Bilinear Kernel Reduced Rank Regression for Facial Expression Synthesis

Dong Huang and Fernando De la Torre

Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA

Abstract. In the last few years, Facial Expression Synthesis (FES) has been a flourishing area of research driven by applications in character animation, computer games, and human computer interaction. This paper proposes a photo-realistic FES method based on Bilinear Kernel Reduced Rank Regression (BKRRR). BKRRR learns a high-dimensional mapping between the appearance of a neutral face and a variety of expressions (e.g. smile, surprise, squint). There are two main contributions in this paper: (1) Propose BKRRR for FES. Several algorithms for learning the parameters of BKRRR are evaluated. (2) Propose a new method to preserve subtle person-specific facial characteristics (e.g. wrinkles, pimples). Experimental results on the CMU Multi-PIE database and pictures taken with a regular camera show the effectiveness of our approach.

1 Introduction

Photorealistic facial expression synthesis (FES) has recently become an active research topic in computer vision and graphics. Applications of FES can be found in diverse fields such as character animation for movies and advertising, computer games, interactive education [1], video teleconferencing [2], avatars [3,4], and facial surgery planning [5]. Generating photo-realistic facial expressions still remains an open research problem due the uncanny ability of people to perceive subtle details in people's faces.

Learning-based methods (e.g. [6,7]) have become a popular approach for FES. However, the use of these methods has several challenges: (1) Muscle deformations due to expression changes can have a large number of degrees of freedom. There are more than 20 groups of facial muscles innervated by facial nerves [8]. The combinations of their movements are nearly innumerable. To model all this variability learning-based methods typically require large amounts of training samples for accurate FES. (2) Synthesis of some facial expressions requires to model subtle facial deformations, for instance wrinkles during squinting. (3) A good model should be able to decouple the identity of the subject from the expression, pose, and illumination while preserving person-specific details (e.g. pimples, beard). (4) Typically the dimensionality of the images is large in comparison with the amount of training samples which causes over-fitting of the model. To address these problems, this paper proposes Bilinear Kernel Reduced Rank Regression (BKRRR) to learn a nonlinear mapping between the frontal neutral image and images with different facial expressions of a subject. Fig. 1 illustrates the process for FES using BKRRR.

The two main contributions of this paper are: (1) Propose BKRRR for FES. BKRRR learns a nonlinear mapping from a neutral face to other facial expressions (e.g. smile,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 364–377, 2010.

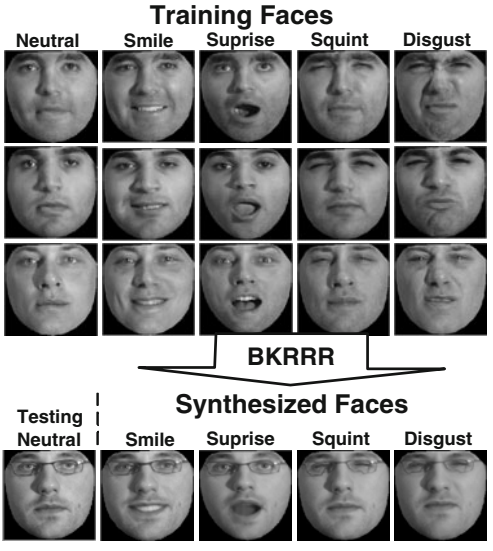


Fig. 1. Synthesizing facial expressions from a neutral face using BKRRR

surprise, squint) that effectively decouples the identity and expression changes. We explore the use of three algorithms for learning the parameters in KRRR and BKRRR, that are based on Subspace Iteration (SI), generalized eigen-decomposition, and Alternated Least Square (ALS). We evaluate the accuracy and computational complexity of each method. (2) Propose a modification of BKRRR to capture subtle person-specific facial features (e.g. glasses, pimples, wrinkles, beard).

The rest of the paper is organized as follows. Section 2 reviews related work on FES. Section 3 describes the KRRR model and three algorithms to learn the KRRR parameters. Section 4 formulates the Bilinear KRRR model, and explores its use to preserve subtle facial details not present in the training samples. Section 5 describes the experimental results, and Section 6 finalizes the paper with the conclusions.

2 Previous Work

Liu et al. [6] proposed a geometric warping algorithm in conjunction with the Expression Ratio Image (ratio between the neutral image and the image of a given expression) to synthesize new expressions preserving subtle details such as wrinkles and cast shadows. Zhang et al. [7] synthesized facial expressions using a local face model. Each region of the face was reconstructed as a convex combination of the corresponding regions in the training set. The synthesized face regions were later blended along the region boundaries. Regression-based approaches find solutions as the weighted combinations of the training data. However, it is unclear how the combination of training data can reproduce subtle local appearance features presented only in the testing samples such as wrinkles, glasses, beard, or pimples. In related work, Nguyen et al. [9] used

extensions of Principal Component Analysis (PCA) to remove glasses and beards in images, and used regression techniques to fill out the missing information.

Tensor-based approaches [10,11,12] perform Higher-Order Singular Value Decomposition (HOSVD) to factorize the normalized face appearance into identity, expression, pose, and illumination. Given the factorization, FES [13,14,15] is done by first computing the identity coefficients of the new testing person, and then reassembling the identity factor with expression factors learned by the HOSVD. A drawback of tensor-based approaches is the need of fully labeled examples across illumination, expressions, and pose. Moreover, it's also unclear how tensor-based methods can preserve subtle person-specific features (e.g. wrinkles, pimples).

Other methods learn the dynamics of the facial expression changes given several video sequences of different subjects performing the same expression. Bettinger et al. [16] used a sampled mean shift and a variable length Markov model to generate person-specific sequences of facial expressions. Zalewski et al. [17] clustered the shape and texture components with a mixture of probabilistic PCA. Each cluster corresponds to a facial expression and clusters are used for FES. Chang et al. [18] introduced a probabilistic model to learn a nonlinear dynamical model on a manifold of expressions containing the neutral and six universal expressions. In the field of computer graphics, several works used 3D models to dynamically animate avatars [19,20,21]. See [22] for a more extensive review of facial expression synthesis methods.

