INTEGRATING HEAD POSE TO A 3D MULTI-TEXTURE APPROACH FOR GAZE DETECTION

Hanan Salam[1] , Renaud Seguier[1] and Nicolas Stoiber[2]

[1]IETR/Team SCEE, Supelec, Avenue de la Boulaie, Césson Sévigné, France
hanan.salam@supelec.fr, renaud.seguier@supelec.fr
[2]Dynamixyz, 80 Avenue du Bottes de Coesmos, Rennes, France
nicolas.stoiber@dynamixyz.fr

ABSTRACT

Lately, the integration of gaze detection systems in human-computer interaction (HCI) applications has been increasing. For this to be available for everyday use and for everybody, the imbedded gaze tracking system should be one that works with low resolution images coming from ordinary webcams and permits a wide range of head poses. We propose the 3D Multi-Texture Active Appearance Model (MT-AAM): an iris model is merged with a local eye skin model where holes are put in the place of the sclera-iris region. The iris model rotates under the eye hole permitting the synthesis of new gaze directions. Depending on the head pose, the left and right eyes are unevenly represented in the webcam image. Thus, we additionally propose to use the head pose information to ameliorate gaze detection through a multi-objective optimization: we apply the 3D MT-AAM simultaneously on both eyes and we sum the resulting errors while multiplying each by a weighting factor that is a function of the head pose. Tests show that our method outperforms a classical AAM of the eye region trained on people committing different gaze directions. Moreover, we compare our proposed approach to the state-of-art method of Heyman et al. [12] which manually initialize their algorithm: without any manual initialization, we obtain the same level of accuracy in gaze detection.

KEYWORDS

Active appearance model, iris tracking, gaze detection

1. INTRODUCTION

”The Eyes are the window to the soul”, a very famous English proverb, demonstrates a very important feature of the face: the eye. This feature with its actions that can be grouped into gazing, blinking and winking, carry information about the person’s intentions, thinking and interior emotions. Moreover, the eyes language is known among all cultures where people communicate with their eyes to send messages to each others. For instance, staring might mean attraction and rolling eyes means dislike of the other’s speech. Concerning Human-Human Interaction, detecting and interpreting eyes actions is an easy task. Our brains are used to such analysis. However, when it comes to Human Computer Interaction (HCI), this task is indeed difficult.

Lately, the integration of automatic gaze detection systems in HCI applications has been increasing. Such applications include entertainment interactive applications such as virtual reality or video games [21,16], aiding disabled people through eye typing or eyes for moving cursors [19] and applications anticipating human behavior understanding through eyes. For an automatic gaze detection system to be available for everyday use and for everybody, the imbedded gaze tracking system should work with low resolution images coming from ordinary webcams, and can cope with the fast movements of the eyes, with large head rotations and with different eye closure. In Salam et al. [27], we have proposed to separate the appearance of the eye skin from that of the iris texture. To do this, we merge an iris AAM and a local eye model where holes are put in the place of the sclera-iris region. The iris model slides under the eye hole permitting the synthesis of new gaze directions. The method models the iris texture in 2D. This does not take into consideration the fact that the iris looks elliptical when it comes to extreme gaze directions. In order to account for this, we extend this work by modelling the eyeball as a sphere.

The idea of separating the appearance of the eye skin from that of the iris resembles that of Orozco et al. [23] since they use two separate appearance trackers for the iris and the eyelids. However, their approach differs from ours by several points. First, we do not use two separate trackers. Rather, we combine an iris texture with the eye skin model and the texture slides under the skin, resulting in one model for the eye region capable of modelling different gaze directions. Second, they manually initialize their trackers on the first frame which ensures good results during their tracking, whereas our method is fully automatic. Third, they do not tackle the problem of big head pose.

Few methods integrate head pose to improve the eye localization in the eye. In this context, we propose to use the information of the pose for the amelioration of iris localization through a multi-objective AAM. We apply an iris localization algorithm simultaneously on both eyes and sum the resulting errors while multiplying each by a weighting factor that is a function of the head pose.

This paper is organized as follows: In section 2 we provide an overview of the related research. In section 3 detail our system together with its different rubrics. In section 4, we detail the multi-objective optimization. In section 5, we show some experimental results. Finally, in section 6, we draw our conclusions.

2. STATE-OF-THE-ART

Research in the area of gaze detection is very active and many methods exist in the literature. These methods can be classified into: shape-based, appearance-based and hybrid methods. A detailed state-of-the-art on eye and gaze tracking techniques can be found in Hansen and Ji [10].

2.1. Shape-based methods

Shape-based methods contain those based on deformable shape templates of the eye. They scan the image to find image pattern candidates that resemble the shape template and then filter candidates using some similarity measure [3,4]. Such methods need the definition of a set of initial parameters for the template that should be initialized close to the eye. Furthermore, they usually record a failure with large head rotations. Some methods use ellipse fitting [26].

These succeed at finding the location of the iris or the pupil on the condition that high resolution images are provided. Another drawback is their incapability to cope with the different states of closure of the eyelids. Other methods are edge detection based. [20] proposed an iris tracker based on the Hough Transform. Their approach needs constant illumination and little to no head translation or rotation; as such, they do not tackle the problem of head pose. [33] used isophotes based approach. However, their method fails with closed eyes, very bright eyes, strong highlights on the glasses, eyes with variable lighting, and on large head poses. In their approach, the eye region is assumed frontal. The following formatting rules must be followed strictly.

2.2. Appearance-based methods

Appearance based methods include image template based approaches [9] which rely on the construction of an eye template that is compared to image patches in the face. Such methods have problems with scale and head pose variation. Hillman et al. [13] construct the image patch model starting from a learning database through eigenfaces.

Other methods train classifiers such as Huang and Wechsler [15] that uses neutral networks to detect the eye. These need large training databases to work well.

2.3. Hybrid methods

Hybrid methods are more robust. They exploit both of the shape and appearance of the eye to find the iris location. Some methods present manually designed models [25,22]. To avoid manual design of the eye model, one can train Active Appearance Models (AAM) [7] on real data. Building an AAM for iris localization requires training a set of hand-labelled faces with different gaze directions. Thus, the appearance of the eye is learned simultaneously with that of the face [2,17]. The inconvenience is the need for a large training database in order to arrive at a reliable AAM for iris localization.

Other methods use InfraRed light sources [11,18] to detect the reflection of one or more InfraRed light sources in the pupil. These methods are efficient, but they have the drawback of being strongly dependent on the brightness of the pupils which can be influenced by the eye closure and occlusion, external illumination and the distance of the user from the camera. In addition, the requirement of IR light sources is itself a constraint, and such methods are limited to indoor use.

2.4. Head pose in gaze direction

All of the above mentioned methods neglect face orientation in the detection of the iris location. However, we can find in the literature methods that use iris location results to detect pose [24] or vice versa [32]. Reale et al [25] also use head pose to find the position of the eyeball from which they use to detect the iris locations. Yet, very few integrate head pose to improve the eye localization except Valenti et al. [33] that normalizes the eye regions by the pose to improve eye centre localization. Heyman et al. [12] perform gaze detection with respect to the head pose instead of the camera using blob detection for iris tracking.

Compared to the state of art, the proposed system works with low resolution images, it does not constrain the user with special requirements (IR illumination, hardware equipment. . .) and it makes use of the appearance and shape of the eye while avoiding explicit manual design of the

model. With respect to classical AAM, it has the advantage of restricting the learning database of AAM to people in frontal view and looking in front of them where there is no need to learn on people with different gaze direction.

3. PROPOSED SYSTEM

In this section, we introduce our global system for the detection of the iris location which is based on a 2.5D AAM [28] and we detail its different rubrics.

3.1 2.5D Active appearance models

We use the 2.5D AAM of Sattar et al. [28]. The latter is an extension of the 2D AAM of Cootes et al. [7]. It is constructed using 3D annotations of the face forming the shapes and 2D textures of the frontal view of the facial images. With 2.5D AAM, there is no need to learn on people with different poses in order to align faces with different poses. This is convenient for us since we aim to reach a model that is capable of detecting the iris starting from a learning base that only contains people in frontal view and looking in front of them.

