3D Face Tracking and Expression Inference from a 2D Sequence Using Manifold Learning

Wei-Kai Liao and Gerard Medioni
Computer Science Department
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA 90089-0273
{wliao, medioni}@usc.edu

Abstract

We propose a person-dependent, manifold-based approach for modeling and tracking rigid and nonrigid 3D facial deformations from a monocular video sequence. The rigid and nonrigid motions are analyzed simultaneously in 3D, by automatically fitting and tracking a set of landmarks. We do not represent all nonrigid facial deformations as a simple complex manifold, but instead decompose them on a basis of eight 1D manifolds. Each 1D manifold is learned offline from sequences of labeled expressions, such as smile, surprise, etc. Any expression is then a linear combination of values along these 8 axes, with coefficient representing the level of activation. We experimentally verify that expressions can indeed be represented this way, and that individual manifolds are indeed 1D. The manifold dimensionality estimation, manifold learning, and manifold traversal operation are all implemented in the N-D Tensor Voting framework. Using simple local operations, this framework gives an estimate of the tangent and normal spaces at every sample, and provides excellent robustness to noise and outliers. The output of our system, besides the tracked landmarks in 3D, is a labeled annotation of the expression. We demonstrate results on a number of challenging sequences.

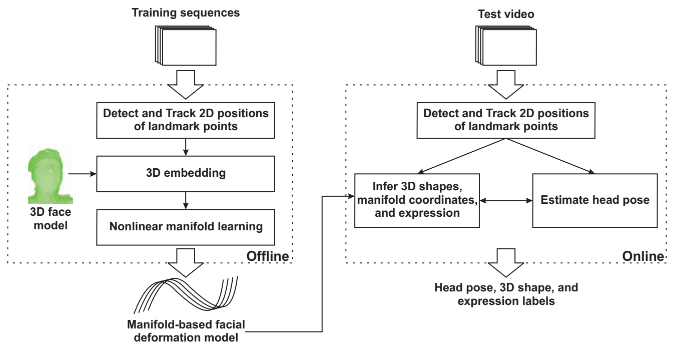
1. Introduction

Nonrigid deformation is an important property of human faces, as it conveys information about a human's mental state. A lot of research has been devoted to investigate deformable face models based on linear subspace analysis. The 2D Active Shape Model (ASM) and Active Appearance Model (AAM) [5, 6, 15] approximate the shape deformation as a linear combination with some 2D basis shapes. The model is learned using Principal Component Analysis (PCA). The AAM inherits the idea of deformable shape, but also learns an appearance model for texture variation. A 3D deformable model has also been proposed. In [22], Xiao et al extended the 2D AAM to a combined 2D+3D AAM. In [2, 3], Blanz and Vetter built a 3D morphable model for facial animation and face recognition.

More recently, in [11], Gu and Kanade proposed a 3D deformable model consisting of a set of sparse 3D points and patches associated with each point. Based on this model, an EM style algorithm is proposed to infer head pose and face shapes. In [23], Zhu and Ji proposed a normalized SVD to estimate the pose and expression. Based on this, a non-linear optimization method is also proposed to improve the tracking result. Vogler et al [21] proposed an integration system to combine 3D deformable model with 2D ASM. The proposed system uses ASM to track reliable features and 3D deformable model to infer the face shape and pose from tracked features.

In the above papers, the construction of a deformable model is built on top of the linear subspace approach. However, linear subspace methods are inadequate to represent the underlying structure of real data, and nonlinear manifold learning approaches are proposed [19, 20]. Nonlinear dimensionality reduction techniques provide a good alternative to model the high dimensional visual data. In [4], Chang et al proposed a probabilistic approach based on the appearance manifold for expression analysis. In [9, 10], the author proposed a manifold based approach for 3D body pose tracking and general tracking.

Most nonlinear manifold learning techniques characterize the intrinsic structure by recovering the low-dimensional embedding. For example, ISOMAP [20] finds a low-dimensional embedding that preserves geodesic distances in the input space. Locally-linear embedding (LLE) [19] searches for a manifold based on the local linearity principle. Recent works argue that this may not be the best way to parameterize the manifold, especially for the pur-


Training sequences
Detect and Track 2D positions of landmark points
3D embedding
3D face model
Nonlinear manifold learning
Offline
Manifold-based facial deformation model
Test video
Detect and Track 2D positions of landmark points
Infer 3D shapes, manifold coordinates, and expression
Estimate head pose
Online
Head pose, 3D shape, and expression labels
Figure 1. Flow chart of proposed system

pose of handling noisy data and out-of-sample generalization [1, 7, 8]. They propose different algorithms to estimate the local tangent hyperplane on the manifold, and use the estimated tangent to manipulate novel points. Besides, [18] proposed computational tools for statistical inference on a Riemannian manifold.

