Hands Free System Control using Face Tracking and Speech Recognition

Shruti Tandur[1] Dilip Kale[2] Abhiruchi Tayade[3]
[1, 2, 3]Computer Engineering Department
[1, 2, 3]Rajiv Gandhi Institute of Technology, Mumbai University

Abstract--- This paper presents a simple prototype system for real time tracking of human face and speech. It uses a simple and effective algorithm. This algorithm is efficient in terms of computation and has the ability to work in different environment. The objective of this paper is to demonstrate a system which uniquely utilizes real time video of the users' face captured using the in-built web-camera and allowing speech inputs from the user to perform operations on the system. The position of the face is tracked and converted into two-dimensional coordinates on a computer screen. Additionally, speech recognition is used to operate the machine and give more functionality to it.

I. INTRODUCTION

Using mouse and keyboard has been most followed trend in the human race till date to interface with a computer. But, the same thing may not apply to the people with certain disabilities.

In recent years, there have been many efforts taken to develop new system to enhance human and computer interaction. It has be interesting and challenging to build system which can be operated without using one's hand and these systems are major advancement towards transparent computer assistant. These systems are capable of perceiving the environment and human behaviour instead of asking the user to learning the skills required to operate the system and even investing a lot to buy the product.

People who are physically disabled can voluntarily make limited movements. Many systems are developed for those who have limited movements such as, devices to detect small muscle movements or eye blinks, infrared or near infrared camera based systems to detect eye movements, electrode based systems to measure the angle of the eye, even systems to detect features in EEG. And these have played a vital role in many people lives and in the advancement of the technology. But the major problem with all such type of system is that they required hardware which is costly and hence it is not affordable to all. Additionally, to operate these systems the person need some basic knowledge of the hardware and the system and required basic skill to use them.

Therefore, the proposed system describes the functionality which is similar to the mouse and keyboard, the basic and important parts of the machine, by using human face tracking and speech recognition. Where face tracking is used for the mouse pointer movement and speech recognition for controlling system by performing actions given by the user.

II. SYSTEM DESIGN

The overall system design is as shown in the following block diagram (Fig. 1), which comprises of two processing units and interpreter. First is video processing, which will trace the movement of the face and accordingly control the mouse pointer. Second is speech processing, which will recognize the commands given by the user as voice input. Interpreter will interpret these commands and perform the actions.

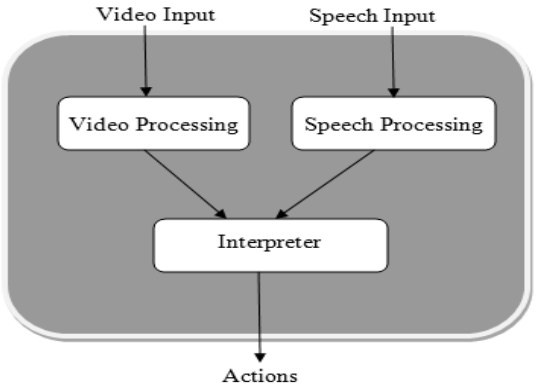

Fig. 1: Proposed System Block Diagram

III. VIDEO PROCESSING

In video processing, it basically tracks face and eye blinking as shown in the Fig. 2.

It initially performs initialization of camera for video streaming. After initialization, each frame is grabbed and face and features are identified. The difference between the current and the initial location of these features is calculated and average of this is used for calculating the translation. According to this translation the mouse pointer is moved.

For performing clicks it is most important that the eye pair must be detected in feature identification phase. If user blinks only left eye, left click is performed. Similarly, if user blinks only right eye, right click is performed

A. Face And Feature Identification

The face and its feature detection is implemented by Haar Feature based Cascade classifier [6]. The object detector described below has been initially proposed by Paul Viola [8][9] and improved by Rainer Lienhart [2].

First, a classifier (namely a cascade of boosted classifiers working with haar-like features) is trained with a few hundred sample views of a particular object (i.e., face, eye pair, nose, mouth), called positive examples, that are scaled to the same size (say, 20x20), and negative examples - arbitrary images of the same size.

After a classifier is trained, it can be applied to a region in an input image. The classifier output is "1" if the region is likely to show the object else "0". To search for the object in the whole image one can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily "resized" in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the

image itself. So, to find an object of an unknown size in the image the scan procedure should be done several times at different scales.

The word "cascade" in the classifier name means that the resultant classifier consists of several simpler classifiers (stages) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. The word "boosted" means that the classifiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques [3].