3.2. System overview

Let us consider that the subject is in front of the screen where a webcam is installed (first block of figure 1). Depending on the face orientation, the left and right eyes are unevenly represented in the webcam image. We thus propose to analyze gaze direction using a multi-objective optimization (with 3D Multi-Texture AAM (3D MT-AAM) for each eye): The contribution of each eye to the final gaze direction is weighted depending on the detected face orientation.

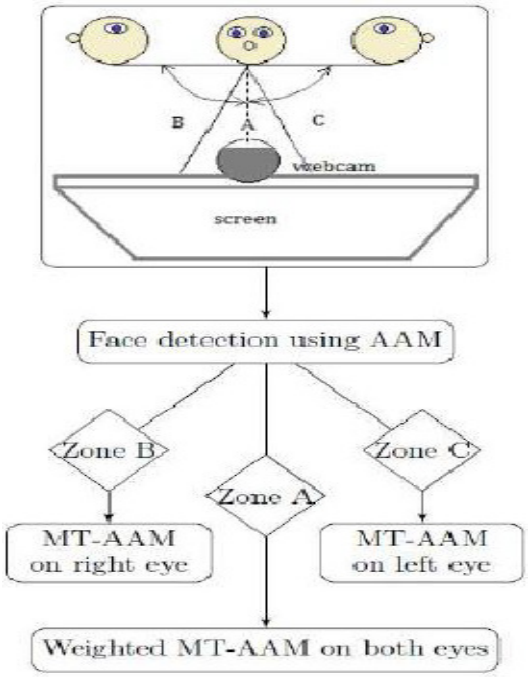


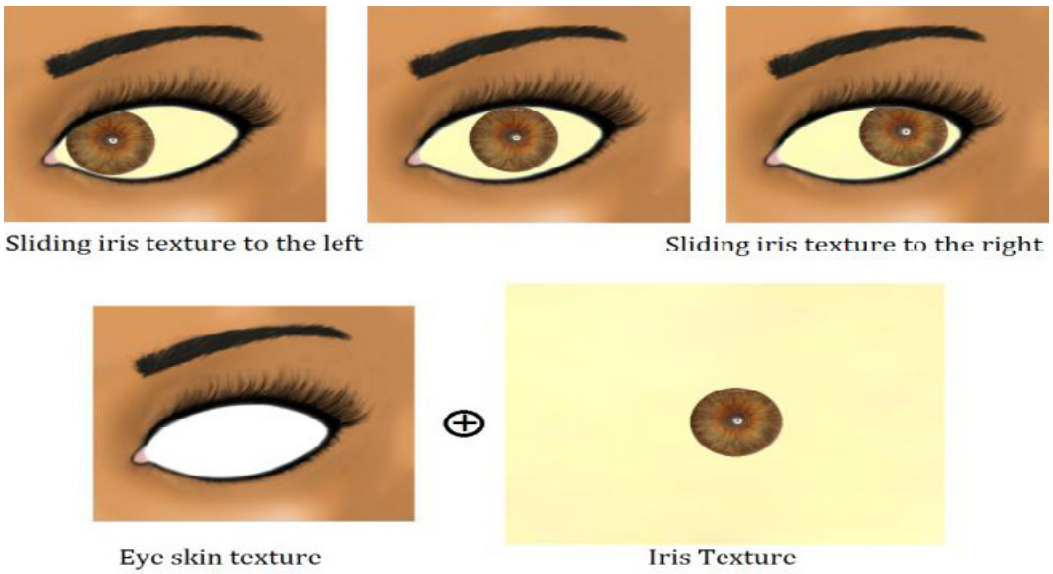
Figure 1.Global system overview. Zone A means that the face rotation is sufficiently small such that both eyes appear in the camera. Zones B and C signify that the right or the left eye appear more in the camera respectively

Figure 1 depicts the steps of our global system. We distinguish between three main zones depending on head orientation.

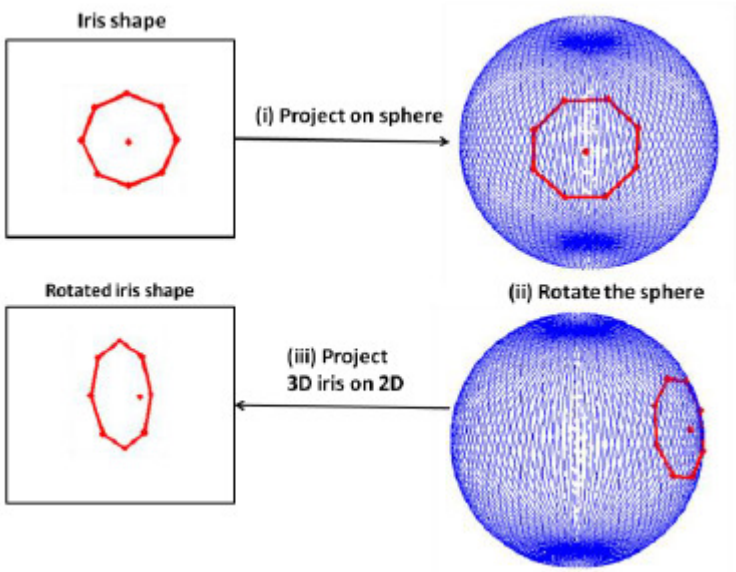
A. The subject could have a head pose such that both of his eyes appear clearly on the screen;
B. The subject shows a large head rotation to the left such that the right eye appears the most in the camera;
C. The subject shows a big head pose to the right such that the left eye appears more;

The algorithm works as follows: The first step is the detection of the head pose for which a 2.5D global AAM model [28] is applied. If the head pose corresponds to the Zone A, then both eyes will be integrated in the detection of the gaze using a weighting function (see section 4). If it is in the Zone B, an MT-AAM will be applied only on the right eye. If it is in the Zone C, then MT-AAM will be applied only on the left one.

In the following, we describe how the iris location is calculated for one eye using the proposed MT-AAM. Then in section 4 we present the multi-objective optimization which considers two MT-AAM.



(a) Multi-texture idea illustration. Moving the iris texture behind the eye skin surface creates different gaze directions



(b) Illustration of modelling the iris as a part of a sphere. See how the iris appearance becomes elliptical in appearance for extreme gaze directions.

Figure 2. Idea illustration

3.3. 3D Multi-texture AAM for iris localization

The basic idea of a multi-texture AAM (MT-AAM) is that the interior of the eye is considered as a separate texture from that of the face (Salam et al. (2012)) (cf. figure 2(a)). As we see from the figure, when the iris texture slides to the extreme left or the extreme right using simple translation in 2D, the appearance of the eye becomes unrealistic. In reality, the iris is a part of a spherical eyeball. As the eyeball rotates to extreme positions, the iris's appearance becomes elliptical rather than circular. Thus, modelling the iris in 2D is not sufficiently realistic and may cause problems in detection. This is why we propose to model the iris as a part of a 3D eyeball through the 3D MT-AAM (cf. figure 2(b)).

Modeling the interior of the eye as a sphere and rotating it under the skin surface, any gaze direction can be synthesized. Consequently, the iris motion is parameterized and realistically modelled. This requires the fusion of 2 AAMs, one for the iris texture and one for the surrounding skin (where a hole is put inside the eye).

We would like to point out that in this attempt of parametrizing the iris motion; we follow the same reasoning as [6]. They separate the head pose from the facial appearance by assigning a separate set of parameters for the head pose. In the same manner, we exclude the position of the iris in the eye from the appearance parameters of AAM by putting a hole in the place of the iris when making the eye model, and we assign a special pose vector for determining the iris location (this is explained in more details in the following sections).

We now present each of the models in sections 3.3.1 and 3.3.2, and their fusion in section 3.3.3, and we further explain how we model the iris as a part of an eyeball. In the rest of the paper, when we want to talk about something that is common between the 2D MT-AAM and the 3D

MT-AAM, we call it the MT-AAM. When we want speak about their differences, we used the pre-suffices 2D and 3D.

3.3.1. Modelling: building eye skin and iris models

Local eye skin AAM -- The presence of the iris which undergoes variability in scale, colour and position adds important variability to the eye (along with the eyelids, eyebrows, etc.). This variation is accounted by active appearance models when modelling the eye as a whole, which causes the number of appearance parameters to increase. For this reason and for the purpose of decorrelating the eye skin from the iris, we put holes inside the eyes in the place of the iris-sclera part.

