THE INTERNATIONAL JOURNAL OF
SCIENCE & TECHNOLEDGE

Nouse Based Technology for Physically Challenged Persons

G.Sumathy
Assistant Professor,Information Technology, Jeppiaar Engineering College, Chennai, India
Dr. Arokia Renjith
Professor,M.E ,CSE Department, Jeppiaar Engineering College, Chennai, India

Abstract:
This paper is mainly for physically challenged persons(especially without hands) ,where they have to use their nose tip as cursor and left and right eyes as left and right button of the mouse ,to handle mouse clicks of a computer instead of mouse. This is done by tracking the areas of nose, eyes based on dark areas between the eyes. In Tongue Drive System, the challenged persons have to use their tongue to move the cursor. Detection of cursor becomes difficult in that case.

Keywords: *Face detection, Tracking system, SSR Filter, Transformation, Feature extraction*

1. Introduction

In our work we were trying to compensate people who have hands disabilities that prevent them from using the mouse by designing an application that uses facial features (nose tip and eyes) to interact with the computer. In existing system different devices are used in HCL (Human Computer Interface) such as infrared cameras, sensors, microphones, which are used to detect and track the facial feature. And also consists of a large computation for facial features.

- Requires a special detecting device.
- Large computation is needed.
- Level of accuracy is low.

To present an algorithm that distinguishes true eye blinks from involuntary ones, detects and tracks the desired facial features precisely, and fast enough to be applied in real-time. We used an off-the-shelf webcam that affords a moderate resolution and frame rate as the capturing device in order to make the ability of using the program affordable for all individuals.

- Requires a moderate resolution and frame rate webcam.
- No need for large computation.
- 98% level of accuracy can be obtained.
-

Facial feature lie in a rigid plane and change in position of the projected features is approximated to global affine transformation. Correlation is used to find the correspondences between the two consecutive frames and then estimated the parameters of the affine transformation.

2. System Analysis

In our new project, the application needs a webcam of moderate resolution and preferable frame rate of 30 frames per second. Whereas in the existing system a very high frame rate camera is required. Apart from this it also requires an addition device like infrared sensor to track the facial features.

3. The Process Flow

3.1. Frames

The video input is in the form of frames, each frame is verified and passed to the next level. The categories of the incoming frames are

- The frame dimension should be 320X240
- The frames should be in 'RGB' format.
- The rate of flow of frames should be 30 frames per second.

The video input consists of video images of the capturing device namely the webcam. The input is first checked whether it is suitable for our project.

