Eyes and Eyebrows Detection for Performance Driven Animation

Juliano L. Moreira, Adriana Braun, Soraia R. Musse
Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS
Graduate Programme in Computer Science
Virtual Human Laboratory -www.inf.pucrs.br/∼vhlab
Porto Alegre, Brazil

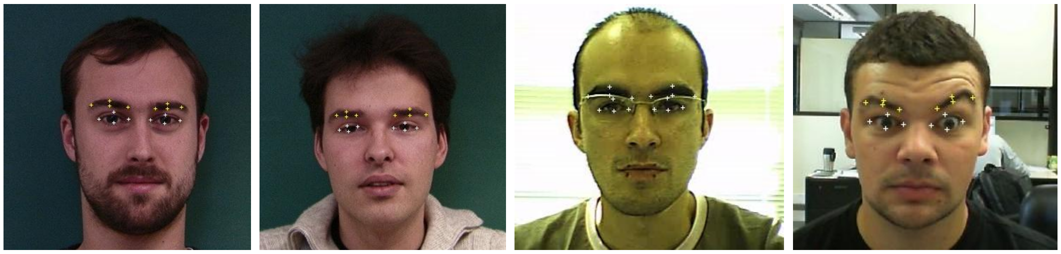


Figure 1. Detection of feature points in real people eyes and eyebrows.

***Abstract*—Real-time localization of eyes and eyebrows in video sequences can find several applications in nowadays. However, this task is still a challenge, mainly if we consider images of low quality as those achieved by web cams. This paper describes a methodology to accurately detect meaningful feature points in eyes and eyebrows of people in frontal pose, in images captured by web cams. Also it discusses the mapping of the movement of the eyes of a real person to an avatar, in order to provide Performance Driven Animation (PDA). Furthermore, this paper presents obtained results and the limitations of the proposed methodology.**

***Keywords*-Facial features detection; computer vision; performance driven animation.**

I. Introduction

The correct localization in real-time of eyes and eyebrows in video sequences can find several applications such as human computer interfaces, characters animation, face recognition and face expression analysis, among others. A very important device for acquisition of images concerning the scalability of applications is certainly the web cam. Such device has become quite common in current days and the number of softwares that use web cams have increased. However, the quality of acquired images is normally not high which certainly offers a challenge for some of web cam applications. In the case of our work, mainly focused on facial components detection using simple web cams, another challenge is certainly the variability of filmed people, their background as well as illumination conditions.

As a step toward the solution of this problem, we propose a methodology to detect eyes and eyebrows features in face images, captured by web cams. This methodology is based on anthropometric measures and color space manipulation of face images of people in nearly frontal pose. This last assumption is claimed to be reasonable since people normally stay in frontal pose when they are using web cam applications on their desktops or notebooks.

As the main application of the presented methodology, we propose to code the detected information into the MPEG4 standard [1] by providing accurate data for animation of 3D characters. Indeed, according to Tang [2], there are 3 different ways to drive an avatar: text-driven, speech-driven and performance-driven animation (PDA). Respectively, they are related with text-to-speech engine which converts from text to visemes; interfaces which transform acoustic features in visual parameters and finally, the last one which animates a face model according to live video input. This paper describes computer vision algorithms to detect eyes information which are used to animate in real-time avatars based on live video of real faces.

The MPEG4 standard [1] for facial animation requires the displacement of feature points in the face, measured in terms of *Feature Animation Parameters Units (FAPUs)*, which are related with measure unities for the faces. Figure 2 illustrates the FAPUS defined for the eyes regions. MPEG4 requires 4 feature points for each eye and 3 feature points for each eyebrow. So, given the facial measures of a real person (in FAPUs), we can map the displacement of each feature point to the avatar, if we have the measures of FAPUS of the avatar's face. Section IV will discuss more about this mapping process.

In Section II some existing work in literature are presented once they provide approaches to detect facial feature points. However, some of them are more focused on find high level patterns such as open, closed, or partially closed eyes. Others

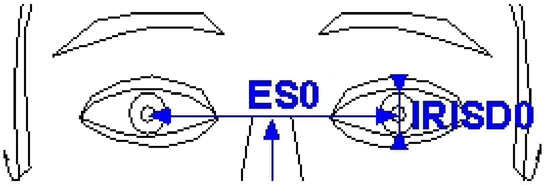


Figure 2. MPEG4 Face Parameter Units for the eyes.

are focused on locating the facial features, i.e, establish a rough position for the eyes and eyebrows, since these characteristics can be perceived. In spite of good results achieved by related work, we believed some of them can not be applied in PDA´s applications. The main reason is because we have constraints of accuracy and time coherency, since features points are directly used to control avatar´s animation. Also, computation time of methods is a relevant requirement of our application.

