

# 3D Model Based Expression Tracking in Intrinsic Expression Space

Qiang Wang, Haizhou Ai, Guangyou Xu

Department of Computer Science and Technology, Tsinghua University,  
State Key Laboratory of Intelligent Technology and Systems, Beijing 100084, P.R. China.

## Abstract

*In this paper, a novel method of learning the intrinsic facial expression space for expression tracking is proposed. First, a partial 3D face model is constructed from a trinocular image and the expression space is parameterized using MPEG4 FAP. Then an algorithm of learning the intrinsic expression space from the parameterized FAP space is derived. The resulted intrinsic expression space reduces even to 5 dimensions. We will show that the obtained expression space is superior to the space obtained by PCA. Then the dynamical model is derived and trained on this intrinsic expression space. Finally, the learned tracker is developed in a particle-filter-style tracking framework. Experiments on both synthetic and real videos show that the learned tracker performs stably over a long sequence and the results are encouraging.*

## 1. Introduction

Tracking the face position and the facial expression parameters from video is a challenging problem. There are some existing approaches for tracking 3D facial motion from a sequence of images: The geometric-model-based optical flow approach uses either 3D wireframe/surface model [5][12] or morphable model [2] for facial movement analysis. And optimization techniques are employed to fit the model to the target image using geometric feature correspondence or optical flow constraints. The appearance-model-based image alignment approach uses both shape and texture information for analysis. The appearance model could either be deterministic such as textured deformable 3D face model [11] or statistical such as AAM model [4] and its variants. The analysis-by-synthesis technique is then applied to fit the model to the target image by image warping.

The above expression tracking methods can be generally regarded as extraction of facial motion with different geometric and/or appearance model information. The motion state to be tracked is the variational mode of the model. Since the state space of all facial expression motions is a high dimensional space, different methods are proposed to simplify the state space representation. The exemplar-based approach [1][7][14] assumes that the expression motion state lies in a linear manifold (up to an

error term) and can be linearly reconstructed from a set of key-frame motion templates (exemplars). Similar to the exemplar-based approach, the expression morphable model also assumes the linearity of the expression deformation space but the model could be learned from training data [3].

Although the existing approaches work successfully under certain conditions, there are still some limitations. One limitation is that although linearity is a good assumption for a subspace of the expression space in coarse level, it is not true in real situations. So there is a trade-off between choosing a comprehensive expression space that is not specific enough or a “small” expression space that may ignore many interesting expression modes. Another limitation is the expression dynamical model. Existing methods just assume simple dynamic models that are not suitable for a high dimensional tracking problem since the temporal prediction becomes important.

In this paper, we present a model-based intrinsic expression tracking method. The basic idea is to learn a low-dimensional non-parametric expression state space. A mapping is then established between this intrinsic expression space and a generic high-dimensional parametric expression space so that we can use most of current tracking techniques. A density model of expression spaces is also constructed for probabilistic tracking. Based on the expression density model, a dynamical model of expression is then derived and factorized into a mixture of linear dynamical models. Finally, the learned intrinsic expression model is integrated into a particle filter style framework for tracking.

The intrinsic expression tracking method can be viewed as a data driven tracking approach. Significant improvements are made to establish the continuous intrinsic state space from discrete training data for tracking. The contributions of this paper include: 1) A non-parametric form of mapping between the generic parametric expression space and the learned intrinsic expression space. 2) A factorized intrinsic dynamical model for expression in the form of mixture of linear dynamical models.

The paper is organized as follows: Section 2 briefly describes the 3D face model and expression. Section 3 introduces an algorithm of learning the intrinsic expression mode. Section 4 describes the modeling and learning of the intrinsic dynamical model. Section 5 describes an analysis by synthesis procedure with an

image measurement model and the whole tracking algorithm implemented in a probabilistic tracking framework. Section 6 gives the experimental results. And finally, Section 7 presents the summary and discussions.

## 2. Face model and expression

The 3D face model is constructed by adapting a generic wire-frame model to a calibrated trinocular image. The generic model consists of 352 triangular meshes and covers most of the facial region except hair, eye and ear. We then calculate 3D points from the trinocular image correspondences and use a face adaptation algorithm [15] to get a 3D specific face model. The trinocular image is used because generally three views are needed to accurately determine major feature correspondences in face region. In the experiment, the three views are a frontal view, profile view and an up-front camera view with about 20-degree tilt down angle. The three views are calibrated so that epipolar constraints can be used to guide feature selections. An adaptation result is shown in Figure.1.

