Face-Based Perceptual Interface for Computer-Human interaction

Cristina Manresa-Yee, Javier Varona and Francisco J. Perales
Universitat de les Illes Balears
Departament de Matemàtiques i Informàtica
Ed. Anselm Turmeda. Crta. Valldemossa km. 7.5
07122 Palma
{cristina.manresa, javier.varona, paco.perales}@uib.es

ABSTRACT

Nowadays accessibility to new technologies for everyone is a task to accomplish. A way to contribute to this aim is creating new interfaces based on computer vision using low cost devices such as webcams. In this paper a face-based perceptual user interface is presented. Our approach is divided in four steps: automatic face detection, best face features detection, feature tracking and face gesture recognition. Using facial feature tracking and face gesture recognition it is possible to replace the mouse's motion and its events. This goal implies the restriction of real-time response and the use of unconstrained environments. Finally, this application has been tested successfully on disabled people with problems in hands or arms that can not use the traditional interfaces such as a mouse or a keyboard.

Keywords

Visual Tracking, Gesture Recognition, Perceptual User Interfaces, Assistive Technologies, e-Inclusion, Information and Communication Technologies, Human-Computer Interfaces

1. INTRODUCTION

Nowadays accessibility to new technologies for everyone is a must-do task to accomplish. One way of achieving this aim is to provide new natural user interfaces at a low cost. Perceptual User Interfaces (PUIs) offer a more natural human-computer interaction (HCI) by means of trying to "perceive" what the user is doing using computer vision, speech recognition and/or sketch recognition [Lan02a].

The use of computer vision techniques for creating perceptual interfaces is a growing field in research because they offer great advantages over other systems: non intrusive devices on the user and hand-free control.

These kind of systems are oriented to take into account the needs of all users, including those with disabilities. Assistive technologies (AT) and e-Inclusion have become a major driving force for an open information society. Therefore, these interfaces can be very useful for disabled people with problems in hands or arms or it can be used for leisure purposes.

Taking advantage of the benefits that vision-based interfaces yield and using a low cost standard webcam, a system for replacing the traditional devices for computer interaction such as a mouse or a keyboard can be implemented.

After an accurate bibliographic revision, some systems are being already commercialized and other more sophisticated solutions are in research phases. These approaches differ in the selection of the facial feature to track. Moreover, they don't include gesture recognition in their solutions. Toyama's work

SHORT COMMUNICATION proceedings
ISBN 80-86943-05-4
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.
Copyright UNION Agency – Science Press

[Toy98a] uses the whole face, Crea Ratón Facial [Cre05a], Tash CameraMouse™ [Bet02a][Tas05a] and Nouse™ [Gor02a], use the nose, and EyeTech Digital System Quick Glance [Eye05a] uses the gaze and infrared illumination.

The system presented here works with a standard webcam and non attached devices on the user. It is based on face features tracking and face gesture recognition. The system aims to act as a human-computer interface replacing the mouse's actions and with a complementary graphical keyboard, the user can interact with the computer completely hands-free. By means of the nose tracking the mouse's position is carried out and the mouse's events can be achieved by gesture recognition.

Important requirements to take into account are that the system has to work in real-time, under unconstrained backgrounds and with conventional light conditions. Furthermore, the quality of the images is poor due to the use of webcams.

The paper is organized as follows. Section 2 describes the system's architecture, the face detection, the face features selection, the features tracking and smoothing and the gestures recognition. Section 3 presents the performance evaluation and the last section concludes this paper.

2. FACE-BASED PERCEPTUAL USER INTERFACE SYSTEM

System Architecture

The application presented in this paper uses the nose region as facial feature for tracking. The nose region is the selected feature because it is visible for everyone in all face positions facing the computer screen, fact that does not occur always with other features that can be occluded by beards, moustaches or glasses.

In figure 2.1, a graph of the system process can be seen.

