Available online at www.sciencedirect.com

ScienceDirect

Journal of Network and
Computer Applications 31 (2008) 357–374

Journal of NETWORK and COMPUTER APPLICATIONS

www.elsevier.com/locate/jnca

Hands-free vision-based interface for computer accessibility

Javier Varona*, Cristina Manresa-Yee, Francisco J. Perales

Departament de Matemàtiques i Informàtica, Universitat de les Illes Balears (UIB), Ed. Anselm Turmeda, Crta. Valldemossa km. 7.5, 07122 Palma, Spain

Received 13 August 2007; received in revised form 5 March 2008; accepted 14 March 2008

Abstract

Physically disabled and mentally challenged people are an important part of our society that has not yet received the same opportunities as others in their inclusion in the Information Society. Therefore, it is necessary to develop easily accessible systems for computers to achieve their inclusion within the new technologies. This paper presents a project whose objective is to draw disabled people nearer to new technologies. It presents a vision-based user interface designed to achieve computer accessibility for disabled users with motor impairments. The interface automatically finds the user's face and tracks it through time to recognize gestures within the face region in real time. Subsequently, a new information fusion procedure is proposed to acquire data from computer vision algorithms and its results are used to carry out a robust recognition process. Finally, we show how the system is used to replace a conventional mouse device for computer interaction and as a communication system for non-verbal children.

Keywords: Human–computer interaction; Multimodal interfaces; Accessibility

1. Introduction

Information and Communication Technologies (ICT) are present in many of our daily activities. Even if information systems are regularly used in education, work, leisure or

*Corresponding author. Tel.: +34 971172005; fax: +34 971173003.

E-mail addresses: xavi.varona@uib.es (J. Varona), cristina.manresa@uib.es (C. Manresa-Yee), paco.perales@uib.es (F.J. Perales).

1084-8045/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2008.03.003

domestic purposes by the majority of modern societies' citizens, still some sectors such as disabled people are at a disadvantage. In recent years, many research activities have focused on designs that aim to produce universally accessible systems that can be used by everyone, regardless of their physical or cognitive skills (Obrenovic et al., 2007). Challenges that are presented are the development of new technologies and systems accessible to everyone and offering assistive technology. Multimodal interaction is a characteristic of everyday human communication, with which we speak, fix our gaze, gesture and move in an effective flow of communication. Enriching human–computer interaction (HCI) with these elements of natural human behaviour is the primary task of multimodal user interfaces. For these reasons, the creation of non-invasive and more natural human–computer interfaces based on speech recognition or computer vision techniques can offer an easier and friendlier interaction with computers to disabled people, rather than using a standard mouse or keyboard, which in some cases may not be possible.

Of all the communication channels through which interface information can travel, vision provides a lot of information that can be used for the recognition of human actions and gestures, which can be analysed and applied to interaction purposes (Turk and Kolsch, 2004). Specifically, when sitting in front of a computer heads and faces can be assumed to be visible to a webcam, a very common input device nowadays. Therefore, it is natural to think of an interface based on head movements, face gestures or facial expressions. Of course, difficulties can arise from in-plane (tilted head, upside down) and out-of-plane (frontal view, side view) rotations of the head, facial hair, glasses, lighting variations and cluttered background. With the goal of solving these difficulties, this paper presents a human–computer interface based on computer vision technology that can automatically detect and track a head and facial features in real time from a standard USB webcam image stream.

Let us first define the main concepts of computer vision-based interfaces. *Detection* determines the image region where an object is located, for example, a face. *Tracking* means to locate the object and report its changes in position over time. It can be considered as a repeated frame-by-frame object detection, which usually implies more than discontinuous processing. In order to improve the tracking by adding a temporal continuity and a prediction for limiting the space of possible solutions, filters for modelling the object's temporal progression such as linear regression or Kalman filters can be used. By using computer vision technology, the interface feedback can be precise and robust in real time. Besides, current technology does not need specific interface requirements, that is, the user's work environment conditions could be normal (office, house or indoor environments), that is, with no specific lighting or static background. Finally, by using a standard webcam to provide the image stream to process, the interface allows the achievement of a low-cost system.

Several research works have tried to achieve similar functionality based on computer vision techniques. Earlier research on automatic head tracking analysed characteristic facial cues such as their colour distributions, head geometry or motion (Bradski, 1998; Toyama, 1998). Alternatively, there are works based on tracking one or more facial features. The location and motion of features is used to estimate the head position. For example, the Camera Mouse of Betke et al. (2002) locates visible features on the user's face: eye, nose and mouth, and then tracks face movement by searching for similar looking regions in subsequent frames. The searching process is based on the appearance of facial regions. An object's appearance describes its colour and brightness properties on every point of its surface, taking into account the texture, surface structure, lighting and view direction. These attributes are view-dependent; therefore, it will only make sense to talk about them from a given point. However, in their system, the user must manually select a feature to track. The most recent Camera Mouse downloads have implemented a relatively robust automatic face finding process. The importance of this work is its contribution of using this technology in order to assist users with disabilities. Another example of head location estimation by means of facial feature tracking can be found in work of Gorodnichy et al. (2004). In this case, the goal is to track a nose, whose main characteristic is that it extends in front of the face and ends with a somewhat universal rounded shape; therefore it is relatively easy to identify and detect as the user moves about. Therefore, the head position is estimated using the offset of the nose from a head centre point. However, the initialization process and the steps to be taken by the user before starting the tracking process are not clearly explained in the above-mentioned work. Their system is composed of two webcams in order to achieve 3D estimations and constrain the image facial feature search. It is possible to consider that 3D estimations are not necessary for a robust feature tracking. In this sense, recently, Morris and Chauhan (2006) presented a system for cursor control. This research work makes an excellent analysis of the difficulties caused by using webcams taking into consideration problems such as low image resolution and bad image quality. However, in order to function in any environment, the system needs a previous calibration stage to establish several process parameters. The usability of vision-based cursor control interfaces is analysed in the work of Kjeldsen (2006). As a conclusion, he presents an accurate cursor position control that takes the dynamics of human motion into account to give smoother and more responsive motion estimation. Nevertheless, it also requires a phase of predefined user's movements before the head tracking process starts. This calibration phase requires a level of head control that in certain kinds of disabilities is not possible.

