A New Facial Point Detector using Active Appearance Models

3er Congreso Internacional de Ingeniería Mecatrónica - UNAB

Alejandro Parada Mayorga
School of Electrical and
Electronic Engineering
Universidad Industrial de Santander
Bucaramanga, Colombia
Email: alejandro_parada.m@hotmail.com

Arturo Plata Gómez
School of Physics
Universidad Industrial de Santander
Bucaramanga, Colombia
Email: aplata@uis.edu.co

***Abstract*—In this paper we present a new scheme for facial point detection using a new face detection system based on Adaboost and color segmentation. The main idea of this work is the use of the frame of detected face with this combined approach of color and texture, for the construction and validation of active appearance models.**

I. Introduction

Facial point detection is one of the most important challenges in facial expression analysis [1] [2], actually it is considered as an open problem, but there exists powerfull tools for deal with it. The active appearance models (AAM), developed by Cootes et al [2] [3], have proven to be an excellent tool for this purpose [1]. These are built from a statistical model of shape and a statistical model of texture, but they have some limitations because the results of how well the model fits the object in the image strongly depend on the initial condition.

In this work, in order to avoid these limitations of AAM we propose to use a new scheme of face detection, and we built the construction and evaluation of the active appearance models only after this process.

The first stage of this scheme (see figure 1) it's based on a new propose developed by the authors, in which the color information is used to improve the results obtained by Adaboost developed by Viola and Jones [4]. The color segmentation techniques are based on the Zhang proposal called Color Centroids Segmentation [5]. Moreover we use a color image normalization developed by Fynlayson [6].

The second stage is built using active appearance models. In this we use the frame of face detection for initialize the deformable model of faces, in which we expect to find the most relevant points for facial expression analysis.

The paper is organized as follows, first we describe all the details of face detection, introducing the comprehensive colour image normalization, the color centroid segmentation and finally the well known Adaboost. After that, we present the active appearance models. Moreover there is a section about implementation, and another section on the results and discussion.

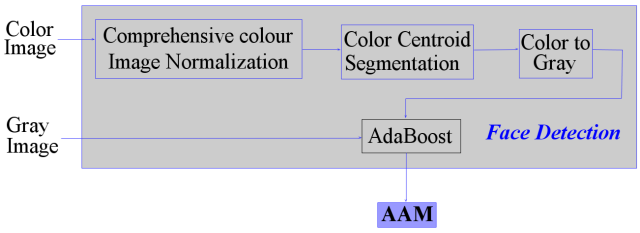

Fig. 1. Facial Point Detection Scheme

June 14, 2011

II. Face Detection

The first step in the face detection process consist of an comprehensive colour image normalization developed by Finlayson in [6]. The second is an stage based on color centroid segmentation developed by Zhang in [5], and finally is the stage related to the face detection using Adaboost developed by Viola and Jones in [4].
The main idea of this scheme is to discard non-facial information using the color of the image, in the case of a color image, and then use Adaboost for obtain more accurate results in face detection. If the image is on gray scale, is used Adaboost directly. In figure 1 you can see the basic system for face detection.

A. Comprehensive Colour Image Normalization

One image, under different lighting conditions can appear with different values in the RGB space either due to lighting geometry or due to illuminant color [7]. Therefore a colour normalization technique is necessary for an accurate color segmentation. Finlayson et al [6], have developed a iterative technique for color image normalization. Finlayson proof that successively using the classic[1] equations for compensate

[1]See [7] for the details and definitions about this

lighting geometry and illuminant colour, it is possible to find, an image that is idempotent and unique, and method is always convergent. Moreover rate of convergence is very high. Convergence is typically achieved in five or six iterations.

This method is described as follows:

Let $\mathcal{I}_{i,j} \quad j = R, G, B$ a image in the RGB space of color, i is the pixel. Let R the normalizer of rows operator defined as

$$R\left(\mathcal{I}_{i,j}\right) = \frac{\mathcal{I}_{i,j}}{\sum_{k=R,G,B} \mathcal{I}_{i,k}} \tag{1}$$

and let C be the normalizer of columns operator, defined as

$$C\left(\mathcal{I}_{i,j}\right) = \frac{N\mathcal{I}_{i,j}}{3\sum_{k=1}^{N} \mathcal{I}_{k,j}} \tag{2}$$

N is the pixel's number in the image.

Then to normalize the color in an image:

1. *Initialization*: Take $\mathcal{I}_{i,j}^{(0)} = \mathcal{I}_{i,j}$ as the initial value of the iterative process.
2. *Iteration Step*: Make $\mathcal{I}_{i,j}^{(r+1)} = C\left(R\left(\mathcal{I}_{i,j}^{(r)}\right)\right)$.
3. *Termination Step*: If $\mathcal{I}_{i,j}^{(r+1)} = \mathcal{I}_{i,j}^{(r)}$ stop the process and make $\mathcal{I}_{i,j}^{normalized} = \mathcal{I}_{i,j}^{(r)}$, if not go to step 2.