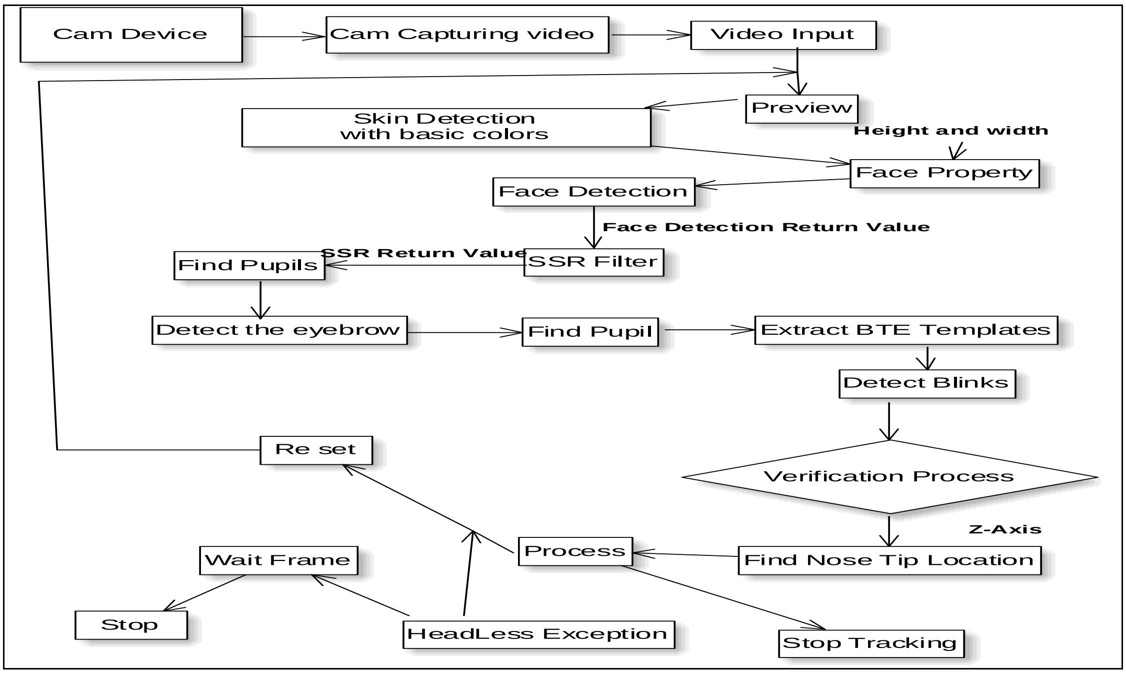

Figure 1: Data flow diagram

3.2. Feature Extraction

Though transform is a feature extraction technique used in digital image processing. In order to achieve this, one must be able to detect a group of pixels that is in a straight line or a smooth curve. In our paper the left and right eye and nose tip is extracted. The purpose of a class diagram is to depict the classes within a model. In an object oriented application, classes have attributes (member variables), operations (member functions) and relationships with other classes. The UML class diagram can depict all these things quite easily. The fundamental element of the class diagram is an icon the represents a class.

A class diagram is a pictorial representation of the detailed system design. Class diagrams are widely used to describe the types of objects in a system and their relationships.

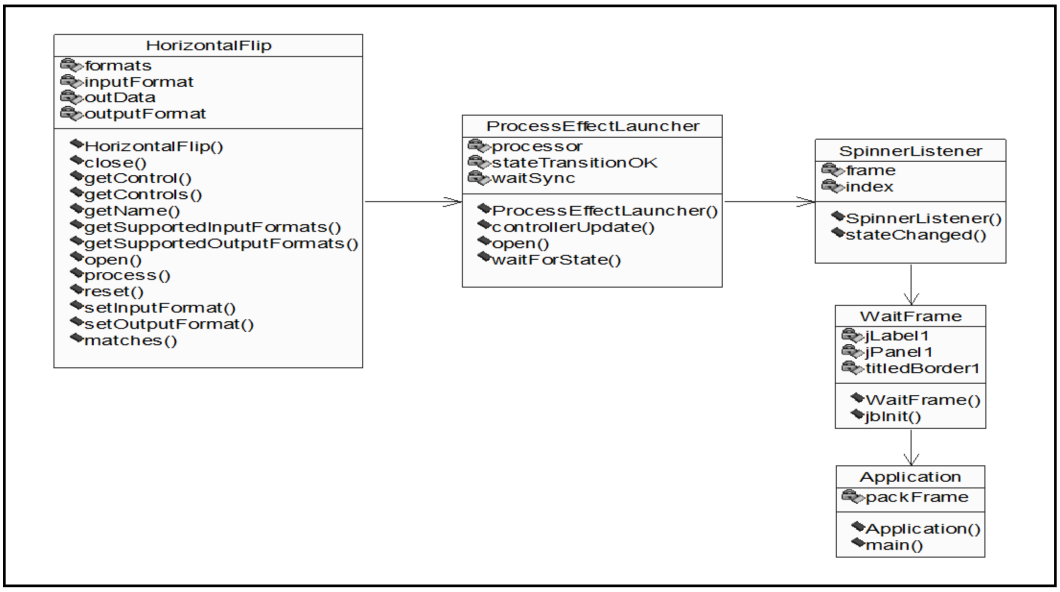

Figure 2: Class diagram 1

4. Implementation

The project implementation consists of four modules which are

- Design module.
- Detection of facial feature.
- Translate nose movement to cursor movement.
- Replacing left/right eye blinks to left/right mouse click event.

