A Head Pose and Facial Actions Tracking Method Based on Effecient Online Appearance Models *

Xiaoyan Wang[1], Xiangsheng Huang [2], Huiwen Cai[2], Xin Wang[1]

[1]College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou,310023,China
{xiaoyanwang, xinw}@zjut.edu.cn

[2]Institute of Automation
Chinese Academy of Sciences
Beijing,100190, China
{xiangsheng.huang, huiwencai}@ia.ac.cn

Abstract: Target modeling and model fitting are the two important parts of the problem of object tracking. The former has to provide a good reference for the latter. Online appearance models (OAM) has been successfully used for facial features tracking on account of their strong ability to adapt to variations, however, it suffers from time-consuming model fitting. Inverse Compositional Image Alignment (ICIA) algorithm has been proved to be an efficient, robust and accurate fitting algorithm. In this work, we introduce an efficient online appearance models based on ICIA, and apply it to track head pose and facial actions in video. A 3d parameterized model, CANDIDE model, is used to model the face and facial expression, a weak perspective projection method is used to model the head pose, an adaptive appearance model is built on shape free texture, and then the efficient fitting algorithm is taken to track parameters of head pose and facial actions. Experiments demonstrate that the tracking algorithm is robust and efficient.

Key–Words: Visual tracking, Online appearance models, Inverse Compositional Image Alignment, model learning, facial feature tracking

1 Introduction

The tracking of head pose and facial actions in video is one of the significant problems in fields of computer vision and graphics. It plays an important role in applications such as Human-Computer Interaction, surveillance, entertainment, and is highly relevant to the techniques of facial expression analysis, face recognition, realistic 3D face model generation, etc. The aims of facial feature tracking include tracking the rigid movement of head and the nonrigid transformation caused by expression and actions, which make it quite challenging.

It is a key problem of visual tracking to adapt the temporal appearance and background changes. However, most of the algorithms can only track the object under a control environment for a short time, and easy to fail when great variation appears. Simple hypothesis of no significant object variations was made as a prerequisite, although the problem can be fixed by trying some more expressive features or adding more effective predictions. Unlike the fixed template models and statistical models, Online Appearance Model combines the training stage with the searching stage. It don't need training samples and can adapt to appearance changes very well with the online learning.

Object tracking is one of the important and challenging problems in computer vision, and has many applications, such as actor-driven animation [12] and video surveillance [13]. The two main portions of tracking problems are target modeling and model fitting. Target model should have the ability to handle object changes caused by different reasons, and provide a proper reference for the fitting. Meanwhile fitting algorithms need to be promoted according to the four aspects presented in [8]: efficiency, robustness, accuracy and automation. Various algorithms have been proposed and improved to meet these requirements [14][15][16]. However most of these methods emphasize particularly only on one aspect. In this work, an efficient online appearance models (EOAM) based on ICIA is presented to improve both the object model and the fitting.

Fixed template models cannot adapt to appearance changes, while statistical texture models are restricted by training sets and will fail if the imaging conditions are significantly changed. Recently online-learning methods become very popular. They combine the training stage with the searching part and have achieved very good results. Adaptive Gaus-

*This work was supported by the National Natural Science Foundation of China (NSFC-60870002, 60573123), NCET, and the Science and Technology Department of Zhejiang Province (2009C21008, Y1090592,Y200907765)

sian mixture methods have been chosen for real-time video background substraction and object tracking [4][3][17][2], because of its good features in both theory and realization. Jepson et al. [1] introduced one kind of adaptive Guaussian mixture model named online appearance model (OAM) for visual tracking. It is a generative model which combines both stable and motion constraints. To our advantage, it earns the ability to adapt to appearance variations caused by head pose and face expression.

When it comes to the fitting part, an EM algorithm is used in [1]. A gradient descent method and particle filters are combined to track the parameters in [2][5]. The main disadvantage of these methods is time-consuming. Inverse Compositional Image Alignment (ICIA) is an efficient algorithm proposed by [6]. It has been used to improve the search efficiency of AAM [7] and extended to 3D morphable models by Romdhani and Vetter [8]. Both applications have proved that the algorithm is efficient, robust and accurate.

In this study, we develop an efficient tracking framework based on OAM and ICIA, and apply it to track head pose and facial features. The highlights of this paper are described as follows:

1. The target model is an adaptive mixture appearance model established from the observation model, which is a modified OAM. The mixture Gaussian appearance model is built the same as [2], while the model learning is improved.

2. According to the mixture Gaussian model, the observation expectation is evaluated as an reference for the fitting, and the cost function is defined based on the log-likelihood of observation appearance.

3. A 3D wire-frame model is used to model the facial actions, and a weak perspective projection model is built to model head pose. The parameters of head pose are redefined for an unified use of ICIA.

4. Given coefficients, a piece-wise affine transform is taken to warp the image and get an observation. Jacobian is evaluated on condition that the warping operation is divided into three steps.

The paper is structured as follows: in the next section, we will establish an efficient tracking framework for combining a modified online appearance model with Inverse Compositional Image Alignment algorithm. Section 3 presents the details of how to build an observation and apply the combined algorithm to track head pose and facial features. Section 4 presents experimental results from an implementation of the algorithm. Finally we draw a conclusion of this work and have a discussion of the future work.

2 An efficient online appearance models (EOAM)

In this part, we describe a tracking framework of combining online appearance model with Inverse Compositional Image Alignment algorithm. A modified version of OAM is used to model the appearance and then ICIA is adopted to fit the model.

2.1 Online Appearance Models

Three Gaussian models are used in the original online appearance models presented in [1]. The first component S depicts the stable image observation, the second part L represents data outliers, and the third part W accounts for the two-frame variation. Occlusion can be modeled in other easy ways, using robust statistics for example. Therefore according to the WSF model proposed in [2], a fixed template F takes the place of the original 'lost' component L. It can be a shape-free facial texture without expression from a frontal view in facial expression tracking.

