Signal Processing 143 (2018) 171–180

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A semi-supervised framework for topology preserving performance-driven facial animation☆

Jian Zhang[a,*], Na Li[a], Yun Liang[b]

[a] School of Science and Technology, Zhejiang International Studies University, Hangzhou 310012, China
[b] College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China

ARTICLE INFO

Article history:
Received 11 March 2017
Revised 29 August 2017
Accepted 1 September 2017
Available online 6 September 2017

Keywords:
Performance-driven facial animation
Facial expression retargeting
Face driving
Semi-supervised framework
Local affine transformation

ABSTRACT

In this paper, we divide performance-driven facial animation into two data transformation problems, facial expression retargeting and face driving, and report a semi-supervised framework to solve the two problems. The objective function includes two parts. In the first part, we unify the temporal and geometrical characteristics of facial expressions and face models as topology characteristics, and preserve the topology characteristics in manifold subspace during data transformation. In the second part, some given data are used as labels to guide the transformation. The proposed semi-supervised framework can be efficiently solved by a least square method. Experimental results show that the proposed framework outperforms existing methods in both facial expression retargeting and face driving.

1. Introduction

Due to the ability to create vivid and dynamic virtual characters, performance-driven facial animation technique has been successfully applied to digital film making and interactive entertainment. Hence, it attracts considerable attention from both academic and industrial worlds.

Performance-driven facial animation comprises two key techniques, i.e., facial expression retargeting and face driving. The former technique means transforming the facial motion of a real human actor captured by certain specialized equipment to the facial motion of other virtual 3D characters. The facial motion is usually referred to motion data including sequences of frames, with each frame represented by coordinates of a group of facial feature points at that time instance. Face driving is deforming a 3D face through a group of facial feature points to generate certain kind of facial expression. We rephrase the human actor as the source object, and rephrase the virtual character as the target object. Similarly, their expressions are named as source expression and target expression respectively.

Basically, there are two classes of facial expression retargeting techniques. The first class is Blend Shape methods [1–9], which simulate the unknown facial expression by blending several existed key expressions through certain weights. The second kind of retargeting technique is facial expression cloning [10–12], which infers the unknown facial expression based on pairwise examples including existed face and its expression. The motion vectors are extracted from the sample pairs, transformed in direction and magnitude, and applied to the given face to generate facial expression.

Face driving technique can be divided into four classes. Some methods build physical muscle models for human face and drive the models based on some animation standards [13,14], which transform the displacements of the feature points to the variations of some animation parameters. Some methods achieve face driving by piecewise linear interpolation [15]. These methods triangulate the facial feature points, and project each face vertex onto a specific triangular mesh. They compute the displacement of the vertex by interpolating the displacements of the feature points. The scattered data interpolation methods use the displacements of feature points as known condition, and interpolate for the displacements of other vertices. Most frequently used interpolation method is radial basis function [16]. Some methods preserve the topology of face model during face driving, under the goal that the deformed face matches the expression feature points well [17].

However, there still exist several disadvantages in the aforementioned methods. Some facial expression retargeting methods heavily depend on the key expressions, which need to be built for each virtual character and lack reusability [3,4]. Some methods

☆ This paper is supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LY17F020009 and No. LQ14F020003, the National Natural Science Foundation of China under Grant No. 61303143, the Science and Technology Planning Project of Guangdong Province (No. 2016A050502050, No. 2015A020209124), and the NSFC-Guangdong Joint Fund (No. U1301253).

* Corresponding author.
E-mail address: jeyzhang@outlook.com (J. Zhang).

http://dx.doi.org/10.1016/j.sigpro.2017.09.002
0165-1684/

fail to preserve the temporal characteristics when transplanting the actor's facial motion to the virtual character [1,2]. In addition, some face driving methods are confined to pre-defined animation parameters, thus lack flexibility [14].

According to our definition, expression retargeting aims to solve the target expression based on the source object's face, the source expression and a few known frames of the target expression, while face driving aims to solve the vertex displacements of the source object, given the displacements of a small set of facial feature vertices. They are very similar in that both problems can be described by some semi-supervised objective function, with similar lost function constructed by certain temporal or spatial constraint. To this end, we propose a semi-supervised framework which unifies facial expression retargeting and face driving in this paper. The objective function is composed of two terms, the first term aims to preserve the temporal characteristics implied in the source expression and the spatial characteristics contained in the source object. The second term is designed to match the unknown target expression or deformed face to some given data. In this framework, the constraint is represented using manifold method, and the given data are referred to as several known frames of target expression or new positions of some facial feature points, depending on specific problems. We can optimize the objective function using least square approach. It is worthwhile to highlight several advantages of the proposed semi-supervised framework as follows:

(1) This framework does not need any key expressions for facial expression retargeting, thus remove the restriction from the key expressions and free expression retargeting from tedious manual work.
(2) Manifold method provides unified approach to representing the temporal constraint of facial expression and the spatial constraint of face driving, thus we can address the two problems in the same way.
(3) The framework can be efficiently solved by a least square method, this ensures optimized results of expression retargeting and face driving.

The rest part of this paper is organized as follows: Section 2 shows related works. In Section 3, we present the semi-supervised framework in detail, and describe how to achieve facial expression retargeting and face driving through this framework. In Section 4, we show some experimental results, and conclusion of this paper is given in the last section.

2. Related works

2.1. Facial expression retargeting

Facial expression retargeting methods can be classified into two groups. The first group of methods are Blend Shape methods [1–9], which obtain each frame of the target expression by linearly blending a set of key shapes using certain combination weights. A variety of expressions emerge when the weights are changed. The works described in [6,9] fit each frame of the source expression to the key shapes of the source object's face to get the combination weights, which were subsequently used to combine the key shapes of the target object's face to generate the target expression. Song and co-workers [1,2] parameterized the space spanned by the source object's key shapes, and learned the mapping from the parameters to the combination weights of the target object's key shapes, then created target expression through the weights. In [3,4], the author built key shapes for the target object using a sample-based method, and obtained the weights of these key shapes by learning the spatial relations among the frames of the source expression.

The second group of methods are expression cloning methods [10–12], which extract motion vectors of the vertices from source expression and adjust their directions and magnitudes to apply them to the target object's face to generate target expression. Igor et al. created target expression using the motion vectors of the source expression directly based on normalization strategy which eliminated mismatch between source and target objects' faces [11]. The work of [12] adopted radial basis function to simulate the relation between the source and the target objects' faces, and estimated the motion vectors of the target object's face from the motion vectors of the source object's face.

The aforementioned methods did not consider the characteristics of facial expressions in temporal domain, so the estimated target expression failed to present natural transitional effect. Some researchers have noticed the problem. Deng et al. [5] learned the mapping from sample faces' motion data to the weights of their key shapes, then estimated the key-shape weights of the target object by applying the mapping to the motion data of the source object. The temporal characteristics were implied by the motion data. Weise et al. [7] combined weight computation and source expression tracking into one framework and provided unified optimization method to guarantee the temporal constraint. Seol et al. [8] adopted temporal difference method to represent facial expression, and obtained the weights of target object's key shapes by solving Poisson equation. However, these methods still need key shapes, which should be built carefully by hand in advance, and this is very labor intensive work.