3 Kernel Reduced Rank Regression (KRRR)

Since its introduction in the early 1950s by Anderson [23], the reduced-rank regression (RRR) model has inspired a wealth of diverse applications in several fields such as signal processing [24] (also known as reduced-rank Wiener filtering), neural networks [25] (also known as asymmetric PCA), time series [23], and computer vision [26]. This section describes KRRR and explores three methods to compute its parameters.

3.1 Error Function for Kernel Reduced Rank Regression (KRRR)

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \Re^{d_x \times n}$ (see the footnote for notation[1]) be a matrix containing the vectorized images of neutral faces for n subjects, and $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n] \in \Re^{d_y \times n}$ contains the vectorized images of the same subjects with a different expression.

Due to lack of training samples to constrain the regression parameters, learning a linear regression between two high-dimensional data sets is usually an ill-posed problem. Consider learning the regression matrix $\mathbf{T}$ that optimizes $\min_{\mathbf{T}} ||\mathbf{Y} - \mathbf{T}\mathbf{X}||_F^2$. The optimal $\mathbf{T}$ can be found in closed-form as $\mathbf{T} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$. If $d_x > n$ the matrix $\mathbf{X}^T\mathbf{X}$ will be rank deficient. In this situation dimensionality reduction or regularization is often necessary. A common approach is to independently learn low-dimensional

[1] Bold capital letters denote matrices $\mathbf{X}$, bold lower-case letters a column vector $\mathbf{x}$. $\mathbf{x}_j$ represents the j^{th} column of the matrix $\mathbf{X}$. All non-bold letters represent scalar variables. x_{ij} denotes the scalar in the row i and column j of the matrix $\mathbf{X}$ and the scalar i^{th} element of a column vector $\mathbf{x}_j$. $\|\mathbf{x}\|_2^2$ denotes the $L2$-norm of the vector $\mathbf{x}$. $tr(\mathbf{A}) = \sum_i a_{ii}$ is the trace of the matrix $\mathbf{A}$ and $diag(\mathbf{a})$ denotes an operator that generates a diagonal matrix with the elements of the vector $\mathbf{a}$. $\|\mathbf{A}\|_F^2 = tr(\mathbf{A}^T\mathbf{A}) = tr(\mathbf{A}\mathbf{A}^T)$ designates the Frobenious norm of matrix $\mathbf{A}$.

models for each data set using PCA/KPCA, and then learn a linear or nonlinear relation between projections using any supervised learning technique (e.g. neural networks). Applying PCA/KPCA separately to each set preserves the directions of maximum variance within sets, but these do not necessarily correspond to the direction of maximum covariation between sets [26]. That is, independently learning low-dimensional models may result in a loss of important details relevant to the coupling between sets. The RRR model [23,24,25] finds a linear mapping, $\mathbf{T} \in \Re^{d_x \times d_y}$, that minimizes the LS error subject to rank constraints on $\mathbf{T}$, effectively reducing the number of free parameters to estimate. The RRR model minimizes $||\mathbf{Y} - \mathbf{TX}||_F^2$ subject to $rank(\mathbf{T}) = k$. A mathematically convenient way to impose $rank(\mathbf{T}) = k$ is to explicitly factorize $\mathbf{T} = \mathbf{BA}^T$, where $\mathbf{A} \in \Re^{d_\varphi \times k}$ and $\mathbf{B} \in \Re^{d_y \times k}$ are regression matrices, and k denotes the rank of the reduced rank model.

The Kernel RRR (KRRR) model minimizes the following energy function:

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{Y} - \mathbf{BA}^T\varphi(\mathbf{X})\|_F^2, \tag{1}$$

where $\varphi(\cdot)$ is a nonlinear function that transforms $\mathbf{X}$ to a (usually) high-dimensional feature space. The surface of Eq. (1) has a unique minimum, up to an invertible $k \times k$ affine transformation [27].

3.2 Learning Parameters for KRRR

This section explores three numerical schemes to optimize Eq. (1). The three methods are the Matlab function eigs to solve Generalized Eigenvalue Problems (GEPs), the Subspace Iteration (SI) method, and Alternated Least-Squares (ALS) procedure. We compare the computational cost as well as the error achieved by the algorithms.

1-Matlab Eigs function (EIGS): Without loss of generality the matrix $\mathbf{A}$ in Eq. (1) can be expressed as a linear combination of $\varphi(\mathbf{X})$, i.e. $\mathbf{A} = \varphi(\mathbf{X})\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \Re^{n \times k}$. $\mathbf{K} = \varphi(\mathbf{X})^T\varphi(\mathbf{X})$ is the kernel matrix such that each entry $k_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \langle\varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j)\rangle$ measures the similarity between two samples by means of a kernel function. Optimizing over $\mathbf{B}$ (i.e. $\mathbf{B} = \mathbf{YK}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}$) and substituting the optimal $\mathbf{B}$ value in Eq. (1) results in the following minimization w.r.t $\boldsymbol{\alpha}$:

$$\min_{\boldsymbol{\alpha}} tr\left\{(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}(\boldsymbol{\alpha}^T\mathbf{KY}^T\mathbf{YK}\boldsymbol{\alpha})\right\}. \tag{2}$$

Solving $\boldsymbol{\alpha}$ is a GEP, and we used the Matlab eigs function. Once $\boldsymbol{\alpha}$ is known, $\mathbf{B} \in \Re^{d_y \times k}$ can be computed with standard regression as:

$$\mathbf{B} = \mathbf{YK}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}. \tag{3}$$

