Adaptive Multiresolution filter Based Multiview Face Recognition

Anil Prabhu.R

Department of Software Engineering
Vins Christian College of Engineering, Nagercoil
Anna University, Chennai, India

Abstract: Face images under wild atmosphere suffer from the changes of heterogeneous factors such as camera view, illumination, expression, etc. Tensor analysis brings a trail of analyzing the influence of heterogeneous factors on facial variety. However, the Tensor Face model develop an obstacle in symbolizing the non-linearity of view subspace. In this paper, to snap this limitation, we furnish a adaptive multiresolution based technique which contains a view-manifold-based Tensor Face in which the latent view manifold preserves the local distances in the multiview face space. Moreover, a Kernelized Tensor Face for multiview face recognition is named to freeze the structure of the latent manifold in the image space. Both methods bring a generative model that entangles a continuous view manifold for hidden view representation. Along with this, we present a Multiresolution filter like using Gabor filter and to extract the edge features using wavelet transform. Conclusively, an expectation maximization like algorithm is evolved to work out the identity and view parameters iteratively for a face image of an unknown/hidden view. The research on the PIE database demonstrates the authority of the manifold construction method. Immense comparison experiments on Weizmann and Oriental Face database for multiview face recognition exhibits the superiority of the proposed adaptive multiresolution based face recognition over the view-based principal component analysis and other state-of-the-art approaches for such purpose.

Keywords- Manifold learning, multiview face recognition non linear tensor decomposition, subspace analysis, Tensor Face, Gabor feature extraction.

1. INTRODUCTION

FACE recognition is an operative research topic in the field of pattern recognition, computer vision, and machine learning. It is also a demanding topic because real-world face images are formed with the interaction of multiple aspects on the imaging condition, including illumination, head poses, etc. Three-dimensional face recognition by comprising 3-D face models has recently been studied. In this paper, we focus on multiview face recognition based on 2-D face images. We intend to build a compact data-driven face representation more specifically, we study two main issues: 1) how to unfold a general multiview face modeling framework that is able to handle multiple unseen views for face recognition and 2) how to estimate the view and identity parameters effectively given a face image with an unknown/hidden view. Correspondingly, our research involves two technical components. First, we want to inspect the low-dimensional intrinsic structure, i.e., the view manifold, in the view subspace. It beams a continuous-valued view manifold, based on which we can build two generative multiview face models. Second, we propose an expectation maximization (EM)-like algorithm to assess the identity and view of a new face image iteratively.

We confirm the effectiveness of the proposed algorithms on two multiview face databases, i.e., the Oriental Face and Weizmann database. To manage the multifactor face modeling and recognition, an elegant multilinear tensor decomposition based on the Tensor-Face framework was proposed. However, Tensor Face unveils its bottleneck which is a limitation in dealing with hidden views due to the discrete nature of the view subspace.

In this paper, we want to embed a continuous-valued nonlinear view manifold in the Tensor Face framework and name it view-manifold-based Tensor Face (V-Tensor Face). V-Tensor Face is able to deal with hidden views by non-linear interpolation along the sequential view points in the low dimensional view subspace. However, the neighborhood relationship in the view manifold may not be well preserved in the high dimensional face image space in V-Tensor Face particularly for the interpolated view points. Thereby, we want to establish a bidirectional mapping between the view manifold and the image space that preserves the local neighborhood structure of views. For this purpose, we employ the kernel trick in tensor decomposition and develop a kernelized Tensor Face (K-Tensor Face) for multi-view face modeling. Both Tensor Face and V-Tensor face methods can be generalized from K-Tensor face.

