"HANDS-FREE PC CONTROL" CONTROLLING OF MOUSE CURSOR USING EYE MOVEMENT

Akhil Gupta, Akash Rathi, Dr. Y. Radhika

Abstract- The paper presents hands free interface between computer and human. This technology is intended to replace the conventional computer screen pointing devices for the use of disabled or a new way to interact with mouse. the system we describe is real time, on-intrusive, fast and affordable technique for tracking facial features. The suggested algorithm solves the problem of occlusions and is robust to target variations and rotations. It is based on novel template matching technique. A SSR Filter integral image, SVM is used for adaptive search window positioning and sizing.

Index Terms- face Recognition, SRR, SVM, and template matching

I. Introduction

This technology is intended to be used by disabled people who face a lot of problems in communicating with fellow human beings. It will help them use their voluntary movements, like eyes and nose movements: to control computers and communicate through customized, educational software or expression building programs. People with severe disabilities can also benefit from computer access and take part in recreational activities, use internet or play games. This system uses a usb or inbuilt camera to capture and detect the user‟s face movement. The proposed algorithm tracks the motion accurately to control the cursor, thus providing an alternative to computer mouse or keyboard.

Primarily approaches to camera-based computer interfaces have been developed. However, they were computationally expensive, inaccurate or suffered from occlusion. For example, *the head movement tracking system* is the device that transmits a signal from top of computer monitor and tracks a reflector spot placed on user forehead. This technique is not completely practical as some disabled cannot move their head and it becomes in accurate when someone rotates its head. *Electro-oculography (EOG) is* a technology where a electrode around user eye record the movement. The problems with this technique is that for using this the disabled person needs someone help to put it and also the system is quite expensive.

Another example is *CAMSHIFT* algorithm uses skin color to determine the location and orientation of head. This technique is fast and does not suffer from occlusion; this approach lacks precision since it works well for translation movement but not rotational movement.

II. FLOW OF USING THE APPLICATION

2.1 Face Detection

Face detection has always been a vast research field in the computer vision world. Considering that it is the back bone of any application that deals with the human face. The face detection method can be organized in two categories:

2.1.1 Feature-based method:

The first involves finding facial features (e.g. noses, eye brows, lips, eye pupils) and in order to verify their authenticity performs by geometrical analysis of their locations, areas and distances from each other. This analysis will eventually lead to localization of the face and the features that it contains. The feature based analysis is known for its pixel-accuracy, features localization and speed, on the other hand its lack of robustness.

2.1.2 Image-based method:

The second method is based on scanning the image of interest with a window that looks for faces at all scales and locations. This category of face detection implies pattern recognition, and achieves it with simple methods such as template matching or with more advanced techniques such as neural networks and support vector machines. Before over viewing the face detection algorithm we applied in this work here is an explanation of some of the idioms that are related to it.

2.2 SIX SEGMENTED RECTANGULAR FILTER [ssr]

At the beginning, a rectangle is scanned throughout the input image. This rectangle is segmented into six Segments as shown in Fig. (1).

S1	S2	S3
S4	S5	S6

Fig 1: segments of rectangle

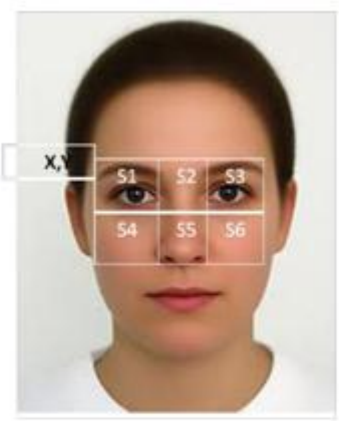


Fig 2: SSR Filter

We denote the total sum of pixel value of each segment (S1-S6). The proposed SSR filter is used to detect the Between-the-Eyes[BTE] based on two characteristics of face geometry. (1) The nose area (S_n) is brighter than the right and left eye area (eye right (S_{er}) and eye left (S_{el}), respectively) as shown in Fig.(2), where

S_n = S2 + S5

S_{er} = S1 + S4

S_{el} = S3 + S6

Then,
Sn >
Ser (1)

$S_n > S_{el}$ (2)

(2) The eye area (both eyes and eyebrows) (S_e) is relatively darker than the cheekbone area (including nose) (S_c) as shown in Fig. (2), where

S_e= S1 + S2 + S3

S_c= S4 + S5 + S6

Then,

$S_e < S_c$ (3)

When expression (1), (2), and (3) are all satisfied, the center of the rectangle can be a candidate for Between-the-Eyes

2.3 INTEGRAL IMAGE

In order to assist the use of Six-Segmented Rectangular filter an immediate image representation called “Integral Image” has been used. Here the integral image at location x, y contains the sum of pixels which are above and to the left of the pixel x, y [10] (Fig. 3).