Haar-like features are the input to the basic classifiers, and are calculated as described below. The current algorithm uses the following Haar-like features:

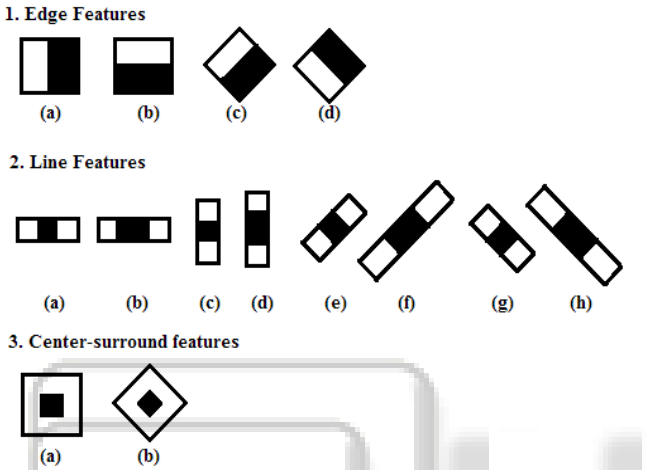

Fig. 3: Haar-like Features

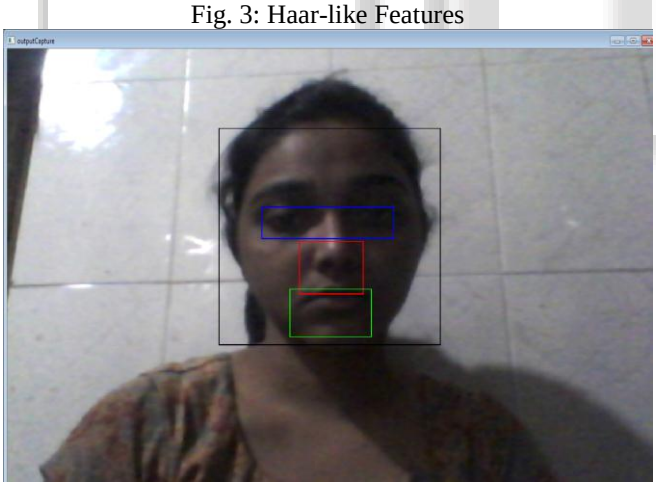

Fig. 4: Output of face & feature identification

Initially face like regions are detected [4][5]. In case, multiple faces are detected, the face which as largest area is selected. Further, on that region eyes, nose and mouth are detected as shown in the Fig. 4. Calculate the average of all the features detected and calculate the difference between average features of current and earlier frame. The mouse pointer is translated by the difference calculated above.

B. Single Eye Blink

The eye pair detected using haar cascade classifier is taken as the region of interest (ROI). The ROI is divided into two parts left eye and right eye. Histogram equalization is performed to remove noise from ROI. Thresholding is performed on each part of the ROI to detect the pupil. Next, contour of the left eye and right eye is carried out to check whether any of the eyes is open or not. Hence, if user's only left eye is closed left click is performed else if user's only right eye is closed right click is performed else no clicks are performed.

IV. TECHNOLOGY USED

A. Opencv With C++

OpenCV (Open Source Computer Vision) is a library of programming functions for real time computer vision. It is developed by Willow Garage, which is also the organization behind the famous Robot Operating System[1].

MATLAB can also be used for Image Processing but OpenCV has following advantages over MATLAB.

1) Speed

Matlab is built on Java, and Java is built upon C. So when you run a Matlab program, your computer is busy trying to interpret all that Matlab code. Then it turns it into Java, and then finally executes the code. OpenCV, on the other hand, is basically a library of functions written in C/C++. You are closer to directly provide machine language code to the computer to get executed. So ultimately you get more image processing done for your computers processing cycles, and not more interpreting. As a result of this, programs written in OpenCV run much faster than similar programs written in Matlab. So, OpenCV is fast when it comes to speed of execution.

2) Resources needed

Due to the high level nature of Matlab, it uses a lot of your systems resources. Matlab code requires lot of RAM to run through video. In comparison, typical OpenCV programs only require ~70mb of RAM to run in real-time. The difference is highly considerable.

3) Cost

List price for the base (no toolboxes) MATLAB (commercial, single user License) is around USD 2150. OpenCV (BSD license) is free!