2.1.1 WSF Mixture Appearance Model

By assuming that the pixels of object observation are independent of each other, the WSF appearance model at time t, is defined as a mixture of three Gaussians $A_t = \{W_t, S_t, F_t\}$, with mixture centers $\{\mu_{i,t}, i = w, s, f\}$, and corresponding variances$\{\sigma^2_{i,t}, i = w, s, f\}$, and the mixing probabilities $\{\omega_{i,t}, i = w, s, f\}$. Note that $\{\mu_{i,t}, \sigma^2_{i,t}, \omega_{i,t}, i = w, s, f\}$ are all d-vectors, where d is the length of the observation. Suppose that the observation is Y_t, its likelihood is written as:

$$p(Y_t|\theta_t) = \textstyle\prod_{j=1}^{d}\{\sum_{i\in\{w,s,f\}} p(G_i)p(Y_t|G_i)\}$$

$$= \prod_{j=1}^{d}\{ \sum_{i\in\{w,s,f\}} \omega_{i,t}(j)N(Y_t(j);\mu_{i,t}(j),\sigma^2_{i,t}(j))\} \tag{1}$$

where G_i represents the three Gaussian components, $N(x;\mu,\sigma^2)$ denotes a normal density:

$$N(x;\mu,\sigma^2) = (2\pi\sigma^2)^{-1/2}exp\{-\varphi(\frac{x-\mu}{\sigma})\} \tag{2}$$

$$\varphi(x) = \frac{1}{2}x^2 \tag{3}$$

2.1.2 Model learning

Initially, after an observation Y_0 is obtained by a detection algorithm, we set A_1 with Y_0. To keep the generative mixture model adaptive, its parameters, namely the means, variances and mixing probabilities, need to be updated when a new observation Y_t is available. The learning process of WSF mixture model is described as follows:

1. Compute the probabilities of new observation Y_t under each gaussian model:

$$p_{i,t} = \begin{cases} \omega_{i,t} \cdot N(Y_t; \mu_{i,t}, \sigma_{i,t}^2), if|\frac{Y_t - \mu_{i,t}}{\sigma_{i,t}}| < T_\sigma \\ \\ 0 \end{cases} \tag{4}$$

where $i \in \{s, f, w\}$, and T_σ is a control variable.

2. If $\sum_j p_j > 0$, compute the expected posterior of each gaussian model:

$$q_{i,t} = p_{i,t} / \sum_{j \in \{s,f,w\}} p_{j,t}; i \in \{s, f, w\} \tag{5}$$

Then update the mixing probabilities as:

$$\omega_{i,t+1} = (1 - \alpha) \cdot \omega_{i,t} + \alpha \cdot q_{i,t} \tag{6}$$

where α is the learning rate.

If $p_{s,t} > 0$, update the mean and variance of the stable model as follows:

$$c_{t+1} = c_t + q_{s,t} \tag{7}$$

$$\mu_{s,t+1} = (1 - \eta) \cdot \mu_{s,t} + \eta \cdot Y_t, \tag{8}$$

$$\sigma_{s,t+1}^2 = (1 - \eta) \cdot \sigma_{s,t}^2 + \eta \cdot (Y_t - \mu_{s,t})^2 \tag{9}$$

where $\eta = q_{s,t} \cdot (\frac{1-\alpha}{c_{t+1}} + \alpha)$

3. The parameters of W-component and F-component are set as:

$$\mu_{w,t+1} = Y_t, \sigma_{w,t+1}^2 = \sigma_{s,1}^2 \tag{10}$$

$$\mu_{f,t+1} = \mu_{f,1}, \sigma_{f,t+1}^2 = \sigma_{f,1}^2 \tag{11}$$

The basic learning procedure of the weights and component S follows the formulation in [3]. Note that the variable c_t counts the number of effective observations for the S-component, and is used to compute an appropriate learning rate.

2.2 Fitting the model to an image

The adaptive appearance model is demonstrated in the context of object tracking. Given an input image, a fitting algorithm is required to optimize the coefficients we are going to track. Inverse Compositional Image Alignment (ICIA) is an efficient one proposed by Baker and Matthews [6]. It has been applied to Active Appearance Model to improve its search performance and has achieved good results [7]. When it comes to the object tracking, we encounter the same problem as AAM search. Generally speaking, gradient descent is a good approach to register the appearance model to a new input image. However it is very time consuming if the gradient matrix is recomputed in each iteration, like in [5]. ICIA algorithm gives a key advantage of pre-calculating the derivatives. Consequently it is a good choice for us.

2.2.1 Observation expectation

Now we would like to determine which portion of the mixture model A_t are most likely to describe Y_t. In [2], the observation of last frame is simply used as the template for tracking. However Gaussian distributions with the most supporting evidence and the least variance are chosen as the background model in [4]. In respect that the variances of two Gaussian components are not updated and higher mixing probability implies more feasibility of happening, the Gaussian with the maximum mixing probability can work equally well [3]. To be more precise, the expected value of observation is adopted here as both the representation of the mixture model and the template T in ICIA algorithm:

$$T = E(Y_t) = \textstyle\sum_{i \in \{s,f,w\}} E(Y_t|G_i)p(G_i)$$

$$= \sum_{i \in \{s,f,w\}} \omega_i \cdot \mu_i \tag{12}$$

2.2.2 Cost function

Our goal is to estimate the optimal parameters of observation model from input image. Assuming that the observation model at time t is denoted by $Y_t = W(X, \rho)$, where X is the input data, and ρ is the coefficient of some warping operation W to project the data to an observation. The log-likelihood of Y_t can be expressed as

$$L(Y_t|A_t) = \sum_{j=1}^{d} log \left[\sum_{i=w,s,f} \omega_{i,t}(j) N(Y_t(j); \mu_{i,t}(j), \sigma_{i,t}^2(j)) \right] \tag{13}$$