The main contribution of this paper is to provide a model of eyes and eyebrows components detection explicitly focused on PDA´s application, which should work using simple capture devices, e.g. web cam in real-time. Obtained results discuss the quality of detection algorithms, if compared with other approaches available on literature, and also present examples of an animation driven by such captured data.

The remaining of this paper is organized as follows: In Section III we expose the proposed methodology, while Section IV discusses the PDA process. Finally, Section V discusses the obtained results and some final remarks will be addressed in Section VI.

II. Related Work

The first studies on Performance Driven Animation of virtual faces were concerned to track the movement of facial components (mouth, eyes, eyebrows, among others) of people wearing markers on their faces. In this research field it is possible to achieve a very precise level of detail. One example is the work of [3] who proposes a scheme for the acquisition process of markers on the face. This type of approach is currently used to generate pre-recorded animations of virtual characters for cinema and games. In such cases accuracy is important and real-time performance of animation is not necessary, since animations should be played after acquisition and processing. However, the advent of new computer vision techniques in recent years allowed the researchers to focus on finding and tracking face components of regular users, without markers. It is currently an important challenge in this research field, i.e. the absence of complex apparatus for image acquisition. In the most ambitious scenario, the idea is to detect the face components and track their movements with high precision in images sequences acquired by simple web cams in real time. In our case, we are interested about an adequate precision in order to animate coherently the eyes regions of a virtual character.

A very common approach to this end is the Active Shape Models (ASM - [4]) and the Active Appearance Models (AAM - [5]) which main focus is to perform tracking of face components. These approaches assume the existence of a previous method for feature points detection in order to build a Point Distribution Model (PDM) and a training step in order to learn the allowable ways to deform the face. Among all authors who use such techniques we can highlight the work of Yuen et al. [6], Tong et al.[7], Ari et. al. [8] and Haj et al. [9]. Milborrow and Nicolls [10] proposed an extension of ASM in order to specialize it to detect facial features.

Other approaches use Facial Animation Tables (FATs) in order to convert the data acquired by video in Facial Animation Parameters, established in the work of Ekman [11]. Besides the success of these FATs in performing the conversion from low level information coming from computer vision to high level information like “the eyes are closed”, they do not allow the personalization of the movement. This strategy was used, for example, by Tang and Huang [12].

Another very known approach consists in analyzing the gradients of a gray scale image in order to establish an “Optical Flow”, i.e, the flow of pixel intensities through different frames. This method was used in the work of He and colleagues [13] for facial features tracking. Achieved results are very good, however computational time can impact the application on PDAs. Su and Huang [14] used Particle Filtering for multiple facial feature tracking. Other technique also used is Deformable Templates [15]. The advantage of such technique is its accuracy in comparison with others, but it is very time consuming if the templates are initialized far from their final positions.

Cristinacce and Cootes [16] propose a method which uses a set of feature templates and a shape constrained search technique, as well as a statistical shape model. Given the new feature locations, templates are updated using a nearest neighbour approach to select likely feature templates from the training set. Cristinacce [16] method was used in this paper as comparative references, helping evaluate our results. In addition, Milborrow [10] implementation of Active Shape Models was also used, as can be seen in Section V. The main reasons to use Cristinacce [16] and Milborrow [10] work as comparative with our model are the relevance of proposed methods, and the available tools, which allows to use such methods for evaluation.

Besides the very good results achieved in mentioned papers, we can observe that the compromise with accuracy × computational time, inherent in our main application is not always the main problem dealt with in other models. Also, some of cited authors are more interested about to provide a method to detect high level events, than robustly find feature points as a function of time, e.g. if a person is blinking, smiling and so on. Yet, we are interested in detecting facial components in real time, by using simple web cams, which are not requirements for some other applications.

Next Section shows our model developed in order to extract information about eyes and eyebrows of people in near frontal view on video sequences, acquired by web cams.

