ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Non Contact Human Computer Interaction Using Eye Blinks

Ms. S. Vijiyalakshmi*
Department of Information Technology
Christ College of Engineering and Technology
Puducherry-605 005, India
sviji7195@yahoo.com

Dr. S. Sivasathya
Department of Computer Science
Pondicherry University
Puducherry-605 014, India
ssivasathya@gmail.com

R.K.Jananee N.Karthiga
Department of Information Technology
Christ College of Engineering and Technology
Puducherry-605 005, India
jananee.bonbon@gmail.com

Abstract: In this paper, a non-contact HCI system for the physically impaired persons is presented. The persons who were paralyzed with some diseases are unable to move their body parts except their eyes. So for those disabled persons this non-contact system is designed with the facility to interact with their eyes. Blink patterns are used to interact with the system to open an application as specified. Eye blinks are the normal action that a person does without any strain; thereby this interaction is applicable for elderly persons also. Simple USB cameras or the web cameras are used to detect the face, to locate the eyes and finally to detect the eye blinks in real time environment. Our experiment says that this system works for static background; the background should not have any moving objects, since the movement of the eye has to be tracked. The eye blink is taken as an action to open an application with using any input device like mouse or keyboard.

Keywords: Blink patterns; eye blink; face detection; HCI (Human Computer Interaction); locating the eyes; cameras; non-contact interaction.

I. INTRODUCTION

To make the computer systems more usable even with the disabled and impaired people we make use of the Artificial Intelligence, where the human computer interaction is applied. The usages of computers among all sorts of people are made possible using new technologies. We here describe about the interaction with the system in contactless manner. It is possible to use the computer without the physical touch of the devices. With the web camera and the computer one can work easily.

The main stage of this system is the 'blink detection'; it is the simplest way to interact with the system as an input action. Blink detection involves the following stages:

a. ***Face detection***-where the skinned region along the features (eyes, eye brows, nose and mouth) are detected and displayed as grey image.

b. ***Locating the eyes***-here the eye pair is tracked and a template of the eye pair is created, eyes are located with the movement of the eyeballs.

c. ***Blink detection***-voluntary blinks are made to detect the eyes, two rectangle boxes indicates that the blink is made.

Usually face detection involves the open CV's method; the Principle Component Analysis is used to detect the face. Blink detection is widely used for various purposes, in various fields. It is used in editing the home video and summarizing it [1] where the eye blinks are taken as a part of human behavior in watching a video, in other ways the blinks are important to find out one's fatigue while working with the system [2]. Also the blinks are counted to predict the average view of a person driving during the night time [3] and in image processing for eye detection [4]. Before implementing the blink detection we must detect the face to identify the eyes and to track the head movement. Then the eye must be tracked in pair to detect the eye blinks.

In an innovative way, we proposed a new usage of the eye blinks for disabled persons. The blinks that is interacted to open an application from the system. It is a low cost method of applying non-contact interaction with the commonly available hardware like a computer with a web camera or a laptop with a web camera.

II. RELATED WORK

Blink detection previously was implied as a part of multimedia and other computer applications [1], [2], [4], [5]. A research on face detection uses the blink detection concept, where the eyes are the continuously moving part in face. Blinks are of two types voluntary and involuntary; voluntary blinks are applicable for interaction and the involuntary blinks for face detection and/or for eye detection [6].

Face detection (recognition) methods have existed for decades, and is continually improving. One of the oldest methods that has stood the test of time is Eigen faces[7], developed by Matthew Turk and Alex Pentland, which utilizes Principal Component Analysis (PCA) for recognition. Newer facial recognition systems use 3D models [8], which were considered for this project along with the grey prediction model [9].

Blink detection was just a part of the related work that signifies the interest of the person in a video with other features like head movement, saccade and facial expressions [10].

III. APPROACH

In our proposed system, face is detected in order to locate and detect the eyes for tracking the eye blinks. A step by step approach is explained from the face detection and the methods followed for each module in our project. Figure 1 shows the overall system architecture of our project.

