



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

Simulation of Mouse Using Human Face (HCI)

Snehal Dongre¹, Sachine patil²

Student of M.E. (C.E.), G.H. Rasoni College of Engineering & Management, Ahmednagar, Pune, India

Asst. Prof, Dept. Of I.T., G.H. Rasoni College of Engineering & Management, Wagholi, Pune, India

ABSTRACT: This project is used for an application that it's capable of swapping mouse with human face for interaction with PC. Facial structures (tip, eyes and nose) are traced and tracked in to use their movements of mouse events. Coordinates of the nose tip in the video feed are interpreted to develop the coordinates of the mouse on the screen. The right /left blinks control right/left events for click. The external device will be cam for the video stream.

KEYWORDS: Cursor control system, Eye detection, Eye movement, Face detection, Human-Computer interaction, Support Vector Machine (SVM).

I. INTRODUCTION

In the past few years knowledge becomes progressed and less costly. With high performed processors and economical cams, people are using real-time applications using image processing. The artificial intelligence is one of them. Our work aims to use human face which is interacting with machine. Our system captures feature along with cam and note its act to interpret to events which is communicating with the machine. We are compensating people those have hands disabilities. Our system prevents them from using mouse. Our system uses facial expression which interacts with the machine. The nose tip was selected as the pointing device; the reason behind that decision is the location and shape of the nose; as it is located in the middle of the face it is more comfortable to use it as the feature that moves the mouse pointer and defines coordinates, it is placed on axis that the face take turns about. It is in convex shape which makes it easier to track as the face moves. Eyes were used to simulate clicks. User now fire events. He blinks their eyes while different devices were used in our system. (e.g. infrared cameras, sensors, microphones) we used an off-the-shelf cam 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. We will try to present an algorithm that distinguishes true eye blinks from involuntary ones, detects and tracks the desired facial features.

II. RELATED WORK

With the growth of attention about machine vision, the interest in our project has increased proportionally. As we mentioned before different human features and monitoring devices were used to achieve our project, but during our research we were interested only in works that involved the use of facial features and webcams. We noticed a large diversity of the facial features that were selected, the way they were detected and tracked, and the functionality that they presented for the project. Researchers chose different facial features: eye pupils, eyebrows, nose tip, lips, eye lids' corners, mouth corners for each of which they provided an explanation to choose that particular one. Different detection techniques were applied (e.g. feature based, image based) where the goal was to achieve more accurate results with less processing time. To control the mouse pointer various points were tracked ranging from the middle distance between the eyes, the middle distance between the eyebrows, to the nose tip. To simulate mouse clicks; eye blinks, mouth opening/closing, and sometimes eyebrow movement were used. Each our project method that we read about had some drawbacks, some methods used expensive equipment, some were not fast enough to achieve real-time execution, and others were not robust and precise enough to replace the mouse. We tried to profit from the experience that other researchers gained in the project field and added our own ideas to produce an application that is fast, robust, and useable.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

III. ALGORITHM

A. Design Considerations:

Face Detection Algorithm Overview:

Two main categories:

1. Feature-based methods
2. Image-based methods.

Feature-based methods: first involves finding facial features (e.g. eye brows, lips, eye pupils....)

Image-based methods: scanning the image of interest with a window that looks for faces at all scales and locations.

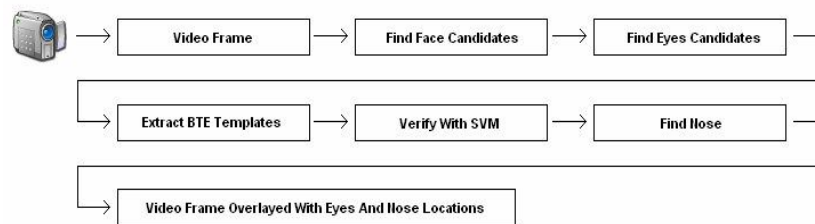


Figure 1: Face detection general steps.