2.2. Feature-based face driving

Prevalent feature-based face driving can be divided into four classes. The first class of methods achieve face driving based on some animation standards, such as Facial Action Coding System (FACS) [13] and MPEG-4 standard [14]. These methods build physical models for human face according to anatomy experience, and use muscle vectors to simulate the effect of face muscles. The animation parameters of FACS or MPEG-4 standard are then converted to the muscle vectors for face driving. The weakness of these methods is that the motion vectors computed from the feature points have to be converted into the animation parameters, and the positions of muscle vectors are difficult to determine. Usually, the determination process needs tedious manual work and is a process of trial and error. In addition, these methods fail to demonstrate detailed characteristics of facial expression.

The second class of methods are piecewise linear interpolation. This kind of methods triangulate the facial feature points to form a sparse triangular mesh structure, then project each vertex of face model onto a specific mesh. Given motion vectors of the feature points contained in the mesh, the displacement of each vertex can be interpolated from these motion vectors [15]. When the number of feature points is small, the mesh structure cannot cover the whole face. Consequently, it would be difficult to compute the displacement of every vertex. Therefore, this kind of methods cannot obtain good driving result with few facial feature points.

The third kind of methods are scattered data interpolation. The most frequently used interpolation approach is radial basis function. Given motion vectors of a set of feature points, the authors of [16] used radial basis function to interpolate for the displacements of all other vertices. However, the interpolation is based on Euclidean distance measurement between the feature points and the other vertices, hence is unable to generate decent result in case of small number of feature points. This was clearly demonstrated in [18].

Recently, some topology preserving methods were applied to face driving [17,19,20]. The key point is maintaining the local spatial constraint when deforming the face model using facial feature

points. Nevertheless, the constraint adopted by these methods lacks enough representative ability for the rotation of the local facial structure during face driving. As a result, the deformed face often shows unnatural wrinkles or other artifacts in local regions.

2.3. Topology preserving learning

Deep learning has been a hot research topic in recent years. Unsupervised deep learning methods [21] discover the topology of the training data by learning the feature representations that can best reconstruct the original data. Supervised deep learning methods [22] preserve the data topology by connecting the neurons related to the salient data features in the current network layer to the neurons in the adjacent and upper layer. However, deep learning methods need a large amount of training data [23–26] that may be unavailable in many applications.

Sparse coding and multi-view learning have found their usages in image ranking [27–29] and classification [30]. The method introduced in [28] represented the topology of the original images by sparsely selected features from a hyper-graph structure. The work of [29,30] learned the topology of the images by combining varied image features into a unified representation while preserving the interdependency among the image features. However, it is quite unsure how to apply these methods to facial animation. The study introduced in [31] used a multi-view Hessian regularized logistic regression (mHLR) for human action recognition. Specifically, this method enhances the multi-view logistic regression by incorporating the multi-view Kernel penalizer for reducing complexity and the multi-view Hessian regularization term for preserving manifold structure. One of the contributions is to use Hessian regularization to preserve the local geometry.

Manifold learning is known as a group of topology preserving learning methods in that they preserve the positional interrelationship between the original data points while transforming the data points into low dimensional embeddings. Traditional manifold learning is unsupervised method, but some researcher have proposed various semi-supervised manifold learning methods by adopting prior knowledge [32–37]. Typical methods include semi-supervised Laplacian eigenmap [34], semi-supervised locally linear embedding [35]. In [36], the author expanded locally linear embedding, local tangent space alignment and ISOMAP to semi-supervised version respectively. Recently, several semi-supervised frameworks were proposed to cover variable manifold learning algorithms [38–40]. Nevertheless, these works regard manifold learning only as a tool for dimensionality reduction. The manifold regularization can be integrated with empirical risk minimization [41] for classification. In [42], the authors enhanced this framework using importance re-weighting such that the method achieves classification with noisy labels. In addition, they solved the problem of noise rate estimation in this study.

Some recently emerged methods share similarity with manifold learning in the ability of topology preservation [17,43–45]. However, these methods were not designed for dimensionality, but for object tracking in video streams [43] and 3D [17] or 2D [44,45] object deformation. In [43], the authors assumed that the video sequence and the object motion trajectory lied on two manifolds that had similar topology structure, then they proposed a semi-supervised method to address the object tracking problem by use of some prior knowledge about the object position. This method adopts linear regression to model the correlation between video frames and object positions, therefore the estimation of the object trajectory is not accurate, especially when the prior knowledge is not enough. In [17], the authors assumed that the deformed face had the same local geometric structure as the original face, and the local structure was depicted by the linear reconstruction error of each vertex by its neighboring vertices. Given new coordinates of some facial feature points, the method achieved face driving in a semi-supervised way. The weakness of this method is that the local structures are not sensitive to rotation, therefore cannot represent local facial details when rotation of local facial region exists. Though the local structures could be enhanced by appending local rotations for each vertex, this problem still cannot be perfectly solved.

3. The semi-supervised framework for facial animation generation

3.1. The semi-supervised framework

In facial expression retargeting, the temporal characteristics of the source expression are required to be maintained. In face driving, the geometrical characteristics of the source object's face should be preserved. If we regard both the frames contained in expression sequence and the vertices of face model as data points, facial expression retargeting and face driving can be converted into similar data transformation process which transforms source data into target data. Meanwhile, the temporal and geometrical characteristics can be unified by certain topology characteristics of the data set. Recent research indicates that data sets with similar topology characteristics have similar low-dimensional manifold structures [46–49], which are usually constructed by some local low-dimensional structures. To this end, we preserve the topology characteristics of data set by utilizing the similarity between the local low-dimensional structures of the source data and the target data, then adopt a number of labeled data points as supervision information to construct the semi-supervised framework. This is the main concern of our method.

Specifically, we construct a local structure using a data point and its neighbors, and represent its low-dimensional counterpart as local tangent coordinates based on a local principal component analysis (PCA). We assume that the local tangent coordinates at a data point of the source data set differ from that of the target data set by a local linear transformation. It is known that a local structure of the target data set can be easily reconstructed by its local tangent coordinates through an inverse projection of the local PCA. Hence, the unknown target data can be computed through a serials of local affine transformations, each was imposed on the local tangent coordinates at a specific data point of the source data set. The framework is depicted in Fig. 1.

3.1.1. The objective function

We denote the source data set as $X = \{x_1, x_2, \ldots, x_N\}$ where $x_i \in R^D$ is a data point, and represent the local structure at x_i as $X_i = \{x_{i_1}, x_{i_2}, \ldots, x_{i_{k_i}}\}$ including x_i itself. The local tangent coordinates of the local structure can be computed by a local PCA as

$$\tilde{x}_i^j = U_i^T (x_{i_j} - \bar{x}_i), \quad j = 1, \ldots, k_i \tag{1}$$

where $\bar{x}_i$ is the mean of X_i, $U_i = [u_i^1, \ldots, u_i^d]$ can be computed as the left singular vectors corresponding to the d largest singular values of $X_i - \bar{x}_i e^T$, and d is the dimension of the manifold. Eq. (1) indicates that a source data point x_{i_j} can be reconstructed as

$$x_{i_j} \approx U_i \tilde{x}_i^j + \bar{x}_i. \tag{2}$$

Suppose $Y = \{y_1, y_2, \ldots, y_N\}$ is the target data set, and $Y_i = \{y_{i_1}, y_{i_2}, \ldots, y_{i_{k_i}}\}$ is the local structure at y_i, the local tangent coordinates of Y_i can be similarly computed as

$$\tilde{y}_i^j = V_i^T (y_{i_j} - \bar{y}_i), \quad j = 1, \ldots, k_i, \tag{3}$$

where $\bar{y}_i$ is the mean of Y_i, and $V_i = [v_i^1, \ldots, v_i^d]$ are the left singular vectors corresponding to the d largest singular values of

