Manifold based Face Synthesis from Sparse Samples

Hongteng Xu
ECE Department, Georgia Tech
Atlanta, GA 30332
hxu42@gatech.edu

Hongyuan Zha
College of Computing, Georgia Tech
Atlanta, GA 30332
zha@cc.gatech.edu

Abstract

Data sparsity has been a thorny issue for manifold-based image synthesis, and in this paper we address this critical problem by leveraging ideas from transfer learning. Specifically, we propose methods based on generating auxiliary data in the form of synthetic samples using transformations of the original sparse samples. To incorporate the auxiliary data, we propose a weighted data synthesis method, which adaptively selects from the generated samples for inclusion during the manifold learning process via a weighted iterative algorithm. To demonstrate the feasibility of the proposed method, we apply it to the problem of face image synthesis from sparse samples. Compared with existing methods, the proposed method shows encouraging results with good performance improvements.

1. Introduction

In many practical applications of computer vision and signal processing, we need to generate new data points beyond those in the original data set. This problem is generally known as data synthesis. For example, given an image sequence of a rotating object, we may need to create an image corresponding to an unobserved angle based on the given sequence. This learning problem can be viewed as estimating a nonlinear function mapping from a learned parameter space to the sample space (e.g., image space). We want to emphasize that two key characteristics of data synthesis distinguish it from the traditional function interpolation problem that can be solved, for example, by simple regression methods: 1) the parameter space is not given and needs to be learned in conjunction with the nonlinear mapping; and 2) the high-dimensionality of the image space is what really makes the problem challenging.

Many nonlinear synthesis algorithms have been proposed to solve this problem. Particularly, because of their strong capability of extracting low-dimensional structural information from high-dimensional data, manifold based methods are widely applied for synthesis and learning problems for high-dimensional data sets. Focusing on human pose estimation and tracking, the works in [10, 11, 14, 19] achieve good results. However, these works are generally focused on recovering features of images, e.g., the skeleton model of body in [14, 19], the location information in [10], rather than the real image data at the pixel level. Because the dimension of parameter space is much lower than that of the image space, the capability of those methods in terms of synthesizing real high-dimensional data are still yet to be proven.

On the other hand, the works in [2, 21, 13, 5, 9, 4, 1] aim at synthesizing images directly. In [21], a Locally-Linear-Embedding (LLE) based method is proposed to reconstruct head pose images. In [5], Locally-Smooth-Manifold-Learning (LSML) is proposed to learn a warping function from a point on an manifold to its neighbors. In [3], the mapping from low-dimensional tangent space to high-dimensional sample space and the corresponding inverse mapping are learned simultaneously, which is used to recover missing data. Although these methods can synthesize full images, they can only handle relatively small local changes in the images, such as the opening and closing of eyes or mouths [2, 5], or the random missing pixels in digit images [3].

Recently, several works also try to synthesize the images having more substantial global changes. In [22], faces under unknowing lighting are synthesized according to spherical harmonic basis morphable model. In [13], the nonlinear mapping is approximated by a mixture of local PCA models and the dynamical texture image is interpolated based on a sequence. A method called local generative units and global affine transformation (LGGA) is proposed in [9], which can capture the global changes of images, including shifting, rotation and scaling. The work in [1], on the other hand, shows robustness to noisy image. However, a common issue for all the methods above is that because the manifold is fitted locally by linear models, they can produce good results only if the number of samples are relatively large. In the case of sparse samples, the underlying manifold will be poorly captured because enough neighbors of sample can-

not be found so that a good linear fit will not be feasible.

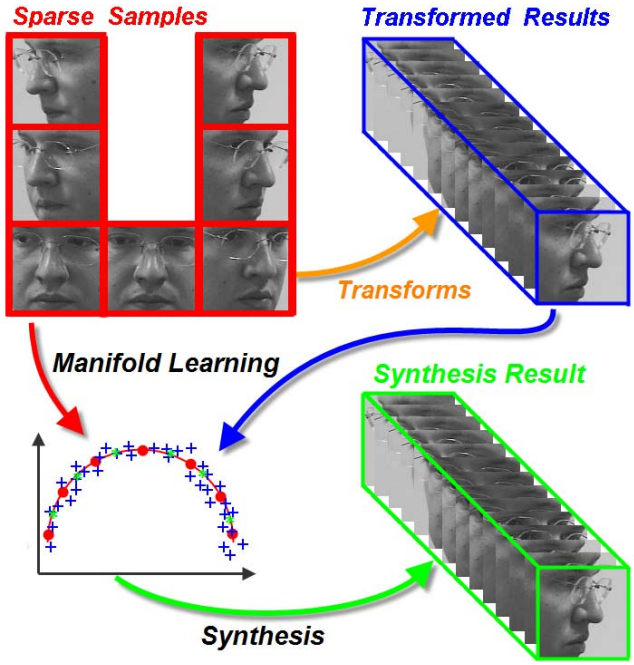

Figure 1. The scheme of the proposed method.

In this paper, we prove that it is possible to synthesize some special images based on sparse samples by leveraging the methodology of transfer learning in manifold-based image synthesis. Specifically, we use the idea of leveraging auxiliary data to enhance the learning for the target domain [18, 15, 17, 4, 16, 12]. For example, for the image classification problem in [12], for those classes with very few training samples, new training samples are borrowed from transformed samples generated from other classes. Also in [4], the visual light face images are synthesized from infrared images. Such an auxiliary data based strategy can also be applied in image synthesis. Given sparse samples, most regions of the manifold are not adequately covered. Fortunately, with the help of certain transformations, we can generate auxiliary data and then obtain a more comprehensive albeit noisier coverage of the manifold. The key difference between the proposed method and the works mentioned above is that we do not have external data set — transformations are applied to the original data points in order to generate the auxiliary data. To incorporate the noisy auxiliary data, we develop a new de-noising scheme into the manifold based synthesis methods.

