

# Robust 3D Face Tracking on Unknown Users with Dynamical Active Models

Dianle Zhou and Patrick Horain

Institut Telecom, Telecom & Management SudParis  
9 rue Charles Fourier, 91011 Évry Cedex France  
{Dianle.Zhou, Patrick.Horain}@IT-SudParis.eu

**Abstract.** The Active Appearance Models [1] and the derived Active Models (AM) [4] allow to robustly track the face of a single user that was previously learnt, but works poorly with multiple or unknown users. Our research aims at improving the tracking robustness by learning from video databases. In this paper, we study the relation between the face texture and the parameter gradient matrix, and propose a statistical approach to dynamically fit the AM to unknown users by estimating the gradient and update matrices from the face texture. We have implemented this algorithm for real time face tracking and experimentally demonstrate its robustness when tracking multiple or unknown users' faces.

**Keywords:** Face Tracking, Active Appearance Models, Face Animation, Virtual Reality.

## 1 Introduction

Head pose and facial gesture estimation is a crucial task in computer vision applications such as human-computer interaction, biometrics, etc. It is a challenge because of the variability of facial appearance within a video sequence. This variability is due to changes in head pose (particularly out-of-plane head rotations), facial expression, lighting, or occlusions, or a combination of all of them.

Classical active appearance models (AAM) [1], [2] work by globally registering a face model onto images. They require initially learning off-line a statistical model from example images of a user. This model describes appearance variations, *i.e.* image variation, with respect to model parameters variations as a matrix of gradients. Parameter variations can then be retrieved from residual appearance variations using an update matrix computed as a pseudo-inverse of the gradients matrix. Face tracking then consists in optimally registering the face model that was learnt onto input images using a gradient descent.

Feature-based approach such as elastic bunch graph matching (EBGM) [3] is robust to homogeneous illumination changes and affine deformations of the face image. The AAM approach was found to be more precise than the feature based approach [1], but unfortunately AAM tracking highly depends on learning the user to be tracked, so it fails with unknown users.

Following Dornaika and Ahlberg [4], we consider an Active Model (AM) that is simplified version of AAM, *i.e.* we learn the shape variations and grab face texture online while we do not learn texture variations.

In this paper we address tracking unknown users by enhancing the AM approach. We show there is a relation between the face model texture and the gradient matrix that we model by using a combined principal component analysis (PCA) [5] on the texture and the gradient matrix. So, we can generate an update matrix for each user, no matter if that user was learnt or not. We demonstrate this with a real-time application to capture head pose and facial gesture using a commodity web camera, which works even on unknown users with controlled lighting conditions and camera.

The rest of the paper is organized as follows. In Section 2, we analyze the AM and point out its short-coming, and propose the new algorithm and framework. Then we give the implement detail in Section 3. Experimental result and conclusion are presented in Sections 4 and 5.

## 2 Tracking Framework

We divided the framework of our tracking application into two parts: offline learning and online tracking. For the offline learning part, we will describe how to get the statistic information from training data. Then we will introduce how to use those information in Active Model search.

### 2.1 Offline Learning

Offline learning consists in estimating:

- (1) Face Shape
- (2) Face Texture
- (3) Update matrix
- (4) Relation between the update matrix and the face texture.

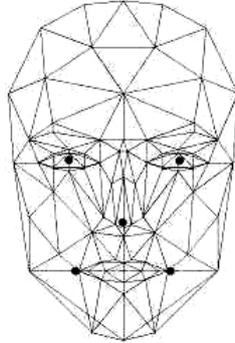
Steps (1) to (3) are classical in AM learning [4]. We introduce (4) for handling unknown users

#### (1) Face Shape

Face shape consists in face morphology and current face expressions. These should be captured by registering the 3D face model. To obtain this “shaped 3D face model”, we displace the vertices of a generic 3D-face model such as CANDIDE-3 [6]. Fig. 1 shows the CANDIDE-3 wireframe.