III. Model Description

The methodology proposed in this work follows a specific pipeline:

1) Face detection
2) Computation of Regions of Interest (ROIs)
3) Color Channel selection
4) Exponential Transform
5) Image Binarization
6) Blobs detection
7) Blobs selection
8) Eyelids, eye corners and eyebrows extreme points localization
9) Data filtering

The first step consists of detecting a face in the video scene. To this end we use the method proposed by Viola and Jones [17], which is a well known algorithm based on Haar classifiers for finding objects in images. In our experiments, we used a dataset of faces of people from different ethnics in frontal pose in the training phase. This algorithm returns a rectangle where the detected face is included. So we can work with a point corresponding to the center of the face (x_c, y_c) and two parameters (height - h_f and width - w_f) for the size of the face.

Given this starting point, we used anthropometric measures in order to define 4 Regions of Interest (ROIs) that probably contain the left and right eyes and the left and right eyebrows. To define the left eye ROI, we crop a rectangular portion of the input image, obtaining R_l (Equation 1) and symmetrically R_r.

$$R_l = \begin{pmatrix} I_{(x_c-w_e,y_c-h_e)} & \cdots & I_{(x_c,y_c-h_e)} \\ \vdots & \ddots & \vdots \\ I_{(x_c-w_e,y_c)} & \cdots & I_{(x_c,y_c)} \end{pmatrix}. \qquad (1)$$

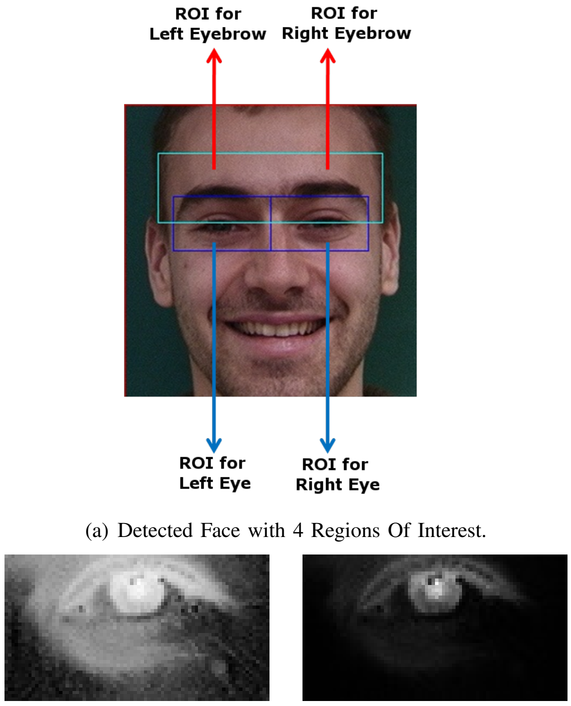
In this Equation, w_e is the width of the eye region, h_e is the height of the eye region while I is the input image. To define the eyebrows ROIs, we consider $y'_c = y_c + \frac{h_e}{2}$. Equation 2 presents the definition of left eyebrow region Rb_l:

$$Rb_l = \begin{pmatrix} I_{(x_c-w_e,y'_c-h_e)} & \cdots & I_{(x_c,y'_c-h_e)} \\ \vdots & \ddots & \vdots \\ I_{(x_c-w_e,y'_c)} & \cdots & I_{(x_c,y'_c)} \end{pmatrix}. \qquad (2)$$

Indeed, ROIs are computed based on the face dimensions and anthropometric based parameters α and β, which describe face proportions, as it can be seen in Equation 3 and illustrated in Figure 3-a.

$$w_e = \alpha w_f, \quad h_e = \beta h_f. \qquad (3)$$

In our experiments, we consider $\alpha = \frac{2}{5}$ and $\beta = \frac{1}{6}$ for the eyes and $\alpha = \frac{3}{7}$ and $\beta = \frac{1}{5}$ for the eyebrows, because these are the smallest values that make all eyes and eyebrows in tested databases are included in the respective ROIs. Figure 3-a shows an example of results achieved in the first 2 steps in the pipeline.



(a) Detected Face with 4 Regions Of Interest.

(b) ROI in inverted red.

(c) Exponential Operator applied in image (b).

Figure 3. (a) describes the step 2, while (b) and (c) shows the step 3 and 4 of the pipeline.

The next 4 phases (3 - 6) of the method are dedicated to segment the iris, eyelids and eyebrows in their respective ROIs. To this end we choose to increase the contrast in the images, highlighting iris, eyelids and eyebrows. Our hypothesis is that we should highlight non-skin pixels. It is well known in literature [18] that skin pixels have a high level of red intensity in RGB color system. Also, concerning problems caused by saturation of light, clear regions in the image should present high levels of red. We decided to investigate the red channel in order to verify if this could be a good color space for eyes and eyebrows segmentation. Our hypothesis is that iris, eyelids contours and eyebrows have low red intensities if compared to skin pixels. Figure 3-b shows the image of a ROI in complementary red channel (CR), i.e for each pixel we computed 255 minus the red channel value of the pixel. It is only performed in order to enhance the human perception of non-skin pixels. Other color spaces were tested as it can be seen in Table I in the Section V, where we justify our decision concerning chosen color space.