To build this model, 22 landmarks that describe the whole eye area, including the eyebrows and the texture surrounding the eye, are used. Figure 4(a) is an illustration of the mean texture of the eye skin model showing the hole inside the right eye with the annotations to obtain this model.

Iris AAM -- In order to build this AAM, we need an iris-sclera texture that is capable of sliding under the eye skin. We construct our iris training database starting from the iris images of Colburn et al. [5] and Dobes et al. [8].

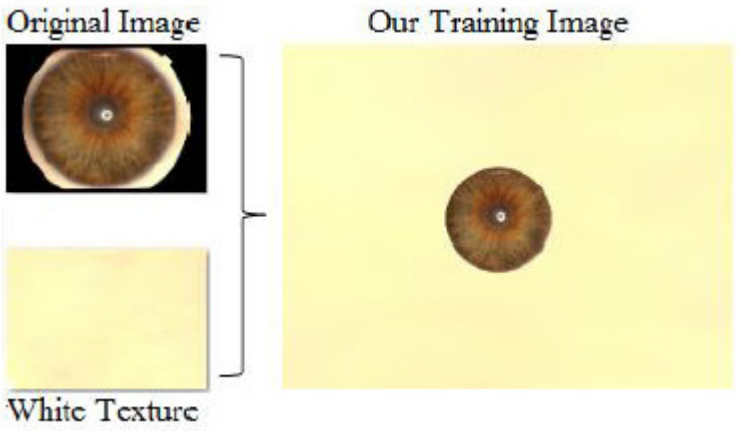


Figure 3. An example of a training iris image before and after processing
(remark that the original image is of a very high resolution)

We reprocess these images to obtain the iris part. We then merge it with a white texture. We make the sclera texture by cropping a small area from the sclera in the original iris images and resizing it. Cropping from the sclera in each image to reproduce the white texture, results in different textures which are not totally white. This permits to learn different white information in the training phase of the iris AAM, thus making the model capable of coping with the variation of the sclera colour from one person to another. The original iris images are of high resolution and they present unnecessary details in the iris texture. We resize these images and apply a circular averaging low pass filter to decrease the amount of information in the iris area. Figure 3 is an illustration of the construction of our iris training images.

For training these iris images we use a model of 13 landmarks of which 8 describe the shape of the iris in frontal view and 1 describes the approximate position of its centre; to learn the white texture around the iris, 4 additional landmarks forming a rectangular shape around the iris are

placed. Figure 4(b) is an illustration of the mean texture of the iris model with the corresponding annotations.

3.3.1. Searching: fusion of the iris and the eye skin models

Fusion of the eye skin model and the iris one is done in the searching phase. First we find the optimal parameters for the eye skin (using the eye skin model) in a prior step. We then use the found parameters to reconstruct the image describing the eye skin. The iris model rotates under it with the pose vector T^{iris} describing the iris position with respect to eye.

$$T^{iris} = [S^{iris}, \theta^{iris}_{hor}, \theta^{iris}_{ver}]$$

Where S_{iris} is the scale of the iris, θ^{iris}_{hor} is the horizontal rotation of the eyeball and θ^{iris}_{ver} is its vertical rotation.

As we see from the iris pose vector, modeling the iris as a part of a sphere instead of a plane permits to give the gaze angle directly. This shows another advantage of the 3D MT-AAM over the 2D MT-AAM of Salam et al. [27]. In the latter, the pose of the iris is in terms of vertical and horizontal translations. Thus, to compute the gaze angle, further calculations should be done.

To merge the eye skin and iris models, we simply replace the hole in the skin model (figure 4(a)) with the pixels of the iris model (figure 4(b)). After replacement, a problem of discontinuity between the two models arises. This is shown in figure 4(c). As we see, the resulting eye model seems unrealistic, especially at the borders of the eye skin model. In order to resolve this, we apply a circular averaging low pass filter on the skin and white parts while preserving the iris. It smoothes the discontinuity between the eyelid and the iris, and also reproduces the shadow effect of the eyelid on the iris. We remark that the filter is not applied on the iris since it is essential to preserve a good image quality of the iris in order to guarantee the localization. This is done using the mask shown in figure 4(d). The filter applied on all the pixels of the white area of the mask. We remark that some pixels of the perimeter of the iris are non-intentionally affected by the application of the filter using the mask. This is because the landmarks of the iris are not abundant (8 landmarks). Thus, it causes that some pixels of the perimeter of the iris to be included in the region where we apply the mask. However, this does not affect the detection since it only concerns few pixels. This results in the final model describing the eye region (figure 4(e)).

The detailed algorithm goes as follows:

1. Localize the eye using the eye skin model

2. From the webcam image, extract the texture of the eye (g_i) (Fig. 2(f)).

3. Using the optimal parameters found by the eye skin model, synthesize the eye skin (g_m^{eye}) (Fig. 2(a)).

4. Until the stop condition (number of iterations reached) do:

(a) Create the model texture of the iris (g_m^{iris}) based on the pose and the appearance parameters of the iris model (Fig. 2(b)).
i. Project the 2D shape of the iris on the sphere (Fig. 1(b)(i)).
(b) Rotate the eye and the sphere in 3D (Fig. 1(b)(ii)).
(c) Project the 3D iris on 2D (Fig. 1(b)(iii)).
(d) Map the iris texture on the rotated shape (cf. Fig. 2(b)). This will give the appearance of the iris when the person is looking to the extreme position.
(e) Merge the two textures g_m^{eye} and g_m^{iris} to obtain the texture g_m (Fig. 2(c)).
(f) Apply a low pass filter to get the final eye region model g_m (Fig. 2(e)). This low pass filter serves to smooth the boundary between the eye skin and iris model.
(g) Evaluate the error $E = g_m - g_i$ in the interior of the eye region (fig. 1(g) and fig. 1(h)).
(h) Tune the pose T^{gaze} and appearance C^{iris} of the iris model.

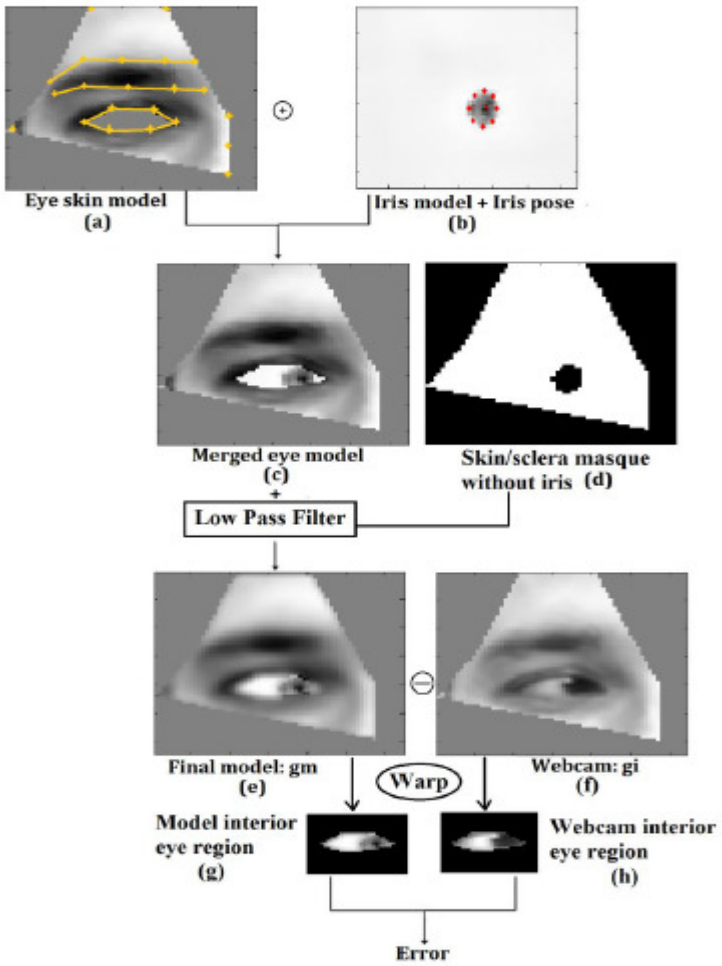


Figure 4. Error calculation at one iteration. At each iteration, the eye skin model is merged with the iris one to obtain the final eye model which is compared to the real eye to get the error.

Note that the evaluation of the error in computed in the interior of the eyes and not over the whole eye region. This is because the eyes skin texture is the optimal one, so it is unnecessary to include the skin texture in the error calculation. It will only add noise to the latter.