Once the convergence is achived, it's necessary to make scaling on the values of the resultant image. We propose the scaling as follows:

$$\mathcal{I}_{i,j}^{final} = \mathcal{I}_{i,j}^{(0)} \mathcal{I}_{i,j}^{normalized} \tag{3}$$

Ebner has shown in [7], that this is not the only way for making the rescaling, but we think is the best way.

B. Color Centroid Segmentation

Zhang [5] has proposed a new technique for color segmentation. This is based on the transformation from the RBG space to a two dimensional space using a new coordinate system and the centroid of a triangle. Figure 2 shows this transform. Each component of the vector $(\mathcal{I}_{i,R}, \mathcal{I}_{i,G}, \mathcal{I}_{i,B}) = (r, g, b)$ is mapped to a new axes system, which are separated by 120 degrees. The new R axis is on 90 degrees to the horizontal. Then, a triangle can be constructed, it will have centroid (u, v), where (u, v) is the point in the new space transformation. It's mathematically as follows

Let $\mathcal{T} : \mathbb{R}^3 \longmapsto \mathbb{R}^2$ be the transformation operator from the RGB space to $\mathbb{R}^2$. The transformation is defined as follows

$$\mathcal{T}\left(\mathcal{I}_{i,j}\right) = u_i \hat{\mathbf{e}}_{\mathbf{x}} + v_i \hat{\mathbf{e}}_{\mathbf{y}} \tag{4}$$

where

$$u_i = \frac{1}{3}\left[\mathcal{I}_{i,B} \cos\left(\frac{\pi}{6}\right) - \mathcal{I}_{i,G} \cos\left(\frac{\pi}{6}\right)\right]$$

$$v_i = \frac{1}{3}\left[\mathcal{I}_{i,R} - \mathcal{I}_{i,B} \sin\left(\frac{\pi}{6}\right) - \mathcal{I}_{i,G} \sin\left(\frac{\pi}{6}\right)\right]$$

$\hat{\mathbf{e}}_{\mathbf{x}}, \hat{\mathbf{e}}_{\mathbf{y}}$ are the unit vectors in x, y coordinates of $\mathbb{R}^2$

According to Zhang, after this transformation is done, we represent the data in polar coordinates (θ, r). Once done, it is necessary to take a lot of skin samples and thresholds should be selected for the values of r and θ. So that the region where we expect to find pixels associated with skin color can be

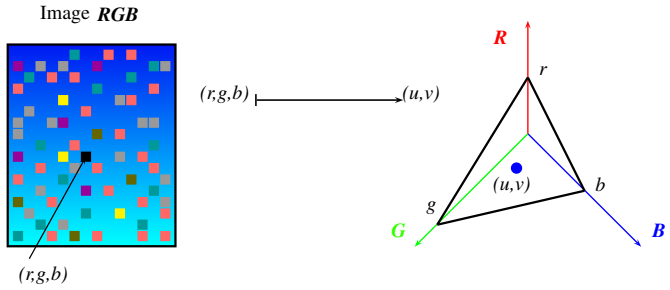

Fig. 2. Transformation from RGB to $\mathbb{R}^2$.

defined as $\{(\theta, r) | \, r \in [r_{min}, r_{max}], \quad \theta \in [\theta_{min}, \theta_{max}]\}$.

In this work we represent the distribution region of skin pixels, as the interior of a polygon. The model of this polygon is obtained considering a distribution in the (θ, r) space with 517880 skin samples taked from FERET [8] [9], Cohn-Kanade [10] and PIE [11], databases with a miscellaneous of 100 facial images of World Wide Web . Thus the limits of this region are the following

$$r - 54,84 \leq -1.2758(\theta - 54.24) \tag{5}$$

$$r - 54.84 \leq 2.2074(\theta - 54.24) \tag{6}$$

$$r \geq 2.5 \tag{7}$$

We name the preceding region as $\mathfrak{M}_{\mathfrak{s}} \subset \mathbb{R}^2$, such that the skin region distribution is defined as follows

$$\{(\theta, r) | \, (\theta, r) \in \mathfrak{M}_{\mathfrak{s}}\} \tag{8}$$

In this way we obtain a flexible model, but that eliminates many parts that do not belong to the distribution of skin colors. We obtain this mathematical description, in an empirical way. As Zhang took some empirical values for delimit θ and r, we take this lines as the limit of interest region.

