Nonlinear Shape and Appearance Models for Facial Expression Analysis and Synthesis

Chan-Su Lee and Ahmed Elgammal
Rutgers University
{chansu,elgammal}@cs.rutgers.edu

Abstract

Facial expression passes through nonlinear shape and appearance deformations with variations in different people and expressions. We present nonlinear shape and appearance models for facial expression analysis and synthesis using nonlinear generative models for different facial expressions in different people. To achieve accurate shape normalized appearance models, we utilize nonlinear warping using thin plate spline (TPS). A novel nonlinear generative model using conceptual manifold embedding and empirical kernel maps for facial expressions provides facial shape and appearance samples according to the configuration, personal style, and expression parameters. We can recognize facial expressions based on estimated facial expression parameters after iterative estimations of facial expression and style. In addition, the model provides accurate synthesis of facial expression sequences even with high nonlinear deformations of shape and appearance during facial expressions.

1. Introduction

Recently, demands for accurate modeling and analysis of facial expressions are growing with new applications of facial motion analysis like deception detection and affective computing. Recognition of emotional states and synthesis of facial expressions are one of the key components for intelligent affective human computer interactions [15]. However, it is difficult to model subtle facial motions with current linear subspace-based models as facial motions pass through nonlinear shape and appearance deformations with variations in different people and expressions.

Most of current systems for analysis and tracking of facial motions from shape and appearances are based on linear models. Active shape models (ASM) [8] are well known linear models for facial motion analysis and tracking using point distribution models in linear subspace. By constraining deformation of point distributions into a linear subspace of the training shapes during local search, it achieves robust fitting of face models [8]. Active appearance models (AAM) combine the linear shape model of points distributions in ASM and the linear texture appearance model by aligning appearance models into a normalized shape using piecewise affine warping [7]. Iterative model refinement algorithms are proposed based on a prediction model, which is learned as a regression model using observations of linear shape and appearance model parameter variations after perturbations.

Bilinear and multilinear models are applied to improve accuracy in modeling facial expression recognitions [1, 20] and facial expression synthesis [5]. All these models are based on linear shape and appearance models with extensions for multiple factors. In addition, any of these models do not count dynamics in facial expressions except [12]. Feature-based facial expression recognition system may overcome the limitation of linear subspace of templates by directly tracking features like facial action units [13]. Feature based approaches, however, are hard to model appearance variations in facial expression and may not be able to synthesize appearance of new facial expressions.

We propose nonlinear shape and appearance models for facial expression analysis and synthesis. When dealing with dynamic facial expressions, image sequences lie on low dimensional nonlinear manifolds embedded in a high dimensional input space.

We model facial expressions by explicit modeling of configuration manifolds and decomposing variability due to different people and expression types. Nonlinear generative models using low-dimensional conceptual manifold embedding and empirical kernel mapping are developed to learn nonlinear shape and appearance model in low dimensional spaces as in Sec. 3. This generative model provides a global shape and appearance deformation model during facial expressions. Iterative estimation of the model parameters allows recognition of facial expression for a given image as in Sec. 3.1. To achieve accurate shape-normalized appearance images for learning our models, we employed thin-plate spline (TPS) warping as described in Sec. 2. Ex-

perimental results in Sec. 4 using Cohn-Kanade AU coded facial expression database [10] show accurate synthesis of facial expressions and better recognition results of facial expressions than multilinear models based on linear subspace extensions [20].

2 Shape-normalized appearance models using thin-plate spline warping

To develop nonlinear generative shape and appearance models, we first represent shapes by distributions of landmark points and compute a mean shape that is used for appearance normalization. We describe the ith shape by n landmark points as a vector $\boldsymbol{p}_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2}, \cdots, x_{in}, y_{in})$. We compute the mean shape after shape alignments by weighted similarity transformation [8] from collected landmark points from different people with different expressions. We weighted each landmark based on the reliability of the face components as some components, like a nose, are more reliable than other face components, like a mouse contour, in facial expressions. By shape normalization, appearance vectors will establish good correspondences between each element of the shape-normalized appearance vectors and, therefore, meaningful algebraic operations between appearance vectors can be achieved.