Iris radius – Since the radius of the iris changes from one iteration to another due to scale parameter. The radius of the sphere changes also. To calculate it, the radius of the iris is computed at each iteration and thus, the sphere radius is equal to the current iris radius divided by an approximate constant ratio between the radius of the iris and that of the sphere. The average diameter of the iris in the human eye is 12mm and the average eyeball diameter is 25mm. Thus the constant ratio between the two is equal to 0.48mm.

Constraints – Since the iris location and scale are constrained by the size of the eye, constraints are added in order to tighten up the search space in the searching phase.

2DMT-AAM: Constraints are based on anthropometric averages which give a very tight bound on the size of the iris relative to the size of the eye and on the fact that iris movements have limits imposed on them by the width of the eye. Iris average width is approximated at 1/1.6 the width of the eye.

3D MT-AAM: The horizontal rotation of the sphere is limited to +40° and -40° and the vertical rotation is limited to +10° and -10° which is found to give plausible projections of the 3D iris shape.

For both 2D and 3D MT-AAM, the scale is varied around an initial scale calculated using the width of the iris and that of the mean iris. The horizontal and vertical translation parameters cannot exceed half of the eye width and height respectively, taking the midpoint of the distance between the eye corners as the origin point.

4. MULTI-OBJECTIVE OPTIMIZATION

Since normally the two eyes have highly correlated appearance and motion, we use only one pose vector and one appearance vector to describe the pose and appearance of both irises. Technically, it should be sufficient to analyze the iris of one eye to obtain its position and appearance in both eyes. Yet, the information from both eyes can lead to a more robust system unless when the person commits large head movements around the vertical axis where one of the eyes can be partially or completely occluded. We achieve this using a multi-objective AAM framework. We follow an approach similar to that of [30] and [14]. Sattar and Seguier [30] propose the so called 2.5D multi-objective AAM for face alignment where several images from different cameras of the same subject are fitted simultaneously using a multi-objective Genetic Algorithm (GA) optimization.

Hu et al. [14] fit a 2D+3D AAM to multiple images of the same subject acquired at the same instance by optimizing the addition of the errors corresponding to these images, thus moving from multi-objective to single-objective optimization. We thus propose to apply a similar approach for the eyes. The idea is that we deal with the eyes as if it they were two separate images acquired at the same time: MT-AAM (see section 3.3) for iris localization is applied simultaneously on both eyes and at each iteration, the resulting errors are summed while multiplying each by a weighting factor that is a function of the head pose.

In this system, a single iris model is merged simultaneously with the left and the right eye skin models. Then the resulting models are overlaid on both the right and the left eyes from the camera to get the left and right errors. These are weighted according to the head orientation and summed to get one global error that is minimized using a Genetic Algorithm (GA) [29]. This error becomes:

$$E = \alpha E^{left} + \beta E^{right}$$

Where E^{left} and E^{right} are the errors corresponding to the left and right eyes respectively. α and β are the weighting factors. They are functions of the head rotation around the z-axis (R_{yaw}), evaluated just after the face detection, and they both follow a double logistic law:

$$\alpha(R_{yaw}) = \begin{cases} 0.5 & if \; -d \le R_{yaw} \le d \\ 0 & if \; -90 \le R_{yaw} \le -22 \\ 1 & if \; 22 \le R_{yaw} \le 90 \\ 0.5(1 + l(1 + \exp\frac{(-R_{yaw} - ld)^2}{\sigma^2})) & else \end{cases}$$

$$\beta(R_{yaw}) = 1 - \alpha$$

Where $l = sign(R_{yaw})$, σ is the steepness factor and d is the band such that the two functions α and β are equal to 0.5. σ is found empirically and d is chosen to be 7◦ such that for such a value we consider that the orientation of the head is negligible and that both eyes contribute equally. A head rotation of 22◦ is considered to be big enough to make one of the eyes appear more than the other in the image, and thus, exclusively taken into consideration in the detection of the iris.
In this way, the face orientation is taken into account by the relevant information from both eyes.

GA is a population-based algorithm that consists of iterated cycles of evaluation, selection, reproduction, and mutation. Each cycle is called a generation. The genes (iris pose and c parameters) are concatenated to form a chromosome. A group of chromosomes forms a population.

The initial population is generated randomly between the upper and the lower limits of the parameters with a uniform distribution. The fitness function is the penalized sum of the pixel errors of the left and the right eyes presented in equation 4.

We have chosen GA as an optimization scheme because of its good exploration ability (ability to span all of the search space). This is convenient for iris pose optimization because the iris could be anywhere inside the small eye region. Moreover, it is also convenient for tracking at low frame rate.

As a matter of fact, the fast saccadic eye movements are such that the iris can have a different position from one frame to another at a low frame rate. Consequently, GA would not fall into a local minimum in such case because of its exploration capability.

5. RESULTS AND DISCUSSION

We conduct four experiments: one to check the accuracy of the eye skin model and the dependence of the proposed model on the eye detection method, one to test the Multi-Objective AAM versus a Single-Objective AAM, one to compare the Multi-Texture AAM, 3D Multi-Texture AAM and a classical AAM for iris detection, and finally, one to compare the 3D-AAM to a state-of-the-art method.

Tests are conducted on the Pose Gaze (PG) database [1] and the UIm Head Pose and Gaze (UImHPG) database [35]. The first database was recorded using a simple monocular webcam. It contains 20 color video sequences of 10 subjects committing combinations of head pose and gaze. The second was recorded using a digital camera. It contains 20 persons with 9 vertical and horizontal gaze directions for each head pose of the person (range 0◦ to 90◦ in steps of 10 for the yaw angle and 0◦, 30◦, 60◦ azimuth and −20◦ and 20◦ elevation for the pitch angle).

5.1. Accuracy of the eye skin model

This section discusses two issues: first, the comparison between the performances of the eye skin model used in this paper (holes put in place of sclera-iris region) and that of a classical local eye model where the appearance of the interior of the eye is learned with that of the face skin; second, the dependence of the MT-AAM on the localization of the eyelids. The first issue serves at showing that putting holes instead of the sclera-iris increases the accuracy of the eyelids localization since it removes perturbations in the appearance due to gaze. On the other hand, the second has as objective to show how dependent the MT-AAM is of the eye localization method.

5.1.1. Eyelids localization: model with holes vs. model without holes

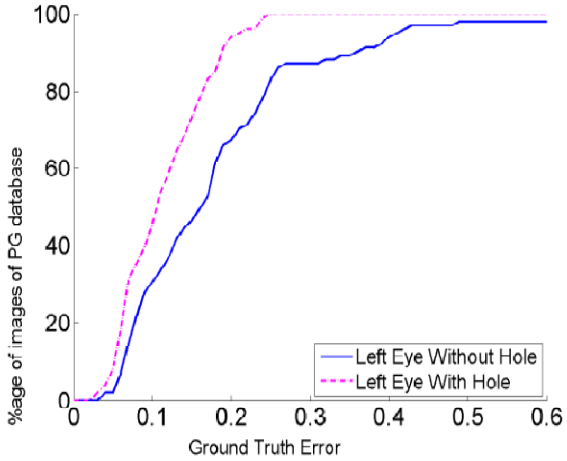
In order to compare the performance of the eye skin model where holes are put inside the eye to that where the interior of the eyes is kept, we plot the ground truth error of the eyelids versus the percentage of images in the database. The GTE is defined as follows:

$$GTE_{eyelid} = \frac{mean(\{d_i\}_{i=1:6})}{d_{eyes}}$$

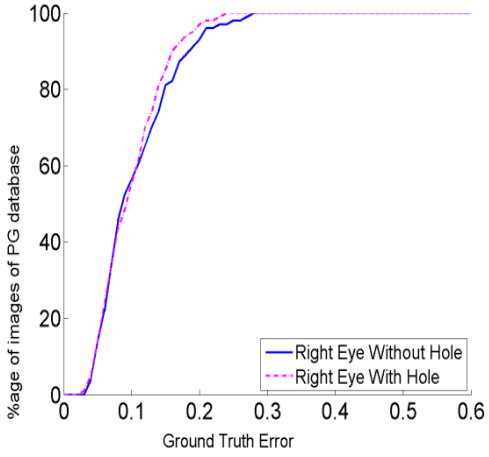
where $\{ d_i\}$ are the distances between the ground truth and each of the found points by the eye skin model. d_{eyes} is the distance between the eyes when the person is looking in front of him. For this test, both models where trained on 104 neutral images of the Bosphoros database [31]. The reason why we train on this database is that we want to make the test in generalization. Figure 5 shows the GTE_{eyelid} of both methods for both right and left eyes. Tests were done on 100 images of the PG database and 185 images of the UImHPG database. The figure shows that for the four cases (left and right eyes of both databases) we have a higher GTE curve in the case of an eyelid model with a hole inside the eye.