Step 4 consists in emphasizing the difference among the intensity of pixels in the CR image. To this end, we chose the exponential operator in order to enhance the difference among pixels' intensity [19]. The exponential operator consists in computing the following transformation to each pixel of the input image $I = CR$:

$$I'_{i,j} = exp\left[I_{i,j}.k\right], \tag{4}$$

where k adopts the value of $\frac{ln255}{255}$, which keeps the resulting pixels intensity values in the expected interval $[0, 255]$. The results of this transformation can be seen in Figure 3-c.

Since we highlighted non-skin pixels, the 5th step consists of eyes and eyelids segmentation in the images. It is performed by binarizing the image I' choosing the pixels which intensities represent outliers in the distribution. So, the binary image Bw, where outliers pixels are white and the remaining are black is given by:

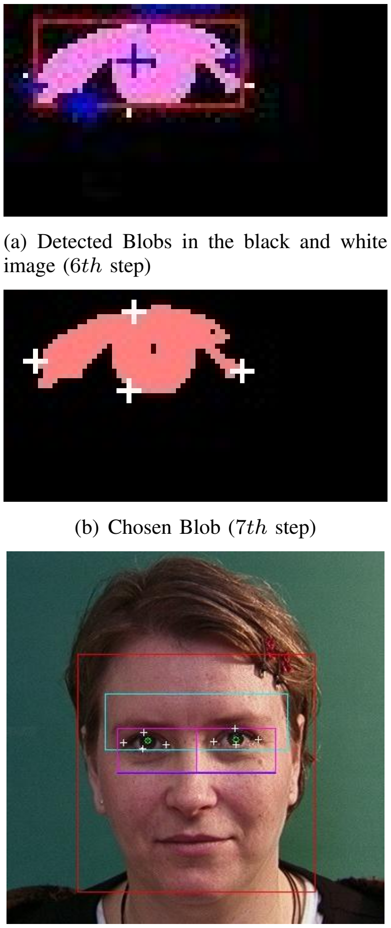
$$Bw = \begin{cases} 1, & \text{if } I' > (\bar{I}' + Z\sigma) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $\bar{I}'$ and σ are the average and the standard deviation of pixels' intensity of I' respectively, and Z is a score that controls the amount of pixels that will be selected to belong to the binary image Bw. The larger the value of Z, the larger the number of pixels to be selected. This parameter must be calibrated in order to achieve the best results. In our experiments, an optimum value empirically determined for Z is 0.9. At the end of this step we have an image like the one showed in Figure 4-a.

It is important to note that in the case of eyebrows detection, we exclude the regions belonging to the eye's blobs from the ROIs before performing the binarization step. To this end, we use a mask that defines the search region for eyebrows (Figure 5)-b. This avoids to consider eyelids and iris as eyebrows.

The 6th step consists in finding the connected elements in each Bw image which represent possible candidates for the eyes and eyebrows. Each 8-connected "blob" of pixels is then analised in order to eliminate clusters that could not represent the searched features (7th step - Figure 4-b). First of all, the blob representing the features of interest must have at least 10% of ROI's area. In eyebrows' case, if more than one blob pass by the area criteria, we select the one which centroid is closer to the center of the ROI. For the eyes' region, the chosen blob will be the one which centroid is lower in the image. Figure 6 illustrates this process.

In step 8, once we found the blobs, we finally determine the location of feature points, choosing the leftmost and the rightmost points within the blobs. These two points determine a line segment l_1, as illustrated in Figure 7. Then we define the line normal to l_1 (l_2) passing on l_1 center, and determine that the upper and lower feature points for each eye will be the last pixel that belongs to the blob and to l_2 simultaneously.

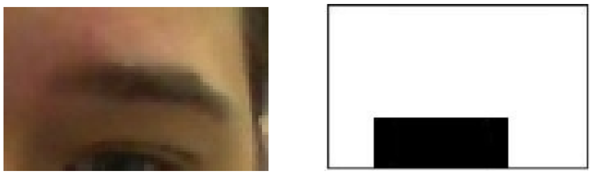


(a) Detected Blobs in the black and white image ($6th$ step)

(b) Chosen Blob ($7th$ step)

(c) Feature points Located

Figure 4. Steps 6 to 8, showing the segmentation of an eye and the detection of feature points.