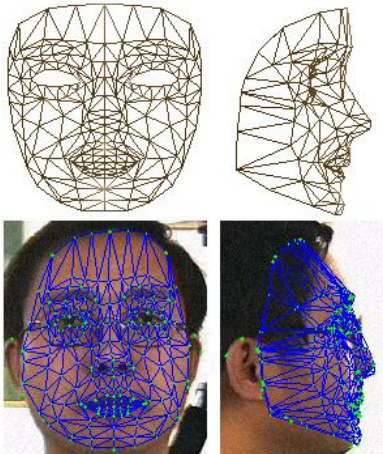


Figure 1. A 3D face model in a frontal view and a side view. The first row is a generic face model and the second row is an adapted face model

The facial expression action is coded using the MPEG4 Face Animation Parameters (FAP) [9]. The FAPs are designed to encode animation of faces reproducing expressions, emotions and speech pronunciation. There are totally 68 FAP parameters. They represent a complete set of basic facial actions and therefore allow representations of most natural facial expressions. The FAP values are defined in the Face Animation Parameter Units (FAPU), which account for face models of different sizes and proportions. So we can get consistent facial animations on different face models with the same FAP set.

## 3. Learning an intrinsic expression model

The FAP representation of facial expression is a generic high-dimensional parameterization. Since different FAPs are independently coded, the FAP state space of expression is not compact and constrained enough as the result it is not appropriate for tracking. In this section, we will describe how to learn a low-dimensional intrinsic expression model. We will start with a non-linear dimensionality reduction algorithm on discrete training data, based on which we establish a continuous mapping between the generic FAP state space and the low dimensional intrinsic state space. Then a density model capturing the global structure of the expression state is constructed and the model parameters are estimated using maximum likelihood algorithm.

### 3.1. Non-linear dimensionality reduction

We consider a high dimensional expression data set  $\mathbf{X}=\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  with  $\mathbf{x}_i \in \mathbb{R}^D$ , each data item corresponds to a FAP frame that represents one-frame facial expression motion. Since most facial motions are strongly coupled in lower face, we could assume that  $\mathbf{X}$  lies on a  $d < D$  dimensional, possibly a nonlinear manifold. The goal of dimensionality reduction is to express  $\mathbf{X}$  in the intrinsic coordinates of a manifold:  $\mathbf{Y}=\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$  with  $\mathbf{y}_i \in \mathbb{R}^d$ . Thus the method aims at obtaining a compact expression representation from examples, that is, to discover a few degrees of freedom that underlie observed modes of continuous variability without use of a priori knowledge.

We use the ISOMAP algorithm [13] for dimensionality reduction. The aim of ISOMAP is to preserve the topological structure of the original data, i.e. the Euclidean distances in  $\mathbb{R}^d$  should correspond to the geodesic distances (distances on the manifold) in  $\mathbb{R}^D$ . The algorithm makes use of a neighborhood graph to find the topological structure of the data and if we set the neighborhood of each data point as the whole data set, the ISOMAP algorithm degenerates to the PCA algorithm.

The size and quality of the training data is important for learning the intrinsic expression model. There are totally 4 training FAP sequences, including Chinese speech, English speech with expression, German speech and basic expression sequences. Of the total 10000 frames, about 2500 key frames are automatically selected as the training data and they have been verified by our face animation system. Each FAP frame contains 16 FAP values, which mainly account for the lower facial motion and the symmetric redundancy is eliminated. By applying the dimensionality reduction algorithm, we get the intrinsic coordinates of the FAP data manifold with dimensionality  $d=5$ , accounting for 95% of the total variance.

### 3.2. Mapping between state spaces

Given the discrete high dimensional FAP data set  $\mathbf{X}$  and the corresponding learned intrinsic coordinate data set  $\mathbf{Y}$ , we now need to establish a continuous mapping between the FAP state space and the intrinsic state space. We propose to use a non-parametric form of mapping based on local linear projection (LLP). The idea basically has two assumptions: 1) each data point and its neighbors are lying on a locally linear patch of the manifold; 2) the mapping from a high-dimensional patch to a low-dimensional patch is a linear projection.