After thresholding, it is necessary to make the correction process using non-linear thresholding. Zhang made this process as follows: Let $\mathcal{I}_{i,gray}$ the gray scale of $\mathcal{I}_{i,j}$, then the non-linear correction is

$$\mathcal{I}_{i,binary} = \frac{\ln\left(1 + 255\mathcal{I}_{i,gray}\right)}{k \ln\left(1 + 255\right)} \tag{9}$$

where $k = 2$. With this process, the noise is reduced. The k value, indicate the number of clusters in the resultant image. So that the final mask for segmentation is obtained as

$$\mathcal{I}_{i,masc} = \mathcal{I}_{i,ccs} \mathcal{I}_{i,binary} \tag{10}$$

$\mathcal{I}_{i,ccs}$ is the masc of CCS thresholding. After that, we fill the holes of the $\mathcal{I}_{i,masc}$. So the final mask for segmentation is

$$\mathcal{I}_{i,Fmasc} = \mathbf{F}_{\text{fillHoles}}\left(\mathcal{I}_{i,masc}\right) \tag{11}$$

$\mathbf{F}_{\text{fillHoles}}$ is the filling holes operator.

C. Adaboost

Viola and Jones have proposed in [4] a robust face detector based on Adaboost. They have achieved a high recognition rate. This scheme is so far, the best face detector [1]. This technique is mainly based on the use of extracted features with Haar-type functions, and the use of a cascade of classifiers, each of which is constructed from weak classifiers. Using Adaboost weak classifiers are designed so that the strong classifiers are a superposition of the first. To make this training efficiently Viola and Jones introduced the concept of integral image, by which you can efficiently get all the features of the image in a new image computed from the original by cumulative sums.

The principal facts about Adaboost algorithm can be sumarized as follows [4] [1]

0. (Input)
   (1) Training examples $\mathcal{Z} = \{(x_1, y_1), \ldots (x_N, y_N)\}$, where $N = a + b$, such that there are a examples with $y_i = 1$ and b examples with $y_i = -1$.
   (2) The number M of weak classifiers to be combined.
1. (Initialization)
   $$w_i^{(0)} = \frac{1}{2a} \quad \text{For those examples with} \quad y_i = 1$$
   $$w_i^{(0)} = \frac{1}{2b} \text{For those examples with} \quad y_i = -1.$$
2. (Forward Inclusion)
   For $m = 1, \ldots, M$:
   (1) Choose optimal h_m to minimize the weighted error:
   $$\epsilon_m = \sum_i w_i^{(m-1)} 1 \left[\text{sign}\left(h_m\left(x_i\right)\right) \neq y_i\right]$$
   (2) Choose α_m according to
   $$\alpha_m = \log\left(\frac{1 - \epsilon_m}{\epsilon_m}\right)$$
   (3) Update $w_i^{(m)} \leftarrow w_i^{(m)} e^{-y_i \alpha_m h_m(x_i)}$ and normalize to $\sum_i w_i^{(m)} = 1$
3. (Output)
   Classification function: H_M as
   $$H_M(x) = \frac{\sum_{m=1}^{M} \alpha_m h_m\left(x\right)}{\sum_{m=1}^{M} \alpha_m}$$

Once all the classifiers have been designed, a attentional cascade is used for achieve an efficient search in the image. Each classifier uses an increasing number of features. Such that the firsts play an important role for rule out regions with no facial information. Therefore only regions with richer information are accepted as facial regions. For all details about the procedure, the author refer the reader to the work of [4].

III. Active Appearance Models

In this work, we follow the ideas proposed by Cootes et al [2] [12] [13] [3] and the work done by Stegmann [14], in which combining a shape model variation with a texture model variation forms that we today know as Active Appearance Model. By texture we mean the gray pattern in the image.

A. Shape Model

To build the model, we require a training set of images annotated with respective landmarks. As our interest is on facial images, we use the Cohn-Kanade database [10] images after the process of face detection. The landmarks indicate the main features of a face (see figure 3 a)). In order to obtain the shape model, We apply the Procrustes analysis for shape alignment [15] [16] [17] [18] [14], and then build a statistical shape model.

1) Shape Aligment: Let

$$\mathbf{x}_i = \left(x_i^{(1)}, x_i^{(2)}, x_i^{(3)}, \ldots, x_i^{(n)}, y_i^{(1)}, y_i^{(2)}, y_i^{(3)}, \ldots, y_i^{(n)}\right)^T$$

be a column vector of $2n$ components whose first n components are the abscissas and the components from $n+1$ to $2n$ are the ordinates of the landmarks that represent the shape of the i facial image of the training set. The shape alignment steps are as follows [16] [15] [14]

Initialize

1 Translate each example so that it's centre fo gravity is at the origin.
2 Choose one example $\mathbf{x}_i$ as an initial estimate of the mean shape $\bar{\mathbf{x}}$ and scale so that $\|\bar{\mathbf{x}}\| = 1$.
3 Name $\bar{\mathbf{x}}_0 = \bar{\mathbf{x}}$

