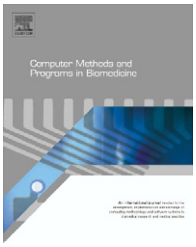
ELSEVIER

journal homepage: www.intl.elsevierhealth.com/journals/cmpb



Robust real time eye tracking for computer interface for disabled people

*Alberto De Santis, Daniela Iacoviello**

Dept. Informatica e Sistemistica "A. Ruberti", Sapienza University of Rome, Via Ariosto 25, 00185 Rome, Italy

ARTICLE INFO

Article history:
Received 18 July 2008
Received in revised form
23 March 2009
Accepted 24 March 2009

Keywords:
Human computer interface
Eye tracking
Image segmentation
Level set
Recursive estimation

ABSTRACT

Gaze is a natural input for a Human Computer Interface (HCI) for disabled people, who have of course an acute need for a communication system. An efficient eye tracking procedure is presented providing a non-invasive method for real time detection of a subject eyes in a sequence of frames captured by low cost equipment. The procedure can be easily adapted to any subject and is adequately insensitive to changing of the illumination. The eye identification is performed on a piece-wise constant approximation of the frames. It is based on a discrete level set formulation of the variational approach to the optimal segmentation problem. This yields a simplified version of the original data retaining all the information relevant to the application. Tracking is obtained by a fast update of the optimal segmentation between successive frames. No eye movement model is required being the procedure fast enough to obtain the current frame segmentation as one step update from the previous frame segmentation.

1. Introduction

An HCI facility for a disabled person is a complex system composed of physical devices (such as video cameras, brain activity sensors, head movement sensors, special contact lenses, etc.) and signal analysis algorithms. The purpose consists in determining with sufficient accuracy the point of gaze of a subject exploring a pc screen, to provide a smart tool improving the subject autonomy both in terms of communication and environment interaction [1,2]. The various solutions proposed have different degree of invasiveness, depending on the technology involved: generally speaking the more sophisticated the devices the more invasive and higher cost [3–5] the HCI; correspondently signal analysis is quite simple. The use of non-invasive low cost technology demands complex algorithms to deal with the real time constraint. We consider indeed the latter situation: the subject is positioned in front of the screen of a general purpose pc endowed with a low cost video camera. A general plot of an eye tracking system is represented in Fig. 1.

Three main reference systems can be identified [6]: the eye reference system, the camera reference system and the screen reference system. The tracking system analyzes the image captured by the camera, determines the line of gaze of the subject watching the screen and identifies the point of the screen the subject is looking at. The position of the pupil center and the pupil shape on the image plane depend on the point on the screen the pupil is directed to, on the location of the user head, on the parameters of the eye model, on the position of the screen with the respect to the camera and on the camera intrinsic parameters.

Eye detection procedures usually exploit the pupil reflectance power to perform image zoning to separate the subject head from the environment and therefore identify the pupil shape and center by template matching. As suggested in [7], there are mainly three kinds of methods: template,

* *Corresponding author*. Tel.: +39 0677274061; fax: +39 0677274033.
E-mail addresses: desantis@dis.uniroma1.it (A. De Santis), iacoviel@dis.uniroma1.it (D. Iacoviello).

0169-2607/$ – see front matter © 2009 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2009.03.010

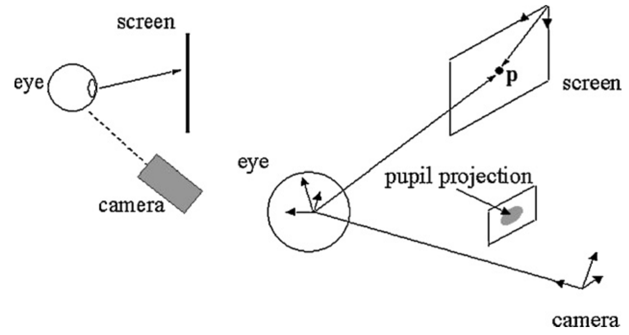


Fig. 1 – General eye tracking set-up and coordinate references.

appearance and feature-based methods. In the first one, a deformable template is designed to fit at best the eye shape in the image; the eye identification is usually accurate but needs a proper initialization, is computationally expensive and requires good image contrast [8,9]. The appearance methods require large amounts of data to train a classifier with the eye photometric appearance in different subjects under different face orientations [10,11]; a good performance can be obtained provided that the classifier has a good generalization property, which depends on the completeness of the training set. Feature-based methods rely on characteristics such as the pupil darkness, the sclera lightness, the iris circular shape, the shape of the eye corners, etc., to distinguish the human eye from the context. In [12] for instance, a picture zoning is performed by the Turk method [13] and eye blinking detection is accomplished to initialize the location of the eyes providing the position of the feature points for the eye tracking by means of the Lucas Kanade's feature tracker [14]. All these steps require a high image contrast to detect and track the eye corners. In [15], after a picture zoning is performed, the iris lower semi circle is detected as the curve that maximizes the normalized flow of the luminance gradient. In [7] eye detection and tracking is based on the active remote IR illumination approach, combined along with an appearance-based method. IR illumination approaches are simple and generate alternate bright/dark pupil images so that pupil detection is obtained after thresholding the difference image [16]; non-perfect illumination, occlusion and eye closure may limit the success of the eye tracking process. These drawbacks are avoided in [7] by using a support vector machine; consequently eye tracking is performed by a Kalman filtering approach reinforced by a mean shift tracking algorithm. In [17] eye corners, eyelids and irises are tracked and their motion is analyzed to determine gaze direction and blinking: here template matching is exploited to identify the eye elements whereas their motion is determined by the optical flow associated with their edge segments; head motion is estimated to detect head-independent eye movements such as saccades or smooth pursuit. In [18], the *between-the-eyes* pattern is considered for feature tracking, and head movement cancellation is performed for easier eyelids movement detection.

In this paper we present a robust and efficient procedure of image analysis to identify the eyes in a video sequence. This is the first block in the functional flow chart shown in Fig. 2; the other blocks are just standard and a possible implementation can be found in [6].

The new method is feature-based and relies on an image segmentation technique developed in [19] and applied to the identification of pupil shape parameters in [20]. In these papers the image segmentation process is accomplished by a region-growing algorithm based on a discrete level set formulation of an optimal approximation problem. It exploits a very general convex cost function that takes into account a fit error term, that evaluates the approximation error between the image and its segmentation, and regularity terms penalizing a segmentation with ragged boundary and small area sub domains. A further term penalizes large values of the level set function, and makes the cost function convex. The unique optimal solution provides a piece-wise constant segmentation of the original image with a lower number of gray levels and therefore with clear-cut edges, preserving all the information relevant to the considered application. The numerical efficiency and the quality of the segmentation of the discrete level set approach versus competing segmentation algorithms, in the class of edge detection, global thresholding and continuum level set, were tested in [19,20].