The main contributions of our work are: 1) we propose a novel framework for generating auxiliary data from sparse image samples based on sample point transformations; 2) we propose a weighted data synthesis method, which adaptively selects useful samples during the manifold learning process via a weighted iterative algorithm; and 3) in the case of face synthesis, we demonstrate that under suitable transformations, the auxiliary data can be viewed as noisy data on the image manifold, which helps us to estimate the structure of manifold. The overall idea of our work is illustrated in Fig. 1.

The organization of the rest of the paper is shown as follows. Section 2 gives a brief review of manifold learning and data synthesize from dense samples. Section 3 presents the strategy for designing transformations for sparse samples and the proposed learning algorithm. The transformations applied to head pose images and the synthesis results are discussed in section 4. Finally, conclusion and future work are shown in section 5.

2. Manifold Learning and Data Synthesis from Dense Samples

The basic assumption of manifold learning is that the high-dimensional data can be viewed as a manifold embedded in the sample space. Manifold learning aims to find the low-dimensional structure of data and establish a connection (e.g., a bijection function) between samples and their coordinates in the low-dimensional latent space. According to the coordinate in the latent space and the mapping function, we can synthesize new data.

Let $X = \{\mathbf{x}_i\}_{i=1}^N$ be the high-dimensional data set, in which $\mathbf{x}_i \in R^D$. Assume they are close to a smooth d-dimensional manifold $\mathcal{M} : X \mapsto Y$, in which their coordinates in the low-dimensional parameter space are $Y = \{\mathbf{y}_i\}_{i=1}^N$. Here $\mathbf{y}_i \in R^d$ and $d \ll D$. What we want to do include: 1) learning $\mathcal{M}$ for dimensionality reduction; 2) finding the corresponding inverse mapping $\mathcal{M}^{-1}$ for data synthesis. In [5], these two problems can be addressed simultaneously by learning a warping function that maps a point on the manifold to its neighbors. Denote the warping function as $\mathcal{W} : X \mapsto X$. We have $\mathcal{W}(\mathbf{x}_i, \boldsymbol{\epsilon}) = \mathcal{M}^{-1}(\mathbf{y}_i + \boldsymbol{\epsilon})$. Here $\boldsymbol{\epsilon}$ is the distance between the samples in the latent space. Using the first order Taylor expansion of $\mathcal{M}^{-1}$, we have $\mathcal{W}(\mathbf{x}_i, \boldsymbol{\epsilon}) = \mathbf{x}_i + \mathcal{H}(\mathbf{x}_i, \boldsymbol{\Theta})\boldsymbol{\epsilon}$. The columns of $\mathcal{H}$ are partial deviations of $\mathcal{M}^{-1}$ to the elements of $\mathbf{y}_i$. $\boldsymbol{\Theta}$ is the parameter vector specifying the parametric form of $\mathcal{H}$.

To this end, we assume that $\mathcal{H}(\mathbf{x}_i, \boldsymbol{\Theta}) = \mathbf{F}_i\boldsymbol{\Theta}$, where $\boldsymbol{\Theta} \in R^{f\times d}$ is the parameter matrix we need to learn. $\mathbf{F}_i \in R^{D\times f}$ is the feature matrix extracted from $\mathbf{x}_i$. So, the manifold learning becomes a problem about learning parameter $\boldsymbol{\Theta}$ and $\boldsymbol{\epsilon}$. Denote $N_i = \{j \mid \mathbf{x}_j \text{ is the neighbor of } \mathbf{x}_i\}$. The difference between $\mathbf{y}_i$ and $\mathbf{y}_j$ is $\boldsymbol{\epsilon}_{ij} = \mathbf{y}_j - \mathbf{y}_i$. The loss function of manifold learning can be written as follows,

$$\min_{\boldsymbol{\Theta},\mathbf{y}_i} \sum_{i=1}^N \sum_{j\in N_i} \|\mathbf{x}_j - \mathbf{x}_i - \mathbf{F}_i\boldsymbol{\Theta}(\mathbf{y}_j - \mathbf{y}_i)\|_2^2 \quad (1)$$
$$\text{subject to} \quad \mathbf{Y}\mathbf{Y}^T = \mathbf{I}_d.$$

Here $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]$. $\mathbf{I}_d$ is the d-dimensional identity matrix. Each term of Eq. (1) measures the error between

the point $\mathbf{x}_i$ on the manifold and its local linear estimate. The constraint in Eq. (1) normalizes the coordinates in the latent space.

It should be noted that Eq. (1) in its current form is not well-posed, we we need to add additional constraints to $\mathbf{Y}$ and $\mathbf{\Theta}$ in order to obtain unique solutions. We address this issue using a simple strategy for reducing the computational complexity in the iterative process for the optimization problem with details shown in section 3.

Given optimal $\mathbf{\Theta}$ and $\mathbf{y}_i$, the new data corresponding to $\mathbf{y}_{new}$ can be synthesized as

$$\mathbf{x}_{new} = \sum_{j \in N_{new}} \mathbf{x}_j + \mathbf{F}_j \mathbf{\Theta} \boldsymbol{\epsilon}_{j,new}. \tag{2}$$

N_{new} is a set containing the indices of $\mathbf{y}_{new}$'s neighbors.

In the case of dense samples, the manifold can be recovered with high accuracy. However, when samples are sparse, the first-order approximation of $\mathcal{M}^{-1}$ will be very poor, mostly relying sample points that are not close to the point in question, so that we can no longer use piecewise linear model to fit the manifold. Our main idea to overcome this is to introduce auxiliary data points into the data set resulting in noisy dense samples.