Based on these assumptions, the process of computing the output FAP state  $\mathbf{x}$  from the input intrinsic state  $\mathbf{y}$  has the following steps:

- 1) Identify the neighbors of  $\mathbf{y}$  in the training data set  $\mathbf{Y}$  and denote it as  $\mathbf{P}_y$ . The high dimensional corresponding neighborhood set is denoted as  $\mathbf{R}_y$ .
- 2) Calculate the linear projection matrix  $A$  according to  $\mathbf{P}_y$  and  $\mathbf{R}_y$ .
- 3) Calculate  $\mathbf{y}$  as:  $\mathbf{x} = A\mathbf{y}$ .

The calculation of the matrix  $A$  can be described as follows. Given  $\mathbf{R}_y = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ ,  $\mathbf{x}_i \in \mathbb{R}^D$ ,  $\mathbf{P}_y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$ ,  $\mathbf{y}_i \in \mathbb{R}^d$ , the linear projection assumption satisfies:

$$\mathbf{x}_i = A\mathbf{y}_i \quad i=1, \dots, k \quad (1)$$

where  $A$  is a  $D \times d$  matrix and thus we have:

$$A = XY^T(YY^T)^{-1} \quad (2)$$

$$X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k)^T, Y = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)^T.$$

A comparison of the mapping algorithm by LLP and PCA is shown in Figure 2. Each algorithm preserves 5 dimensions in low dimensional space and the errors are measured in high dimensional space after a round mapping.

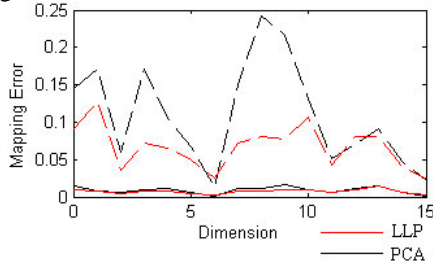


Figure 2. Comparison of mapping algorithms performed on the whole data set (10000 frames). The solid lines represent the mean error while the dashed lines represent the maximum error.

### 3.3 Expression density model

The density of expression state is modeled by mixture of factor analyzers (MFA) [6]. The factor analysis is a statistical method for modeling the covariance structure of high dimensional data using a small number of latent variables. Different from PCA, the FA model defines a

proper density model for the data and independent noise. The MFA model is its extension by allowing different local factor models in different regions of the input space, and thus concurrently performs clustering and dimensionality reduction. The resulted expression density model can be used in the subsequent probabilistic prediction and inference for tracking.

For the completeness of the paper, we will briefly describe the MFA model and its model parameter estimation. Interested reader can refer to [16] for more detail.

In FA model, the generative model is given by:

$$\mathbf{x} = \Lambda\mathbf{y} + \mathbf{u} \quad (3)$$

Where  $\mathbf{x}$  is a  $D$ -dimensional real-valued data vector,  $\mathbf{y}$  is a  $d$ -dimensional vector of the real-valued factor and  $d$  is generally much smaller than  $D$ .  $\Lambda$  is known as the factor loading matrix. The factor  $\mathbf{y}$  is assumed to be  $N(\mathbf{v}, \Sigma)$  distributed (a Gaussian distribution with mean  $\mathbf{v}$  and covariance  $\Sigma$ ). The  $D$ -dimensional random variable  $\mathbf{u}$  is  $N(\boldsymbol{\mu}, \Psi)$  distributed, where  $\Psi$  is a diagonal matrix,  $\mathbf{y}$  and  $\mathbf{u}$  are assumed to be independent. For a mixture of  $M$  factor analyzers indexed by  $w$ ,  $w=1, \dots, M$ , the generative model is:

$$\mathbf{x} | w = \Lambda_w \mathbf{y} | w + \mathbf{u}_w \quad (4)$$

From (4), the overall distribution of the MFA model is given by:

$$P(\mathbf{x}, \mathbf{y}, w) = P(\mathbf{x} | \mathbf{y}, w)P(\mathbf{y} | w)P(w) \quad (5)$$

$$P(\mathbf{x} | \mathbf{y}, w) = N(\Lambda_w \mathbf{y} + \boldsymbol{\mu}_w, \Psi_w) \quad (6)$$

$$P(\mathbf{y} | w) = N(\mathbf{v}_w, \Sigma_w) \quad (7)$$

In the MFA model,  $\mathbf{y}$ ,  $w$  are the latent variables and the parameters of the MFA model are  $\{\Lambda_w, \boldsymbol{\mu}_w, P(w), \Psi_w\}$ . The algorithm to estimate the model parameters can be found in [16] and the resulted MFA model of expression is shown in Figure 3.