The work presented here differs from the previous vision-based interfaces in several ways. First of all, it was designed specifically for disabled people with motor impairments in hands or arms. For this reason, the interface is completely automatic for the user, and it does not require any calibration phase with predefined user's actions or movements. In addition, it also includes facial gesture recognition for achieving basic event recognition. We consider as facial gestures the atomic facial feature motions such as eye blinking or winking (Grauman et al., 2001; Morris et al., 2002; Grauman et al., 2003). Other systems contemplate head gesture recognition, which imply overall head motions (Xiao et al., 2003), or facial expression recognition, the latter combines different facial feature changes to express an emotion (Fasel and Luettin, 2003; Pantic and Rothkrantz, 2003). In addition, a computer application for computer accessibility, has been developed using the hands-free vision-based interface and it can be accessed freely on the Web by users around the world (Manresa et al., 2006a). Thus, it is continually tested by disabled and non-disabled people who have the possibility of giving feedback on its performance. Finally, we have developed another application, *BlissSpeaker,* whose objective is improving the social integration of children with physical disabilities and speech problems (Manresa et al., 2006b).

The paper is organized as follows. In Section 2 we describe the hands-free vision-based interface in general terms. Section 3 explains the system's initialization by means of an automatic learning process of the user's facial features. Then, in Section 4, we explain how to estimate facial features' positions through time by means of the tracking process. The facial gesture recognition process for detecting eye winks is detailed in Section 5. In

Section 6, the presented algorithms are used for achieving a hands-free interface for computer accessibility replacing the mouse device. In Section 7, the interface performance evaluation results are described. And finally, in the last Section 8, two interface applications are presented for interactive expositions and communication for non-verbal children.

2. The hands-free vision-based interface architecture

In this section, the vision-based interface architecture is presented. This interface is conceived particularly for people with physical disabilities on the upper-body limbs and it allows them to interact with the computer during everyday life. In order to achieve this function, the system's feedback is in real-time and is precise and robust. A standard USB webcam is used for image acquisition, providing a low-cost system. The user's work environment conditions are normal (office, house or indoor environments) with no special lighting or static background. In addition, the use of attachments or paintings on the user's head is not necessary. The system's only requirement is that, at the beginning, the users position their head facing the screen avoiding any type of orientation: by positioning their head in pan, tilt or roll angles they may cause the failure of the automatic face and facial features initial detection. For example, in certain cases of cerebral palsy, users need assistance to position their head correctly. Nevertheless, once the system is initialized it works correctly for head orientations (providing that facial features are visible).

To achieve an easy and user-friendly perceptual user interface, the system is composed of two main modules: Initialization and Processing (see Fig. 1). The Initialization module is responsible of extracting the user's facial features models. This process locates the user's face, learns their skin colour and detects the initial facial feature locations and their image properties such as appearance and colour. Moreover, this process is completely automatic,

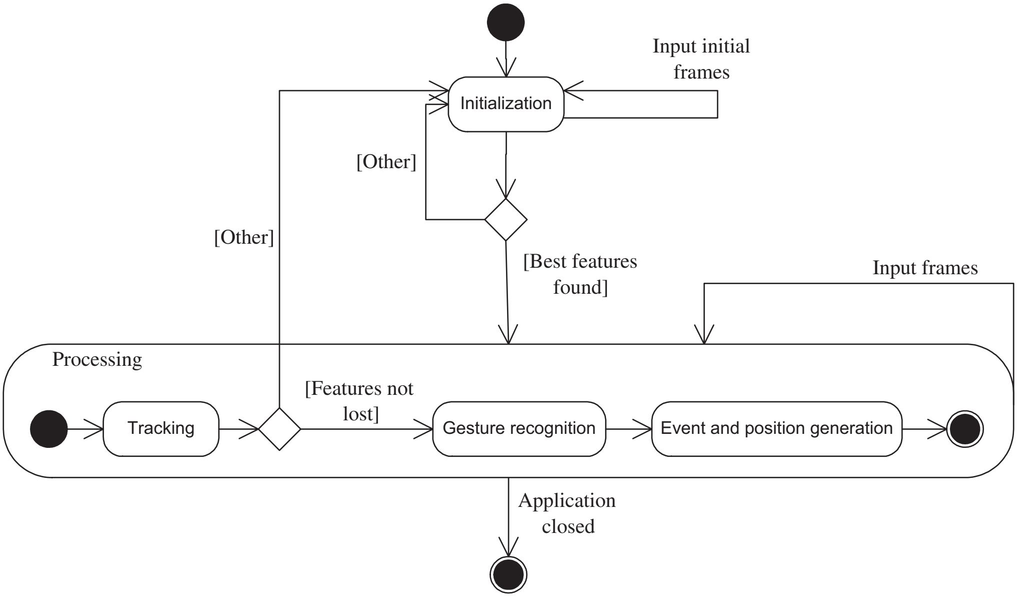

Fig. 1. The vision-based interface is divided in two main modules: Initialization and Processing.

and it can be considered as a learning process of the user's facial features. The chosen facial features are the nose for head tracking and the eyes for gesture recognition. We decided to use the nose as the tracking feature because it is almost always visible in all positions of the head when facing the screen and it is not occluded by beards, moustaches or glasses. Regarding the gesture recognition module, the controlled chosen gestures are right and left eye winks.

The selected facial features' positions are robustly estimated through time by two tasks: nose tracking based on image features and eye tracking by means of colour distributions. It is important to point out that the system is able to react when the image features get lost, detecting when it occurs and restarting the system calling the Initialization module.

3. Learning the user's facial features

It is very important that the interface be more natural; consequently, the system should not require any calibration process where the user is involved. To accomplish this necessity, the system automatically detects the user's face by means of the algorithm of Viola and Jones (2004). The user should just stay still for a few frames so that the process can be initialized. The face detection is considered robust if the face is detected in approximately the same image position for a few frames. Specifically, best results have been achieved using 15 frames. In Fig. 2(a) examples of face detection for different users are shown. It is now possible to define the image region for searching the user's facial features. Based on anthropometrical measurements, the face region can be divided in three sections: eyes and eyebrows, nose, and mouth region.