Repeat

4 Align all the shapes with the current estimate of the mean shape[2].
5 Re-estimate mean form aligned shapes:

$$\bar{\mathbf{x}} = \frac{1}{n_S} \sum_{j=1}^{n_S} \mathbf{x}_j \tag{12}$$

where n_S is the namber of shapes that will be aligned.
6 Allign $\bar{\mathbf{x}}$ to the initial reference $\bar{\mathbf{x}}_0$
7 Normalize the mean so that $\|\bar{\mathbf{x}}\| = 1$

Until convergence

2) Principal Component Analysis: First, we compute de covarince of the data $\mathbf{S}$ as follows [17] [18] [14]

$$\mathbf{S_s} = \frac{1}{n_S - 1} \sum_{i=1}^{n_S} \left(\mathbf{x}_i - \bar{\mathbf{x}}\right)\left(\mathbf{x}_i - \bar{\mathbf{x}}\right)^T \tag{13}$$

and then we compute the eigenvectors $\boldsymbol{\phi}_{\mathbf{s}k}$ corresponding to the eigenvalues $\lambda_{\mathbf{s}k}$, sorted so that $\lambda_{\mathbf{s}k} \geq \lambda_{\mathbf{s}(k+1)}$. If $\boldsymbol{\Phi}_\mathbf{s}$ is the matrix whose column vectors are $\boldsymbol{\phi}_{\mathbf{s}k}$ eigenvectors corresponding to the $q_\mathbf{s}$ largest eigenvalues, the value of any of $\mathbf{x}_i$ can be aproximated by

$$\mathbf{x} \approx \bar{\mathbf{x}} + \boldsymbol{\Phi}_\mathbf{s} \mathbf{b}_\mathbf{s} \tag{14}$$

where $\mathbf{b}_\mathbf{s}$ is a vector that defines a set of parameters of a deformable model. By varying the elements of $\mathbf{b}_\mathbf{s}$ we can vary the shape $\mathbf{x}$, using equation 14. The variance of the i^{th} parameter, $b_{\mathbf{s}i}$, across the training set is given by $\lambda_{\mathbf{s}i}$ [19]. The number of eigenvectors to retain, $q_\mathbf{s}$, can be chosen so that the model represents some portion fo the total variance of the data

[2]For the align of two shapes you can see [15] for all the details

[19] [17] [18], according to Cootes et al we take as criterion the eigenvectors whose eigenvalues represent the 98% of the total variance such that

$$\frac{\sum_{k=1}^{q_s} \lambda_k}{\sum_{k=1} \lambda_k} \geq 0.98$$

B. Texture Model

As in the case of the shape model, we require a set of training images. Indeed this set of training images is the same in both cases. Once we have the training images we use a normalization procedure to built the texture model in a similar way as in the case of the shape model [17] [18]. It is important that the texture samples, are taken inside from the convex hull obtained from the non aligned shapes [14].

1) Image Warping: Using image warping we map the texture samples inside a shape to another [14]. For this we construct a tringular mesh taking into account the landmarks and using de *Delaunay algorithm*, then we map gray values of pixels inside of each triangle to the corresponding triangle of the another shape respectively [14]. Let $\boldsymbol{v}_{(1)}, \boldsymbol{v}_{(2)}, \boldsymbol{v}_{(3)}$ the vector that represent the vertices of a triangle $\triangle_{\boldsymbol{v}_{(1,2,3)}}$ on the object's mesh taken counterclockwise direction, and $\boldsymbol{v}^{'}_{(1)}, \boldsymbol{v}^{'}_{(2)}, \boldsymbol{v}^{'}_{(3)}$ the corresponding vector of vertices on the triangle $\triangle_{\boldsymbol{v}^{'}_{(1,2,3)}}$ in the another shape, then the value of the pixel located in $\boldsymbol{v} = [x, y]^T$ inside the triangle $\triangle_{\boldsymbol{v}_{(1,2,3)}}$ is mapped to $\boldsymbol{v}^{'} = [x^{'}, y^{'}]^T$ inside the triangle $\triangle_{\boldsymbol{v}^{'}_{(1,2,3)}}$ as follows

$$\mathcal{T}_{warp}(\boldsymbol{v}) = \left[x^{'}, y^{'}\right]^T = \alpha \boldsymbol{v}^{'}_{(1)} + \beta \boldsymbol{v}^{'}_{(2)} + \gamma \boldsymbol{v}^{'}_{(3)} \tag{15}$$