The design module consists of three classes namely

- Frame
- Frame_AboutBox
- DevicesFinder

The Frame and Frame_AboutBox classes contain the necessary coding for the design of the application.The DevicesFinder class is used to detect the system whether a suitable webcam is attached and supports the project to run successfully.The second module of the project is the detection of facial features, for this we are using the SSR Filter. This module consists of the following classes

- FaceDetector
- FaceTracker
- ImageProcessor
- ConnectedComponents

4.1. Face detection algorithm overview

The general steps of the algorithm are outlined in the following figure 4.

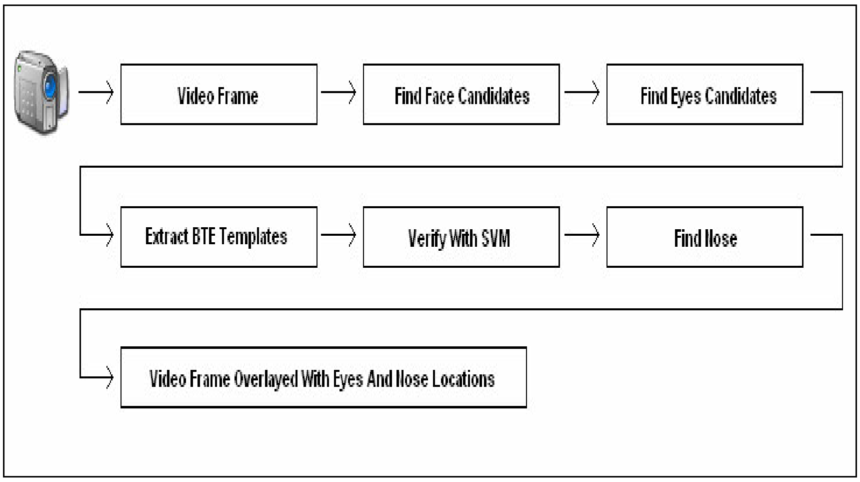

Figure 3:Face Detection Overview

4.2. Find Face Candidates

We will be using feature based face detection methods to reduce the area in which we are looking for the face, so we can decrease the execution time. To find face candidates the SSR filter will be used in the following way: At first we calculate the integral image by making a one pass over the video frame using these equations

$s(x, y) = s(x, y-1) + i(x, y)$

$ii(x, y) = ii(x-1, y) + s(x, y)$

Where $s(x, y)$ is the cumulative row sum, $s(x,-1) = 0$, and $ii(-1, y) = 0$.

4.3. Cluster Face Candidates

Checking all face candidates will be computationally heavy and unnecessary. What we are going to do is to find the clusters of face candidates and consider the center of each cluster as the final candidate. The clustering algorithm that was used is the following:

Passing the image from the upper left corner to the lower right one; for each face candidate fc:

- If all neighbors are not face candidates assign a new label to fc.
- If one of the neighbors is a face candidate assign its label to fc.
- If several neighbors are face candidates assign the label of one of them to fc and make a note that the labels are equal. After making the first pass we will do another one to assign to each group of equal labels a unique label, so the final labels will become the clusters' labels

We propose a real-time face detection algorithm using Six-Segmented Rectangular (SSR) filter, distance information, and template matching technique. Between-the-Eyes is selected as face representative in our detection because its characteristic is common to most people and is easily seen for a wide range of face orientation(combination of featured and image based approach).