Here, we propose a new framework to model the deformable shape using nonlinear manifolds. The main contribution is two-fold. First, instead of using a linear subspace analysis, we argue the 3D facial deformations are better modeled as a combination of several 1D manifolds. Each 1D manifold represents a mode of deformation or expression, such as smile, surprise, blinking, etc. By learning these manifolds, a 3D shape instance, usually represented by a very high dimensional vector, can be mapped into a low-dimensional manifold. The coordinate on the manifold corresponds to the magnitude of facial deformation along that mode. We thus call it the "level of activation". Second, we propose a novel framework of nonlinear manifold learning based on N-D Tensor Voting [16, 17]. Tensor Voting estimates the local normal and tangent spaces of the manifold at each point. The estimated tangent vectors enable us to directly navigate on the manifold.

The proposed 3D deformable shape model is applied to nonrigid face tracking. We develop an algorithm to infer the nonrigid 3D facial deformations with the head pose and expression iteratively, based on the proposed model. Without learning complex facial gestures dynamics, the proposed algorithm can track a rich representation of the face, including the 3D pose, 3D shape, expression label with probability, and the activation level. The flow chart of our proposed system is outlined in figure 1.

The rest of this paper is organized as follows: We start with the offline construction of the manifold-based facial deformation model. The formulation and learned manifolds are presented in section 2. The manifold learning and inference is implemented in the N-D Tensor Voting framework, shown on section 3. Based on the proposed model and inference tool, we develop an iterative algorithm to track the nonrigid facial deformation with 3D head pose, as detailed in section 4. In section 5, we conduct several experiments to analyze and evaluate the proposed model and algorithm. Finally, conclusions and discussions are given in section 6.

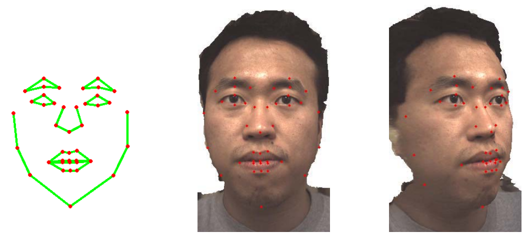

(a) 2D (b) 3D Frontal (c) Non-frontal

Figure 2. Shape model. The red points indicate the landmarks

2. Manifolds of 3D Facial Deformations

Using a deformable model allows to find a compact parametrization to represent the nonrigid facial shape and motion. Linear subspace techniques approach this problem by searching for a set of optimal basis, whose span covers varying shapes. For a given shape instance, the projection into this subspace is an approximation:

$$S = S_0 + \Phi b + \epsilon \qquad (1)$$

where S_0 is the mean shape, ϵ is the noise term, and Φ is the matrix of basis, which is usually learned by applying Principal Component Analysis (PCA) on the training data. b is the parameter that represents the weight of each principal component and controls the variation of the shape instance.

On the other hand, the 3D shape itself is conceptually aligned on the manifolds. The 3D facial shape is governed by two factors, the type of expression and the magnitude of the deformation. The type of expression indicates the style of deformation, such as smile, surprise, or blinking, and warps the neutral face based on this deformation. The magnitude of the deformation controls the "level of activation", such as onset, apex, or offset, for this shape instance. The mathematical formulation of the above ideas is:

$$S = S_0 + f(L, c) + \epsilon \qquad (2)$$

where S_0 now is the neutral face. c is the variable indicates the type of deformation, and L is the variable controling the magnitude of this deformation. f is a nonlinear function that maps L and c into the 3D shape space. $X = S - S_0$ is the 3D facial deformation. If S_0 is fixed and known, modeling the facial deformation, X, and modeling the 3D shape, S, are equivalent.

Under this setting, for a specific expression c, the deformation X should lie on a 1D manifold, since L is conceptually a real-valued scalar that controls the magnitude of this expression. Because the face can mix several different expressions, we have a set of 1D manifolds and any facial deformation is a combination of this set of manifolds.

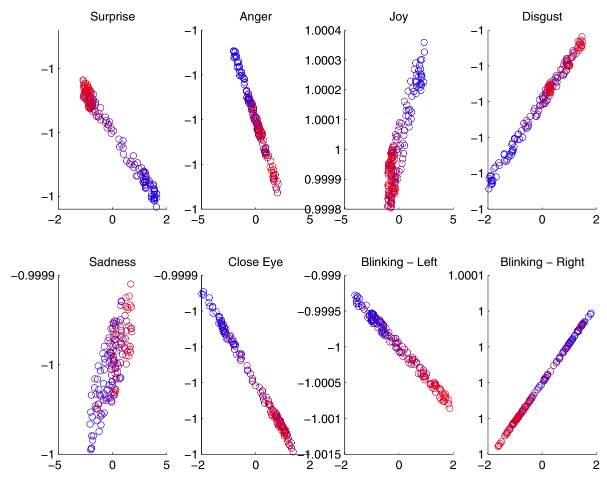


Figure 3. Learned manifolds of 3D facial deformations

	% classified as 1D
Surprise	90.1%
Anger	85.3%
Joy	93.4%
Disgust	95.3%
Sadness	90.1%
Close and open eyes	88.7%
Left-eye blinking	92.4%
Right-eye blinking	93.9%

Table 1. Estimated dimensionality of 3D facial deformations using Tensor Voting

To validate this deformation manifold concept, we conduct an off-line learning process as follows: A face shape is represented by 42 landmark points. For a collected expression video, we use a 2D active shape model to detect landmark points. These points are tracked across frames using the Lucas-Kanade feature tracker [14]. After tracking, we have the 2D image coordinates for each landmark point and they represent the 2D shape variation. We project these landmark points into 3D by using a previously generated 3D face model. Currently, we use a person specific face model, but it might be sufficient to use either a generic one, or even a cylinder. Figure 2 illustrates the shape model. The deformation vector X is the stack of 3D point displacement and it is the input to the manifold learning algorithm.