Figures 5(c) and 5(f) show qualitative results of both models on some images of the PG and UImHPG databases respectively. As we see from the figure, the eye skin model finds the good localization of the eyelid while the model without a hole does not. The reason is that the information inside the eye (colour of the iris and the different iris locations) influence the localization of the points of the eyelids when the interior of the eye is learned with the model. We

can see from these results how the model always follows the position of the iris and so it diverges. By deleting this information, we have succeeded to delete its disturbance and we are able to better localize the eyelids.

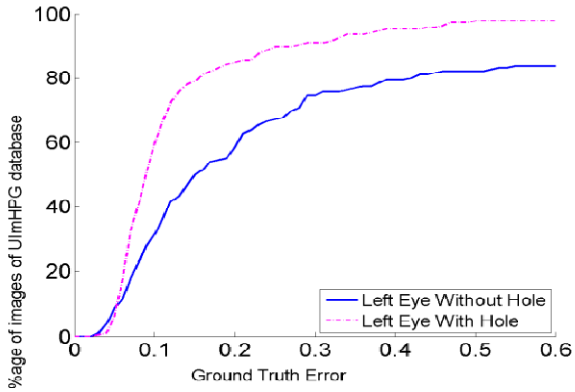


(a) PG : $GTE_{LeftEyelid}$

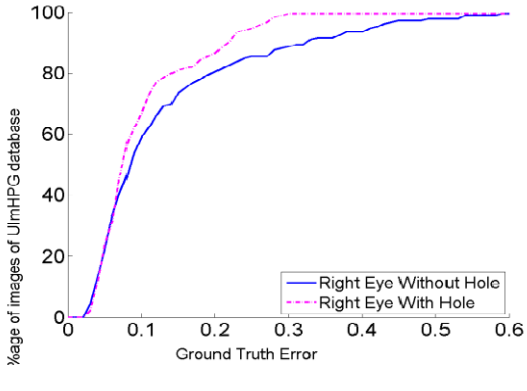


(b) PG : $GTE_{RightEyelid}$

(c) Comparison of results for the PG database: model without hole (left image) and that of the eye model with hole (right image)



(d) UImHPG : $GTE_{LeftEyelid}$

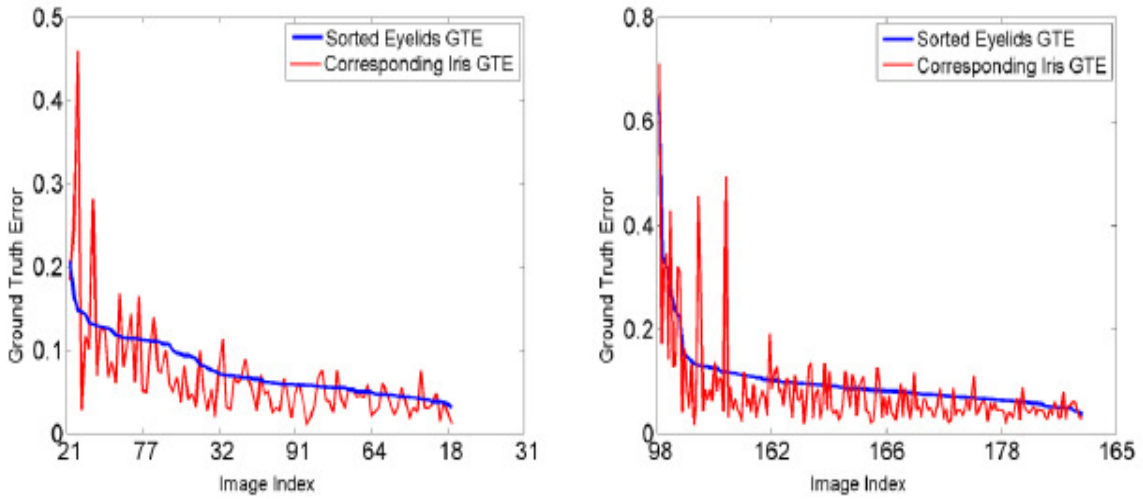


(e) UImHPG : $GTE_{LeftEyelid}$

(f) Comparison of results for the UImHPG database: model without hole (left image) and that of the eye model with hole (right image)

Figure 5.Comparison between the GTEeyelidwithhole and GTEeyelidwithouthole of the right and left eyelids for the PG and the UImHPG databases

5.1.2. Dependency of the MT-AAM on the eyelids localization



(a) PG database (b) UImHPG database

Figure 6. GTE_{eyelid} vs. GTE_{iris} sorted in descending order

To study the dependency of the proposed iris localization method (the MT-AAM) on the eyelids localization method, we plot the GTE2eyelids for each image sorted in decreasing order. We then sort the GTEiris according to the indices of the sorted images of the GTE2eyelids . The idea is to see how the iris GTE acts with the decrease of the GTE2eyelids . In other words, we can assume that if both errors decrease together then they are dependent from each other. Thus bad eyelid localization would result in bad gaze detection.

The GTE_{iris} is the mean of the distance (Euclidean distance) between ground truth (real location of iris centre) marked manually and the iris centre given by the gaze detection method normalized by the distance between the eyes. The GTE_{iris} is given by:

$$\begin{cases} \dfrac{mean(d_{left}, d_{right})}{d_{eyes}} & if\ -d \le R_{yaw} \le d \\ \dfrac{d_{right}}{d_{eyes}} & if\ -90 \le R_{yaw} \le -22 \\ \dfrac{d_{right}}{d_{eyes}} & if\ 22 \le R_{yaw} \le 90 \\ \dfrac{\alpha d_{left} + \beta d_{right}}{d_{eyes}} & else \end{cases}$$

Where dleft and dright are the Euclidean distances between the located eyes and the ground truth, deyes is the distance between the two eyes from a frontal face. R_{yaw} is the horizontal head pose, α and β are the weights calculated using the double logistic function, and d is the band such that α and β are equal to 0.5.

The $GTE_{2eyelids}$ is calculated from the GTE_{eyelid} of the right and the left eyes. Actually, according to the eye that was used in the detection of the iris, the corresponding GTE_{eyelid} is taken into account.

$$GTE_{2eyelids} = \begin{cases} mean(GTE_{lefteyelid} + GTE_{righteyelid}) & if\ -d \le R_{yaw} \le d \\ GTE_{righteyelid} & if\ -90^o \le R_{yaw} \le -22^o \\ GTE_{lefteyelid} & if\ 22^o \le R_{yaw} \le 90^o \\ \alpha GTE_{lefteyelid} + \beta GTE_{righteyelid} & else \end{cases}$$

R_{yaw} is the horizontal head pose, α and β are the weights calculated using the double logistic function, and $d = 7^0$ is the band such that α and β equal to 0.5

In figure 6, we plot the $GTE_{2eyelids}$ of the PG (figure 6(a)) and UImHPG (figure 6(b)) databases sorted in descending order vs. the GTE_{iris} . From this plot, we can see how the GTE of the iris decreases as the $GTE_{2eyelids}$ does. This confirms that if the localization of the eyelid was precise enough, the MT-AAM will be precise. As a conclusion, we can state that one of the drawbacks of our proposed method is its dependency on the eye localization method. Thus, we choose the eye skin model with a hole to locate the eyelids.

5.2. Multi-Objective AAM vs. Single-Objective AAM

In order to test the power of the Multi-Objective AAM (MOAAM), we compare it to a Single-Objective AAM (SOAAM). For this experiment we choose to test on the PG database and not on the UImHPG. Actually, the UImHPG database does not contain continuous head poses (the head poses are increments of 10) in contrary to the PG database. Consequently, we choose the latter

since the Multi-Objective method weights the errors according to the head pose. On the other hand, for this experiment, we use the Bosphoros database as a learning base for the eye skin model. The reason is that we want to do the test in generalization, and training on it gives accurate results for the eyelids. In MOAAM, both left and right eyes are given the same weight of 0.5 in the error calculation. In the MOAAM, the proposed double logistic function (cf. section 4) is used to evaluate the weights corresponding to the errors of each of the eyes.