Thus the cost function can be defined as:

$$e_t(\rho) = \frac{2}{d} \sum_{j=1}^{d} \sum_{i=w,s,f} \left\{ \omega_{i,t}(j) \cdot \varphi \left[\frac{Y_t(j) - \mu_{i,t}(j)}{\sigma_{i,t}(j)} \right] \right\} \tag{14}$$

2.2.3 Optimization procedure

In this section, a framework of combing ICIA with OAM is built. As we mentioned before, ICIA is used to optimize the parameters of observation and the expectation of observation is its template. By reason that the template is time-varying, only the Jacobians can be calculated at the beginning. The whole process can be described as follows:

1. Pre-compute the Jacobian $\frac{\partial W}{\partial \rho}$ when $\rho = 0$;
2. Initialization:
$$\mu_{i,1} = Y_0, c_1 = 0; i \in \{s, f, w\} \tag{15}$$
3. At time t, set the texture model T for ICIA, in accordance with equation (12).
4. Evaluate the gradient $\bigtriangledown T$ of the template T;
5. Evaluate the steepest descent images SD and the Hessian matrix H:
$$SD = \bigtriangledown T \cdot \frac{\partial W}{\partial \rho} \tag{16}$$
$$H = \sum_x \left[\bigtriangledown T \cdot \frac{\partial W}{\partial \rho} \right]^T \left[\bigtriangledown T \cdot \frac{\partial W}{\partial \rho} \right] \tag{17}$$
6. Iterate until converged:
   (a) Normalize the input data using some warp operation with the current parameters ρ, and get a new observation denoted as Y_c.
   (b) The new observation has to register with the current mixture model A_t, and the error is evaluated according to equation (14). Supposing that e_t records the error of last ρ_c, and if $e_c(\rho_c) < e_t$, update the error and the parameter:
$$e_t = e_c(\rho_c), \rho_t = \rho_c, Y_t = Y_c \tag{18}$$
   If $e_c(\rho_c) \geq e_t$, the error can no longer be reduced, thus convergence is declared, then parameters ρ_t and observation Y_t are returned.
   (c) Compute the shift of parameters:
$$\Delta\rho_c = H^{-1} \sum_x SD^T \cdot [Y_c - T] \tag{19}$$
   (d) Compose the incremental warp with the current warp $W(x, \rho_t) \circ W(x, \Delta\rho_c)^{-1}$ and get a new parameter $\rho_c{}'$. Small steps are performed to the update: $\rho_c \leftarrow \rho_c + \lambda(\rho_c{}' - \rho_c)$ with a factor $\lambda \ll 1$.
7. Update the mixture appearance model A_{t+1} using Y_t, according to the steps in section 2.1.2, and set t = t + 1, then return to step 3.

3 Head pose and facial expression tracking

So far, the efficient online appearance models based on ICIA has been detailed. Now we are interested in applying it to head pose and facial actions tracking.

3.1 Building image observation

We explore the application of automatic facial action tracking. To this end, both facial expression and head-pose have to be recovered. In this section, models of facial action and head-pose are built, and then an observation is obtained from them.

3.1.1 Modeling the facial actions

Our study is based on a 3D wire-frame face model Candide built by Ahlberg [10], which is designed to depict the diversity between different human beings as well as different face expressions. The model is given as:

$$g = \overline{g} + \sum_{i=1}^{n_s} s_i \cdot S_i + \sum_{i=1}^{n_a} a_i \cdot A_i \tag{20}$$

where $\overline{g}$ is the 3D standard shape, S_i denote the shape modes, A_i denote the animation modes, n_s and n_a are the numbers of shape and animation modes used respectively, s_i and a_i are the respective parameters.

Note that the shape modes represent inter-person variety, and are ascertained by some detection procedure at the beginning. Therefore, only the tracking of animation parameters are considered:

$$a = (a_1, a_2, \ldots, a_{n_a})^T \tag{21}$$

This procedure is regarded as a 3D to 3D projection:

$$M((x, y, z); a) \rightarrow (x, y, z) \tag{22}$$

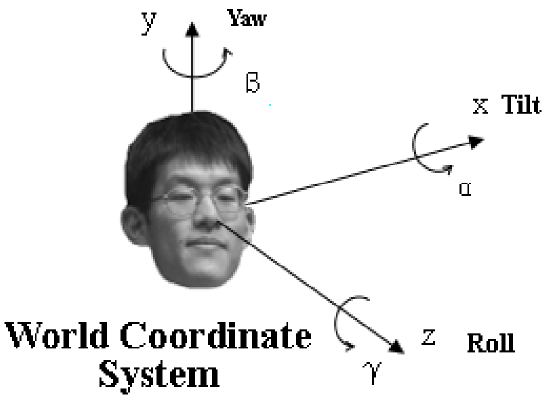

Figure 1: Coordinate system and pose parameters used in our approach

3.1.2 Modeling the head-pose

After the face model is built, it is necessary to project the 3D wire-frame to 2D image coordinate system. A weak perspective projection model is used:

$$g' = f \cdot R \cdot (\overline{g} + \sum_{i=1}^{n_s} s_i \cdot S_i + \sum_{i=1}^{n_a} a_i \cdot A_i) + t \quad (23)$$

where f is the camera focal length, $t = (t_x, t_y)$ is denoted as the translation vector. Figure 1 shows the coordinate system used in our approach. Supposing that R' is the 3×3 matrix based on the three rotation angles around the respective axes, and is denoted as following:

$$R' = R_\alpha \cdot R_\beta \cdot R_\gamma = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \quad (24)$$

where the three rotations around each axis are defined as:

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \quad (25)$$

$$R_\beta = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \quad (26)$$

$$R_\gamma = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (27)$$

The projection matrix R is denoted by:

$$R = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{bmatrix} \quad (28)$$

After the translation, a 2D mesh g' is obtained. To try to unify the operations for easy computation, we define some new parameters as following:

$$q_i = \begin{cases} f \cdot a_i - 1, (i = 1, 5) \\ a_i, (i = 2, 3, 4, 6) \end{cases} \quad (29)$$

$$q_7 = t_x, q_8 = t_y \quad (30)$$

The corresponding mesh vectors are defined as:

$$g_0^* = (x_1, y_1, x_2, y_2, \cdots, x_v, y_v)^T \quad (31)$$

$$g_1^* = (x_1, 0, x_2, 0, \cdots, x_v, 0)^T \quad (32)$$

$$g_2^* = (y_1, 0, y_2, 0, \cdots, y_v, 0)^T \quad (33)$$

$$g_3^* = (z_1, 0, z_2, 0, \cdots, z_v, 0)^T \quad (34)$$

$$g_4^* = (0, x_1, 0, x_2, \cdots, 0, x_v)^T \quad (35)$$

$$g_5^* = (0, y_1, 0, y_2, \cdots, 0, y_v)^T \quad (36)$$

$$g_6^* = (0, z_1, 0, z_2, \cdots, 0, z_v)^T \quad (37)$$

$$g_7^* = (1, 0, 1, 0, \cdots, 1, 0)^T \quad (38)$$

$$g_6^* = (0, 1, 0, 1, \cdots, 0, 1)^T \quad (39)$$

Then the form of weak perspective projection model is changed to:

$$N((x, y, z); q) \rightarrow g^*(x, y) = g_0^* + \sum_{i=1}^{6} q_i \cdot g_i^* \quad (40)$$

At the same time the number of pose parameters to be tracked increases to eight from the original six. This is not considered to be a problem, as the computation of the projection has become more direct. In conclusion, the tracking problem now consists of updating the animation parameters a_i and the eight projecting parameters q_i. The total parameters of both pose and expression are denoted as:

$$\rho = [q, a]^T = [q_1, q_2, \ldots, q_8, a_1, a_2, \ldots, a_{n_a}]^T \quad (41)$$

3.1.3 Image warping

To obtain an normalized texture observation, we make the standard shape $\overline{g}$ as the reference mesh by projecting it onto the image system using a centered frontal pose. Afterward each input image I is warped so that the vertices of new mesh with parameter ρ match the corresponding ones of the reference mesh.

The reference frames of the new mesh and the base mesh are considered to be different, and denoted as (x, y) and (u, v) respectively. A mesh is set of triangles. Suppose that the three vertices of each pixel

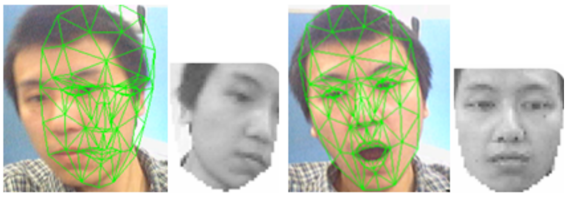

Figure 2: Example of observation corresponding to a wrong model parameter and a correct model parameter

$(u,v)^T$ in reference mesh are $(u_i,v_i)^T$, $(u_j,v_j)^T$ and $(u_k,v_k)^T$. The corresponding vertices in new mesh are $(x_i,y_i)^T$, $(x_j,y_j)^T$ and $(x_k,y_k)^T$. Then the pixel $(u,v)^T$ can be expressed as:

$$(u,v)^T = \lambda_1(u_i,v_i)^T + \lambda_2(u_j,v_j)^T + \lambda_3(u_k,v_k)^T \tag{42}$$

where λ_1, λ_2 and λ_3 are the barycentric coordinates and satisfy:

$$0 \leq \lambda_1, \lambda_2, \lambda_3 < 1; \lambda_1 + \lambda_2 + \lambda_3 = 1 \tag{43}$$

A piece-wise affine transform is used, and the point simply maps to:

$$(x,y)^T = \lambda_1(x_i,y_i)^T + \lambda_2(x_j,y_j)^T + \lambda_3(x_k,y_k)^T \tag{44}$$

Finally, a shape-free image patch as the observation is obtained:

$$W((x,y);\rho) \rightarrow (u,v) \tag{45}$$

$$Y(u,v) = I(W((x,y);\rho)) = I(x,y) \tag{46}$$

According to [11], all the barycentric coordinates can be pre-calculated to reduce the CPU time of warping process. Figure 2 shows a example of observation corresponding to a wrong model parameter and a correct model parameter.

What is left to do now is to compute the Jacobian and incremental warp inversion and then get new parameters, when using ICIA. These steps are detailed in next two sections.