Region-growing algorithms are known to be robust with respect to various signal degradations such as additive noise, blurring, illumination changes. On the other hand they may be time consuming and therefore their use in an eye tracking application needs some adaptation. This issue is the main motivation of this work. The method in [19] is modified by considering a simpler cost function: only the fit error and the convexity terms are retained. Indeed, the segmentation boundary regularity term are computational expensive and their suppression in the considered application does not affect significantly the quality of the segmentation. The interconnection between frames is obtained by updating the current frame segmentation starting from the previous frame optimal segmentation. This also significantly contributes to the algorithm speed increase, and would not be possible with simple thresholding methods. An efficient picture zoning is performed to quickly identify the user head position, aiming to reduce the size of each frame to be analyzed and therefore to decrease the processing time.

The paper is organized as follows. In Section 2 design issues and the general set-up are presented. Then the optimal segmentation procedure is described in Section 2.1, whereas the feature extraction and pupil identification are presented in Section 2.2. The automatic calibration procedure is explained in Section 2.3 and the eye tracking procedure is found in Section 2.4. In the experimental Section 3 the eye tracking procedure is evaluated in terms of accuracy and robustness,

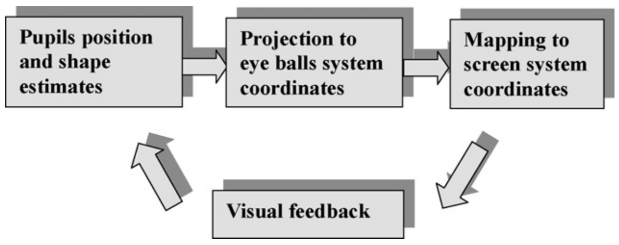

Pupils position and shape estimates
Projection to eye balls system coordinates
Mapping to screen system coordinates
Visual feedback

Fig. 2 – Flow chart of the eye tracking system.

and results of the application to real video sequences are discussed. Conclusions and further developments can be found in Section 4.

2. Design

The standard set-up we refer to is a regular pc facility: the user is sat in front of a screen endowed with a single low cost video camera, pointed on the subject face, Fig. 3. The videotaping occurs in a regular room illumination, no additional light sources are considered, like IR lamps.

To test the algorithm, described in the next sections, in critical conditions, we did not produce any ad hoc video sequence but rather used two sequences taken from the web, http://www.ecse.rpi.edu/homepages/cvrl/database/database.html, courtesy of Prof. Qiang Ji. The sequences are taken at 15 frames per second, each frame is an 8-bit gray level image with resolution of 320 × 240; each sequence presents moderate facial expression changes. In these sequences the subjects perform eye movements far more complex than those expected in an application for disabled people, like wide head rotation and eye occlusions.

2.1. *The optimal segmentation procedure*

Consider a simple 2D monochromatic image I with just one object over the background. The object boundary can be represented by the *boundary set* ϕ_0 of a function ϕ: $D \to R$, where $D \subset R^2$ is a grid of points (pixels) representing the image domain. The boundary set is defined as follows

$$\phi_0 = \{(i,j) : sign(\phi_{h,k}) \neq sign(\phi_{i,j}), \quad \text{for at least one } (h,k) \in [i \pm 1, j \pm 1]\} \tag{1}$$

Let us assume that the region $\{(i,j): \phi_{i,j} \geq 0\}$ coincides with the object; the pixels not belonging to ϕ_0 are either in the interior of the object or in the background. In this case it is easy to obtain a binary representation I_s of the original picture

$$I_s = c_1 \chi_{(\phi \geq 0)} + c_2 \chi_{(\phi < 0)} \tag{2}$$

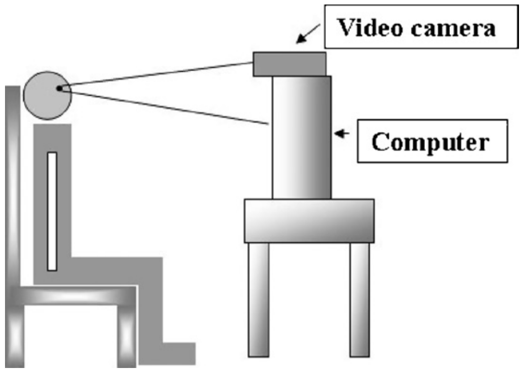


Fig. 3 – Experimental set-up.

where

$$\chi_{(\phi \geq 0)} = \begin{cases} 1 & \phi_{i,j} \geq 0 \\ 0 & \text{elsewhere} \end{cases}, \quad \chi_{(\phi < 0)} = \begin{cases} 1 & \phi_{i,j} < 0 \\ 0 & \text{elsewhere} \end{cases} \tag{3}$$

and c_1, c_2 are two different constant positive gray level values. Image I_s provides a piece-wise constant segmentation of the original picture I: it is a simpler representation of the data with a clear-cut between the object and the background; the information about the object shape is preserved. Function ϕ is called *level set function* and, according to (2), operates the image segmentation. Segmentation (2) can be obtained by solving an optimal approximation problem defined as follows:

$$\begin{aligned} \min_{(c_1,c_2,\phi)} E(c_1, c_2, \phi) &= \min_{(c_1,c_2,\phi)} \left[\lambda \|I - I_s\|^2 + \alpha \|\phi\|^2\right] \\ &= \min_{(c_1,c_2,\phi)} \left[\lambda \sum_{i,j} (I_{i,j} - c_1)^2 H(\phi_{i,j}) \right. \\ &\quad \left. + \lambda \sum_{i,j} (I_{i,j} - c_2)^2 (1 - H(\phi_{i,j})) + \alpha \sum_{i,j} \phi_{i,j}^2 \right] \end{aligned} \tag{4}$$

where H is the Heaviside function, λ and α are two positive parameters. The first two terms represents the fit error between the original data and the piece-wise constant approximation; the third term is a regularity term that makes the cost function convex [19]. Function E is a particular form of the energy functional used in [19,20]; it does not contain terms weighting the area and the boundaries length of the segmentation sub-regions, as the contribution of these terms is marginal to the current application and their computation is time consuming. Following the argument in [19], a smooth version of the cost function is advisable and it is obtained by substituting in (4) the Heaviside function by a smooth approximant

$$H_\varepsilon(\phi) = \frac{1}{1 + \exp(-\phi/\varepsilon)} \tag{5}$$

It can be proved that the smooth version of problem (4) admits necessary and sufficient conditions for a unique global minimum that, by standard calculus, are obtained as follows

$$c_1 = \frac{\sum_{i,j} H_\varepsilon(\phi_{i,j}) I_{i,j}}{\sum_{i,j} H_\varepsilon(\phi_{i,j})}, \quad c_2 = \frac{\sum_{i,j} [1 - H_\varepsilon(\phi_{i,j})] I_{i,j}}{\sum_{i,j} [1 - H_\varepsilon(\phi_{i,j})]} \tag{6}$$

$$\alpha \phi_{i,j} + \lambda [(I_{i,j} - c_1)^2 - (I_{i,j} - c_2)^2] \delta_\varepsilon(\phi_{i,j}) = 0 \tag{7}$$

where δ_ε is the derivative of function H_ε. Different kind of approximant can be considered, as for instance indicated in [21], but expression (5) is more convenient from the computational point of view. From (6) we note that c_1 is the image gray level average over the region in which $\phi \geq 0$, whereas c_2 is the average over the complementary region. Eq. (7) provides an implicit equation for the level set function in every pixel (i,