Figure 4:Examples of successful Between-the-Eyes detection

4.4. Integral Image

The SSR filter is computed by using intermediate representation for image called integral image. For the original image i(x, y). The integral image can be computed in one pass over the original image by the following pair of recurrences.

$$s(x,y)=s(x,y-1)+i(x,y) \quad (1)$$

$$ii(x,y) = ii(x-1, y) + s(x,y) \quad (2)$$

Where *s(x ,y)* is the cumulative row sum, *s(x , -1) = 0,* and *ii(-1, y) = 0.*

Using the integral image, the sum of pixels within rectangle D (r_s) can be computed at high speed with four array references.

$$s_r = (ii(x,y) + ii(x - W, y - L)) - (ii(x - W, y) + ii(x, y - L)) \quad (3)$$

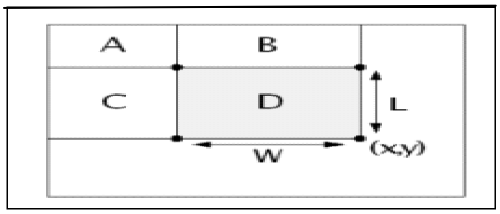

Figure 5: Integral image

4.5. SSR filter

At the beginning, a rectangle is scanned throughout the input image. This rectangle is segmented into six segments as shown in Figure 7.

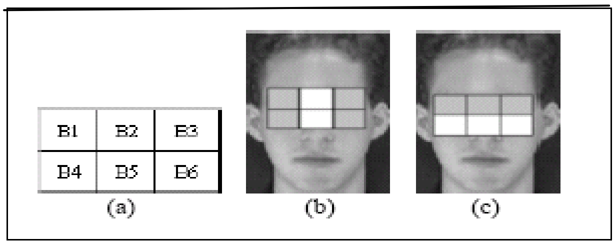

Figure 6: SSR Filter

We denote the total sum of pixel value of each segment (B1 ¡ B6) . The proposed SSR filter is used to detect the Between-the-Eyes based on two characteristics of face geometry. The nose area (n S) is brighter than the right and left eye area (er S and el S , respectively) as shown in Figure 7(b), where

$S_n=S_{b2}+S_{b5}$

$S_{er}=S_{b1}+S_{b4}$

$S_{el}=S_{b3}+S_{b6}$

Then,

$$S_n>S_{er} \quad (4)$$

$$S_n> S_{el} \quad (5)$$

(1) The eye area (both eyes and eyebrows) (e S) is relatively darker than the cheekbone area (including nose) (c S) as shown in Figure 7(c), where

$S_e=S_{b1}+S_{b2}+S_{b3}$

$S_c=S_{b4}+S_{b5}+S_{b6}$

Then,

$$S_e < S_c \quad (6)$$

When expression (4), (5), and (6) are all satisfied, the center of the rectangle can be a candidate for Between-the-Eyes.

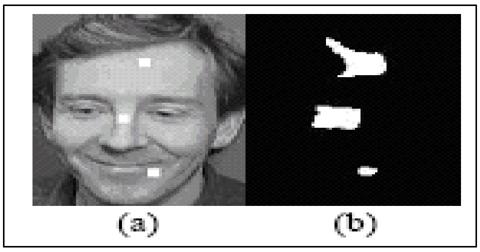

Figure 7: Detecting areas between the eyes

Between-the-Eyes candidates from SSR filter. In Figure 7, the Between-the-Eyes candidate area is displayed as the white areas and the non-candidate area is displayed as the black part. By performing labeling process on Figure 8(b), the result of using SSR filter to detect Between-the-Eyes candidates.

4.6. Filter Size Estimation

In order to find the most suitable filter size, we use 400 facial images of 40 people, were taken at different time, under various lighting condition, at different gesture, and with and without eyeglasses. Each image size is 92×112 with 256 gray levels.