The image points that can easily be tracked on the nose region are searched for in the image; these points are characterized by a high image derivative energy which is perpendicular to the prominent direction (Shi and Tomasi, 1994). This condition theoretically selects the corners or the nostrils in the nose region. However, the ambient lighting, which can produce unwanted shadows, can cause the selection of useless and unstable features; this fact is visible in Fig. 2(b). Ideally, the desired selected points should be found on both sides of the nose and characterized by certain symmetry conditions. Therefore, an enhancement and a re-selection of the initially found features must be carried out taking into account these constraints. Fig. 2(c) shows the final reselected features. This process selects the tracking points for the nose facial feature and it will contribute to the process' robustness. Fig. 2(d) illustrates the final point considered: the mean of the features, which due to the re-selection of points, should be centred on the nose.

The nose-detection process has been evaluated using the BioID face database. We have chosen this database because the image resolution and acquisition conditions are similar to the ones belonging to our application. The BioID is a head-and-shoulder image face database that stresses "real-world" conditions featuring a large variety of illumination and face sizes with complex backgrounds in normal conditions with no restrictions. The database consists of 1521 frontal view images of 23 different test persons with a resolution of 384 × 286 pixel (Jesorsky et al., 2001). Tests conducted with these images have shown that 95.79% of faces are successfully detected, and among the detected faces about 96.08% of nose features are detected with enough precision. In order to measure the precision of detection, we take advantage of the fact that the database images have manually annotated several facial feature points (http://www.bioid.com/downloads/facedb/index.php). Specifically, we use the "tip of the nose" mark for comparison with the results of our

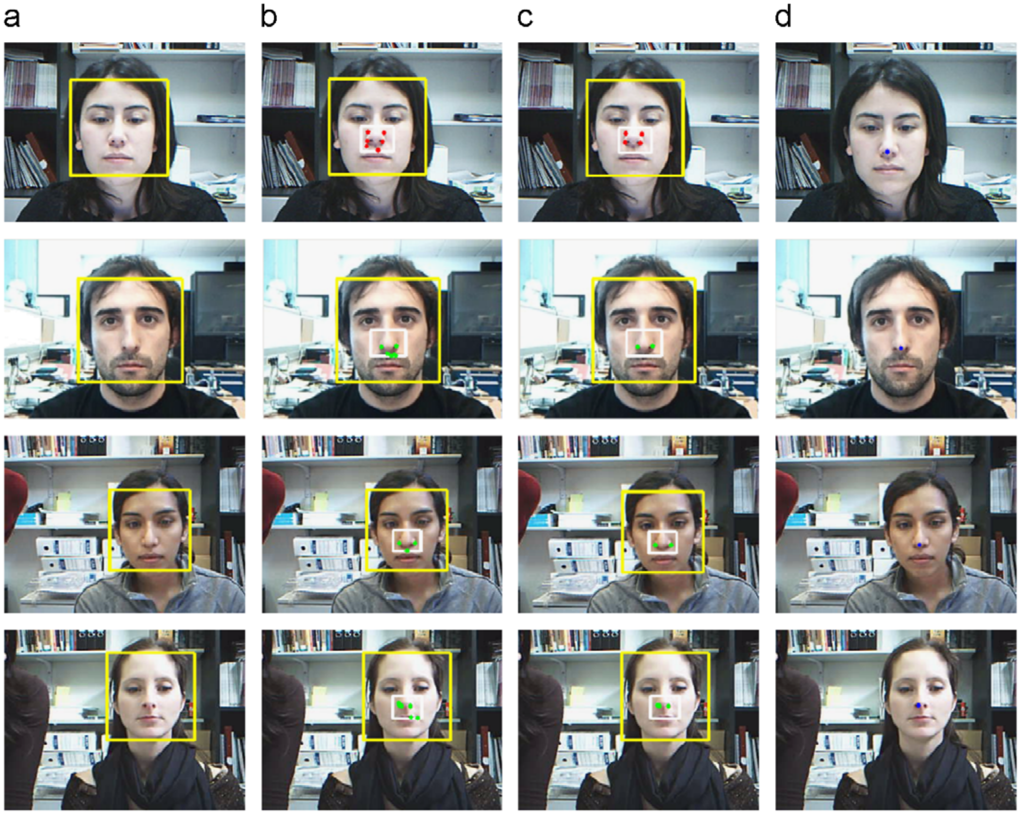

Fig. 2. (a) Automatic face detection, (b) initial set of features, (c) best feature selection using symmetrical constraints, (d) mean of the selected features: tracked nose point.

Table 1
Precision results of the nose-detection process (in pixels)

Displacement	Mean	Standard dev.	Maximum	Minimum
Total	6.03	4.66	29.95	0.03
Horizontal	2.34	2.05	15.79	0.00
Vertical	4.98	4.86	29.43	0.00

nose-detection algorithm. The precision was measured computing distance between the mean of the features and the “tip of the nose” mark. The computed distance is the root-squared difference between the two points. In addition, we also have computed the differences in X and Y. Precision results are summarized in Table 1. Note that errors in face detection are due to incorrect placements of the head in the image, which result in incomplete visibility in the image (see Fig. 3(a)). Besides, errors in the nose detection are mainly due to lighting conditions that lead to different brightness on either side of the face. This causes the feature selection algorithm to fail to find symmetrical features (see Fig. 3(b)). In relation to the precision of nose detection in Table 1, we can see that the error

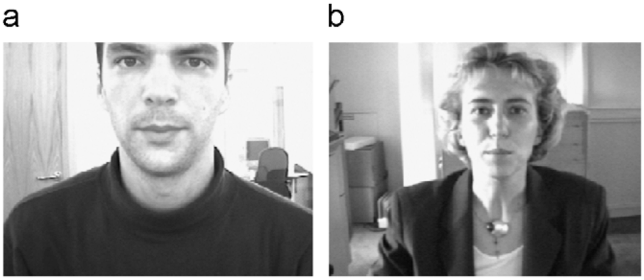

Fig. 3. BioID database samples showing the main causes of detection errors for face (a) and nose (b).

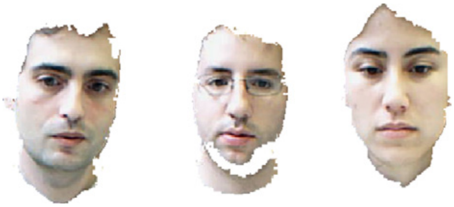

Fig. 4. Skin masks for different users obtained by the learning of each user's skin colour model.

in *Y* is greater than the error in *X*. This is caused by the fact that symmetrical features are detected in most cases near the nostrils (see Fig. 2); therefore the mean of found features is properly located horizontally, but with a little displacement in the *Y*-axis towards the bottom of the image. Nevertheless, taking the acquisition conditions of the database images and the computed displacements into account, we can conclude that the precision of the presented nose-detection process is acceptable for our purposes.