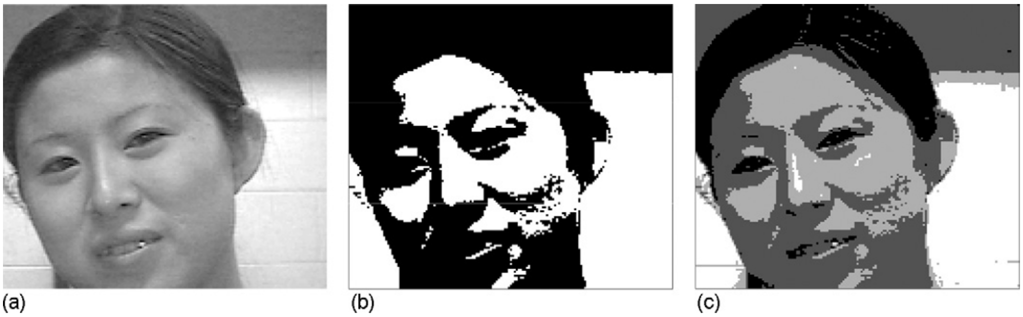


Fig. 4 – Image segmentation by the discrete level set procedure; (a) original; (b) first binarization; (c) four levels segmentation.

j); it can be solved by the following iterative scheme

$$\alpha\phi^{n+1}_{i,j} + \lambda[(I_{i,j}-c_1^n)^2 - (I_{i,j}-c_2^n)^2]\delta_\varepsilon(\phi^n_{i,j}) = 0 \qquad (8)$$

that has a fast convergence to the unique solution of (7), starting from any initial condition ϕ^0. The computational cost of any update of Eq. (8) can be easily obtained in terms of the image size $N \times M$; once constants c_1 and c_2 are known, the level set function can be updated according to (8) with $3(N \times M)$ sums and $3(N \times M)$ products. According to (6), constant c_1 requires $2(N \times M)$ sums, $N \times M$ products and one division; constant c_2 requires $3(N \times M)$ sums, $N \times M$ products and one division. Therefore one run of the algorithm (6)–(8) is linear in the image size $N \times M$ in terms of sums and products.

The level set method has the amenable property that during the level set evolution (8) the boundary set ϕ_0^n, starting from *any* initial shape ϕ_0^0, can *merge* and *split* in order to easily deal with the complex topology of the real world images.

Should we need to preserve other object appearance details, the number of gray levels can be increased to four by further segmenting the regions in (3) by the use of two more level set functions ϕ_+, ϕ_-

$$I_s = c_{11}\chi_{(\phi\geq0)}\chi_{(\phi_+\geq0)} + c_{12}\chi_{(\phi\geq0)}\chi_{(\phi_+<0)} + c_{21}\chi_{(\phi<0)}\chi_{(\phi_-\geq0)} + c_{22}\chi_{(\phi<0)}\chi_{(\phi_-<0)} \qquad (9)$$

where c_{ij}, i, *j* = 1, 2, are the constant gray level values within each of the four sub-regions individuated by the two level set functions. Segmentation (9) is therefore accomplished by a hierarchical procedure obtained by successive binarizations. First the image domain is partitioned in two regions {(i, *j*): $\phi_{i,j} \geq 0$} and {(i, *j*): $\phi_{i,j} < 0$}. Then each of these sub domains is further partitioned in two sub domains with the help of the additional level set functions ϕ_+, ϕ_-:

$$\{(i,j):\phi_{i,j}\geq 0\} = \left(\{(i,j):\phi_{i,j}\geq 0\}\bigcap\{(i,j):\phi_{+i,j}\geq 0\}\right)\bigcup\left(\{(i,j):\phi_{i,j}\geq 0\}\bigcap\{(i,j):\phi_{+i,j}< 0\}\right)$$

and

$$\{(i,j):\phi_{i,j}< 0\} = \left(\{(i,j):\phi_{i,j}< 0\}\bigcap\{(i,j):\phi_{-i,j}\geq 0\}\right)\bigcup\left(\{(i,j):\phi_{i,j}< 0\}\bigcap\{(i,j):\phi_{-i,j}< 0\}\right)$$

By the same argument the number of gray levels can be further increased, but for the eye tracking problem the four levels segmentation (9) proved to be sufficient to represent all the information relevant to the features extraction process. Fig. 4 shows the result of the hierarchical procedure.

The binarization of Fig. 4b will be referred to as *first binarization*. The four levels segmentation provides a *cartoon image* of the original data where the eyes always belong to the darkest part, whatever the colour of the eyes and the race of the subject.

2.2. *Features extraction and pupils detection*

The frame four levels segmentation, as obtained in Fig. 4, is not suitable for the identification of the subject pupils. We need a more detailed segmentation to distinguish the pupils from the other eye elements. To this aim a binary image can

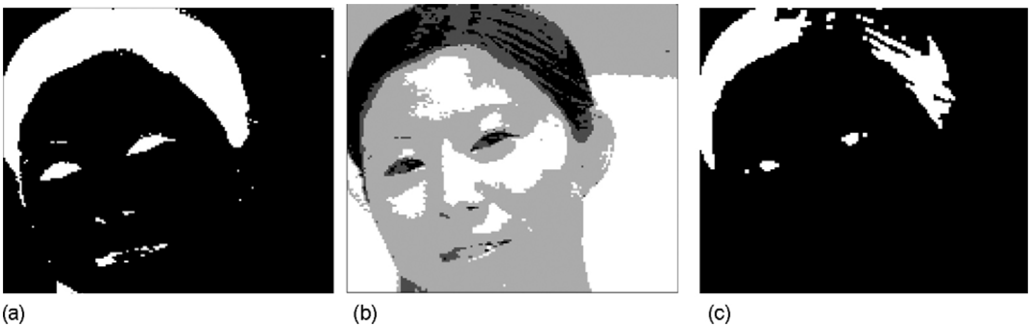


Fig. 5 – Pupils identification. (a) Image mask selecting the darkest elements in the early four levels segmentation in Fig. 4c; (b) final four levels segmentation where the pupils are separated from the sclera; (c) darkest elements of (b).