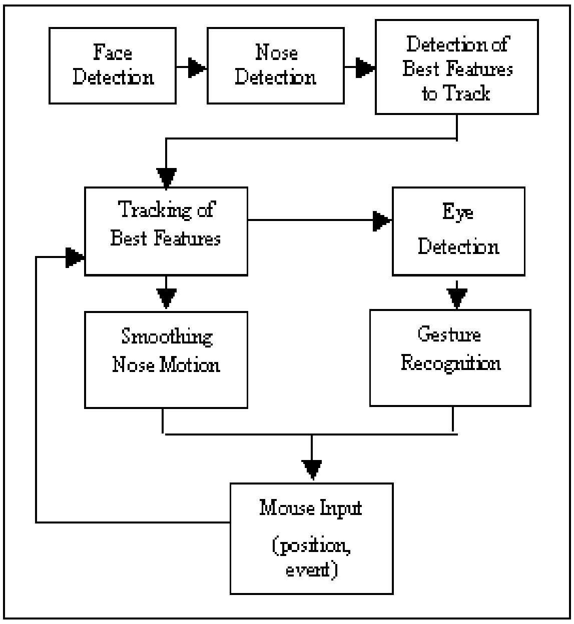

Face Detection
Nose Detection
Detection of Best Features to Track
Tracking of Best Features
Eye Detection
Smoothing Nose Motion
Gesture Recognition
Mouse Input (position, event)

Figure 2.1 System Process

Face Detection

The system automatically detects the face by means of Viola and Jones algorithm [Jou04a], which does not require any previous calibration process. This approach slides a window across the image and applies a binary classifier that discriminates between a face or the background. This classifier is trained using a boosting machine learning meta-algorithm [Vio01a].

When a face is detected, for it to be accepted as a face and for continuing with the image process, the user must keep it steady for a number of frames. When the user executes the system, he has to place himself correctly and steady in front of the screen and the webcam. In figure 2.2, correct and incorrect face detection examples are shown.

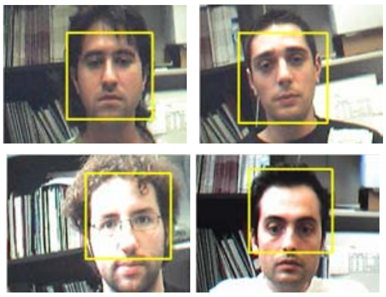


Fig. 2.2 Face detection examples: upper row correct; down row incorrect.

The face bounding box is defined as $F = \{\vec{X}_F, \vec{S}_F\}$, where $\vec{X}_F$ is the rectangle's origin point and $\vec{S}_F$ is the rectangle's size.

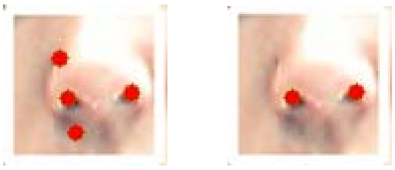
Best Face Features Detection & Tracking

Once the face is detected, the system looks for the nose section. Based on anthropometrical face measurements, the nose is found in approximately the second third of the face. The nose rectangle is defined as $N = \{\vec{X}_N, \vec{S}_N\}$ where the nose rectangle origin is $\vec{X}_N = (\vec{X}_F + \vec{S}_F/3)$ and the nose rectangle size is $\vec{S}_N = (\vec{S}_F/3)$.

Over the nose rectangle, the best features to track are selected controlling those points for which the derivative energy perpendicular to the prominent direction is above a threshold. This typically selects corners in images [Shi94a].

When applying the algorithm, due to the illumination, some points can appear and not be in the nose corners. The real position point of the mouse to track will be a mean of all the features found, due to this, the selected features to track should be in both sides of the nose and symmetrical, because this point should be as centred as possible.

With the points found with the algorithm, the final list of points $\vec{P} = (\vec{p}_1, \vec{p}_2 ... \vec{p}_n)$ to track, where n is the number of features and $\vec{p}_i = (px_i, py_i)$, is created considering symmetry constraints. In figure 2.3, the features found applying directly the Shi & Tomasi algorithm and the final list of features that will be considered due to their symmetry respect to the vertical axis are shown .



(a) **(b)**

Fig. 2.3 Best Features to Track (a) Shi and Tomasi best features to track (b) Enhancement of the features for our purposes

For tracking the selected feature points, the spatial intensity gradient information of the images is used for finding the best image registration [Jou04b]. The Lucas-Kanade algorithm is robust for handling rotation, scaling and shearing, so the user can move in a more flexible way. However, due to illumination changes or a fast movement, the length between a particular feature to track and the mean of the features can be greater than the length of the nose rectangle diagonal. If this occurs, this feature is not tracked anymore, as only the points beneath the nose region are in the region of interest.