where

$$\alpha = 1 - (\beta + \gamma) \tag{16}$$

$$\beta = \frac{y v_{x_3} - v_{x_1} y - v_{x_3} v_{y_1} - v_{y_3} x + v_{x_1} v_{y_3} + x v_{y_1}}{-v_{x_2} v_{y_3} + v_{x_2} v_{y_1} + v_{x_1} v_{y_3} + v_{x_3} v_{y_2} - v_{x_3} v_{y_1} - v_{x_1} v_{y_2}} \tag{17}$$

$$\gamma = \frac{x v_{y_2} - x v_{y_1} - v_{x_1} v_{y_2} - v_{x_2} y + v_{x_2} v_{y_1} + v_{x_1} y}{-v_{x_2} v_{y_3} + v_{x_2} v_{y_1} + v_{x_1} v_{y_3} + v_{x_3} v_{y_2} - v_{x_3} v_{y_1} - v_{x_1} v_{y_2}} \tag{18}$$

and $\boldsymbol{v}_{(i)} = [v_{x_i}, v_{y_i}]^T$.

2) Texture Alignment: To minimize the effect of global lighting variation, we normalize the training images as follows [17] [18] [1]:

Let $\mathbf{g}_i$ the column vector that contains the texture samples (i.e. the gray values of the gray image) of the image i inside the convex hull of the non aligned shape $\mathbf{x}_i$.

1. Warp all the texture samples from inside the convex hull of $\mathbf{x}_i$ of the original image i to the mean shape placed in the center of the image and with a scaling that is the average of the of size of all faces in the training images $\mathbf{g}_i \leftarrow \mathcal{T}_{warp}(\mathbf{g}_i)$
2. Compute for all $\mathbf{g}_i$

$$\mathbf{g}_i = \frac{\mathbf{g}_i - \beta \mathbf{1}}{\sqrt{\alpha}} \tag{19}$$

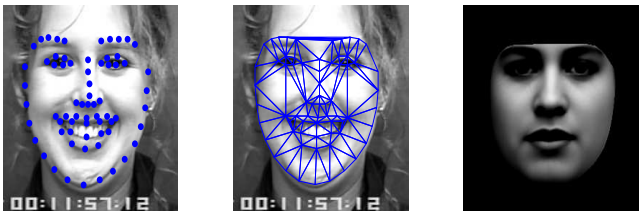

Fig. 3. Left to Right: a) Landmarks on a face, b)The triangular mesh constructed using the landmarks, c) The mean texture on the mean shape

Where

$$\beta = \frac{\mathbf{g}_i \cdot \mathbf{1}}{n_S} \tag{20}$$

$$\alpha = \frac{\|\mathbf{g}_i\|^2}{n_S} - \beta^2 \tag{21}$$

here $\mathbf{1}$ is the vector whose components are equal to 1.

3) Principal Component Analysis: Once the texture samples are normalized, We compute the covariance matrix [17] [18] [14] as

$$\mathbf{S}_g = \frac{1}{n_S - 1} \sum_{i=1}^{n_S} (\mathbf{g}_i - \bar{\mathbf{g}})(\mathbf{g}_i - \bar{\mathbf{g}})^T \tag{22}$$

and then we compute the eigenvectors $\boldsymbol{\phi}_{\mathbf{g}k}$ corresponding tho the eigenvalues $\lambda_{\mathbf{g}k}$, sorted so that $\lambda_{\mathbf{g}k} \geq \lambda_{\mathbf{g}(k+1)}$. If $\boldsymbol{\Phi}_{\mathbf{g}}$ is the matrix whose column vectors are $\boldsymbol{\phi}_{\mathbf{g}k}$ eigenvectors corresponding to the $q_{\mathbf{g}}$ largest eigenvalues, the value of any of $\mathbf{g}_i$ can be approximated by

$$\mathbf{g} \approx \bar{\mathbf{g}} + \boldsymbol{\Phi}_{\mathbf{g}} \mathbf{b}_{\mathbf{g}} \tag{23}$$

where $\mathbf{b}_{\mathbf{g}}$ is a vector that defines a set of parameters of the model. By varying the elements of $\mathbf{b}_{\mathbf{g}}$ we can vary the texture $\mathbf{g}$, using equation 23. The variance of the i^{th} parameter, $b_{\mathbf{g}i}$, across the training set is given by $\lambda_{\mathbf{g}i}$ [19]. The number of eigenvectors to retain, $q_{\mathbf{g}}$, can be chosen so that the model represents some portion of the total variance of the data [19] [17] [18], according to Cootes et al we take as criterion the eigenvectors whose eigenvalues represent the 98% of the total variance as in the case of shape model.