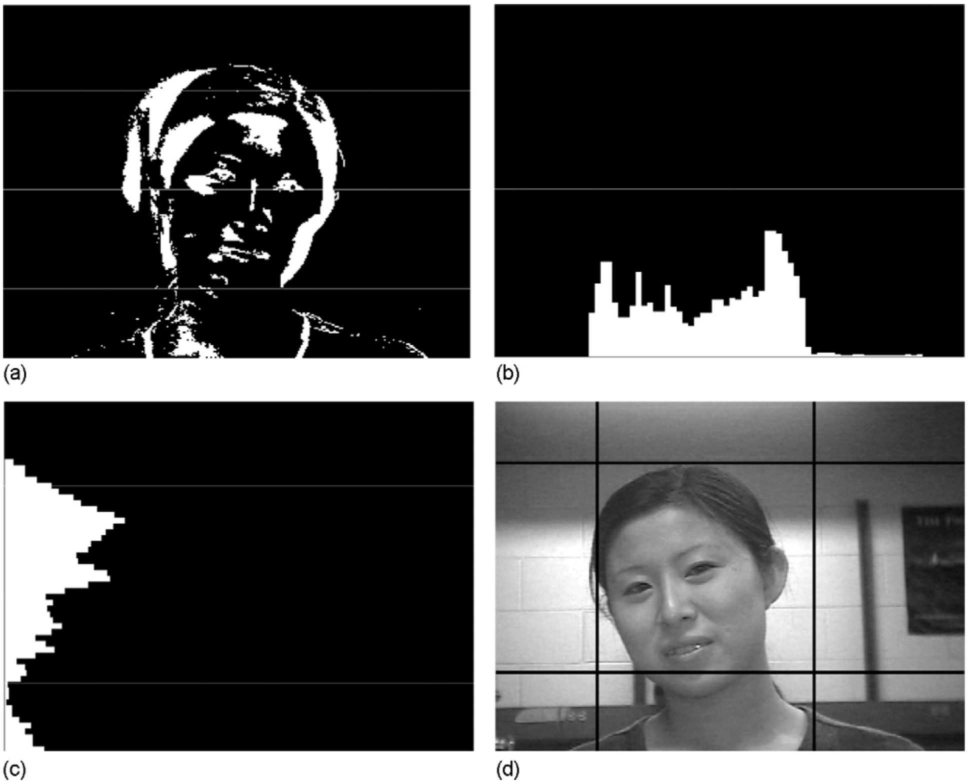


Fig. 6 – Frame zoning. (a) Thresholded difference picture between a frame and the average background; (b) histogram of the horizontal distribution of the white pixels in (a); (c) histogram of the vertical distribution of the white pixels in (a); (d) original frame with the zoning boundaries obtained by the analysis of histograms (b) and (c).

be obtained from the previous four levels segmentation, by a simple boolean operation, thus highlighting the dark elements (and therefore the eyes) and leaving the rest of the picture in the black background, see Fig. 5a.

The obtained binary image is a mask that is used to select two regions on the original image to be segmented, obtaining again a four levels segmentation where, this time, the pupils are well separated from the sclera. Indeed, a simple boolean operation allows now to identify the pupils as the darkest elements of Fig. 5b, thus obtaining the picture on Fig. 5c, that will be referred to as *final binarization*.

The Matlab©instruction *bwlabel* can now label all the white objects over the black background and the instruction *regionprops* evaluates the morphological characteristics of any labeled object. Among the others, the shape parameters of interest are: the area, the centroid coordinates, the major and minor axes length, the orientation. The area is just the number of pixels that build up the object. The centroid coordinates are computed as the coordinates of the object center of mass. The major and minor axes belong to the ellipse that has the same second order moment of the object, computed according to [22]. The orientation measures the angle formed by the major axis and the horizontal axis. All these quantities can be used to identify the subject pupils, among the other objects on the final binarization, as will be explained in the next subsection.

2.3. Automatic calibration

The main problem in an eye tracking procedure is a trade-off between accuracy and processing time. In this respect

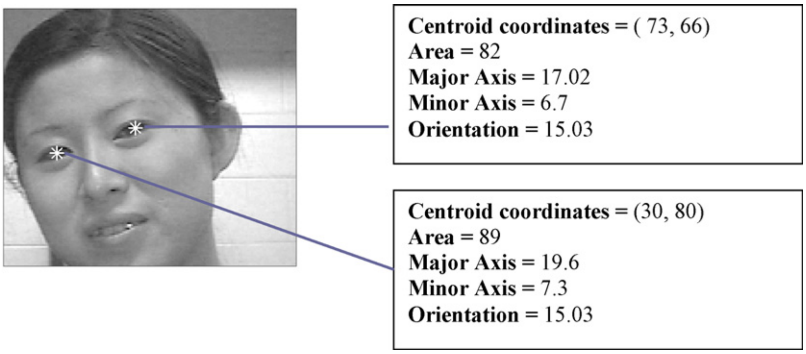


Fig. 7 – Estimation of pupils position and some shape parameters.

the data size is an issue and therefore it may be necessary to reduce it for further speed increase. This can be obtained by a frame zoning procedure aimed at identifying the frame portion containing the subject head. Consider that between adjacent frames (within 1/15-th of a second) the picture background practically remains the same whereas the subject position changes only slightly. Before the tracking starts, few seconds of the video acquisition can be used to estimate the pixel-to-pixel average and standard deviation of the gray level. Therefore the pixel-to-pixel comparison of one frame to the average clearly indicates the frame portion where the head movement has occurred, see Fig. 6a. In this picture, in the pixels marked as white, the difference between the gray level and the average was, in absolute value, greater than the standard deviation.

The spatial distribution of the white pixels in Fig. 6a along the horizontal and the vertical directions can be easily obtained. The horizontal spatial distribution, for instance, is determined by dividing the picture into vertical strips, that represent the histogram bins, and counting the number of white pixels into each strip. The distribution concentrates in the picture zone where the head movement occurred, Fig. 6b (for the vertical distribution see Fig. 6c). The average and the standard deviation of the two distributions are finally used to build a box that contains the subject head with a suitable tolerance, Fig. 6d. This zoning need not to be updated, since any subject working while sitting in front of a pc does not drastically move his body. Should this be the case the zoning could be easily repeated, at a rate slower than the frame acquisition rate of course, and used as a feedback to the tracking algorithm. Once the zoning has been performed, the calibration is completed by a reliable subject pupils detection. This is obtained by determining the final binarization, according to the procedure of Section 2.2, on the last frame acquired in the early short acquisition. Now the pupils can be identified since they are a couple of objects with very similar size and shape characteristics, Fig. 7.