B. Description of the Algorithm:

Our system is believed that people from different races have same skin color but they differ in lighting intensity of that color (e.g. In HSB color model according to this idea people have close H and S values but different B values). In order to discover a suitable skin color model, according to the previously mentioned idea, we will use the pure r and g values which are the R and G values of the RGB color model in the absence of brightness, and they are calculated with the following equations:

$$r = R / (R + G + B) \dots\dots\dots (1)$$

$$g = G / (R + G + B) \dots\dots\dots (2)$$

We move the r and g values to the (a, b) color space with the following equations:

$$a = r + g / 2 \dots\dots\dots (3)$$

$$b = \sqrt{3} / 2 g \dots\dots\dots (4)$$

The range of 'a' is from 0 to 1, while the range of 'b' is from 0 to $\sqrt{3} / 2$ [12].

$$0.49 < 'a' < 0.59 \dots\dots\dots (5)$$

$$0.24 < 'b' < 0.29 \dots\dots\dots (6)$$

We 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 discover face candidates the (SSR) Six Segmented Rectangular filters, filter used in the following way:

Initially we calculate the integral image. We are making a one pass over the video frame using these equations [3]:

$$s(x, y) = s(x, y-1) + i(x, y) \dots\dots\dots (7)$$

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

$$ii(x, y) = ii(x-1, y) + s(x, y) \dots \dots (8)$$

Here's(x, y) is cumulative row sum, s(x,-1) = 0, and ii(-1, y) = 0. Figure 2 shows an ideal location of the SSR filter, where its center is considered as a face candidate.

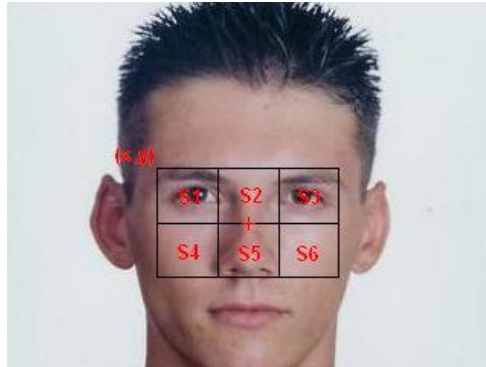


Figure 2: Ideal SSR filter location for a face candidate.

(x, y) is the location of the filter (upper left corner).

The plus sign is the center of the filter which is the face candidate.

We can notice that in this ideal position the eyes fall in sectors S1 and S3, while the nose falls in sector S5.

Since the eyes and eye brows are darker than the BTE and the cheek bones , we deduce that:

$$S1 < S2 \quad \&\& \quad S2 > S3 \dots \dots (9)$$

$$S1 < S4 \quad \&\& \quad S3 < S6 \dots \dots (10)$$

We are using a threshold that is relevant to the size of the currently used SSR filter, to eliminate clusters that are small. The center of each cluster that is big enough is set with the following equations:

$$x = [\sum x(i)] / n \dots \dots (11)$$

$$y = [\sum y(i)] / n \dots \dots (12)$$

i is the pixel from the cluster, n is the cluster's area.

Find Nose Tip:

Now that we located the eyes, the final step is to discover the nose tip.

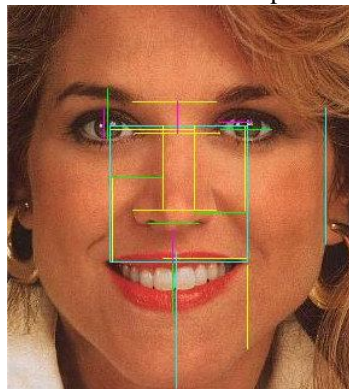


Figure 3: A human face with different lines that define the facial features proportions.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

From figure 3, we can see that the blue line defines a square of the pupils. corners of the mouth; the nose tip should fall inside this square, so this square becomes our region of interest in finding the nose tip (see fig.4). So the first step is to extract the ROI; in case the face was rotated we need to rotate the ROI back to a horizontal alignment of the eyes.

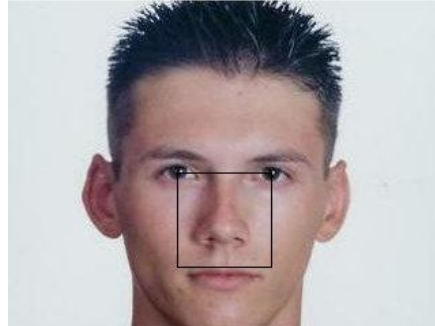


Figure 4: The square that forms the ROI, and the ROI after extraction.