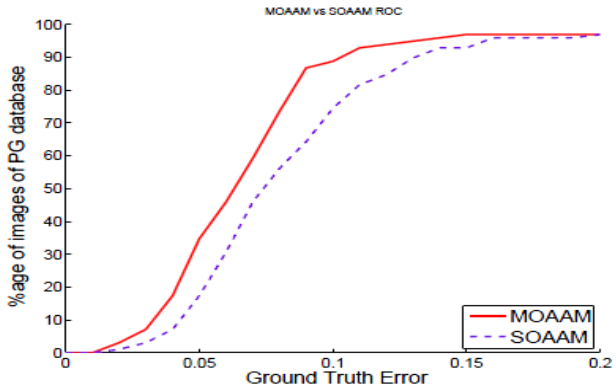
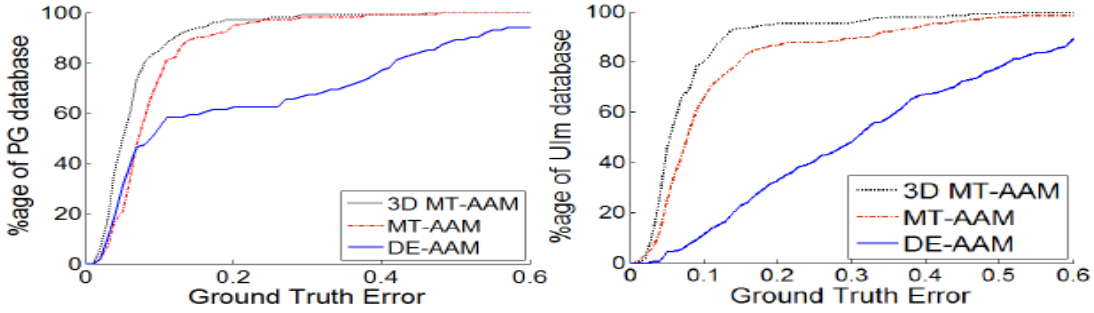


Figure 7. MOAAM vs. SOAAM

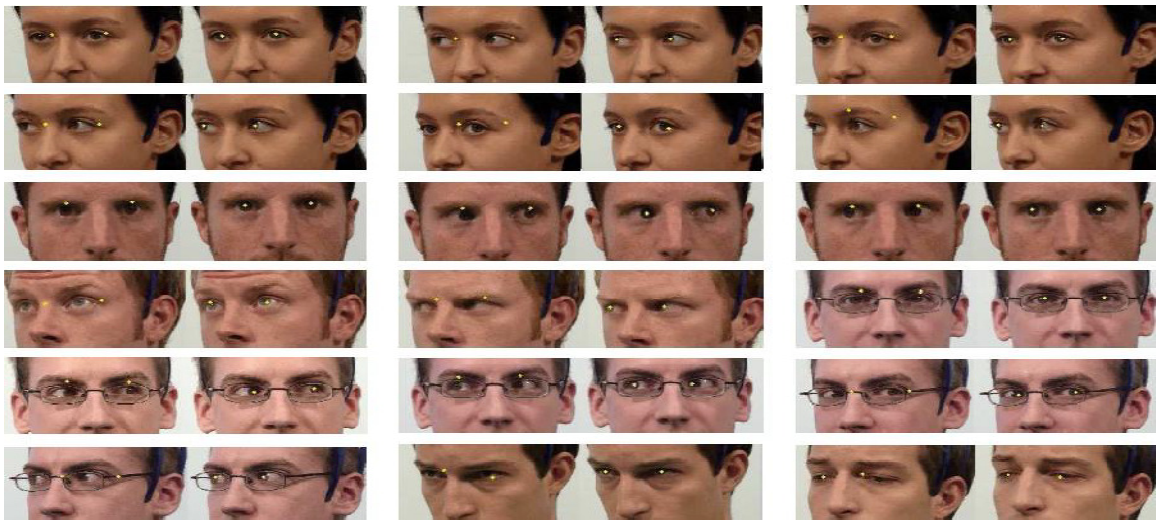
To eliminate the noise of pose detection and to have a fair comparison showing the strength of integrating the head pose in the calculation of the gaze, we use the ground truth of the pose instead of the results given by the global AAM. We remark that tests were done on subjects with a pose in the range of $[-21^0, -8^0]$ and $[8^0, 21^0]$ where the information of both eyes is taken into account by the double logistic function. Other poses where not taken into consideration for this experiment since it will not be fair for the SOAAM (for poses $\geq 22^0$ one eye is taken into account for MOAAM).

Figure 7 shows comparison between the GTE curves of the MOAAM and the SOAAM for the PG database. We can see that MOAAM improves iris localization with respect to SOAAM by 14.29% (iris is well detected for 88.78% of the images with an error level of 10% for MOAAM vs. 74.49% for SOAAM with the same error). As we see, integrating the head pose in the calculation of the final error of the MT-AAM improves iris localization.

5.3. 3D MT-AAM vs. 2D MT-AAM vs. classical AAM



(a) GTE of PG database (b) GTE of UImHPG database



(c) Comparison between the multi-texture approach (right columns) and the DE-AAM approach (left columns).

Figure 8. 3D MT-AAM vs. 2D MT-AAM vs. Double Eyes AAM

We compare the 3D MT-AAM to the 2D MT-AAM and a classical AAM, the 2.5D Double Eyes AAM (2.5D DE-AAM). In the following, we describe how the three methods were trained and on what images they were tested.

Models training – The 2.5D DE-AAM was built using a total of 28 landmarks. Each eye is annotated by 7 landmarks of which 1 landmark is for its centre. To take into consideration the texture surrounding the eyes landmarks at the bottom of each eyebrow were placed.

To train the model, we use 50 images of the PG database as a learning database. It consists of 10 persons with frontal head pose, each committing 5 gaze directions (1 to the extreme left, 1 to the extreme right, 1 to the front and 2 intermediate gaze directions).

For the 2D MT-AAM and the 3D MT-AAM, the same training database as that for the 2.5D DE-AAM was used to train the eye skin model. The iris AAM is trained using a group of 23 iris textures starting from the images of Dobes et al. [8] (see section 3.3.2).

Concerning the two MT-AAMs, as presented in section 3, to find the head orientation, a 2.5D global active appearance model was used. The model is trained on 104 neutral face images of the 3D Bosphoros database of Savran et al. [31]. A total of 83 landmarks are used, of which 78 are marked manually on the face and 5 landmarks on the forehead estimated automatically from the landmarks of the eyes.

Optimization – Concerning the 2D MT-AAM, a Genetic Algorithm (GA) followed by a gradient descent is used to optimize the iris appearance and pose parameters. Concerning the 3D MT-AAM, only a GA is used for optimization. Concerning the Classical AAM two consecutive Newton gradient descent algorithms are used. In the first one the learning of the relationship between the error and the displacement of the parameters is done offline during the training phase as proposed by Cootes et al. [6]. In the second, this relationship is learned online. The parameters from the first optimization scheme are entered into the second in order to refine the results.

Testing – The UIm testing database contains 185 images chosen randomly from the initial database. The PG testing database contains the same persons of the training database (mentioned in the paragraph concerning training) but with varying head poses and gaze directions. The number of images in it is 100 of which are chosen randomly from the initial database.

In figure 8, we compare the Ground Truth Error of the iris (GTE_{iris} presented in section 5.1.2) versus the percentage of aligned images in the two databases.

Figure 8(a) displays results obtained by testing on the Pose Gaze database. Figure 8(b) presents ones obtained on the UlmHPG database. Both figures contains 3 curves for both databases, a GTE_{iris} for the 3D MT-AAM, one for the 2D MT-AAM, and one for the 2.5D DE-AAM.