The dimensionality of facial deformation manifolds are examined. Figure 3 plots the learned manifolds of 8 common facial deformations, and table 1 reports the estimated dimensionality using Tensor Voting. The 8 modes of deformations are surprise, anger, joy, sadness, disgust, close and open eyes, left eye blinking, and right eye blinking, and each learned manifold is projected into a 2D space. It is verified experimentally that the dimensionality of the manifold is very close to 1, which confirms our assumption.

To visualize the meaning of the parameter L, figure 3 plots each sample in a different color based on its order in the original sequences. We record videos with the subject going from neutral to the apex of the expression and back to neutral. The samples in the beginning and end are plotted in blue and the samples in the middle of the video are plotted in red, which is assumed to be the peak of this expression.

3. Tensor Voting and Nonlinear Manifold Learning

Tensor Voting is a computational framework to estimate geometric information. It was originally developed in 2D for perceptual grouping and figure completion, and later to 3D and ND for other problems, such as stereo matching and motion processing. Since the focus of this paper is not the Tensor Voting framework, we only introduce the fundamental concepts, particularly in the context of learning conceptual manifold, and refer readers to [16, 17] for the complete presentation and implementation details.

Suppose we have a set of samples in a high dimensional space V and these samples lie on a manifold M of much lower dimension. Our objective is to infer the geometric structure of this manifold. In other words, we try to find the vectors that span the normal and tangent space at each point, and use them to characterize this manifold.

Tensor voting is an unsupervised approach to estimate a structure tensor T at each point. Here, T is a rank-2, symmetric tensor, whose quadratic form is a symmetric, non-negative definite matrix, representing underlying geometry. Given training samples, $\{X_i\}$, Tensor Voting encodes each sample as a ball tensor, which indicates an unoriented token. Each X_i receives a vote from X_j, $T_{j\to i}$, and X_i sums up all incoming tensors. $T_{j\to i}$ is generated by taking X_j's tensor and relative orientation between X_i and X_j into account, and weight the distance between them. The result of this process can be interpreted as a local, nonparametric estimation of the geometric structure at each sample position.

After accumulating all cast tensors, the local geometry can be derived by examining its eigensystem. Recall that a tensor can be decomposed as

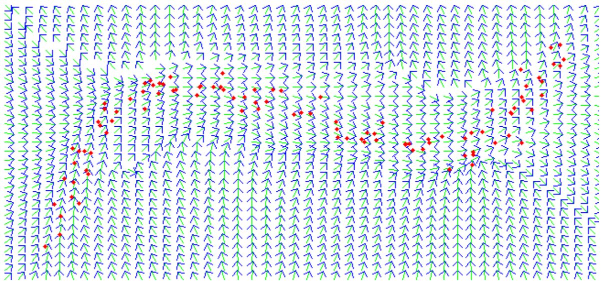
$$T=\sum_{i=1}^{N}\lambda_i e_i e_i^T=\sum_{i=1}^{N-1}[(\lambda_i-\lambda_{i+1})\sum_{k=1}^{i} e_k e_k^T]+\lambda_N\sum_{i=1}^{N} e_i e_i^T \quad (3)$$

where $\{\lambda_i\}$ are the eigenvalues arranged in descending order, $\{e_i\}$ are the corresponding eigenvectors, and N is the dimensionality of the input space.

Equation 3 provides a way to interpret the local geometry from T. The difference between two consecutive eigenvalues, $\lambda_i - \lambda_{i+1}$, encodes the salience of certain structure:

$$\lambda_i - \lambda_{i+1} \gg \lambda_j - \lambda_{j+1}, \quad \forall j \in \{1, \cdots, N-1\}, j \neq i$$

means the geometric structure, whose normal space is i-D and the tangent space is $(N-i)$-D, is most salient. The eigenvectors $\{e_1, ..., e_i\}$ span the normal space of this point,


(a) Tangent Directions: e_2

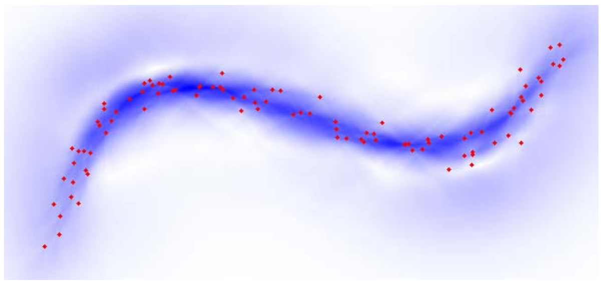

(b) Saliency Values: $\lambda_1 - \lambda_2$

Figure 4. Example of Tensor Voting result in 2D

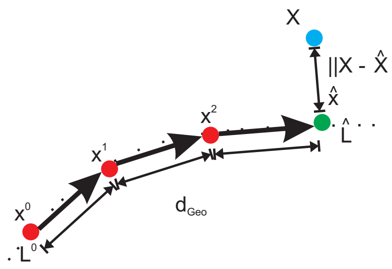


Figure 5. Traversing the manifold with tangent vectors

while $\{e_{i+1}, ..., e_N\}$ represent the tangent space. For example, if $\lambda_{N-1} - \lambda_N$ is the most salient structure, this point is considered in an 1D manifold and e_N is its tangent vector.