C. Combined Model Formulation

We can obtain a combined formulation of the appearance model applying principal component analysis to the vectors

$$\mathbf{b} = \begin{bmatrix} \mathbf{W}_s \mathbf{b}_s \\ \mathbf{b}_g \end{bmatrix} = \begin{bmatrix} \mathbf{W}_s \boldsymbol{\Phi}_s^T (\mathbf{x} - \bar{\mathbf{x}}) \\ \boldsymbol{\Phi}_g^T (\mathbf{g} - \bar{\mathbf{g}}) \end{bmatrix} \tag{24}$$

If We denote the matrix of the new eigenvectors as $\hat{\boldsymbol{\Phi}}$, then the new model has the following form

$$\mathbf{b} = \hat{\boldsymbol{\Phi}} \mathbf{c} \tag{25}$$

The vectors $\mathbf{c}$ are called the appearance parameters [2] [3] [20]. With this, we can control the shape an the texture models. Since the shape and texture have zero mean, $\mathbf{c}$ does have too zero mean [17] [18]. Thus the combined model, can be written as follows

$$\mathbf{x} = \bar{\mathbf{x}} + \boldsymbol{\Phi}_s \mathbf{W}_s^{-1} \hat{\boldsymbol{\Phi}}_{cs} \mathbf{c} \tag{26}$$

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{\Phi}_g \hat{\mathbf{\Phi}}_{cg} \mathbf{c} \tag{27}$$

where

$$\hat{\mathbf{\Phi}} = \begin{bmatrix} \hat{\mathbf{\Phi}}_{cs} \\ \hat{\mathbf{\Phi}}_{cg} \end{bmatrix} \tag{28}$$

1) Choice of shape Parameters Weights $\mathbf{W}_s$: We use the scheme proposed by Cootes et al, $\mathbf{W}_s$ is selected as follows:

$$\mathbf{W}_s = r\mathbf{I} \tag{29}$$

where $\mathbf{I}$ is the identity matrix, and $r > 0$ is such that

$$r^2 = \frac{\sum_{k=1}^{q_\mathbf{g}} \lambda_{\mathbf{g}k}}{\sum_{k=1}^{q_\mathbf{s}} \lambda_{\mathbf{s}k}} \tag{30}$$

D. Active Appearance Model Search

For make the search in the image, it is necessary to apply a similarity transformation over the model. With this transformation, we represent in the image frame the variation of the combined model. This transformation is defined for any point $\boldsymbol{v} = [x, y]^T$ as follows

$$\mathcal{T}_\mathbf{t}(\boldsymbol{v}) = \begin{bmatrix} 1 + s_x & -s_y \\ s_y & 1 + s_x \end{bmatrix} \boldsymbol{v} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{31}$$

$\mathbf{t} = (t_1, t_2, t_3, t_4) = (s_x, s_y, t_x, t_y)$ where $s_x = s\cos(\theta) - 1$, $s_y = s\sin(\theta)$. θ is the rotation, and s the scaling, whereas (t_x, t_y) is the translation.

1) Learning to Correct Model Parameters: The main idea for learn to correct the parameters value, is to use the known conditions of images used in the model. With this we make a perturbation on these, and then We store the error. Then, all the errors are stored in a matrix that is used after that for make the search in the image frame.

Let $\mathbf{c}_{i,0}$ and $\mathbf{t}_{i,0}$ the values of the model parameters and the similarity transformation with which the models generates the element $\mathbf{g}_i$ of the image i. We make a known perturbation $\delta\mathbf{c}, \delta\mathbf{t}$ for each parameter individually as

$$\mathbf{c} = \mathbf{c}_{i,0} + \delta\mathbf{c} \tag{32}$$

$$\mathbf{t} = \mathbf{t}_{i,0} + \delta\mathbf{t}, \tag{33}$$

such that

$$\mathcal{T}_{\mathbf{t}_{i,0}+\delta\mathbf{t}}(\mathbf{x}_m) = \mathcal{T}_{\mathbf{t}_{i,0}}(\mathcal{T}_{\delta\mathbf{t}}(\mathbf{x}_m)) = \hat{\hat{\mathbf{x}}}_m \tag{34}$$

$$\mathbf{g}_m = \bar{\mathbf{g}} + \mathbf{\Phi}_g \hat{\mathbf{\Phi}}_{cg}(\mathbf{c}_{i,0} + \delta\mathbf{c}) \tag{35}$$

$$\mathbf{x}_m = \bar{\mathbf{g}} + \mathbf{\Phi}_g \hat{\mathbf{\Phi}}_{cg}(\mathbf{c}_{i,0} + \delta\mathbf{c}) \tag{36}$$

Let $\hat{\hat{\mathbf{g}}}_m$ the generated texture object in the image frame obtained by warping $\mathbf{g}_m$ into $\hat{\hat{\mathbf{x}}}_m$, then we compute the error as follows

$$\mathbf{\Xi}_\mathbf{g} = \mathbf{g}_s - \hat{\hat{\mathbf{g}}}_m \tag{37}$$

Where $\mathbf{g}_s$ is the sample textures inside the convex hull of $\hat{\hat{\mathbf{x}}}_m$ that is the shape model in the image frame.