3. Proposed Methods

3.1. Create Auxiliary Data via Transformations

We propose to create auxiliary data by applying certain class of transformations to the original sparse samples. Let $\mathcal{T} : R^D \mapsto R^D$ be a certain class of transformation. To each $\mathbf{x}_i$, its results of transformation are denoted as $X_i = \{\mathbf{x}_{im} | \mathbf{x}_{im} = \mathcal{T}^m(\mathbf{x}_i), m = 1, ..., M.\}$. Here $\mathcal{T}^m$ means applying $\mathcal{T}$ m times. We can view $\mathcal{T}$ as the warping function of a synthetic manifold M_i. Then, X_i is the sample set of $\mathcal{M}_i$. In our work, we assume that $\mathcal{M}_i$ is in the same latent space with the target manifold $\mathcal{M}$. It is obvious that $\mathcal{M}_i$ can be defined uniquely according to $\mathcal{T}$ and $\mathbf{x}_i$. Given a suitable $\mathcal{T}$ we can make $\mathcal{M}_i$ satisfy the following condition.

- **Condition 1:** $\mathcal{M}_i(\mathbf{x}_i) = \mathcal{M}(\mathbf{x}_i)$, and to $m = 1, ..., M$, there exists an upper bound δ such that $\|\mathcal{M}_i(\mathcal{T}^m(\mathbf{x}_i)) - \mathcal{M}(\mathcal{T}^m(\mathbf{x}_i))\| \leq \delta$.

In other words, although transformed samples are no longer on the target manifold in general, they may not be very far from the target manifold, which can be viewed as "noisy" samples for the recovery of the target manifold.

Fitting local structure merely is not enough for recovering $\mathcal{M}$ in the case of sparse samples. For example, $\mathbf{x}_i$ and $\mathbf{x}_j$ are two original samples. We can apply transformations respectively to get $\mathcal{M}_i$ and $\mathcal{M}_j$ in which X_i and X_j meet condition 1. To recover the manifold between $\mathbf{x}_i$ and $\mathbf{x}_j$, we need to find a path between them. It requires that the two synthetic manifolds have some overlaps, so that the path is composed of X_i and X_j. So, we further introduce following condition.

- **Condition 2:** To arbitrary $\mathbf{x}_i \in X$, we apply transformation $\mathcal{T}$ to create new samples X_i, whose element satisfies condition 1. A sample in X_i is available if the set of its K nearest neighbors is not a subset of X_i.

In summary, condition 1 ensures that $\mathcal{M}_i$ can fit the local structure of the target manifold. While the condition 2 ensures that the samples in $\mathcal{M}_i$ are connected with other synthetic manifolds, so that the global structure of the target manifold can be captured. As a result, the samples of $\mathcal{M}_i$ can be borrowed for learning the structure of $\mathcal{M}$ around $\mathbf{x}_i$. Fig. 2 gives an illustration of proposed method.

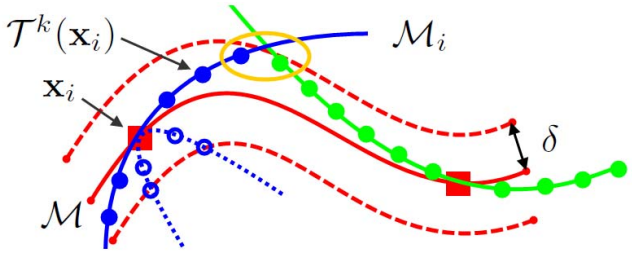

Figure 2. The red curve is the target manifold $\mathcal{M}$ and the red squares are sparse samples. The blue and green curves are synthetic manifolds created by two transformations. The "•"s are transformed samples who are in the regions defined by δ (condition 1). For some samples, their neighbors are close to the other manifold (condition 2), which is labeled by orange circles. An unavailable transformation is given by blue dot curves. The samples are blue "o"s, whose neighbors can only be found in the set created by the corresponding transformation.

By applying available transformations, we create a series of synthetic manifolds. All of the synthetic manifolds and the target manifold we want to recover are fused together in the same latent space. The auxiliary data we created is the samples of synthetic manifolds meeting condition 1 and 2. By borrowing auxiliary data from synthetic manifolds, the partial structural information of synthetic manifolds is shared by the unknown target manifold. As a result, we can fit the target manifold with synthetic manifolds, rather than a linear hyperplane. The principle of this method is based on transfer learning, which is inspired by the work in [12] where the feature space of different classes are shared and their samples can be borrowed by each other. Here, all the manifolds are in the same latent space and parts of their samples can be shared with each other. The difference is that the proposed method is not dependent on external data set — we apply transformations on the original data set to generate the auxiliary data.

Another nonlinear manifold learning work appears in [8], which fits manifold with piecewise polynomial regression, but it still requires dense samples to estimate the model

parameters. As a contrast, we do not restrain the form nonlinear regression directly. The form of the regression function is decided by the transformation, which makes the proposed method more flexible.

A concrete example of the our main idea is illustrated by face images. For sparse images of the horizontal rotation of head, we use shifting, flipping and rotation, to create auxiliary data. To fuse the manifolds into the same parameter space, we set the dimension of latent space to $d = 2$. The result of dimensionality reduction is shown in Fig. 3[1]. We observe that the auxiliary data indeed can help to recover the original manifold.

