# Face Recognition and Pose Estimation with Parametric Linear Subspaces

Kazunori Okada[†][*] and Christoph von der Malsburg[‡]

[†]Imaging and Visualization Department
Siemens Corporate Research
Princeton, NJ 08540 USA
kazunori.okada@siemens.com

[‡]Institut für Neuroinformatik
Ruhr-Universität Bochum
Bochum, D-44801 Germany
malsburg@neuroinformatik.ruhr-uni-bochum.de

**Abstract**

We present a general statistical framework for modeling and processing head pose information in 2D grayscale images: analyzing, synthesizing, and identifying facial images with arbitrary 3D head poses. As a basic component, LPCMAP model offers a compact view-based data-driven model with bidirectional mappings between face views and 3D head angles. We call a mapping from face to pose analysis mapping and that from pose to face synthesis mapping. A model matching is also defined by an analysis-synthesis chain that concatenates the two mappings. Such a mapping-based model implicitly captures 3D geometric nature of the problem without explicitly reconstructing 3D facial structure from data. The model is learned by using efficient PCA and SVD algorithms resulting in linear functional forms. They are however only locally valid due to the linear design. We further extend this local model to mitigate the shortcomings. PPLS model extends the LPCMAP for covering a wider pose range by piecing together a set of LPCMAPs. Multiple-PPLS model further extends the PPLS for generalizing over different individuals. These proposed models are applied to solve pose estimation and animation by using the analysis and synthesis mappings, respectively. A novel pose-insensitive face recognition framework is also proposed by exploiting the PPLS model to represent each known person. In our recognition framework, the model

[*] The corresponding author can also be reached at 3641 Lavell Drive Los Angeles, CA 90065, kazokada@earthlink.net. Main part of the presented work was conducted while both authors were at University of Southern California, Los Angeles USA.

matching with the PPLS models provides a flexible pose alignment of model views and input faces with arbitrary head poses, making the recognition invariant against pose variation. Implementations of the proposed models are empirically evaluated with a database of various views of 20 people rendered from Cyberware-scanned 3D face models. The results demonstrated sub-degree pose estimation and sub-pixel shape synthesis accuracy, as well as high degree of generalization to unseen head poses within ±50 degree range of full 3D head rotation. For recognition and interpersonalized pose estimation, the results also indicated robustness against unseen head poses and individuals while compressing the data by a factor of 20 and more.

# 1 Introduction

Face recognition is one of the most interesting and challenging problems in computer vision and pattern recognition. In past many aspects of this problem have been rigorously investigated because of its importance for realizing various applications and understanding our cognitive processes. For reviews, see [37, 6, 42]. Past studies in this field have revealed that our utmost challenge is to reliably recognize people in the presence of image/object variations that occur naturally in our daily life [32]. Among others, head pose variation is one of most common variations because humans and their heads can move freely. Thus, handling of head pose variation is an extremely important factor for virtually any realistic and practical application scenarios. There have been a number of studies which specifically addressed the issue of pose invariance in face recognition [2, 31, 23, 20, 38, 3, 1, 13, 11, 16, 12, 15, 41, 30, 40, 5]. Despite the accumulation of studies and relative readiness of the problem, however, performance of the state-of-the-art has unfortunately remained inferior to human ability and sub-optimal for practical use when there is no control over subjects and when one must deal with an unlimited range of full 3D pose variation.

Our main goal is to develop a simple and generalizable framework which can be readily extended beyond the specific focus on head poses (e.g., illuminations and expressions), while improving the pose processing accuracy of the state-of-the-art. For this purpose, we propose a general statistical framework for compactly modeling and accurately processing head pose information in 2D images. The framework offers means for analyzing, synthesizing, and identifying facial images with arbitrary head poses. The model is data-driven in a sense that a set of labeled training samples are used to learn how facial views change as a function of head poses. For realizing the compactness, the model must be able to learn from only a few samples by generalizing to unseen head poses and individuals. Linearity is emphasized in our model design, which simplifies the learning process and facilitates generalization by avoiding typical non-linear pitfalls such as over-fitting [4].

For pose-insensitive face recognition, previous works can roughly be categorized into two types: single-view and multi-view approach. The single-view approach is based on the person-independent transformation of test images [23, 20, 16]. The pose invariance is achieved by representing each known person by a facial image with a fixed canonical head pose and by transforming each test image to the canonical pose. This forces that head poses of the tests and gallery entries are always aligned when they are compared for identification. An advantage of this approach is the small size of the known-person gallery, however their recognition performance tends to be low due to the difficulty of constructing an accurate person-independent transformation. On the other hand, the multi-view approach [2, 3, 12, 41] is based on the multi-view gallery, which consists of multiple views of various poses for each known person. Pose-invariance is achieved by assuming that, for each input face, there exists a view with the same head pose as the input for each known person in the gallery. These studies have reported generally better recognition performance than the single-view approach. The large size of the gallery is, however, a disadvantage of this approach. The recognition performance and the gallery size have a trade-off relationship; to improve the performance requires denser sampling of the continuous pose variation, increasing the gallery size. This increase of the gallery size makes it difficult to scale the recognition systems to the large number of known people and makes the recognition process more time-consuming.

One solution to the trade-off problem is to represent each known person by a compact model. Given the multi-view gallery, each set of views of a known person can be used as training samples to learn such a personalized model, reducing the gallery size while maintaining high recognition performance. The parametric eigenspace method of Murase and Nayar [25] and the virtual eigensignature method of Graham and Allinson [13] are successful examples of this approach. These methods represent each known person by compact manifolds in the embedded subspace of the eigenspace. Despite their good recognition performance, generalization capability is their shortcoming. Both systems utilized non-linear methods (cubic-spline for the former and radial basis function network for the latter) for parameterizing/modeling the manifolds. Such methods have a tendency to overfit peculiarities in training samples, compromising capability to generalize over unseen head poses. The proposed solution emphasizes on linearity in model design, facilitating such generalization, as well as model compactness.

Our investigation explores the model-based solution of pose estimation, pose animation, and pose-insensitive face recognition using parametric linear subspace models. As a basic component, we propose LPCMAP model that offers a compact view-based model with bidirectional mappings between face views and 3D head angles [27]. We call a mapping from face to pose analysis mapping and that from pose to face synthesis mapping. A model matching is also defined by an analysis-synthesis chain that concatenates the two mappings. Availability of such mappings avoids the necessity of an exhaustive search in the parameter space. Its parametric nature also provides an intuitive interface that permits clear interpretation of image variations and enables the model to continuously cover the pose variation thereby improving accuracy of the previous systems. Such a mapping-based model implicitly captures 3D geometric nature of the problem without explicitly reconstructing 3D facial structure from data. The model is learned by using efficient PCA and SVD algorithms resulting in a linear form of the functions. They are however only locally valid due to the linear design. Therefore this local model is further extended to mitigate the shortcomings. PPLS model extends the LPCMAP for covering a wider pose range by piecing together a set of LPCMAPS [29]. Multiple-PPLS model further extends the PPLS for generalizing over different individuals [26]. The discrete local models are continuously interpolated, improving the structurally discrete methods such as the view-based eigenface by Pentland et al. [31].

These proposed models are successfully applied to solve pose estimation and animation by using the analysis and synthesis mappings, respectively. A novel pose-insensitive face recognition framework is also proposed by exploiting the PPLS model to represent each known person. In our recognition framework, the model matching with the PPLS models provides a flexible pose alignment of model views and input faces with arbitrary head poses, making the recognition invariant against pose variation. As a pure head pose estimation application, the analysis mapping can also be made to generalize inter-personally by using the multiple-PPLS model that linearly combines a set of personalized PPLS models.

The rest of this article is organized as follows. In Section 2, we overview our framework and introduce some basic terminologies. Section 3 describes the LPCMAP and PPLS models in details. Section 4 shows how we can extend the PPLS model inter-personally. In Section 5, we empirically evaluate effectiveness of the proposed models. In Section 6, we conclude this article by summarizing our contributions and discussing future work.

# 2 Problem Overview and Definitions

The problem of our focus is to learn a data-driven statistical model of how facial appearance changes as a function of corresponding head angles and to apply such learned models for building a face recognition system that is insensitive to the pose variation. The following introduces formal descriptions of our problem, as well as terminologies used throughout this paper.

## 2.1 Statistical Models of Pose Variation

### 2.1.1 Analysis and Synthesis Mappings

We employ the parametric linear subspace (PLS) model [27, 29] for representing such pose-modulated statistics. A PLS model consists of bidirectional, continuous, multivariate, mapping functions between a vectorized facial image $\vec{v}$ and 3D head angles $\vec{\theta}$. We call a mapping $\mathcal{A}_\Omega$ from the image to angles *analysis mapping*, and its inverse $\mathcal{S}_\Omega$ *synthesis mapping*.

$$\begin{aligned} \mathcal{A}_\Omega &: \vec{v} \xrightarrow{\Omega} \vec{\theta} \\ \mathcal{S}_\Omega &: \vec{\theta} \xrightarrow{\Omega} \vec{v}(\Omega) \end{aligned} \tag{1}$$

$\Omega$ denotes a model instance that is learned from a set of training samples. We suppose that a set of $M$ training samples, denoted by $M$ pairs $\{(\vec{v}_m, \vec{\theta}_m)|m = 1, .., M\}$, is given where a single labeled training sample is denoted by a pair of vectors $(\vec{v}_m, \vec{\theta}_m)$, $\vec{v}_m$ is the $m$-th vectorized facial image, and $\vec{\theta}_m = (\theta_{m1}, \theta_{m2}, \theta_{m3})$ are the corresponding 3D head angles of a face presented in $\vec{v}_m$.