Using the previous idea we tried to locate the nose tip with intensity profiles. In horizontal intensity profiles we add vertically to each line the values of the lines that precedes it in the ROI (see fig.5), so since that the nose bridge is brighter than the surrounding features the values should accumulate faster at the bridge location; in other words the maximum value of the horizontal profile gives us the 'x' coordinate of the nose tip.

IV. PSEUDO CODE

- Step 1: Place the camera in front of human face.
- Step 2: Then human face detection.
- Step 3: Find the face of candidate.
 - If find, go to Step 4
 - else
 - Try again human face detection.
- Step 4: Then find the eye and nose of the candidate.
- Step 5: Then do all the process, which is in Step 6, 7, 8, 9, 10, 11.
- Step 6: Connect the web cam device to machine which gives us RGB format output.
- Step 7: We will fetch all devices which give us RGB format and we will select first one.
- Step 8: We will set a cam on a face to recognize it as a human face and process will get enable.
- Step 9: After we will fetch the region of interest i.e. eye pupils and nose tip.
- Step 10: At real time process will take place. Continuously we fetch the data from webcam in form of frames.
- Step 11: We will operate our mouse pointer according to region of interest on our face.
- Step 12: End.

V. SIMULATION RESULTS

To digitize the (a, b) space we round values to multiplicands of 0.01, so we get 101 values for 'a', and 88 values for 'b'. Finally, to discover the skin color model we extracted 735 skin pixel samples from each of 771 face images which were taken from [6, 7], so the total number of skin pixel samples was 566685 samples. For each of the samples we calculated the 'a' and 'b' values, and increased the counter of that value ('a' has 101 counters, 'b' has 88 counters). The values that were most frequent are considered as skin pixel values, so after plotting the 'a' and 'b' results (see fig. 5 & 6) we deduced the following thresholds for 'a', and 'b' to consider a pixel as a skin pixel.

Using the previous idea we tried to locate the nose tip with intensity profiles. In horizontal intensity profiles we add vertically to each line the values of the lines that precedes it in the ROI (see fig.5), so since that the nose bridge is brighter than the surrounding features the values should accumulate faster at the bridge location; in other words the maximum value of the horizontal profile gives us the 'x' coordinate of the nose tip.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

ROI: Region of interest.

We add horizontally to each column the values of the columns that precedes it in the ROI (see fig. 7); the same as in the horizontal profile, the values accumulate faster at the nose tip position so the maximum value gives us the 'y' coordinate of the nose tip. From both, the horizontal and vertical profiles we were able to locate the nose tip position (see fig. 8), but unfortunately this method did not give accurate results because here might be several max values in a profile that are close to each other, and choosing the correct max value that really points out the coordinates of the nose tip is a difficult task.

Each NBP will add with its S2 sector a certain amount to the accumulated sum in the horizontal profile, but the NBP at the nose trills location will add a smaller amount (S2 sector of the NBP at the nose trills location is darker than S2 sector of other NBPs), so if we calculated the first derivate of the values of NBPs' S2 sectors (the first derivate of the values of the maximum value of the horizontal profile at each ROI line) we will notice a local minima at the nose trills location (see fig. 9); by locating this local minima we take the NBP that corresponds to it as the nose trills location, and the next step is to look for the nose tip above the nose trills. Since the nose tip is brighter than other features it will donate with its S2sector to the accumulated sum more than other NBPs, which means a local maxima in the first derivate (see fig. 9); so the location of the nose tip is the location of the NBP that corresponds to the local maxima that is above the local minima in the first derivate.

In tracking mode the results were very robust when:

1. The frame rate is 20 fps and higher; the user can move very fast without having the program losing his facial features.
2. The glasses reflect light and cause bright spots that sometimes force our program to lose track of the eyes.

For the detection and tracking to be accurate and robust the lighting conditions must be set so the light is frontal in a way that it will spread evenly on the face, because side light will cause false face detection and will affect eventually the tracking process.