The skin colour feature will help the tracking and gesture recognition by constraining the processing to the pixels classified inside a "skin mask". In order to learn the user's skin colour, the pixels inside the previously detected facial region are used as colour samples for building the learning set. Being an automatic process, the sample region must be easy to find. Having divided the face into three regions, the region with more skin colour and with fewer obstacles (e.g. the eyes or mouth) is the nose region; therefore, the samples are taken from this image region. Examples of nose regions are marked as white rectangles in images of Fig. 2. Actually, before taking these samples, the darker points are excluded to avoid colour samples from regions such as nostrils and moustaches. A Gaussian model in 3D RGB is chosen to represent the skin colour probability density function due to its good results in practical applications (Alexander and Buxton, 2001). The Gaussian model parameters, mean and covariance matrix, are computed from the sample set using standard maximum likelihood methods (Bishop, 1995). Once the model is calculated, the probability of a new pixel being recognized as skin is computed. Finally, we apply a probability threshold and a hole-filling process in order to have a compact region for searching, the "skin mask" of the user's face, see Fig. 4.

The last step of the Initialization is to build the user's eye model. Using the region of the eyes and eyebrows found in the face detection phase both eyes can be located. First of all, the facial region is binarized to find the dark zones, and the algorithm selects the bounding

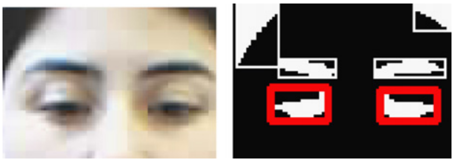

Fig. 5. Example of eyes' region detection: the image regions that are selected (in red) are symmetrical and are nearer to the nose region.

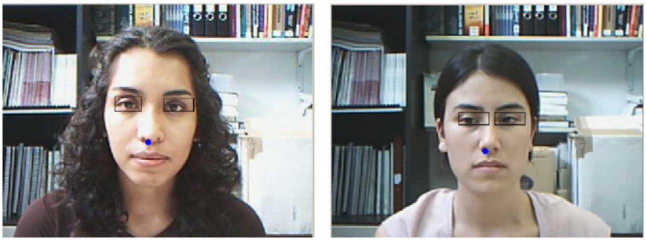

Fig. 6. Final eye regions for tracking (in black).

boxes of the pair of image regions that are symmetrical and are located nearer to the nose region. This process may fail in the case that the user wears eye glasses. The main reasons for failure are: the light reflected in the glasses caused by certain lighting conditions and the variety of eye glasses models that mislead eye location.

Next, in order to remove the eyebrows or the face borders, see Fig. 5, we apply that the eye colour differs from the rest of the facial features' colour (taking into account that the eye colour distribution is composed by sclera and iris colours). Besides, this characteristic colour distribution will be used for the eye tracking. Therefore, the interface can be used by people with clear (blue or green) or dark (black or brown) eyes. Eye models are obtained through histogramming techniques in the RGB colour space of the pixels belonging to the detected eye regions.

The model histograms are produced with the function b: $R^2 \rightarrow \{1 \ldots m\}$ where m is the number of bins. The function associates the pixel at location $\boldsymbol{x}_{\mathrm{i}}$ the index $b(\boldsymbol{x}_i)$ of its bin in the quantized feature space. In the experiments performed, the histograms are calculated in the RGB space using $16 \times 16 \times 16$ bins. The reliability of the colour distribution is increased by applying a weight function to each bin value depending of the distance between the pixel location and the eye centre. In Fig. 6, samples of the final eye regions for tracking are shown.

4. Facial features tracking

The facial feature tracking process consists of two tasks: eye and nose tracking. The eye tracking is based on the eye's characteristic colour. As was explained in the previous section, by weighting the eye model depending of the distance between the pixel location and the eye centre, the tracking reliability is increased. In addition, when the weighting function is an isotropic kernel, it is possible to use a gradient optimization function, such as the mean-shift algorithm, to search for the eye model in the new image. Practical details and a discussion about the complete algorithm can be found in Comaniciu et al. (2003). As it can be seen in the performance evaluation of the application for mouse replacing in Section 7, this algorithm performs well and in real time. It is important to comment that small positional errors can occur. However, it is not important because the eye tracking results are only used to define the image regions where the gesture recognition process is to be performed. Besides, to add robustness to this process only the pixels belonging to the user's ''skin mask'' are considered as search region.

The positional results for the interface are reported by the nose-tracking algorithm, where the selected image features of the nose region are used. In this case, the spatial intensity gradient information of the images is used to find the image registration (Baker and Matthews, 2004). By means of this algorithm, each selected image feature in the new frame is found. Then, the mean is computed from the features found and defined as the new nose position for that time instant. This algorithm is designed to handle rotation, scaling and shearing, so the user can move more freely. Lighting or fast movements can cause the loss or displacement of the tracking image features. Since only the image features beneath the nose region are in the region of interest, an image feature will be discarded when the distance between this feature and the nose position is greater than a predefined value. When all useless features are discarded, the Initialization process is called to start the process once again.

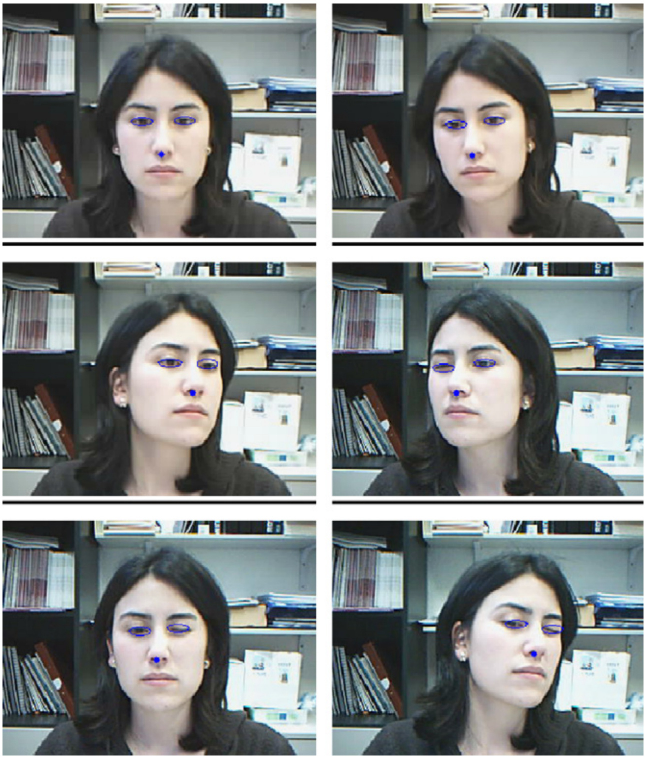

Fig. 7. Facial feature tracking results.