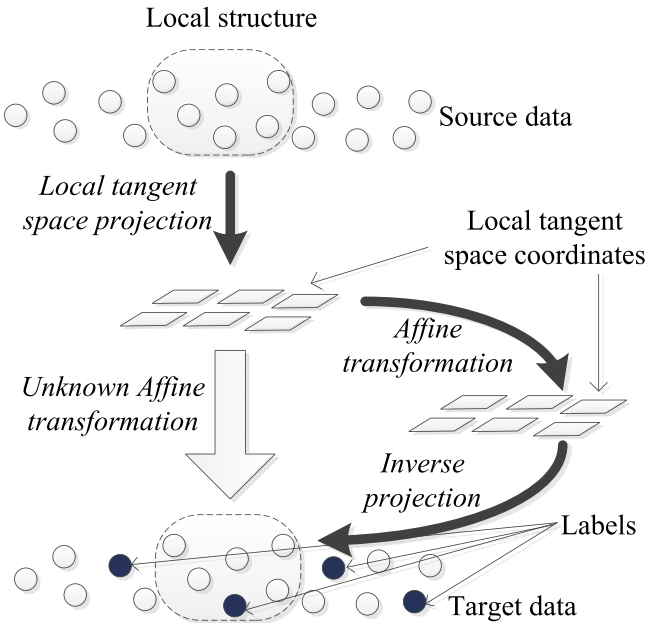

Fig. 1. The proposed semi-supervised framework.

$Y_i - \bar{y}_i e^T$. Similar to (2), we have

$$y_{i_j} \approx V_i \tilde{y}_i^j + \bar{y}_i. \tag{4}$$

According to our assumption, there exists a local linear transformation between the local tangent coordinates at a data point of the source data set and that of the target data set. Let Q_i be the local linear transformation at the ith data point, we accordingly have

$$\tilde{y}_i^j = Q_i \tilde{x}_i^j. \tag{5}$$

Substituting (5) into (3), we then obtain $Q_i \tilde{x}_i^j = V_i^T (y_{i_j} - \bar{y}_i)$, from which a target data point y_{i_j} can be reconstructed as $V_i Q_i \tilde{x}_i^j + \bar{y}_i$. Let $V_i Q_i = D_i$, $\bar{y}_i = c_i$, we obtain

$$y_{i_j} \approx D_i \tilde{x}_i^j + c_i$$

which indicates that the target data point differs from the local tangent coordinate of corresponding source data point by an affine transformation. The errors of the optimal affine transformation at the ith data point is then given by

$$\min_{D_i, c_i} \sum_{j=1}^{k_i} \|y_{i_j} - (D_i \tilde{x}_i^j + c_i)\|_2^2. \tag{6}$$

To compute the target data points $\{y_1, y_2, \ldots, y_N\}$, we need to solve the following problem

$$\min_{D_i, c_i, y_{i_j}} \sum_{i=1}^{N} \sum_{j=1}^{k_i} \|y_{i_j} - (D_i \tilde{x}_i^j + c_i)\|_2^2. \tag{7}$$

In the meanwhile, we choose a number of data points from the source data set, and assign new values for these data points. These new values act as labels which are used to guide the transformation process. For simplicity, we can rearrange the positions of the data points such that the first M data points are labeled. Let $\hat{y}_i (i = 1, \ldots, M)$ be the labels of the selected data points, we then add the following term to problem (7):

$$\sum_{i=1}^{M} \|y_i - \hat{y}_i\|_2^2. \tag{8}$$

Considering both (7) and (8), the semi-supervised framework can be denoted as:

$$\min_{D_i, c_i, y_i} \sum_{i=1}^{N} \sum_{j=1}^{k_i} \|y_{i_j} - (D_i \tilde{x}_i^j + c_i)\|_2^2 + \beta \sum_{i=1}^{M} \|y_i - \hat{y}_i\|_2^2 \tag{9}$$

where β is a regularization parameter.

3.1.2. The solution to the objective function

We rewrite the objective function of problem (9) as

$$\sum_{i=1}^{N} \|Y_i - (D_i \tilde{X}_i + c_i e^T)\|_F^2 + \beta \sum_{i=1}^{M} \|y_i - \hat{y}_i\|_2^2 \tag{10}$$

where $\tilde{X}_i = \{\tilde{x}_i^1, \tilde{x}_i^2, \ldots, \tilde{x}_i^{k_i}\}$.

Since $c_i = \bar{y}_i$, the first term of (10) can be rewritten as

$$\sum_{i=1}^{N} \|Y_i (I - ee^T/k_i) - D_i \tilde{X}_i\|_F^2, \tag{11}$$

and the optimal transformation matrix D_i is given by

$$Y_i (I - ee^T/k_i) \tilde{X}_i^+ \tag{12}$$

where $\tilde{X}_i^+$ is the Moore–Penrose generalized inverse of $\tilde{X}_i$. Substituting (12) into (11), we have

$$\sum_{i=1}^{N} \|Y_i (I - ee^T/k_i)(I - \tilde{X}_i^+ \tilde{X}_i)\|_F^2. \tag{13}$$

Let $W_i = (I - ee^T/k_i)(I - \tilde{X}_i^+ \tilde{X}_i)$, and suppose S_i is the 0-1 selection matrix s.t. $YS_i = Y_i$, (13) can be represented into matrix form:

$$\|YSW\|_F^2, \tag{14}$$

where $W = diag(W_1, \ldots, W_N)$, $S = [S_1, \ldots, S_N]$.

Let $B = SWW^T S^T$, (14) is equivalent to $trace(YBY^T)$, therefore the matrix representation of problem (9) is

$$\min_Y trace(YBY^T) + \beta \|Y - Y^l\|_F^2 \tag{15}$$

where $Y^l = [\hat{Y}, 0]$ with $\hat{Y} = [\hat{y}_1, \ldots, \hat{y}_M]$.

We can divide Y into two parts as $Y = [Y_1, Y_2]$ where Y_1 represent the data points that have labels and Y_2 are those without labels. Hence, the first term of the objective function of (15) can be transformed to

$$trace\left(\begin{bmatrix} Y_1 & Y_2 \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix} \begin{bmatrix} Y_1^T \\ Y_2^T \end{bmatrix} \right) \tag{16}$$

where B_{11} is a symmetric matrix of size $M \times M$, corresponding to the labeled data, and B_{22} is of size $(N - M) \times (N - M)$, corresponding to the unlabeled ones.