Smoothing the Nose Motion

When the user wants to move the mouse position to a particular place, there is a tendency in the direction of the movement. For avoiding jitter and for smoothing the movement, it is applied a linear regression method to a particular number of tracked nose positions through consecutive frames. The computed nose points of several consecutive frames, s, are adjusted to a line, and therefore the mouse's motion can be carried out over that line direction.

Additionally, for avoiding discontinuity problems between the estimations of the nose positions at each time step, previous estimated points are considered for computing the current frame nose position.

When the new smoothed point for the frame t, n_t is calculated, the system calculates the horizontal distance and the vertical distance in pixels between n_t and the previous nose point estimation.

$$Dx(n_t) = (nx_t - nx_{t-1}),$$
$$Dy(n_t) = (ny_t - ny_{t-1}),$$

Finally, these distances will be used in the computation of the final screen coordinates S for the new point n_t

$$S_x = V_x + Dx(n_t) \cdot Kx$$
$$S_y = V_y + Dy(n_t) \cdot Ky$$

where $\vec{V} = (V_x, V_y)$ is the previous mouse position point in the computer screen and Kx and Ky are predefined constants

Finally, the computed mouse screen coordinates $\vec{S}$ are sent as real mouse inputs for placing the cursor in the desired region.

Gesture Recognition

The PUI can be enhanced including gesture recognition for replacing the different clicks of the mouse, for example, by means of eye blinking detection. This is a very difficult task due to the poor quality of the images and no use of infrared lights.

The eye blink detection is based on finding the iris. That is, if the iris is detected in the image the eye will be considered as open, if not, the eye will be considered closed. For achieving this aim, a region of interest, ROI, is calculated for including the eyes' region. This eyes' ROI is proportional to the face dimensions and it avoids including hair or the face frontiers. Then the eyes' ROI is segmented by a threshold for finding the dark zones. After that, the Sobel operator is used for detecting the vertical edges of the eyes' ROI (searching only for vertical contours eliminates the eyebrows if they were included in the image). The two sides of the eye are the ones with the two longest vertical edges. The left eye or right eye is distinguished controlling the centre of the image. In figure 2.4 the eye blinking detection process is shown.

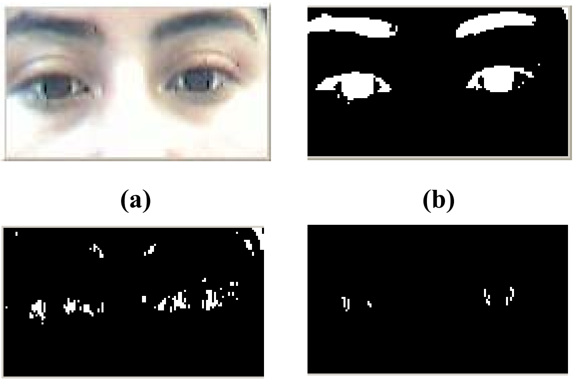

(a) (b) (c) (d)

Fig 2.4 (a) Original image, (b) Thresholding results, (c) Sobel contours, (d)Eye vertical edges

If the two vertical edges of an eye don't appear in the image for a number of frames (for gesture consistency) the system will assume that the eye has been closed and will send the mouse event associated.

Some examples of the blink recognition are shown in figure 2.5.