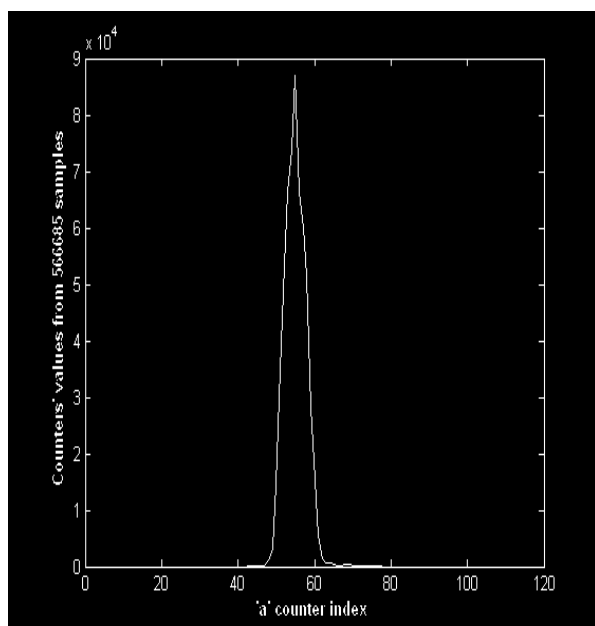


Figure 5: A plot of 'a' counters over 566685 skin samples.

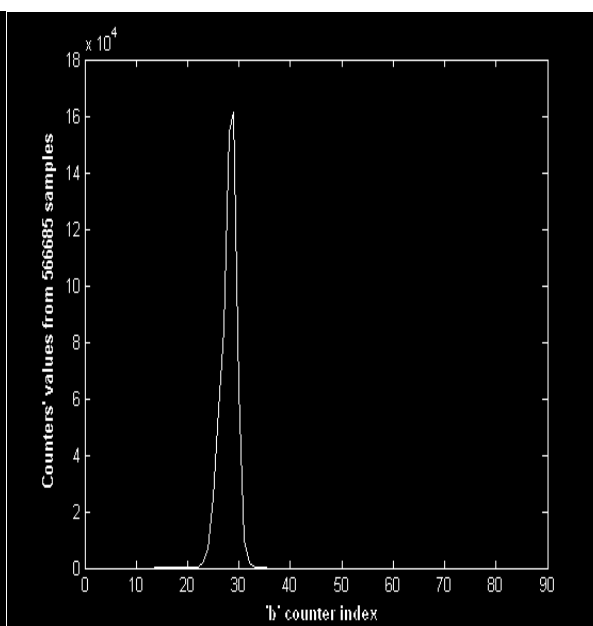


Figure 6: A plot of 'b' counters over 566685 skin samples.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

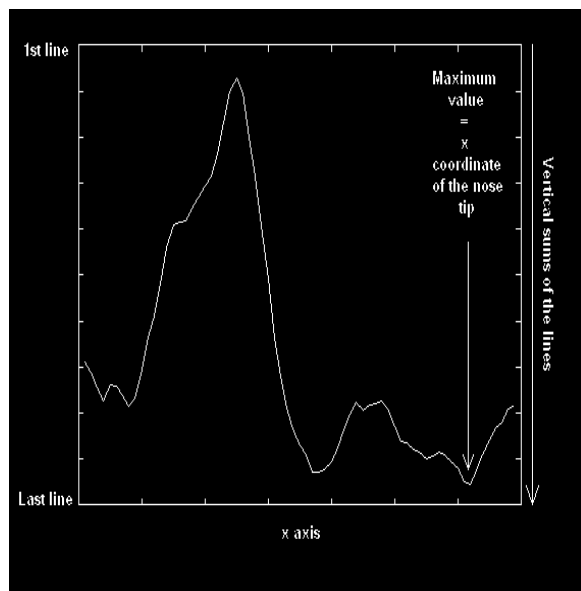


Figure 7: The horizontal profile of the ROI from figure 4.