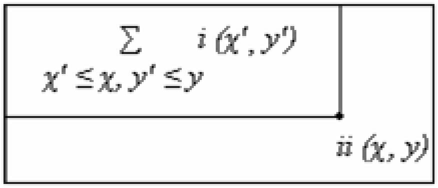


Fig 3: Integral Image; i: Pixel value; ii: Integral Image

So, the integral image is defined as:

$ii\ (\chi, y) = \Sigma\ i(\chi', y')$ (1)

$\chi' \leq \chi,\ y' \leq y$

With the above representation the calculation of the SSR filter becomes fast and easy. No matter how big he sector is, with 3 arithmetic operations we can calculate the pixels that belong to sectors (Fig. 4).

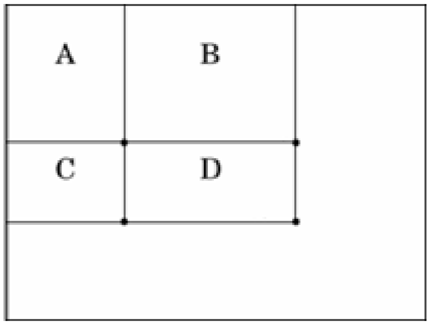


Fig. 4: Integral Image, The sum of pixel in Sector Dis computed as Sector =D - B - C + A

The Integral Image can be computed in one pass over the original image (video image) by:

$s(\chi,y) = s(\chi,y-1)+i(\chi,y)$ (2)
$ii\ (\chi,y)=ii(\chi -1,y)+s(\chi,y)$ (3)

Where $s(\chi , y)$ is the cumulative row sum, $s(\chi, -1) = 0$, and $ii(-1, y) = 0$. Using the integral image, any rectangular sum of pixels can be calculated in four arrays (Fig. 4).

2.4 SUPPORT VECTOR MACHINES [SVM]

SVM are a new type of maximum margin classifiers: In “learning theory‟ there is a theorem stating that in order to achieve minimal classification error the hyper plane which separates positive samples from negative ones should be with the maximal margin of the training sample and this is what the SVM is all about. The data samples that are closest to the hyper plane are called support vectors. The hyper plane is defined by balancing its distance between positive and negative support vectors in order to get the maximal margin of the training data set.

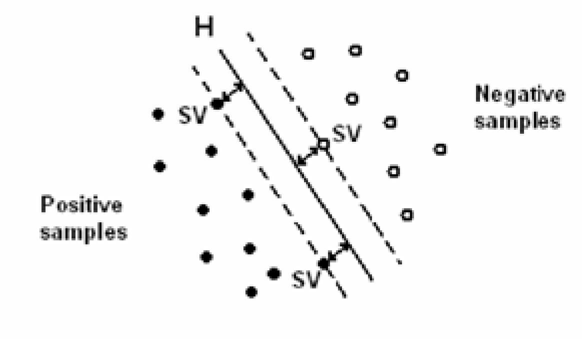


Fig 5: Hyper plane with the maximal margin.

Training pattern for SVM

SVM has been used to verify the BTE template. Each BTE is computed as: Extract 35 wide by 21 high templates, where the distance between the eyes is 23 pixels and they are located in the 8throw (Fig 5, 6, 7). The forehead and mouth regions are not used in the training templates to avoid the influence of different hair, moustaches and beard styles.