2-Subspace Iteration (SI): The SI method [28] is an extension of the Power method to solve GEPs. Given two symmetric matrices, $\mathbf{S}_1 \in \Re^{n \times n}$ and $\mathbf{S}_2 \in \Re^{n \times n}$, and an initial random matrix $\boldsymbol{\alpha}_0 \in \Re^{n \times k}$, the SI method [28] alternates the following steps:

$$\mathbf{S}_1\hat{\boldsymbol{\alpha}}_{t+1} = \mathbf{S}_2\boldsymbol{\alpha}_t \tag{4}$$

$$\mathbf{S} = \hat{\boldsymbol{\alpha}}_{t+1}^T\mathbf{S}_1\hat{\boldsymbol{\alpha}}_{t+1} \qquad \mathbf{T} = \hat{\boldsymbol{\alpha}}_{t+1}^T\mathbf{S}_2\hat{\boldsymbol{\alpha}}_{t+1} \tag{5}$$

$$\mathbf{SW} = \mathbf{TW}\boldsymbol{\Delta} \tag{6}$$

$$\hat{\boldsymbol{\alpha}}_{t+1} = \hat{\boldsymbol{\alpha}}_{t+1}\mathbf{W} \qquad \hat{\boldsymbol{\alpha}}_{t+1} = \hat{\boldsymbol{\alpha}}_{t+1}/||\hat{\boldsymbol{\alpha}}_{t+1}||_F.$$

where t denotes the iteration step. In our case, $\mathbf{S}_1 = \mathbf{K}^2$ and $\mathbf{S}_2 = \mathbf{K}\mathbf{Y}^T\mathbf{Y}\mathbf{K}$. The first step, Eq. (4), of the SI algorithm solves a linear system of equations to find $\hat{\boldsymbol{\alpha}}_{t+1}$. In the second step, Eq. (5), the data is projected onto the estimated subspace. In order to impose the constraints that $\boldsymbol{\alpha}_{t+1}^T\mathbf{S}_1\boldsymbol{\alpha}_{t+1} = \boldsymbol{\Lambda}$ and $\boldsymbol{\alpha}_{t+1}^T\mathbf{S}_2\boldsymbol{\alpha}_{t+1} = \mathbf{I}_k$, a normalization is done by solving the following $k \times k$ generalized eigenvalue problem, $\mathbf{S}\mathbf{W} = \mathbf{T}\mathbf{W}\boldsymbol{\Delta}$, Eq. (6), where $\mathbf{W} \in \Re^{k\times k}$ is the eigenvector matrix. It can be shown [28] that as t increases, $\boldsymbol{\alpha}_{t+1}$ will converge to the eigenvectors of problem (2) and $\boldsymbol{\Delta}$ to the eigenvalues. The convergence is achieved when $\frac{|\delta_t^{k+1} - \delta_t^k|}{\delta_t^{k+1}} < \epsilon \ \forall i$, where δ_i^k denotes the k^{th}-largest generalized eigenvalue.

3-Alternated Least Squares (ALS): The ALS algorithm alternates between fixing $\boldsymbol{\alpha}$ and solving for $\mathbf{B}$ with Eq. (3), and fixing $\mathbf{B}$ and solving for $\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{Y}^T\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}$.

For all methods we used probabilistic PCA to factorize the matrix $\mathbf{K}$ as $\mathbf{K} \approx \mathbf{U}\mathbf{S}\mathbf{U}^T + \sigma^2\mathbf{I}_n$. This factorization is beneficial to regularize the solution and make some algorithms more efficient (e.g. solving Eq. 4). See [29] for more information.

Comparison of EIGS, SI and ALS

To evaluate the computational complexity and accuracy of the three approaches to compute the parameters in KRRR, we used 50% of the subjects from session 1 in the CMU Multi-PIE [30] database as training set. The neutral and smiling faces were used for training. We used a Gaussian kernel and the local bandwidth is selected as the mean pair-wise distance. The dimension of the images is $d_y = 35999$ pixels. The number of people $n = 125$, and k is set to $k = 37$, that preserves 99.9% of the $\mathbf{K}^2$ energy (an upper bound on the rank of the GEP).

The performance is measured using the Gradient Mean Square Error (GMSE) [12]:

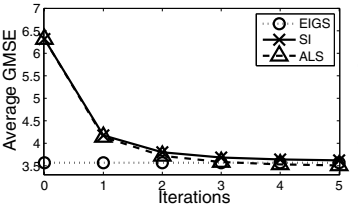
$$\text{GMSE} = \frac{1}{rc}\sum_{i=1}^{rc}\left\| \begin{bmatrix} \nabla\mathbf{F}_x(i) \\ \nabla\mathbf{F}_y(i) \end{bmatrix}_{true} - \begin{bmatrix} \nabla\mathbf{F}_x(i) \\ \nabla\mathbf{F}_y(i) \end{bmatrix}_{syn} \right\|_F^2, \tag{7}$$

between the synthesized expression and the ground truth image, where $\mathbf{F} \in \Re^{r\times c}$ is the face image of size $r \times c$ pixels, $[\mathbf{F}(i)]_{syn}$ and $[\mathbf{F}(i)]_{true}$ represent the gray level of the i^{th} pixel in the synthesized expression and the ground truth image respectively. $\begin{bmatrix} \nabla\mathbf{F}_x(i) \\ \nabla\mathbf{F}_y(i) \end{bmatrix}$ is the gradient at the i^{th} pixel. GMSE measures the difference in gradients.

Fig. 2 (a) shows the average GMSE (Eq. (7)) over 125 training subjects. As shown in Fig. 2 (a), all methods achieve similar errors. Table 2 (b) shows the computational complexity to compute $\boldsymbol{\alpha}$ using the Matlab function eigs (EIGS), the SI and ALS procedure. The time in seconds on a PC with 2.2GHz CPU was $0.077s$, 0.036s, and 1.100s for EIGS, SI and ALS respectively. The SI method achieved comparable accuracy and was more computationally efficient than ALS or eigs from Matlab.