2. REVIEW OF TENSOR FACE

View variations cause dramatic differences in face images, which have strong influence on the appearance-based face recognition. Thus, the study on view subspace appears to be important. Tensor decomposition provides a trail to analyze the multiple factors of face images independently. The factorized model of multi view faces, which is called Tensor Face, separates the view and identity factors with their corresponding subspaces. The notations are explained as follows. Scalars are denoted by lowercase letters (a, b,…), vectors by bold lowercase letters (a, b, …), matrices by bold uppercase letters (A, B,…), and higher order tensors by calligraphic uppercase letters (A, B,…).
The tensor data can be regarded as a multi-dimensional vector. In multi-linear algebra, each dimension of the N-order tensor $D = (x_1 x_2 x_3 \ldots x_L)$ is associated with a mode. Tensor decomposition through high-order SVD (HOSVD) seeks for N orthonormal mode matrices. To obtain the Tensor Face model of multiview faces, we recenter face images along pixel, identity, and view variations as tensor $Y$. Then, we apply HOSVD on it to get:

$$Y = [C \times U_{\text{pixel}} \times U_{\text{identity}} \times U_{\text{view}}]$$

Where $x_n$ represents the mode-n product between core tensor $C$ and mode matrix $U_n$. The core tensor governs the interaction among mode matrices, $U_{\text{pixel}}$ orthonormaly spans the space of eigenimages. Rows of $U_{\text{identity}}$ and $U_{\text{view}}$ spans the parameter space of different identities and views, respectively, which separates the influences of identity and view factors.

We extract an identity vector $p^k$ from $U_{\text{identity}}$ and a view vector $x_i$ and $U_{\text{view}}$ to represent person $k$ and view $i$, respectively. Tensor Face enables us to represent the training image $Y$ of the $k$th identity under the $i$th view based on $C x_1 U_{\text{pixel}} x_2 U_{\text{identity}} x_3 U_{\text{view}}$ as follows:

$$Y = C x_1 U_{\text{pixel}} x_2 U_{\text{identity}} x_3 U_{\text{view}}$$

Given a test facial image $y$, face recognition is used to identify the closest identity vectors in $U_{\text{pixel}}$. However, without the view information, we cannot calculate the identity vector of $Y$ based on (2). Tensor Face assumes that the view vector of $Y$ is the $i$th row of $U_{\text{view}}$. Thus, the test view is regarded as one of the training views. The identity vector $p^k$ under view $i$ can be obtained as:

$$p^k = y \times^{i-1}$$

Where $B$ is the flattened matrix of sub tensor (Cupixel) along the identity dimension. Then, face recognition is used to find the nearest neighbor of $p^k$ in the identity subspace $U_{\text{identity}}$, i.e.,

$$\hat{r} = \text{argmin}_{k=1} \| p^k - p^k \|$$

Where $\{1, \ldots, V\}$ and $k(1, \ldots, K)$ denotes the Euclidean distance between two vectors.

Tensor Face-based multi-view face recognition traverses the discrete identity and view subspace obtained from the training data to find a best match combination of identity and view vectors. The discrete nature of the view subspace in Tensor Face lacks the power of representing the view structure. Thus, it may not handle the unseen views which are significantly diverging from the training ones. Moreover, the effective solution of the parameters instead of using an exhaustive search is another issue of Tensor Face-based recognition.

3. V- TENSOR FACE

The key point of V- Tensor Face is to build a view manifold which can represent the view nonlinearity and preserve the structure of the view subspace. Thus, V-Tensor Face will lead to a better representation of multi-view faces than Tensor Face. There are two problems to be solved: 1) how to build such a view manifold and ensure its validity at the same time and 2) how to incorporate the manifold into the Tensor Face model.

3.1 View-Manifold-Based TensorFace Modeling

To represent the view nonlinearity, we embed the view manifold in TensorFace to build V-TensorFace. In this model, face image $y$ of the $k$th identity $i$th view can be generated by (6)

$$y = C x_1 U_{\text{pixel}} x_2 p^k x_3 M(x_i)$$

Here, $p^k$ is the identity coefficient of the $k$th subject. $M(x_i)$ is the ith view coefficient on the view manifold $M$. The major difference between (2) and (6) is the mode-3 multiplier. In (2), the mode-3 multiplier is a view coefficient selected from the discrete mode matrix $U_{\text{view}}$ (the red line in Fig. 1). This manifold helps to generate the face images of $p^k$ under unseen views on the view manifold. In this model, tensor decomposition is used once, and the low-dimensional view nonlinearity and continuity are represented by the view manifold.