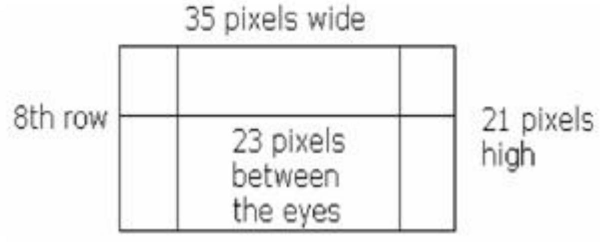


Fig 6: How to extract the training template

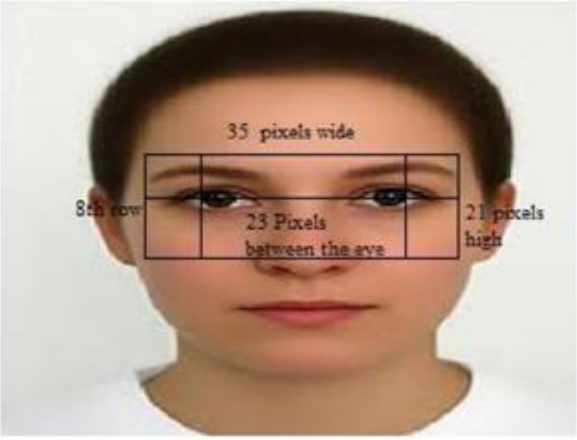


Fig. 7: Illustration of the training template

The training sample attributes are the templates pixels‟ values, i.e., the attributes vector is 35*21 long. Two local minimum dark points are extracted from (S1+S3) and (S2+S6) areas of the SSR filter for left and right eye candidates. In order to extract BTE temple, at the beginning the scale rate (SR) were located by dividing the distance between left and right pupil‟s candidate with 23, where 23 is the distance between left and right eye in the training template and are aligned in 8th row. Then:

- Extract the template that has size of 35 x SR x 21, where the left and right pupil candidate are align on the 8 x SR row and the distance between them is 23 x SR pixels.

And then scale down the template with SR, so that the template which has size and alignment of the training templates is horizontally obtained, as shown in Fig. 8.

Fig. 8: Face pattern for SVM learning

In this section we have discussed the different algorithm‟s required for the face detection that is:

SVM, SSR, Integral image and now we will have a detailed survey of the different procedures involved: face tracking, motion detection, blink detection and eyes tracking.

III. FACE DETECTION ALGORITHM

OVERVIEW:

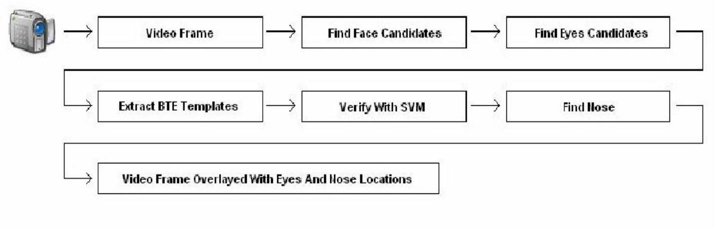


Fig. 9: Algorithm 1 for finding face candidates

3.1 Face Tracking

Now that we found the facial features that we need, using the SSR and SVM, Integral Image methods we will be tracking them in the video stream. The nose tip is tracked to use its movement and coordinates as them movement and coordinates of the mouse pointer. The eyes are tracked to detect their blinks, where the blink becomes the mouse click. The tracking process is based on predicting the place of the feature in the current frame based on its location in previous ones; template matching and some heuristics are applied to locate the feature’s new coordinates.

3.2 Motion Detection

To detect motion in a certain region we subtract the pixels in that region from the same pixels of the previous frame, and at a given location (x,y); if the absolute value of the subtraction was larger than a certain threshold, we consider a motion at that pixel .

3.3 Blink detection

We apply blink detection in the eye’s ROI before finding the eye’s new exact location. The blink detection process is run only if the eye is not moving because when a person uses the mouse and wants to click, he moves the pointer to the desired location, stops, and then clicks; so basically the same for using the face: the user moves the pointer with the tip of the nose, stops, then blinks. To detect a blink we apply motion detection in the eye‟s ROI; if the number of motion pixels in the ROI is larger than a certain threshold we consider that a blink was detected because if the eye is still, and we are detecting a motion in the eye‟s ROI, that means that the eyelid is moving which means a blink. In order to avoid multiple blinks detection while they are a single blink the user can set the blink‟s length, so all blinks which are detected in the period of the first detected blink are omitted.