3.2 Computing the Jacobian

The warping operation here consists of three steps and can be expressed as:

$$W \circ N \circ M((x,y);\rho) = W(N(M((x,y,z);a);q);\rho) \tag{47}$$

Therefore, to compute the Jacobian at a point (x, y) when $\rho = 0$, we can get the following equations:

$$\frac{\partial W \circ N \circ M((x,y);0)}{\partial a} = \frac{\partial W \circ M((x,y,z);0)}{\partial a} \tag{48}$$

$$\frac{\partial W \circ N \circ M((x,y);0)}{\partial q} = \frac{\partial W \circ N((x,y);0)}{\partial q} \tag{49}$$

Supposing that there are ν vertices composing the 3D wireframe and the 2D mesh, which are denoted as:

$$g = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_\nu, y_\nu, z_\nu) \tag{50}$$

$$g^* = (x_1, y_1, x_2, y_2, \ldots, x_\nu, y_\nu) \tag{51}$$

Consequently the chain rule is applied to warp and gives $\frac{\partial W \circ M}{\partial a}$ as:

$$\sum_{i=1}^{\nu}\left[\frac{\partial W \circ M}{\partial x_i}\frac{\partial x_i}{\partial a} + \frac{\partial W \circ M}{\partial y_i}\frac{\partial y_i}{\partial a} + \frac{\partial W \circ M}{\partial z_i}\frac{\partial z_i}{\partial a}\right] \tag{52}$$

where $\frac{\partial W \circ M((x,y,z);0)}{\partial x_i}$, $\frac{\partial W \circ M((x,y,z);0)}{\partial y_i}$ and $\frac{\partial W \circ M((x,y,z);0)}{\partial z_i}$ are determined by if (x_i, y_i, z_i) is one vertex of the triangle (x,y,z) belonged to. $\frac{\partial x_i}{\partial a}$, $\frac{\partial y_i}{\partial a}$ and $\frac{\partial z_i}{\partial a}$ are the corresponding values in animation modes A.

The same way is taken to calculate $\frac{\partial W \circ N}{\partial q}$:

$$\frac{\partial W \circ N}{\partial q} = \sum_{i=1}^{\nu}\left[\frac{\partial W \circ N}{\partial x_i}\frac{\partial x_i}{\partial q} + \frac{\partial W \circ N}{\partial y_i}\frac{\partial y_i}{\partial q}\right] \tag{53}$$

3.3 Computing new coefficients

A first order approximation of the inverse incremental warp is derived [6]:

$$W \circ N \circ M((x,y);\triangle\rho)^{-1} = W \circ N \circ M((x,y);-\triangle\rho) \tag{54}$$

We compute the warp composition and acquire new parameters in a way proposed in [7]. At first, the destination of standard shape $\overline{g}$ under the composition of $W \circ N \circ M((x,y),\rho)$ and $W \circ N \circ M((x,y),\triangle\rho)^{-1}$ is obtained, which is denoted as:

$$\xi = (W \circ N \circ M)((x,y);\rho) \circ (W \circ N \circ M)((x,y);-\triangle\rho) \tag{55}$$

Therefore new parameters ρ_{new} should satisfy the following condition:

$$(W \circ N \circ M)((x,y);\rho_{new}) = \xi \tag{56}$$

Finally the update parameters is computed as follows:

$$q_i = g_i^* \cdot (\xi - \overline{g}) \tag{57}$$

$$a_i = A_i \cdot (N(\xi;q)^{-1} - \overline{g}) \tag{58}$$

However the first order approximation has been proved to be not very accurate in [8], and a novel selective approach is taken instead [9]. For the reason of time limit, we leave it for further study.

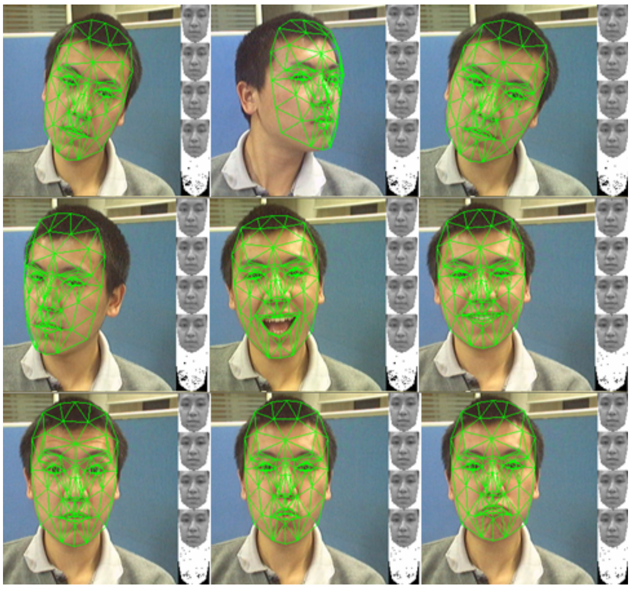

Figure 3: Head pose and facial expression tracking results obtained with a 860-frame-long sequence

4 Experiments

The tracking approach proposed in this paper is tested on a series of videos. The size of video frames is 320×240, and the resolution of observation is set to be 40×46. The platform is a PC with an Intel C2D 3.0GHz CPU. In experiments, fourteen shape modes and six animation modes of the Candide model are used. The six animation units contain: (1) upper lip raiser A_1; (2) jaw drop A_2; (3) lip stretcher A_3; (4) eyebrow lowerer A_4; (5) lip corner depressor A_5; and (6) outer brow raiser A_6. Most common facial expression can be represented by these actions.

Model initialization is needed to detect the face and facial feature points, then determine the shape parameter, and give the pose parameter an initial value. The model initialization is carried out using the first frame of the sequence, under the assumption that the target is facing the camera (out-plane rotations $\alpha = 0$ and $\beta = 0$) and has a neutral expression (all action parameters are 0) at this time. In our work, an Adaboost classifier is used to detect the face, and a weighted AAM algorithm [18] is adopted to detect facial feature points. Both of them are very fast. Robust statistics is used to deal with occlusions.

Figure 3 displays the tracking results associated with several frames of a 860-frame-long test sequence. The right part of each picture shows five observations, which are the means of Component S, Component F and Component W of the mixture model, observation expectation and outlier respectively from top down. Figure 4 shows the six estimated values of pose of the sequence frames. Note that it was eight parameters when tracking. Figure 5 shows the six estimated values of animations of the sequence frames. It can be seen that the sequence contains large head movements and facial actions and the algorithm responds well to them. We find that most of the mixing weights of component S ascend as time goes on in this sequence, which make sense because this component is designed to represent the stable features of all the past observations, and its credibility is promoted by learning if the object keeps moving as the frames going. It can be seen that our algorithm is robust and corresponds well to the actual head and facial actions. Both the mouth and eyebrow act well in response to the actual expression.

