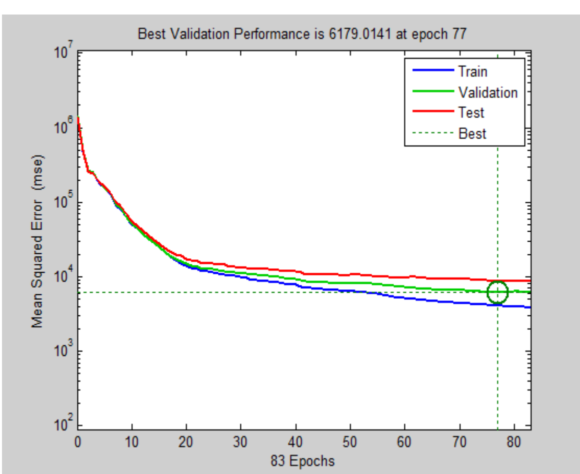

Figure 3. Performance Plot after training, validation and test.

TABLE I.

S.No.	Mouse pointer coordinates				Root mean square error
	Real X	Real Y	Predicted X	Predicted Y	
1.	1307	50	1297.10	48.74	9.97
2.	1345	159	1351.99	162.76	7.94
3.	1270	347	1269.99	341.92	5.07
4.	418	623	424.59	615.85	9.72
5.	233	608	239.66	605.90	6.98
6.	410	380	419.60	384.60	10.64
7.	775	158	783.20	152.12	10.09
8.	537	227	540.71	227.33	3.72
9.	1160	421	1159.29	429.89	8.92
10.	1012	574	1013.09	583.47	9.54
11.	181	162	178.21	166.39	5.19
12.	734	415	728.83	416.54	5.39
13.	623	351	616.54	356.98	8.80
14.	473	297	469.02	295.80	4.14
15.	317	593	319.43	589.64	4.14
16.	1028	206	1035.23	203.88	7.54
17	374	314	372.43	319.85	6.05
18.	762	733	763.58	742.79	9.91
19.	742	462	742.55	465.47	3.52
20.	609	503	612.72	494.67	9.11

A. TABLE SHOWING ROOT MEAN SQUARE ERROR OF X AND Y COORDINATE OF SOME DATA.

Size of data set : 5478

Total root mean square error in X coordinate : 83.9638 pixel

Total root mean square error in Y coordinate : 61.9470pixel

Resolution of the screen tested : 1366X768 pixels

Percent error in X coordinate: 6.146 %

Percent error in Y coordinate: 8.066 %

Figure 4. Image of eye which is fed into ANN indicating the position of iris in the eye.

V. CONCLUSION

In this article, we have proposed a algorithm by which we can feed the movement of iris in the eye image as a good feature into the artificial neural network, increasing the efficiency of training. We have got an error of 6.146% along X coordinate and 8.066% along Y coordinate. The usage of image moments and the Euclidean distance between the eyes and nose serve the purpose of change in face orientation while gazing. The neural network has to be trained once and it provides good results in all lightening conditions since we have increased the contrast using Histogram Equalization.

VI. REFERENCES

[1] Liu, T. & Pang, C. *Eye-Gaze Tracking Research Based on Image Processing* Image and Signal Processing, 2008. CISP '08. Congress on **2008**, Vol. 4, pp. 176 -180

[2] Wang, J.-G., Sung, E. & Venkateswarlu, R**.** *Estimating the eye gaze from one eye* Comput. Vis. Image Underst., Elsevier Science Inc.,**2005**, Vol. 98, pp. 83-103

[3] Yamazoe, H., Utsumi, A., Yonezawa, T. & Abe, S. *Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions* Proceedings of the 2008 symposium on Eye tracking research &38; applications ACM,**2008**, pp. 245-250

[4] Yoo, D.H. & Chung, M.J. *A novel non-intrusive eye gaze estimation using cross-ratio under large head motion* Comput. Vis. Image Underst., Elsevier Science Inc.,**2005**, Vol. 98, pp. 25-51

[5] Zhang, W., Zhang, T.-N. & Chang, S.-J. *Gazing estimation and correction from elliptical features of one iris* Image and Signal Processing (CISP), 2010 3rd International Congress on **2010**, Vol. 4, pp. 1647 -1652

[6] S. Baluja and D. Pomerleau, "*Non-intrusive gaze tracking using artificial neural networks*", Tech. Rep. CMU-CS-94-102, School of Computer Science, CMU, CMU Pittsburgh, Pennsylvania 15213, Jan. 1994.

[7] D. Torricelli, S. Conforto, M. Schmid and T. D'Alessio, "*A neural-based remote eye gaze tracker under natural head motion*", Comput. Meth. Prog. Bio, pp. 66–78, 2008.

[8] Ibrahim Furkan Ince and Jin Woo Kim2, *"A 2D eye gaze estimation system with low-resolution webcam images*", Ince and Kim EURASIP Journal on Advances in Signal Processing 2011.