4. K-TENSORFACE

---

**Fig 1:** Architecture of Adaptive multiresolution filter based multiview face recognition

**Fig 2:** Block Diagram
The purpose of building a multiview face model is to find the latent structure of multiview faces then build a homomorphism mapping between the latent variable space and the image space. Most nonlinear dimensionality reduction approaches focus on preserving the local distance of the high-dimensional data in the latent space. For example, the view manifold in V-TensorFace preserves the local view adjacency relationship. However, this model does not guarantee that the local distances in the latent view subspace will be preserved in the image space, particularly for the interpolated unseen views. Since kernel mapping can provide a smooth interpolation from the low- to high-dimensional space, we introduce it to V-TensorFace model to build a smooth bidirectional mapping, the manifold generation mechanism guarantees the locality preservation in the latent space, while the locality property of the adopted Gaussian kernel ensures the manifold structure preservation in the image space.

4.1 Kernelized TensorFace Modeling

We briefly nonlinear tensor decomposition in the context of multiview face modeling. This method can provide a compact low-dimensional representation of multiview face images by exploring both the linear(e.g., the identity) and nonlinear(e.g., the view) variable spaces. We refer the readers to for more details. Given the multiview face images $y_{1:n}$ of $K$ persons, each person has $N$ views, which are represented by $\{x_{1}, x_{2}, \ldots, x_{N}\}$ in the nonlinear view manifold. We use kernel $\psi(\cdot)$ of the generalized radial basis function (RBF) to build the nonlinear mapping between the view manifold and the high-dimensional image space as

$$y^i = \sum_{j=1}^{N} \omega_j \psi(\|x - z_1\|) \delta + I(x) b^i$$

Where $y^i$ denotes the $i$th element in $y$. $z$ are kernel centers sampled from the view manifold, and is the weight of each kernel. $b^i$ is the mapping coefficient of linear polynomial $l(x) = [1, x]$. The mapping between the multiview face images of the $K$th person and their low-dimensional view coefficients can be represented by

$$\begin{bmatrix}
\psi(x_1) \\
\psi(x_2) \\
\vdots \\
\psi(x_N)
\end{bmatrix}
= D^k$$

Where A=[[l(x_1), l(x_2), \ldots, l(x_N), O_{r+1}r+1]]. The multiplication between A and $D^k$ ensures their orthogonality and makes the solution of $D^k$ well posted. $\psi(x) = [\phi(||x - z_1||), \ldots, \phi(||x - z_{N}||), 1, x]$ is a vector of dimension $1 \times (N + e + 1)$. $D^k$ is a $(N + e + 1) \times d$ matrix. Its jth column is $[\phi_0, \phi_1, \ldots, \phi_d]$. For the $K$th person under view $I$, the mapping can be represented by

$$y^i = \psi(x_i)D^k. \quad (9)$$

The linear part $I(x)$ of $\psi(x)$ supports the conditionally positive property of the RBF mapping [52], [53]. It also provides a means to solve each $x$ without solving the inverse of the nonlinear part of $\psi(x)$. In (8) and (9), $y$ is identity dependent. However, $\psi(x)$ is identity free. We deduce that the identity information is embedded in the linear matrix $D$ in this mapping. To extract the identity information, we stack $D^{1:k}$ to form a tensor $D = [D^1, \ldots, D^k]$. Then, we apply HOSVD to $D$ to abstract the low-dimensional identity coefficients $p^i \in R^k$. This results in the generative multiview face model K-TensorFace. In this model, $y^i \in R^k$ can be synthesized by identity $p$ and view $x$, coefficients as

$$y^i = Cx_1 \psi(x_2) p^i x_1 \psi(x_1)$$

Where $C$ is a 3-rd core tensor.