3.4 Eyes Tracking

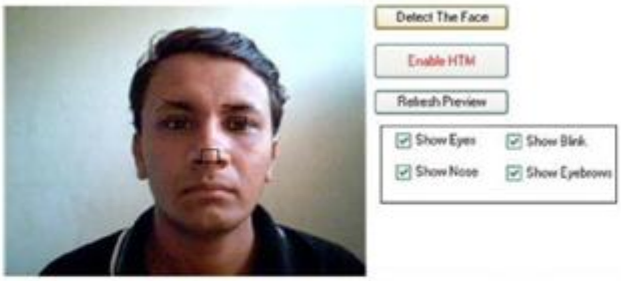
If a left/right blink was detected, the tracking process of the left/right eye will be skipped and its location will be considered as the same one from the previous frame (because blink detection is applied only when the eye is still). Eyes are tracked in a bit different way from tracking the nose tip and the BTE, because these features have a steady state while the eyes are not (e.g. opening, closing, and blinking) To achieve better eyes tracking results we will be using the BTE (a steady feature that is well tracked) as our reference point; at each frame after locating the BTE and the eyes, we calculate the relative positions of the eyes to the BTE; in the next frame after locating the BTE we assume that the eyes have kept their relative locations to it, so we place the eyes‟ ROIs at the same relative positions to the new BTE. To find the eye‟s new template in the ROI we combined two methods: the first used template matching, the second searched in the ROI for the darkest5*5 region (because the eye pupil is black), then we used the mean between the two found coordinates as the new eye‟s location.

IV. IMPLEMENTATION AND APPLICATIONS

4.1 Interface

The following setup is necessary for operating our “hands-free pc control”. A user sits in front of the computer monitor with a generic USB camera widely available in the market, or a inbuilt camera.

Our system requires an initialization procedure. The camera position is adjusted so as to have the user‟s face in the center of the camera‟s view. Then, the feature to be tracked, in this case the eyes, eyebrows, nostril, is selected, thus acquiring the first template of the nostril (Figure 10). The required is then tracked, by the procedure described in above section, in all subsequent frames captured by the camera.



☑ Show Eyes ☑ Show Blink
☑ Show Nose ☑ Show Eyebrows

Fig 10: A screen shot of initialization window.

4.2 Applications

Hands-free PC control can be used to track faces both precisely and robustly. This aids the development of affordable vision based user interfaces that can be used in many different educational or recreational applications or even in controlling computer programs.

4.3 Software Specifications

We implemented our algorithm using JAVA in Jcreator a collection of JAVA functions and a few special classes that are used in image processing and are aimed at real time computer vision.

The advantage of using JAVA is its low computational cost in processing complex algorithms, platform independence and its inbuilt features Java Media Framework[JMF]2.1 which results in accurate real time processing applications.

V. CONCLUSION AND FUTURE DIRECTIONS

This paper focused on the analysis of the development of hands-free PC control - Controlling mouse cursor movements using human eyes, application in all aspects. Initially, the problem domain was identified and existing commercial products that fall in a similar area were compared and contrasted by evaluating their features and deficiencies. The usability of the system is very high, especially for its use with desktop applications. It exhibits accuracy and speed, which are sufficient for many real time applications and which allow handicapped users to enjoy many compute activities. In fact, it was possible to completely simulate a mouse without the use of the hands. However, after having tested the system, in future we tend to add additional functionality for speech recognition which we would help disabled enter password’s/or type and document verbally.

ACKNOWLEDGMENT

We would like to thank Prof Mrs. D.Rajya Lakshmi for their guidelines in making this paper.

REFERENCES

[1] H.T. Nguyen,”.Occlusion robust adaptive template tracking.”. Computer Vision, 2001.ICCV 2001 Proceedings

[2] M. Betke. “the camera mouse: Visual Tracking of Body Features to provide Computer Access For People With Severe Disabilities.” IEEE Transactions on Neural Systems And Rehabilitation Engineering. VOL. 10. NO 1. March 2002.

[3] Abdul Wahid Mohamed, “Control of Mouse Movement Using Human Facial Expressions” 57, Ramakrishna Road, Colombo 06, Sri Lanka.

[4] Arslan Qamar Malik, “Retina Based Mouse Control (RBMC)”, World Academy of Science, Engineering and Technology 31,2007

[5] G .Bradsky.” Computer Vision Face Tracking for Use in a Perceptual User Interface”. Intel Technology 2nd Quarter ,98 Journal.

[6] A. Giachetti, Matching techniques to compute image motion. Image and Vision Computing 18(2000). Pp.247-260.

AUTHORS

First Author – Akhil Gupta, 4/4 B.Tech, Department of Information Technology, Gitam Institute of Technology, Email- akhil09gpt@gmail.com

Second Author – Akash Rathi**,** 4/4 B.Tech, Department of Computer Science and Engineering Gitam Institute of Technology, Email- akash.rapidfire@gmail.com

Third Author - Dr. Y Radhika(PhD), Associate professor, Department of Computer Science and Engineering, Gitam Institute of Technology, Email- radhika@gitam.edu