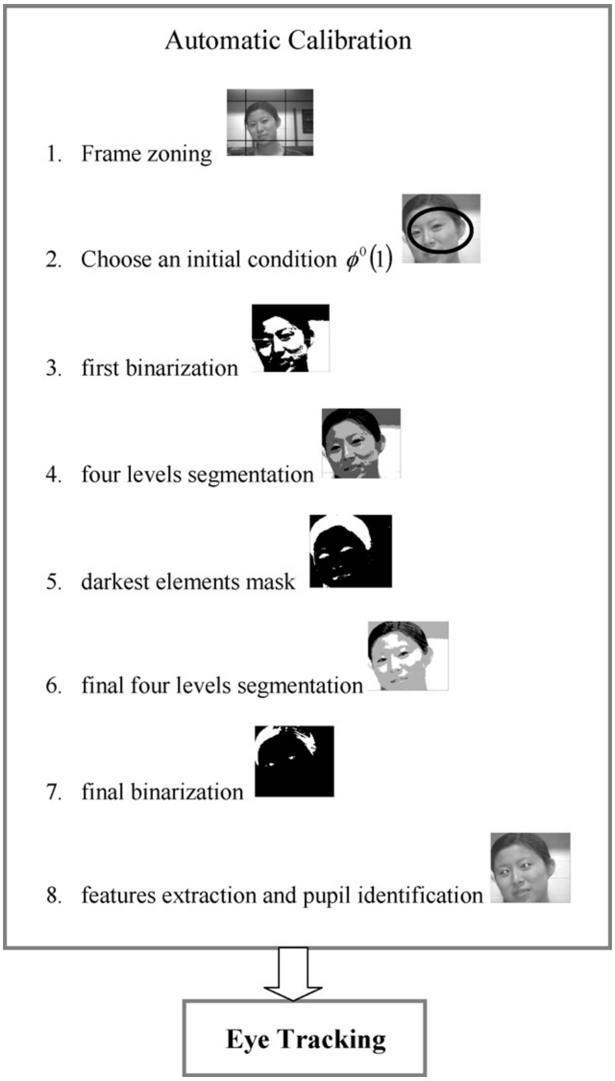

Automatic Calibration
1. Frame zoning
2. Choose an initial condition $\phi^0(1)$
3. first binarization
4. four levels segmentation
5. darkest elements mask
6. final four levels segmentation
7. final binarization
8. features extraction and pupil identification
Eye Tracking

Fig. 8 – Flow chart of the initial automatic calibration procedure. The computed eye signature is passed to the next block to start the eye tracking, along with the final configuration $\phi(1)$ of the level set function that has performed the first binarization at step 3.

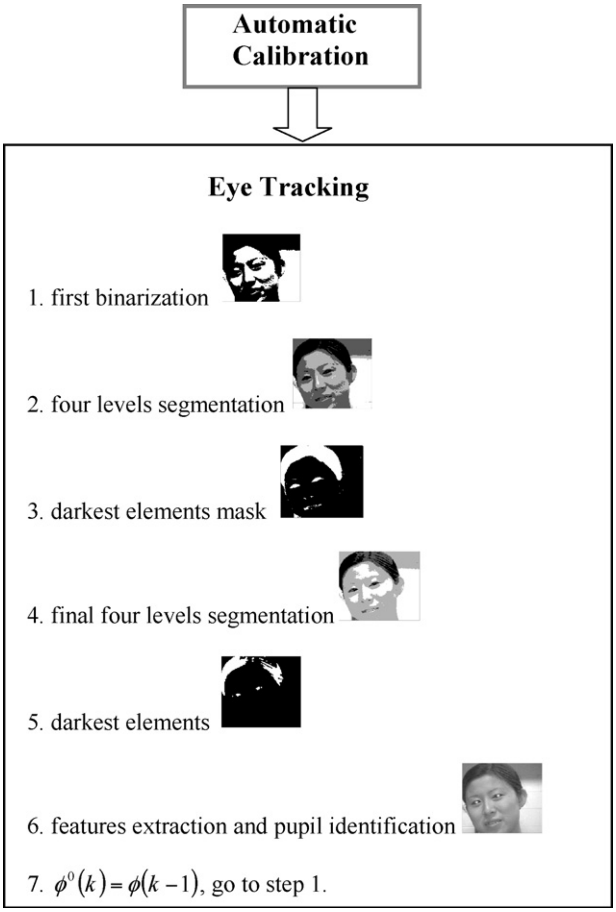

Automatic Calibration
Eye Tracking
1. first binarization
2. four levels segmentation
3. darkest elements mask
4. final four levels segmentation
5. darkest elements
6. features extraction and pupil identification
7. $\phi^0(k)=\phi(k-1)$, go to step 1.

Fig. 9 – Flow chart of the eye tracking procedure. Step one is executed for the first frame after calibration by initializing the level set function at $\phi(1)$ coming from the previous block.

These quantities can be stacked in two features vectors that represent the *eyes signature*. If needed, to avoid false detections due to artefacts, anthropometric measure can be used to take into account, for instance, the between-the-eyes span.

This procedure is applied only in the calibration step and, for the available sequences, requires no more than 5 runs of algorithm (6)–(8) for each segmentation leading to the final binarization. During the eye tracking, the pupils can be directly identified by a fast procedure explained in the next subsection. The flow chart of Fig. 8 summarizes the initial automatic calibration procedure.