(a)

(b)

Figure 5. Example of eyebrow's ROI and its correspondent mask.

A similar process occurs in the eyebrows: we find the leftmost and the rightmost points within the eyebrow's blob and consider them as extreme points from the eyebrow. Afterwards, two segments are computed like performed for the eyes (l_1 and l_2). However, one of extreme points from l_2 is not considered since we are only interested in 3 points for each eyebrow. Indeed, inferior point of l_2 is not considered

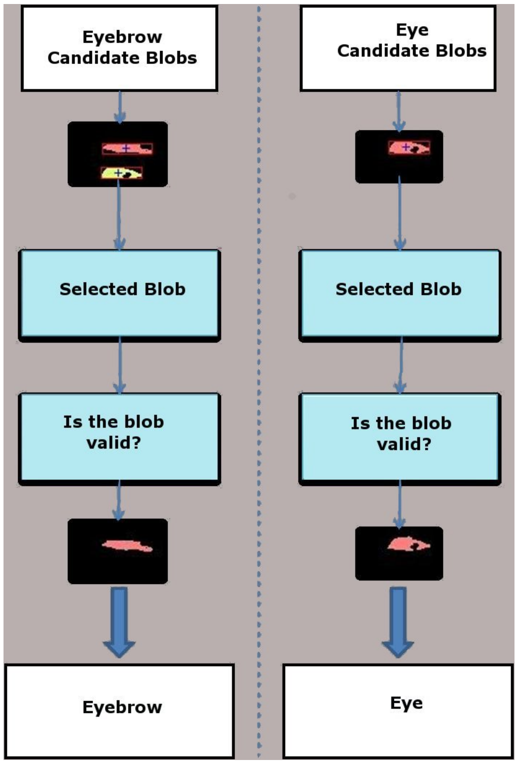


Figure 6. Sequence of images showing the process of blobs selection.

according to MPEG4 standard [1].

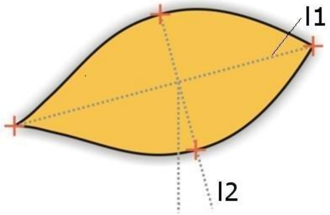


Figure 7. Illustration of Eyes feature points.

The 9th step only makes sense in image sequences with temporal coherence. Due to small variations from one frame to another in video sequences, the detection may present a instability of some pixels. In order to reduce this instability, we filter the feature point position $X = (x, y)$ using a linear filter in the current feature point location, considering the feature point positions in the n previous frames, according to the Equation 6:

$$X_t = \sum_{i=1}^{n} b_i X_{t-i}, \tag{6}$$

where b is an array of weights. In our implementation we used $n = 3$ and $b = (0.5, 0.3, 0.2)$.

At the end of these 9 steps we will have 3 feature points for each eyebrow and 4 feature points for each eye, as illustrated in Figure 1. We can perceive that 4 points for each eyebrow are displayed, however, we only consider 3 of them to animate avatars, as previously explained.

IV. Avatar Animation

Once we have found the specified feature points (4 for each eye and 3 for each eyebrow), their displacements as a function of time must be mapped to the avatar in the virtual world. We define a reference frame in the video and computed a mapping to the reference frame of 3D world. Since we do not have information about the three-dimensional form of the user, just the projection on the video frame, we just mapped the bi-dimensional displacements in image, without considering perspective transformation, since the user kept his or her face in a constant distance from the camera in mainly frontal view.

For scale variation, we considered the Face Parameter Units proposed on MPEG4 standard, showed in Figure 2. It was requested to the users in the experiments to keep their faces stopped for one second at the beginning of the recording. The first 20 frames was then used as training source in order to establish a reference frame. In this training phase, we computed the ESO and IRISDO Facial Animation Units represented in Figure 2. We also establish the average position of each feature point, keeping it in a set called $\mu = \{(\bar{x}_1, \bar{y}_1), (\bar{x}_2, \bar{y}_2), ..., (\bar{x}_{14}, \bar{y}_{14})\}$.

