Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 1, January 2014, pg.418 – 422

RESEARCH ARTICLE

Real Time HCI using Eye Blink Detection

R.T.Narmadha[1], T.Mythili[2], R.T.Nivetha[3]

[1]Information Technology & Info Institute of Engineering, India
[2]Information Technology & Info Institute of Engineering, India
[3]Computer Science and Engineering & Sri Eshwar College of Engineering, India
[1] narmadhahoney@gmail.com; [2] sathyamythili@gmail.com; [3] r.t.nivetha@gmail.com

Abstract— ***This project aims to design an effective and efficient HCI both in terms of performance and cost. It also designs a Hands-free interface which helps to interact with the computer by using the facial features. To select a robust facial feature, we use the pattern recognition paradigm of treating features. We are using an off-the-shelf webcam that affords a moderate resolution and frame rate as the capturing device. It compensates people who have hand disabilities that prevent them from using the mouse by designing an application that uses facial features to interact with the computer.***

Keywords— HCI; Hands free interface; Pattern Recognition Paradigm; Facial feature; Off-the-shelf webcam

I. INTRODUCTION

The focus of new human-computer interfaces has become a growing field today, which aims to attain the development of more natural, intuitive, unobtrusive and efficient interfaces. The trend uses Perceptual User Interfaces (PUIs) that are turning out to be very popular as they seek to make the user interface more natural [1]. It also takes the advantage of the ways in which people naturally interact with each other and with the world. When sitting in front of a computer and with the use of webcams, very common devices nowadays, heads and faces can be assumed to be visible. Therefore, system's based in head or face feature detection and tracking, and face gesture or expression recognition can become very effective human computer interfaces. Different approaches have been used for non invasive face/head-based interfaces [2]. For the control of the position some systems analyze facial cues such as color distributions, head geometry or motion [3]. Other works track facial features or gaze including infrared lighting. To recognize the user's events it is possible to use facial gesture recognition [4]. In our project we consider facial gestures and the atomic facial feature motions.

II. PROPOSED HANDS FREE INTERFACE SYSTEM

We consider the problem of tracking faces using a video camera and focus our attention on the design of vision-based perceptual user interfaces. This is the system which uses a video camera to track user's face position in 3D in order to convert it to a position of a cursor or another virtual object in 2D screen.

The concept of the second order change detection, which allows one to discriminate the local (most recent) change in image, such as blink of the eyes, is introduced. This concept sets the base for designing complete face-operated control systems, in which, using the analogy with mouse, "pointing" is performed by nose and "clicking" is performed by blinking of the eyes.

Edge-based features create a problem in tracking the objects which may rotate, since these features are not invariant to the rotation and the change of scale of the object. In order to select a robust facial feature, we use the pattern recognition paradigm of treating features. According to this paradigm, a feature is associated with a vector made of feature attributes. Feature attributes can be pixel intensities or they can be the parameters of geometric primitives.

Our system also distinguishes true eye blinks from involuntary ones. It is very fast enough to implement in real time. We are using 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.

The general flow emphasizes the importance of appropriate feature selection for successful tracking, introduces the concept of a convex-shape feature and defines the nose feature. Then the flow describes the face tracking technique and eye blinks detection techniques.

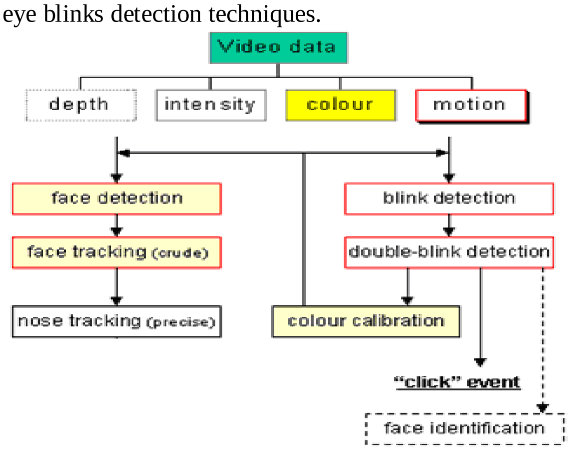

Fig.1 Hierarchy of Tasks in the Eye-Blinks System

III. FEATURE EXTRACTION

A. Desired Feature Properties

In order to select a robust facial feature, the pattern recognition paradigm of treating features is used. According to this paradigm, a feature is associated with a vector made of feature attributes. Feature attributes can be pixel intensities or they can be the parameters of geometric primitives. In the case of template-based feature tracking, feature attributes are the intensity values obtained by centering a peephole mask on the position of the feature. For a facial feature to be easily recognized and robustly tracked it should possess the uniqueness and robustness properties, defined as follows. The uniqueness property states that a feature vector should lie as far as possible from other vectors in multidimensional space of feature attributes. The robustness property, on the other hand, states that a feature vector should not change much during the tracking process.