In theory, it would be possible to use Kalman filters to smooth the positions (Bar-Shalom et al., 2001). However, Kalman filters are not suitable in this case, because they do not achieve good results with erratic movements such as head motion (Fagiani et al., 2002). Therefore, our smoothing algorithm is based on the motion's tendency of the nose positions, i.e. the head motion tendency. A Linear Regression method is applied to a number of tracked nose positions through consecutive frames (Bishop, 1995). The computed nose points of n consecutive frames are adjusted to a line, and therefore, the nose motion can be carried out over that line direction. To avoid discontinuities the regression line adjusts to every new point that arrives. Several frames of the tracking sequences are shown in Fig. 7.

5. Facial gesture recognition

The gestures to take into consideration are eye winks. Most previous work used quality images and good image resolution in the eye zones. However, wink recognition with webcam quality images is difficult. Besides, this process depends on the user's head position. The wink detection process is based on a search for the iris contours. That is, if the iris contours are detected in the image, the eye will be considered as open, if not, the eye will be considered closed. It is important to point out that, this process is robust because it is only carried out in the tracked eye regions by the mean-shift procedure described before.

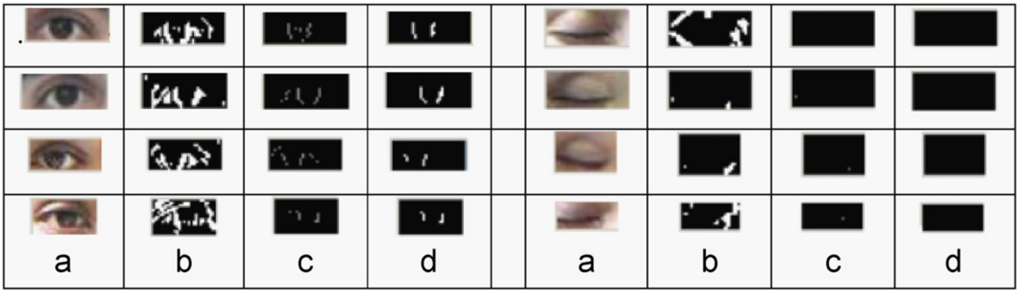

Fig. 8. Process for recognizing winks: (a) original image, (b) segmentation, (c) vertical edges, (d) iris contours.

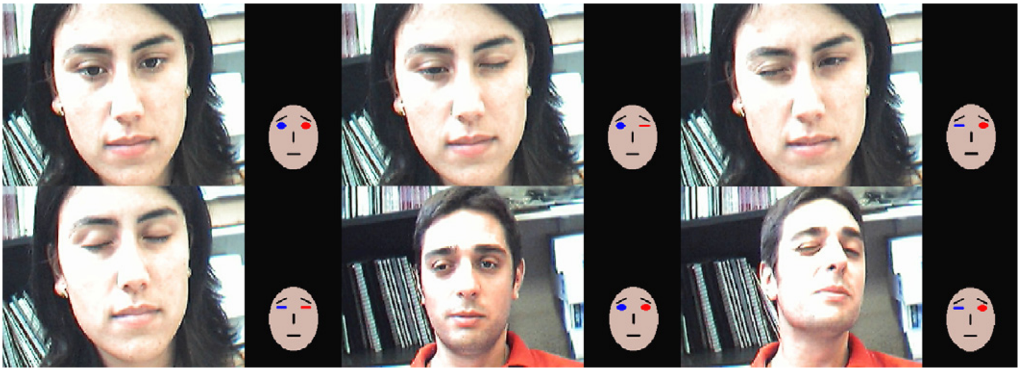

Fig. 9. Wink detection examples.

The process starts by detecting the vertical contours in the image. To avoid false positives in this process, the vertical contours are logically combined with a mask, which is generated by binarization of the current image. Finally, the two longest vertical edges of each eye region are maintained if they appear to be the eye candidates. If these two vertical edges, which correspond to the eye iris edges, do not appear after a predefined number of consecutive frames, it is assumed that the eye is closed. In Fig. 8 the image process for gesture recognition is described, and in Fig. 9, examples of the wink detection are shown.

6. Replacing the mouse for hands-free computer accessibility

All the techniques mentioned before have been applied in conjunction to build an application whose objective is to completely replace the traditional mouse device functionality used to interact with computers. Moreover, a benefit of this vision-based interface is its potential users. As the interaction is controlled by head motion and gesture recognition, the need for hands is not required. Therefore, physically disabled users with limitations in upper-limb movement (i.e. brain palsy, multiple sclerosis) can benefit from this hands-free computer accessibility. Other interface uses could be for entertainment and leisure, such as computer games or exploring immersive 3D graphic worlds offering new and more powerful interaction experience to any kind of audience.

From a technical point of view, the precision required is sufficient to move the screen cursor to the desired position by means of a user's head motions. Reproduction of the mouse motion can be done in two different forms: absolute and relative. In the absolute type, the nose position is mapped directly onto the screen, but this type would require a very accurate tracking, since a small tracking error in the image would be magnified on the screen. Therefore, relative head motion is used to control the mouse's motion, which is not as sensitive to the tracking accuracy, since the cursor is controlled by the relative motion of the nose facial feature in the image. When the user wants to move the mouse position to a particular position, there is a predictable tendency in the direction of its head movement.