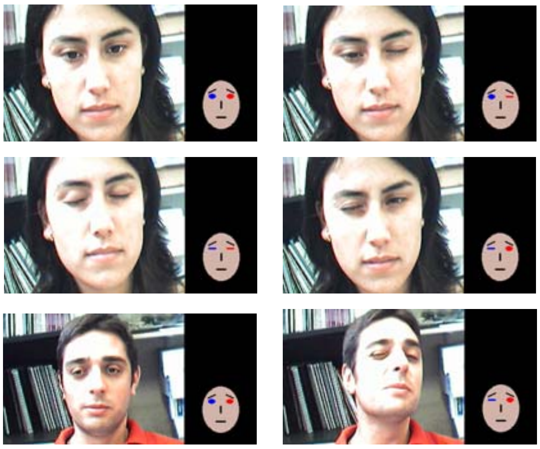


Figure 2.5 Examples of eye blinking detection

3. SYSTEM'S PERFORMANCE EVALUATION

In this section we show the accuracy and robustness of the nose tracking system. The application is implemented in Visual C++ using OpenCV libraries. The images were captured with two webcams: a Genius VideoCAM Express USB Internet Video Camera and with a Logitech QuickCam Messenger. The cameras are not assumed to be calibrated and they provide 320x240 images at a rate of 25 frames per second. The computer configuration was a Pentium IV, 3.2 GHz, 1GB RAM. Although the system has been tested on machines with less resources and it worked fine.

The user placement is very important too. The user was seating in a comfortable position without stretching his neck or forcing a strange pose. The webcam was placed on the computer screen at a forehead height. See figure 3.1

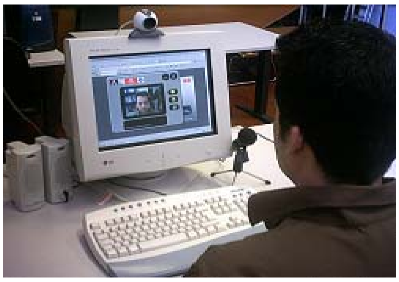


Figure 3.1 Correct user placement

In order to prove the robustness, the user was asked to rotate and move his head always facing the computer screen and therefore the webcam too. The results were that the user could move in a quite wide range without loosing the features to track. In figure 3.2 images of the range of motion can be seen. Although the system is quite robust, fast movements or illumination changes can cause the loss of the good features to track, in this case, the system is re-initialized for detecting the face and the best features to track.

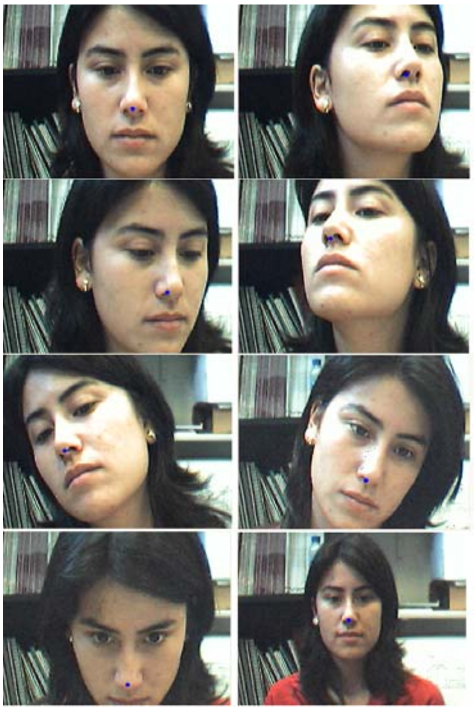


Figure 3.2 Face Motion Range

Two datasets were implemented for evaluating the accuracy of the application and users with different skills tested the system. The dataset A, a 4 x 4 point grid was presented in the computer screen. Each point has a radius of 15 pixels. Fig. 3.3 illustrates the pattern. This test was performed on a set of 13 persons without any training. The user had to try clicking on every point and distance data between the mouse's position click and the nearest point in the grid was stored.

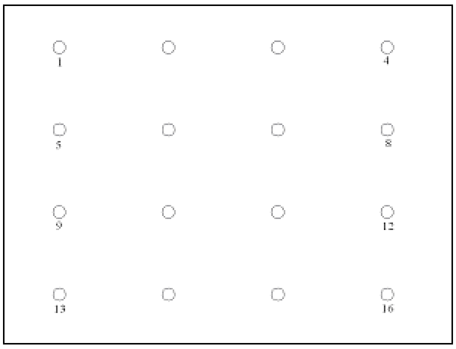


Figure 3.3 Dataset A: a point grid of 4x4, where the circle radius is 15 pixels

In figure 3.4 it is shown the percentage of error caused by clicks on incorrect positions for the first set of users without training. The graph shown in figure 3.5 represents the mean distance in pixels of the errors originated trying to click on a point for the users without experience.