3.3 FES with KRRR

This section shows experimental results using KRRR for FES. Given a neutral face of an untrained subject $\mathbf{x}_t$, we can synthesize a new facial expression $\mathbf{y}_t$ as a linear combination of facial expressions from the training set (i.e. $\mathbf{Y}$):



(a)

EIGS	SI	ALS
$O(n^3)$	$O(n^2k)$	$O(n^2d_y)$

(b)

Fig. 2. (a) Average GMSE to compute parameters of KRRR using Matlab eigs function (EIGS), SI, and ALS. **(b)** Computational complexity to compute $\boldsymbol{\alpha}$ using the EIGS, SI, and ALS methods.

$$\mathbf{y}_t \approx \mathbf{B}\boldsymbol{\alpha}^T\mathbf{k}(\cdot,\mathbf{x}_t) = \mathbf{Y}\mathbf{K}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}\boldsymbol{\alpha}^T\mathbf{k}(\cdot,\mathbf{x}_t) = \mathbf{Y}\mathbf{g}_t, \tag{8}$$

where $\mathbf{g}_t = \mathbf{K}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}\boldsymbol{\alpha}^T\mathbf{k}(\cdot,\mathbf{x}_t) \in \Re^{n\times 1}$ is the coefficient that weights the contributions of each training sample. $\mathbf{k}(\cdot,\mathbf{x}_t) \in \Re^{n\times 1}$ is the column vector of the kernel between the training samples and $\mathbf{x}_t$.

Note that in Eq. (8), the overall pixel intensity of $\mathbf{y}_t$ depends on the elements of the kernel vector $\mathbf{k}(\cdot,\mathbf{x}_t)$, which are close to 1 when $\mathbf{x}_t$ is close to the training data $\mathbf{X}$. However, the kernel values are smaller than 1 when $\mathbf{x}_t$ is far away. To normalize the kernel (i.e. $\sum_{j=1}^{n} g_{tj} \approx 1$), we use the Soft-Max kernel [31]:

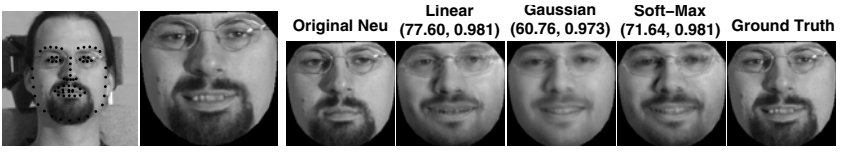
$$k(\mathbf{x}_i,\mathbf{x}_j) = \frac{exp\left(\frac{-\|\mathbf{x}_i-\mathbf{x}_j\|_2^2}{\sigma^2}\right)}{\sum_l exp\left(\frac{-\|\mathbf{x}_l-\mathbf{x}_j\|_2^2}{\sigma^2}\right)}, \quad i,j = 1,\cdots,n. \tag{9}$$

Fig. 3 shows an example of smiling facial expression synthesis using KRRR on subjects from session 1 (249 subjects) of the CMU Multi-PIE [30]. We used 50% of the subjects for training and the remaining for testing. All selected faces have been manually labeled with 66 landmarks and warped to a normalized template (see Fig. 3 (a)). The warping was done by interpolating the triangular meshes between the original landmarks and the canonical template. Note that the wrapping alone cannot result in realistic synthesis of expressions because it cannot model appearance changes (e.g. wrinkles and teeth). We compared the synthesis capabilities for three kernels: linear, Gaussian, and Soft-max. We provided two measures of error between the synthesized expression (syn) and the ground truth image (true): the average Gradient Mean Square Error (GMSE) defined in Eq. (7) and the Normalized Inner-Product (NIP):

$$\text{NIP} = \frac{1}{rc}\frac{\sum_{i=1}^{rc}[\mathbf{F}(i)]_{true}[\mathbf{F}(i)]_{syn}}{\|[\mathbf{F}]_{true}\|_F\,\|[\mathbf{F}]_{syn}\|_F}. \tag{10}$$

GMSE measures the difference in gradients, while NIP measures the correlation between gray-level values.

As can be seen in Fig. 3 the Soft-Max kernel synthesized more photo-realistic images being able to reproduce the teeth while preserving the facial hair. It also achieved the higher NIP value.



(a) (b)

Fig. 3. (a) 66 facial landmarks and the geometrically normalized face. **(b)** Synthesizing smiling faces with KRRR. Neutral test image, linear kernel, Gaussian kernel, Soft-Max kernel, and ground truth. The first number in the brackets indicates the average GMSE and the second represents the average NIP, defined in Eq. (7) and (10) respectively.

4 Bilinear Kernel Reduced Rank Regression

In the previous section, we have shown how LRRR and KRRR can be used for FES. However, observe that RRR and KRRR are unsuccessful in preserving details of the original images (e.g. wrinkles, pimples, glasses). This is because the synthesized image is a combination of the training set images, and in the training set many of these features are not present (see Fig. 3). In this section, we propose to use Bilinear KRRR (BKRRR) to effectively decouple identity and expression factors by enforcing the same identity in the synthesis of different expressions. The BKRRR is able to preserve person-specific facial features and greatly improve the synthesis performance.