The process of model learning and image synthesis is shown in Fig. 2. In the learning phase, the multiview face images $y^i$ are mapped to their low-dimensional viewpoints $x$ in the manifold $M$ according to the RBF mapping in (9). The localized Gaussian kernels in RBF mapping attempt to preserve the latent structure of view manifold in the high-dimensional image space even for the interpolated viewpoints. Then, the linear matrix $D$ of each person is stacked to form a tensor $D$. We decompose $D$ with HOSVD to obtain the identity coefficients $p^i$ and the core tensor $C$. Finally, multiview face images are synthesized in the synthesis part.

4.2 Discussion on the Choice of View Manifolds

The RBF mapping can build a nonlinear mapping between face images and different kinds of view manifolds. The performance of K-TensorFace will be significantly affected by the involved view manifold. In order to cope with the view nonlinearity in the K-TensorFace model, we briefly discuss three kinds of view manifold generation methods: data-, concept-, and hybrid data-concept-driven view manifolds.

View manifolds can be deduced from the training data using nonlinear dimensionality reduction methods like LLE or ISOMAP. However, those view manifolds are person dependent and cannot be used for multiview face modeling that requires a commonly shared view manifold. The concept-driven manifolds originate from an ideal conceptual design. For example, gait observations of full walking cycle under a fixed view can be embedded on a 2-D circular-shaped manifold. Moreover, gait observations from multiple views can be embedded on a 3-D torus. These conceptual manifolds were shown effectively on exploring the intrinsic low-dimensional data. In our case, multiview face images rotating from the leftmost to the rightmost are obtained by placing cameras with equal angle spacing along a semicircular bracket. Hence, conceptually, we design a view manifold on a semicircle. In V-TensorFace, the view coefficients in $U_{view}$ construct the global nonlinear structure of the view manifold (data driven). Moreover, the view coefficients interpolated by spline fitting construct the
locally linear structure of the view manifold (concept driven), which use the prior knowledge of manifold smoothness. It is a hybrid data-concept-driven manifold generation method.

A basic requirement of learning K-TensorFace is that the view manifold must be shared by all persons, which means the view manifold is invariant to identity changes. This makes identity-dependent data-driven view manifolds unusable here. The conceptual view manifold uses the prior knowledge of view rotating in the physical space. Although it is purely determined by the conceptual design, which is independent to the identity changes, it may not necessarily capture the intrinsic nonlinear view structure of the multiview face images. Therefore, the hybrid view manifold seems to be a better choice since it has a proper balance between the generality and the specificity.

5. GABOR FILTER

The features extraction for face recognition using Gabor filters yield good results. So we use here a Gabor wavelets based feature extraction technique. We use the following family of two-dimensional Gabor kernels:

\[
\psi(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x^2 + y^2}{\lambda}\right)
\]

Where \((x, y)\) specify the position of a light impulse in the visual field and \(\lambda, \phi, \sigma, \gamma\) are parameters of the wavelet. We have chosen the same parameters used by Wiskott.

**TABLE 1:** Parameters of Gabor Wavelets

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orientation</td>
<td>(\Theta)</td>
<td>({0, 2\pi, 4\pi, 6\pi, 8\pi})</td>
</tr>
<tr>
<td>Wavelength (h)</td>
<td>(\lambda)</td>
<td>({4, 4\sqrt{2}, 8, 8\sqrt{2}, 16})</td>
</tr>
<tr>
<td>Phase</td>
<td>(\varphi)</td>
<td>({0, \frac{\pi}{2}})</td>
</tr>
<tr>
<td>Gaussian Radius</td>
<td>(\sigma)</td>
<td>(\sigma = 1)</td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>(\gamma)</td>
<td>1</td>
</tr>
</tbody>
</table>

A set of Gabor filters is used with 5 spatial frequencies and 8 distinct orientations, this makes 40 different Gabor filters represented as,

![Real part of filter](image1)

![Magnitude of filter](image2)

Fig 3: Gabor Filter

We have selected 5 sets of Gabor filters with different orientations.