An application of the analysis mapping can be considered as *pose estimation.* Given an arbitrary facial image $\vec{v} \notin \{\vec{v}_1, .., \vec{v}_M\}$, $\mathcal{A}_\Omega$ provides a 3D head angle estimate $\hat{\vec{\theta}} = \mathcal{A}_\Omega(\vec{v})$ of a face in $\vec{v}$. On the other hand, an application of the synthesis mapping provides a means of *pose transformation* or *facial animation.* Given an arbitrary 3D head angle $\vec{\theta} \notin \{\vec{\theta}_1, .., \vec{\theta}_M\}$, $\mathcal{S}_\Omega$ provides a synthesized sample or model view $\hat{\vec{v}} = \mathcal{S}_\Omega(\vec{\theta})$ whose head is rotated according to the given angle but its appearance is due to the learned model $\Omega$.

### 2.1.2 Personalized and Interpersonalized Models

The type of training samples used for learning a model determines the nature of specific PLS model instance. A model is called *personalized* when it is learned with pose-varying samples from a single individual. In this case, both analysis and synthesis mappings become specific to the person presented in the training set. Therefore the synthesis mapping output $\vec{v}(\Omega)$ exhibits personal appearance that solely depends on $\Omega$ encoding specificities of the person, while its head pose is given by an input to this mapping. On the other hand, a model is called *interpersonalized* when the training set contains multiple individuals. For pose estimation, this provides a more natural setting where the analysis mapping $\mathcal{A}_\Omega$ continuously covers not only the head pose variations but also variations over different individuals.

## 2.2 Pose-Insensitive Face Recognition

### 2.2.1 Model Matching

Given an arbitrary person's face as input, a learned PLS model can be fit against it by concatenating the analysis and synthesis mappings. We call this model matching process the *analysis-synthesis-chain* (ASC) process.

$$\mathcal{M}_\Omega : \vec{v} \xrightarrow{\Omega} \vec{\theta} \xrightarrow{\Omega} \vec{v}(\Omega) \tag{2}$$

The output of ASC is called the *model view* $\vec{v}(\Omega)$. It provides a facial view $\vec{v}(\Omega)$ of the person learned in $\Omega$ whose head pose is aligned to the input

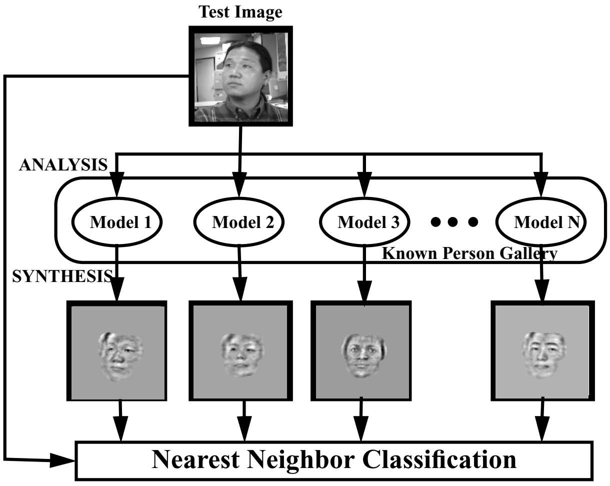


Figure 1: Pose-insensitive face recognition framework with parametric linear subspace models used to represent each known person.

face in $\vec{v}$. This process not only fits a learned model to the input but also gives simultaneously a 3D pose estimate as a byproduct that can be used for other application purposes.

Note that, when matching a personalized PLS model to different person's face, the resulting model view can be erroneous due to the pose estimation errors caused by the identity mismatch. To overcome this issue, $\mathcal{A}_\Omega$ of an interpersonalized model [26] can be exploited for reducing pose estimation error. For the purpose of face recognition, however, these errors actually serve as an advantage because it makes model views of mismatched individuals less similar to the input helping to single out the correct face. Moreover such errors are typically small due to the geometrical proximity of different faces.

### 2.2.2 Overview of the Proposed Recognition Framework

Figure 1 illustrates our framework for pose-insensitive face recognition. The framework employs the PLS model as the representation of a known person. For each known person, a personalized model is learned from the pose-varying samples of the person. We call a database of $P$ known people, as a set of learned personalized models $\{\Omega_p | p = 1, .., P\}$, the *known-person gallery*. Given a test image of an arbitrary person with an arbitrary head pose, each model in the gallery is matched against the image by using its ASC process. The process results in pose-aligned model views of all known persons. After this pose alignment, the test image is compared against the model views in a nearest neighbor classification fashion. In this scheme, the view-based comparison only occurs between views of the same pose improving the recognition performance.

Figure 2 illustrates the advantage of the proposed model-based method

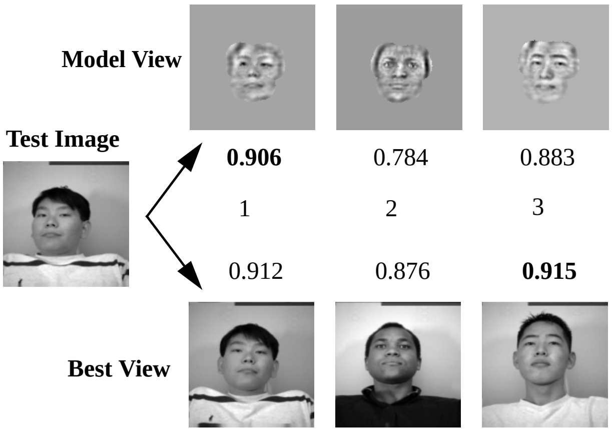


Figure 2: An illustrative example of face recognition with pose variation using the model-based and the example-based systems.

over an example-based multi-view method using a gallery of three known persons. A set of training samples for each person is used to construct the multi-view gallery entry. The top row displays model views of three learned models; the bottom row displays the best views of each known person that are most similar to the test image. Decimal numbers shown adjacent to the images denote their similarity to the test. There were no views in gallery whose head pose was the same as the test image shown in the left. Therefore head pose of the test and the best matched views are always different. This results in a mis-identification by the multi-view system. On the other hand, the model-based solution, constructed by using the exactly same samples as in the multi-view system, identifies the test image correctly. This is realized by the model's ability to generalize to unseen views, resulting in model views whose head pose is better aligned to the test.

The proposed framework flexibly aligns head pose of the inputs and model views at an arbitrary pose, exploiting the PLS's continuous generalization capability to unseen views. Figure 3 and Table 1 illustrate this advantage in comparison with two other recognition frameworks: the multi-view (MVS) and single-view (SVS) systems. Given facial images with arbitrary head poses shown in the first raw, a PLS, learned for this face, can provide model views whose head pose is well aligned to the inputs. MVS provides the most similar view (best view) to the input among the training samples used to learn the model, while SVS employs always the same frontal view that represents the person single-handedly. The figure shows that the proposed model is appeared to provide better pose-alignment than the two other systems. Table 1 shows actual facial similarity values between the inputs and the three different types of model or best view. The standard deviation shown in the right column indicates the degree of invariance against the pose variations by each framework. The parametric linear model provided the smallest standard deviation among the three, demonstrating

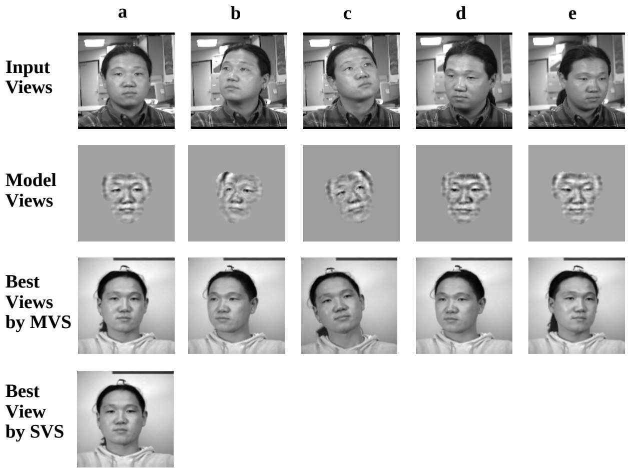


Figure 3: Comparison of three recognition frameworks in terms of pose-alignment ability. Model views shown in the second raw are given by the proposed method. MVS: example-based multi-view system; SVS: example-based single-view system. See texts for their description.

| **Model Type** | **a** | **b** | **c** | **d** | **e** | **std.dev.** |
|---|---|---|---|---|---|---|
| Model Views | 0.915 | 0.871 | 0.862 | 0.891 | 0.878 | 0.0184 |
| Best Views by MVS | 0.930 | 0.872 | 0.876 | 0.913 | 0.897 | 0.0220 |
| Best View by SVS | 0.926 | 0.852 | 0.816 | 0.878 | 0.862 | 0.0359 |

Table 1: Similarity scores between the test input and different types of model and best views in Figure 3.

the model's favorable characteristics towards pose-insensitivity.