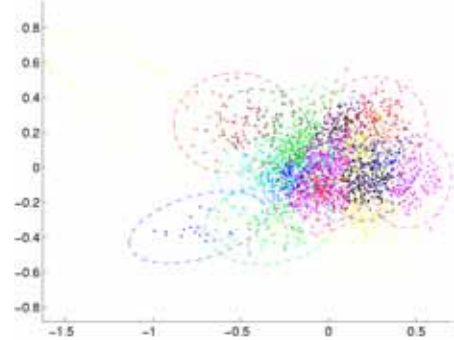


Figure 3. The resulted MFA model for the first 2 dimensions. There are 22 mixture components, where '+' represents mixture centers and ellipses represent equal probability curves of 3 s.t.d.

## 4. Intrinsic dynamical model

Based on the density model of the intrinsic expression state, we can derive its dynamical model. Assume a second order Markov property of the dynamical process, the dynamical model can be represented by  $P(\mathbf{y}_t | \mathbf{y}_{t-1}, \mathbf{y}_{t-2})$  and it can be factorized as follows:

$$P(\mathbf{y}_t | \mathbf{y}_{t-1}, \mathbf{y}_{t-2}) = \sum_{w_t} P(\mathbf{y}_t | \mathbf{y}_{t-1}, \mathbf{y}_{t-2}, w_t) P(w_t | \mathbf{y}_{t-1}, \mathbf{y}_{t-2}) \quad (8)$$

Where  $P(\mathbf{y}_t | \mathbf{y}_{t-1}, \mathbf{y}_{t-2}, w_t)$  is the dynamical model under fixed mixture label and can be modeled as a second order Gaussian Auto-Regressive Process (ARP):

$$P(\mathbf{y}_t | \mathbf{y}_{t-1}, \mathbf{y}_{t-2}, w_t) = N(A_1^{w_t} \mathbf{y}_{t-1} + A_2^{w_t} \mathbf{y}_{t-2} + \mathbf{d}^{w_t}, C^{w_t}) \quad (9)$$

$P(w_t | \mathbf{y}_{t-1}, \mathbf{y}_{t-2})$  is the dynamical model of mixture label. We assume that it is first order and can be modeled as:

$$P(w_t | \mathbf{y}_{t-1}, \mathbf{y}_{t-2}) = \frac{P(\mathbf{y}_{t-1} | w_t) P(w_t)}{\sum_{w_{t-1}} P(\mathbf{y}_{t-1} | w_{t-1}) P(w_{t-1})} \quad (10)$$

where  $P(\mathbf{y}_{t-1} | w_t)$  is assumed to be  $N(\mathbf{e}^{w_t}, \Omega^{w_t})$  distributed.

The parameters in the whole dynamical model are  $\Xi = \{A_1^l, A_2^l, \mathbf{d}^l, C^l, \mathbf{e}^l, \Omega^l\}$ ,  $l=1, \dots, M$ . The parameters can be learned from temporal training sequence using the MLE algorithm as described in [10].

## 5. Analysis-by-Synthesis Tracking

### 5.1 Image synthesis and measurement

Given the textured 3D face model and the estimated FAP parameters, we can get the synthesized expression image. The synthesized image is then compared to the target image to get an observation probability which can be used in probabilistic tracking.

For the synthesis, the 3D wireframe model is first deformed according to the FAP parameters. Then the 3D model's vertices are projected to the image plane to get a target 2D mesh model. Denote the reference texture image as  $I_0$ , the current target image as  $I_t$ , the reference and target mesh set is  $\{s_1^0, \dots, s_m^0\}$  and  $\{s_1^t, \dots, s_m^t\}$  respectively, then the warping process from the target image to the reference image is denoted as:

$$I_t^0(s_i^0) = T_{\alpha_i}(I_t(s_i^t)) \quad (11)$$

Where  $I_t(s_i^t)$  is the image defined in triangle  $s_i^t$ .  $T_{\alpha_i}$  is the corresponding triangular warping function with

parameter  $\alpha_i$ .