**Table 2:** Sets of Gabor Filters for different orientations

<table>
<thead>
<tr>
<th>The Number of Gabor Filters</th>
<th>The Orientations</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>(\Theta = [0])</td>
</tr>
<tr>
<td>10</td>
<td>(\Theta = [0, 2\pi/8])</td>
</tr>
<tr>
<td>15</td>
<td>(\Theta = [0, 2\pi/8, 4\pi/8])</td>
</tr>
<tr>
<td>20</td>
<td>(\Theta = [0, 2\pi/8, 4\pi/8, 6\pi/8])</td>
</tr>
<tr>
<td>25</td>
<td>(\Theta = [0, 2\pi/8, 4\pi/8, 6\pi/8, 8\pi/8])</td>
</tr>
</tbody>
</table>

When Gabor filter are applied to each pixel of the image, the dimension of the filter vector can be very large(proportional to the image dimension). So, it will leave to expensive computation and storage cost. To alleviate such problem and make the algorithm robust, Gabor features are obtained only at the 10 extracted fiducial points. If we note \(\mathcal{C}\) the number of Gabor filters, each points will be represented by a vector of \(\mathcal{C}\) components called “Jet”. When applying the Gabor filters to all fiducial points, we obtain a jets vector of 10\(\mathcal{C}\) real coefficients characterizing the face.

6. GENERAL FRAMEWORK FOR MULTIVIEW FACE REPRESENTATION

We cast the TensorFace-, V-TensorFace-, and K-TensorFace-based multiview face models in a general framework to help readers understand the relationship among them. Then, we give an iterative parameter searching method for face recognition.

There are two main differences among TensorFace, V-TensorFace, and K-TensorFace. One is the view representation, and the other is the mapping between the view subspace and image space. In the following, we want to revisit TensorFace and V-TensorFace from the K-TensorFace point of view to examine the aforementioned two main differences.

To get TensorFace, we define in \(\psi(x) = x\) (10), where \(x \in \mathbb{U}_{\text{view}}\). Then on linear view variation cannot be accurately represented in TensorFace because of the sparse view representation. For this purpose, V-TensorFace is proposed, for which we define \(\psi(x) = \text{M}(x)\) in (10). The interpolated views in the manifold \(\mathcal{M}\) may not be preserved very well in the image space. Thus, the Gaussian kernel \(\varphi(\cdot)\) was incorporated to build a smooth bidirectional mapping between the view manifold and the image space. It results in the K-TensorFace model where \(\psi(x) = [\varphi(\|x - z_1\|), \ldots, \varphi(\|x - z_K\|), 1, x]\) and \(x \in \mathbb{M}\). Both V-TensorFace and K-TensorFace involve a view manifold, with which we develop an iterative EM-like parameter searching method. V-TensorFace can be regarded as the linearization of K-
TensorFace. When the view coefficients, V-TensorFace degenerates into TensorFace. In this perspective, we call K-TensorFace a unified multiview TensorFace model.

In the previous section, we have described a multiview face with the V- and K-TensorFace models. It is also necessary to model the inverse process. In other words, given a view manifold $M \subset \mathbb{R}^d$ and a new input $y \in \mathbb{R}^d$, we need to estimate its view parameter $x \in M$ and identity parameter $p^i$ by minimizing the reconstruction error. Unfortunately, the solution of each parameter depends on the existing of another one. The EM-like algorithm is suited to this type of “chicken-and-egg” problem. Usually, the face variations caused by illumination and viewing direction may be larger than the interperson variation in distinguishing identities. As the distance between images with different identities under the same view($D_i$) may be smaller than that between images with the same identity but different views ($D_j$). Therefore, identity coefficients are more sensitive than view coefficients to new observations, and thus, we start with view initialization.