A. Face Detection/Image Differencing:

The first objective of this project is to identify human faces in an image and video. As mentioned above, the first stage in Face Recognition is Face Detection. The OpenCV library makes it fairly easy to detect a frontal face in an image using its Haar Cascade Face Detector (also known as the Viola-Jones method). We utilized a pre trained cascade in the OpenCV library, which obtained acceptable accuracy and used in this project. Grey prediction modal is used to produce the grey image of the input image so as to obtain the output. In this face detection itself the threshold is maintained to see the output grey image where the face with its features is obtained. Two frame windows are used; one to show the input image and the other to show the grey image i.e. the output image.

Grey image shows the pixel ratios of the black and white colors where the white/grey color indicates the features like close points together. This approach gives 80% accuracy where the background environment is static that is without any moving objects. We proceed to the next stage only after getting the grey image of the real time input image.

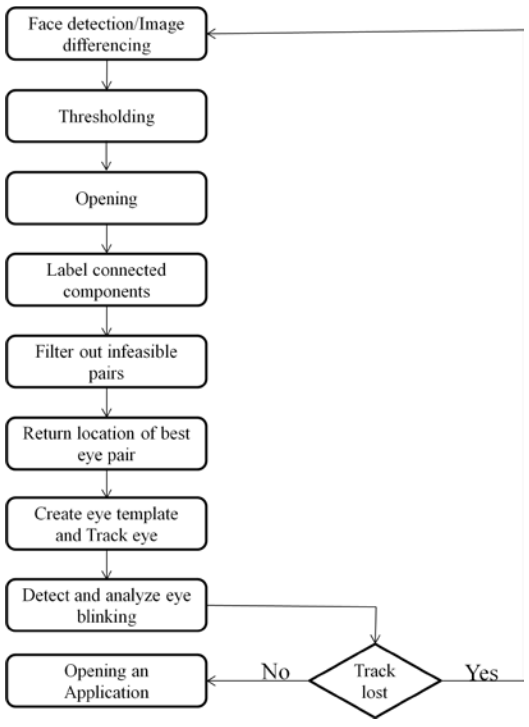

Face detection/Image differencing
Thresholding
Opening
Label connected components
Filter out infeasible pairs
Return location of best eye pair
Create eye template and Track eye
Detect and analyze eye blinking
Opening an Application
No
Track lost
Yes

Figure 1. Overall system architecture

B. Eye Detection and Locating the Eye Pair:

In this stage the eyes are put into movement since we use the motion technique in identifying the eyes in the face. Usually people blink up to 10 times per second, which is considered as involuntary blinks, even this involuntary blinks may identify the eyes. But as far as the blink detection the OpenCV's library works accurately with certain compilers and operating systems. Linux (fedora, ubuntu, etc) has the best computing environment with accuracy. Here the motion analysis is made for the eyes, eyes are subjected to move and the eyes are detected. Eyes move in pair so the eye has to be located in pair.

And once the center is found the location of the eye is returned to the system to proceed further. Returning the eye location follows the following code and eye pair is returned along with the minimum location size and the maximum location size and their height and width of the eye. Two rectangle boxes are used to indicate the location of the eye (here it simply refers to the tracker).

C. Blink Detection:

Blink detection is followed after a template creation of the eye pair. Those templates last for few seconds indicating that the eyes were detected and located as pair. In blink detection stage voluntary eye blinks are made until they are detected. The eye template servers as a base for blink detection, as the user opens and closes his eyes the correlation score varies depending on the eye movement.

As the user's eye is in the normal open state, very high correlation scores of about 0.85 to 1.0 are reported. As the user blinks, the scores fall to values of about 0.5 to 0.55. Finally, a very important range to note is the one containing scores below about 0.45. Scores in this range normally indicate that the tracker has lost the location of the eye. In such cases, the system must be reinitialized to relocate and track the new position of the eye. For indicating the blink detection, as soon as the blink is detected a message is displayed as "blink detected".

D. Opening Application:

This blink action is taken into consideration for interacting with the system to open any user specified application. Only after the message is displayed the interaction is applicable, we check if the blink is detected and if it is true then the tracker gets lost and it opens the specified application(Notepad for example) and the frame windows are exited.