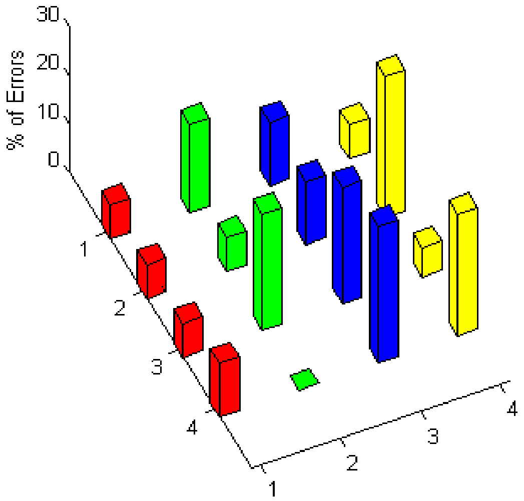


Figure 3.4 % of errors of Test 1: non-trained users

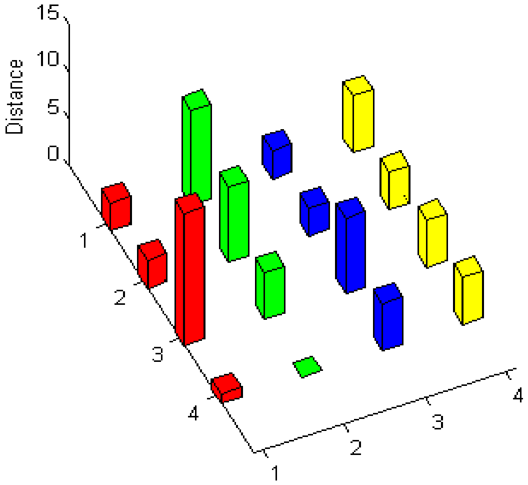


Figure 3.5 Distance errors (in pixels) of Test 1: non-trained users

In figures 3.4 and 3.5 it is shown that a person without previous training can place the screen cursor at the desired position in the main number of cases. Besides, the distance error is related with the circle position, that is, it augments when the user tries to click near the screen boundaries.

Next, the following experiments will show how after a short training period, the user improves fast his skills. Therefore the results improve. To show this fact a dataset B, a 5 x 5 point grid was presented in the computer screen. Each point has a radius of 15 pixels. Figure 3.6 illustrates the pattern.

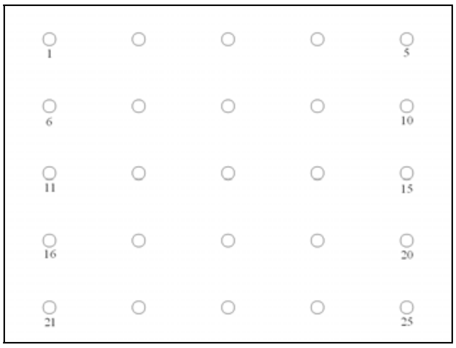


Figure 3.6 Dataset B. A point grid of 5x5. The circle radius is 15 pixels

This second test was performed on a set of 9 persons that already had used the system several times. The graphs shown in figures 3.7 and 3.8 illustrate how the number and distance of errors decrease dramatically (users only failed once). Besides, in this case (with trained users), there is not a relation between the screen positions and the accuracy of the system. However, in order to state this assumption this test should be completed with more persons.

Finally, to point out, the continuous use of this system produced some neck fatigue over several users. Therefore, some errors clicking the point grid could be caused due to this reason.

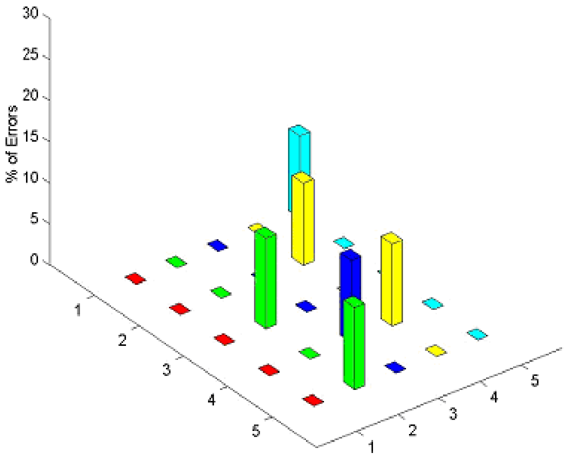


Figure 3.7 Errors of Test 2: trained users.