4.1 Error Function for BKRRR

Let $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \Re^{d_x \times n}$ be a matrix containing the d_x dimensional input vectors representing neutral faces for n different subjects, and $\mathbf{Y}_l = \{\mathbf{y}_1, \cdots, \mathbf{y}_n\} \in \Re^{d_y \times n}$ be a matrix containing the vectorized images of the same n subjects with the l^{th} expression ($l = 1, \cdots, r$) (e.g. smile, surprise, disgust, squint, and scream). BKRRR extends KRRR, Eq. (1), by minimizing:

$$E(\boldsymbol{\alpha}, \mathbf{B}_l^{Exp}, \mathbf{B}^{Neu}) = \sum_{l=1}^{r} \|\mathbf{Y}_l - \mathbf{B}_l^{Exp}\boldsymbol{\alpha}^T\mathbf{K}\|_F^2 + \|\mathbf{X} - \mathbf{B}^{Neu}\boldsymbol{\alpha}^T\mathbf{K}\|_F^2, \quad (11)$$

recall that $\mathbf{A} = \varphi(\mathbf{X})\boldsymbol{\alpha}$ and it represents the space of identity, while $\mathbf{B}^{Neu}$ is a basis to reconstruct neutral faces and $\mathbf{B}_l^{Exp}$ is a basis for reconstructing the l^{th} facial expression. Unlike KRRR, BKRRR seeks to approximate all r expressions and the neutral face with the same identity coefficients. Observe that reconstructing the neutral testing image (second term in Eq. 11) will be a key component of our algorithm to decide which person-specific features will be able to be reconstructed as a combination of the training set. $\mathbf{K} \in \Re^{n \times n}$ is the kernel matrix containing the similarity between the neutral faces in the training samples. $\boldsymbol{\alpha}$, $\mathbf{B}^{Neu}$ and $\mathbf{B}_l^{Exp}$ are respectively computed as:

$$\min_{\boldsymbol{\alpha}} tr\left\{(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}\left[\boldsymbol{\alpha}^T\mathbf{K}\left(\sum_{l=1}^{r}\mathbf{Y}_l^T\mathbf{Y}_l + \mathbf{X}^T\mathbf{X}\right)\mathbf{K}\boldsymbol{\alpha}\right]\right\},$$

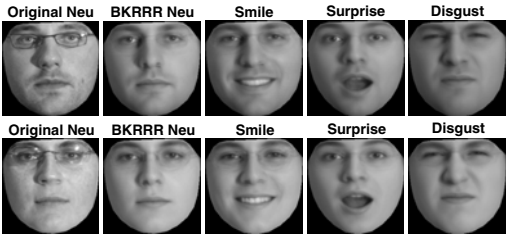


Fig. 4. Synthesis of facial expressions with BKRRR. First column shows the input image, the second the synthesized neutral image, the third, fourth and fifth show the synthesized smile, surprise and disgust expression respectively. Observe that BKRRR can not reconstruct the glasses.

$$\mathbf{B}^{Neu} = \mathbf{X}\mathbf{K}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}, \tag{12}$$

$$\mathbf{B}_l^{Exp} = \mathbf{Y}_l\mathbf{K}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}, \quad (l = 1, \cdots, r). \tag{13}$$

Similar to Section 2, solving $\boldsymbol{\alpha}$ is a GEP and we use the SI method.

The matrix $\boldsymbol{\Theta} = \boldsymbol{\alpha}^T\mathbf{K} \in \Re^{k\times n}$ in BKRRR contains subspace of identity variation. Given a new testing image $\mathbf{x}_t$, the synthesized expression can be obtained as:

$$\mathbf{y}_t = \mathbf{B}_l^{Exp}\boldsymbol{\alpha}^T\mathbf{k}(\cdot, \mathbf{x}_t), \tag{14}$$

where $\mathbf{k}(\cdot, \mathbf{x}_t)$ is the kernel vector for $\mathbf{x}_t$. Similarly, for the neutral face:

$$\mathbf{x}_t^{Neu} = \mathbf{B}^{Neu}\boldsymbol{\alpha}^T\mathbf{k}(\cdot, \mathbf{x}_t), \tag{15}$$

which approximates the neutral expression of the testing sample using the training data (2^{nd} column of Fig. 4). The synthesis of the neutral face image from the training images is important to recover subtle person-specific features and its use will be discussed in the next section. Fig. 4 also shows other synthesized expressions (smile, surprise and disgust) using the BKRRR model.

4.2 Preserving Person-Specific Features

Fig. 3 and Fig. 4 show a fundamental problem of regression approaches: the synthesized image is a combination of the data, and it is usually difficult to reconstruct subtle person-specific features of the testing image as holistic combinations of training samples. Moreover, it is not realistic to assume that the training data includes all possible iconic variations (e.g. types of glasses, beards, eyes half closed). In addition, the BKRRR minimizes a least-square error, which typically does not preserve subtle person-specific features such as pimples that might have small energy (see Fig. 3 and 4). This section shows how to combine the regression results with the synthesized neutral image to preserve subtle person-specific features.

Fig. 5 illustrates the process to preserve person-specific facial details. Given a neutral test face $\mathbf{x}_t$ (Fig. 5 (a)), we first synthesize the neutral image as a combination of the

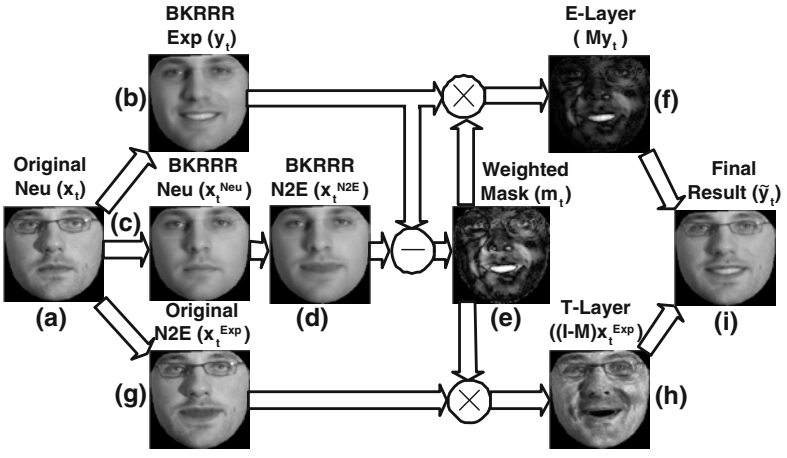


Fig. 5. FES using BKRRR that preserves person-specific facial features such as glasses and beard