## 3 Parametric Linear Subspace Model

This section describes two instances of the PLS model: the linear principal component-mapping (LPCMAP) model [27] and the parametric piecewise linear subspace (PPLS) model [29]. The PPLS model employs a set of LPCMAP models, each of which realizes the continuous analysis and synthesis mappings. For maintaining the continuous nature in a global system, we consider that local mapping functions cover the whole parameter space, without imposing a rigid parameter window. Due to the linearity, however, the range over which each local mapping is accurate is often limited. In order to cover a wide range of continuous pose variation, a PPLS model pieces together a number of local models distributed over the 3D angle space of head poses. In order to account for the local model's parameter-range limitation, each model is paired with a radius-basis weight function. The PPLS

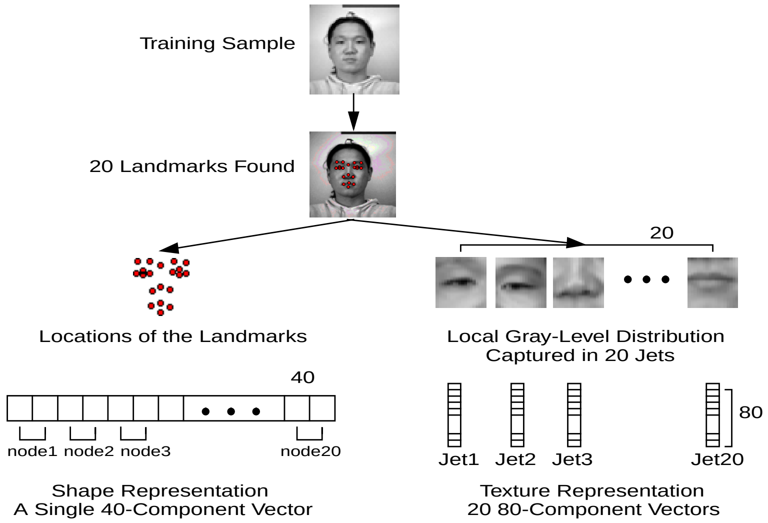


Figure 4: Shape and texture decomposition process, illustrating parameter settings used for our experiments in Section 5. The number of landmarks $N = 20$ and the length of a texture vector $L = 80$ with a bank of 5-level and 8-orientation 2D complex Gabor filters.

then performs a weighted linear combination of local model's outputs, realizing a continuous global function. The following introduces details of these models.

## 3.1 Linear PCMAP Model

The LPCMAP is a PLS model that realizes the continuous, but only locally valid, bidirectional mapping functions. It consists of a combination of two linear systems: 1) *linear subspaces* spanned by principal components (PCs) of training samples and 2) *linear transfer matrices*, which associate projection coefficients of training samples onto the subspaces and their corresponding 3D head angles. It linearly approximates the entire parameter space of head poses by a single model.

### 3.1.1 Shape-Texture Decomposition and Image Reconstruction

The LPCMAP model treats shape and texture information separately in order to utilize them for different purposes. It has also been shown in literature that combined feature of shape and shape-free texture improves recognition performance [38, 7, 21]. Figure 4 illustrates the process of decomposing shape and texture information in facial images. First, $N$ predefined landmarks are located in each facial image $\vec{v}_m$ by a landmark finder or other means. Using this location information, shape and texture representations $(\vec{x}_m, \{\vec{j}_{m,n}\})$ are extracted from the image. The shape representation $\vec{x}_m \in \mathbf{R}^{2N}$ stands for an array of object-centered 2D coordinates

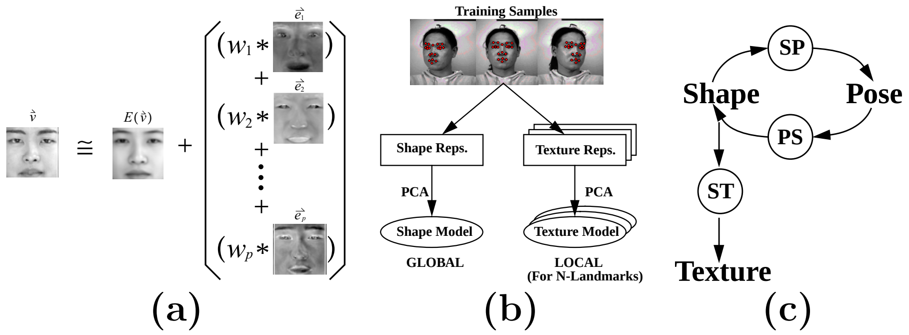


Figure 5: Learning processes of the LPCMAP model: (a) PCA subspace model by Sirovich and Kirby [36], (b) shape and texture models using linear subspaces, and (c) linear transfer matrices relating different model parameters A rectangle in (b) denotes a set of training samples and an ellipse denotes a PCA subspace model.

of the $N$ landmarks. On the other hand, the texture information is represented by a set of spatially sparse local features sampled at the $N$ landmark points. The multi-orientation and multi-scale Gabor wavelet transformation [8] is used to define such local features. The texture representation $\{\vec{j}_{m,n} \in \mathbf{R}^L | n = 1, .., N\}$ stands for a set of Gabor jets ($L$-component complex coefficient vector of the Gabor transform) sampled at the $N$ landmarks [19, 39, 28]. Let $\mathcal{D}_x$ and $\mathcal{D}_j$ denote operations of the shape and texture decomposition, respectively.

$$\vec{x}_m = \mathcal{D}_x(\vec{v}_m) \quad \vec{j}_{m,1}, .., \vec{j}_{m,N} = \mathcal{D}_j(\vec{v}_m) \tag{3}$$

As an inverse operation, a gray-level facial image can be reconstructed approximately from a pair of shape and texture representations $(\vec{x}_m, \{\vec{j}_{m,n}\})$, following the work by Poetzsch et al. [33]. $\mathcal{R}$ denotes this reconstruction operation.

$$\vec{v}_m = \mathcal{R}(\vec{x}_m, \vec{j}_{m,1}, .., \vec{j}_{m,N}) \tag{4}$$

### 3.1.2 Learning Linear Subspace Models

As the first step of the model learning process, we extract a small number of significant statistical modes from training facial images using Principal Component Analysis (PCA), as illustrated in Figure 5. Given training samples $\{(\vec{v}_m, \vec{\theta}_m) | m = 1, .., M\}$, a set of extracted shape representations $\{\vec{x}_m\}$ is subjected to PCA [34], solving the eigen decomposition problem of the centered sample covariance matrix, $XX^t\vec{y}_p = \lambda_p\vec{y}_p$, where $X$ is a $2N \times M$ column sample matrix. This results in an ordered set of $2N$ principal components (PCs) $\{\vec{y}_p | p = 1, .., 2N\}$ of the shape ensemble. We call such PCs *shape PCs*. The local texture set $\{\vec{j}_{m,n}\}$ at a landmark $n$ is also subjected to PCA, resulting in an ordered set of $L$ PCs $\{\vec{b}_{s,n} | s = 1, .., L\}$. We cal such

PCs *texture PCs*. Performing this procedure for all the $N$ landmarks results in a set of local texture PCs $\{\vec{b}_{s,n} | s = 1, .., L; n = 1, .., N\}$.

The subspace model [36, 31] is defined by a vector space spanned by a subset of the PCs in decreasing order of their corresponding variances as illustrated in Figure 5(a). An image $\vec{v}$ is approximated as the sum of the average image $E(\vec{v})$ and the PCs $(\vec{e}_1, .., \vec{e}_p)$. The weight vector $(w_1, .., w_p)$ is defined by orthogonal projection onto the subspace and serves as a compact representation of the image $\vec{v}$. Due to the orthonormality of PCs, a linear combination of the PCs with the above mixing weights provides the best approximation of an original representation which minimizes $L_2$ reconstruction error.

As illustrated in Figure 5(b), a *shape model* $Y$ is constructed by the first $P_0 \leq 2N$ shape PCs, $Y = (\vec{y}_1, .., \vec{y}_{P_0})^t$. And a *texture model* $\{B_n\}$ is then constructed by the first $S_0 \leq L$ texture PCs at each landmark $n$, $\{B_n = (\vec{b}_{1,n}, .., \vec{b}_{S_0,n})^t | n = 1, .., N\}$. These subspace models are used to parameterize and compress a centered input representation by orthogonally projecting it onto the subspace.

$$\vec{q}_m = Y(\vec{x}_m - \vec{u}_x) \quad \vec{u}_x = \frac{1}{M}\sum_{m=1}^{M} \vec{x}_m \tag{5}$$

$$\vec{r}_{m,n} = B_n(\vec{j}_{m,n} - \vec{u}_{jn}) \quad \vec{u}_{jn} = \frac{1}{M}\sum_{m=1}^{M} \vec{j}_{m,n} \tag{6}$$

We call projection coefficient vectors for shape representation $\vec{q}_m \in \mathbf{R}^{P_0}$ *shape parameters* and those of texture representation $\vec{r}_{m,n} \in \mathbf{R}^{S_0}$ *texture parameters*, respectively. We also refer to these parameters (equivalent to the weight vector in Figure 5(a)) as *model parameters* collectively.

$$\vec{x}_m \approx \vec{u}_x + Y^t \vec{q}_m \tag{7}$$

$$\vec{j}_{m,n} \approx \vec{u}_{jn} + (B_n)^t \vec{r}_{m,n} \tag{8}$$

### 3.1.3 Learning Linear Transfer Matrices

As the second step of the learning process, model parameters are linearly associated with head pose parameters for realizing direct mappings between $\vec{v}$ and $\vec{\theta}$, as illustrated in Figure 5(c).