4) Portability

MATLAB and OpenCV run equally well on Windows, Linux and MacOS. However, when it comes to OpenCV, any device that can run C, can, in all probability, run OpenCV.

V. SPEECH PROCESSING

Presently, there has been great advancement in speech recognition and has been used widely for many application. In this paper, we use speech recognition to add more functionality to the system. One of the most popular and popular and simpler technique for speech recognition is Hidden Markov Model (HMM) [7].

Hidden Markov Model is a form of finite state machine having a set of hidden states, an output, transition probabilities, emission probabilities and initial state probabilities. It is the simplest dynamic Bayesian network. Training this network is also simpler.

The Speech Processing works as shown in the block diagram (Fig. 5). Here, the proposed system uses Microsoft Speech API 5 for performing speech recognition, which takes the speech input and gives the text as the output. Further, this text is matched with the commands and accordingly the actions are performed.

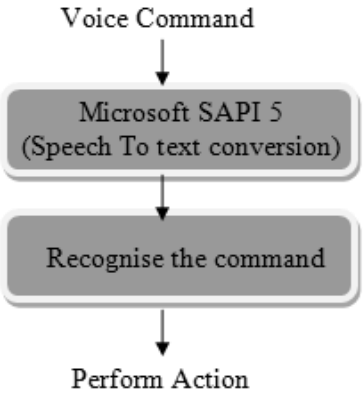

Fig. 5: Working of Speech Processing

The voice commands given by the user can be as follows:
Calculator: To open calculator.
Notepad: To open notepad.
Paint: To open paint editor.
Save: To save the currently open document.
Word: To open word pad.
Write: To perform write into the document that is open.
To perform keyboard operations:
Press followed by the key, e.g. Press a.

For certain keys like Delete, Backspace, Home, etc. the voice command can be simply Delete, Backspace, Home, etc. respectively.

In this way, we can have many additional commands to control the system.

VI. Conclusion

This is an effective technology which ensures maximum usage of current generation computers in an innovative manner. Also it proves to be time efficient and user friendly. Additionally, it is cost effective as it does not require any external hardware as required in most of the existing systems.

VII. Future Scope

This concept mentioned in the paper can be used for many applications such as developing games, virtual reality, security, surveillance, study and protection of any inaccessible spots and places where physical protection to user is essential. One such application can be as follows; in wild life sanctuary for the study of wild life it is not always possible for the human to go closer to the animal. Hence, a robot with similar features can be built with a camera which moves according to the users face movement giving the user view of that place and speech for movement of the robot. Similarly, this concept can be used in many fields giving a virtual view to the user of an inaccessible place.

Acknowledgement

We wish to express our sincere gratitude to Dr. U. V. Bhosle, Principal and Prof. S. B. Wankhade, H.O.D of Computer Department of Rajiv Gandhi Institute of Technology for providing us an opportunity to publish this on “Hands Free System Control". This project bears an imprint of many peoples.

References

[1] Gary Bradski and Adrian Kaehler, “Learning OpenCV: Computer Vision with OpenCV Library”, O’reilly Publication.
[2] Lienhart, R. & Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. Proceedings of the International Conference on Image Processing (ICIP), pp. I-900- I-903, September 2002, IEEE, Rochester, New York, USA.
[3] M.Gopi Krishna, A. Srinivasulu, “Face Detection System On AdaBoost Algorithm Using Haar Classifiers” International Journal of Modern Engineering Research (IJMER) Vol. 2, Issue. 5, Sep.-Oct. 2012 pp-3556-3560 ISSN: 2249-6645
[4] OpenCV dev team (2011-2013), OpenCV documentation on face recognition.
[5] OpenCV dev team (2011-2013), OpenCV documentation of Cascade Classification.
[6] Phillip Ian Wilson And Dr. John Fernandez,” Facial Feature Detection Using Haar Classifiers*”JCSC 21, 4 (April 2006)
[7] Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. ASSP Magazine, IEEE, Vol. 3, No. 1, 4-16, ISSN: 0740-7467
[8] Viola, P., & Jones, J. J. (2001). Robust Real-Time Face Detection. Proceedings of the IEEE International Conference on Computer Vision (CVPR), pp. 122-130, ISBN:0-7803-7965-9, July 2003, IEEE Computer Society, Washington, DC, USA
[9] Viola, P., & Jones, J. J. (2004) Robust Real-Time Face Detection. International Journal of Computer Vision, Vol. 57, No. 2, May 2004, 137-154, ISSN: 0920-5691.