training data using Eq. (15), this image is denoted as $\mathbf{x}_t^{Neu} \in \Re^{d_x \times 1}$ (Fig. 5 (c)). The resulting image $\mathbf{x}_t^{Neu}$ is warped onto the normalized template of the expression we want to target, $\mathbf{x}_t^{N2E} \in \Re^{d_y \times 1}$ (Fig. 5 (d)). We then apply BKRRR to generate $\mathbf{y}_t$ using Eq. (14) (Fig. 5 (b)). A weighted mask (Fig. 5 (e)) is computed by subtracting $\mathbf{x}_t^{N2E}$ from $\mathbf{y}_t$ as: $\mathbf{m}_t = exp\left(\frac{|\mathbf{x}_t^{N2E} - \mathbf{y}_t|}{\beta}\right)$, where $\mathbf{m}_t \in \Re^{d_y \times 1}$ denotes the weighted mask, β is a scalar selected to ensure that elements of $\mathbf{m}_t$ are between 0 and 1. The weighted mask has high values in regions where the appearance changes due to the expression variation (e.g. teeth and cheeks), and low values where the training data can not reconstruct person-specific features (e.g. glasses).

An expression layer (Fig. 5 (f)) is computed by multiplying the mask $\mathbf{M} = diag(\mathbf{m}_t) \in \Re^{d_y \times d_y}$ by the synthesized expression $\mathbf{y}_t$, that is: $\mathbf{M}\mathbf{y}_t$. This layer contains only appearance variations due to expression changes (e.g. teeth and wrinkles on the cheeks). We normalize the original neutral face $\mathbf{x}_t$ to the expression template and obtain $\mathbf{x}_t^{Exp} \in \Re^{d_y \times 1}$ (Fig. 5 (g)). Later a person-specific texture layer (Fig. 5 (h)) is created as: $(\mathbf{I} - \mathbf{M})\mathbf{x}_t^{Exp}$. Finally, the expression face $\tilde{\mathbf{y}}_t$ (Fig. 5 (i)) is computed as the combination of the expression layer and the texture layer:

$$\tilde{\mathbf{y}}_t = \mathbf{M}\mathbf{y}_t + (\mathbf{I} - \mathbf{M})\mathbf{x}_t^{Exp}. \tag{16}$$

The final result $\tilde{\mathbf{y}}_t$ greatly improves the resemblance to the original neutral test face over the result of BKRRR because it has merged person-specific features that could not be modeled by the BKRRR model.

4.3 Illumination Adaption

Fig. 6 (c) shows an example of synthesizing a smiling face using one image taken with a regular camera with uncontrolled illumination (Fig. 6 (a)). As can be observed, the

Fig. 6. Illumination normalization for FES

poor synthesis is the result of the different illumination conditions between training and testing. This section proposes a simple method to normalize illumination changes.

Fig. 6 (b) shows the illumination bias computed as the difference between the original test face and the mean face of the training set (neutral face). As can be observed, the high values of the illumination bias on the right cheek show a large difference between the training and testing lighting conditions. Fig. 6 (d) shows the results obtained after the illumination normalization.

Given the test image $\mathbf{x}_t \in \Re^{d_x \times 1}$, and the mean training face $\overline{\mathbf{x}}$, we create a representation that contains both the spatial and textural information of the image. That is, $\mathbf{F}_t = [\mathbf{l}_h, \mathbf{l}_v, \mathbf{x}_t]^T \in \Re^{3 \times d_x}$ and $\mathbf{F}^{mean} = [\mathbf{l}_h, \mathbf{l}_v, \overline{\mathbf{x}}]^T \in \Re^{3 \times d_x}$ respectively, where $[\mathbf{l}_h, \mathbf{l}_v]$ denotes the spatial location of the pixels along the horizontal and vertical axis respectively. Then we compute the linear transformation $\mathbf{M} \in \Re^{3 \times 3}$ that minimizes $||\mathbf{F}^{mean} - \mathbf{M}\mathbf{F}_t||_F^2$. The optimal matrix is $\mathbf{M} = \mathbf{F}^{mean}(\mathbf{F}_t)^+$, where $()^+$ denotes generalized pseudo-inverse. Then $\mathbf{F}_t^* = [\mathbf{l}_h, \mathbf{l}_v, \mathbf{x}_t^*]^T = \mathbf{F}^{mean}(\mathbf{F}_t)^+\mathbf{F}_t$, where $\mathbf{x}_t^*$ represents the illumination normalized testing image. Finally, to normalize the contrast of the image, the image is processed as: $\widetilde{\mathbf{x}_t} = \frac{std(\overline{\mathbf{x}})}{std(\mathbf{x})}(\mathbf{x}_t^* - mean(\mathbf{x}_t^*)) + mean(\mathbf{x}_t^*)$, where $\widetilde{\mathbf{x}_t}$ is the resulting normalized image. $std(\cdot)$ and $mean(\cdot)$ are operators that compute the standard deviation and mean respectively. Then $\widetilde{\mathbf{x}_t}$ is used to synthesize the smile expression $\widetilde{\mathbf{y}_t}$. As shown in Fig. 6 (d), the adaption algorithm greatly improves FES in images with untrained lighting conditions.

5 Experiments

This section provides quantitative and qualitative (visual) evaluation of the techniques proposed in this paper. We used all subjects (336) from the four sessions of the CMU Multi-PIE database [30]. We selected the subset of frontal faces containing 919 neutral faces, 249 smiling faces from session 1, 203 surprise faces from session 2, 203 squint faces from session 2, 228 disgust faces from session 3 and 239 scream faces from session 4 respectively. All selected faces have been manually labeled with 66 landmarks and warped to a normalized template (see Fig. 3 (a)).