Method	Tracking Speed	Fitting Speed
GDM + SOAM	50.7 ms	34.8 ms
GDM + MOAM	64.4 ms	42.9 ms
ICIA + SOAM	40.9 ms	22.1 ms
ICIA + MOAM	51.2 ms	30.5 ms

Table 1: Speed comparison on a Pentium IV 3.0 GHz PC. These results show the average time of tracking and fitting process for each image on a 1388-frame-long test sequence.

For validation purpose, we use the talking face video with ground truth data from the Face and Gesture Recognition Working group for tracking. The video consists of 5000 frames taken from a video of a person engaged in conversation. This corresponds to about 200 seconds of recording. The sequence was taken as part of an experiment designed to model the behavior of the face in natural conversation [19]. Figure 6 displays the tracking results associated with several frames of the test sequence. The right part of each picture shows five observations, which are the means of Component S, Component F and Component W of the mixture model, and observation expectation respectively from top down. It can be seen that our algorithm responds well to them.

The talking face video data set has been tracked with an AAM using a 68 point model. To be precise, we select 49 labeled points that are close to the corresponding Candide model points from 68 annotated points per frame. Figure 7 shows the Candide model with points used for evaluation and the average error over the whole video sequence for each point using a single Gaussian component(SOAM) and a mixture Gaussian model (MOAM). The results shows that MOAM is more robust.

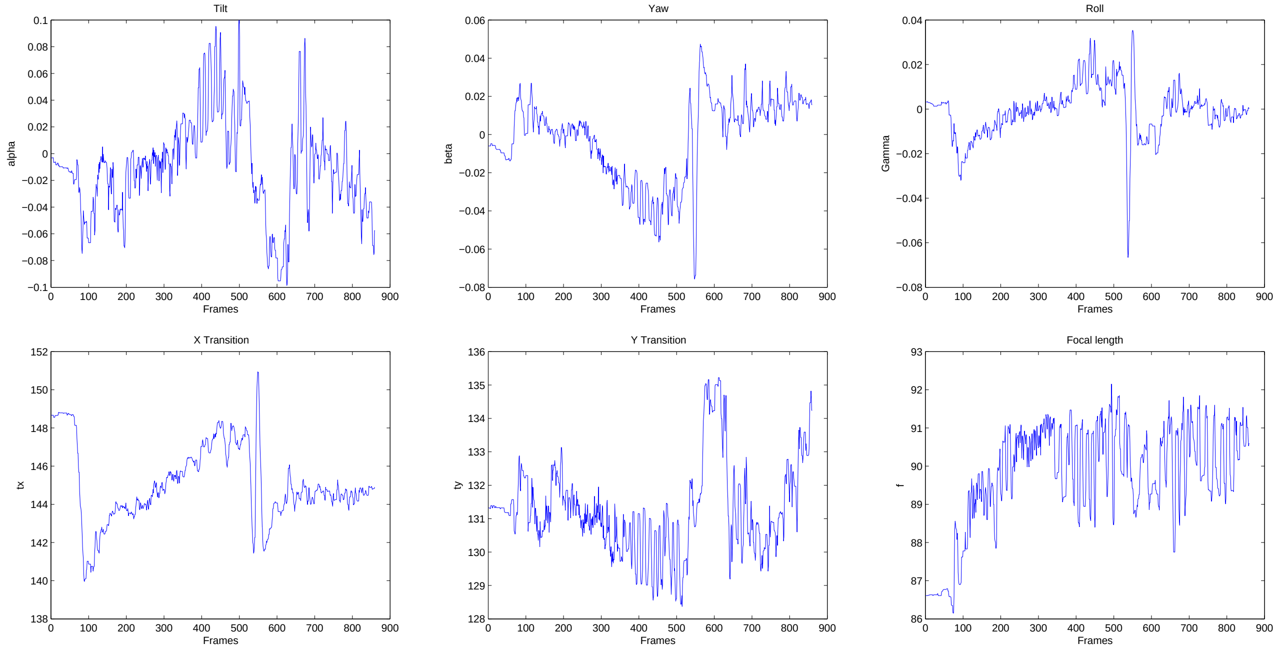

Figure 4: Tracking results of the six pose parameters obtained with a 860-frame-long sequence using EOAM