Clearly, the model parameters are non-linearly related to the 3D head angles therefore the intrinsic mapping between them is non-linear. In order to linearly approximate such non-linear mapping, we first transform the 3D head angles $\vec{\theta}_m = (\theta_{m,1}, \theta_{m,2}, \theta_{m,3})$ to *pose parameters* $\vec{\varphi}_m \in \mathbf{R}^{T \geq 3}$ such that the mapping between the pose and model parameters can be linearly approximated. We consider the following trigonometric function $\mathcal{K}$ for this purpose.

$$\begin{array}{c}\vec{\varphi}_m = \mathcal{K}(\vec{\theta}_m) = (\cos(\tilde{\theta}_{m,1}), \sin(\tilde{\theta}_{m,1}), \cos(\tilde{\theta}_{m,2}), \sin(\tilde{\theta}_{m,2}), \cos(\tilde{\theta}_{m,3}), \sin(\tilde{\theta}_{m,3})) \\ \tilde{\theta}_{m,i} = \theta_{m,i} - u_{\theta i} \quad \vec{u}_\theta = (u_{\theta 1}, u_{\theta 2}, u_{\theta 3}) = \frac{1}{M}\sum_{m=1}^{M} \vec{\theta}_m \end{array} \tag{9}$$

There exists an inverse transformation $\mathcal{K}^{-1}$ such that

$$\vec{\theta}_m = \mathcal{K}^{-1}(\vec{\varphi}_m) = \vec{u}_\theta + (\arctan(\frac{\varphi_{m,2}}{\varphi_{m,1}}), \arctan(\frac{\varphi_{m,4}}{\varphi_{m,3}}), \arctan(\frac{\varphi_{m,6}}{\varphi_{m,5}})) \tag{10}$$

For both the analysis and synthesis mappings, the pose parameters $\vec{\varphi}_m$ are linearly related only with the shape parameters $\vec{q}_m$.

$$\vec{\varphi}_m = F\vec{q}_m \tag{11}$$

$$\vec{q}_m = G\vec{\varphi}_m \tag{12}$$

A $T \times P_0$ transfer matrix $F$ (denoted as **SP** in Figure 5(c)) is learned by solving an overcomplete set of linear equations, $FQ = \Phi$, $Q = (\vec{q}_1, .., \vec{q}_M)$, $\Phi = (\vec{\varphi}_1, .., \vec{\varphi}_M)$. The Singular Value Decomposition (SVD) [34] is used to solve this linear system. Moreover, a $P_0 \times T$ transfer matrix $G$ (denoted as **PS** in Figure 5(c)) is also learned by solving, $G\Phi = Q$, in the same manner. For the synthesis mapping, the shape parameters $\vec{q}_m$ are linearly related with the texture parameters $\vec{r}_{m,n}$ at each landmark $n$.

$$\{\vec{r}_{m,n} = H_n\vec{q}_m | n = 1, .., N\} \tag{13}$$

A set of $S_0 \times P_0$ transfer matrices $\{H_n\}$ (denoted as **ST** in Figure 5(c)) is learned by solving, $H_nQ = R_n, R_n = (\vec{r}_{1,n}, .., \vec{r}_{M,n})$, by using SVD for all the $N$ landmarks.

### 3.1.4 Model Definition

As a result of the above two learning steps, we generate a set of data entities which collectively capture facial appearance in a given set of training samples. A LPCMAP model $LM$ is defined by such data entities that are learned from training samples

$$LM := \{\vec{u}_x, \{\vec{u}_{jn}\}, \vec{u}_\theta, Y, \{B_n\}, F, G, \{H_n\}\} \tag{14}$$

where $\vec{u}_x$ and $\{\vec{u}_{jn}\}$ are average shape and texture representations, $\vec{u}_\theta$ is an average 3D head angle vector, $Y$ and $\{B_n\}$ are shape and texture models, $F$ and $G$ and $\{H_n\}$ are shape-to-pose, pose-to-shape, and shape-to-texture transfer matrices.

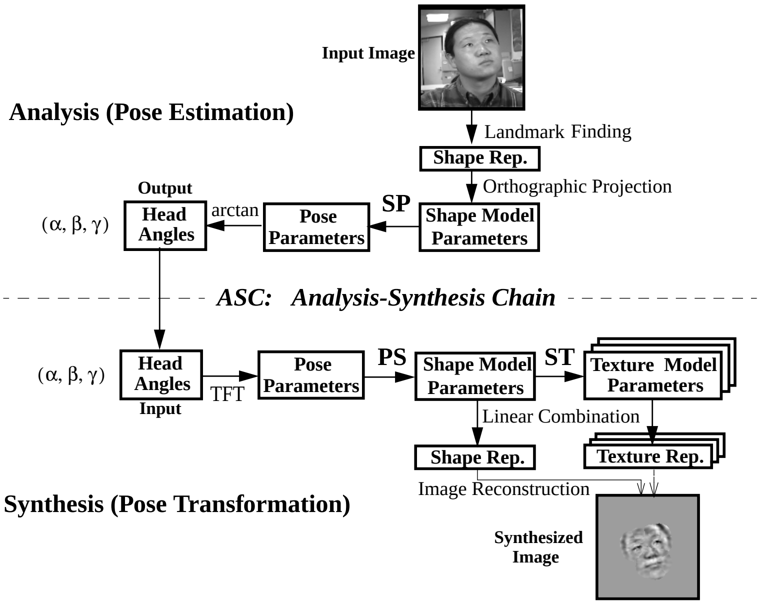


Figure 6: Analysis and synthesis mapping and analysis-thesis-chain functions. Trigonometric transfer functions $\mathcal{K}$ and $\mathcal{K}^{-1}$ are denoted by **TFT** and **arctan**, respectively. **SP**, **PS** and **ST** denote the transfer matrices shown in Figure 5(c).

### 3.1.5 Mapping and Chain Functions

The analysis and synthesis mappings are constructed as a function of the learned LPCMAP model $LM$, as illustrated in Figure 6. The analysis mapping function $\mathcal{A}_{LM}(\vec{v})$ is given by combining formulae (3), (5), (11), and (10).

$$\hat{\vec{\theta}} = \mathcal{A}_{LM}(\vec{v}) = \vec{u}_\theta + \mathcal{K}^{-1}(F \cdot Y \cdot (\mathcal{D}_x(\vec{v}) - \vec{u}_x)) \qquad (15)$$

The analysis function only utilizes the shape information of faces, following results of our preliminary experiments in which the head angles are better correlated with the shape representations than the texture representations [26]).

The synthesis mapping function $\mathcal{S}(\vec{\theta})$ is given by relating the 3D head angles to the shape coefficients and the shape coefficients to the texture coefficients, Because the shape and texture decomposition, we address distinct synthesis processes for shape and texture. We refer to shape and texture synthesis mapping functions as $\mathcal{SS}$ and $\mathcal{TS}$, respectively.

The shape synthesis mapping function $\mathcal{SS}_{LM}(\vec{\theta})$ is given by combining formulae (9), (12), and (7), using only the shape information similar to the analysis function. On the other hand, the texture synthesis mapping function $\mathcal{TS}_{LM}(\vec{\theta})$ is given by formulae (9), (12), (13), and (8), utilizing correlation between shape and texture parameters. The synthesis mapping function $\mathcal{S}_{LM}(\vec{\theta})$ is then given by substituting the shape and texture syn-

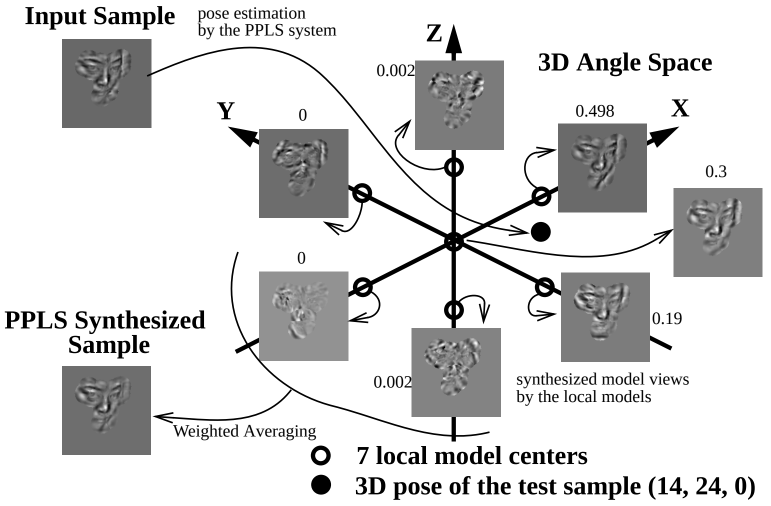


Figure 7: A sketch of the PPLS model with seven LPCMAP models. An input image is shown at the top-left. Model centers of the LPCMAPs are denoted by circles. Pose estimation is performed by applying the analysis mapping $\mathcal{A}_{PM}$, resulting the global estimate denoted by a block dot. On the other hand, pose transformation is performed by applying the synthesis mapping $\mathcal{S}_{PM}$. Model views, shown next to the model centers, are linearly combined with Gaussian weights, resulting in global synthesis shown at the bottom-left.