To achieve shape normalization, piecewise-affine warping is frequently used in linear appearance models [14]. However, piecewise-affine warping can cause artifacts around boundaries for non-rigid shape deformation due to facial motions [6]. We use thin plate spline (TPS) warping for the non-rigid registration of appearance images to the mean shape. TPS warping have been widely used in medical image alignments and non-rigid deformations [16] after popularization by Bookstein [3]. Given an image sequence $\boldsymbol{I}_1, \boldsymbol{I}_2, \cdots, \boldsymbol{I}_{N_K}$, where N_K is the number of image frames, with corresponding shape vectors $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_{N_K}$, we can obtain the mean shape $\boldsymbol{p}_0$. We need to warp every image $\boldsymbol{I}_j$ with shape vector $\boldsymbol{p}_j$ into new image $\boldsymbol{I}_j^{tps} = \boldsymbol{I}(W_{\boldsymbol{p}_0}(\boldsymbol{p}_j))$. $\boldsymbol{I}(W_{\boldsymbol{p}_0}(\boldsymbol{p}_j))$ denotes the warped image of $\boldsymbol{I}_j$ by TPS warping with shape landmark points $\boldsymbol{p}_j$ into $\boldsymbol{p}_0$. In actual computation, we perform backward warping because of the discrete nature of raster images and computational efficiency as explained in the following.

TPS warping leads to smooth deformations and accurate normalization of non-rigid deformations of appearances in facial expressions. In case of backward warping we need to warp output image coordinates into input image coordinates by TPS warping from $\boldsymbol{p}_0$ to $\boldsymbol{p}_j$ and interpolate intensity values based on the warped coordinate. TPS warping specifies

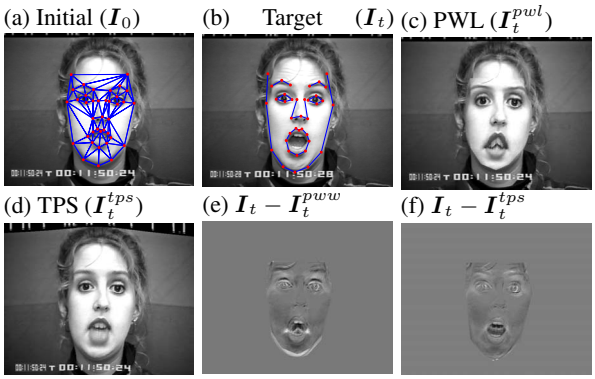


Figure 1. Image warping:(a) The original image with shape landmarks. It also shows delaunay triangulation [2] result. (b) The target image with its shape landmarks. (c) Piecewise linear affine transformation based on shape triangulation for shape landmark points. (d) TPS warping based on shape landmark correspondences. (e) (f) Image difference between target image and warped image by piecewise warping (PWL) and by TPS warping (TPS)

a mapping in the form

$$f(x,y) = a_1 + a_x x + a_y y + \sum_{i=1}^{2n} w_i U(|\boldsymbol{p}_0 - (x,y)|) \quad (1)$$

which minimizes bending energy

$$E_f = \iint_{\mathbb{R}^2} \left(\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right) dxdy$$

where $U(x,y) = r^2 log r^2, r = \sqrt{x^2+y^2}$. When we perform backward warping from the shape $\boldsymbol{p}_0$ to a shape $\boldsymbol{p}_j$, the coefficients in the mapping function Eq. 1 can be computed as follows:

$$(w_1 \cdots w_{2n} | a_1 a_x a_y)^\mathsf{T} = \boldsymbol{L}^{-1} \boldsymbol{Y}, \quad (2)$$