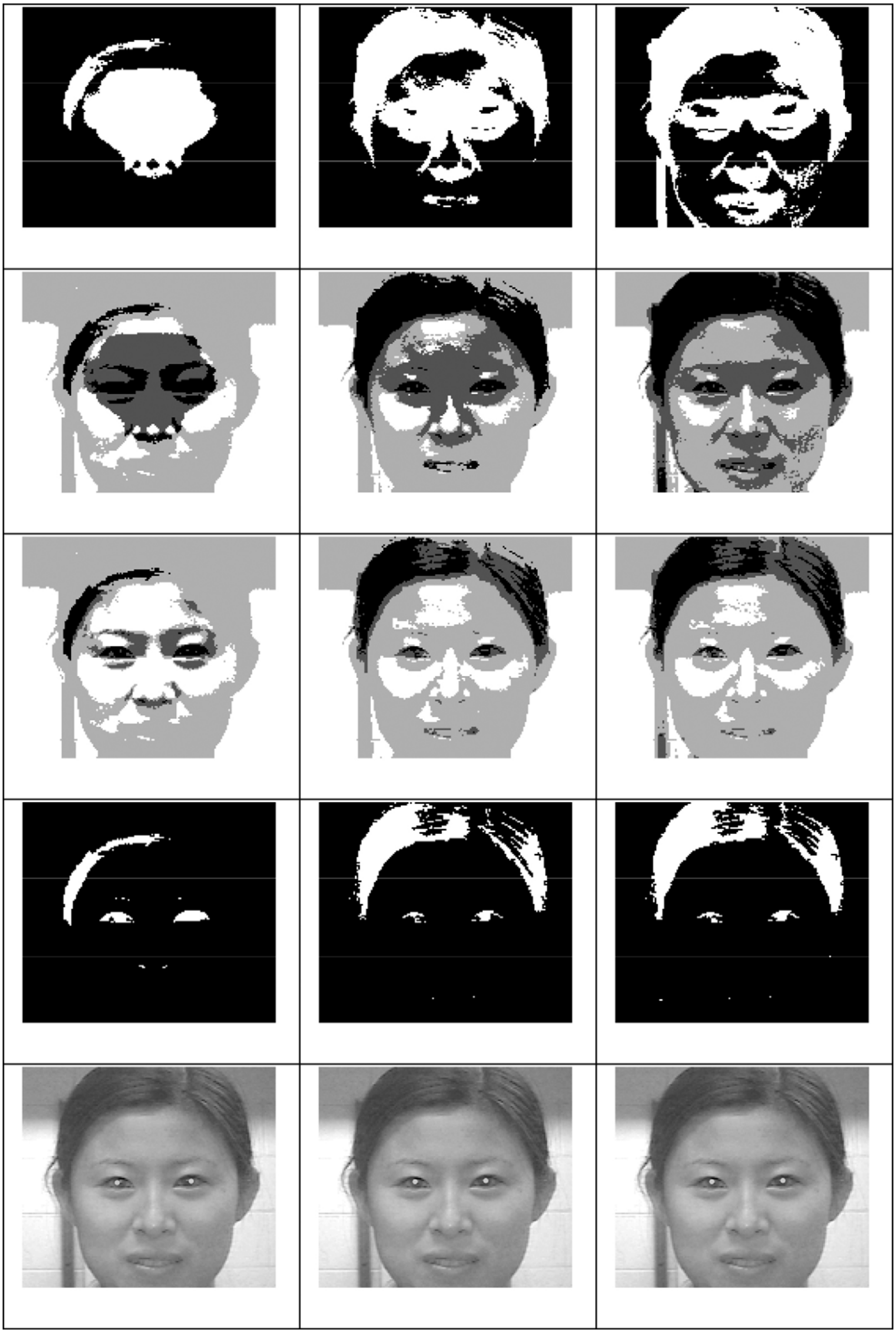


Fig. 10 – Segmentation results with λ = 200 (first column), λ = 500 (second column), λ = 10^3 (third column). In the first row the first binarizations are displayed. The second and third rows show the first and second four levels segmentations, respectively. In the fourth row the final binarizations are shown and in the last row the estimated pupils position are marked on the original frame.

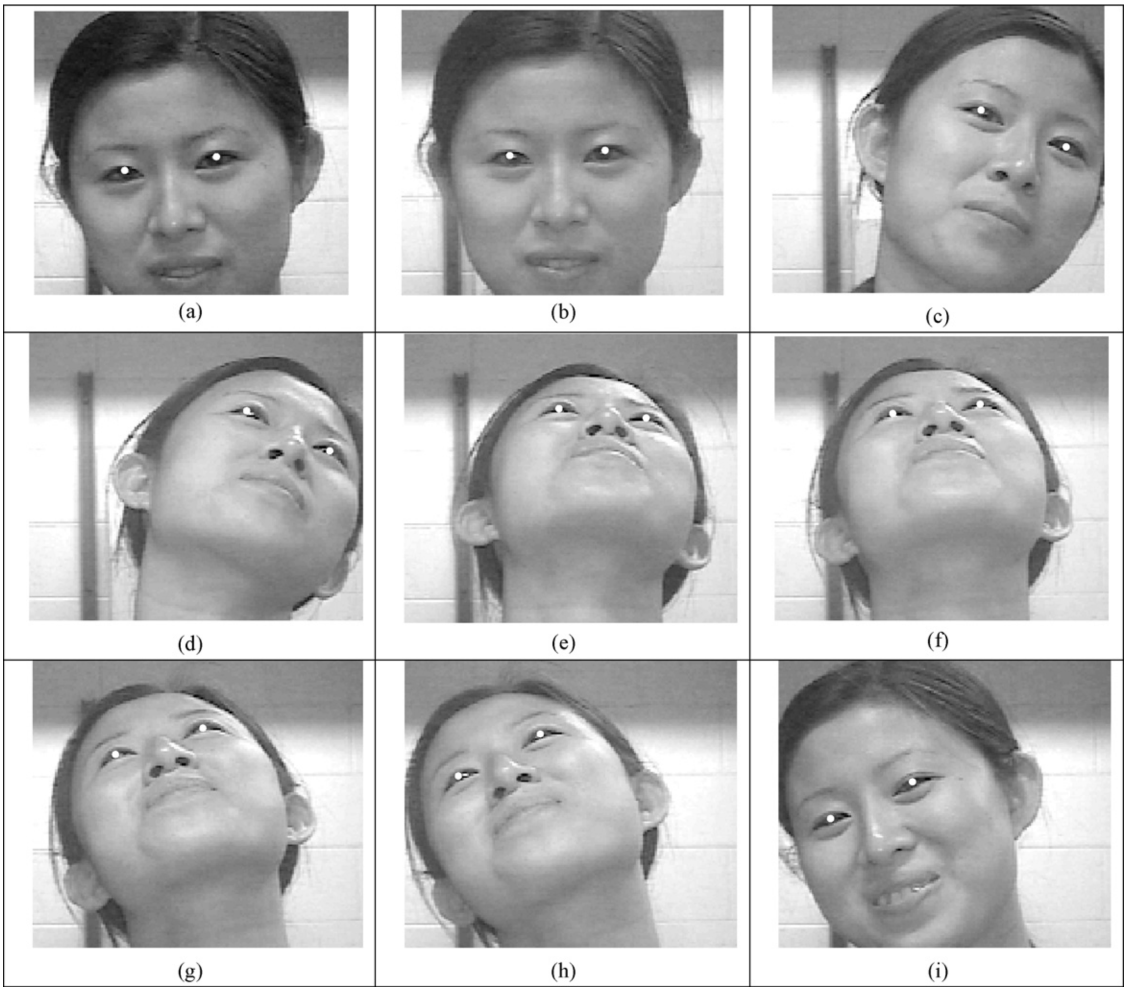


Fig. 11 – Pupils tracking, some significant shots with gamma correction equal to 0.5.

2.4. *Eye tracking*

The eye tracking algorithm requires that the current frame be segmented quickly so to obtain in real time the eye features vectors. Let us suppose that the frame $(k-1)$ has been processed: this means that the hierarchical procedure of Section 2.1 has been completed and the level set function $\phi(k-1)$, that accomplished the first binarization, has reached its final configuration that solves (8). To apply the segmentation to the next frame k we need to choose an initial configuration $\phi^0(k)$ of the level set function to be evolved according to (8). It can be chosen equal to $\phi(k-1)$ to improve the rate of convergence of Eq. (8) since, due to the similarity of adjacent frames, the initial condition $\phi^0(k)=\phi(k-1)$ is very close to the solution of (8) for frame k. Indeed, each binarization step to obtain the final binarization of the k-th frame requires just one run of the algorithm (6)–(8).

On the final binarization for the current frame k all the objects are labeled and their shape parameters evaluated. Now the eyes signature computed at the step $(k-1)$ are exploited to quickly identify the pupils among all the labeled objects. This is obtained by simply matching the features vectors of these objects to the eyes signature of the previous frame. In this way false detections can be easily avoided. This matching procedure allows also to resolve eye occlusions, for instance due to blinking; indeed the eyes position is not updated until matching occurs. The matching is obtained when the normalized euclidean distance of the eyes signature between adjacent frames is within a chosen percentage. Normalization is taken with respect to the norm of the previous frame eyes signature vectors. The eye tracking procedure is outlined in the flow chart of Fig. 9.