Figure 4 shows an example of applying Tensor Voting in 2D. The red dots represent the training samples, which come from a 1D curve. We perturb each point with noise, thus they are not perfectly aligned in the manifold. After Tensor Voting, we have the tangent direction and saliency value of each position. In figure 4(a), arrows indicate direction of tangent vector and it is clear that for position near the conceptual manifold, the estimated tangent direction is accurate. On the other hand, figure 4(b) is the saliency map of $\lambda_1 - \lambda_2$: If it is more salient, its color is close to blue, otherwise it is close to white. We can observe that positions close to the curve have higher saliency. Combining these two figures, we can easily identify those points that are more likely to be on the manifold, and also estimate their tangent vectors. The estimated tangent and saliency value are used to traverse the manifold, as shown in the next section.

3.1. Traversing the Manifold with Tangent Vectors

The estimated tangent plane enables us to directly navigate on the manifold. The key idea is to "walk down the manifold" along the estimated tangent hyperplane. Figure 5 illustrates the idea of traversing a manifold with tangent hyperplane. To be more specific, we present 4 tasks related to the 3D facial deformation modeling. The first two tasks are about learning the manifold from training data, and the last two are inference tasks after we have learned the manifold.

Training Task 1. Estimate $d_{\text{Geo}}(X_i, X_j)$, the geodesic distance between X_i and X_j.

$d_{\text{Geo}}(X_i, X_j)$ can be estimated as the minimum traveling distance between X_i and X_j along the manifold. Let X_i be starting point and destination is X_j, and set $X^0 = X_i$. In each X^k, we move toward the destination under the guidance of moving direction D^k:

$$D^k = H^k H^{kT}(X_j - X^k) \tag{4}$$

where H^k is the matrix of tangent vectors estimated by Tensor Voting. We then move iteratively to estimate the geodesic distance:

$$\begin{aligned} X^{k+1} &= X^k + \alpha^k D^k \\ d_{\text{Geo}} &= d_{\text{Geo}} + \|\alpha^k D^k\| \end{aligned} \tag{5}$$

until X^k reaches an ϵ-neighborhood of X_j.

The step length α is chosen to ensure we are still within the manifold. In other works [1, 7, 8], the authors have to use a very small step to avoid leaving the support of manifold. In contrast, Tensor Voting provides a mechanism to prevent this undesirable situation, as saliency indicates the underlying structure. If we reach a point outside the manifold, the saliency value is low and we change to a smaller step. This is clearly an advantage over other approaches.

Training Task 2. Recover the manifold coordinate L of X.

Given 2 samples X_i and X_j, the geodesic distance $d_{\text{Geo}}(X_i, X_j)$ is assumed to be the same as $\|L_i - L_j\|_{\text{manifold}}$, the distance in the embedded manifold. Therefore, if L_i is known, we can recover L_j as the byproduct of estimating the geodesic distance.

To recover the manifold coordinate of a given training set $\{X_i\}$, we first identify a sample as the starting point. In practice, we include a vector $\mathbf{0}$ in the training data. Its L is defined as 0, since it means zero deformation and represents the neutral shape. Starting from this point, we recursively move outward to estimate the geodesic distance and manifold coordinate.

Inference Task 1. Given L, find its mapping $\hat{X}$ in input space.

This is a nonlinear function approximation problem, as the objective is to learn the function f_{TV}:

$$\hat{X} = f_{\text{TV}}(L) \tag{6}$$

However, we can solve it as a search problem, using the estimated tangent hyperplane: First, we find point L^0, whose coordinate is X^0 in input space. In practice, L^0 is the nearest point of L in training set. We iteratively move X^k and L^k following the estimated tangent vectors, until we reach L. The procedure is outlined in algorithm 1.

```
input  : $L$
output: $\hat{X}$
Initialize $L^0$ and $X^0$;
while $\|L - L^k\| \geq \epsilon$ do
    $H^k \leftarrow$ Tensor Voting at $X^k$;
    $D^k \leftarrow H^k(L - L^k)$;
    $X^{k+1} \leftarrow X^k + \alpha^k D^k$;
    $L^{k+1} \leftarrow L^k + \alpha^k (L - L^k)$;
end
$\hat{X} \leftarrow X^k$;
```

Algorithm 1: Map L to $\hat{X}$

Inference Task 2. Given an arbitrary X in input space, find its optimal projection $\hat{X}$ in the manifold.