where $\boldsymbol{Y} = \begin{bmatrix} x_{j1} & x_{j2} & \cdots & x_{jn} \\ y_{j1} & y_{j2} & \cdots & y_{jn} \end{bmatrix}$, $\boldsymbol{L} = \left[\begin{array}{c|c} \boldsymbol{K} & \boldsymbol{P} \\ \hline \boldsymbol{P}^\mathsf{T} & \boldsymbol{O} \end{array}\right]$, $\boldsymbol{K}_{ij} = U(||(x_{0i}, y_{0i}) - (x_{0j}, y_{0j})||)$ and the i-th row of $\boldsymbol{P}$ is $(1, x_{0i}, y_{0i})$. Computing the coefficients, we need to compute the inverse of matrix $\boldsymbol{L}$ only once as it only depends on $\boldsymbol{p}_0$. Fig. 1 (e) and (f) shows intensity differences between target image (b) and warped images (c) and (d) within target shape contour. When we measure errors between true target image appearance and warped ones by pixel intensity differences, the TPS warped appearance shows 20% less errors than the piecewise linear one as its warping based on corresponding landmark points describe more accurately nonlinear deformation of appearance in large shape deformation during facial expressions. The piecewise linear warping result may be improved using more dense landmark points and elaborate triangulation.

Given an image sequence, the kth image $\boldsymbol{I}_k$ is represented by the aligned shape $\boldsymbol{p}_k$ and shape-normalized appearance $\boldsymbol{a}_k$, where the appearance vector, $\boldsymbol{a}_k$, is the vector representation of pixels which are inside the mean shape contour after warping the image to the mean shape by TPS, i.e.,

$$\boldsymbol{a}_k = \boldsymbol{I}_{\boldsymbol{\xi} \in \boldsymbol{p}_0}^{k} (W_{\boldsymbol{p}_0}(\boldsymbol{\xi}, \boldsymbol{p}_k)). \quad (3)$$

We combine the shape vector and the appearance vector as a new vector $\boldsymbol{y}_k = [\boldsymbol{p}_k^\mathsf{T} \boldsymbol{a}_k^\mathsf{T}]^\mathsf{T}$ for the facial expression analysis. In this case, the dimension of $\boldsymbol{y}$ is $N_{as} = 2n + N_a$ where the number of pixels within the mean shape is N_a. The combined shape and appearance vectors are used in modeling facial motions using the nonlinear generative model as will be described in the next section.

3 Nonlinear generative models with manifold embedding

Nonlinear dimensionality reduction has been recently exploited to model manifold structures in face recognition [18] and facial expression analysis [4]. When unsupervised data-driven manifold embedding techniques are used, resulting embedded manifolds of the same type of facial expressions performed by different people will be quite different and it is hard to find a unified representation of the manifolds. But, conceptually all these manifolds are the same 1-dimensional circular curves for expressions, which move from neutral expressions to target expressions and return back to neutral expressions. Using this conceptual manifold embedding and nonlinear mapping, we can model dynamics of facial expressions in a low dimensional space similar to [9, 12].

A set of image sequences which represent full cycles of the facial expressions are used for conceptual embedding of facial expressions using a unit circle. The image sequences are not necessarily to be of the same length. We denote each sequence by $Y^{se} = \{y_1^{se} \cdots y_{N_{se}}^{se}\}$ where e denotes the expressions and s is person's identity. Let N_e and N_s denote the number of expressions and number of people in the training data respectively, i.e., there are $N_s \times N_e$ sequences. Each sequence is temporally embedded at equidistances on a unit circle such that $x_i^{se} = [cos(2\pi i/N_{se}) \ \ sin(2\pi i/N_{se})], i = 1 \cdots N_{se}$.