E. Computational Consideration:

These platforms are limited in both computational power and memory space. Throughout this project, there was a consideration in what algorithms to use, and how our code is implementing. To keep our application on the system as responsive as possible, we broke our application into three parts:

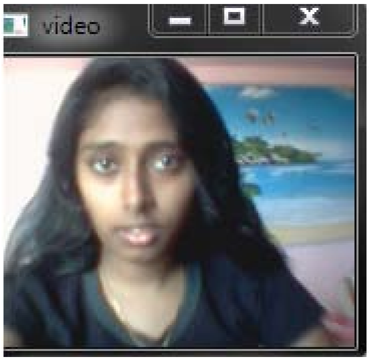
a. Preprocessing: We do as much computation as we could off the system, such as checking for the availability of the web camera, opening the camera to capture the image, creating the difference windows for grey image and input image, and calculating the time taken to detect the blinks. These results were analyzed and were loaded at run-time.

b. ***Run-time:*** When the application is run it loads the camera and produces the window frames. The application then uses this to calculate the Principle Component Analysis features and grey prediction modal. This causes our application to take about 0.5-1 seconds to load, which is a reasonable amount of time.

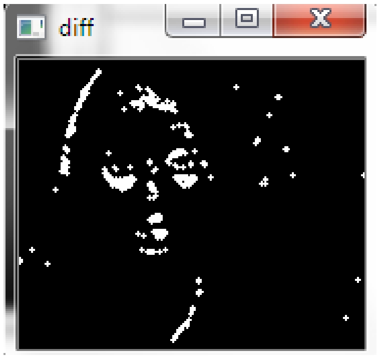
c. ***Interaction with the system:*** Finally, the calculations are made once the application is opened. First, it applies the motion analysis and PCA & grey prediction model to find the faces, and then it applies the preprocessing as described before. Finally, it paints rectangle boxes to indicate the location of the eyes, displays it on the system along with the message "blink detected", and saves it to the system. All this takes about 3-5 seconds, which is completely reasonable for system-based on real time computing.

IV. RESULT

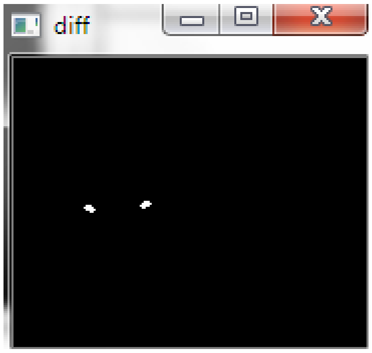
The results of our project are shown as snap shots below: it starts with the input image as real time color image and after a threshold of 5 seconds a grey image is produced in another frame which exactly concentrates on the eye pair so that it locates them and identifies them.