The 3D shape of this deformable 3D model is described as vertices coordinates. The 3D face model is given by the 3D coordinates of the vertices  $P_i$ ,  $i = 1, 2, 3 \dots m$  where  $m$  is the number of vertices. Thus, the shape up to a global scale can be fully described by the  $3m$ -vector  $\mathbf{g}$  that is the concatenation of the 3D coordinates of all the vertices. The shape with its variations is approximated by a linear relation:

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{S}\boldsymbol{\sigma} + \mathbf{A}\boldsymbol{\alpha}, \quad (1)$$



**Fig. 1.** The CANDIDE-3 wireframe [6]

where  $\mathbf{g}$  is the standard shape of the model, and the columns of matrices  $\mathbf{S}$  and  $\mathbf{A}$  are Shape and Animation Units. A Shape Unit provides a way to deform the 3D wireframe, such as to adapt the eye width, the head width, the eye separation distance, etc.  $\boldsymbol{\sigma}$  and  $\boldsymbol{\alpha}$  are coordinate vectors in the shape and animation spaces. Thus,  $\mathbf{S}\boldsymbol{\sigma}$  accounts for shape variability (inter-person variability) while  $\mathbf{A}\boldsymbol{\alpha}$  accounts for the facial animation (intra-person variability). We assume that the two kinds of variability are independent.

The generic 3D face model is meant to describe various persons with various face expressions [5]. This avoids the need for a statistical model for face shape as originally introduced by Cootes [1]. This is similar to the Direct Appearance Models by Hou [7] where the relationship between the texture and the shape is many to one that is one shape can correspond to many textures, but no texture is associated with more than one shape. They claim that one can thus predict the shape directly from the texture, which leads to a faster and more reliable search algorithm.

## (2) Face Texture

A face texture is a geometrically normalized image of a face that is independent from shape, so appearance variations resulting from shape variations are removed. It is created by first interactively adjusting the face model to an image captured, and then mapping that image onto the model, and finally reshaping the face model to the mean shape in the front view. Normalized face textures all have the same size, 64 by 64 in our experiments.

Given the vector  $\mathbf{b}$  of the model pose and animation parameters and an input image  $\mathbf{y}$ , the face texture  $\mathbf{T}(\mathbf{b})$  is the result of a warping function  $\mathbf{W}$ :

$$\mathbf{T}(\mathbf{b}) = \mathbf{W}(\mathbf{y}, \mathbf{b}), \quad (2)$$

$\mathbf{W}$  is an affine transform with respect to each model mesh triangle. It requires two costly computations for each pixel: its mesh triangle, and the three barycentric coordinates in that triangle. Fortunately, these time-consuming operations can be pre-computed offline [8].

In this step, face textures from various users with various poses and expressions are obtained. They are used to build a statistical model of the face textures with a PCA

which outputs eigenfaces, so describing the texture variation around the mean texture [5]. This can be done either on a single user or dedicated to multiple users. The former is a *person specific model*, while the later is a *generic model* [9].

### (3) Update Matrix

Tracking consists in registering the 3D model onto a sequence of input images  $\mathbf{y}$  so that it minimizes their difference with the projected model texture. Equivalently, but faster because the reference texture is smaller than input images, we rather minimize the residue image  $\mathbf{r}(\mathbf{b})$  between the warped input image  $W(\mathbf{y}, \mathbf{b})$  and the reference face texture  $\mathbf{T}_{\text{model}}$ :

$$\mathbf{r}(\mathbf{b}) = W(\mathbf{y}, \mathbf{b}) - \mathbf{T}_{\text{model}}. \quad (3)$$

$\|\mathbf{r}(\mathbf{b})\|^2$  is to be minimized with respect to  $\mathbf{b}$ , the vector of the pose and animation parameters, that is composed of the global model rotation  $\boldsymbol{\theta}$ , translation  $\mathbf{t}$  and the animation vector  $\boldsymbol{\alpha}$ .

$$\mathbf{b} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \boldsymbol{\alpha}]. \quad (4)$$

This is classically achieved by gradient descent. A first order Taylor expansion gives:

$$\mathbf{r}(\mathbf{b} + \Delta\mathbf{b}) = \mathbf{r}(\mathbf{b}) + \frac{\partial \mathbf{r}(\mathbf{b})}{\partial \mathbf{b}} \Delta\mathbf{b}. \quad (5)$$

Iterations consist in updating the value of  $\mathbf{b}$  with  $\Delta\mathbf{b}$  such that:

$$\Delta\mathbf{b} = \mathbf{U}\mathbf{r}(\mathbf{b}), \quad (6)$$

where:

$$\mathbf{U} = -(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T, \quad (7)$$

$$\mathbf{G} = \frac{\partial \mathbf{r}(\mathbf{b})}{\partial \mathbf{b}}. \quad (8)$$