Given a set of distinctive representative and arbitrary points on the unit circle $\{\boldsymbol{z}_i \in \mathbb{R}^2, i = 1 \cdots N\}$, we can define an empirical kernel map[17] as $\psi_N(\boldsymbol{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}^N$ where

$$\psi_N(\boldsymbol{x}) = [\phi(\boldsymbol{x}, \boldsymbol{z}_1), \cdots, \phi(\boldsymbol{x}, \boldsymbol{z}_N)]^\mathsf{T}, \quad (4)$$

given a kernel function $\phi(\cdot)$. For each input sequence $\boldsymbol{Y}^{se}$ and its embedding $\boldsymbol{X}^{se}$ we can learn a nonlinear mapping function $f^{se}(\boldsymbol{x})$ that satisfies $f^{se}(\boldsymbol{x}_i) = \boldsymbol{y}_i, i = 1 \cdots N_{se}$ and minimizes a regularized risk criteria. The whole mapping can be written as

$$f^{se}(x) = \boldsymbol{B}^{se} \cdot \psi(\boldsymbol{x}) \quad (5)$$

where $\boldsymbol{B}$ is a $d \times N$ coefficient matrix. We have d simultaneous interpolation functions each from 2D to 1D. The mapping coefficients can be obtained by solving the linear system $[\boldsymbol{y}_1^{se} \cdots \boldsymbol{y}_{N_{se}}^{se}] = \boldsymbol{B}^{se}[\psi(\boldsymbol{x}_1^{se}) \cdots \psi(\boldsymbol{x}_{N_{se}}^{se})]$. Using these nonlinear mappings, we can capture nonlinearity of facial expression in different people and expressions.

The nonlinear mappings are different for different people and for different expressions. Higher-order singular value decomposition (HOSVD) [11, 19], which is a generalization of singular value decomposition (SVD), can be applied to decompose the mapping coefficients into multiple orthogonal factors. After converting each coefficient matrix into a $N_c = d \cdot N$ dimensional vector, b^{se}, all the coefficient vectors can then be arranged in an order-three facial expression coefficient tensor $\boldsymbol{\mathcal{B}}$ with dimensionality $N_s \times N_e \times N_c$. The coefficient tensor is then decomposed as

$$\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{S} \times_2 \boldsymbol{E} \times_3 \boldsymbol{F}, \quad (6)$$

where $\boldsymbol{\mathcal{Z}}$ is a core tensor, with dimensionality $N_s \times N_e \times N_c$ which governs interactions among different mode basis matrices, $\boldsymbol{S}$, $\boldsymbol{E}$, and $\boldsymbol{F}$, representing the basis for people, expressions and pixels respectively.

Given this decomposition and given any N_s dimensional person face vector $\boldsymbol{s}$ and any N_e dimensional expression vector $\boldsymbol{e}$ we can generate coefficient matrix $\boldsymbol{B}^{se}$ by unstacking the vector $\boldsymbol{b}^{se}$ obtained by tensor product $\boldsymbol{b}^{se} = \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{s} \times_2 \boldsymbol{e}$. This can be expressed abstractly also in the generative form by arranging the tensor $\boldsymbol{\mathcal{Z}}$ into an order-four tensor $\boldsymbol{\mathcal{C}}$

$$\boldsymbol{y}_t^{se} = \boldsymbol{\mathcal{C}} \times \boldsymbol{s} \times \boldsymbol{e} \times \psi(\boldsymbol{x}_t) \ , \quad (7)$$

where dimensionality of core tensor $\boldsymbol{\mathcal{C}}$ is $d \times N_s \times N_e \times N$. The result of the tensor multiplication $\boldsymbol{\mathcal{C}} \times \boldsymbol{s} \times \boldsymbol{e}$ is a reconstruction of the coefficient matrix $\mathbf{B}^{se}$. We can analyze facial expression image sequences by estimating the state parameters $\boldsymbol{s}$, $\boldsymbol{e}$, and $\boldsymbol{x}_t$.