thesis functions to formula (4).

$$\begin{aligned} \hat{\vec{v}} = \mathcal{S}_{LM}(\vec{\theta}) = \mathcal{R}(\mathcal{SS}_{LM}(\vec{\theta}), \mathcal{TS}_{LM}(\vec{\theta})) \\ \hat{\vec{x}} = \mathcal{SS}_{LM}(\vec{\theta}) = \vec{u}_x + Y^t \cdot G \cdot \mathcal{K}(\vec{\theta} - \vec{u}_\theta) \\ \{\hat{\vec{j}}_n | n = 1, .., N\} = \mathcal{TS}_{LM}(\vec{\theta}) = \{\vec{u}_{jn} + B_n \cdot H_n \cdot G \cdot \mathcal{K}(\vec{\theta} - \vec{u}_\theta) | n = 1, .., N\} \end{aligned} \tag{16}$$

Finally, the ASC function $\mathcal{M}(\vec{v})$ is given by concatenating Eq. (15) and Eq. (16) as shown in Figure 6.

$$\hat{\vec{v}} = \mathcal{M}_{LM}(\vec{v}) = \mathcal{R}(\mathcal{SS}_{LM}(\mathcal{A}_{LM}(\vec{v})), \mathcal{TS}_{LM}(\mathcal{A}_{LM}(\vec{v}))) \tag{17}$$

## 3.2 Parametric Piecewise Linear Subspace Model

### 3.2.1 Model Definition

The parametric piecewise linear subspace (PPLS) model [29] extends the LPCMAP model by using the piecewise linear approach [35]. Due to the linear approximation, LPCMAP model can only be accurate within a limited range of pose parameters. Piecewise linear approach approximates the nonlinear pose variation within a wider range by piecing together a number of locally valid models distributed over the pose parameter space. The PPLS model $PM$ consists of a set of $K$ local models in the form of the above-described LPCMAP model.

$$PM := \{LM_k | k = 1, .., K\} \tag{18}$$

We assume that the local models are learned by training data sampled from appropriately distanced local regions of the *3D angle space*: the 3D finite parameter space spanned by the head angles. Each set of the local training samples is associated with a *model center*, the average 3D head angles $\vec{u}_\theta^{LM_k}$, which specifies the learned model's location in the 3D angle space. Figure 7 illustrates seven local models distributed in the 3D angle space. Model centers are denoted by circles and model views of the input are also shown next to them. Missing components of shape representations due to large head rotations are handled by the *mean-imputation method* [22], which fills in each missing component by a mean computed from all available data at the component dimension.

### 3.2.2 Mapping and Chain Functions

The analysis mapping function $\mathcal{A}_{PM}$ of the PPLS model is given by averaging $K$ local pose estimates with appropriate weights as illustrated in Figure 7.

$$\hat{\vec{\theta}} = \mathcal{A}_{PM}(\vec{v}) = \sum_{k=1}^{K} w_k \mathcal{A}_{LM_k}(\vec{v}) \tag{19}$$

Similarly, the synthesis mapping function $\mathcal{S}_{PM}$ is given by averaging $K$ locally synthesized shape and texture estimates with the same weights as illustrated in Figure 7.

$$\begin{array}{r} \hat{\vec{v}} = \mathcal{S}_{PM}(\vec{\theta}) = \mathcal{R}(\mathcal{SS}_{PM}(\vec{\theta}), \mathcal{TS}_{PM}(\vec{\theta})) \\ \hat{\vec{x}} = \mathcal{SS}_{PM}(\vec{\theta}) = \sum_{k=1}^{K} w_k \mathcal{SS}_{LM_k}(\vec{\theta}) \\ \{\hat{\vec{j}}_n\} = \mathcal{TS}_{PM}(\vec{\theta}) = \sum_{k=1}^{K} w_k \mathcal{TS}_{LM_k}(\vec{\theta}) \end{array} \tag{20}$$

A vector of the weights $\vec{w} = (w_1, .., w_K)$ in Eq. (19) and Eq. (20) must be responsible for localizing the output space of the LPCMAP models, since their outputs themselves are continuous and unbounded. For this purpose, we defined the weights, as a function of the input pose, by using a normalized Gaussian function of distance between an input pose and each model center

$$w_k(\vec{\theta}) = \frac{\rho_k(\vec{\theta} - \vec{u}_\theta^{LM_k})}{\sum_{k=1}^{K} \rho_k(\vec{\theta} - \vec{u}_\theta^{LM_k})} \quad \rho_k(\vec{\theta}) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{\|\vec{\theta}\|^2}{2\sigma_k^2}) \tag{21}$$

where $\sigma_k$ denotes the $k$-th Gaussian width. It is set by the standard deviation of the 3D head angle vectors used for learning $LM_k$ and determines the extent to which each local model influences the outputs $\hat{\vec{\theta}}$ and $\hat{\vec{v}}$. The weight value reaches maximum when the input pose coincides with one of the model centers; it decays as the distance increases. Outputs of local models that are located far from an input pose can become largely distorted because of the pose range limitation. However, these distorted local outputs do not greatly influence a global output because their contribution is strongly inhibited by relatively low weight values.

The ASC function $\mathcal{M}(\vec{v})$ is again given by connecting an analysis output to a synthesis input.

$$\hat{\vec{v}} = \mathcal{M}_{PM}(\vec{v}) = \mathcal{R}(\mathcal{SS}_{PM}(\mathcal{A}_{PM}(\vec{v})), \mathcal{TS}_{PM}(\mathcal{A}_{PM}(\vec{v}))) \tag{22}$$

### 3.2.3 Gradient Descend-based Pose Estimation

Note that Eq. (19) cannot be solved in closed-form because its r.h.s. include the weights as a function of an unknown $\vec{\theta}$. To overcome this issue, a gradient descent-based iterative solution is formulated. Let a shape vector $\vec{x}$ be an input to the algorithm. Also let $\vec{x}_i$ and $\vec{\theta}_i$ denote the shape and angle estimates by the $i$-th iteration. The algorithm iterates the following formulae until the mean-square error $\|\Delta\vec{x}_i\|^2$ becomes sufficiently small.

$$\begin{array}{c} \Delta\vec{x}_i = \vec{x} - \vec{x}_i, \\ \Delta\vec{\theta}_i = \sum_{k=1}^{K} w_k(\vec{\theta}_i)\mathcal{A}'_{LM_k}(\Delta\vec{x}_i), \\ \vec{\theta}_{i+1} = \vec{\theta}_i + \eta\Delta\vec{\theta}_i, \\ \vec{x}_{i+1} = \sum_{k=1}^{K} w_k(\vec{\theta}_{i+1})\mathcal{SS}_{LM_k}(\vec{\theta}_{i+1}), \end{array} \tag{23}$$

where $\eta$ is a learning rate and $\mathcal{A}'$ is a slight modification of Eq. (15) that has a shape vector interface. The initial conditions $\vec{x}_0$ and $\vec{\theta}_0$ are given by the local model whose center shape $\vec{u}_x^{LM_k}$ is most similar to $\vec{x}$.

Note that the weighted sum of the analysis mappings in (23) is used as an approximation of the gradient of $\vec{\theta}$ with respect to $\vec{x}$ at the current shape estimate $\vec{x}_i$. In the PPLS model, such gradients are only available at the $K$ discrete model centers. The second formula in (23), therefore, interpolates the $K$ local gradient matrices for computing the gradients at an arbitrary point in the 3D angle space. The good local accuracy of the LPCMAP model shown in [27] supports the validity of this approximation. When a sufficient number of local models are allocated in the 3D angle space, the chance of being trapped at a local minimum should decrease. In our experimental setting described in the next section, the above initial condition settings resulted in no local minimum trappings significantly away from the global minima. Note also that the algorithm performs pose estimation and shape synthesis simultaneously since it iterates between pose and shape in each loop. This gives an alternative for the shape synthesis, although the global synthesis mapping in (20) remains valid.

# 4 Interpersonalized Pose Estimation

As mentioned in Section 2.1.2, a single system should be able to solve the pose estimation task across different individuals by exploiting geometrical proximity of faces. In the previous sections, we focused on how to model pose variations by using an example of personalized PLS models. This section discusses how to extend the PLS framework to capture variations due to head pose and individual differences simultaneously. The resulting *interpersonalized model* is applied to realize pose estimation across different people.