The update  $\mathbf{U}$  matrix can be computed offline before tracking. It is known as the negative pseudo-inverse of the gradient matrix  $\mathbf{G}$ . It is computed by numerical differentiation, so the  $j^{\text{th}}$  column of  $\mathbf{G}$  is computed with:

$$\mathbf{G}_j = \frac{\mathbf{r}(\mathbf{b} + h\mathbf{q}_j) - \mathbf{r}(\mathbf{b} - h\mathbf{q}_j)}{2h}. \quad (9)$$

where  $h$  is a step value and  $\mathbf{q}_j$  is a vector with all elements zero except the  $j^{\text{th}}$  element that equals to one.

### (4) Relation between the Update Matrix and the Face Texture

In thesis [10], Ahlberg builds a training database from  $N = 330$  images of 6 persons. He also builds the update matrix for various head poses and facial expressions. The estimated gradient is simply estimated as an average of the  $N$  images gradient matrices.

$$\bar{\mathbf{G}}_j = \frac{1}{N} \sum_{n=1}^N \frac{\mathbf{r}(\mathbf{b} + h\mathbf{q}_j) - \mathbf{r}(\mathbf{b} - h\mathbf{q}_j)}{2h}. \quad (10)$$

Here  $n = 1, \dots, N$ , is the images number in the training set with various head poses and facial expressions. Ahlberg assumes that the gradient matrix is similar for the different users, which we find is not a valid assumption.

Let's look at the gradient matrix and the update matrix  $\mathbf{U}$ . It is the negative pseudo-inverse of the gradient matrix  $\mathbf{G}$ . In fact, the matrix  $\mathbf{G}$  is a 3D matrix with the size  $K \times L \times M$ , where  $K$  is the number of parameters,  $L$  is the width of the texture and  $M$  is the height of the texture. To make it easy to analyze matrix  $\mathbf{G}$ , we will separate it into  $K$  channels, so each of the matrix  $\mathbf{G}$  can be described by  $K$  images. These are named Steepest Descent Images (SDIs) [12] because they provide the direction in which the descent toward the optimum is steepest. By looking into the SDIs from different persons, Fig. 2, we can see that the gradient matrix is extremely different from user to user. So averaging user gradient matrices into a single gradient matrix is not suitable. In our experiments, we found that while tracking a user that was learnt works good, averaging multiple users' gradient matrices gives poor tracking results (see Fig. 4, first row), because the gradient matrix is highly related to the user.

Since the gradient matrix is user specific, it must be adjusted for each user. This is usually achieved with tedious training with the user to be tracked. Even worse, this prevents tracking unknown users. Tracking multiple or unknown users requires adjusting the gradient matrix to each user. Such an approach is supported by the intuitive evidence that the gradient matrix is related to the face texture, because the texture can be directly used to predict the shape [7].

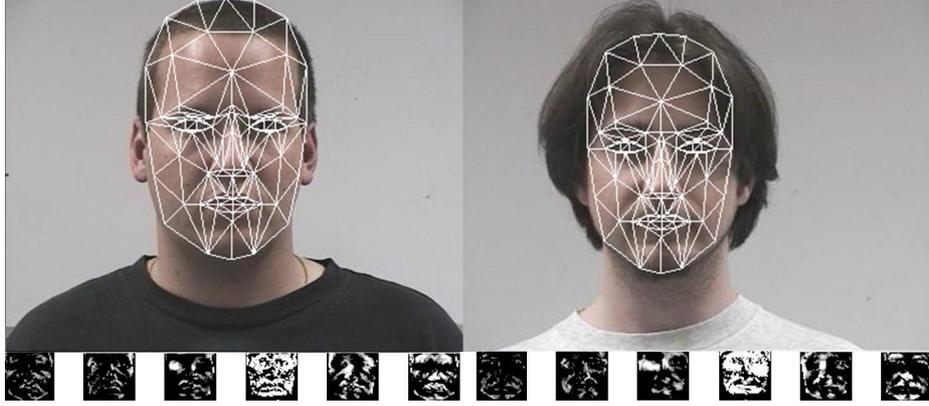
Matthews and Baker [12] have pointed out that there is linear relationship between  $\mathbf{G}$  and  $\nabla_{\mathbf{x}}\mathbf{T}$ . Since the user specific texture  $\mathbf{T}_{\text{model}}$  does not vary during tracking, so equations (8) can be rewritten:

$$\mathbf{G} = \frac{\partial \mathbf{T}(\mathbf{b})}{\partial \mathbf{b}} = \nabla_{\mathbf{x}}\mathbf{T} \cdot \nabla_{\mathbf{b}}\mathbf{x}. \quad (11)$$

where  $\mathbf{x}$  is the pixel coordinate in the texture, so  $\nabla_{\mathbf{x}}\mathbf{T}$  is the image gradient of the face texture.  $\nabla_{\mathbf{x}}\mathbf{T}$  holds all the information related to "color" in the gradient matrix  $\mathbf{G}$ , for example user face color, illumination condition and so on. The second part  $\nabla_{\mathbf{b}}\mathbf{x}$  is only related to the geometrical transform that is the "shape" part of  $\mathbf{G}$ . So the relationship between  $\mathbf{G}$  and  $\nabla_{\mathbf{x}}\mathbf{T}$  appears to be independent from either illumination conditions or the user being tracked.

We propose to model statistically the dependency (11) of  $\mathbf{G}$  on  $\nabla_{\mathbf{x}}\mathbf{T}$  so that a gradient matrix can be estimated from any user face texture that is readily available while tracking. Learning is achieved on some database of face images. For each image  $n$  in the training database, we manually set the position, rotation and the animation parameters. With those parameters, we extract texture  $\mathbf{T}_n$ , and compute the corresponding

gradient matrix  $\mathbf{G}_n$  for image  $n$  using expression (8). We use PCA on both the texture gradient and the gradient matrices:



**Fig. 2.** Example of Steepest Descent Images from two users in the training database [16]. The left image is the training image with the model registered manually. Each set of 6 images on the bottom is the steepest descent images for the 6 head pose.

$$\begin{aligned}\nabla_x \mathbf{T} &= \overline{\nabla_x \mathbf{T}} + \mathbf{E}_T \mathbf{p}_T \\ \mathbf{G} &= \overline{\mathbf{G}} + \mathbf{E}_G \mathbf{p}_G\end{aligned}\quad (12)$$

where  $\mathbf{E}_T$  and  $\mathbf{E}_G$  are the eigenvectors of texture gradient space and gradient matrix space and  $\mathbf{p}_T$  and  $\mathbf{p}_G$  are eigenvalue vectors.

For each example we generate the concatenated vector:

$$\mathbf{p} = \begin{pmatrix} \mathbf{p}_T \\ \mathbf{p}_G \end{pmatrix}.\quad (13)$$

Because  $\mathbf{p}$  has a zero average value, a further PCA gives:

$$\mathbf{p} = \mathbf{E}_c \mathbf{c},\quad (14)$$

where:

$$\mathbf{E}_c = \begin{pmatrix} \mathbf{E}_{cT} \\ \mathbf{E}_{cG} \end{pmatrix}.\quad (15)$$

By substituting  $\mathbf{p}_T$  and  $\mathbf{p}_G$  back in (12) we get:

$$\begin{aligned}\nabla_x \mathbf{T} &= \overline{\nabla_x \mathbf{T}} + \mathbf{E}_T \mathbf{E}_{cT} \mathbf{c} = \overline{\nabla_x \mathbf{T}} + \mathbf{Q}_T \mathbf{c} \\ \mathbf{G} &= \overline{\mathbf{G}} + \mathbf{E}_G \mathbf{E}_{cG} \mathbf{c} = \overline{\mathbf{G}} + \mathbf{Q}_G \mathbf{c}\end{aligned}\quad (16)$$

where:

$$\begin{aligned} \mathbf{Q}_T &= \mathbf{E}_T \mathbf{E}_{cT} \\ \mathbf{Q}_G &= \mathbf{E}_G \mathbf{E}_{cG} \end{aligned} \quad (17)$$

So we model  $\nabla_x \mathbf{T}$  and the  $\mathbf{G}$  with a single coordinates vector  $\mathbf{c}$ . With this function we can predict the user specific gradient matrix from the current texture that was captured from the input video sequence.