In 3D facial deformation modeling, this is the most important interpretation task, since we want to recover the "true" position from noisy observation using deformation model. Given an observation X, it is represented as $X = f(L) + \epsilon$, where ϵ is the noise. Let $\hat{X} = f(L)$ be the optimal projection of X on the manifold. This "projection" operation can be called filtering, since it discards the noise.

If we further assume the equal prior of L, $\hat{X}$ will be:

$$\begin{aligned} \hat{L} &= \arg\min_L \|X - f_{\text{TV}}(L)\|_{\text{input}} \\ \hat{X} &= f_{\text{TV}}(\hat{L}) \end{aligned} \tag{7}$$

Therefore, $\hat{X}$ is the point nearest to X on the manifold. In practice, searching $\hat{X}$ is formulated as an optimization problem, and solved by algorithm 2. We choose to initialize X^0 as the nearest point to X in the training samples. The convergence criteria is either $\|X - X^k\| < \epsilon$ or $\|X - X^k\|$ reaches a local minimum.

```
input  : $X$
output: $\hat{X}$
Initialize $X^0$;
repeat
    $H^k \leftarrow$ Tensor Voting at $X^k$;
    $D^k \leftarrow H^k H^{kT}(X - X^k)$;
    $X^{k+1} \leftarrow X^k + \alpha^k D^k$;
until convergence ;
$\hat{X} \leftarrow X^k$;
```

Algorithm 2: Search for optimal projection of X

3.2. Working with Multiple Manifolds

The recovered manifold coordinate and projection can be used to determine the likelihood of the input point belonging to this manifold. Since there may exist several manifolds in input space, and these manifolds may have the same dimensionality, as we have shown that there are multiple 1D deformation manifolds of a human face, we would like to measure such likelihood, and infer the posterior probability of each manifold.

Given a point X and recovered low dimensional embedding L for manifold c, we define the joint probability of X and L using mixture modeling:

$$\begin{aligned} P(X, L|c) &= \sum_m w_m P_m(X|L, c) P_m(L|c) \\ &= \sum_m w_m^c P_G(X|f_{\text{TV}}(L; c), \Sigma_m^c) P_G(L|\mu_m^c, \sigma_m^c) \end{aligned} \tag{8}$$

where $f_{\text{TV}}(.; c)$ is the mapping using manifold c. $P_G(X|f_{\text{TV}}(L; c), \Sigma_m^c)$ is the Gaussian pdf with mean $f_{\text{TV}}(L; c)$ and covariance matrix Σ_m^c. For simplicity, we assume Σ_m^c is diagonal. $P_G(L|\mu_m^c, \sigma_m^c)$ is the Gaussian pdf with mean μ_m^c and standard deviation σ_m^c.

Equation 8 relates the relative position between X and its optimal projection $\hat{X}$ with its manifold coordinate L. Based on this equation, the posterior probability of expression c can be computed as:

$$P(c|X, L) \propto P(X, L|c) P(c) \tag{9}$$

4. Tracking the Nonrigid Facial Deformation with Head Pose

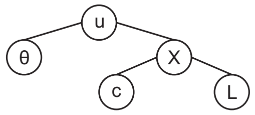


Figure 6. The generative model of 2D shape

The proposed deformation manifold method can be used to track the 3D head pose with nonrigid facial deformation. The 3D shape, 2D observation, and the head pose can be represented by a generative model in figure 6. Here, $\theta = \{\omega_x, \omega_y, \omega_z, t_x, t_y, t_z\}$ is the 3D head pose, and u is 2D observed positions of landmark points in the image. To extract u from images, we train a feature detectors for each point. The detector is learned using a boosting approach. Each detector is trained by real-Adaboost and its structure is nesting [12]. For each incoming frame, we first use a Lucas-Kanade feature tracker to initialize the 2D landmarks, and apply the feature detectors to update their positions.

Using this generative model, the inference of head pose θ and nonrigid deformation X is divided into two steps:

- Estimate θ. Given the 2D-3D correspondence u and X, 3D head pose estimation $\hat{\theta}$ can be computed by minimizing the reprojection error:

$$\hat{\theta} = \arg\min_\theta \sum_i \|F(P_i; \theta) - p_i\|_2^2 \tag{10}$$

where P_i and p_i are the 3D and 2D position of i-th landmark point, respectively. Equation 10 can be solved by using standard optimization algorithms, such as Gauss-Newton.