B. Nose Feature

A human face has a salient convex shape feature, which is the tip of the nose. Due to the symmetry and the convex shape of the nose, the nose feature is always visible in the camera, and it stays almost the same during the rotations of the head. It also does not change much with head moving towards and from the camera. Thus, the nose tip defined above can always be located, which is a very important property of the nose which does not hold for any other facial feature. It gives a user the flexibility and convenience of head motions.

IV. FACE DETECTION

A. Facial Features Detection

1) Nose Tip Detection: At first, the user must stay steady for a few frames for the process to be initialized. Face detection will be considered robust when during a few frames the face region is detected without changes Fig 2. Then, it is possible to define the initial user's face region to start the search of the user's facial features. Over the nose region, we look for those points that can be easily tracked, that is, those whose derivative energy perpendicular to the prominent direction is above a threshold. Basically it selects the nose corners or the nostrils. However, the ambient lighting can cause the selection of points that are not placed over the desired positions; this fact is clearly visible in Fig.3. Fig. 4 shows the selected features that we consider due to their symmetry respect to the vertical axis. This reselection process will achieve the best features to track and it will contribute to the tracking robustness. The final point illustration is shown in Fig. 5, that is, the mean point of all the final selected features that due to the reselection of points will be centered on the nose. This mean point is referred as "Nose Tip".

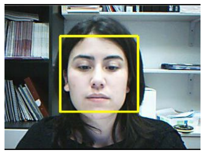

Fig.2 Automatic Face Detection

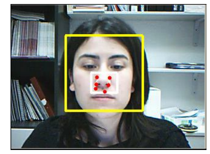

Fig.3 Initial set of Features

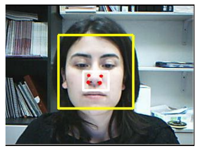

Fig.4 Best Feature Selection using Symmetrical Constraints

Fig.5 Mean of all Features: Nose Point

2) *Eye Detection:* The next step of is to build the user's eyes models. Both eyes can be located using the skin color model and clustering concept. First, the region is binarized to find the dark zones, and then we keep the bounding boxes of the pair of blobs that are symmetrical and are located nearer to the nose region. This way, the eyebrows or the face borders should not be selected. Detection of eyes is performed on the idea that eye color is different to the other facial features color (taking in account that the eye color distribution is composed by sclera and iris colors). Then, other eye candidates are detected using their variations in color distributions. Similarly our system could be used by users with clear (blue or green) or dark (black or brown) eyes. Eye models are obtained through histogram techniques in the RGB color space of the pixels belonging to the detected eye regions.

Fig.6 Example of Eyes' Detection: the blobs that are selected (in red color) are symmetrical and are nearer to the nose region

V. TRACKING

Hands-free control is achieved by tracking the convex-shape nose feature in video stream captured by the camera. Since the convex-shape features may not possess the uniqueness property, the first step is to calculate the area for local search of the nose feature using skin color, frame subtraction and the knowledge of the feature position in the previous frame, if available. The second step consists in scanning the local search area and finding pixel 'u' which has the shape the closest to that of the selected feature in terms of correlation with the template feature. Before proceeding to this step, the image is reprocessed with the Gaussian filter to smooth the defects of images caused by low quality of the cameras. The final step is to refine the position of the best match, using the evidence-based convolution filter.

A. Gesture Recognition

The gestures considered here are eye ball motions. The major part of the work uses high quality images and good image resolution in the eyes' zones. However, movement recognition with webcam quality images is difficult. Besides, this process depends on the user's head position. Robustness is the degree to which a system operates correctly in the presence of exceptional inputs or stressful environmental conditions [6][8][11].

The user is asked to move the head in all possible rotations while still looking at the computer screen: "yes" (up-down), "no" (left-right), and "don't know" (clockwise) motions and also "scale" (body) motion. The range of head motion within which the convex-shape nose tip feature successfully tracked is determined. The nose tip is tracked for up to about 35-40 degrees of rotation of head in all three directions. This range of allowable motion practically covers the range a user may exhibit while looking at the screen.

The different positions of the head and their correct detections and the corresponding tracking are shown in the Fig 7.