6.1. Initialization: It contains three steps. It can be the hybrid data-conceptual manifold.

6.1.1. View estimation under each identity. Given a test image $Y$, we assume its identity coefficients are $p^k$, $k=1,\ldots,K$, of the training data. Then, $K$ view coefficients $x^k \in \mathbb{R}^p$ with respect to each identity are solved by taking the linear part of $\psi(\cdot)$ in (11), in which $D^k$ is the flattened matrix of subtensor $C_{j_1} U_{j_2 x_3 p^k}$. $\psi(x^k)$ is a preset variance controlling the algorithm sensitivity. $M(x^k)$ is the viewpoint on the manifold projected from $x^k$.

6.1.2 View Coefficient Synthesis: For each coefficient $y$, we synthesize its view coefficient by $x^k = \sum_k p(x^k | y)$, where $p(x^k)$ denotes the probabilities of $x^k$ belonging to $M$. It can be determined under the Gaussian assumption as

$$p(x^k) \propto \exp\left(-\frac{1}{2\sigma^2} \left(\|x^k - M(x^k)\|^2 + 2\sigma^2\right)\right)$$

where $\sum_k p(x^k) = 1$ and $\sigma$ is a preset variance controlling the algorithm sensitivity. $M(x^k)$ is the viewpoint on the manifold projected from $x^k$.

6.1.2 View projection to the manifold. The view coefficient is initialized by projecting $X$ to the manifold as $X = M(x)$.

Identity Estimation: Given the estimated view coefficient $X$, we calculate $\psi(X) = [\phi_1(X), \phi_2(X), \ldots, \phi_n(X)]$. Then, identity recognition can be performed in the CD in terms of $p^i$ as

$$k_{CD} = \arg \min_i \|p^i - p^h\|$$

Or we consider $Y$ as drawn from a Gaussian mixture model centered at the reconstructed identity class $K$. Therefore, the likelihood function of observation $p(y|k, x)$ belonging to person $k$ can be formulated as

$$p(y|k, x) \propto \exp\left(-\frac{1}{2\sigma^2} \left(\|y - C_{j_1} U_{j_2 x_3 p^k} \sum_k p(x^k | y)\|^2\right)\right)$$

With the equal probability assumption of $p(k)$ and $p(k|x)$, the posterior probability of identity $K$ is reduced to $p(k|x, y) = p(y|k, x) / \sum_k p(y|k, x)$.

Then, identity recognition can be performed in the reconstructed ID as

$$k_{ID} = \arg \max_y p(y|x, y)$$

We will test both schemes with experiments. A new identity. View estimation can be obtained as

$$p^i = \sum_k \frac{p(x^k | y)}{\sum_{k'} p(x^{k'} | y)}$$

View Estimation: Given the test image $Y$ and the estimated $p^i$, we can solve a new view coefficient $x^i \in \mathbb{R}^d$ in (10) based on (11). Then, the updated view coefficient that is constrained on the view manifold is obtained by $x = M(x^i)$. Therefore, the identity view coefficients can be solved iteratively until the termination condition is met.

Fig 4: Distance comparison between identity and view variations

7. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we briefly introduce the experimental databases first, i.e., Oriental Face, Weizmann, and PIE. To validate the proposed multiview face model, we report the results obtained on multiview faces in the Weizmann and Oriental Face databases. The view and identity recognition results of TensorFace and V-TensorFace on both databases are given to show the effectiveness of the hybrid data-concept-driven view manifolds on K-TensorFace, we recognize the identity of Oriental Face database in both ID and CD with the concept driven view manifold [concept manifold(CM)] and hybrid data-concept-driven view manifold (HM). Finally, we extend the proposed method to multi-illumination face recognition. The manifold generation method is verified on the illumination (order unknown) manifold.
construction on the PIE database.