Consequently, problem (15) can be represented as

$$\min_Y trace\left(\begin{bmatrix} Y_1 & Y_2 \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix} \begin{bmatrix} Y_1^T \\ Y_2^T \end{bmatrix} \right) + \beta \|Y_1 - \hat{Y}\|_F^2. \tag{17}$$

The objective function of (17) is quadratic. Under weak assumptions, it can be shown that this function has a symmetric positive definite Hessian matrix, therefore, its minimization can be computed by solving the following linear system of equations:

$$\begin{bmatrix} B_{11} + \beta I & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix} \begin{bmatrix} Y_1^T \\ Y_2^T \end{bmatrix} = \begin{bmatrix} \beta \hat{Y}^T \\ 0 \end{bmatrix}. \tag{18}$$

Obviously, we can represent the closed form solution of the problem as:

$$Y = \begin{bmatrix} \beta \hat{Y} & 0 \end{bmatrix} \begin{bmatrix} B_{11} + \beta I & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix}^{-1}. \tag{19}$$

The algorithm of the semi-supervised framework can be summarized as Algorithm 1.

Algorithm 1 The algorithm of the semi-supervised framework for performance-driven facial animation.

Input:

source data set $X = \{x_1, x_2, \ldots, x_N\}$, and a group of labels $\hat{Y} = \{\hat{y}_1, \ldots, \hat{y}_M\}$ assigned to some selecteddata points, $\{x_1, \ldots, x_M\}$ for simplicity, from the sourcedata set.

Output:

The target data set $Y = \{y_1, y_2, \ldots, y_N\}$;

1: Initialize matrix B as zero.
2: **for** All the data points x_i in X **do**
3: Determine $k_i - 1$ nearest neighbors of x_i according tosome strategy, and form the local structure X_i;
4: Compute the d largest unit singular vectors $g_1, g_2, \ldots, g_d$ of $(X_i - \bar{x}_i e^T)^T (X_i - \bar{x}_i e^T)$, and set $G_i = [e/\sqrt{k_i}, g_1, g_2, \ldots, g_d]$;
5: Update matrix B by $B(I_i, I_i) = B(I_i, I_i) + I - G_i G_i^T (i = 1, \ldots, N)$, where I_i is each local structure's index set;
6: **end for**
7: Obtain the closed form solution of target data Y through (19).

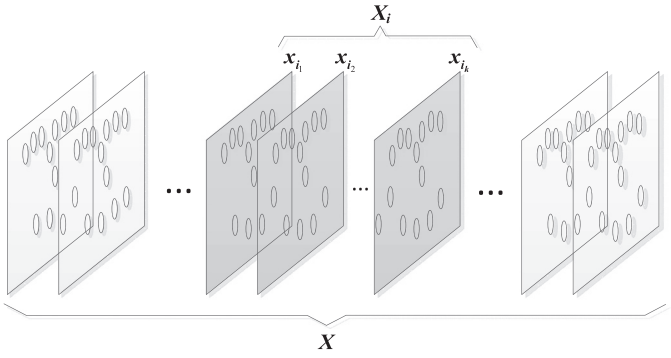

Fig. 2. The local structure of a source expression.

3.2. The application of the framework to facial expression retargeting

In facial expression retargeting, we represent the source expression X as a sequence of motion data, with each frame of the motion data x_i denoting a data point, and define the local structure at x_i as a consecutive sequence of frames including the current frame and several adjacent frames before and after the current frame in temporal domain. Fig. 2 shows a local structure X_i of a source expression X, where the frames belonging to X_i are rendered in dark color.

Similarly, we represent the target expression Y as the unknown motion data of a new virtual character, and solve the motion data under the condition that some frames of the target expression $\hat{Y}$ are already known. $\hat{Y}$ can be seen as labels, and can be rearranged to be the foremost frames of the motion data. As discussed in Section 3.1, this is a semi-supervised problem, and can be accordingly solved by Algorithm 1. Fig. 3 explains how to apply Algorithm 1 to facial expression retargeting.

3.3. The application of the framework to face driving

In face driving, we represent the source object's face X as a 3D face model comprising a large number of 3D vertices, with each vertex x_i denoting a data point, and define the local structure at x_i as several vertices including the current vertex and its surrounding vertices. Fig. 4 shows a local structure X_i of a source face, where the vertices belonging to X_i are rendered in dark color.

Similarly, we represent the target object's face Y as the unknown 3D face after model deformation, and solve the 3D face under the condition that some vertices of the 3D face $\hat{Y}$ are already known. $\hat{Y}$ can be seen as labels, and can be rearranged to be the foremost vertices of the face model. This is also a semi-supervised problem, and can be solved by Algorithm 1. Fig. 5 explains how to apply Algorithm 1 to face driving.

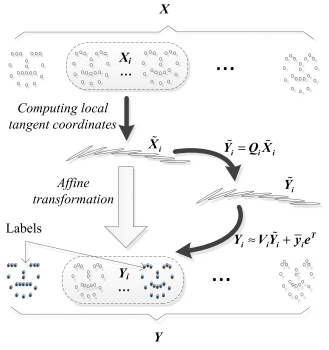

Fig. 3. Facial expression retargeting based on the proposed framework.

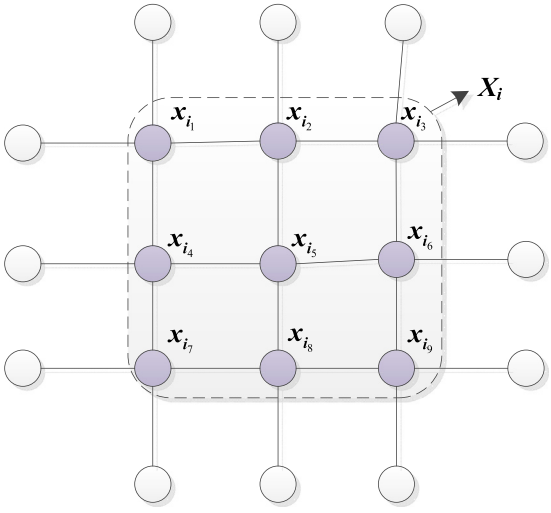

Fig. 4. The local structure of a source object's face.

4. Experiments and discussions

In this section, we will present several experiments to demonstrate the superiority of the proposed framework to some existing methods in both facial expression retargeting and face driving. First, the number of the labeled data is quite important to the proposed semi-supervised framework, so we will evaluate the influence of the labels' number to the learning results at first. After determining the number of labels, we will compare the facial expression retargeting result and face driving result of several approaches, including the proposed framework, to testify the accuracy of the framework. At last, we demonstrate the visual effect of the performance-driven facial animation based on the proposed framework.

To evaluate facial expression retargeting result and face driving result of different approaches, we need not only the source objects' faces and source expressions but also the corresponding target objects' faces and target expressions so as to compare the synthesized target faces and target expressions with the true ones. To this end, we build pairwise 3D virtual face models using 3Ds Max software, where each pair includes a source face with no expression and corresponding target face with certain expression. Also, we ask an experienced animator of our team to generate facial expression sequences of source faces and the same expression sequences of

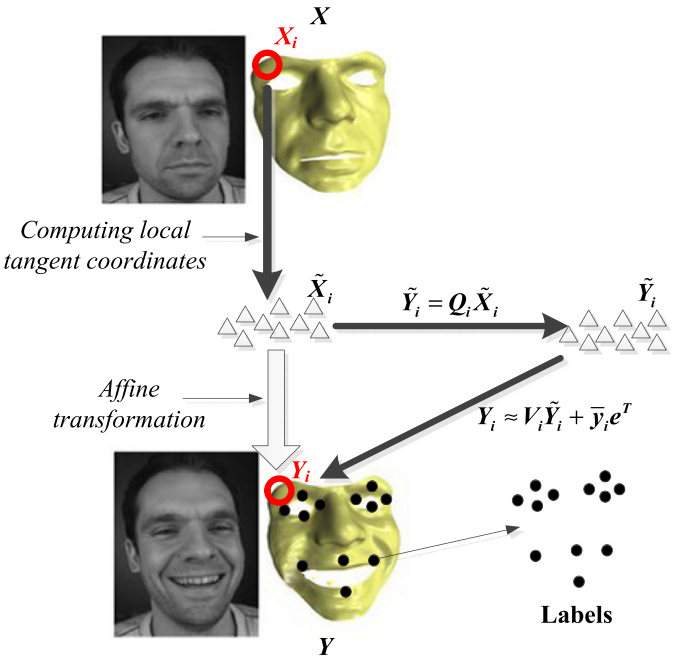

Fig. 5. Face driving based on the proposed framework.