3. Experiments

Before testing the tracking algorithm we want to support the choice of simplifying the segmentation cost function, as in (4), with respect to the general expression given in [19] where, besides the fit error, the segmentation boundary regularity was taken into account. In the first case the segmentation algorithm, denoted by A1, depends on parameters α and λ, while in the general case algorithm, denoted by A2, two more parameters have to be chosen: μ that weighs the area of the segmentation sub domains, ν that weighs the length of their boundary. Parameter ε only defines the degree of approximation of the generalized functions H and δ, with their approximants H_ε and δ_ε, and is usually chosen equal to 1. The cost function expression in both cases can be normalized to the value of α, that is definitely set equal to 1. The reciprocal importance of the parameters for A2 was already assessed in [19], showing that the segmentation is more sensitive to λ; therefore also parameters μ and ν are set equal to 1. The choice of λ depends on the image contrast, but it is independent of its brightness: the lower the contrast the higher λ; values ranging from 10^2 to 10^5 have proven to give satisfactory results in general applications. For all these reasons, A1 is expected to be numerically more efficient than A2, with acceptable degradation of the accuracy. Indeed, we performed a four levels segmentation for 50 frames of the available sequence, and measured each time the fit error. In all the binarizations of the hierarchical segmentation procedure, 10 runs of the level set function update were considered with $\lambda = 10^3$, both for A1 and A2. For A2 a fit error with mean value of -6.8×10^{-4} and standard deviation of 0.06 was obtained within a cputime of 2.9 s, whereas for A1 the fit error mean value was equal to 1.3×10^{-4} with a standard deviation of 0.08, and a shorter cputime of 1.01 s. The algorithms have been implemented in Matlab© on a 2.6 Ghz pc. This result clearly indicates that the simplified cost function is the best choice when real time is an issue in the chosen application. This early experiment also shows the need for a frame resizing to reduce considerably the processing time.

Next we justify the choice of the value of λ. In Fig. 10, the results of every step of the hierarchical segmentation procedure are displayed in columns for values of $\lambda = 200$, 500, 10^3, respectively.

It can be seen that the segmentation results improve as λ increases, and are already good for $\lambda = 500$. Nevertheless on frames when the subject is looking aside the screen, the eye zone may suffer a contrast decrease, so that the higher value of λ is anyway preferred.

Now we perform the eye tracking on two sequences obtained from the web at http://www.ecse.rpi.edu/homepages/cvrl/database/database.html. For better signal conditioning a gamma correction equal to 0.5 was performed on the sequences frames, and $\lambda = 10^3$ was chosen. In Fig. 11, nine snapshots of the first video sequence are displayed; the subject performs a wide head movement so that various eyes orientations are present. Even though eye occlusion does not really happen, a very critical situation occurs when the subject is looking up. This indeed shows the robustness

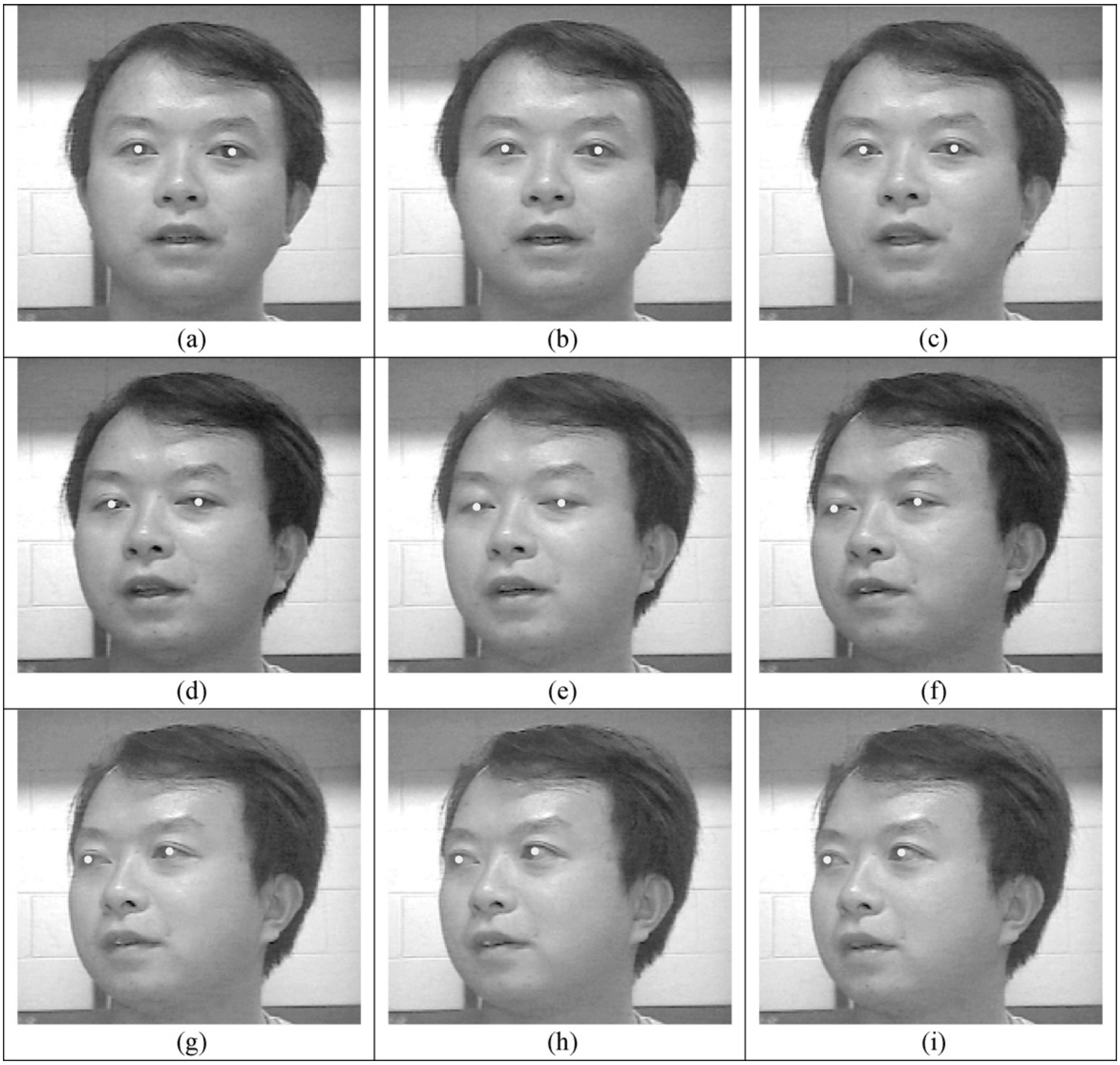


Fig. 12 – Pupils tracking, some significant shots with gamma correction equal to 1.5.

of the procedure with respect to different face orientation. This is mainly due to the tracking mechanism based on the segmentation update between successive frames and the matching of the eyes signature. The algorithm was able to track the pupils in all the 300 frames and the error on the pupils position, compared to the positions estimate provided by IR illumination as described in [7], was characterized by an average of 1 pixel with a standard deviation of 0.8. The used ground truth data was available on the database.