Databases:

In the Oriental Face database, we use 1406 face images of 74 oriental people, and each one has 19 views in the range of [−90◦, . . . , 90◦] for pan sampled at around the 10 interval along a semicircle. Weizmann face database contains 28 individuals, and each one has five views. In the PIE database, the frontal face images of 60 persons obtained by Camera 27 are chosen. Each person has 21 illuminations caused by 21 flash firing variation with the room lights off. Since most subspace-learning-based face recognition methods are sensitive to spatial misalignments, all images are shifted to align the eyes and nose tip position found manually. The automatic alignment can be realized by reconstructing from the misaligned images. This method is applicable in our method to achieve automatic face alignment. The images in the Weizmann database are in the size of 64*50 pixels since the faces in this database are a bit longer than others. The rest of the images in the size of 50*50 pixels.

The Experiments on the Oriental Face Database

The original Face database contains a long range of view variations. On this database, experiments are compared among VPCA, TensorFace, V-TensorFace and K-TensorFace. In V-TensorFace, the identity parameter is solved in both CD and ID without iteration and iteratively. In K-TensorFace, when the cases of CM and HM are combined with recognition in CD or ID, we have four experiments to test. They K-TensorFace with CM/CD, CM/ID, HM/CD and HM/ID respectively, in which the parameter estimation is iterative. In addition, we test K-TensorFace with iterative parameter estimation and HM (K-TensorFace HM/Itera).

The recognition rates of V-TensorFace are 6.44%–8.31% higher than those of TensorFace. It indicates that the manifold works better than the discrete view coefficients. Comparing the results in Tables V and VI, we can find that the identity recognition rate of TensorFace on the Oriental Face database is improved around 30% than that on the Weizmann database, which has slightly bigger view intervals. However, the proposed V- and K-TensorFace models are more robust than TensorFace to view sampling distance.

Even TensorFace and VPCA are comparable with regards to the identification rate. TensorFace represents multiview face images in a unified basis, which provides a general parametric face model with explicit physical meaning of different factors. Although the cameras are distributed at about 10 interval along a semicircle during the imaging of the oriental faces, these images do not obey a uniform distribution in the view subspace. We calculate the distance between the neighboring view pairs are quite different. It is obvious that intervals View5–View6 and View14–View15 are too sparseto support valid view interpolation. Thus, the recognition results in these intervals are usually lower than those in other intervals. Note that, in Table VI, Mean-17 is the mean of the recognition rates of View2−View18. K-TensorFace with CM performs worse than TensorFace in both CD and ID, which shows that the concept driven view manifold is not effective in capturing in intrinsic structure of the view subspace. The obtained multiview images do not obey a uniform distribution in the view subspace, which is particularly obvious in the intervals View5–View6 and View14–View15. The low recognition rates of HM in these intervals show that he performance of K-TensorFace depends on the distribution density of the manifold.

![Fig 4: Comparision between tensorface and proposed adaptive multiresolution filter based technique in terms of accessing speed.](image-url)

The recognition rates of K-TensorFace with HM are improved by 6.75%–9.36% and 12.96%–14.66% than VPCA in CD and ID, respectively. Compared with TensorFace, the recognition rates of K-TensorFace with HM are improved around 7% and 12% in CD and ID, respectively. The iterative ID-based K-TensorFace method is 18.23%–18.85% better than TensorFace in identification. As seen from our intensive experiments, the iterative parameter estimation algorithm converges in 6-10 steps. An example of the converge curve is shown in Fig. 9. The convergence rate depends on the initialization. The V- and K-TensorFace models have almost the same recognition rates in CD. However, the recognition rate of K-TensorFace is around 3% higher than that of V-TensorFace in ID whether with or without the iterative parameter estimation. Since the recognition is excuted in the reconstructed ID, we attribute this...
improvement to the smooth kernel mapping in K-TensorFace.

**Validation of the GA Shortest Path-Based Manifold Generation Method:**

View manifold generation is used to preserve the sequential order of multiple views in the image space. The view order is easily observed. Thus, we test the GA shortest path-based manifold generation method on the data without knowing the view order. In Fig. 7(c), the illumination order of the 21 faces are unknown. Thus, we extend the manifold the manifold generation method to explore the manifold topology of illumination to validate its effectiveness. To obtain the identity-independent illumination coefficients, we apply HOSVD on the PIE data to get the mode matrix of illumination, which contains the coefficient vectors of 21 illuminations. We illustrate the first two dimensions of the illumination coefficients by the blue triangles in Fig. 10 and connect them with the shortest path. The faces connected by the red line are demonstrated in the lower part of the figure, which shows that the connected illumination coefficients are sorted in a responsible illumination-varying order.