The Facial Animation Parameters (FAPs) we send to Facial Animation Module, consist in displacements, in each frame, of the feature points from μ and are computed as follows:

$$Fap_i = \frac{\mu_i - X_i}{FAPU_{real}}.FAPU_{avatar}, \tag{7}$$

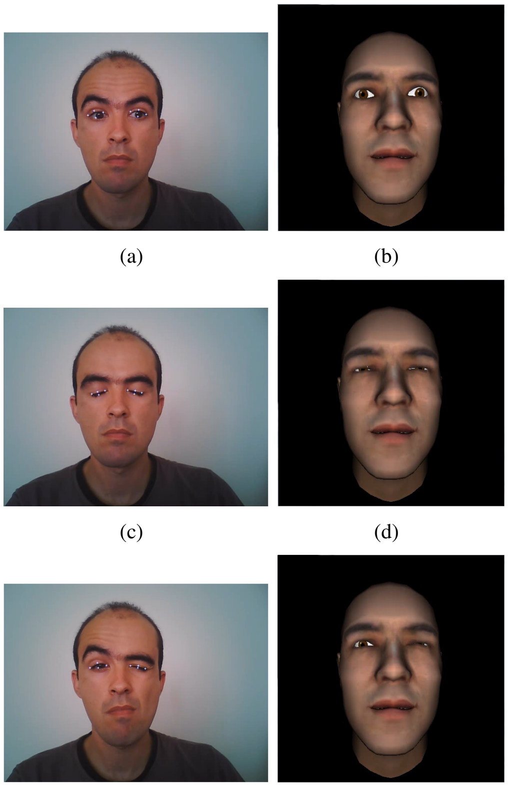
where $FAPU_{real}$ is the Facial Animation Units measured in training phase and $FAPU_{avatar}$ is the Facial Animation Units of the 3D model of the avatar. The arrows in Figure 2 give the motion direction of each Fap_i.

Figure 8 shows snapshots of animation results.

In Figure 8-a, 8-c and 8-e, we can see 3 frames of a video where eyes regions are detected through 4 feature points. In Figure 8-b, 8-d and 8-f, we can observe the result of avatar´s animation considering detected feature points. At the moment, our animation model is not working for eyebrows animation, mainly due to the fact the eyebrows are still textures and not yet geometrically defined in 3D faces. This is some work we should do in next steps, in the context of facial animation. Important to highlight that in this paper we did not present facial animation details, since it is not the main scope of this work.

V. Results

The most straightforward way to verify the accuracy of the method is to measure the Euclidean distance d of the computed points (x, y) in relation to the real points in



(a) (b)

(c) (d)

(e) (f)

Figure 8. Frames showing the mapping from the real user to the avatar.

image (x_r, y_r), according to Equation 8. The real points $Xr = (x_r, y_r)$ are determined by a person that specifies the location of feature points in the image, also called ground truth.

$$d = \sqrt{(x - x_r)^2 + (y - y_r)^2}. \tag{8}$$

The location of ground truth points in the image, however, is very imprecise, since more than one pixel can be considered as the correct location. As it can be seen in Figure 9, the correct right corner of a right eye can be located in more than one pixel, depending on the image resolution. Due to this fact, we established as an acceptable error d_{err} the distance of 5% between the 2 iris (d_{iris}) in the image. The reader may establish other thresholds, so our results will be presented in terms of percentiles of values for the relative error $\frac{d}{d_{iris}}$.

The image in Figure 9 was selected from the IMM face database [20], which consists of a ground truthed public database of 40 people in 6 different poses. We used this dataset in our experiments. The images have resolution of 640x480 pixels. Since we are working with color images we exclude the images of 3 people which were provided in gray scale. We also eliminated from the analysis the images where people were not in frontal view, since the face detector were not trained to locate faces in lateral position. This process results in 142 analyzed images, which have been used to achieve results presented in next analyses.

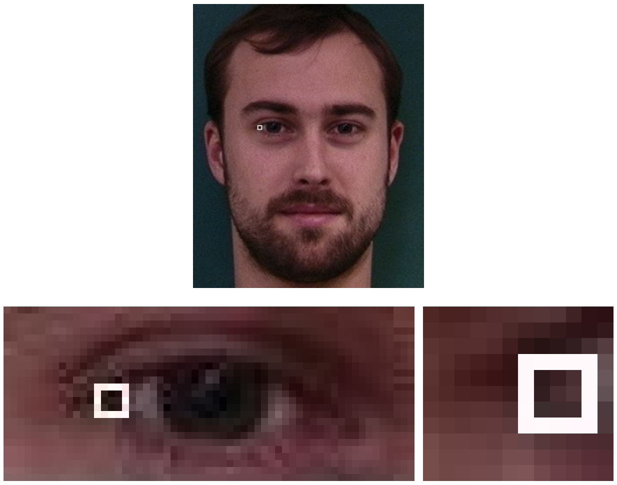


Figure 9. Magnification of the region of the eye, showing the difficulty in specifying a unique point for the corner of the eye in a 640x480 resolution image. The left image has a 12x11 pixels of resolution.