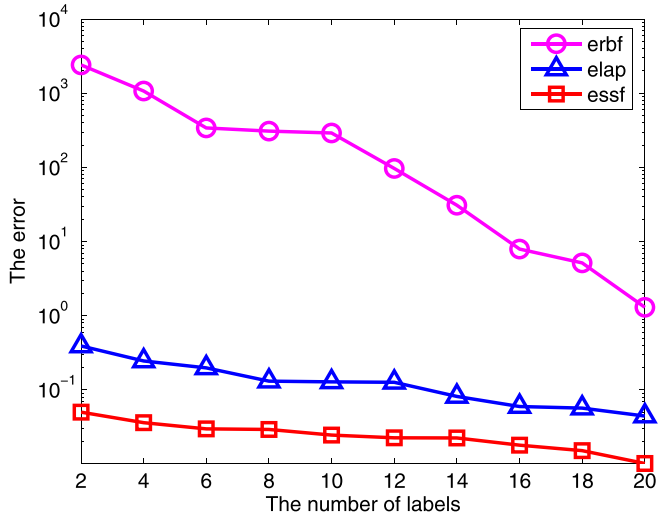

Fig. 6. The error of facial expression retargeting with different number of labels.

target faces using 3Ds Max, and select some facial feature points from the source and target sequences to construct the source expressions and target expressions. The selected feature points aim to simulate the marker points on the objects' faces. Since the labels used for facial expression retargeting and face driving are chosen from the target expression sequences and the target faces respectively, the labels are supposed to contain no noise. To compensate for this weakness, we are planning to incorporate noise reduction scheme in the proposed framework in our following study.

4.1. The influence of the labels' number to facial expression retargeting

In face driving, the labels used to deform the given source face are a set of facial feature points. As discussed above, we use these facial feature points to simulate the marker points used in the process of motion capture. Typically, the number of marker points is designated by a professional animator to generate optimized

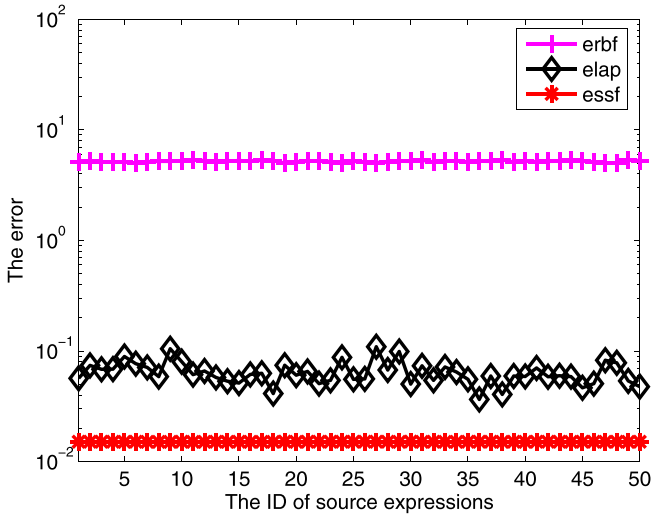

Fig. 7. The error of facial expression retargeting with respect to different source expressions.

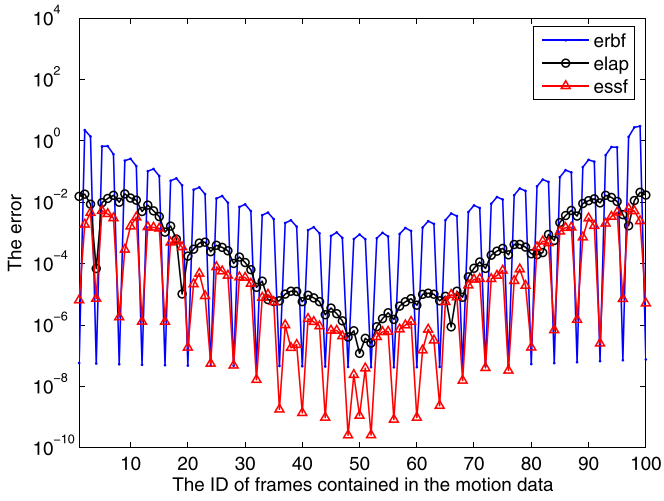

Fig. 8. The error of facial expression retargeting with respect to each frame contained in the motion data.

motion capture data, so we fix the number of facial feature points during face driving.

In facial expression retargeting, the labels are some known frames corresponding to the selected frames from the source expression, with each frame represented by a number of facial feature points depicting the expression appearance at that time instance. Different from face driving, we do not have ideal method to determine the number of labeled frames, so we need to evaluate the influence of the labels' number to expression retargeting and find an optimal value for expression retargeting.

To this end, we generate 50 source expressions and same kind of 50 target expressions, with each source expression or target expression constructed by 100 frames. Then we select a number of frames from each target expression as labels to perform the proposed framework using each source expression as input. We vary the number of labels and compute the mean error of the 50 tests with respect to different label numbers. For the sake of robustness, we conduct facial expression retargeting for each facial feature point, and integrate the motion of all feature points together according to their positions to form the target expression. In addition, the frames chosen as labels are evenly distributed to guarantee the robustness of the algorithm. We use the feature