The relative control yields smoother movements of the cursor, due to the non-magnification of the tracking error. So, if $\mathbf{n}_t = (x_t, y_t)$ is the new nose-tracked position in the image at time t, to compute the new mouse screen coordinates, $\mathbf{s}_t$, at time t, Eq. (1) is applied. Where α is a vector with two predefined constants that depend on the horizontal and vertical screen sizes and translates the image coordinates to screen coordinates. The computed screen coordinates are sent to the system as mouse events to place the cursor on the desired position:

$$\mathbf{s}_t = \mathbf{s}_{t-1} + \alpha(\mathbf{n}_t - \mathbf{n}_{t-1}). \tag{1}$$

In the application, a graphical event keyboard is provided with the main functionalities to select the mouse events: left click, double left click, right click and maintained left click for dragging operations. It also offers two special events to disable all the buttons for no event functionality and to disable the application. This is useful when for example the user wants to read an already open document and the screen cursor control is not necessary. When selecting the desired event, the user must stay still for a few seconds positioning the screen cursor over the desired event button (with the first version of the product) or with the wink of an eye they can pass onto the next button. Once selected, if the user fixates on any part of the screen for a few seconds, the event will be carried out. By means of the additional graphical keyboard and the previously described vision-based algorithms, the system will accomplish a complete hands-free interaction between the user and the computer.

7. Performance evaluation

The accuracy and robustness of the presented hands-free interface is shown in this section. The application is implemented in Visual C++ using the OpenCV libraries. The images for the performance evaluation 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 320 × 240 pixel images at a rate of 25 frames/s. The computer's configuration where the tests were carried out was a Pentium IV, 3.2 GHz, 1 GB RAM, although the system has been successfully tested on machines with fewer resources.

For the application to work correctly, the user placement is also very important. During tests the users were seated in a comfortable position without stretching their necks or forcing strange positions. The webcam was placed on the computer screen at approximately forehead height, see Fig. 10. In order to prove the robustness, the users were asked to rotate and move their heads always facing the computer screen and therefore, the webcam too. The results showed that the users could move in a wide range of ways without the application loosing the tracked features.

In Fig. 11, sample 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 tracked image features, in this case, the system re-initializes itself in order to detect the face and the best features to track.

To evaluate the application's performance, the new interface was tested by two sets of different users, one set had never experienced the application whereas the other set had been previously trained with the interface. A grid of 25 targets arranged in five rows was presented on the computer screen, and the users were asked to click on each target; each target had a radius of 15 pixels, see Fig. 12. Distance data between the screen cursor position and the nearest target on the grid was stored when the user clicked outside of the target to compute the distance of errors. Two experiments were implemented for evaluation of the interface accuracy for users with different skills. The first experiment was performed on a group of 13 people without any previous training. In this case, the results

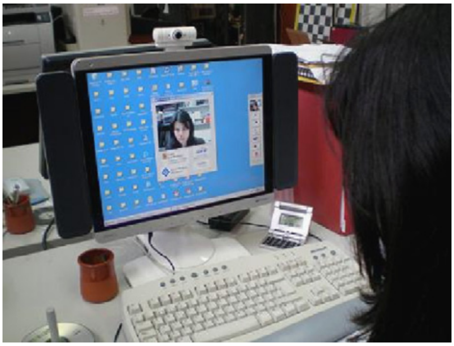

Fig. 10. Correct user placement.

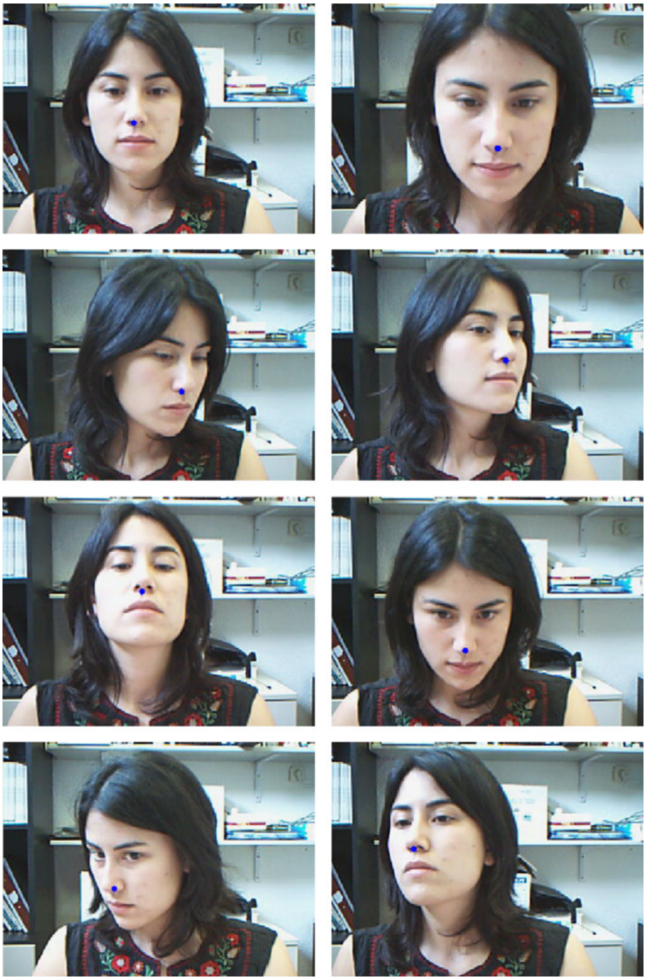

Fig. 11. Head motion range.

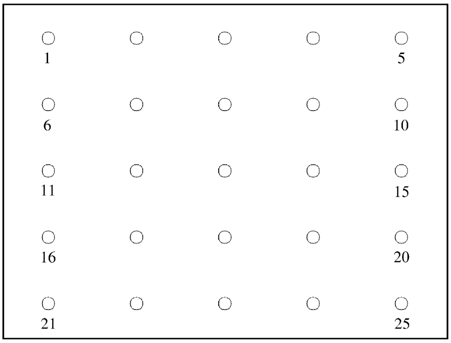

Fig. 12. The point grid pattern used for the interface's performance evaluation (the circle radius is 15 pixels).

Table 2
Results obtained in the performance evaluation experiments

Users group	Recognized clicks (%)	Distance error (mean)
New users	85.9	5 pixels
Trained users	97.3	2 pixels

showed that a user with no preparation can place the screen cursor correctly on the desired position in most cases. Also, the distance error is related to the circle position on the screen, that is, it increased when the user tries to click near the screen boundaries. Specifically, 80.4% of the errors occur when the user tries to click in one of the four targets placed farthest from the screen centre, i.e., on the targets near the screen corners. The second experiment was performed on a group of nine people that had already used the system several times before. In this case, as summarized in Table 2, the errors and their distance decrease dramatically. In this case, with trained users, there is no relation between the screen positions and the accuracy of the system.