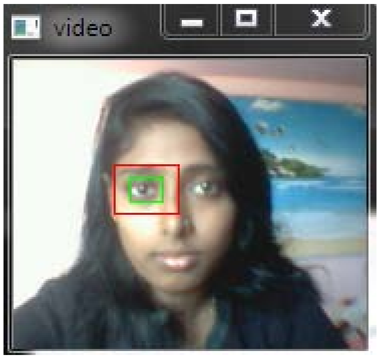
A. Figure: 2 Input Image



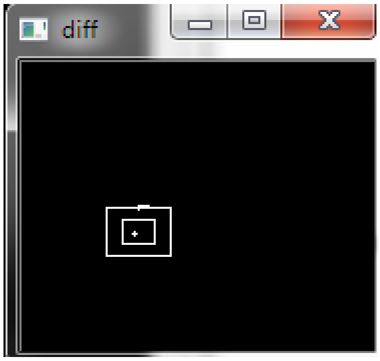
B. Figure: 3 Grey Image



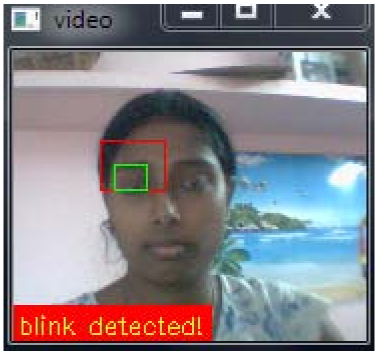
C. Figure: 4 Eye Detection



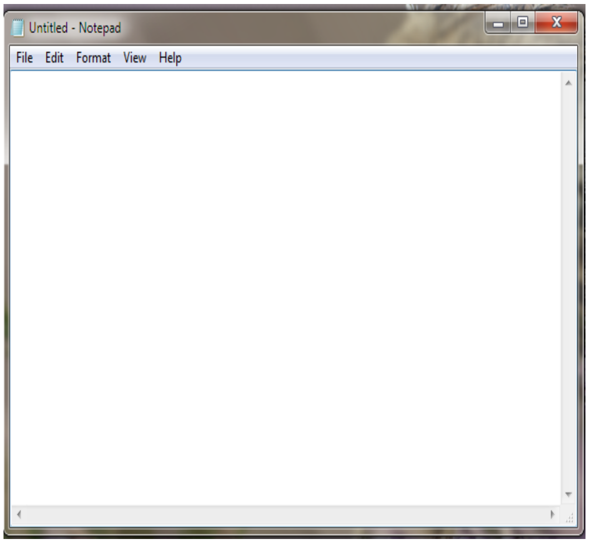
D. Figure: 5 Locating the eye



E. Figure: 6 Locating eye in gray image



F. Figure: 7 Blink Detected



G. Figure: 8 Applications Opened

V. CONCLUSION

Thus we successfully created a system for disabled persons who cannot move their hands and cannot speak. The system worked effectively by opening an application of user's wish. Here the time complexity is much less where there is no dynamic environment, we opened the application with the blink action in just 2-3 seconds which is much faster than searching that specific application in our system through mouse or keyboard devices. The code using OpenCV is a way of shortcut for opening an application through eye blinks.

VI. FUTURE ENHANCEMENT

We give an enhancement as counting the number of blinks for opening different applications on those counts and a specified time delay to close the current application. Since this is the first step to interact with an application using eye blinks we just interacted with only one application. But this can be used to open as many as required by the user depending the counts of blinks. Once the application is opened it must be closed, to do so the eyes can be closed for a certain amount of time (which indirectly indicates the closing of the application).

VII. REFERENCES

[1] Wei-Ting Peng and Wei-Ta Chu," Editing by Viewing: Automatic Home Video Summarization by Viewing Behavior Analysis", ieee transaction on multimedia, Vol.13, No. 3, June 2011.

[2] Eng-Jon Ong, and Richard Bowden," MIMiC: Multimodal Interactive Motion Controller", ieee transactions on multimedia,Vol. 13, No. 2, April 2011

[3] S.Saravanakumar and N.Selvaraju," Eye Tracking and Blink Detection for Human Computer Interface", International Journal of Computer Applications (0975 – 8887) Volume 2 – No.2, May 2010.

[4] Horng W. B., Chen C. Y., Chang Y., Fan C. H.(2004), "Driver fatigue detection based on eye tracking and dynamic template matching", Proc. of ieee Int. Conf. on Networking, Sensing and Control, pp. 7-12

[5] Aleksandra Krolak, Pawel Strumillo, "Eye-Blink Detection System for Human-Computer Interaction", the International Conference on Universal Technologies Oslo University College, Oslo, Norway, May 19-20, 2010.

[6] C. Blehm, et al.: "Computer Vision Syndrome: A Review", Survey of Ophthalmology, Elsevier Inc., vol. 50, no. 3, 2005.

[7] 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.

[8] P. Viola and M. J. Jones, "Robust real-time face detection," Int. J.Comput. Vis., vol. 57, no. 2, pp. 137–154, 2004.

[9] R. Valenti and T. Gevers, "Accurate eye center location and tracking using isophote curvature," Proc. IEEE Comput. Vis. Pattern Recognit., pp. 1–8, 2008.

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