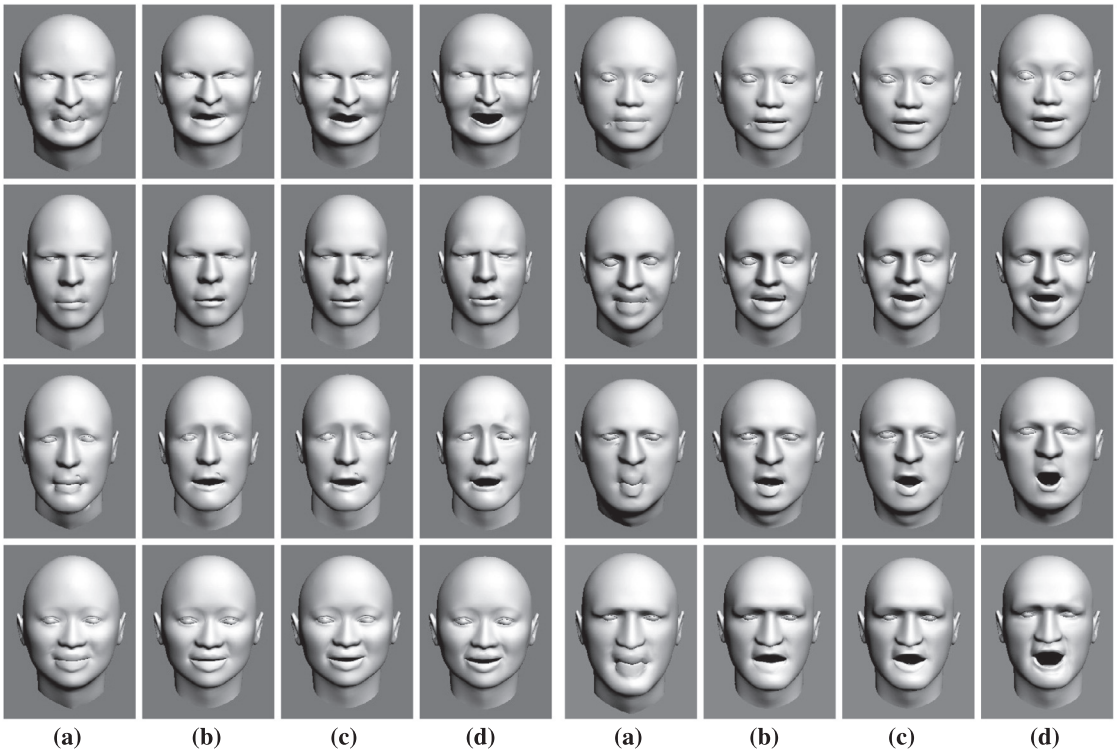

Fig. 9. The visual comparison of the face driving result using different methods, (a) radial basis function (b) Laplacian method (c) the semi-supervised framework (d) the true face.

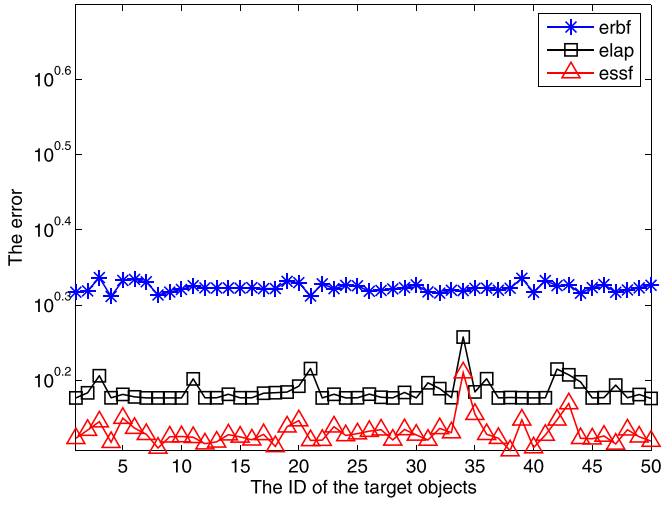

Fig. 10. The error of the face driving result using different methods.

point-based strategy in the following experiments about facial expression retargeting to ensure robustness.

Denote each source expression as $X^j(j = 1, \ldots, 50)$ and each target expression as $\hat{Y}^j(j = 1, \ldots, 50)$, and let $Y^j(j = 1, \ldots, 50)$ be the generated target expression, we compute the mean error of the 50 tests as $\frac{1}{50}\sum_{j=1}^{50} \|Y^j - \hat{Y}^j\|_F$. Note that X^j here represents 100 consecutive facial expression frames, with each frame denoted by a set of facial feature points, and so does Y^j. Fig. 6 shows the mean error of facial expression retargeting with respect to different label numbers, where erbf represents the error of radial basis function (RBF) interpolation [16], elap represents the error of Laplacian method [17], and essf denotes the error of the proposed semi-supervised framework. In the rest part of the paper, erbf, elap and essf have the same meaning as what they represent here. Fig. 6 indicates that for an expression sequence of 100 frames, 6 labels are enough for our approach to generate decent target expression. The error tends to be very tiny when the number of labels rises to 20. Nevertheless, the error of both RBF and Laplacian methods is much higher than that of our approach. The reason is that our approach can preserve the temporal characteristics of the source expression. In the following experiments, we set the number of labels as 10 for facial expression retargeting.

4.2. The adaptability of the framework to facial expression retargeting

To testify the applicability of the proposed framework in expression retargeting, we need to evaluate the error of expression retargeting performed on various source expressions. In particular, the framework is executed on 50 different source expressions to generate 50 target expressions. Similarly, the error is computed as $\|Y^j - \hat{Y}^j\|_F$ where Y^j is the generated target expression and $\hat{Y}^j$ is the true target expression. The error is shown in Fig. 7 which indicates that the proposed framework has the smallest error on the randomly generated 50 source expressions. The Y^j here are also represented as 100 consecutive facial expression frames.

Since expression is constructed by frames, we also need to evaluate the error of each frame. To this end, we average the error of expression retargeting performed on the 50 different source expressions, and demonstrate the mean error of each frame in Fig. 8 where the error is computed as $\|y_i - \hat{y}_i\|_F$, with y_i represents the generated frame and $\hat{y}_i$ represents the true frame. Fig. 8 indicates that the proposed framework has smallest error on most frames of each expression, except for several frames in the beginning and at the end of the expression. Our method performs better because the method can preserve the source expression's temporal characteristics. In the beginning of an expression, the proposed method might lose its superiority to other methods because the temporal constraint can be weak due to the small

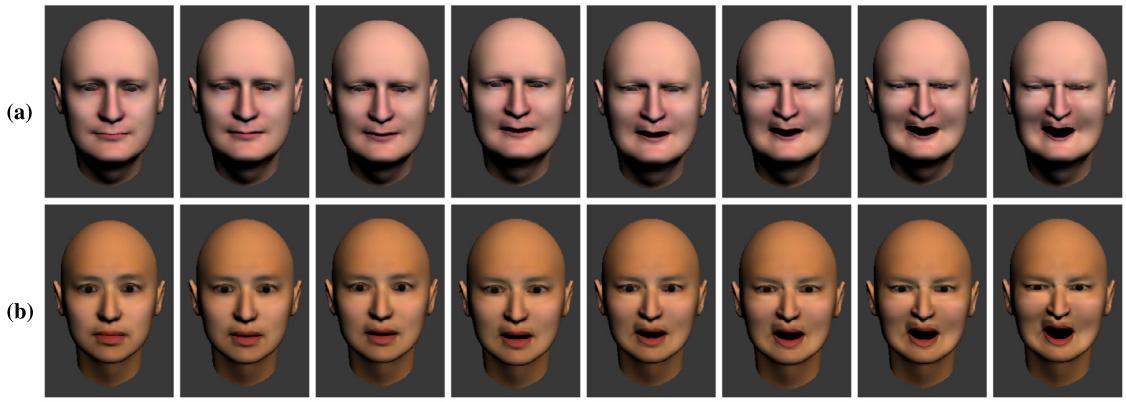

Fig. 11. The result of the performance-driven facial animation (angry expression), (a) the source expression (b) the target expression.

Fig. 12. The result of the performance-driven facial animation (fear expression), (a) the source expression (b) the target expression.