- Estimate X. Given u and θ, the initial guess of 3D facial deformations, $\tilde{X}$, can be computed by backprojecting u into 3D using a 3D face model. $\tilde{X}$ is a noisy observation of true deformation, and we rely on the offline learned manifold-based facial deformation model to refine the estimation. For each expression c, algorithm 2 is used to infer the optimal projection $\hat{X}_c$ from $\tilde{X}$ and the posterior probability is computed using equation 9. The estimated deformation $\hat{X}$ is the expectation over posterior probability:

$$\hat{X} = \sum_c \hat{X}_c P(c|\tilde{X}, \hat{L}_c) \tag{11}$$

Algorithm 3 outlines the tracking procedure. Currently, the implemented system runs near 1 fps, the slowest part is the traversal of the manifold (algorithm 2). We believe real time can be achieved by using the power of the GPU.

```
Initialize tracker;
foreach frame do
    $\hat{\theta} \leftarrow$ 3D rigid head tracker;
    $\tilde{u} \leftarrow$ Lucas-Kanade feature trackers;
    repeat
        $\hat{u} \leftarrow$ use feature detector at $\tilde{u}$;
        $\tilde{X} \leftarrow$ backproject $\hat{u}$ using $\hat{\theta}$ and 3D face model;
        $\hat{L}_c, \hat{X}_c \leftarrow$ use algorithm 2 with $\tilde{X}$;
        $P(c|\tilde{X}, \hat{L}) \leftarrow$ use equation 9;
        $\hat{X} \leftarrow$ use equation 11;
        $\hat{\theta} \leftarrow$ optimization using equation 10;
        $\tilde{u} \leftarrow$ project 3D landmark points using $\hat{\theta}$;
    until convergence ;
end
```

Algorithm 3: Tracking algorithm

5. Experiments

5.1. Evaluation of Proposed Method

The objective of this evaluation is to compare the performance of our proposed nonlinear manifold approach with other approaches for recovering correct face shape under noise. We manually label the ground truth position of all landmark points and their 3D coordinates are reconstructed using 3D face model, as described in section 2. For each true 3D shape X, we perturb the 3D position of landmark points independently with Gaussian noise. This step is repeated 10 times to produce 10 test samples. We have 20 ground truth for each expression, total $20 \times 8 = 160$ true shapes and 1600 test samples. The noisy shape, X', is considered as the initial observation of 3D shape. We then use different methods to estimate $\hat{X}$ from X'. This evaluation can be interpreted as examining the "denoising" ability, as we try to filter out the noise and estimate the true shape.

For our proposed method, we estimation $\hat{X}_{\text{TV}}$ using algorithm 2. For the linear subspace approach, we use PCA to learn the person-specific model. We compare with two types of PCA estimation, naive PCA estimation and shrinkage PCA estimation. Recall that PCA is to find the optimal projection of X in linear subspace:

$$X = X_{\text{mean}} + \Phi b$$

where Φ is the matrix of eigenvectors of principal components.

The naive PCA estimation $\hat{X}_{\text{NPCA}}$ of observation X is:

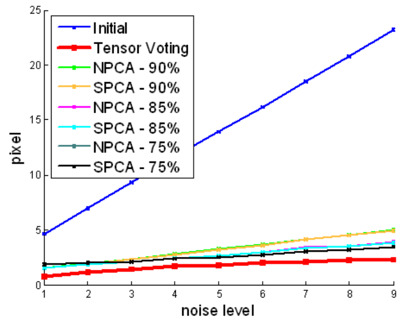
$$\bar{X}_{\text{NPCA}} = X_{\text{mean}} + \Phi\Phi^T(X - X_{\text{mean}}) \tag{12}$$

Although naive PCA enjoys the advantage of easy implementation and inexpensive computation, it has several drawbacks. One is it may generate "unallowable" shapes. This has been investigated in the literature. For example, in [13], Li and Ito have proposed a shape parameter space optimization technique for 2D active shape model. A set of complicated rules is devised to eliminate the unallowable configurations. On the other hand, for 3D shape, [11] propose a shrinkage process to restrict the variation in parameter space. The shrinkage PCA estimation $\hat{X}_{\text{SPCA}}$:

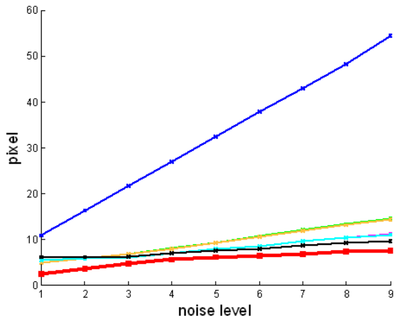
$$\begin{aligned} \hat{b_i} &= \beta_i b_i \\ \beta_i &= \lambda_i/(\lambda_i + \sigma) \\ \hat{X}_{\text{SPCA}} &= \Phi\hat{b} \end{aligned} \tag{13}$$

where b_i is the i-th element of b, λ_i is the eigenvalue of i-th principal component, and σ is the sum of residual energy. β_i is the shrinkage coefficient that controls the feasible range of i-th component; a significant component has larger variation range while a less significant one has smaller range.