## 2.2 Online Tracking: Active Model Search

### (1) Initialization

The fitting algorithm of AM may get trapped in some local minima so it requires coarse initialization. Thanks to the robustness of the AM, in most cases a face detection algorithm can meet the conditions. A survey on face detection can be found in [11]. We used the AdaBoost approach proposed by Viola and Jones [13], the implementation is freely available from the OpenCV library [14].

### (2) On line Tracking Flow

For each iteration, the parameter update  $\Delta \mathbf{b}$  is computed using expression (6). Notice that the update matrix  $\mathbf{U}$  is now dynamically calculated from input texture. Given a texture  $\mathbf{T}$ , function (16) provides the combined texture and gradient coordinates in the PCA space:

$$\mathbf{c} = \mathbf{Q}_T (\nabla \mathbf{T} - \overline{\nabla \mathbf{T}}), \quad (18)$$

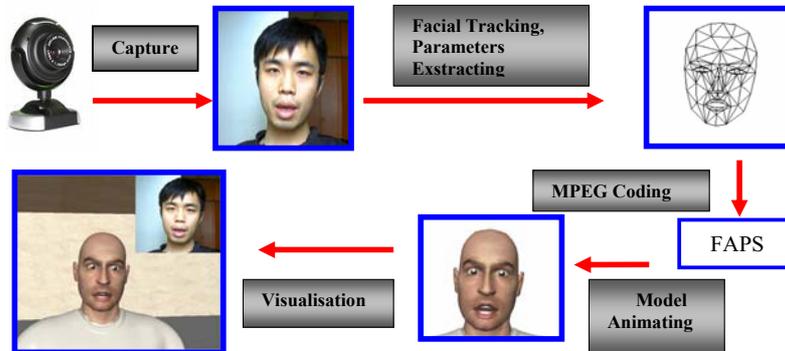
The predicted gradient matrix can be calculated using function (16). Notice that the gradient matrix is updated dynamically during tracking, so we call it dynamical AM while the classical AM relies on a constant gradient matrix.

## 3 Implementation

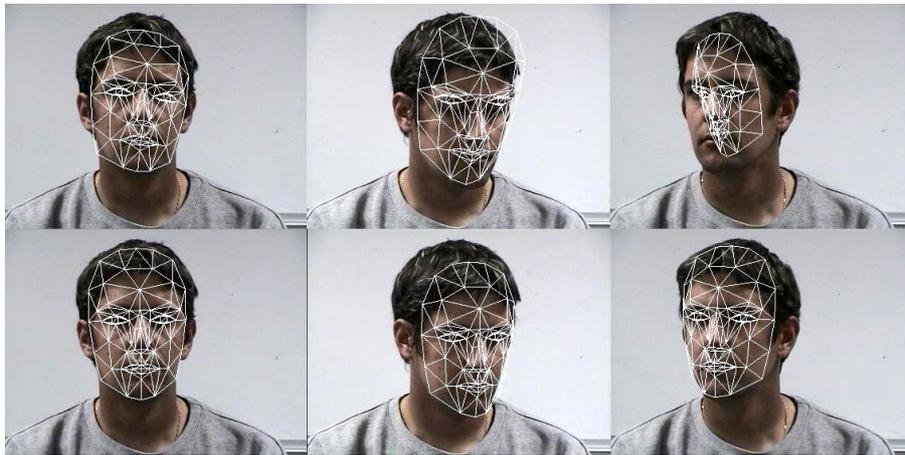
The algorithm has been implemented on a low-end PC with an Intel 1.8 GHz Pentium IV processor and 512 MB memory. We use a consumer web camera to grab the video sequences. OpenGL is used for texture mapping and OpenCV for video capture and processing. We retained the following six animation parameters of the Candide model for tracking the facial gestures described in [4]:

- |                       |                           |
|-----------------------|---------------------------|
| (1) upper lip raiser, | (4) lip corner depressor, |
| (2) jaw drop,         | (5) eyebrow lowerer,      |
| (3) mouth stretch,    | (6) outer eyebrow raiser. |

Based on the algorithm described above, six head position parameters and six animation parameters are separately trained and tracked. For comparison purposes, both the original and the dynamical AM have been implemented. They run at 25 and 20 frames / second, respectively. The dynamical AM still can work in near real time since dynamically predicting the gradient matrix only requires a further PCA in a low dimensional space.