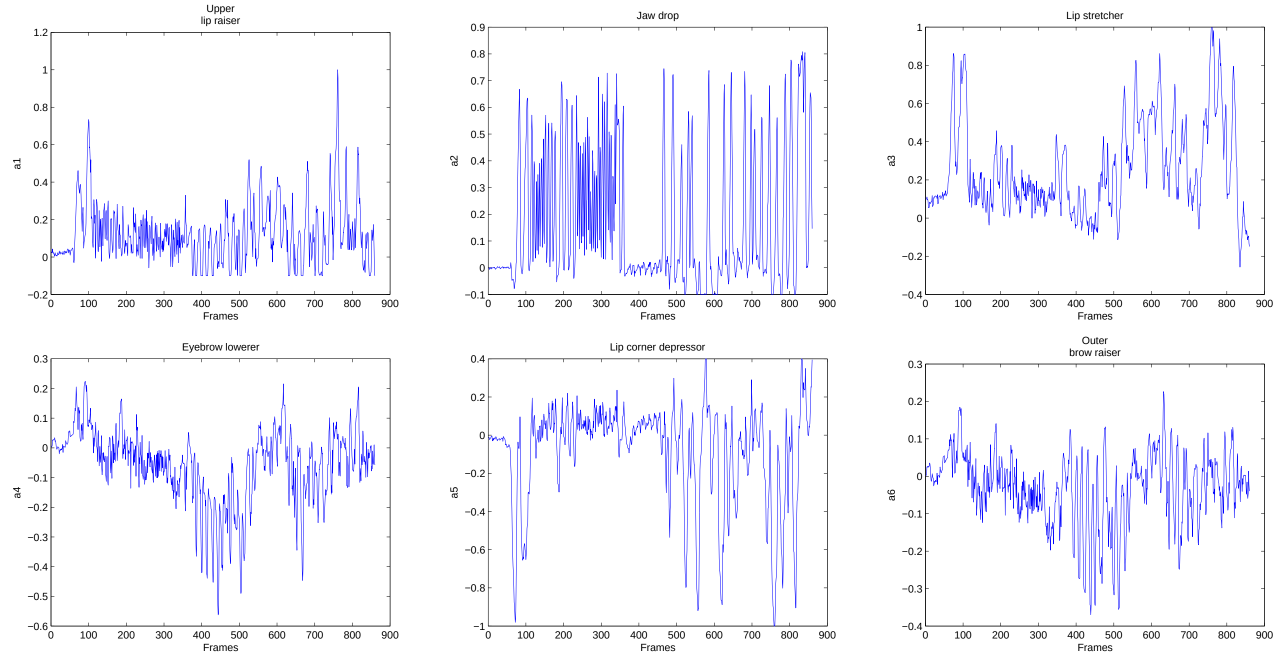

Figure 5: Tracking results of the six facial expression obtained with a 860-frame-long sequence using EOAM