3.1 Facial Expression Recognition

We can recognize facial expressions by the estimated expression parameter vector $\boldsymbol{e}$. Given an input image $\boldsymbol{y}$, we need to estimate configuration $\boldsymbol{x}$, expression parameter $\boldsymbol{e}$, and personal face parameter $\boldsymbol{s}$ which minimize the reconstruction error

$$E(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{e}) = || \ \boldsymbol{y} - \boldsymbol{\mathcal{C}} \times_1 \boldsymbol{s} \times_2 \boldsymbol{e} \times_3 \psi(\boldsymbol{x}) \ || \quad (8)$$

We assume expression vector for a given image can be written as a linear combination of expression class vectors in

the training data, i.e., we need to solve for linear regression weights α such that $\boldsymbol{e} = \sum_{k=1}^{K_e} \alpha_k \boldsymbol{e}^k$ where each $\boldsymbol{e}^k$ is one of expression class vectors in the training data. Similarly for the personal face, we need to solve for weights β such that $\boldsymbol{s} = \sum_{k=1}^{K_s} \beta_k \boldsymbol{s}^k$ where each $\boldsymbol{s}^k$ is one of K_s face class vectors.

If the expression vector and the person face vector are known, then Eq. 8 is reduced to a nonlinear 1-dimensional search problem for configuration $\boldsymbol{x}$ on the unit circle that minimizes the error. On the other hand, if the configuration vector and the person face vector are known, we can obtain expression conditional class probabilities $p(\boldsymbol{e}^k|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{s})$ which is proportional to observation likelihood $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{s}, \boldsymbol{e}^k)$. Such likelihood can be estimated assuming a Gaussian density centered around $\mathcal{C} \times_1 \boldsymbol{s}^k \times_2 \boldsymbol{e} \times_3 \psi(\boldsymbol{x})$, i.e.,

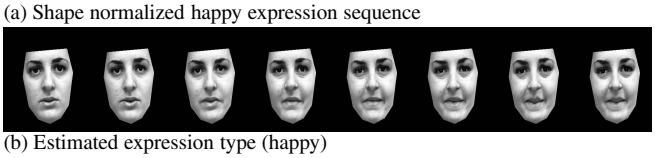
$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{s}, \boldsymbol{e}^k) \approx N(\mathcal{C} \times_1 \boldsymbol{s}^k \times_2 \boldsymbol{e} \times_3 \psi(x), \Sigma^{\boldsymbol{e}^k}).$$

Given expression class probabilities we can set the weights to $\alpha_k = p(\boldsymbol{e}^k \mid \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{s})$. Similarly, if the configuration vector and the expression vector are known, we can obtain face class weights by evaluating image likelihood given each face class $\boldsymbol{s}^k$ assuming a Gaussian density centered at $\mathcal{C} \times_1 \boldsymbol{s}^k \times_2 \boldsymbol{e} \times_3 \psi(\boldsymbol{x})$.

This setting favors an iterative procedure for solving for $\boldsymbol{x}, \boldsymbol{e}, \boldsymbol{s}$. However, wrong estimation of any of the vectors would lead to wrong estimation of the others and leads to a local minima. For example wrong estimation of the expression vector would lead to a totally wrong estimate of configuration parameter and therefore wrong estimate for person face parameter. To avoid this we use a deterministic annealing like procedure where in the beginning the expression weights and person face weights are forced to be close to uniform weights to avoid hard decisions about expression and face classes. The weights gradually become discriminative thereafter. To achieve this, we use a variable expression and person face class variances that are uniform to all classes and are defined as $\boldsymbol{\Sigma}^e = T_e \sigma_e^2 \boldsymbol{I}$ and $\boldsymbol{\Sigma}^s = T_s \sigma_s^2 \boldsymbol{I}$ respectively. The parameters T_e and T_s start with large values and are gradually reduced and in each step and a new configuration estimate is computed. Several iterations with decreasing T_e and T_s allow estimations of the expression vector, the person face vector and face configuration iteratively and allow estimations of expression and face from a single image.