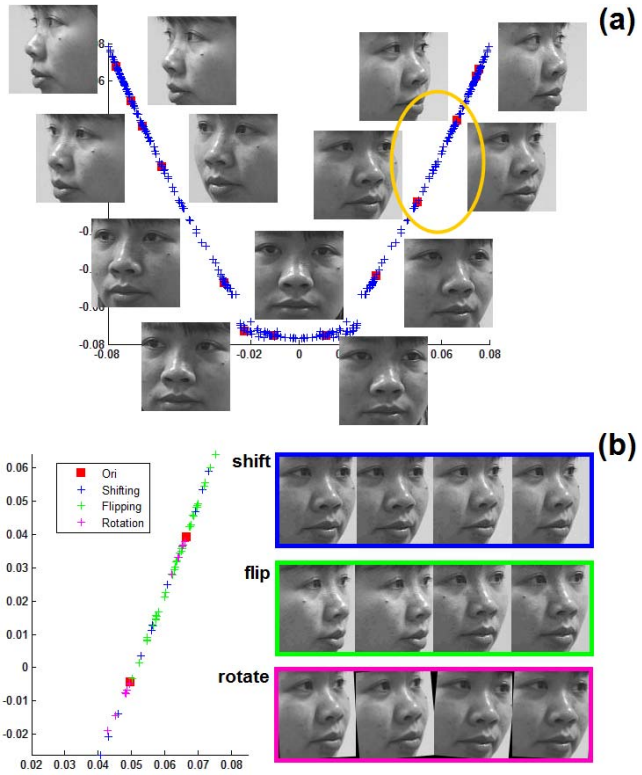

Figure 3. (a) The learning result based on original samples and the auxiliary data. Each red square corresponds to the latent variable of original image. The blue "+"s are latent variables of transformed images. (b) The enlarged figure corresponding to the yellow circle in (a). The images gotten by shifting, flipping and rotation are labeled by blue, green and pink "+"s. Some examples are given as well.

3.2. Modifications of the Loss Function

Given auxiliary data $\{\mathbf{x}_i, i = 1+N, ..., N+L\}$ created by several transformations $\{\mathcal{T}_m\}_{m=1}^{M}$, we get a new data set, where the first N samples are the original samples while the rest L ones are auxiliary data. According to the analysis above, only the samples in the auxiliary data satisfying condition 1 and 2 can be used. The rest are the outliers which may change the structure of the target manifold. To select suitable samples, we first modify the neighbor selection method according to condition 2. The scheme is described in Algorithm 1 which removes samples not meeting condition 2 from the data set.

Algorithm 1: Neighbor Selection
To $\mathbf{x}_i$, $i = 1, ..., N+L$, apply KNN
to get the set of neighbors of $\mathbf{x}_i$, $\mathcal{N}_i$.
For $i = N+1$ to $N+L$
  If $\mathbf{x}_i$ is created by $\mathcal{T}_m$, and all
  of its neighbors $\mathbf{x}_j$, $j \in \mathcal{N}_i$ are created
  by $\mathcal{T}_m$ as well, then remove
  $\mathbf{x}_i$ from data set and every $\mathcal{N}_j$, $j = 1, ..., N+L$.

For samples meeting condition 2, their neighbors are also selected adaptively. Recently, an adaptive neighbor selection algorithm is proposed based on PCA in [23]. In [6], the constraint of sparsity is added on neighbors. Both of those methods measure the weight of a neighbor based on its similarity with its center. In our work, we determine the weight with the help of a nonparametric estimate. For $\mathbf{x}_i$, the normalized weight of its neighbor $\mathbf{x}_{ik}$, denoted as $\hat{a}_{ik}$, is calculated as

$$a_{ik} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_{ik}\|_2^2}{2\sigma^2}}, \quad \hat{a}_{ij} = \frac{a_{ij}}{\sum_{k\in\mathcal{N}_i} a_{ik}}. \tag{3}$$

Here σ is the bandwidth of the Gaussian kernel. Besides weighting neighbors, we further introduce weight to each error term of the loss function. During the learning phase, the weight corresponding to the large error term should be small while that corresponding to the small error term should be large. Weighting neighbors of samples and error terms ensures that the influence of the samples far from the target manifold will be suppressed, so that the condition 1 will be satisfied. As a result of the modifications discussed above, the loss function is now rewritten as

$$\begin{aligned}
&\min_{\mathbf{\Theta},\mathbf{y}_i,\mathbf{w}} \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} \hat{a}_{ij}\|\mathbf{x}_j - \mathbf{x}_i - \mathbf{F}_i\mathbf{\Theta}(\mathbf{y}_j - \mathbf{y}_i)\|_2^2 \quad (4)\\
&+ \sum_{i=N+1}^{N+L} w_i \sum_{j\in\mathcal{N}_i} \hat{a}_{ij}\|\mathbf{x}_j - \mathbf{x}_i - \mathbf{F}_i\mathbf{\Theta}(\mathbf{y}_j - \mathbf{y}_i)\|_2^2\\
&+\lambda\|\mathbf{1} - \mathbf{w}\|_1,\\
&s.t. \quad \mathbf{Y}\mathbf{Y}^T = \mathbf{I}_d, \quad w_i = 0 \text{ or } 1.
\end{aligned}$$

Here $\mathbf{w} = [w_{N+1}, ..., w_{N+L}]^T$ is the weight vector denoting the auxiliary data meeting condition 2. $\mathbf{1}$ is a vector whose elements are all ones. The last term of the objective function in Eq. (4) is the regularization term of $\mathbf{w}$, which ensures that the weight vector is sparse.

[1]The proposed method will be given in Section 3.2. The parameters of transformations and number of neighbors will be given in Section 4.