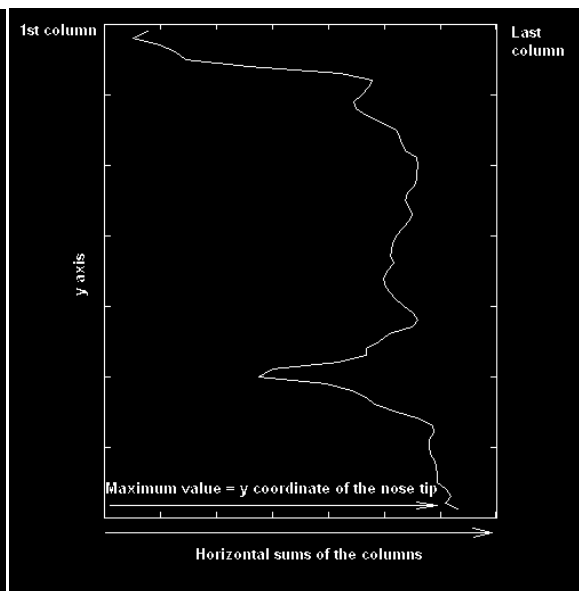


Figure 8: The vertical profile of the ROI from figure 4.

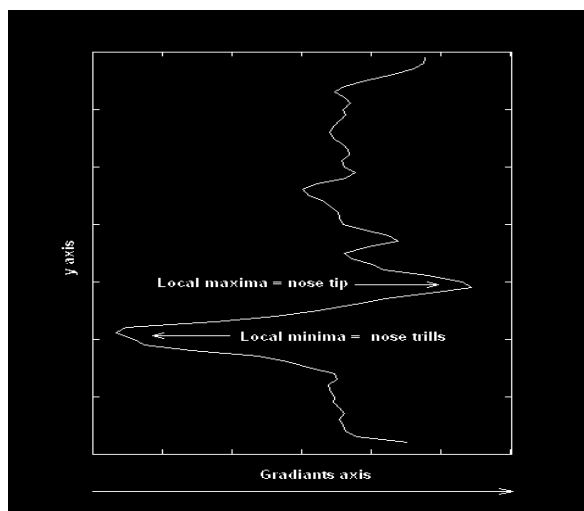


Figure 9: First derivate of the values of NBPs' S2 sectors.

VI. CONCLUSION AND FUTURE WORK

We are very promising; the following conclusions were made based on the experiments that were held during the expo. In detection mode the eyes and nose tip were located accurately when the following conditions were fulfilled:

1. The face is not rotated more than 5° around the axis that passes from the nose tip (as long as the eyes fall in sectors S1 and S3 of the SSR filter).
2. The face is not rotated more than 30° around the axis that passes from the neck (profile view).
3. Wearing glasses does not affect our detection process.
4. As for different scales it is best to get about 35 cm close to the cam, because when the face is a bit far from the screen the program may detect a false positive especially when the background is jammed (crowded).



ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

Feature works may include improving the tracking robustness against lighting conditions; perhaps by using more sophisticated and expensive capturing devices such as infrared cameras that can operate in absence of light and give more accurate tracking results. Adding the double left click (detecting the double left eye blink) and the drag mode (enabling/disabling with the right double eye blink) functionalities. Adding voice commands to launch the program, start the detection process, and to enable/disable controlling the mouse with the face.

REFERENCES

1. M.Mangaiyarkarasi and A.Geetha, “**Cursor control system using facial expressions for human- computer interactions**” .International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 8 Issue 1 –APRIL 2014.
2. Shrunkhala Satish Wankhede, Ms. S.A. Chhabria and Dr. R.V. Dharaskar, “**Controlling mouse cursor using eye movements**” .International Journal of Application or Innovation in Engineering & Management (IJAIEM) 2013.
3. Tarun Dhar Diwan, Rajesh Tiwari, “**Automatic Eye Blink Tracking and Detection in an Image Sequence**”. International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 2. (5), 2011.
4. Akhil Gupta, Akash Rathi, Dr. Y. Radhika, “**Hands free pc control**” controlling of mouse cursor using eye movements. International Journal of Scientific and research publication, Volume 2, Issue 4, April 2012.
5. V. Pradeep, “**A simple algorithm for using face as a pointing device using openvc**”. International Journal of advanced research in computer science and software engineering, Volume 2, Issue 2, February 2012.
6. Erik Hjelm and Boon Kee Low, “**Face detection : A survey**”.Machine Vision and Image Understanding 83, 236–274 (2001).

BIOGRAPHY

Snehal Dongre is a Student in the Computer Engineering Department, College of G. H. Raisoni College of engineering and management, Chas, Ahmednagar, Pune University. She received B.E degree in 2012 from Nagpur, MH, India. Her research interests are Image Processing, HCI, Algorithms etc.