Frame-based Facial Expression Recognition: For any given image frame, we can estimate facial expression parameter using iterative estimation of expression, style and configuration parameters. Estimated expressions weight can be used directly to recognize facial expression type. The maximum weight expression class corresponds to the maximum likelihood class for a given image frame. Most of the facial expression recognition systems use a peak expression image for learning and recognition of facial expression. In our model, all the frames can be used for the estimation of facial expression. In addition, estimated configuration parameter shows how close to peak expression from embedding coordinate. Fig. 2 shows estimation of expression (b) from shape normalized appearance sequence images (a). As the expression starts from a neutral expression, which is similar in all different expressions, in the database [10], estimated expression weights are similar to all four expression classes. As expression progress to the peak expression configuration in the sequence, the estimated expression weights become discriminative. From sequence of an expression we may recognize expression more robustly by selecting majority of recognized expression. However, we used peak expression image in testing facial expression classification in order to be comparable to other recognition systems using only peak expression frames in facial expression recognition.

(a) Shape normalized happy expression sequence



(b) Estimated expression type (happy)

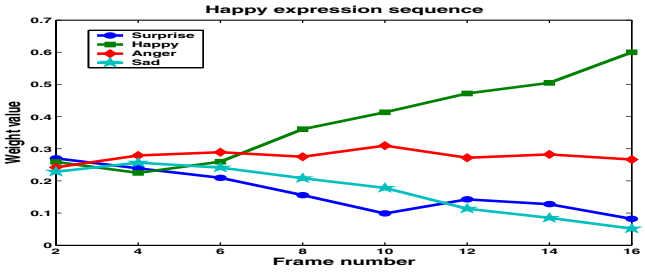


Figure 2. Frame based facial expression estimation: (a) Shape normalized happy expression sequence. (b) Estimated weights of expression class types.

The facial expression estimation in peak expression becomes an iterative estimation of expression vector and style vector as we know the configuration, the low dimensional embedding. We can embed the peak expression at the opposite location from the neutral (initial) expression in the unit circle embedding. So, we don't need to estimate configuration by full search in the embedding space. The facial expression recognition becomes finding the closest expression class with the estimated expression parameter. Fig. 3 shows typical examples of expression weight changes through the iterations. Sec. 4.2 shows experimental results using peak expression image from CMU AU coded facial expression database [10].

3.2 Facial Expression Synthesis

Our model can synthesize facial expressions by combinations of facial expression parameters and personal face

(a) Expression: surprise (b) Expression: happy

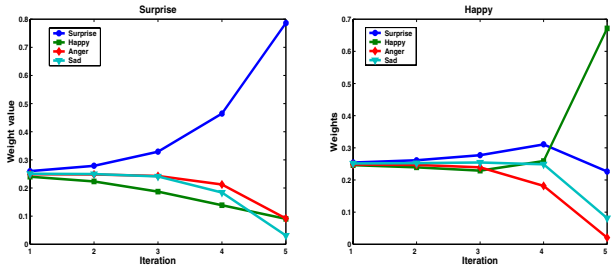


Figure 3. Expression class weight changes in iterations:(a) True expression is surprise. (b) True expression is happy. Here the highest weight changed during iteration as style estimations also changed from mean style to specific style.

parameters. As we have decomposed the mapping space that captures nonlinear deformation in facial expressions, the linear interpolation of the face style and facial expression still captures nonlinearity in the facial expression. A new personal face vector and a new facial expression vector can be synthesized by linear interpolation of existing personal face class vectors and expression class vectors using parameter α_i as follows:

$$\boldsymbol{e}^{new} = \alpha_1 \boldsymbol{e}_1 + \alpha_2 \boldsymbol{e}_2 + \cdots + \alpha_{N_e} \boldsymbol{e}_{N_e} \tag{9}$$

where $\sum_i \alpha_i = 1$. We can synthesize new personal face $\boldsymbol{s}^{new}$ similarly. A new facial expression image can be generated using new personal style and expression parameters.

$$\boldsymbol{y}_t^{new} = \mathcal{C} \times_1 \boldsymbol{s}_t^{new} \times_2 \boldsymbol{e}_t^{new} \times_3 \psi(\boldsymbol{x}_t) \tag{10}$$