Fig 5: Comparision between tensorface and proposed adaptive multiresolution filter based technique in terms of recognition rate.

Usually when the illumination variation is smooth, the linear face recognition method achieves very good results. However, face recognition on sparsely sampled illuminations is a challenging task because of the nonlinear illumination changes. To handle this nonlinearity, we use K-TensorFace to build a multi-illumination face model. We adopt only five illuminations, namely, Illum1, Illum4, Illum10, Illum12, and Illum16 as the training data, which are marked with yellow squares in Fig. 10. The identity recognition is compared among VPCA, TensorFace, V-TensorFace and K-TensorFace. In the last two methods, parameters are solved iteratively. The identity in each illumination is recognized by the leave-one-out style cross-validation on those five illuminations. The results are given in Table VII. We can see that K-TensorFace, with iteration has better results than TensorFace. The identity recognition rate is improved by around 13%. Which means that K-TensorFace has better ability to handle the illumination nonlinearity.

**TABLE 3:** Comparision based on dataset and speed of processing.

<table>
<thead>
<tr>
<th>% of Dataset</th>
<th>Speed of accessing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional</td>
<td>Proposed</td>
</tr>
<tr>
<td>10</td>
<td>3.7</td>
</tr>
<tr>
<td>20</td>
<td>3.6</td>
</tr>
<tr>
<td>30</td>
<td>3.65</td>
</tr>
<tr>
<td>40</td>
<td>3.7</td>
</tr>
<tr>
<td>50</td>
<td>3.8</td>
</tr>
</tbody>
</table>

The show the smooth projection from the illumination manifold to the image space in K-TensorFace we synthesize new faces using K-TensorFace model under the interpolated illuminations in the manifold. In this experiment, the illumination manifold is derived from only five illuminations. The remaining 36 images are the synthesized ones. We can see that the synthesized images look natural and realistic. Some of them are very close to the real images in the PIE database. The experiments in this section also show that the proposed general model can be extended to multifactor face recognition task.

**TABLE 4:** Comparision based on Rate of Projection and Recognition Rate.

<table>
<thead>
<tr>
<th>Rate of Projection</th>
<th>Recognition Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Conventional</td>
</tr>
<tr>
<td>5</td>
<td>.62</td>
</tr>
<tr>
<td>10</td>
<td>.72</td>
</tr>
<tr>
<td>15</td>
<td>.75</td>
</tr>
<tr>
<td>20</td>
<td>.78</td>
</tr>
<tr>
<td>25</td>
<td>.81</td>
</tr>
<tr>
<td>30</td>
<td>.82</td>
</tr>
<tr>
<td>35</td>
<td>.82</td>
</tr>
</tbody>
</table>

8. CONCLUSION AND FUTURE WORK
In the multiview face recognition tasks, normally, linear methods cannot locate the unfamiliar intermediate views accurately. To represent the nonlinearity in view subspace, we embedded a novel view manifold to TensorFace and validate the effectiveness of the manifold with the shortest path method. In this project, to break this limitation, we present a Multiresolution filter like using Gabor filter and to extract the edge features using wavelet transform, most importantly, we propose a unified framework to generalize Tensor-Face, V-Tensor Face, and K-Tensor Face. Finally, a nearest neighbor classifier is developed to estimate the identity and view parameters iteratively for a face image of an unknown/unseen view. The experiment on the PIE database shows the effectiveness of the manifold construction method. Using the posed face images at four angles: 0°, 30°, 60°, and 90° as templates, the performance of pose estimation of about 80% has been achieved for test images in the entire angle range of 0° - 90°.

REFERENCES