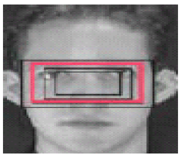

Figure 8: Various size of SSR filter

5. Real-time face detection system:

The third module of this project contains the following classes namely

- SSRFilter
- ProcessEffect
- ProcessEffectLauncher

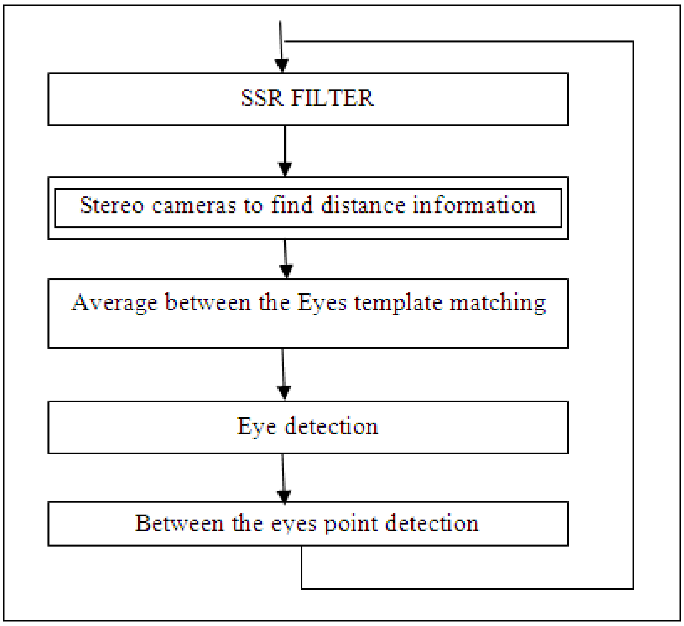

Figure 9: Processing flow of real-time face detection

The SSR Filter class as mention above calculates the integral image, checks the face condition, and nose bridge condition for extraction. The Process Effect class contains all the definition for various process in our project such as Angle limit, Controls the mouse, Blinking, Eyebrows and other operation like open, close, reset of the project. The Process Effect Launcher class contains the implementation of all the process in the Process Effect class and apart from that it also starts the processor for our project.

6. Conclusion

We have demonstrated that facial feature points are obtained and the blink detection process can be successfully tracked in the succeeding frames of an image sequence. The system has worked in real-time and is robust with respect to variations in scaling and lighting conditions, different orientations of the head and presence of distracting objects on the face (such as glasses etc.), and for subjects of different racial origin. However, the system has limitations, primarily because it is intended to track the eyes in near frontal head postures. If the head is rotated more than 45^0 to either side, about a vertical or horizontal axis, the systems will lose track of one eye, but it will continue to track some feature. This will cause tracking of false feature points when the head poses once again becomes more upright. So this needs to reassign the facial feature.

7. Future enhancement

The system also assumes a single user is present in the frame of an image sequence. This limits the possibilities of its application. Future work will need to cope with the presence of multiple faces and multiple users, which can allow multi-user gaze input for applications such as multi-user computer games etc. The same system can be developed using brain signal processing in the future.

8. References

1. Marc Lalonde, David Byrns, Langis Gagnon, Normand Teasdale, Denis Laurendeau, Dept. of Electrical and Computer Engineering, Laval University, Quebec City, QC, Canada, G1K 7P4 http://ieeexplore.ieee.org/Xplore/freeabs_all.jsp?arnumber=4444178
2. G. Chow and X. Li, "Towards a system for automatic facial feature detection", Pattern Recognition, Vol. 26, pp. 1739-1755, 1993.
3. In CameraMouse website: http://www.cameramouse.com/ product.htm.
4. JAVA Complete reference – Herbert Schildt, Tata McGraw Hill Publication, Third Edition
5. Java and JMF website: www.sun.java.com
6. L.V. Calderita and P.Buston, C.SuarezMejias, F.Fernandez, A.Bandera ,Towards an autonomous socially interactive robot for motor and neurorehabilitation therapies for children (IEEE 2013)
7. Jill C.Slaboda, Richard T.Lauer, Emily A.Keshner ., Postural Response of adults with cerebral palsy to combined base of support and visual field rotation. (IEEE 2013)