**Fig. 3.** Render GRETA using dynamical AM face tracking (video available from the website: <https://picoforge.int-evry.fr/projects/svn/myblog3d/video/Gretafaceanimation.avi>)



**Fig. 4.** Example for the tracking result using the DAM algorithm from top to bottom: tracking using averaging multiple users' gradient matrices and tracking with dynamically predict the gradient matrix (images from the Head Pose Image Database [16])

The captured face animation can be converted to MPEG-4 facial animation parameters (FAPs) [4] for rendering with an avatar e.g. the free MPEG-4 player named GRETA [15].

## 4 Experimental Results

In our experiments we use the Head Pose Image Database [16] for training the global parameter face position, rotation and scale, and the BIOMET database [17] for training the animation parameters.

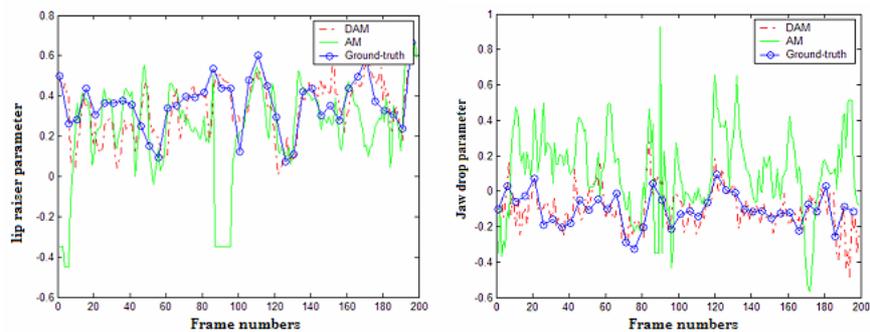
Experiments on head pose tracking were first achieved with training with ten persons with varying skin color and 7 images per person. Fig. 4 presents some tracking

results. We find that the dynamical AM algorithm gives better result than the algorithm averaging multiple users' gradient matrices.



**Fig. 5.** Dynamically tracking on video sequence (video available at: <https://picoforge.int-evry.fr/projects/svn/myblog3d/video/Facetracking.avi>)

In our second experiments (Fig. 5), a person is speaking with lots of facial expressions and head motions. We can observe that even if the background is complex, the facial expressions and the head pose are well recovered. Note that those results are obtained without any previous camera calibration or manual initialization. The AdaBoost-based initialization is done only on the first frame of the sequence, without any interactive operation.



**Fig. 6.** Tracking result of upper lip raiser animation parameter (left) and jaw drop parameter (right)

In order to accurately evaluate the accuracy of tracking system, we mutually set the parameter for the test sequence and as ground-truth. Both the trackers of AM and

dynamical AM have been applied to the video, and we compare all the twelve head position parameters and animation parameters to the true values. The result of two parameters: upper lip raiser and jaw drop are plotted in (Fig. 6), and the root mean square error (RMSE) between the tracking result and the ground-truth is given by Table 1. From the result we can see the dynamical AM algorithm gives better result. Note in the fig. 6 the tracking is lost for the original AM algorithm from the 60<sup>th</sup> to the 140<sup>th</sup> frame.

**Table 1.** The root mean square error (RMSE) between animation parameter (AP) tracking result with the ground-truth. The AP(1~6) is noted for (1) upper lip raiser, (2) jaw drop, (3) mouth stretch, (4) lip corner depressor, (5) eyebrow lowerer, (6) outer eyebrow raiser separately.

Algorithm	RMSE AP(1)	RMSE AP(2)	RMSE AP(3)	RMSE AP(4)	RMSE AP(5)	RMSE AP(6)
AM	0.0534	0.0967	0.0586	0.0394	0.0250	0.0211
DAM	0.0166	0.0206	0.0112	0.0310	0.0093	0.0091

From these experiments, the dynamical AM algorithm is more robust and accurate than AAM for unseen users. Actually, the illumination and camera parameters are “embedded” in the update matrix. This makes the algorithm work robustly when training and tracking the faces in the same acquisition conditions, but it achieves poor results in case of uncontrolled lighting or variation of the camera parameters. Ideally the illumination could be corrected with eigenlight-fields and Bayesian face sub-regions [18] and the camera parameters could be calculated as optimization problem. But both of those processes are not compatible with real-time processing.