The experiment of Fig. 12 shows the robustness of the method with respect to eye occlusions. In this case, when one pupil, or even both, is lost because of the occlusion, the algorithm keeps memory of the last position correctly identified; this is updated as soon as the occlusion is removed and matching with the eyes signature occurs. Should the loss of matching of the eyes signature not recovered in few frames (for example, after occlusion), the procedure should be restarted from the beginning with the calibration procedure. Also in this case the error on the pupils position, compared to the positions estimate provided by IR illumination, was evaluated: it was characterized by an average of 1.45 pixel with a standard deviation of 0.806.

As final experiment the robustness of the tracking procedure with respect to changes of illumination was assessed. To this aim, the first sequence analyzed was artificially darkened/lightened by subtracting/adding a value of 0.3 to the gray level of the frames. Then we applied the eye tracking procedure to the new sequences obtained and evaluated the pupils position error with respect to the ground truth data. For the darkened sequence the error mean value was equal to 1.62 with standard deviation of 0.83, whereas for the lightened sequence the error had a mean value of 1.62 with standard deviation of 0.84. There is an increase in the average of the position error due to the signal variation but the accuracy remains practically the same, this denoting a good degree of robustness with respect to uniform changes of illumination.

The algorithm featured encouraging performances both in terms of accuracy of the pupils identification and speed, the latter being fundamental for real time applications. Indeed, the frame-to-frame processing for eye tracking takes 5 runs of the segmentation algorithm (6)–(8); each run requires a number of products and sums linear in the image size. The whole eye tracking procedure of Section 2.4 took 0.3 s on a pc endowed with a processor Intel Core Duo—2.6 Ghz and 4 Gbyte of RAM memory. Since the acquisition rate was 15 frames per seconds, we were able to process one frame over 5. This fact did not cause any tracking problem in these experiments, but of course it could be an important issue on other applications.

4. Conclusions

Eye tracking systems for disabled people should have some key characteristics: non-invasive low cost equipment, extremely simple calibration and robustness with respect to changes in illumination conditions. A tracking procedure has been developed able to track the subject eyes with no additional IR illumination. The method relies on a segmentation algorithm based on a discrete level set formulation of the optimal segmentation problem. The key feature of the method consists in a fast update of the optimal segmentation between successive frames. The tracking procedure shows a satisfactory degree of robustness with respect to various face orientations, eye occlusions, uniform change of illumination. Real time processing is attainable thanks also to a proper picture zoning reducing the data size. Further developments may include peculiar situations where, for example, the subject is wearing eye glasses, or where a changing in the background takes place.

Conflict of interest

The submission of the present work does not cause any conflict of interest of the Authors with any public Institution or private Company.

Acknowledgments

The authors wish to acknowledge the courtesy of Prof. Qiang Ji at Renssealer Institute for granting us the permission to use data from the database of his lab. They are also grateful to Dr. Jixu Chen at Renssealer Institute for all his collaboration in making the sequences available to them.

REFERENCES

[1] P. Majaranta, K.J. Raiha, Twenty years of eye typing: system and design issues, in: Eye tracking Research & Applications: Proceedings of the Symposium on ETRA, New York, 2002, pp. 15–22.

[2] D. Heckenberg, Performance evaluation of vision-based high DOF human movement tracking: a survey and human computer interaction perspective, in: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop, 2006, pp. 156–164.

[3] L.H. Yu, M. Eizenman, A new methodology for determining point-of gaze in head-mounted eye tracking systems, IEEE Transactions on Biomedical Engineering 10 (51) (2004) 1765–1773.

[4] D. Ishima, Y. Ebisawa, Eye tracking based on ultrasonic position measurement in head free video-based eye-gaze detection, in: Proceedings of IEEE EMBS Asian-Pacific Conference on Biomedical Engineering, 2003, pp. 258–259.

[5] A. Glenstrup, T. Angell-Nielsen, Eye Controlled Media, Present and Future State, Technical report, University of Copenaghen (1995), http://www.diku.dk/users/panic/eyegaze/.

[6] A. Villanueva, R. Cabeza, S. Porta, Eye tracking: pupil orientation geometrical modeling, Image and Vision Computing 24 (2006) 663–679.

[7] Z. Zhu, Q. Ji, Robust real-time eye detection and tracking under variable lighting conditions and various face orientations, Computer Vision and Image Understanding 98 (2005) 124–154.

[8] A. Yuille, P. Hallinan, D. Cohen, Feature extection from faces using deformable templates, International Journal of Computer Vision 8 (2) (1992) 99–111.

[9] K.M. Lam, H. Yan, Locating and extracting the eye in human face images, Pattern Recognition 29 (1996) 771–779.

[10] J. Huang, H. Wechsler, Eye detection using optimal wavelet packets and radial basis functions, International Journal of Pattern Recognition and Artificial Intelligence 13 (7) (1999) 1009–1025.

[11] W.M. Huang, R. Mariani, Face detection and precise eyes location, in: Proceedings of the International Conference on Pattern Recognition, 2000.

[12] T. Morris, P. Blenkhorn, F. Zaidi, Blink detection for real-time eye tracking, Journal of Network and Computer Applications 25 (2002) 129–143.

[13] M.A. Turk, Interactive-time vision: face recognition as a visual behaviour, MIT, Doctoral Thesis.

[14] X. Xie, R. Sudhakar, H. Zhuang, On improving eye feature extraction using deformable templates, Pattern Recognition 27 (1994) 791–799.

[15] Z. Hammal, C. Massot, G. Bedoya, A. Caplier, Eyes segmentation applied to gaze direction and vigilance estimation, in: International Workshop on Pattern Recognition for Crime Prevention, Security and Surveillance, 2005, pp. 236–246.

[16] C. Morimoto, D. Coons, A. Amir, M. Flickner, Pupil detection and tracking using multiple light sources, Image and Vision Computing, Special Issue on Advances in Facial Image Analysis and Recognition Technology 18 (4) (2000) 331–335.

[17] S. Sirohey, A. Rosenfeld, Z. Duric, A method of detecting and tracking irises and eyelids in video, Pattern Recognition 35 (2002) 1389–1401.

[18] S. Kawato, N. Tetsutani, Detection and tracking of eyes for gaze-camera control, Image and Vision Computing 22 (2004) 1031–1038.

[19] A. De Santis, D. Iacoviello, Discrete level set approach to image segmentation, Signal, Image and Video Processing 1 (2007) 303–320.

[20] A. De Santis, D. Iacoviello, Optimal segmentation of pupillometric images for estimating pupil shape parameters, Computer Methods and Programs in Biomedicine, Special Issue on Medical Image Segmentation 84 (2006) 174–187.

[21] T. Chan, L. Vese, Active contours without edges, IEEE Transactions on Image Processing 10 (2) (2001) 266–277.

[22] Haralick-Shapiro, Computer and Robot Vision, Vol. I, Addison-Wesley, 1992.