The image distance is defined based on the piece-wise image SSD:

$$D = \sum_{i=1}^m W_i \sum_{s_i^0} (I_t^0(s_i^0) - I_0(s_i^0))^2 \quad (12)$$

Where  $W_i$  is the weight of mesh  $s_i^0$ . Then we model the image measurement using truncated Laplacian as follows:

$$P(I_t | FAP_t) = \begin{cases} e^{-\lambda * D / D_0} & D \leq D_0 \\ e^{-\lambda} & D > D_0 \end{cases} \quad (13)$$

where  $\lambda$  and  $D_0$  are manually selected constants.

In order to make the image measurement model more specific, we divide the whole face into different regions as shown in Figure 4. There are 3 kinds of regions: rigid regions such as forehead; non-rigid regions such as mouth, cheek and outlier regions such as the nose hole. So we can determine the mesh weight according to the property of its region and the motion to be measured. We also developed an algorithm for detecting mesh occlusions. The occluded meshes do not contribute to the image measurement model.



Figure 4. Face region division. Different colors represent different face regions.

### 5.2 The probabilistic tracking framework

The tracking algorithm adopts a framework of the CONDENSATION [8]. One step of the tracking algorithm is to generate a weighted sample set  $S_t = \{(s_t^{(n)}, \pi_t^{(n)}), n = 1, \dots, N\}$  at time  $t$  from the set  $S_{t-1} = \{(s_{t-1}^{(n)}, \pi_{t-1}^{(n)}), n = 1, \dots, N\}$  at time  $t-1$ . The weighted sample set approximates the conditional object state density where  $s_t^{(n)}$  is a discrete random sample and has a probability proportional to its weight  $\pi_t^{(n)}$ . The one-step tracking algorithm has the following 3 steps:

- 1) Estimate rigid motion parameters by generating the sample set  $\hat{S}_t = \{(\hat{s}_t^{(n)}, \hat{\pi}_t^{(n)}), n = 1, \dots, N\}$  at time  $t$  from set  $S_{t-1} = \{(s_{t-1}^{(n)}, \pi_{t-1}^{(n)}), n = 1, \dots, N\}$  at time  $t-1$ . For each sample there are three steps, re-sampling, prediction and measurement: a)  $s_{t,0}^{(i)} = \text{Importance\_resampling}(S_{t-1})$ ; b)

$\hat{s}_t^{(i)} = \text{Prediction}(s_{t,0}^{(i)})$ : denote  $s_{t,0}^{(i)} = s_{t,0,a}^{(i)} \oplus s_{t,0,b}^{(i)}$ , where  $\oplus$  is the vector concatenation operator,  $s_{t,0,a}^{(i)}$  is the rigid motion parameter and  $s_{t,0,b}^{(i)}$  is the non-rigid intrinsic expression state. The rigid part of  $\hat{s}_t^{(i)}$  is predicted by a second order linear dynamical model from  $s_{t,0,a}^{(i)}$  and the non-rigid part of  $\hat{s}_t^{(i)}$  equals  $s_{t,0,b}^{(i)}$ ; c)  $\hat{\pi}_t^{(i)} = \text{Measurement}(\eta(\hat{s}_t^{(i)}))$ , where  $\eta$  denote the mapping from intrinsic expression state to FAP state and  $\text{Measurement}()$  represents the measurement model described in Section 5.1.

- 2) Calculate the rigid motion parameters' posterior mean  $s_{t,a}$ .
- 3) Estimate non-rigid motion parameters. For each sample, there are still three steps: a)  $s_{t,1}^{(i)} = \text{Importance\_resampling}(\hat{S}_t^{(i)})$ ; b)  $s_{t,1}^{(i)} = \text{Prediction}(s_{t,1}^{(i)})$ : denote  $s_{t,1}^{(i)} = s_{t,1,a}^{(i)} \oplus s_{t,1,b}^{(i)}$ ,  $s_{t,1,a}^{(i)}$  is the rigid motion parameter and  $s_{t,1,b}^{(i)}$  is the non-rigid intrinsic expression state. The rigid part of  $s_{t,1}^{(i)}$  is obtained by adding a small Gaussian noise to  $s_{t,a}$  and the non-rigid part of  $s_{t,1}^{(i)}$  is predicted from  $s_{t,1,b}^{(i)}$  using the intrinsic dynamical model described in Section 4; c)  $\pi_t^{(i)} = \text{Measurement}(\eta(s_{t,1}^{(i)}))$