4 Experimental Results

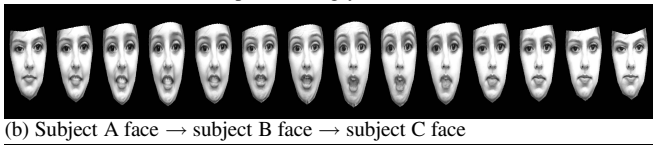
To build nonlinear generative model of facial expression and to test facial expression recognition, we selected 40 sequences of 10 female subjects ($N_s = 10$) with four emotions ($N_e = 4$: surprise, happy, angry and sad) from Cohn-Kanade AU coded facial expression database [10]. We collected shape landmark points for every other frame in each sequence. Collected sequence data contains sequences of 5 to 17 image frames. The total number of frame collected with landmark points was 399 frames. The landmarks have 38 points in each image ($n = 38$). The appearance vector was represented by 288328 pixels ($N_a = 288328$) inside landmark shape contour in the mean shape.

For any given sequence, we embed expression frames on the half circle as expression sequences in the database start from neutral expression and stop in the peak expression instead of returning to the neutral expression. The embedding configuration is parameterized in the range $\gamma = [0, 1]$ to cover half unit circle embedding space. So, $\gamma = 0$ means neutral configuration and $\gamma = 1$ means peak configuration. For facial expression recognition we used embedding which is corresponding to the peak configuration $\gamma = 1$ as described in Sec. 4.2.

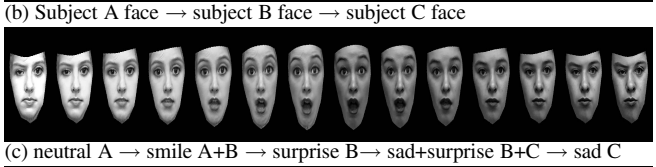
4.1 Synthesis of facial expression with personal style and expression type variations

Our model can synthesize facial expression while changing the person style and expression type parameters during performing the expression.

(a) Neutral → smile → surprise → angry



(b) Subject A face → subject B face → subject C face



(c) neutral A → smile A+B → surprise B→ sad+surprise B+C → sad C

Figure 4. Facial expression synthesis: First row: Expression transfer. Second row: Personal face transfer during smile expression. Third row: simultaneous transfer of expressions and personal faces.

Fig. 4 shows synthesis examples of new facial expressions and personal faces. During synthesis of the new images, we combine control parameter t to embedding configuration γ and interpolation parameter α and β. In case of Fig. 4 (a), the t changed $0 \rightarrow 1 \rightarrow 0$ and new expression parameter $\boldsymbol{e}_t^{new} = (1-t)\boldsymbol{e}^{smile} + t\boldsymbol{e}^{surprise}$ and then $\boldsymbol{e}_t^{new} = (1-t)\boldsymbol{e}^{angry} + t\boldsymbol{e}^{surprise}$. As a result, the facial expression starts from neutral expression of smile and animates new expression as t changes. When $t = 1$, the expression becomes a peak expression of surprise, then the expression t changes to angry and then back to neutral expression again. In the same way, we can synthesize new faces during smile expressions as in (b). Fig. 4 (c) is the simultaneous control of the personal face and expression parameters. In this case, the embedding changed from $0 \rightarrow 1 \rightarrow 0.5 \rightarrow 1$. As a result, the last synthesized expression is the peak expression of the target expression instead of a neutral expression.

4.2 Facial Expression Recognition

We tested facial expression recognition performance for ten subjects with four expressions: surprise (SP), happi-

MT	Non-GM				MLA			
ET	SP	HP	AG	SD	HP	AG	SD	SP
SP	**9**	0	0	1	**9**	1	0	0
HP	0	**10**	0	0	0	**10**	0	0
AG	0	1	**8**	1	0	1	**7**	2
SD	0	0	1	**9**	0	0	4	**6**

Table 1. Facial expression recognition with a peak expression image: Non-GM: Nonlinear Generative Model (proposed method), MLA: Multilinear Analysis in [20], MT: Applied recognition method, ET: Expression type, SP: Surprise, HP: Happy, AG: Angry,SD: Sad

ness (HP), angry (AG), and sadness (SD). Using collected shape normalized appearance, the performance was tested by leave-one-out method: we learn the model with nine people and tested the recognition performance with one person whose data are not used for learning the model. Table 1 shows recognition results when we classified facial expression using maximum expression weight of the last frame from 40 sequences. To compare the performance, we modeled facial expression by multilinear model with the same shape normalized appearance model as in [20].