There are two approaches for realizing such an interpersonalized PLS model. The first is simply to train a LPCMAP or PPLS model by using a set of training samples that contain different-pose views from multiple people. The generic design of the proposed PLS models allows this straightforward extension however we must empirically validate if the learned linear model adequately capture both variations correctly. Numerically after learning, both $LM$ and $PM$ can be used in the same manner described in Section 3 for exploiting the corresponding analysis synthesis mappings and ASC model matching. We refer to this type of model as *single-PLS model.*

The second, on the other hand, is to linearly combine a set of personalized models in the similar way we constructed PPLS using a set of LPCMAPs. We refer to this type of model as *multiple-PLS model.* A multiple-PPLS model $MM$ consists of a set of $P$ personalized models in the form of PPLS.

$$MM := \{PM_p | p = 1, .., P\} \tag{24}$$

We assume that each PPLS model is personalized by learning it with pose-varying samples of a specific person and that the training samples cover an adequate range of head poses in the 3D angle space. The analysis mapping function $\mathcal{A}_{MM}$ of the multiple-PPLS model is then defined by a weighted linear combination of $P$ pose estimates by the personalized models, realizing an interpersonalized pose estimation.

$$\hat{\vec{\theta}} = \mathcal{A}_{MM}(\vec{v}) = \sum_{p=1}^{P} w_p \mathcal{A}_{PM_p}(\vec{v}) \tag{25}$$

The weight vector $\vec{w} = (w_1, .., w_P)$ is responsible for choosing appropriate personalized models and ignoring models that encodes faces very different from input so as to minimize pose estimation errors. We consider an error of shape reconstruction $err_p(\vec{x})$ by using a shape-only analysis-synthesis-chain of $p$-th PPLS model. Then we let a normalized Gaussian function of such errors, similar to (21), indicate fidelity of personalized models to arbitrary

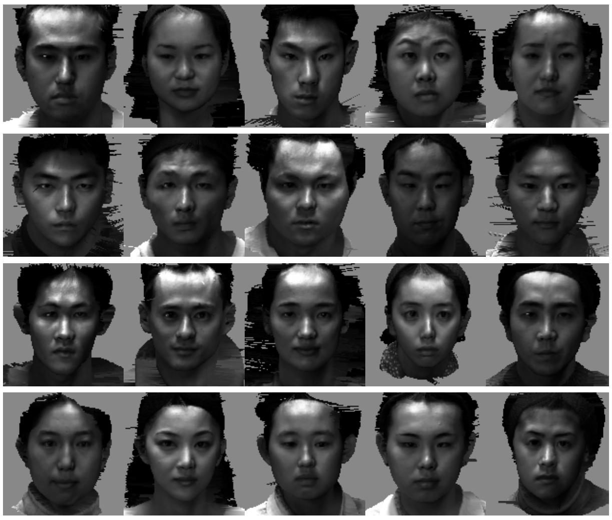


Figure 8: 20 frontal views rendered from the 3D face models.

inputs.

$$\begin{aligned} w_p(\vec{\theta}) &= \frac{\rho_p(err_p(\vec{x}))}{\sum_{p=1}^{P} \rho_p(err_p(\vec{x}))} \\ err_p(\vec{x}) &= \vec{x} - \hat{\vec{x}}_p = \vec{x} - \mathcal{SS}_{PM_p}(\mathcal{A}_{PM_p}(\vec{x})) \\ \rho_p(\vec{\theta}) &= \frac{1}{\sqrt{2\pi}\sigma_p} \exp(-\frac{\|err_p(\vec{x})\|^2}{2\sigma_p^2}) \end{aligned} \tag{26}$$

where a shape synthesis mapping $\mathcal{SS}_{PM_p}$ of the multiple-PPLS model is defined similar to (25) and $\sigma_p$ denotes the $p$-th Gaussian width.

# 5 Experiments

## 5.1 Data Set

For evaluating our system's performance over various head poses, we must collect a very large number of samples with controlled head poses, which is not an easy task. For mitigating this difficulty, we use 3D face models pre-recorded by a Cyberware scanner. Given such data, relatively faithful image samples with an arbitrary, but precise, head pose can easily be created by image rendering. We used 20 heads randomly picked from the ATR-Database [17], as shown in Figure 8.

For each head, we created 2821 training samples. They consist of 7 local sample sets each of which covers a pose range of ±15 degrees at one-degree interval. These local sets are distributed over the 3D angle space such that they collectively cover a pose range of ±55 degrees along each axis of 3D rotations; their model centers are distanced by ±40 degrees from the frontal pose (origin of the angle space). We also created 804 test samples for each head. In order to test the model's generalization capability to unknown head poses, we prepared the test samples whose head poses were not included in the training samples. Head angles of some test samples were in-between multiple local models and beyond their ±15 degree range. They cover a pose range of ±50 degrees. For more details of the data, see our previous reports [26, 29].

For each sample, the 2D locations of 20 landmarks of inner facial parts, such as eyes, nose and mouth, are derived by rotating the 3D landmark coordinates, initialized manually, and by projecting them onto an image plane. The explicit rotation angles of the heads also provide 3D head angles of the samples. The rendering system provides the self-occlusion information. Up to 10% of the total landmarks were self-occluded for each head.

## 5.2 Personalized Pose Estimation and View Synthesis

We compare the PPLS and LPCMAP models learned using the same training samples described above. The resulting PPLS model consists of 7 local linear models, each of which is learned from one of the local training sets. On the other hand, the resulting LPCMAP model consists of a single model learned from the total 2821 samples. The shape and texture representation are extracted using the specification $N = 20$ and $L = 80$ described in Figure 4. The PPLS model uses $\sigma_k$ set to the sample standard deviation and the gradient descent-based system with 500 iterations and $\eta$ set to 0.01. The learned models are tested with both 2821 training samples themselves and 804 disjoint test samples with unknown poses. We refer to the former by *accuracy test* and the latter by *generalization test*.

Figure 9(a) compares average pose estimation errors of the PPLS and LPCMAP models in both accuracy and generalization tests. In the accuracy test, the average angular error with the first 8 PCs was $0.8 \pm 0.6$ and $3.0 \pm 2.4$ degrees and the worst error was 5.6 and 18.9 degrees for the PPLS and LPCMAP models, respectively. In the generalization test, the average error was $0.9 \pm 0.6$ and $2.4 \pm 1.4$ degrees, and the worst error was 4.5 and 10.2 degrees for the two models. Figure 9(b) compares average shape synthesis errors of the two models in the two test cases. In the accuracy test, the average landmark position error with the first 8 PCs was $0.8 \pm 0.4$ and $2.2 \pm 1.2$ pixels, and the worst error was 3.0 and 7.6 pixels for the PPLS and LPCMAP models, respectively. In the generalization test, the average error was $0.9 \pm 0.4$ and $2.4 \pm 0.7$ pixels, and the worst error was 2.7 and 5.6 pixels for the two models. Figure 9(c) compares average similarities of synthesized and ground-truth texture vectors for the two models in the two test cases. Local texture similarity is computed as a normalized dot-product (cosine) of Gabor jet magnitudes, $JetSim := \frac{amp(\vec{j}_n^m) \cdot amp(\hat{\vec{j}}_n^m)}{\|amp(\vec{j}_n^m)\| \, \|amp(\hat{\vec{j}}_n^m)\|}$, where *amp* extracts magnitudes of a Gabor jet in polar coordinates. The similarity values range from 0 to 1, where 1 denotes equality of two jets. In the accuracy test, the average similarity with the first 20 texture PCs was $0.955 \pm 0.03$ and $0.91 \pm 0.04$, and the worst similarity was 0.81 and 0.73 for the PPLS and LPCMAP models, respectively. In the generalization test, the

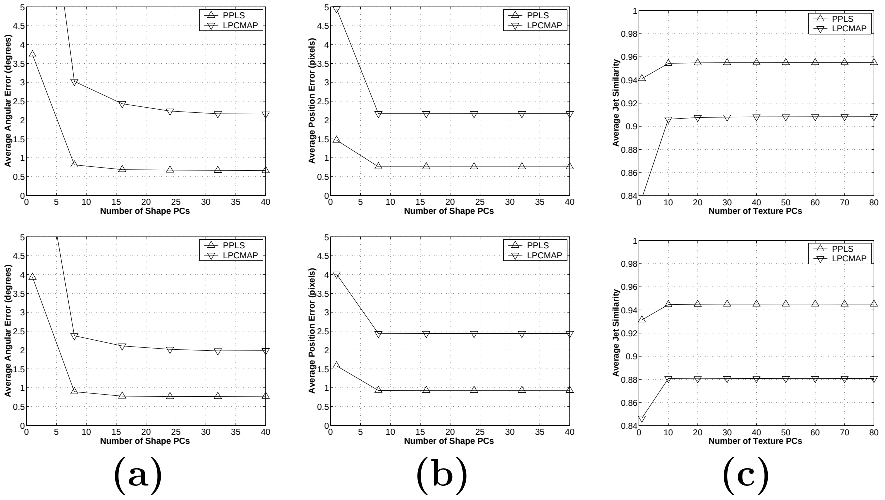


Figure 9: Comparison of the PPLS and LPCMAP models in terms of pose estimation and transformation errors. The first and second rows show results of the accuracy and generalization tests, respectively. Errors (similarities) are plotted over the number of PCs used to construct a subspace. (a) pose estimation errors in degrees averaged over 3 rotation angles. (b) shape synthesis errors in pixels averaged over 20 landmarks. (c) texture synthesis error by Gabor jet similarity averaged over 20 landmarks.

average similarity was $0.945 \pm 0.03$ and $0.88 \pm 0.03$, and the worst similarity was 0.82 and 0.77 for the two models.