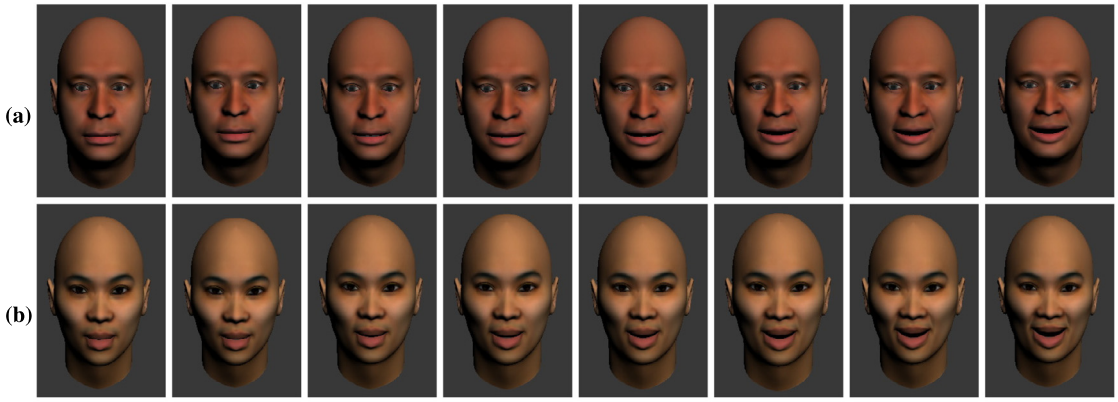

Fig. 13. The result of the performance-driven facial animation (happy expression), (a) the source expression (b) the target expression.

intensity of face variation. The same thing happens at the end of an expression sequence. Though the error of our approach fluctuates between 10^{-10} and 10^{-2}, it would not influence the visual appearance of the generated target expression due to the small magnitude.

4.3. The adaptability of the framework to face driving

We generate 50 pairs of faces, with each pair includes a source face and a target face, and ask the animator to pick up an optimized set of feature points from the target face as labels, then we conduct face driving using the proposed framework. Each face have 6174 vertices and the number of feature points are designated as 80. The driving results of 4 randomly selected source faces by various approaches are demonstrated in Fig. 9 where (a) denotes RBF interpolation, (b) denotes Laplacian method, (c) corresponds to the proposed framework and (d) means the ground truth. We discover from Fig. 9 that RBF interpolation cannot generate decent results, especially when the mouth of target face is open. This is because the RBF interpolation is based on Euclidean distance metric between feature points and other vertices, and Euclidean distance sometimes cannot correctly reflect the true distance between two vertices. Our framework and Laplacian method share similar results, but our results are closer to the true target faces. The reason is that the topology constraint is imposed on each local structure of the face. On the contrary, the constraint of Laplacian

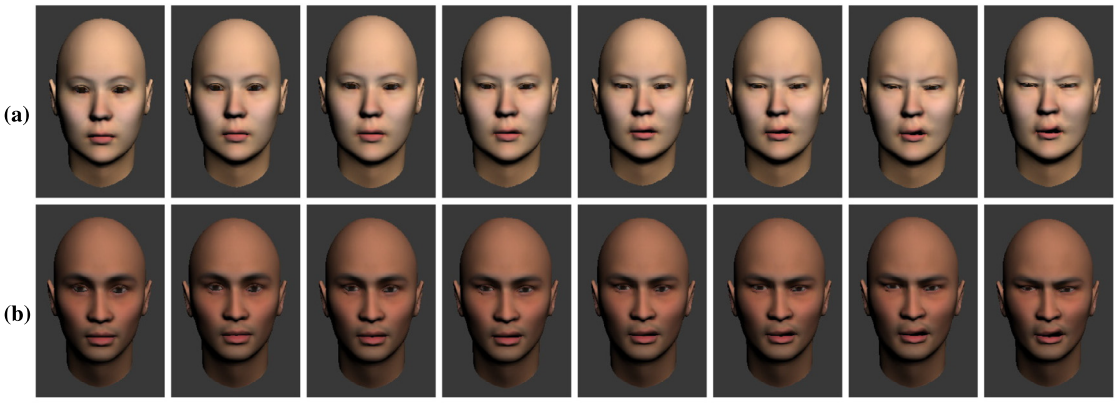

Fig. 14. The result of the performance-driven facial animation (disgust expression), (a) the source expression (b) the target expression.

method is imposed on each vertex of the face, so the coordinates of adjacent vertices may have large deviation.

Fig. 10 plots the error of 50 face driving tests where we can find that the proposed framework has highest accuracy compared with RBF interpolation and Laplacian approach, and this manifests the adaptability of the proposed framework to face driving.

4.4. The visual effect of the performance-driven facial animation

We demonstrate the visual effect of performance-driven facial animation based on the proposed framework. Specifically, we create four source faces and four target faces through 3Ds Max, with each face has 6174 vertices and realistic textures. For each source face, we build an expression sequence which includes 100 face models demonstrating different extent of this expression. Then we synthesize 10 face models as labels based on RBF interpolation using the source face and a target face as interpolation condition. We select 80 facial feature points from each face model to construct source expression, and perform facial expression retargeting based on these facial feature points. At last, we use the synthesized target expression to drive the target face to generate expression deformation. The experiment is repeated four times for the four source expressions, and 8 frames of selected driving results of four typical expressions are shown in the Figs. 11–14. The four figures show that our proposed framework can generate decent facial expression and is very suitable for performance-driven facial animation.

5. Conclusion

This paper reports a semi-supervised framework for performance-driven facial animation. This framework unifies the approaches to facial expression retargeting and face driving into a semi-supervised data transformation method, which uses local data structures' tangent coordinates to construct topology constraints and assigns labels for a few selected data points as prior knowledge. We propose algorithms for both facial expression retargeting and face driving based on the framework. Compared with several existing methods, this framework not only achieves decent facial expression retargeting without the support from any sample data, but also obtains better face driving results. The results of the performance-driven facial animation have excellent visual appearance, this indicates that the proposed framework has potential applied value in character animation generation.