The average recognition rate in our method is $90\%(\frac{36}{40})$, which is better than multilinear model $\big(80\%(\frac{32}{40})\big)$, where facial expressions for unknown person are recognized based on cosine distance of estimated expression vector using one of closest person subspace with the same shape normalized appearance data. Our proposed approach has better a model of person differences by linear combination of existing persons instead of selecting one of the known person in [20]. In addition, our model has better style and expression decomposition model as it decomposes in kernel map space and it can use all the image sequences with different number of frames for training, which is impossible in multilinear analysis in [20] as it requires aligned the same number of frames for training.

5 Conclusions

This paper presented a new approach for facial expression recognition and synthesis. The model utilized nonlinear warping of appearance for shape normalized appearance model and kernel mapping to model nonlinearity of appearance in facial expressions. The dynamics of facial expressions are also modeled using low dimensional manifold embedding of the expression configuration. The model shows better performance in facial expression recognition in addition to accurate synthesis of facial expression with geometry and expression variations simultaneously. The proposed model is a generative and has a low dimensional representation of dynamics, which are useful for the tracking of facial motions within Bayesian framework.

Acknowledgement This research is partially funded by NSF award IIS-0328991

References

[1] B. Abboud and F. Davoine. Bilinear factorisation for facial expression analysis and synthesis. *IEE Proc. Vis. Image Signal Proceess.*, 152(3), 2005.

[2] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software*, 22(4):469–483, 1996.

[3] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. PAMI*, 11(6):567–585, 1989.

[4] Y. Chang, C. Hu, and M. Turk. Probabilistic expression analysis on manifolds. In *Proc. of CVPR*, pages 520–527, 2004.

[5] E. S. Chuang, H. Deshpande, and C. Bregler. Facial expression space learning. In *Proc. of PG*, pages 68–76, 2002.

[6] T. F. Cootes. Statistical models of appearance for computer vision. Technical report, University of Manchester, 2004.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. of ECCV*, volume 2, pages 484 – 498, 1998.

[8] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: Their training and applications. *CVIU*, 61(1):38–59, 1995.

[9] A. Elgammal and C.-S. Lee. Separating style and content on a nonlinear manifold. In *Proc. CVPR*, volume 1, pages 478–485, 2004.

[10] T. Kanade, Y. Tian, and J. F. Cohn. Comprehensive database for facial expression analysis. In *Proc. of FGR*, pages 46–53, 2000.

[11] L. D. Lathauwer, B. de Moor, and J. Vandewalle. A multilinear singular value decomposiiton. *SIAM Journal On Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[12] C.-S. Lee and A. Elgammal. Facial expression analysis using nonlinear decomposable generative models. In *AMFG*, pages 17–31, 2005.

[13] Y. li Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *IEEE Trans. PAMI*, 23(2), 2001.

[14] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[15] R. W. Picard. Affective computing: Challenges. *Int. Journal of Human-Computer Studies*, 59(1–2):55–64, 2003.

[16] K. Rohr, H. S. Stiehl, T. M. Buzug, J. Weese, and M. H. Kuhn. Landmark-based elastic registration using approximating thin-plate splines. *IEEE Trans. on Medical Imaging*, 20(6):526–534, 2001.

[17] B. Schlkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

[18] T. Sim and S. Zhang. Exploring face space. In *Workshop Proceedings FPIV*, 2004.

[19] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis of image ensembles. In *Proc. of CVPR*, 2003.

[20] H. Wang and N. Ahuja. Facial expression decomposition. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 2, pages 958–965, 2003.