Evaluation has demonstrated better performances and accuracy in controlling the screen cursor position due to training. Besides, a fact to take into account is that the prolonged use of the hands-free interface can produce some neck fatigue in certain users.

8. Applications based on the hands-free interface

8.1. Interactive expositions

Nowadays, expositions based on new ways of interaction need contact with the visitors that play an important role in the exhibition contents. Museums and expositions are open to all kind of visitors, therefore, these "sensing expositions" look forward to reaching the maximum number of people. This is the case of "Galicia dixital", an exposition in Santiago de Compostela (Martín et al., 2007). Visitors go through all the phases of the exposition sensing, touching and receiving multimodal feedback such as audio, video, haptics, interactive images or virtual reality.

In one phase there is a slider-puzzle with images of Galicia to be solved, see Fig. 13. There are four computers connected enabling four users to compete to complete the six puzzles included in the application.

Visitors use a touch-screen to interact with the slider-puzzle, but the characteristics of this application make it possible to interact by means of the hands-free interface in a very easy manner. Consequently, the application has been adapted to it and therefore, disabled people can also play this game and participate in a more active way in the exposition.

8.2. Non-verbal communication

By means of human–computer interaction, one ambitious objective is to achieve communication for people with speech disorders using new technologies. Nowadays, there are different augmentative communication systems for people with speech limitations, ranging from unaided communication such as sign languages, to computerized iconic languages with voice output systems such as *Minspeak*^{TM} (Albacete et al., 1998). We

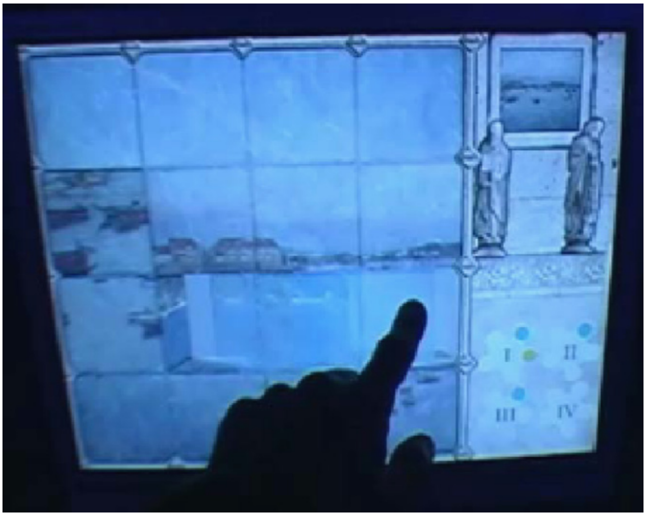

Fig. 13. Slider-Puzzle form Galicia Dixital exposition.

present *BlissSpeaker*, an application based on a symbolic graphical-visual system for non-verbal communication named Bliss (Blissymbolics, 2007).

The Bliss system can be used as an augmentative system or for completely replacing verbal communication. It is commonly used by people with cerebral palsy, but with the following learning aptitude requirements:

1. cognitive abilities;
2. good visual discrimination;
3. possibility of indicating the desired symbol;
4. good visual and auditory comprehension.

Some speech therapists use it in their sessions to help themselves with children with speech disorders and to help in the prevention of linguistic and cognitive delays in crucial stages of a child's life. The Blissymbolics language is currently composed of over 2000 graphic symbols that can be combined and re-combined to create new symbols. The number of symbols is adaptable to the capabilities and necessities of the user, for example, *BlissSpeaker* has 92 symbols that correspond to the first set of Bliss symbols for preschool children (Warrick, 1978). *BlissSpeaker* is an application that verbally reproduces statements built using Bliss symbols, which allows a more "natural" communication between a child using Bliss and a person that does not understand or use these symbols, for example, the children's relatives. The application can work with any language, as long as there is an available compatible SAPI (Speech Application Programming Interface). In Fig. 14, the system's process is shown.

The potential users of *BlissSpeaker* are children with speech disorders; therefore, its operation is to be very simple and intuitive. Moreover, audio, vision and traditional graphical user interfaces combined together configure a very appealing multimodal interface that can help attract and involve the user in its use. Furthermore, the use of the hands-free interface with *BlissSpeaker* will help to fulfil the third requirement of a Bliss

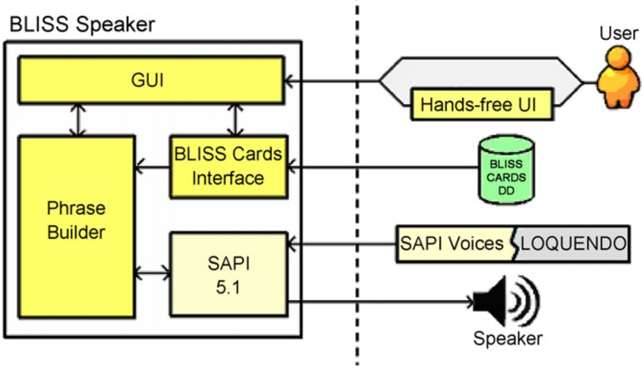

Fig. 14. *BlissSpeaker* diagram.

user, which is the possibility of indicating the desired symbol. It will offer children with upper-body physical disabilities and speech difficulties a way to communicate themselves through an easy interface and their teachers or relatives will understand them better due to the symbols' vocal reproduction. Furthermore, the use of the new interface can make learning of Bliss language more enjoyable and entertaining, and it also promotes the children's coordination, because the interface works with head motion.

This system was evaluated in a children's scientific fair. The system was tested by more than 60 disabled and non-disabled children from 6 to 14 years of age. A short explanation on how it works was given. They operated the application with surprising ease and even if they had never seen Bliss symbols before, they created statements that made sense and reproduced them for their class mates. Children enjoyed interacting with the computer through the functionalities that the face-based interface offered. Moreover, upper-body physical disabled children are grateful for the opportunity of accessing a computer.

9. Conclusion

The design of more natural and multimodal forms of interaction with computers or systems is an aim to achieve. Vision-based interfaces can offer appealing solutions to introduce non-intrusive systems with interaction by means of gestures. In order to build reliable and robust perceptual user interfaces based on computer vision, certain practical constraints must be taken in account: the application must be capable of working well in any environment and should make use of low-cost devices.