5.1 FES with BKRRR

This section compares the performance of Linear Reduced Rank Regression(LRRR), KRRR, BKRRR, Tensor (HOSVD) [13,12] and BKRRRT (BKRRR with texture preservation described in section 4.2). The performance of each method is measured using GMSE (Eq. (7)) and NIP (Eq. (10)).

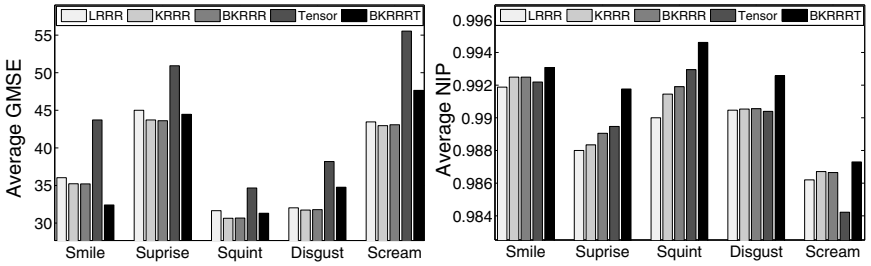


Fig. 7. Comparison of LRRR, KRRR, BKRRR, Tensor (HOSVD) and BKRRRT (BKRRR+Texture) in terms of the average GMSE (the lower the better) and average NIP (the higher the better)

Fig. 8. FES on neutral faces with subtle person-specific features (e.g. hair, wrinkle, glasses, mole and beard). The first number in brackets indicates the average GMSE and the second average NIP.

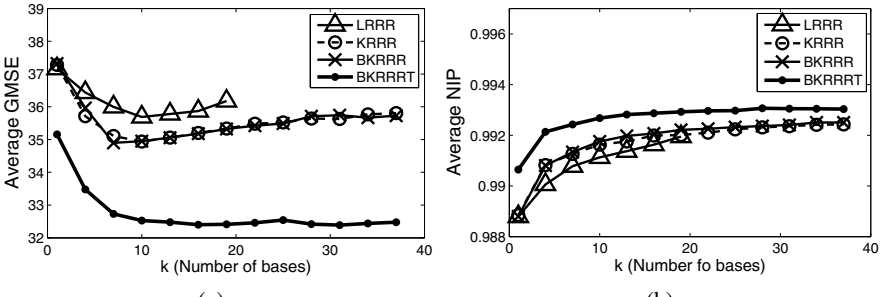
We used 50% of the faces from the CMU Multi-PIE database [30] for training (i.e. 125 for smile, 102 for surprise, 102 for squint, 114 for disgust and 120 for scream) and the remaining 50% for testing and cross-validation. In the tensor method [13], we selected all bases whose singular values are non-zero, to maximize the expressibility of

the model (as done in [13]). For LRRR, KRRR, BKRRR, and BKRRRT we selected the number of basis, k, as the number of eigenvectors that preserve 99.9% of the energy in $\mathbf{K}^2$. This is an upper bound on the rank of the RRR model. For both the KRRR and BKRRR methods, we used the Soft-Max kernel, and the regression matrices were computed using the SI method. The bandwidth parameter for the Soft-Max kernel was selected with cross-validation.

Numerical results are shown in Fig. 7. The LRRR, KRRR, BKRRR and BKRRRT methods all have smaller average GMSE than the tensor method. The BKRRR and KRRR have similar performance. However, recall that the BKRRR method is necessary to synthesize the neutral face as combination of the training samples used in the BKRRRT. The BKRRRT outperforms visually and quantitatively (in NIP) both BKRRR and KRRR. Fig. 8 shows several synthesized faces for all methods. The first column shows the original test image, the second column the neutral image, the third, fourth fifth and sixth column the synthesized image with BKRRR, LRRR, Tensor method [13] and BKRRRT respectively. Finally, the last column shows the ground truth image. Observe that BKRRRT can reconstruct much more accurately subtle facial features (e.g. glasses, skin, pimples, eyelids, hairs, mole and wrinkle) than any other method. Moreover, visually it is able to generate more photo-realistic images. On the other hand, the tensor method produces artifacts in the synthesized faces which reflects in a larger GMSE (worse preservation of edges) and smaller NIP (bad appearance matching). BKRRRT achieves the highest average NIP compared to all other methods. Observe, that occasionally the value for GMSE is higher than LRRR. This is because there is large difference in subtle edges in the original and synthesized image (e.g. rim of glasses slightly shifted), but BKRRRT achieves more photo-realistic results. Fig. 9 shows the average GMSE and NIP error versus the number of bases k to synthesize smile. As expected the error decreases w.r.t. the number of bases and BKRRRT clearly outperforms competitive approaches. For more results see [32].

5.2 FES with Illumination Adaption

This experiment tests the ability of our algorithm to handle untrained illumination conditions. Fig. 10 shows several images that have been taken with a regular camera under



(a) (b)

Fig. 9. Average GMSE **(a)** and average NIP **(b)** versus number of bases. We used 125 testing images from the session 1 in the CMU-MultiPIE.

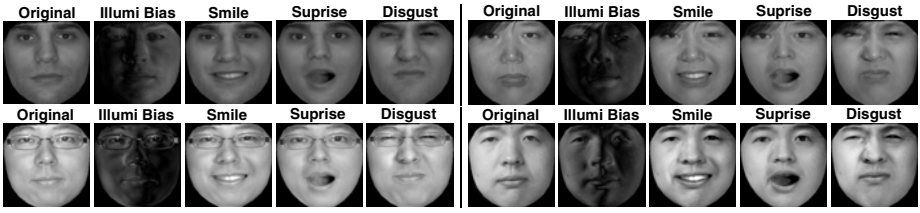


Fig. 10. FES with images taken with a regular camera under different lighting conditions. The input image is denoted by "original" and the illumination bias as "Illumi bias".

different illumination conditions. The images contain subjects of varying ethnicity. After correcting for illumination as explained in Section 4.3, our FES using BKRRRT produces very realistic results.