## 6. Experiments

### 6.1 Synthetic video

In this experiment, we use the synthesized video for tracking. The synthesized video is obtained by animating a textured 3D face model. The synthesized video is assumed as been captured by a real camera during tracking and it is used for 2 reasons: 1) the ground-truth data of facial expression motion is easy to get and so facilitate the evaluation of the tracking result. Since current expression tracking algorithms have different constraints and output forms, comparison with the ground truth is a good choice. 2) We could control the accuracy of image measurement model by adding some noise to the synthesized video. In the following experiment, we add some Gaussian noise with intensity variance 10.

There are 16 non-rigid FAPs and 3 global rotational FAPs for tracking. There are totally 3 training data sets: the training data for learning intrinsic expression model which has about 2500 FAP frames; the training data for learning the intrinsic dynamical model which are two continuous FAP sequence, totally 5000 frames; the training data for learning the rigid motion's dynamical model which is a 1000-frames continuous FAP sequence.

The result of expression tracking on synthetic video is shown in Figure 5 and the corresponding quantified comparison of the tracking results with the ground truth

data is shown in Figure 6. The image size is 136\*172 and the face speaks, smiles with significant head rotation. 1000 samples are used for the tracker to achieve stable result. The tracking speed is about 7s/frame on a Pentium IV 1.4G machine. The mesh occlusions detection module and the image warping module described in section 5.1 take over 90% percent of the whole processing time. The tracking results are encouraging since the tracker tracks stably over a long sequence (about 900 frames) where significant head rotation and expression occur simultaneously. This also implies that during tracking, the learned intrinsic expression state and the rigid global rotational state can be separated naturally.



Figure 5. Synthetic video tracking results. The first row is the original synthetic images corresponding to frame 48, 119, 140, 295, 580, 796, 863. The second row is corresponding synthesized images by applying the tracking result on the same face model. The third row is the synthesized tracking result on another face model.

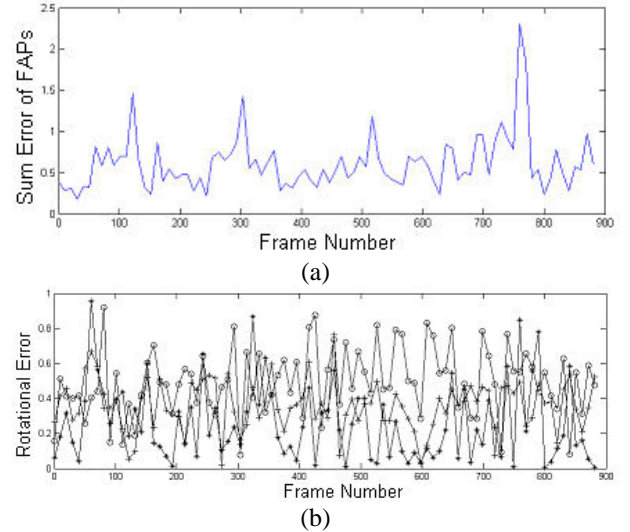


Figure 6. Quantified comparison results. (a) Sum of error of 16 non-rigid FAPs, each FAP's error is normalized by the FAP quantization step. (b) Rotational error. The piecewise lines with 'o', '+', '\*' correspond to tilt, pan and roll respectively and the error is in degree.

### 6.2 Real video

In this experiment, we track video captured by a real camera. The training process is the same as that described in Section 6.1. The image size is 768\*576, 1000 samples are used for the tracker to achieve stable result. The tracking speed is about 15s/frame on a Pentium IV 1.4G machine. Due to space limit, part of the tracking result is shown in Figure 7.