The feature $\mathbf{F}_i$ corresponds to a kernel function of $\mathbf{x}_i$, and we choose radial basis functions (RBFs) as features. To each image $\mathbf{x}$ we can sample it with overlap and obtain a matrix $\mathbf{P}_i \in R^{D\times s^2}$. The t-th row of $\mathbf{P}_i$, denoted as $\mathbf{p}_t$, corresponds to a $s \times s$ patch of $\mathbf{x}$. Cluster patches of all the images into f clusters, whose centers are $\{\boldsymbol{\mu}_n\}_{n=1}^{f}$. The element of $\mathbf{F}_i$ is $F_{tn} = e^{-\frac{\|\mathbf{p}_t-\boldsymbol{\mu}_n\|_2^2}{2h^2}}$.

3.3. Weighted Iterative Manifold Learning

Because of the coping of the three variables, we seek the optimal solution of Eq. (4) by an iterative method. According to the evaluation about various manifold learning algorithms shown in [20], Local-Tangent-Space-Alignment (LTSA) algorithm [24] performs the best. So, we estimate $\mathbf{y}_i$ initially by LTSA, and set $\mathbf{w} = \mathbf{1}$, then the objective function becomes

$$\sum_{i=1}^{N+L}\sum_{j\in\mathcal{N}_i}\hat{a}_{ij}\|\mathbf{x}_j - \mathbf{x}_i - \mathbf{F}_i\boldsymbol{\Theta}\boldsymbol{\epsilon}_{ij}\|_2^2. \tag{5}$$

For the convenience of representation, we rewrite Eq. (5) as following matrix form,

$$\sum_{i=1}^{N+L}\|(\boldsymbol{\Delta}_i - \mathbf{F}_i\boldsymbol{\Theta}\mathbf{E}_i)\mathbf{A}_i\|_F^2. \tag{6}$$

Here $\boldsymbol{\Delta}_i \in R^{D\times|\mathcal{N}_i|}$, each column of which is $\mathbf{x}_j - \mathbf{x}_i$, $j \in \mathcal{N}_i$. Similarly, $\mathbf{E}_i \in R^{d\times|\mathcal{N}_i|}$, whose column is $\boldsymbol{\epsilon}_{ij}$, $j \in \mathcal{N}_i$. $\mathbf{A}_i$ is a diagonal matrix whose diagonal element is $\sqrt{\hat{a}_{ij}}$.

The minimization of Eq. (6) has an analytic solution. Denote $\mathbf{S}_i = \mathbf{E}_i\mathbf{A}_i$, we can get a series of matrices $\{\mathbf{Z}_{ij}\}$ as

$$\mathbf{Z}_{ij} = \mathbf{S}_i^{(j)} \otimes \mathbf{F}_i. \tag{7}$$

Here "$\otimes$" means Kronecker multiplication. $\mathbf{S}_i^{(j)}$ is the jth column of $\mathbf{S}_i$. The optimal $\boldsymbol{\Theta}$ can be calculated as follows.

$$\mathbf{Z} = \sum_{i=1}^{N+L}\sum_{j=1}^{|\mathcal{N}_i|}\mathbf{Z}_{ij}^T\mathbf{Z}_{ij}, \tag{8}$$

$$vec(\boldsymbol{\Theta}) = \mathbf{Z}^{+}\sum_{i=1}^{N+L}\sum_{j=1}^{|\mathcal{N}_i|}\hat{a}_{ij}\mathbf{Z}_{ij}^T\boldsymbol{\Delta}_i^{(j)},$$

where $\boldsymbol{\Delta}_i^{(j)}$ is the jth column of $\boldsymbol{\Delta}_i$ and $vec(\boldsymbol{\Theta})$ is the vector form of $\boldsymbol{\Theta}$.

After getting optimal $\boldsymbol{\Theta}$, we then compute the optimal $\mathbf{w}$ by solving following integer programming problem.

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}}(\mathbf{e} - \lambda\mathbf{1})^T\mathbf{w}, \tag{9}$$
$$s.t. \quad w_i = 0 \text{ or } 1, \quad i = N+1, ..., N+L.$$

Here, $\mathbf{e}$ is the vector of errors, whose element is $\|(\boldsymbol{\Delta}_i - \mathbf{F}_i\boldsymbol{\Theta}\mathbf{E}_i)\mathbf{A}_i\|_F^2$. Furthermore, because the optimal $\mathbf{w}$ needs to be quantized to 0 or 1 for sample selection, Eq. (9) has a fast thresholding method to solve. Given $\mathbf{e}$, if its ith element is larger than λ, then $w_i = 0$, else $w_i = 1$. Based on the newly selected samples, we go back to calculate their neighbors and then repeat the steps above. After several iterations, the final optimal solution is achieved. The scheme of the proposed algorithm is given in Algorithm 2.

Algorithm 2: Manifold Learning Algorithm
Initialization:
Given samples $\{\mathbf{x}_i\}_{i=1}^{N+L}$,
apply Algorithm 1 to get neighbors of $\mathbf{x}_i$, $\mathcal{N}_i$.
Calculate $\{\hat{a}_{ij}\}$ by Eq. (3).
Iteration:
For $k = 1$ to K
1.Apply LTSA to get coordinates $\{\mathbf{y}_i\}$.
2.Minimize Eq. (6) to get $\boldsymbol{\Theta}$.
3.To transformed result $\mathbf{x}_j$, $j = N+1, ..., N+L$,
if $\|(\boldsymbol{\Delta}_j - \mathbf{F}_j\boldsymbol{\Theta}\mathbf{E}_j)\mathbf{A}_j\|_F^2 \geq \lambda$,
$\{\mathbf{x}_i\} = \{\mathbf{x}_i\} \setminus \mathbf{x}_j$.
4.Remove corresponding neighbors of $\mathbf{x}_i$.
5.Calculate $\{\hat{a}_{ij}\}$ again.