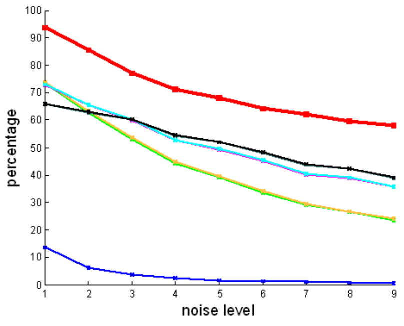
To measure the error, X, X', and $\hat{X}$s are all projected to a 640×480 image. Three errors are computed, average pixel displacement per point, largest pixel distance among all landmark points, and percentage of points within 2 pixels, all measured with respect to the projected position of ground truth. The quantitative evaluation is reported in figure 7. "Initial" indicates the error of initial observation X', and we include it here as the baseline. "Tensor Voting" is the proposed nonlinear manifold approach based on Tensor Voting. "NPCA" and "SPCA" stand for naive PCA and shrinkage PCA, respectively. The attached number 90%, 85%, and 75% is the energy of selected principal components. The low-energy PCA has higher error than high-energy PCA in low noise level, while has lower error in the high noise level. This confirms our understanding of PCA, since decreasing the energy means sacrificing the details, but less sensitive to the noise and outliers. Note that, the SPCA is slightly better than NPCA in this evaluation. Among all cases, the proposed approach is consistently better than other approaches, especially in the high noise level.



(a) Average Error



(b) Max Error



(c) Percentage of Points within 2 Pixels

Figure 7. Quantitative evaluation of our proposed method

5.2. Tracking and Synthesis

Figure 8 shows tracking results from our proposed method. In test sequences, the subject starts from a neutral expression at frontal pose, and performs multiple expressions. The tracker is initialized by an automatic 2D ASM fitting. In figure 8, the first row shows the tracking results. Green arrow indicates the estimated head pose and red points represent the tracked landmark points. We can see that even with out-of-plane rotation, the proposed method still tracks well. Some subtle deformations, such as blinking, can be identified by the proposed algorithm. The second row shows the reprojection of estimated 3D shapes in frontal pose. The last row shows the estimated probability of each expression. There are 8 modes of expression: from left to right is surprise, anger, joy, disgust, sadness, close and open eyes, left-eye blinking, and right eye-blinking.

Probing deeper, figure 9 shows the interplay between the probability and manifold coordinate. "Prob." and "L" are the estimated probability and manifold coordinate of "Surprise", respectively. The bottom row shows the tracked landmarks and example frames, whose index is $0, 5, 10, \ldots, 35, 40$, of this expression. All of them are the output of proposed tracker. "L" is also properly scaled into $[0, 1]$ interval by the maximum value in the training set. It is clear that L can represent the activation level of this expression, which agrees with our interpretation. Besides, the tracker assigns low probability to the correct expression in the onset and offset, since the expression is close to neutral. As the expression starts to activate, the ambiguity decreases and the probability of "Surprise" increases. Figure 10 shows another results from a segment of smile.

The proposed method can be used to generate expression sequences. Figure 11 shows a synthesized sequence of a left-eye blinking expression. The sequence is generated from neutral to the apex of this mode, as we control the manifold coordinate L. We use algorithm 1 to map L to $\hat{X}$ and project them into the 2D image plane.

6. Conclusions and Discussions

We have proposed a new deformable face model based on nonlinear manifolds for 3D facial expressions. The 3D facial deformation model is a combination of several 1D manifolds and each manifold represents a mode of expression. We apply Tensor Voting to learn these nonlinear deformation manifolds. Tensor voting estimates local tangent hyperplane and it provides us a robust estimation tool for unseen, noisy input data. An iterative algorithm is proposed to infer the 3D head pose, 3D deformations, expression class, and manifold coordinate, which indicates the activation level of an expression. Thus, the output of our tracker is a rich representation of the current status of the face.

In the future, we plan to improve the proposed method in several directions. Currently, we are learning person-dependent manifolds. The deformation manifold of each expression should exhibit invariants across different subjects. Investigating this direction will extend the proposed approach be person-independent. The current tracker does not address occlusion explicitly. Including a mechanism to deal with occlusions will make the tracker more robust.

Acknowledgement

The research in this paper was sponsored, in part, by the U.S. Army Research, Development and Engineering Command (RDECOM). The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

References

[1] Y. Bengio, M. Monperrus, and H. Larochelle. Nonlocal estimation of manifold structure. *Neural Computation*, 18(10):2509–2528, Oct. 2006. 2, 4

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. *SIGGRAPH 1999*, pp. 187–194. 1

[3] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *PAMI*, 25(9):1063–1074, Sept. 2003. 1

[4] Y. Chang, C. Hu, and M. Turk. Probabilistic expression analysis on manifolds. *CVPR 2004*, pp. 520–527. 1

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *PAMI*, 23(6):681–685, June 2001. 1