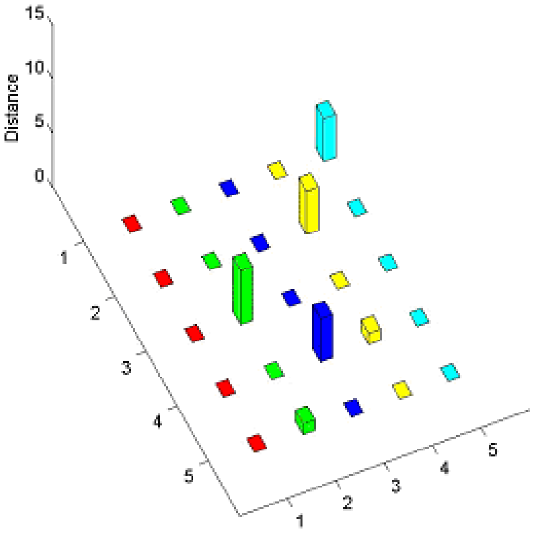


Figure 3.8: Distance errors (in pixels) of Test 2: trained users.

4. CONCLUSIONS AND FUTURE WORK

In this paper a real-time vision-based perceptual user interface has been presented for interacting with the computer. A system based on computer vision algorithms has been proposed. To build this system algorithms for face detection, best feature selection and tracking and face gesture recognition have been used.

The system's performance evaluation results have shown that the system can replace a traditional mouse with a low-cost interface such as a webcam with efficiency and robustness. Besides, the experiments have confirmed that continuous training of the users results in higher skills and, thus, better performances and accuracy for controlling the mouse position.

The system's disadvantages are that the environment has to be controlled regarding to illumination conditions and the user must be capable of moving the neck in quite a wide range.

As future work a more exhaustive evaluation of this system remains. Besides, it is planned to extend the gesture recognition to other face gestures for a more rich interaction with the computer. Therefore, it will be possible to obtain a MPEG4 compliant face description to build and control a user avatar for enhanced communication between users.

5. ACKNOWLEDGMENTS

This work has been subsidized by the national project TIN2004-07926 from the MCYT Spanish Government and TAGrv S.L., Fundación Vodafone, Asociación de Integración de Discapacitados en Red. Javier Varona acknowledges the support of a Ramon y Cajal grant from the Spanish MEC.

6. REFERENCES

[Bet02a] Betke, M., Gips J., Fleming, P. 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, nº1, 2002

[Cre05a] http://www.crea-si.com/esp/rfacial.html

[Eye05a] http://www.eyetechds.com/qglance2.htm

[Gor02a]Gorodnichy, D.O., Malik, S. , Roth, G. Nouse 'Use Your Nose as a Mouse' – a New Technology for Hands-free Games and Interfaces.

[Jou04a]Viola, P., and Jones, M. Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), pp.137-154, 2004

[Jou04b] Baker, S. and Matthews, I. Lucas-Kanade 20 Years On: A Unifying Framework. International Journal of Computer Vision 56(3), pp.221-225,2004
[Lan02a]Landay, J.A., Hong, J.I., Klemmer, S.R.,Lin, J., Newman, M. W. Informal PUIs: No Recognition Required. AAAI 2002 Spring Symposium: Sketch Understanding Workshop. pp. 86–90. 2002

[Shi94a] Shi, J., and Tomasi, C. Good Features to Track. IEEE Conference on Computer Vision and Pattern Recognition. Pp.593-600, 1994

[Tas05a]http://www.tashinc.com/catalog/ca_camera mouse.html

[Toy98a]Toyama, K. Look, Ma – No Hands!"Hands-Free Cursor Control with Real-Time 3D Face Tracking. Proc. Workshop on Perceptual User Interfaces. pp. 49–54, 1998.

[Vio01a] Viola, P. and Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pp. 511–518, 2001.