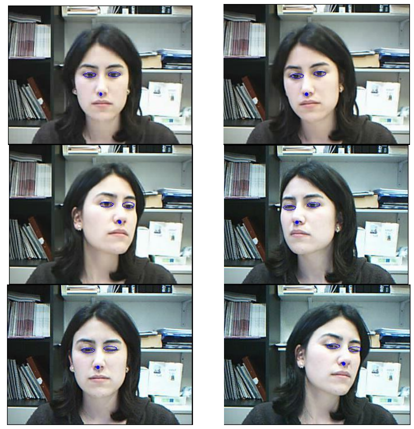

Fig 7. Facial Feature Tracking Results

B. Mean Shift Procedure

Mean shift procedure is also used to detect the eye blink [5]. The process starts by detecting the vertical contours in the image. For avoiding false positives in the process, the vertical contours are logically operated with a mask which was generated by thresholding the original image. Finally we keep the two longest vertical edges of each eye region if they appear to get the eye candidates. If these two vertical edges which correspond to the eye iris edges don't appear after the process for a number of consecutive frames, for gesture consistency, we will assume that the eye is closed. In Fig 8. the process for the above gesture recognition is described.

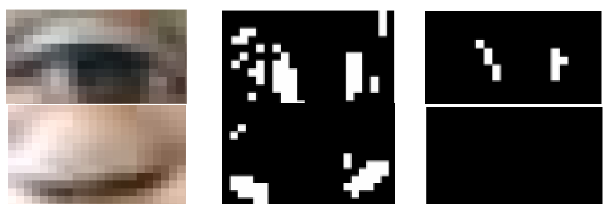

Fig8. Process for Recognizing Blinks. The first row shows the process applied to open eyes. The second row represents the process over closed eyes. (a) Original Image (b) Vertical Edges (c) Iris Contours

VI. CONCLUSIONS

The proposed system includes a new mixture of several computer vision techniques, where some of them have been improved and enhanced to reach more stability and robustness in tracking and gesture recognition. Numerical and visual results are given. In order to build reliable and robust perceptual user interfaces based on computer vision, certain practical considerations must be taken in account. An effective interface is provided in order to replace the standard mouse motions and events. A Hands-free interface is designed which helps to interact with the Computer by using the facial features.

REFERENCES

[1] Cristina Manresa-Yee, Javier Varona, and Francisco J. Perale (2005), "Towards Hands-Free Interfaces Based on Real-Time Robust Facial Gesture Recognition", In Proceedings of the IEEE Workshop on Applications of Computer Vision on 2005, pages: 214-219

[2] Fagiani, C., Betke, M., Gips, J, (2002) "Evaluation of Tracking Methods for Human-Computer Interaction", In: Proceedings of the IEEE Workshop on Applications in Computer Vision, 121–126

[3] Alexander, D.C., Buxton, B.F (2001) "Statistical Modeling of Colour Data", International Journal of Computer Vision 44, 87–109

[4] Oraya Sawettanusorn, Yasutaka Senda, Shinjiro Kawato, Nobuji Tetsutani, and Hironori Yamauchi, (2003) "Real-Time Face Detection using Six Segmented Rectangular Filter", ACM proceedings Internal ATR Media Information Science Laboratories, Kyoto, Japan

[5] T. Moriyama, T. Kanade, J. Cohn, J. Xiao, Z. Ambadar, and H. Gao., (2002) "Automatic recognition of eye blinking in spontaneously occurring behavior". In Proc. Int. Conf. Patt. Rec.,

[6] D. G. Lowe, "Distinctive image features from scale-invariant key points. Int. J. of Computer Vision", 91–110, 2004

[7] Toyama, K., (1998) "Look, Ma – No Hands! Hands-Free Cursor Control with Real-Time 3D Face Tracking": In Proceedings of the Workshop on Perceptual User Interfaces, 49–54

[8] Viola, P., Jones, M, (2004) "Robust Real-Time Face Detection". International Journal of Computer Vision" , 137–154
[9] T. N. Bhaskar, F. T. Keat, S. Ranganath, and Y. V. Venkatesh, (2003) "Blink detection and eye tracking for eye localization", In Proc. TENCON 2003, volume 2, pages 821– 824
[10] P. Ekman, "Facial Action Coding System: a technique for the measurement of facial movement", Consult. Psych. Press, 1978.
[11] Haro, M. Flickner, and I. Essa,(2000) "Detecting and tracking eyes by using their physiological properties, dynamics, and appearance", In IEEE International Conference on Computer Vision and Pattern Recognition, volume 1, pages 163–168