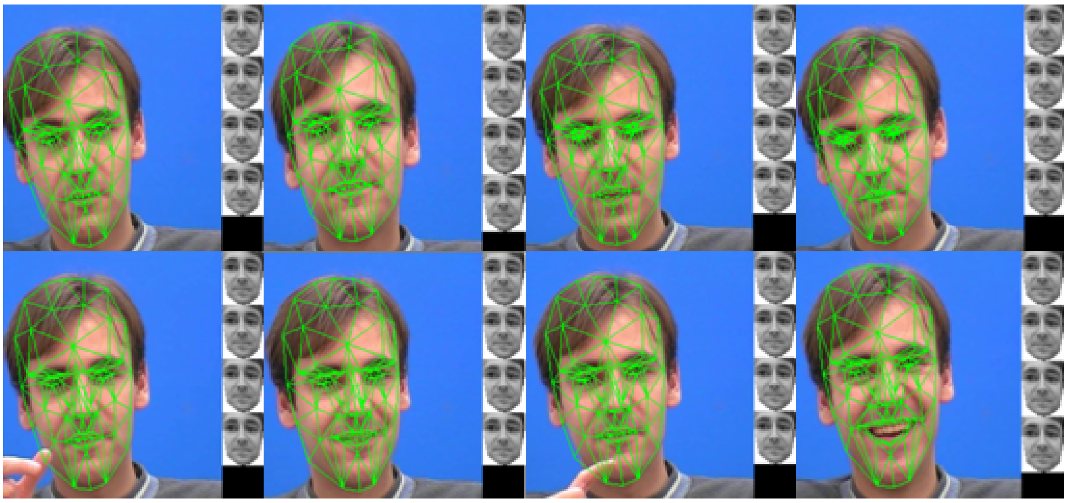

Figure 6: Head pose and facial expression tracking results

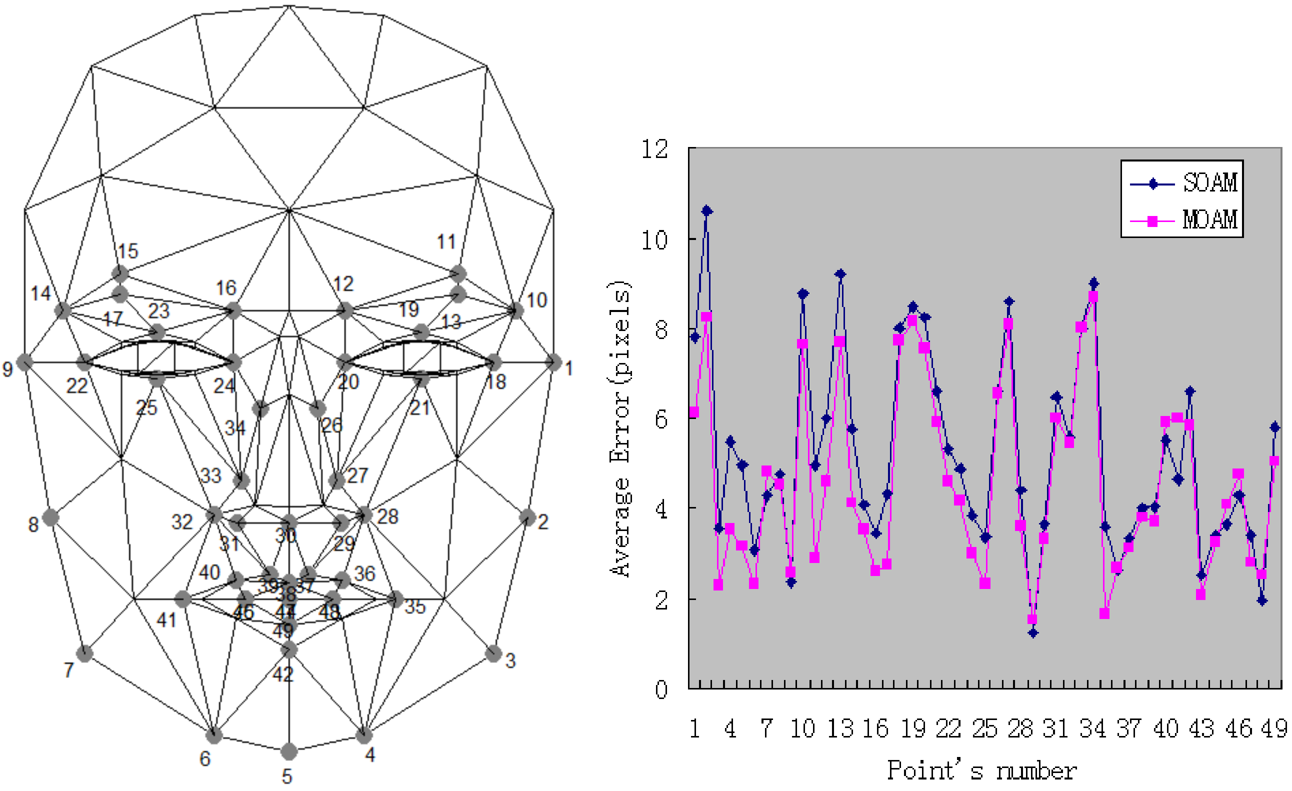

Figure 7: Selected points on model and the average error

Finally the speed of our algorithm is tested and compared with the method using a single Gaussian component with gradient descent method presented in [2][5]. Four combinations of algorithms are compared performing on a 1388-frame-long test sequence. Gradient descent method (GDM) or ICIA is used for fitting, and a single Gaussian component S (SOAM) or the three mixture Gaussian model (MOAM) is taken to model the appearance. The average speed of each method is shown in Table 1. Note that eight pose parameters have to be tracked using ICIA while there are six using GDM. ICIA is still more efficient than GDM, despite more parameters to track. It is slowed down as the template are time-varying and only the Jacobian can be pre-computed before tracking. The learning of mixture Gaussian model is a little time-consuming but it is more robust especially when the head moves quickly or the expression changes fast.

5 Conclusion and future work

A robust and fast tracking algorithm named EOAM is proposed and applied to track head pose and facial actions in this paper. The mixture Gaussian model earns more flexibility than a single one and the ICIA provides an efficient fitting to the model. In our approach, the observation expectation of the mixture Gaussian model is evaluated as the template for ICIA fitting. Facial expression and head pose are modeled, and then a piece-wise affine transform is used to obtain an observation. Experimental results performed on live video sequences demonstrate that our method is robust and efficient.

The actions of mouth and eyebrows can be tracked well with our method, whereas the technique still need to be improved so that more expressions can be expressed and tracked. Initialization is very important in visual tracking, and most algorithms are very sensitive to the veracity of starting position. It is always very difficult to acquire a good initialization automatically. Therefore how to promote the robustness of tracking approaches to the initial position is one of our future work. We don't pay much attention on occlusion in this paper, although robust statistics has been used to deal with it in the experiments. It will be another important direction we are going to study in the future. Future work also contains addressing the robustness of the tracker to important illumination changes and quick movements.

References:

[1] A. D. Jepson, D. J. Fleet, and T. R. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 10, pp. 415–422, 2001.

[2] S. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Process*, vol. 13, no. 11, pp. 1491–1506, 2004.

[3] Dar-Shyang Lee, "Effective gaussian mixture learning for video background subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 827–832, 2005.

[4] C. Grimson and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, no. 6346911, pp. 246–252, 1999.

[5] F. Dornaika and F. Davoine, "On appearance based face and facial action tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 9, pp. 1107–1124, 2006.

[6] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 10, pp. 1090–1097, 2001.

[7] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[8] S. Romdhani and T. Vetter, "Efficient, robust and accurate fitting of a 3d morphable model," *IEEE International Conference on Computer Vision*, vol. 1, no. 8301595, pp. 59–66, 2003.

[9] S. Romdhani and N. Canterakis and T. Vetter, "Selective vs. global recovery of rigid and non-rigid motion," *Tech. Report, University of Basel, Computer Science Dept*, 2003.

[10] J. Ahlberg, "Candide-3 - an updated parameterized face," 2001, Tech. Report No.LiTH-ISY-R-2326. Image Coding Group, Dept.of EE, Linkping University, Sweden.

[11] J. Ahlberg, "Real-Time facial feature tracking using an active model with fast image warping," *International Workshop on Very Low Bitrate Video*, pp. 39–43, 2001.

[12] B. Bascle and A. Blake, "Separability of Pose and Expression in Facial Tracking and Animation," *IEEE International Conference on Computer Vision*, no. 1164, pp. 323–328, 1998.

[13] D. Gavrila, "The visual analysis of human movement: a survey," *Computer Vision and Image Understanding*, vol. 73, no. 1164, pp. 82–98, 1999.

[14] Datong Chen and Jie Yang, "Robust object tracking via online dynamic spatial bias appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2157–2169, 2007.

[15] Qiang Chen, Quan-Sen Sun, Pheng-Ann Heng, and De-Shen Xia, "Robust object tracking via online dynamic spatial bias appearance models," *Pattern Recognition Letters*, vol. 29, no. 2, pp. 126–141, 2008.

[16] S. Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.

[17] Stephen J. McKenna and Yogesh Raja and Shaogang Gong, "Object Tracking using Adaptive Colour Mixture Models," *Image and Vision Computing*, vol. 17, no. 2, pp. 225–231, 1999.

[18] Shuchang Wang and Yangsheng Wang and Xiaolu Chen, "Weighted Active Appearance Models", *International Conference on Intelligence Computing*, pp. 1295–1304, 2007

[19] Face and Gesture Recognition Working group, *http://www-prima.inrialpes.fr/FGnet/html/benchmarks.html*