For all three tasks, the PPLS model greatly improved performance over the LPCMAP model in both test cases, resulting in *sub-degree* and *sub-pixel* accuracy. The results also show that the average errors between the two test cases were similar, indicating good generalization to unknown poses. As a reference for our texture similarity analysis, we computed average texture similarities over 450 people from the FERET database [32]. The average similarity was $0.94 \pm 0.03$ for the same person pairs and $0.86 \pm 0.02$ for the most similar, but different, person pairs. The average similarity of the PPLS model was higher than that of the large FERET database, which validates the results of our texture similarity analysis.

Figure 10 illustrates model views: images reconstructed from samples synthesized by formula (20) of the PPLS model. Note that facial images reconstructed by the Pötzsch algorithm [33] do not retain original picture quality. This is because a transformation $\mathcal{D}_j$ from images to the Gabor jet representations is lossy due to coarse sampling in both spatial and frequency domains. Nonetheless, these images still capture characteristics of faces fairly well. Figure 10(a) compares reconstructed images of original and synthesized training samples. The left-most column shows frontal views while the rest of columns show views with ±45 degree rotation along one axis. Figure 10(b) compares original and synthesized test samples. For

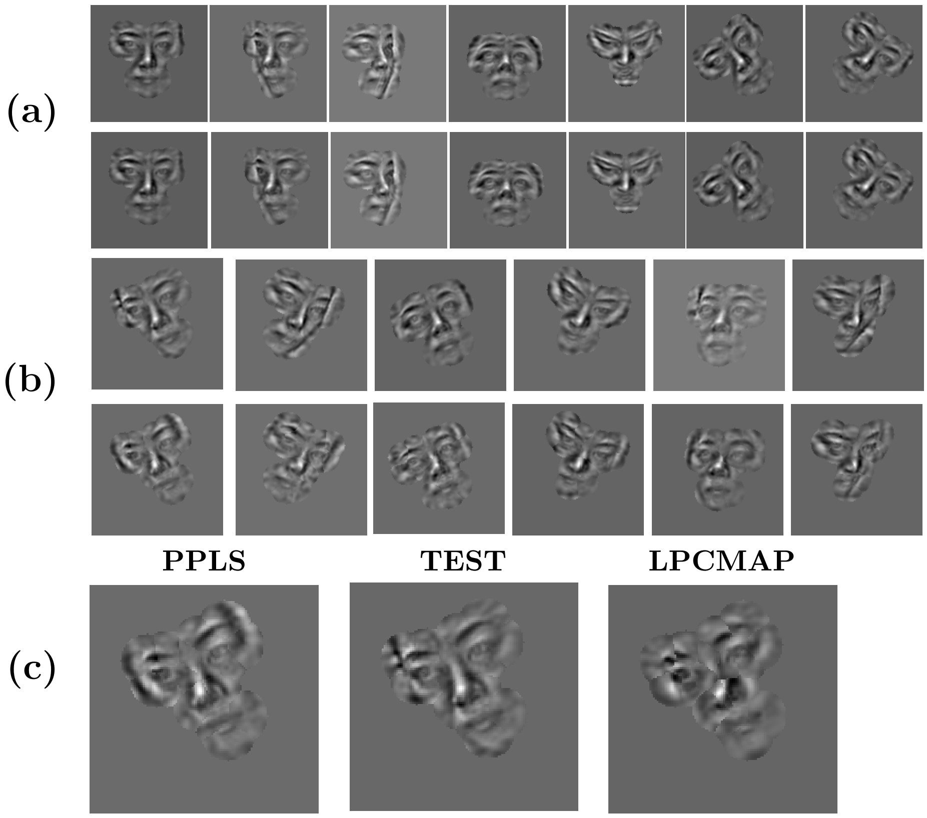


Figure 10: Examples of synthesized model views by the PPLS model. In (a) and (b), model views in the first and second rows are reconstructed from ground-truth and synthesized pose-aligned samples, respectively. (a): training samples with known head pose (accuracy test case); (b): test samples with unknown head poses (generalization test case); (c): illustrative comparison of model views synthesized by the PPLS and LPCMAP models.

all three cases, the original and synthesized model views were very similar, indicating good accuracy and successful generalization to unknown head poses. Figure 10(c) compares model views synthesized by the PPLS and LPCMAP models. The PPLS's model view was more similar to the original than the LPCMAP's model view. This agrees with the results of our error and similarity analyses.

### 5.3 Pose-Insensitive Face Recognition

For comparison, we constructed four recognition systems with 20 known persons: 1) the single-view system (SVS), which represents each known person by a single frontal view, 2) the LPCMAP system with a gallery of LPCMAP models, 3) the PPLS system with a gallery of PPLS models, and 4) the multi-view system (MVS), which represents each person by various raw views of the person. The LPCMAP, PPLS and MVS are constructed

| Test Samples | Identification | Compression |
|---|---|---|
| **SVS** | 59.9±10.6% | 0.035% |
| **LPCMAP** | 91.6±5.0% | 0.74% |
| **PPLS** | 98.7±1.0% | 5% |
| **MVS** | 99.9±0.2% | — |

Table 2: Average correct-identification and relative compression rates for four different systems.

|  | **PPLS** | **LPCMAP** |
|---|---|---|
| **Unknown:** $\mathcal{M}(\vec{v})$ | 98.7±1.0% | 91.6±5.0% |
| **Known:** $\mathcal{S}(\vec{\theta})$ | 99.3±0.7% | 92.4±4.0% |

Table 3: Identification rates when head pose of tests is unknown or given as ground-truth.

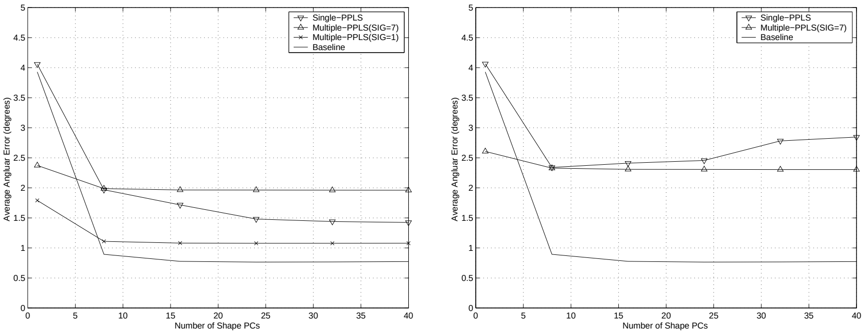
by using the same 2821 training samples per person; the SVS serves as a base-line. For both models, $P_0$ and $S_0$ are set to 8 and 20, respectively. The PPLS models consist of 7 local models and perform 500 iterations with $\eta$ set to 0.01 for each test sample. Each pair of views are compared by an average of normalized dot-product similarities between the corresponding Gabor jet's magnitudes.

Table 2 summarizes the results of our recognition experiments. Identification rates in the table are averaged over the 20 persons; the compression rates represent the size of the known-person gallery in percentage relative to the MVS. The results show that recognition performance of the PPLS system was more robust than the LPCMAP system (7% higher rate). Performance of our model-based systems was much better than the base-line SVS. Identification rates of the PPLS and MVS were almost the same while the former compressed the data by a factor of 20.

In some application scenarios, head pose information can be independently measured by the other means prior to identification. In such a case, the proposed recognition system can be realized by using only the synthesis mapping instead of model matching. Table 3 compares average identification rates of the two cases: with and without the knowledge of head poses. The results show that the knowledge of head poses gave a slight increase in recognition performance, however the increase was minimal.

### 5.4 Interpersonalized Pose Estimation

For comparison, we test both single-PPLS and multiple-PPLS models for two test cases: interpolation and extrapolation tests. We use the same data



**Interpolation Test** **Extrapolation Test**

Figure 11: Comparison of the single-PPLS and multiple-PPLS models for the interpolation and extrapolation tests in terms of interpersonalized pose estimation errors. Baseline plots indicate average pose estimation errors by the personalized models shown in Figure 9 for reference.

described in Section 5.1. For the interpolation (known persons) test, both models are trained with all the 56420 training samples (20 people × 2821 samples). A single-PPLS model with 7 LPCMAPs is trained with all the samples. On the other hand, a multiple-PPLS model is build by training each personalized model with 2821 samples for a specific person. These two models are then tested with the same 16080 test samples from the 20 individuals. For the extrapolation (unknown persons) test, each model is trained with 53599 training samples of 19 individuals, excluding training samples referring to the person being tested. This assure that the model does not contain knowledge of testing faces. The two models are trained in the same way as the interpolation test and tested with the same 16080 test samples. The same parameter settings of LPCMAP and PPLS models are used as described in Section 5.2.