Let $\mathbf{g}_{\partial^k p_i}$ the error computed by perturbation of the parameter p_i by an value of $\zeta(k)$, $\quad k = 1, 2, 3, ..., d$ and let $\mathbb{A}$ the matrix whose column vectors $\mathbf{a}_i$ are of the form

$$\mathbf{a}_i = \frac{1}{d} \sum_{k=1}^{d} w_k \mathbf{g}_{\partial^k p_i} \tag{38}$$

where

$$w_k = \frac{1}{p_k} e^{-\frac{(\delta p_k)^2}{2\sigma_k^2}} \tag{39}$$

σ_k is the standard deviation, if $p_k = c_k$ then $\sigma_k = \sqrt{\lambda_k}$ and if $p_k = t_k$ σ_k has another value depending on the perturbations.

Now, with this and according to [18] [12], construct

$$\hat{\mathbb{R}} = \left(\mathbb{A}\mathbb{A}^T\right)^{-1} \mathbb{A} \tag{40}$$

2) Iterative Model Refinement: For the iterative model refinement we follow the steps proposed by Cootes et al [17] [18] [12]:

1. Project the texture sample $\mathbf{g}_s$ into the texture model frame making zero mean and unit variance, obtaining a new $\mathbf{g}_s$.
2. Evaluate the error $\mathbf{\Xi}_\mathbf{g} = \mathbf{g}_s - \hat{\hat{\mathbf{g}}}_m$, and the current error, $E = \|\mathbf{\Xi}_\mathbf{g}\|^2$.
3. Compute the predicted displacements, $\delta\mathbf{p} = -\hat{\mathbb{R}}\mathbf{\Xi}_\mathbf{g}$, where $\mathbf{p}^T = (\mathbf{c}^T|\mathbf{t}^T)$
4. Update the model parameters $\mathbf{p} \rightarrow \mathbf{p} + \zeta(k)\delta\mathbf{p}$ where initially $k = 1$, indeed

$$\zeta(k)\delta\mathbf{p} = \begin{bmatrix} \zeta_\mathbf{c}(k)\delta c_1 \\ \zeta_\mathbf{c}(k)\delta c_2 \\ \vdots \\ \zeta_\mathbf{c}(k)\delta c_n \\ \zeta_\mathbf{t}(k)\delta t_1 \\ \vdots \\ \zeta_\mathbf{t}(k)\delta t_4 \end{bmatrix} \tag{41}$$

5. Calculate the new points, $\hat{\hat{\mathbf{x}}}'_m$ and model texture in the image frame $\hat{\hat{\mathbf{g}}}'_m$.
6. Sample the image at the new points inside the convex hull of $\hat{\hat{\mathbf{x}}}'_m$ and normalize to obtain $\mathbf{g}'_s$.
7. Calculate a new error vector, $\mathbf{\Xi}'_\mathbf{g} = \mathbf{g}'_s - \hat{\hat{\mathbf{g}}}'_m$.
8. If $\left\|\mathbf{\Xi}'_\mathbf{g}\right\|^2 < E$, then accept the new estimate; otherwise, try at $k = 0.5, k = 0.25$, etc.

Cootes et al [17] [18] and Stegmann in [12] have shown that the optimal choice for the model parameter displacements are according to

$$\delta c_i = \sqrt{\lambda_i} \tag{42}$$

and we choose

$$\zeta_\mathbf{c}(k) = \frac{1}{5}(k - 3.5), \quad \forall k \in [1, 6], k \in \mathbb{Z} \tag{43}$$

for the displacements of the similarity transformation parameters the displacements are as follows

$$\delta t_i = 1 \tag{44}$$

Fig. 4. Some Results. Left to right: a) The original image with the initial position of the mean shape, b)The initial texture over the mean shape, c)The mesh shape after convergence adapted to the real facial shape, d)The final texture of the model after convergence, e) The original Image

and we choose

$$\zeta_{\mathbf{t}|t_1}(k) = \zeta_{\mathbf{t}|t_2} = \frac{1}{5}(k - 3.5), \quad \forall k \in [1, 6], k \in \mathbb{Z} \tag{45}$$

$$\zeta_{\mathbf{t}|t_3} = \zeta_{\mathbf{t}|t_4} = \frac{2(k - 3.5)}{s_{x0} + 1}, \quad \forall k \in [1, 6], k \in \mathbb{Z} \tag{46}$$

where $s_{x0} + 1 = s_{x0}\cos(\theta_0)$ defines the parameters of the similarity transformation without perturbing, such that $\mathbf{g}_i$ is generated in the image frame as $\mathcal{T}_{\mathbf{t}_{i,0}}(\mathbf{g}_i) = \hat{\mathbf{g}}_i$.