3.4. Weighted Data Synthesis

When we synthesize new data, we can also introduce weights to its neighbors for increasing estimation accuracy. Because the new data is not available, the weight of each neighbors is measured by $\boldsymbol{\epsilon}$. Given the coordinate $\mathbf{y}_{new}$ and the set of its neighbors, $\mathcal{N}_{new}$, the normalized weight is

$$b_{j,new} = e^{-\frac{D^2\|\boldsymbol{\epsilon}_{j,new}\|_2^2}{2d^2\sigma^2}}, \quad \hat{b}_{j,new} = \frac{b_{j,new}}{\sum_{i\in\mathcal{N}_{new}} b_{i,new}}. \tag{10}$$

So, the new data $\mathbf{x}_{new}$ is estimated by

$$\mathbf{x}_{new} = \sum_{j\in N_{new}}\hat{b}_{j,new}(\mathbf{x}_j + \mathbf{F}_j\boldsymbol{\epsilon}_{j,new}). \tag{11}$$

4. Experimental Results

To demonstrate the feasibility, we apply the proposed method on the synthesis of face images[2] [7]. The data set contains 15 image sequences of head rotation from 15 people. The sampling rate of horizontal rotation of head is 1 sample per 15°, and the sampling range is $[-90°, 90°]$. In such a situation, only 13 images can be used for synthesis. We resize images to 100×90 and create auxiliary data by following three steps. 1) To each original image, we apply shifting 5 times to create 5 new images. The shifting step we choose is 2 pixels per image. 2) To each shifting result,

[2]http://www-prima.inrialpes.fr/perso/Gourier/Faces/HPDatabase.html

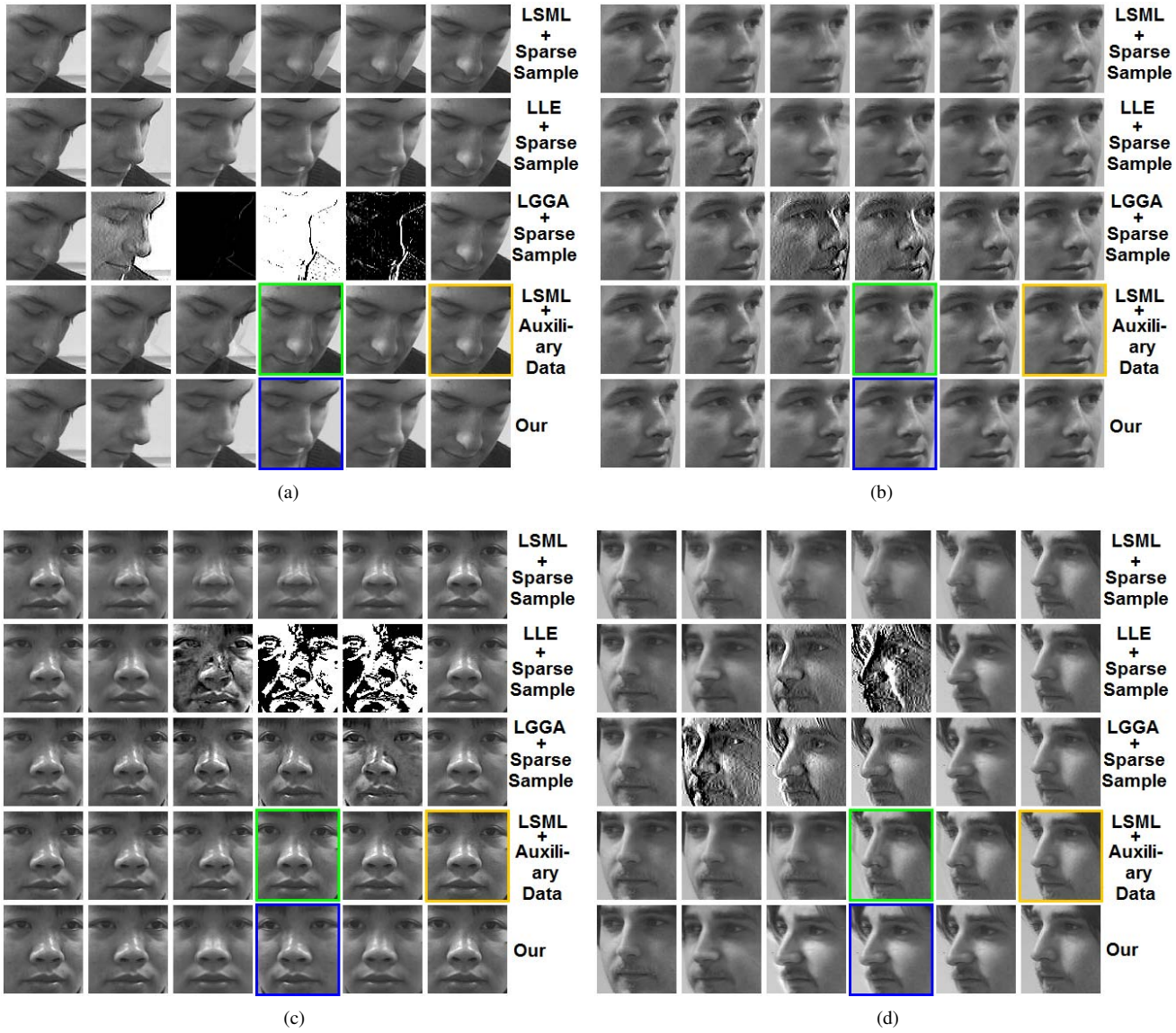

Figure 4. The comparison of synthesis results for various methods. To each sub-figure, the 1st and the 6th column are original images. From the 2nd to the 5th column are synthesis results gotten by LSML [5] with sparse samples, LLE based method [21], LGGA [9], LSML with dense samples and proposed method.

we then apply rotation 5 times to create 5 images more, and the rotation angle we choose is 2 degrees per image. 3) Finally, all the images are flipped.