We do the test of figure 8(a) to prove that even when the testing database contains the same persons as the learning database for the DE-AAM, the multi-texture AAM overcomes the latter. For the DE-AAM, when the person is in frontal view, the model succeeds to localize the iris which is normal because the learning was done on such kind of images. However, when it comes to the different head poses present in the testing database, the DE- AAM will fail while the MT-AAM will not. This is due to the fact that in the DE-AAM it is necessary to increase the number of images in the learning database to increase the accuracy and to include variation in pose. Whereas, with the MT-AAM, the number of images in the learning database is sufficient to localize accurately the eyelids since the interior is removed and we have the same persons and then for the iris location, the general iris texture that is slid under the skin is able to localize it.

We do the test of figure 8(b) in order to test the generalization capabilities for the three models (test on the UImHPG database which is different from the training database).

As we see from figure 8(a), the 3D MT-AAM outperforms the 2D MTAAM which confirms the success of modeling the iris as a part of a sphere which is normal because it is more realistic. In addition, both the 2D and 3D MT-AAM outperform the 2.5D 2.5D DE-AAM. For instance, for an error less or equal to 10% of the inter-eyes distance, the 3D MT-AAM has detected the correct position of the iris on 85.15% of the images, the 2D MT-AAM has detected 74.26% whereas the 2.5D DE-AAM shows only 54.46% at the same error level.

Concerning figure 8(b) (generalization test), we have a good detection of 80% for the 3D MT-AAM versus 65.95 for the 2D MT-AAM and 11.35% for the 2.5D DE-AAM method for the same error level of 10%. This confirms first that the 3D MT-AAM is also better than the 2D MT-AAM in generalization.

Second, it shows that the MT-AAM model outperforms the classical one in generalization. Actually, as we separate the interior of the eye from the eye skin, we are able to parametrize the iris location through the iris pose vector (cf. section 3.3). Thus, it does not enter in the AAM appearance parameters anymore. Consequently, we are able to generalize to new different people while being less dependent from the learning base.

In addition, figure 8(c) shows qualitative results on the UImHPG database to compare between the MT-AAM and the DE-AAM. As the figure shows, MT-AAM succeeds to follow the gaze of persons with eye glasses and with different head poses, whereas the DE-AAM method does not. This assures the fact that with our method we are able to restrict the training base where there's no need to include people wearing eyeglasses in order to get reliable results on such subjects. The

reason behind this is that excluding the appearance of the interior of the eye for localizing the eyelids makes it easier for our model to cope with other kinds of appearance such as the existence of eyeglasses.

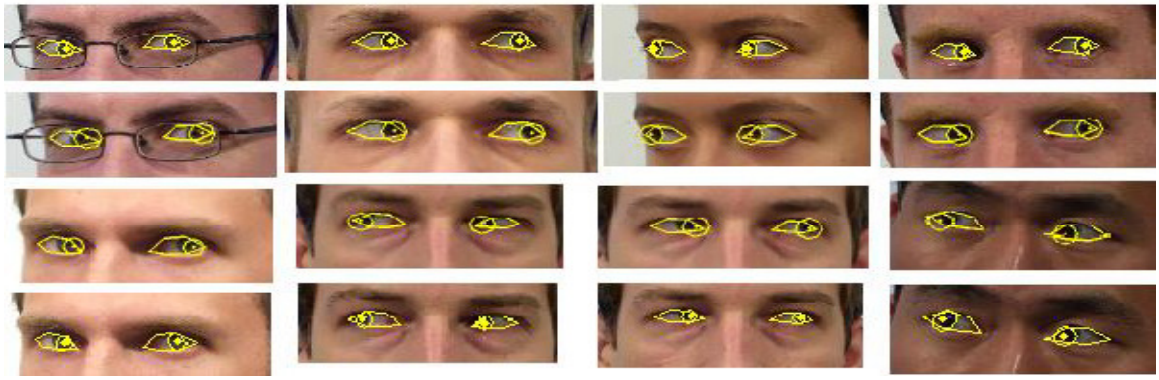


Figure 9. Qualitative comparison between the 3D MT-AAM (lower row of the same person) and the 2D MT-AAM (upper row of the same person)

On the other hand, figure 9 shows some qualitative results comparing the two models: the 3D MT-AAM and the 2D MT-AAM. It is obvious from the figure the superiority of the 3D AAM on the 2D AAM. As we see for extreme gaze directions the iris in the 3D MTAAM will take the shape of an ellipse representing realistically its appearance, however for the 2D MT-AAM, the iris shape is a circle that comes out of the eye in most of the time because it cannot take the real shape of the iris. As a conclusion, we can state that the 3D representation of the iris is more realistic than a 2D one and thus gives better results.

5.4. Comparison with a state-of-the-art method

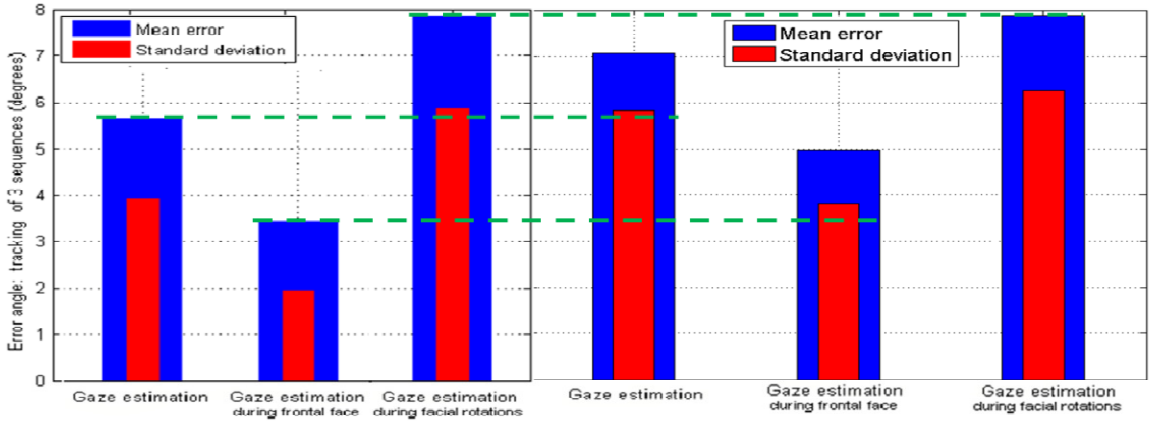


Figure 10. Comparison of the MT-AAM method (to the right) to that of Heyman et al. [12] (to the right)

This part compares the 3D MT-AAM method to the state-of-the-art method of Heyman et al. (2011). The comparison is conducted on 3 image sequences of the UImHPG database (heads 3, 12 and 16). The authors perform 2 trackings of the gaze of each of these sequences using 2 slightly different manual initializations of their head model. Head rotations were restricted to [−30◦, 30◦] and gaze rotations to [−40◦, 40◦]. On the other hand, we only conduct one gaze detection experiment per sequence since our method is fully automatic.

Figure 10 shows the average error angle of our method compared to that of Heyman et al. [12]. The red colour corresponds to the standard deviation and the blue colour corresponds to the mean of the gaze angle. We achieve a 7.07◦ gaze angle error with our method compared to a 5.64◦ with their method. We achieve their accuracy in the case of facial rotations. In the case of frontal face, their method is better than ours by about 1.5◦.

The authors have more accurate results because they manually initialize their first frame for tracking. Actually the 3D face model they use is adjusted manually on the first frame. Thus, the authors guarantee that they will not have high $GTE_{2eyelids}$ as we do (cf. figure 6; when we have high $GTE_{2eyelids}$, we have high GTE_{iris}, i.e. bad gaze detection). As a conclusion, since our method is fully automatic and we achieve similar results in the case facial rotations, we can say that our method is more robust and more appropriate for real time applications.

6. CONCLUSIONS

We have presented a new approach for the detection of the gaze angle. The proposed algorithm acts as a divide-and-conquer algorithm where the eye is divided into two parts: the eye skin and the iris-sclera texture. The latter is modelled as a part of a sphere. We localize the eye using an eye skin AAM where holes are put inside the eyes. Then, we parameterize the iris motion inside the eye by rotating the iris texture under the eye skin model. In order to be robust to head pose changes, a multi-objective framework is employed: The contribution of each eye to the final gaze direction is weighted depending on the detected face orientation. This multi-objective framework improves the gaze detection with respect to a single objective where both of eyes are equally integrated.

ACKNOWLEDGEMENTS

The authors would like to thank the project RePliCA, an ANR funded project for supporting this work.