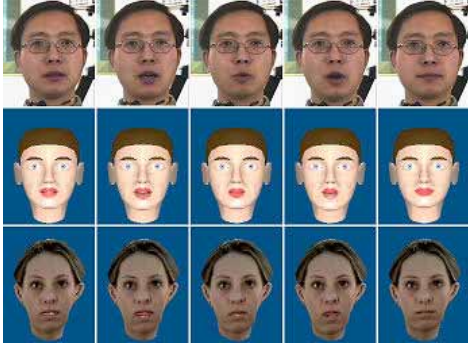


Figure 7. Real video tracking results. The first row is the images corresponding to frame 35, 41, 132, 212, 348. The second and third row are the corresponding synthesized images by applying the tracking results on different face models.

## 7. Conclusions

A method of 3D model based intrinsic expression tracking is presented. The method builds an intrinsic expression state space with density models from a large collection of expression examples for tracking. The obtained intrinsic expression state is compact, specific. With a 3D face model and a region-based image measurement, the tracker can deal with rigid head motion and non-rigid intrinsic facial expression separately. Experiments are done on both synthetic and real videos and the tracker tracks stably over a long sequence where significant head rotation and expression occur simultaneously. The tracked expression parameters can be used for face animation.

The power of the intrinsic expression tracker comes from three points: 1) the compactness and specificity of the intrinsic expression representation; 2) the flexibility of the intrinsic dynamical model for tracking the autonomous expression motion; 3) all the learned models for the tracker are efficiently estimated from unlabeled data.

We intend to explore several avenues in future work: 1) optimize the tracker towards real time. We are planning to reduce training samples following the importance sampling technique; 2) modeling the variations of illumination, facial wrinkle and oral cavity in tracking.

## 8. References

- [1] B. Bascle and A. Blake. Separability of Pose and Expression in Facial Tracing and Animation. *Proc. IEEE ICCV*, pages 323-328, 1998.
- [2] M. Brand and R. Bhotika. Flexible flow for 3D nonrigid tracking and shape recovery. *Proc. IEEE CVPR*, vol. 1, pages 315-322, 2001.
- [3] M. Brand. Morphable 3D models from video. *Proc. IEEE CVPR*, vol. 2, pages 456-463, 2001.
- [4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Proc. ECCV*, pages 484-498, 1998.
- [5] D. DeCarlo and D. Matas. Optical flow constraints on deformable models with applications to face tracking. *IJCV*, 38(2):99-127, 2000.
- [6] Z. Ghahramani and G. Hinton. The EM algorithm for mixtures of factor analyzers. University of Toronto Technical Report, CRG-TR-96-1, 1996.
- [7] T. S. Huang and P. Hong. Exemplar-based face and facial motion tracking. *Proc. IEEE ICASSP*, vol. 4, pages 3600-3603, 2002.
- [8] M. Isard and A. Blake. CONDENSATION – conditional density propagation for visual tracking. *Int. J. Computer Vision*. 29(1): 5-28, 1998.
- [9] MPEG Video. Information technology - Coding of audio-visual objects - Part 2: Visual, Amendment 1: Visual extensions. *ISO/IEC JTC1/SC29/WG11/N3056*, Jan. 2000
- [10] B. North, A. Blake, M. Isard and J. Rittscher. Learning and Classification of Complex Dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1016-1034, 2000.
- [11] F. Pighin, R. Szeliski and D.H. Salesin. Resynthesizing facial animation through 3D model-based tracking. *Proc. IEEE ICCV*, Vol. I, pages 143-150, 1999.
- [12] H. Tao and T. S. Huang. Explanation-based Facial Motion Tracking Using a Piecewise Bezier Volume Deformation Model. *Proc. IEEE CVPR*, pages 611-617, 1999.
- [13] J. B. Tenenbaum, V. deSilva and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319-2323, 2000.
- [14] K. Toyama and A. Blake. Probabilistic tracking with exemplar in a metric space. *IJCV*, 48(1):9-19, 2002.
- [15] Q. Wang, H. Zhang, T. Riegel, E. Hundt and G. Xu. Creating Animatable MPEG4 Face. *Proc. International conference on Augmented, Virtual Environments and three-dimensional imaging*, pages 228-231, 2001.
- [16] Q. Wang, G. Xu and H. Ai. Learning Object Intrinsic Structure for Robust Visual Tracking. *Proc. IEEE CVPR*, vol. 2, pages 227-233, 2003.