This work has proposed a new mixture of several computer vision techniques for facial features detection and tracking and face gesture recognition, some of them have been improved and enhanced to reach more stability and robustness. A hands-free interface able to replace the standard mouse motions and events has been developed using these techniques. In March of 2006 a preliminary version with mouse control was launched, but with the selection of the mouse's events using a wait-until-click over a graphical event keyboard, and it was awarded with the *Fundetec Award 2006*, in the category of "Best Project of a Non-Profit Organization oriented to Citizens". The final release of the mouse replacing application for Microsoft Windows, named SINA, will be available soon under a freeware license in the Web page http://dmi.uib.es/~ugiv/sina. This will allow us to have users around the world testing the application and we will be able to improve the results by analysing their reports.

The interface and both applications are currently being tested in two centres which work with disabled people; one centre works with users affected by cerebral palsy and the other with users affected by multiple sclerosis. In both cases, the main comments of the therapists tutoring the users are related to the interface's usability in order to improve the user's performance. The most important conclusion offered to us by the therapists is that users (mainly children) are really excited about being offered the possibility to access a computer. In addition, users with reduced head mobility have improved its control by means of using this interface, proving its potential as a rehabilitation tool. Of course, more improvements have to be made, including more gestures (equivalents of BLISS commands or other kind of language for disabled people), sounds (TTS and ASR) and adaptive learning capabilities for specific disabilities. Enhancements have been planned as future work, such as including a larger set of head and face gestures and combinations in order to speed up actions.

Acknowledgements

We thank the reviewers for their constructive feedback. This work is subsidized by the national projects TIN2007-67896 and TIN2007-67993 from the MCYT Spanish Government. We would like to thank the M.A.R. group of the University of Santiago de Compostela for the integration of our hands-free interface with their applications in Galixia Dixital. Besides, Dr. J. Varona acknowledges the support of a Ramon y Cajal Grant from the Spanish MEC.

References

Albacete PL, Chang SK, Polese G. Iconic language design for people with significant speech and multiple impairments in assistive technology and artificial intelligence. Lec Not Comput Sci 1998;1458:12–32.

Alexander DC, Buxton BF. Statistical modelling of colour data. Int J Comput Vis 2001;44:87–109.

Baker S, Matthews I. Lucas-Kanade 20 years on: a unifying framework. Int J Comput Vis 2004;56:221–55.

Bar-Shalom Y, Li XR, Kirubarajan T. Estimation with applications to tracking and navigation: theory, algorithms, and software. Wiley; 2001.

Betke M, Gips J, Fleming P. The Camera Mouse: visual tracking of body features to provide computer access for people with severe disabilities. Neural Syst Rehabilitation Eng 2002;10:1–10.

Bishop CM. Neural networks for pattern recognition. Oxford: Oxford University Press; 1995.

Blissymbolics 2007, 〈http://www.blissymbolics.org〉.

Bradski GR. Computer vision face tracking as a component of a perceptual user interface. In: Proceedings of the IEEE workshop on applications of computer vision (WACV). New Jersey, 1998. p. 214–9.

Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell 2003;25:564–77.

Fagiani C, Betke M, Gips J. Evaluation of tracking methods for human–computer interaction. In: Proceedings of the IEEE workshop on applications in computer vision (WACV). Orlando, 2002. p. 121–6.

Fasel B, Luettin J. Automatic facial expression analysis: a survey. Pat Rec 2003;36:259–75.

Gorodnichy DO, Malik S, Roth G. Nouse 'Use your nose as a mouse'—a new technology for hands-free games and interfaces. Image Vis Comput 2004;22:931–42.

Grauman K, Betke M, Gips J, Bradski GR. Communication via eye blinks detection and duration analysis in real time. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). Hawaii, 2001. p. 1010–7.

Grauman K, Betke M, Lombardi J, Gips J, Bradski GR. Communication via eye blinks and eyebrow raises: video-based human–computer interfaces. Univ Access Inf Soc 2003;2(4):359–73.

Jesorsky O, Kirchberg K, Frischholz R. Robust face detection using the Hausdorff distance. In: Bigun J, Smeraldi F, editors. Lecture notes in computer science, vol. 2091. Berlin: Springer; 2001. p. 90–5.

Kjeldsen R. Improvements in vision-based pointer control. In: Proceedings of ACM SIGACCESS conference on computers and accessibility. Portland: ACM Press; 2006. p. 189–96.

Manresa C, Varona J, Perales FJ. Towards hands-free interfaces based on real-time robust facial gesture recognition. In: Proceedings fourth conference on articulated motion and deformable objects (AMDO). Palma de Mallorca, 2006. p. 504–13.

Manresa C, Varona J, Ribot T, Perales FJ. Non-verbal communication by means of head tracking. In: Proceedings of Ibero-American symposium on computer graphics (SIAGC). Santiago, 2006. p. 72–5.

Martín R, Otero A, Gutiérrez E, Flores J. Exposiciones interactivas, caso de estudio: Galicia Dixital. In: Proceedings AIPO 2007.

Morris T, Zaidi F, Blenkhorn P. Blink detection for real-time eye tracking. J Network Comput Appl 2002;25: 129–43.

Morris T, Chauhan V. Facial feature tracking for cursor control. J Network Comput Appl 2006;29:62–80.

Obrenovic Z, Abascal J, Starcevic D. Universal accessibility as a multimodal design issue. Commun ACM 2007; 50(5):83–8.

Pantic M, Rothkrantz LJM. Toward an affect-sensitive multimodal human–computer interaction. Proc IEEE 2003;91(9):1370–90.

Shi J, Tomasi C. Good features to track. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). Seattle, 1994. p. 593–600.

Toyama K, Look, ma–No hands! Hands-free cursor control with real-time 3D face tracking. In: Proceedings of the workshop on perceptual user interfaces (PUI). San Francisco, 1998. p. 49–54.

Turk M, Kolsch M. Perceptual interfaces. In: Medioni G, Kang SB, editors. Emerging topics in computer vision. Prentice-Hall; 2004. p. 456–520.

Viola P, Jones M. Robust real-time face detection. Int J Comput Vis 2004;57:137–54.

Warrick A. Blissymbols for preschool children. Toronto: Blissymbolics Communication Institute; 1978.

Xiao J, Moriyama T, Kanade T, Cohn JF. Robust full-motion recovery of head by dynamic templates and re-registration techniques. Int J Ima Syst Technol 2003;13:85–94.