The reason for choosing shifting transform is based on the work in [9] and the experimental result in section 2. According to [9], the manifold of the image shifting sequence is a curve. Furthermore, we can find in Fig. 3 that a part of the curve can be viewed as a good local fitting for the manifold of rotated head image. In our opinion, this is because that the rotation of head can be approximated by many small shifting steps. Introducing rotation transform makes the image robust to subtle changes of face. Applying flipping transform is based on the symmetry of human face.

In feature extraction phase, the size of patch is 15×15, and the number of clustering center is 50. The parameter h is chosen as one third of the maximum $\|\mathbf{p}_t - \boldsymbol{\mu}_n\|_2$. Similarly, another parameter of the proposed algorithm, σ in Eq. (3), is one third of the maximum $\|\mathbf{x}_i - \mathbf{x}_j\|_2$ as well. The Lagrange factor in Eq. (4), λ, is set to be 1. In the learning phase, the number of neighbor is 16 initially in algorithm 1.

After learning manifold, the location of the new image on the manifold is decided by its coordinates in the tangent space. With the help of introducing transformed samples, there are sufficient neighbors for the new image so that it can be synthesized by Eq. (11). In our work, we choose $|\mathcal{N}_{new}| = 2$ in Eq. (11).

Given original sparse samples and corresponding auxiliary data, we compare the proposed method with other competitors, including the LLE based method [21], LGGA [9] and the state of the art method, LSML [5]. Specifically, for showing the influence of auxiliary data on the result of synthesis, LSML are applied in two cases — purely using original sparse samples and combining sparse samples with auxiliary data. Fig. 4 gives the partial experimental results on different face images. To LLE based method and LGGA, although they can get good synthetic images sometimes, the performances of them are not stable. In Fig. 4, both of them have a risk of getting failures during synthesis.

On the other hand, when samples are sparse, the number of sample is too small to avoid over-fitting phenomenon of parameters in the learning phase. As a result, LSML leads to obvious "ghost effect" — the synthesis result is similar to that of traditional linear interpolation. Even if applying LSML with the help of auxiliary data, without adaptive strategies for sample and neighbor selection, the performance are also inferior to the proposed method. In Fig. 4, we can find that some LSML results in the situation having auxiliary data still have "ghost effect". The reason for this problem is that the outliers in the auxiliary data are not removed in the learning phase. So, in the synthesis phase, it is possible that the neighbors we find include outliers. In such a situation, synthesis results will be corrupted by outliers.

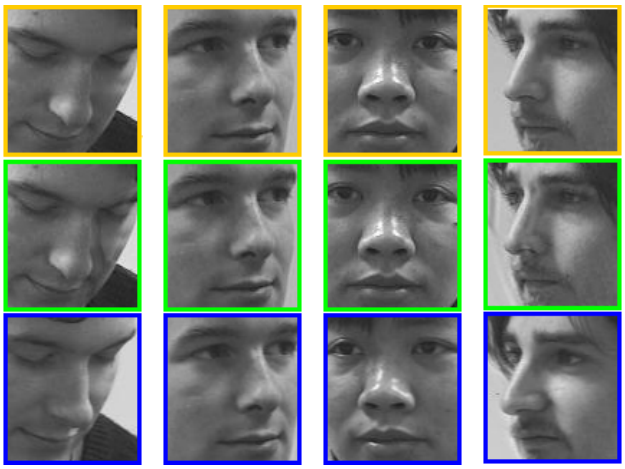

Figure 5. Each sub-figure shows the enlarged images labeled in Fig. 4. To each sub-figure, the image labeled by orange frame is original image. The image labeled by green frame is the synthesis result of LSML. The image labeled by blue frame is the result of the proposed method.

Another problem may happen in the result of LSML with auxiliary data is "missing rotation". In each sub-figure of Fig. 4, the synthesis results of LSML looks like the repeat of original image. Fig. 5 gives enlarged comparison images for illustrating this phenomenon. In our opinion, this problem is also caused by outliers. Because the outliers disobeying condition 2 are not removed, the synthetic manifolds are isolated to each other. As a result, the global structure of the target manifold cannot be learned by LSML. The illustrations of "ghost effect" and "missing rotation" are shown in Fig. 6. The proposed method, on the other hand, makes samples sufficient by transformations and removes outliers during the iterations of learning algorithm. In Fig. 4 and 5, the synthesis results of the proposed method avoid serious "ghost effect". At the same time, the subtle change of image is learned by the proposed method while the synthesis result of LSML is almost the same with original image.

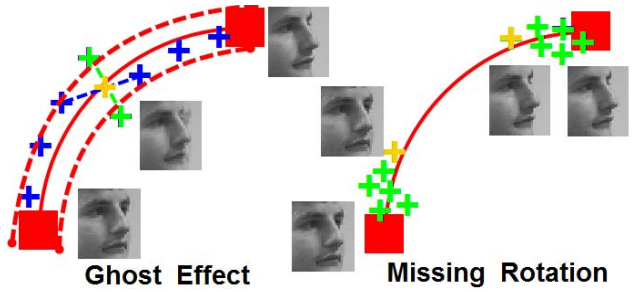

Figure 6. To each sub-figure, the red squares and corresponding images are original samples and red curve shows the target manifold. The blue and green "+"s are transformed samples. The orange "+"s and corresponding images are synthetic results based on wrong samples (green "+"s). The "ghost effect" is caused by choosing the outliers disobeying condition 1 as samples. The "missing rotation" is cause by the failure of learning. The outliers disobeying condition 2 are not removed, which causes that structure of target manifold is not captured.