Color Space	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
RGB - Red	0.04	0.05	0.06	0.08	0.09	0.10	0.62
HSV - Value	0.04	0.05	0.06	0.07	0.09	0.12	0.62
Gray Scale	0.05	0.06	0.07	0.08	0.09	0.13	0.64
YCbCr - Y	0.05	0.06	0.07	0.08	0.08	0.13	0.64
RGB - Green	0.05	0.06	0.08	0.09	0.12	0.16	0.64
RGB - Blue	0.06	0.07	0.08	0.10	0.12	0.16	0.62
HSV - S	0.08	0.10	0.12	0.16	0.18	0.21	0.65
YCbCr - Cr	0.06	0.07	0.10	0.15	0.41	0.55	0.81
YCbCr - Cb	0.24	0.28	0.32	0.38	0.46	0.56	0.81

Table I

COMPARISON OF PERCENTILES OF VALUES OF THE RELATIVE ERROR $\frac{d}{d_{iris}}$ ACHIEVED BY THE PROPOSED METHOD APPLIED TO DIFFERENT COLOR SPACES.

Table I presents the deciles of relative errors $\frac{d}{d_{iris}}$ achieved by applying the method described in Section III in several channels of different color spaces. As it can be seen, 90% of the values of $\frac{d}{d_{iris}}$ are equal or smaller than 0.10 for the red channel in RGB color space, 80% of this relative distance are equal or smaller than 0.09 considering the same channel, and so on. Besides the small difference between the results achieved in channel R of RGB and V of HSV, we choosen the first one, since any color system transformation is necessary and, in this way, the computational time is reduced.

We compare the performance of the proposed method with 2 other approaches presented in the literature. The first one was proposed by Cristianacce [16] and the other one is an extension of ASM developed by Milborrow and Nicolls [10]. However, just some feature points are detected simultaneously by the three compared models. These points can be seen in Figure 10. So, other feature points we can currently detect are ignored in this analysis.

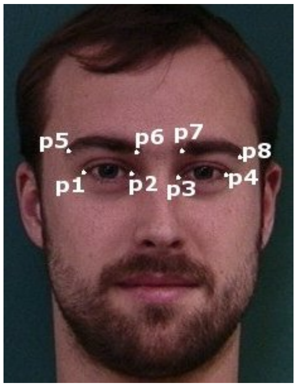


Figure 10. Points detected simultaneously by the models proposed by Cristianacce, Milborrow and ours.

Based on Tables II and III we can note that the proposed model presented better results for locating feature points in the eyes, in comparison with tested approaches. Regarding eyebrows, the boundaries are not always visible, what can impact the result of evaluation (see feature points from p5 to p8, mainly if we consider p5 and p8). In addition, it generates high values for obtained standard deviation as well. Future statistical evaluation could be done to verify the values significance.

The model presented in Section III was implemented in C++, using the OpenCV library [1] for image acquisition and manipulation. It also uses the library CVBlob. The computational cost of presented algorithm applied to typical web cam videos (320 x 240 pixels of resolution) processed in a Mobile DualCore Intel Pentium T2370, 1733 MHz is 27 frames per second (i.e. we achieve real time frame-rates).

VI. Final Considerations

This paper presented a method for eyes and eyelids detection to be used in performance driven animation of avatars. The method presents good results in comparison with other approaches, in quantitative terms. In addition, it works for simple web cams and achieves real time frame rates. However, future work are been employed in order to improve the following aspects:

- We are currently investigating the application of Deformable Templates [15] for point location in Step 8 of the pipeline, in order to increase both precision and accuracy;
- We intend to deal with non-frontal views.
- We are also performing quantitative matching with other tracking approaches like AAM. Qualitative analysis indicate that our method is more accurate then the cited, but further experimentation is required.
- MPEG4 platform provides a limited standard for one to one eye animation since it just allows the parameterization of the top of eyelids. More complex movements of the eyes with wrinkles and curvatures can not be mapped on this standard. We must investigate other animation platforms in order to achieve more reflective animation.