References

[1] J. Song, B. Choi, Y. Seol, J. Noh, Characteristic facial retargeting, Comput. Animat. Virtual Worlds 22 (2–3) (2011) 187–194.
[2] H. Pyun, Y. Kim, W. Chae, H. Kang, S. Shin, An example-based approach for facial expression cloning, in: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, 2003, pp. 167–176.
[3] P. Kim, Y. Seol, J. Song, J. Noh, Facial retargeting by adding supplemental blendshapes, in: Proceedings of the 2011 Pacific Graphics, Kaohsiung, 2011, pp. 89–92.
[4] Y. Yang, N. Zheng, Y. Liu, S. Du, Y. Su, Y. Nishio, Expression transfer for facial sketch animation, Signal Process. 91 (2011) 2465–2477.
[5] Z. Deng, P. Chiang, P. Fox, U. Neumann, Animating blendshapebfaces by cross-mapping motion capture data, in: Proceedings of the 2006 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, Redwood City, 2006, pp. 43–48.
[6] E. Chuang, C. Bregler, Performance driven facial animation using blendshape interpolation, Technical Report CS-TR-2002-02, Stanford University, 2002.
[7] T. Weise, S. Bouaziz, H. Li, M. Pauly, Realtime performance-based facial animation, ACM Trans. Graph. 30 (4) (2011) 77–85.
[8] Y. Seol, J.P. Lewis, J. Seo, B. Choi, K. Anjyo, J. Noh, Spacetime expression cloning for blendshapes, ACM Trans. Graph. 31 (2) (2012) 14.
[9] Q. Zhang, Z. Liu, B. Guo, D. Terzopoulos, H. Shum, Geometry-driven photorealistic facial expression synthesis, IEEE Trans. Vis. Comput. Graph. 12 (1) (2006) 48–60.
[10] J. Noh, U. Neumann, Expression cloning, in: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, 2001, pp. 277–288.
[11] I.S. Pandzic, Facial motion cloning, Graph. Models 65 (2003) 385–404.
[12] M. Fratarcangeli, M. Schaerf, R. Forchheimer, Facial motion cloning with radial basis functions in MPEG-4 FBA, Graph. Models 69 (2007) 106–118.
[13] Y. Zhang, E. Prakash, E. Sung, A new physical model with multilayer architecture for facial expression animation using dynamic adaptive mesh, IEEE Trans. Vis. Comput. Graph. 10 (3) (2004) 339–352.
[14] W. Gao, Y. Chen, R. Wang, S. Shan, D. Jiang, Learning and synthesizing MPEG-4 compatible 3-D face animation from video sequence, IEEE Trans. Circuits Syst. Video Technol. 13 (11) (2003) 1119–1128.
[15] L. Terissi, M. Cerda, J. Gomez, N. Hitschfeld-Kahler, B. Girau, R. Valenzuela, Animation of generic 3D head models driven by speech, in: Proceedings of the 2011 IEEE International Conference on Multimedia and Expo, Barcelona, 2011, pp. 1–6.
[16] J. Zhang, 3D facial expression reconstruction from video via SFM and dynamic texture mapping, J. Comput.-Aided Des. Comput. Graph. 22 (6) (2010) 949–958.
[17] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Roessl, H. Seidel, Laplacian surface editing, in: Proceedings of the 2004 EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing, Nice, 2004, pp. 175–184.
[18] J. Zhang, D. Tao, X. Bian, X. Zhan, Monocular face reconstruction with global and local shape constraints, Neurocomputing 149 (2015) 1535–1543.
[19] X. Wan, X. Jin, Data-driven facial expression synthesis via Laplacian deformation, Multimed. Tools Appl. 58 (1) (2012) 109–123.
[20] J. Zhang, J. Yu, J. You, D. Tao, N. Li, J. Cheng, Data-driven facial animation via semi-supervised local patch alignment, Pattern Recognit. 57 (2016) 1–20.
[21] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (12) (2010) 3371–3408.
[22] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. (2012) 1097–1105.
[23] J. Yu, B. Zhang, Z. Kuang, D. Lin, J. Fan, iPrivacy: image privacy protection by identifying sensitive objects via deep multi-task learning, IEEE Trans. Inf. Forensics Secur. 12 (5) (2017) 1005–1016.
[24] J. Yu, X. Yang, F. Guo, D. Tao, Deep multimodal distance metric learning using click constraints for image ranking, IEEE Trans. Cybern. (99) (2016) 1–11.
[25] J. Yu, J. Sang, X. Gao, Machine learning and signal processing for big multimedia analysis, Neurocomputing 257 (2017) 1–4.

[26] J. Zhang, K. Li, Y. Liang, N. Li, Learning 3D faces from 2D images via stacked contractive autoencoder, Neurocomputing 257 (2017) 67–78.
[27] J. Yu, D. Tao, Y. Rui, M. Wang, Learning to rank using user clicks and visual features for image retrieval, IEEE Trans. Cybern. 45 (4) (2015) 767–779.
[28] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Trans. Image Process. 23 (5) (2014) 2019–2032.
[29] J. Yu, Y. Rui, B. Chen, Exploiting click constraints and multiviewfeatures for image reranking, IEEE IEEE Trans. Multimed. 16 (1) (2014) 159–168.
[30] J. Yu, Y. Rui, T. Y, D. Tao, High-order distance based multiview stochastic learning in image classification, IEEE Trans. Cybern. 44 (12) (2014) 2431–2442.
[31] W. Liu, H. Liu, D. Tao, Y. Wang, K. Lu, Multiview Hessian regularized logistic regression for action recognition, Signal Process. 110 (2015) 101–107.
[32] H. Gong, C. Pan, Q. Yang, H. Lu, S.M. A., A semi-supervised framework for mapping data to the intrinsic manifold, in: Proceedings of the 2005 IEEE International Conference on Computer Vision, Beijing, 2005, pp. 98–105.
[33] H. Huang, H. Feng, Gene classification using parameter-free semi-supervised manifold learning, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (3) (2012) 818–827.
[34] F. Zheng, N. Chen, L. Li, Semi-supervised Laplacian eigenmaps for dimensionality reduction, in: Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hongkong, 2008, pp. 843–849.
[35] J. Ham, D. Lee, L. Saul, Semisupervised alignment of manifolds, in: Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics, Barbados, 10, 2005, pp. 120–127. (4)
[36] X. Yang, H. Fu, H. Zha, J. Barlow, Semi-supervised nonlinear dimensionality reduction, in: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, 2006, pp. 1065–1072.
[37] G. Yu, G. Zhang, C. Domeniconi, Z. Yu, J. You, Semi-supervised classification based on random subspace dimensionality reduction, Pattern Recognit. 45 (3) (2012) 1119–1135.
[38] Y. Song, F. Nie, C. Zhang, S. Xiang, A unified framework for semi-supervised dimensionality reduction, Pattern Recognit. 41 (9) (2008) 2789–2799.
[39] R. Chatpatanasiri, B. Kijsirikul, A unified semi-supervised dimensionality reduction framework for manifold learning, Neurocomputing 73 (10–12) (2010) 1631–1640.
[40] F. Nie, D. Xu, I. Tsang, C. Zhang, Flexible manifold embedding: a framework for semi-supervised and unsupervised dimension reduction, IEEE Trans. Image Process. 19 (7) (2010) 1921–1932.
[41] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, 2013.
[42] T. Liu, D. Tao, Classification with noisy labels by importance reweighting, IEEE Trans. Pattern Anal. Mach. Intell. 38 (3) (2016) 447–461.
[43] Z. Zhang, H. Zha, M. Zhang, Spectral methods for semi-supervised manifold learning, in: Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, 2008, pp. 1–6.
[44] M. Eitz, O. Sorkine, M. Alexa, Sketch based image deformation, in: Proceedings of the 2007 Vision, Modeling, and Visualization Conference, Saarbrcken, 2007, pp. 135–142.
[45] W. Huang, S. Yao, S. Guan, S. Xia, A real-time image deformation model based on line handles, J. Comput.-Aided Des. Comput. Graph. 22 (12) (2010) 2067–2072.
[46] C. Shan, S. Gong, P.W. McOwan, Appearance manifold of facial expression, in: Proceedings of ICCV-HCI 2005, Beijing, 2005, pp. 221–230.
[47] Y. Chang, C. Hu, R.O. Feris, M. Turk, Manifold based analysis of facial expression, in: Image and Vision Computing, volume 24, 2006, pp. 605–614.
[48] Y. Chang, C. Hu, M. Turk, Manifold of facial expression, in: Proceedings of the 2003 IEEE International Workshop on Analysis and Modeling of Faces and Gestures, Nice, 2003, pp. 28–35.
[49] S. Xu, Y. Jia, Facial expression manifold based on expression similarity, J. Softw. 20 (8) (2009) 2191–2198.