We now discuss a limitation of the proposed method — sometimes the samples cannot be created or borrowed correctly. In Fig. 4(a), we can find that the original images do not have hairs in the forehead while the synthetic images have hairs. This is because the samples are created by flipping are not on the original manifold perfectly. However, even though the proposed method is not perfect, it has better visual effect than its competitors. Besides the comparison of visual effect, we give objective measurements for various methods in Table 1. The objective experiment is designed as follows. After learning the manifold and getting the coordinates of all the images in the latent space, we remove one original image and auxiliary data created from it. Based on the rest of images and the coordinates of removed images, we try to synthesize the removed image. The average mean-square-error (MSE) of the synthesis results of 15 people are measured for various methods. According to Table 1, the performances of LLE based method and LGGA is not satisfying because of the failed synthesis results like Fig. 4 shows. The performance of the proposed method is superior to others indeed.

Table 1. Average MSE of Synthesis Results for Various Methods

LSML [5] (Sparse Samples)	481.69
LLE Based [21] (Auxiliary Data)	660.43
LGGA [9] (Auxiliary Data)	577.33
LSML [5] (Auxiliary Data)	258.46
Proposed	**175.58**

5. Conclusion and Future Work

In this paper, a manifold-based face synthesis method is proposed for the case of sparse samples. By combining transfer learning strategy with manifold learning algorithm, the samples are supplied by their transformed results, which provide additional structural information for manifold recovery. Additionally, the auxiliary data is weighted for outlier detection during the learning phase, which improves learning and final synthesis results. To the data set having certain special properties that can be used to design transformations, the proposed method has potential to improve the learning result when the number of samples is insufficient. The core problem of our work is to find suitable transformations for given samples, which is empirical currently. Designing a strategy for finding suitable transformations is the direction for our future work.

Acknowledgement: This work is supported in part by NSF grant DMS-1317424 and a subcontract N00014-13-M-0022 from Object Video.

References

[1] A. M. Álvarez-Meza, J. Valencia-Aguirre, G. Daza-Santacoloma, C. D. Acosta-Medina, and G. Castellanos-Domínguez. Image synthesis based on manifold learning. In *Computer Analysis of Images and Patterns*, pages 405–412. Springer, 2011. 1

[2] C. Bregler and S. M. Omohundro. Nonlinear image interpolation using manifold learning. *Advances in neural information processing systems*, pages 973–980, 1995. 1

[3] M. A. Carreira-Perpinán and Z. Lu. Manifold learning and missing data recovery through unsupervised regression. In *ICDM*, pages 1014–1019. IEEE, 2011. 1

[4] J. Chen, D. Yi, J. Yang, G. Zhao, S. Z. Li, and M. Pietikainen. Learning mappings for face synthesis from near infrared to visual light images. In *CVPR*, pages 156–163. IEEE, 2009. 1, 2

[5] P. Dollár, V. Rabaud, and S. Belongie. Learning to traverse image manifolds. *Advances in neural information processing systems*, 19:361, 2007. 1, 2, 6, 7, 8

[6] E. Elhamifar and R. Vidal. Sparse manifold clustering and embedding. *Advances in Neural Information Processing Systems*, 24:55–63, 2011. 4

[7] N. Gourier, D. Hall, and J. L. Crowley. Estimating face orientation from robust detection of salient facial structures. In *FG Net Workshop on Visual Observation of Deictic Gestures*, pages 1–9, 2004. 5

[8] J. Hinkle, P. Muralidharan, P. T. Fletcher, and S. Joshi. Polynomial regression on riemannian manifolds. In *ECCV*, pages 1–14. Springer, 2012. 3

[9] D. Huang, Z. Yi, and X. Pu. Manifold-based learning and synthesis. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(3):592–606, 2009. 1, 6, 7, 8

[10] C.-S. Lee and A. Elgammal. Modeling view and posture manifolds for tracking. In *ICCV*, pages 1–8. IEEE, 2007. 1

[11] R. Li, T.-P. Tian, and S. Sclaroff. Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series. In *ICCV*, pages 1–8. IEEE, 2007. 1

[12] J. J. Lim, R. Salakhutdinov, and A. Torralba. Transfer learning by borrowing examples for multiclass object detection. In *NIPS*, 2011. 2, 3

[13] C.-B. Liu, R.-S. Lin, N. Ahuja, and M. Yang. Dynamic textures synthesis as nonlinear manifold learning and traversing. In *BMVC*, volume 2, pages 859–868. Citeseer, 2006. 1

[14] K. Moon and V. Pavlovic. Regression using gaussian process manifold kernel dimensionality reduction. In *MLSP*, pages 14–19. IEEE, 2008. 1

[15] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010. 2

[16] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009. 2

[17] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, pages 1481–1488. IEEE, 2011. 2

[18] M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *The Journal of Machine Learning Research*, 10:1633–1685, 2009. 2

[19] M. Torki and A. Elgammal. Regression from local features for viewpoint and pose estimation. In *ICCV*, pages 2603–2610. IEEE, 2011. 1

[20] T. Wittman. Manifold learning matlab demo. *URL: http://www. math. umn. edu/ wittman/mani/index. html*, 2005. 5

[21] C. Zhang, J. Wang, N. Zhao, and D. Zhang. Reconstruction and analysis of multi-pose face images based on nonlinear dimensionality reduction. *Pattern Recognition*, 37(2):325–336, 2004. 1, 6, 7, 8

[22] L. Zhang, S. Wang, and D. Samaras. Face synthesis and recognition from a single image under arbitrary unknown lighting using a spherical harmonic basis morphable model. In *CVPR*, volume 2, pages 209–216. IEEE, 2005. 1

[23] Z. Zhang, J. Wang, and H. Zha. Adaptive manifold learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):253–265, 2012. 4

[24] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing*, 26(1):313–338, 2004. 5