VII. Acknowledgment

This work was developed in cooperation with Hewlett-Packard Brasil Ltda, using incentives of Brazilian Informatics Law (Law n^o 8.248 of 1991).

References

[1] I. Pandzic and R. Fochheimer, *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. Wiley, John and Sons, 2002.

[2] H. Tang and T. Huang, "Mpeg4 performance-driven avatar via robust facial motion tracking," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, Oct. 2008, pp. 249–252.

[3] Z. Deng, U. Neumann, J. P. Lewis, T.-Y. Kim, and et al, "Expressive facial animation synthesis by learning speech coarticulation and expression spaces," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1523–1534, 2006.

[4] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models - their training and application," *Computer Vision and Image Understanding*, vol. 61, pp. 38–59, 1995.

[5] T. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearence models," in *European Conference on Computer Vision 1998*, vol. II. Springer, 1998, pp. 484–498.

[6] P. Yuen, J. H. Lai, and Q. Y. Huang, "Mouth state estimation in mobile computing environment," in *Sixth IEEE Int. Conf. on Automatic Face and Gesture Recognition*. IEEE Computer Society, 2004, pp. 705–710.

[7] Y. Tong, Y. Wang, Z. Zhu, and Q. Ji, "Facial feature tracking using a multi-state hierarchical shape model under varying face pose and facial expression," *Pattern Recognition, International Conference on*, vol. 1, pp. 283–286, 2006.

[8] I. Ari, A. Uyar, and L. Akarun, "Facial feature tracking and expression recognition for sign language," in *23rd Int. Symp on Computer and Information Science*. IEEE Computer Society, 2008, pp. 1–6.

[9] M. A. Haj, J. Orozco, J. Gonzalez, and J. J. J. Villanueva, "Automatic face and facial features initialization for robust and accurate tracking," in *19th International Conference on Pattern Recognition*. IEEE Computer Society, 2008, pp. 1–4.

[10] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," in *Proceedings of European Conference on Computer Vision*, 2008, pp. 504–513.

[11] P. Ekman, W. V. Friesen, and J. C. Hager, *The Facial Action Coding System*. Weidenfeld & Nicolson, 2002.

[1] http://opencv.willowgarage.com/wiki/

Table II
AVERAGE OF RELATIVE ERROR.

Model	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
Proposed Model	0.03	0.07	0.07	0.03	0.09	0.05	0.6	0.11
Cristinacce [16]	0.06	0.07	0.07	0.06	0.12	0.07	0.09	0.09
Milborrow [10]	0.12	0.12	0.10	0.11	0.14	0.13	0.13	0.11

Table III
STANDARD DEVIATION OF RELATIVE ERROR ON COMPARED MODELS.

Model	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
Proposed Model	0.02	0.03	0.04	0.02	0.13	0.6	0.5	0.13
Cristinacce [16]	0.04	0.04	0.04	0.04	0.06	0.04	0.06	0.05
Milborrow [10]	0.23	0.23	0.23	0.22	0.22	0.25	0.25	0.22

[12] H. Tang and T. S. Huang, “Mpeg4 performance-driven avatar via robust facial motion tracking,” in *International Conference on Computer Vision*, 2008, pp. 249–252.

[13] K. He, G. Wang, and Y. Yang, “Optical flow-based facial feature tracking using prior measurements,” in *7th International Conference on Cognitive Informatics*. IEEE Computer Society, 2008, pp. 324–331.

[14] C. Su and L. Huang, “Spatio-temporal graphical-model-based multiple facial feature tracking,” *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 13, pp. 2091–2100, 2005.

[15] A. Yuille, D. Cohen, and P. Hallinan, “Feature extraction from faces using deformable templates,” in *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR ’89., IEEE Computer Society Conference on*, 1989, pp. 104–109.

[16] D. Cristinacce and T. F. Cootes, “Facial feature detection and tracking with automatic template selection,” in *FGR ’06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 429–434.

[17] P. Viola and M. Jones, “Robust real-time object detection,” in *International Journal of Computer Vision*, 2001.

[18] L. G. Kourkoutis, K. I. Panoulas, and L. Hadjileontia, “Automated iris and gaze detection using chrominance: application to human-computer interaction using a low resolution webcam,” in *19th IEEE Int. Conf. on Tools with Artificial Intelligence*, 2007, pp. 536–539.

[19] M. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*, 2nd ed. Academic Press, 2007.

[20] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann, “The IMM face database - an annotated dataset of 240 face images,” Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Tech. Rep., may 2004. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3160