IV. Implementation

The comprehensive colour image normalization is implemented on Matlab. Due to the low computational cost, the programation language no plays a central role.
As in the last case, the implementation of CCS is on Matlab. The program was making using vectorial operations, such that the computational cost is low.
The most popular implementation of Adaboost is avaliable in the openCV library for C++. Masnadi [21] has done an implementation of this technique on Matlab, but it is not the best way due to the computational cost. Therefore we have decided to use the implementation done in the OpenCV library. We use Matlab for the implementation of AAM. We have made many efforts to reduce the computational cost. The mean time in a search is about 5 minutes or less.

V. Results

For construct the model we use 244 images, and 193 images for make the training, i.e, the construction of the $\hat{\mathbb{R}}$ matrix. Finally we use 100 different images, not used neither the model either the training, to validate the model. Figure 4 shows some of the results. The final error has a value of 48.44, 67.13, 92.48 and 77.37 for each individual seen form top to down.

A. Quantitative Evaluation

Our principal interest is about how the mesh of the shape is adapted to the facial expression on the image of the prove. As can be seen in the figure 4, some good results can be obtained with a final error about 90. We consider in this work 95 as a satisfactory value of final error. 84% of the images have an error less or equal to this value.

VI. Discussion and Conclusions

It is clear that the use of a face detection system improves the behaviour of the active appearance models. If the main interest, as in the case of this work, is in how well the shape model fits the facial expression of the image, the texture no plays a central role after the process have achieved convergence. But it is essential for reduce the error in the process of search.

Acknowledgment

The authors want to thanks to FERET program, portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office [8] [9]. Similarly, the authors are grateful for the use of Cohn-Kanade [10] and PIE [11] databases. Alejandro Parada wants to thank the Industrial University of Santander for the scholarship that has been assigned to pursue graduate study and research.

References

[1] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*. Springer, 2005.
[2] C. T. G.J. Edwards and T. Cootes, "Interpreting face images using active appearance models," pp. 1–6, 1998.
[3] T. C. G.J. Edwards and C. Taylor, "Advances in active appearance models," pp. 137–142, 1999.
[4] M. J. P. Viola, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the IEEE Computer Society conference on Computer Vision and Pattern Recognition*, vol. 20, 2001.
[5] . J. Q. Zhang, S.C. Kamata, "Face detection and tracking in color images using color centroids segmentation," *International Conference on Robotics and Biomimetics, Bankok, Thailand*, pp. 1008–1013, feb 2009.
[6] G.D. Finlayson, B. Schiele, J.L. Crowley, "Comprehensive colour image normalization," *ECCV98 Fifth Wuropean Conference on Computer Vision*, vol. 1, pp. 475–490, 1998.
[7] M. Ebner, *Color Constancy*. Wiley, 2007.
[8] P.J. Phillips, H. Wechsler, J. Huang, P. Rauss, "The feret database and evaluation procedure for face recognition algorithms," *Image and Vision Computing J*, vol. 16, pp. 295–306, 1998.
[9] S. R. P. R. P.J Phillips, H. Moon, "The feret evaluation methodology for face recognition algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1090–1104, 2000.
[10] C. J. F. . T. Y. Kanade, T., "Comprehensive database for facial expression analysis," *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), Grenoble, France*, pp. 46–53, 2000.
[11] T. Sim, S. Baker, M. Bsat, "The cmu pose, illumination, and expression (pie) database of human faces," *Carnegie Mellon University*, pp. 295–306, 2000.
[12] C. T. T.F. Cootes, G.J. Edwards, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 681–685, 2001.
[13] T. Cootes and C. Taylor, "Constrained active appearance models," pp. 748–754, 2001.

[14] M. B. Stegmann, *Active Appearance Models: Theory, Extensions and Cases. Master Thesis*. Technical University of Denmark, 2000.
[15] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*. Wiley, 1998.
[16] R. Davies, C. Twining, and C. Taylor, *Statistical Models of Shape, optimization and evaluation*. Springer, 2008.
[17] T. Cootes and C.J.Taylor, *Statistical Models of Appearance for Computer Vision*. University of Manchester, 2000.
[18] T. Cootes and Chris.J.Taylor, *Statistical Models of Appearance for Computer Vision*. University of Manchester, 2004.
[19] I. Jolliffe, *Principal Component Analysis*. Springer, 2002.
[20] X. L. D. T. Xinbo Gao, Ya Su, "A review of active appearance models," *IEEE Transactions on Systems, Man, and cybernetics-Part C: Applications and Reviews*, pp. 145–158, 2010.
[21] H. Masnadi-Shirazi, "Adaboost face detection," *Department of Electrical and Computer Engineering. University of California, San Diego UCSD*, pp. 1–6, 2007.