6 Discussion and Future Work

This paper presents a method for FES based on Bilinear KRRR. The BKRRR model learns a nonlinear mapping between a neutral face image and another image with a different facial expression of the same person. To preserve subtle person-specific features and be robust to untrained configurations, we proposed a method to combine the result of BKRRR with the original image. The results of our method are visually realistic despite the limited amount of training data. Although we have illustrated the BKRRR in the case of FES, the method is more general and can be applied to other problems image synthesis problems. In future work, we plan to improve the performance using local models (e.g. independently modeling eye, mouth and nose regions).

References

1. Gratch, J., Rickel, J., Andre, E., Cassell, J., Petajan, E., Badler, N.: Creating interactive virtual humans: some assembly required. IEEE Intelligent Systems 17, 54–63 (2002)
2. Choi, C., Aizawa, K., Harashima, H., Takebe, T.: Analysis and synthesis of facial image sequences in model-based image coding. IEEE Trans. CSVT 4, 257–275 (1994)
3. Breen, D., Lin, M.: Vision-based control of 3D facial animation. In: SCA, pp. 193–206 (2003)
4. Noh, J., Neumann, U.: Expression cloning. SIGGRAPH 1, 277–288 (2001)
5. Keeve, E., Girod, S., Kikinis, R., Girod, B.: Deformable modeling of facial tissue for craniofacial surgery simulation. Computer Aided Surgery 3, 223–228 (1998)
6. Liu, Z., Shan, Y., Zhang, Z.: Expressive expression mapping with ratio images. In Proc. of Ann. Conf. on Computer Graphics and Interactive Techniques (2001)
7. Zhang, Q., Liu, Z., Guo, B., Shum, H.: Geometry-driven photorealistic facial expression synthesis. IEEE Trans. VCG 12, 48–60 (2006)
8. Chung, K.: Gross Anatomy (Board Review). Lippincott Williams & Wilkins, Hagerstown (2005)
9. Nguyen, M., Lalonde, J., Efros, A., De la Torre, F.: Image-based shaving. Computer Graphics Forum (Eurographics) 27, 627–635 (2008)

10. Vasilescu, M., Terzopoulos, D.: Multilinear analysis of image ensembles: Tensorfaces. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 447–460. Springer, Heidelberg (2002)
11. Tenenbaum, J., Freeman, W.: Separating style and content with bilinear models. Neural Computation 12, 1247–1283 (2000)
12. Wang, H., Ahuja, N.: Facial expression decomposition. In: ICCV (2003)
13. Abboud, B., Davoine, F.: Appearance factorization for facial expression analysis. In: BMVC (2004)
14. Vlasic, D., Brand, M., Pfister, H., Popovic, J.: Face transfer with multiliner models. ACM Trans. Graphics 24, 426–433 (2005)
15. Macedo, I., Brazil, E., Velho, L.: Expression transfer between photographs through multilinear aam's. In: SIBGRAPI, pp. 239–246 (2006)
16. Bettinger, F., Cootes, T., Taylor, C.: Modelling facial behaviours. In: BMVC, vol. 2 (2002)
17. Zalewski, L., Gong, S.: Synthesis and recognition of facial expressions in virtual 3D views. In: AFGR (2004)
18. Chang, Y., Hu, C., Ferisa, R., Turk, M.: Manifold based analysis of facial expression. Image and Vision Computing 24, 605–614 (2005)
19. Kouadio, C., Poulin, P., Lachapelle, P.: Real-time facial animation based upon a bank of 3D facial expressions. In: Proc. of Computer Animation (1998)
20. Pighin, F., Szeliski, R., Salesin, D.: Resynthesizing facial animation through 3D model-based tracking. In: ICCV (1999)
21. Pyun, H., Kim, Y., Chae, W., Kang, H., Shin, S.: An example-based approach for facial expression cloning. In: SIGGRAPH/Eurographics SCA, pp. 167–176 (2003)
22. Parke, F.I., Waters, K.: Computer facial animation. AK Peters, Wellesley (1996)
23. Anderson, T.: Estimating linear restrictions on regression coefficients for multivariate normal distributions. Annals of Mathematics Statistics 12, 327–351 (1951)
24. Scharf, L.: The SVD and reduced rank signal processing. Signal Processing 25, 113–133 (2002)
25. Diamantaras, K.: Principal Component Neural Networks (Therory and Applications). John Wiley & Sons, Chichester (1996)
26. De la Torre, F., Black, M.: Dynamic coupled component analysis. In: CVPR (2001)
27. Baldi, P., Hornik, K.: Neural networks and principal component analysis: Learning from examples without local minima. Neural Networks 2, 53–58 (1989)
28. Bathe, K., Wilson, E.: Numerical Methods in Finite Element. Prentice-Hall, Englewood Cliffs (1971)
29. De la Torre, F., Gross, R., Baker, S., Kumar, V.: Representational oriented component analysis for face recognition with one sample image per training class. In: CVPR (2005)
30. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: The CMU multi-pose, illumination, and expression (multi-pie) face database. Tech. rep., Robotics Institute, Carnegie Mellon University,TR-07-08 (2007)
31. Weinberger, K., Tesauro, G.: Metric learning for kernel regression. In: AISTATS (2007)
32. Huang, D., De la Torre, F.: Bilinear kernel reduced rank regression for facial expression synthesis. Tech. rep., Robotics Institute, Carnegie Mellon University,TR-10-23 (2010)