Figure 11 compares the single-PPLS model and multiple-PPLS model in the two test setting. Down-triangles denote the average angular errors of the single-PPLS model and up-triangles denote those of the multiple-PPLS model with $\forall p \ \sigma_p = 7$. As reference, average pose estimation errors of the personalized model shown in Figure 9 are also included and denoted by solid lines without markers. Errors are plotted against 6 different sizes of the shape model. Our pilot study indicated that $\forall p \ \sigma_p = 7$ is optimal for both interpolation and extrapolation cases. However $\forall p \ \sigma_p = 1$ was optimal when only interpolation test was considered. For this reason, errors with $\sigma_p = 1$ is also included for the interpolation test.

When $\sigma_p$ is set optimally for both test cases, the average errors of the two models were very similar between two test cases. With the first 8 shape PCs, the errors of the two models were the same: 2.0 and 2.3 degrees for the interpolation and extrapolation tests, respectively. For the interpolation test, standard deviation of the errors and the worst error were 0.9 and 5.5

degrees for the single-PPLS model and 0.8 and 5.1 degrees for the multiple-PPLS model. For the extrapolation test, the standard deviation and the worst error were 0.9 and 5.9 degrees for the former and 0.9 and 5.5 degrees for the latter. For both tests, the average errors of the two models are roughly 1 to 1.5 degrees larger than the baseline errors. When $\sigma_p$ is set optimally for the interpolation condition, the multiple-PPLS model clearly outperformed the single-PPLS model, improving the average errors by roughly 1 degree and becoming similar to the baseline result (only 0.2 degree difference). These experimental results indicate that both models are fairly accurate, indicating the feasibility of the proposed approach to generalize over different persons.

# 6 Conclusion

This article presents a general statistical framework for modeling and processing head pose information in 2D grayscale images: analyzing, synthesizing, and identifying facial images with arbitrary 3D head poses. Three types of PLS model are introduced. The LPCMAP model offers a compact view-based model with bidirectional analysis and synthesis mapping functions. A learned model can be matched against an arbitrary input by using an analysis-synthesis chain function that concatenates the two. The PPLS model extends the LPCMAP for covering a wider pose range by piecing together a set of local models. Similarly the multiple-PPLS model extends the PPLS for generalizing over different people by linearly combining a set of PPLSs. A novel pose-insensitive face recognition framework is proposed by using the PPLS model to represent each known person. Our experimental results of 20 people, covering a wide range of ±50 degree 3D rotation, demonstrated the proposed model's accuracy for solving pose estimation and pose animation tasks and robustness for generalizing to unseen head poses and individuals while compressing the data by a factor of 20 and more.

The proposed framework was evaluated by using accurate landmark locations and corresponding head angles computed by rotating 3D models explicitly. In reality, a stand-alone vision application based on this work will require a landmark detection system as a preprocess. Gabor jet-based landmark tracking system [24] can be used to provide an accurate landmark positions, however it requires the landmarks to be initialized by other methods. Pose-specific graph matching [18] provides an another solution but with much lower precision. In general, the landmark locations and head angles will contain measurement errors. Although our previous studies indicated robustness to such errors [26], more systematic investigation on this matter should be performed in future.

Our future goal must address other types of variation such as illuminations and expressions for realizing more robust systems. There have been a number of progresses on both illumination variations [9, 12] and expression variations [10, 14]. However, an issue on combining the variation-specific solutions into a unified system that is robust against all types of variation has not fully been investigated. Our simple and general design approach can be advantageous for extending the presented models towards this goal.

## Acknowledgments

The authors thank Shigeru Akamatsu and Katsunori Isono for making their 3D face database available for this study. This study was partially supported by ONR grant N00014-98-1-0242.

## References

[1] M. S. Bartlett and T. J. Sejnowski. Viewpoint invariant face recognition using independent component analysis and attractor networks. In *Neural Information Processing Systems: Natural and Synthetic*, volume 9, pages 817–823. MIT Press, 1997.

[2] D. Beymer. Face recognition under varying pose. Technical Report A.I. Memo, No. 1461, Artificial Intelligence Laboratory, M.I.T., 1993.

[3] D. Beymer and T. Poggio. Face recognition from one example view. Technical Report 1536, Artificial Intelligence Laboratory, M.I.T., 1995.

[4] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, New York, 1995.

[5] X. Chai, L. Qing, S. Shan, X. Chen, and W. Gao. Pose invariant face recognition under arbitrary illumination based on 3d face reconstruction. In *Proc. AVBPA*, pages 956–965, 2005.

[6] R. Chellappa, C. L. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83(5):705–740, 1995.

[7] I. Craw, N. Costen, T. Kato, G. Robertson, and S. Akamatsu. Automatic face recognition: Combining configuration and texture. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 53–58, Zurich, 1995.

[8] J. G. Daugman. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 36:1169–1179, 1988.

[9] P. Debevec, T. Hawkins, H. P. Tchou, C. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proceedings of Siggraph*, pages 145–156, 2000.

[10] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski. Classifying facial actions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(10):974–988, 1999.

[11] S. Duvdevani-Bar, S. Edelman, A. J. Howell, and H. Buxton. A similarity-based method for the generalization of face recognition over pose and expression. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 118–123, Nara, 1998.

[12] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Generative models for recognition under variable pose and illumination. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 277–284, Grenoble, 2000.

[13] D. B. Graham and N. M. Allinson. Characterizing virtual eigensignatures for general purpose face recognition. In *Face Recognition: From Theory to Applications*, pages 446–456. Springer-Verlag, 1998.

[14] H. Hong. *Analysis, Recognition and Synthesis of Facial Gestures.* PhD thesis, University of Southern California, 2000.

[15] F. J. Huang, Z. Zhou, H. J. Zhang, and T. Chen. Pose invariant face recognition. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 245–250, Grenoble, France, 2000.

[16] H. Imaoka and S. Sakamoto. Pose-independent face recognition method. In *Proc. IEICE Workshop of Pattern Recognition and Media Understanding*, pages 51–58, June 1999.

[17] K. Isono and S. Akamatsu. A representation for 3D faces with better feature correspondence for image generation using PCA. In *Proc. IEICE Workshop on Human Information Processing*, pages HIP96–17, 1996.

[18] N. Krüger, M. Pötzsch, and C. von der Malsburg. Determination of face position and pose with a learned representation based on labeled graphs. Technical report, Institut fur Neuroinformatik, Ruhr-Universität Bochum, 1996.

[19] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42:300–311, 1993.

[20] M. Lando and S. Edelman. Generalization from a single view in face recognition. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 80–85, Zurich, 1995.

[21] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19:743–755, 1997.

[22] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data.* Wiley, New York, 1987.

[23] T. Maurer and C. von der Malsburg. Single-view based recognition of faces rotated in depth. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 248–253, Zurich, 1995.

[24] T. Maurer and C. von der Malsburg. Tracking and learning graphs and pose on image sequences. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 176–181, Vermont, 1996.

[25] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *Int. Journal of Computer Vision*, 14:5–24, 1995.

[26] K. Okada. *Analysis, Synthesis and Recognition of Human Faces with Pose Variations.* PhD thesis, University of Southern California, 2001.

[27] K. Okada, S. Akamatsu, and C. von der Malsburg. Analysis and synthesis of pose variations of human faces by a linear PCMAP model. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 142–149, Grenoble, 2000.

[28] K. Okada, J. Steffens, T. Maurer, H. Hong, E. Elagin, H. Neven, and C. von der Malsburg. The Bochum/USC face recognition system: And how it fared in the FERET phase III test. In *Face Recognition: From Theory to Applications*, pages 186–205. Springer-Verlag, 1998.

[29] K. Okada and C. von der Malsburg. Analysis and synthesis of human faces with pose variations by a parametric piecewise linear subspace method. In *Proc. the IEEE Conf. Computer Vision and Pattern Recognition*, volume I, pages 761–768, Kauai, 2001.

[30] K. Okada and C. von der Malsburg. Pose-invariant face recognition with parametric linear subspaces. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, Washington D.C., 2002.

[31] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. Technical report, Media Laboratory, M.I.T., 1994.

[32] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:1090–1104, 2000.

[33] M. Pötzsch, T. Maurer, L. Wiskott, and C. von der Malsburg. Reconstruction from graphs labeled with responses of Gabor filters. In *Proc. Int. Conf. Artificial Neural Networks*, pages 845–850, Bochum, 1996.

[34] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, New York, 1992.

[35] S. Schaal and C. G. Atkeson. Constructive incremental learning from only local information. *Neural Computing*, 10:2047–2084, 1998.

[36] L. Sirovich and M. Kirby. Low dimensional procedure for the characterisation of human faces. *Journal of the Optical Society of America*, 4:519–525, 1987.

[37] D. Valentin, H. Abdi, A. J. O'Toole, and G. W. Cottrell. Connectionist models of face processing: a survey. *Pattern Recognition*, 27:1209–1230, 1994.

[38] T. Vetter and N. Troje. A separated linear shape and texture space for modeling two-dimensional images of human faces. Technical Report TR15, Max-Plank-Institut fur Biologische Kybernetik, 1995.

[39] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19:775–779, 1997.

[40] L. Zhang and D. Samaras. Pose invariant face recognition under arbitrary unknown lighting using spherical harmonics. In *Proc. Biometric Authentication Workshop*, 2004.

[41] W. Y. Zhao and R. Chellappa. SFS based view synthesis for robust face recognition. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 285–292, Grenoble, 2000.

[42] W. Y. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35:399–458, 2003.