## 5 Conclusion and Future Work

In this paper, we propose an enhancement to the active model approach for real-time tracking the face of unknown users. The main aim of the work is to exploit the existent video database to improve the tracking robustness. The whole flow of the system: initialization, training and tracking is introduced. Our experimental result demonstrate the improvement of the dynamically AM compare to the classical AM. The tracking algorithm has been applied for animating avatar by using MPEG-4 as a common communication and animation standard. The future research work will be focus on facial expressions analysis and emotion recognition from dynamic images by using our face tracking system.

## Acknowledgment

We wish to thank Professor Catherine Pelachaud for providing GRETA and her assistance in using it. And also Dr. Jörgen Ahlberg gives us the CANDIDE-3 face model. At last I would like to thank Dr. Dijana Petrovska who give me suggestion in my whole work period and kindly proved the BIOMET Database.

## References

1. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(6), 681–685 (2001)
2. Xiao, J., Baker, S., Matthews, I., Kanade, T.: Real-time combined 2d+3d active appearance models. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 25, pp. 535–542 (2004)
3. Wiskott, L., Fellous, J.M., Kruger, N., Malsburg, C.: Face recognition by elastic bunch graph matching, Tech. Rep. IR-INI 96–08 (1996)
4. Dornaika, F., Ahlberg, J.: Fast and Reliable Active Appearance Model Search for 3D Face Tracking. *IEEE Transactions on Systems, Man, and Cybernetics–Part 34*, 1838–1853 (2004)
5. Turk, M., Pentland, A.: Eigenfaces for Recognition. *Journal of Cognitive Neuroscience* 3(1), 71–86 (1991)
6. Ahlberg, J.: Candide-3 – an updated parameterized face. Technical Report LiTH-ISY-R-2326, Linköping University, Sweden (2001)
7. Hou, X.W., Li, S.Z., Zhang, H.J.: Direct appearance models. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 828–833 (2002)
8. Ahlberg, J.: Real-Time Facial Feature Tracking Using an Active Model With Fast Image Warping. In: *International Workshop on Very Low Bitrates Video*, pp. 39–43 (2001)
9. Gross, R., Matthews, I., Baker, S.: Generic vs. person specific active appearance models. *Image and Vision Computing* 23(11), 1080–1093 (2005)
10. Ahlberg, J.: Model-based Coding - Extraction, Coding, and Evaluation of Face Model Parameters, PhD Thesis (2002)
11. Yang, M.H., Kriegman, D., Ahuja, N.: Detecting Faces in Images: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 34–58 (2002)
12. Matthews, I., Baker, S.: Active appearance models revisited. *International Journal of Computer Vision* 60(2), 135–164 (2004)
13. Viola, P., Jones, M.J.: Robust real-time object detection. Cambridge Research Laboratory, Technical Report Series (2001)
14. Bradski, G., Kaehler, A.: Pisarevsky: Learningbased computer vision with intel’s open source computer vision library. *Intel Technology Journal* 9(2), 1 (2005)
15. Poggi, I., Pelachaud, C., Derosis, F., Carofiglio, V., Decarolis, B.: GRETA. A Believable Embodied Conversational Agent. In: Stock, O., Zancarano, M. (eds.) *Multimodal Intelligent Information Presentation*. Kluwer, Dordrecht (2005)
16. Gourier, N., Hall, D., Crowley, J.L.: Estimating Face Orientation from Robust Detection of Salient Facial Features. In: *International Workshop on Visual Observation of Deictic Gestures* (2004)
17. Salicetti, S., Beumier, C., Chollet, G., Dorizzi, B., Jardins, J.L.I., Lunter, J., Ni, Y., Petrowska Delacretaz, D.: BIOMET: A multimodal person authentication database including face, voice, fingerprint, hand and signature modalities. In: Kittler, J., Nixon, M.S. (eds.) *AVBPA 2003. LNCS*, vol. 2688, pp. 845–853. Springer, Heidelberg (2003)
18. Gross, R., Baker, S., Matthews, I., Kanade, T.: Face Recognition Across Pose and Illumination. In: Li, S.Z., Jain, A.K. (eds.) *Handbook of Face Recognition*, Springer, Heidelberg (2004)