REFERENCES

[1] Asteriadis, S., Soufleros, D., Karpouzis, K., Kollias, S., (2009). “A natural head pose and eye gaze dataset”, in: Proceedings of the International Workshop on Affective-Aware Virtual Agents and Social Robots, p. 1.

[2] Bacivarov, I., (2009). “Advances in the Modelling of Facial Sub-Regions and Facial Expressions using Active Appearance Techniques”. Ph.D. thesis. National University of Ireland.

[3] Carvalho, F.J., Tavares, J., (2007). “Eye detection using a deformable template in static images”, in: VIPimage-I ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing, pp. 209–215.

[4] Chen, Q., Kotani, K., Lee, F., Ohmi, T., (2009). “An accurate eye detection method using elliptical separability filter and combined features”. International Journal of Computer Science and Network Security (IJCSNS).

[5] Colburn, A., Cohen, M.F., Drucker, S., (2000). “The role of eye gaze in avatar mediated conversational interfaces”. Microsoft Research Report 81, 2000.

[6] Cootes, T., Edwards, G., Taylor, C., (1998). “Active appearance models”, in: IEEE European Conference on Computer Vision (ECCV ’98), p. 484.

[7] Cootes, T.F., Edwards, G.J., Taylor, C.J., (2001). “Active appearance models”. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 681.

[8] Dobes, M., Machala, L., Tichavskı, P., Pospisil, J., (2004). "Human eye iris recognition using the mutual information". Optik-International Journal for Light and Electron Optics 115, 399–404.
[9] Grauman, K., Betke, M., Gips, J., Bradski, G.R., (2001). "Communication via eye blinks-detection and duration analysis in real time", in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, pp. I–1010.
[10] Hansen, D.W., Ji, Q., (2010). "In the eye of the beholder: A survey of models for eyes and gaze". Pattern Analysis and Machine Intelligence, IEEE Transactions on 32, 478–500.
[11] Hansen, J.P., Augustin, J.S., Skovsgaard, H., (2011). "Gaze interaction from bed", in: Proceedings of the 1st Conference on Novel Gaze-Controlled Applications, p. 11.
[12] Heyman, T., Spruyt, V., Ledda, A., (2011). "3d face tracking and gaze estimation using a monocular camera", in: Proceedings of the 2nd International Conference on Positioning and Context-Awareness, pp. 1–6.
[13] Hillman, P., Hannah, J., Grant, P., (2003). "Global fitting of a facial model to facial features for model-based video coding", in: Image and Signal Processing and Analysis, 2003. ISPA 2003. Proceedings of the 3rd International Symposium on, pp. 359–364.
[14] Hu, C., Xiao, J., Matthews, I., Baker, S., Cohn, J., Kanade, T., (2004). "Fitting a single active appearance model simultaneously to multiple images", in: British Machine Vision Conference, p. 10.
[15] Huang, J., Wechsler, H., (1999). "Eye detection using optimal wavelet packets and radial basis functions (rbfs)". International Journal of Pattern Recognition and Artificial Intelligence 13, 1009–1025.
[16] Isokoski, P., Joos, M., Spakov, O., Martin, B., (2009). "Gaze controlled games". Universal Access in the Information Society 8, 323–337.
[17] Ivan, P., (2007). "Active appearance models for gaze estimation". Ph.D. thesis. Vrije University. Amsterdam.
[18] Kaminski, J., Knaan, D., Shavit, A., (2009). "Single image face orientation and gaze detection". Machine Vision and Applications 21, 85–98.
[19] Van der Kamp, J., Sundstedt, V., (2011). "Gaze and voice controlled drawing", in: Novel Gaze-Controlled Applications (NGCA), p. 9.
[20] Khilari, R., (2010). "Iris tracking and blink detection for human-computer interaction using a low resolution webcam", in: Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing, New York, NY, USA. pp. 456–463.
[21] Koesling, H., Kenny, A., Finke, A., Ritter, H., McLoone, S., Ward, T. (2011). "Towards intelligent user interfaces: anticipating actions in computer games", in: Proceedings of the 1st Conference on Novel Gaze-Controlled Applications, p. 4.
[22 Moriyama, T., Kanade, T., Xiao, J., Cohn, J.F., (2006). "Meticulously detailed eye region model and its application to analysis of facial images". IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 738–752.
[23] Orozco, J., Roca, F., Gonz`alez, J., (2009). "Real-time gaze tracking with appearance-based models". Machine Vision and Applications 20, 353–364.
[24] Qiang, J., Xiaojie, Y., (2002). "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance". Real-Time Imaging 8, 357–377.
[25] Reale, M.J., Canavan, S., Yin, L., Hu, K., Hung, T., (2011). "A Multi-Gesture interaction system using a 3-D iris disk model for gaze estimation and an active appearance model for 3-D hand pointing". IEEE Transactions on Multimedia 13, 474–486.
[26] Ryan, W.J., Woodard, D.L., Duchowski, A.T., Birchfield, S.T., (2008). "Adapting starburst for elliptical iris segmentation, in: IEEE second international conference on biometrics : Theory, Applications and Systems", pp. 1–7.
[27] Salam, H., Seguier, R., Stoiber, N., (2012). "A multi-texture approach for estimating iris positions in the eye using 2.5d active appearance models", in: Proceedings of the IEEE 2012 International Conference on Image Processing, p. accepted.
[28] Sattar, A., Aidarous, Y., Gallou, S.L., Seguier, R., (2007). "Face alignment by 2.5d active appearance model optimized by simplex", in: International Conference on Computer Vision Systems (ICVS), pp. 1–10.

[29] Sattar, A., Aidarous, Y., Seguier, R., (2008). "Gagm-aam: a genetic optimization with gaussian mixtures for active appearance models", in: IEEE International Conference on Image Processing (ICIP'08), pp. 3220–3223.
[30] Sattar, A., Seguier, R., (2010). "Facial feature extraction using hybrid ggeneticsimplex optimization in multi-objective active appearance model", in: Fifth International Conference on Digital Information Management (ICDIM), pp. 152–158.
[31] Savran, A., Alyüz, N., Dibeklioglu, H., C¸ eliktutan, O., Gökberk, B., Sankur, B., Akarun, L., (2008). "Bosphorus database for 3D face analysis", in: Proceedings of the First COST 2101 Workshop on Biometrics and Identity Management (BIOD), pp. 47–56.
[32] Stiefelhagen, R., Yang, J., (1997). "Gaze tracking for multimodal human-computer interaction", in: icassp, p. 2617.
[33] Valenti, R., Sebe, N., Gevers, T., (2012). "Using geometric properties of topo- graphic manifold to detect and track eyes for human-computer interaction". IEEE Transactions on Image Processing 21, 802–815.
[34] Valenti, R. ans Gevers, T., (2008). « Accurate eye center location and tracking using isophote curvature", in: CVPR, pp. 1–8.
[35] Weidenbacher, U., Layher, G., Strauss, P.M., Neumann, H., (2007). "A comprehensive head pose and gaze database", in: 3rd IET International Conference on Intelligent Environments (IE 07), pp. 455–458.

AUTHORS

Hanan SALAM received the B.Eng in Electrical and Computer engineering from the Lebanese University, Lebanon, and the M.Eng in Signal and Image processing from the Ecole Centrale, Nantes, France in 2010. She is currently working to get her PHD degree at the SCEE (Communication and Electronic Embedded Systems) lab of Supélec, Rennes, France. Her research interests include face analysis and eye gaze detection for Human-Machine Interface.

Renaud Séguier received the PhD degrees in Signal Processing, Image, Radar in 1995 and the HDR (Habilitation Diriger des Recherches) in 2012 from the University of Rennes I. He worked one year in Philips R&D department on numerical TV and Mpeg2 transport-stream. He joined SCEE Communication and Electronic Embedded Systems) lab of Suplec in 1997 since when he is Assistant Professor and now Professor in Image Processing, Artificial Life and Numerical Implementation. His current research focuses on face analysis and synthesis for Human-Machine Interface.

Nicolas Stoiber graduated from the engineering school Supelec in France 2005. He then obtained a Master of Science in Information and Multimedia Technology at the Technische Universitt Mnchen through a double degree in 2007. In 2010, he completed a PhD in the field of facial expression analysis and realistic facial animation synthesis. He then joined the founders Dynamixyz. He has since then been leading the company R&D work on image analysis, and animation and human facial expressions modeling.