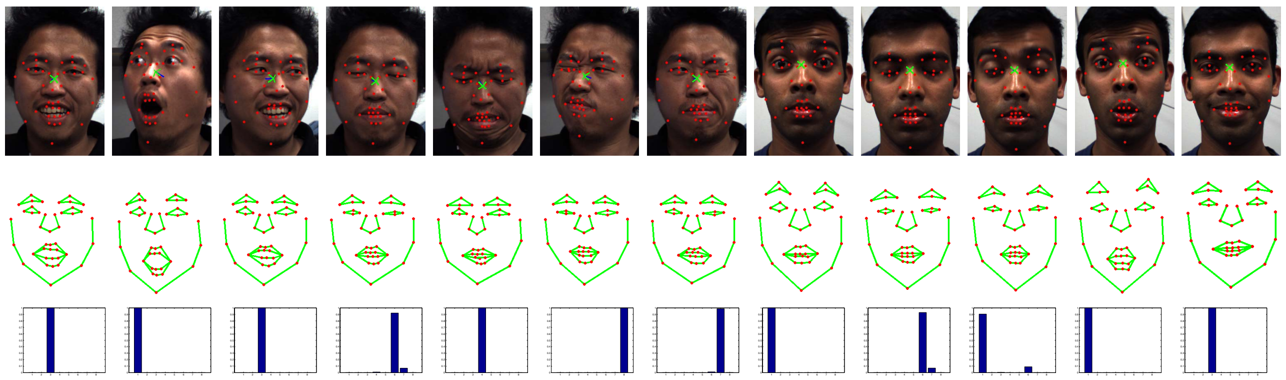


Figure 8. Tracking results

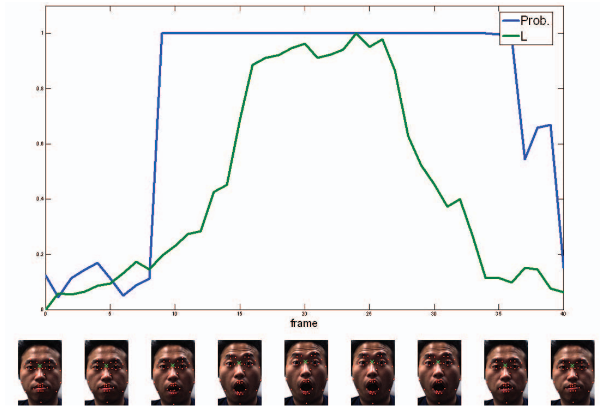


Figure 9. Probability and manifold coordinate of surprise

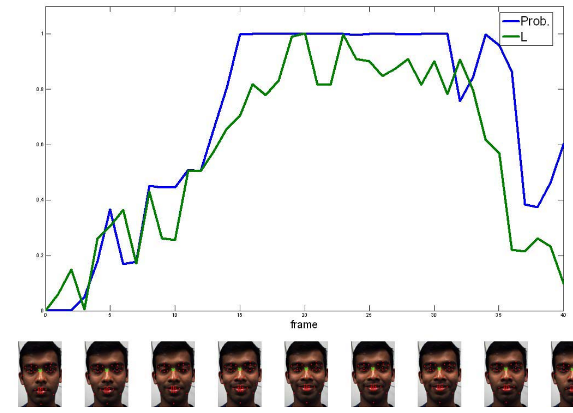


Figure 10. Probability and manifold coordinate of smile

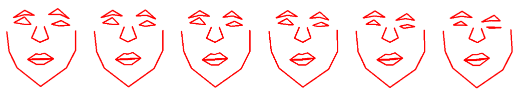


Figure 11. Synthesized shapes of left-eye blinking

[6] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *CVIU*, 61(1):38–59, Jan. 1995. 1

[7] P. Dollar, V. Rabaud, and S. Belongie. Learning to traverse image manifolds. *NIPS 2006*. 2, 4

[8] P. Dollar, V. Rabaud, and S. Belongie. Non-isometric manifold learning: Analysis and an alogrithm. *ICML 2007*, pp. 241 – 248. 2, 4

[9] A. Elgammal. Learning to track: Conceptual manifold map for closed-form tracking. *CVPR 2005*, pp. 724–730. 1

[10] A. Elgammal and C.-S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. *CVPR 2004*, pp. 681–688. 1

[11] L. Gu and T. Kanade. 3d alignment of face in a single image. *CVPR 2006*, pp. 1305–1312. 1, 6

[12] C. Huang, H. Ai, B. Wu, and S. Lao. Boosting nested cascade detectors for multi-view face detection. *ICPR 2004*, pp. 415–418. 5

[13] Y. Li and W. Ito. Shape parameter optimization for adaboosted active shape model. *ICCV 2005*, pp. 251–258. 6

[14] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI 1981*, pp. 674–679. 3

[15] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, Nov. 2004. 1

[16] P. Mordohai and G. Medioni. Unsupervised dimensionality estimation and manifold learning in high-dimensional spaces by tensor voting. *IJCAI 2005*, pp. 798–803. 2, 3

[17] P. Mordohai and G. Medioni. *Tensor Voting: A Perceptual Organization Approach to Computer Vision and Machine Learning*. Morgan and Claypool Publishers, 2007. 2, 3

[18] X. Pennec. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154, July 2006. 2

[19] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, Dec. 2000. 1

[20] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, Dec. 2000. 1

[21] C. Vogler, Z. Li, A. Kanaujia, S. Goldenstein, and D. Metaxas. The best of both worlds: Combining 3d deformable models with active shape models. *ICCV 2007*. 1

[22] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. *CVPR 2004*, pp. 535–542. 1

[23] Z. Zhu and Q. Ji. Robust real-time face pose and facial expression recovery. *CVPR 2006*, pp. 681–688. 1