Parametric Piecewise Linear Subspace Method for Processing Facial Images with 3D Pose Variations

Kazunori Okada[†,*]
†Computer Science Department
University of Southern California
Los Angeles, CA 90089-2520
kazunori@organic.usc.edu

Christoph von der Malsburg[†,‡]
‡Institut für Neuroinformatik
Ruhr-Universität Bochum
Bochum, D-44801 Germany
malsburg@organic.usc.edu

Abstract

We propose a framework for learning a general, accurate and compact representation model of 3D objects from 2D images and demonstrate its application for analyzing and synthesizing facial images with head pose variation. The parametric piecewise linear subspace method accurately covers a wide range of pose variation in a continuous manner through a weighted linear combination of local linear models distributed in a 3D pose parameter space. Its parametric nature provides an explicit interface that permits clear interpretation of image variations and the connection to other functional modules. The linear design helps to avoid typical non-linear pitfalls such as overfitting and time-consuming learning. When learned and tested for a specific person, experimental results show sub-degree and sub-pixel accuracy within ±55 degree full 3D rotations and good generalization capability over unknown head poses.

Index Terms: 3D Head Pose Variation, Pose Estimation, Pose Transformation, Subspace Method, Parametric Model, Piecewise Linear Model, Face Recognition, Gabor Wavelets.

[*] The corresponding author.

1 Introduction

2D Images of objects change their appearance due to variations in both the intrinsic properties of objects and the extrinsic conditions of the surrounding environment. The latter includes the relative geometrical configurations of objects, lighting conditions, image projection settings, and so forth. Our investigation aims to realize a comprehensive computational representation scheme of 3D objects that accommodates all the possible extrinsic variations and accurately describes the intrinsic object properties. To reach this goal, this article discusses design and learning issues of such representation schemes and their application to the specific problem of human faces observed under head pose variations.

1.1 Continuous Parametric Representation Models

A serious difficulty with the aforementioned variations is that they are entangled with each other and are encoded *implicitly* in 2D images. In order to process object images accurately, we must know how to disambiguate these implicitly encoded variations and make them *explicit*. Only after this disambiguation, information on the extrinsic variation source becomes available for intuitively understanding and correctly manipulating the object's variational factors (e.g., pose and expression in faces) in the images. The variation information is also necessary for achieving variation-invariant object recognition by compensating given extrinsic image variations and processing only the intrinsic object properties. We treat this problem as a function approximation problem in which each object representation consists of a continuous bidirectional multivariate mapping function that realizes a direct parameterization of object/facial images by their corresponding variation parameters. Following the convention in the computer vision literature, we call a mapping from images to parameters *analysis mapping*, and its inverse *synthesis mapping*. In this framework, our task becomes to design the structure and to formulate the learning algorithm of the mapping functions such that the parameters describing different variation sources are as independent to one another as possible. We call this type of representation scheme *parametric representation model.*

1.2 Using Piecewise Linear Subspaces for Representing Non-Linear Data Structures

Another difficulty is to realize a *compact* representation that accounts for the non-linear structure of image variations. The number of all possible views for each object is enormous, corresponding to the product space of independent continuous variation sources. The size of each object representation must, however, be as compact as possible because there are also a very large number of objects to be remembered within limited memory resources. The subspace method, based on principal component analysis (PCA), has been successfully applied to realize compact image-based representations of human faces [70, 75, 12, 36]. This method describes an arbitrary facial image as a linear combination of a small number of orthonormal principal components (PCs) learned from training samples, as illustrated in figure 1(a). The method is known to be optimal in terms of least-square errors of data compression and reconstruction [70]. Moreover, by considering components of the weight vector as variation parameters, it can also be considered as a parametric representation model which separates the implicit variations into different linear components.

Despite these advantages, the linear method has the problem of being unable to cope with situation where the data lie near curved manifolds (see figure 1(b)). In such cases, there are many parameter combinations for which there is no possible image, and the parameters cannot be made to correspond to physical variation parameters. A number of previous studies have approached this problem by applying non-linear component analysis (e.g., principal curve [25], curvilinear component analysis [15] and kernel PCA [67, 43]), but they tend to compromise the bidirectional nature of mappings and/or compactness and computational simplicity in comparison with their linear counterparts. Figure 1(c) illustrates the *piecewise linear subspace method*: a linear solution of the non-linear problem in a 2D toy case. It collectively covers the non-linear data cloud with a set of localized linear subspaces, each of which consists of PCs learned from a local set of samples (for other applications of the piecewise linear approach, see [80, 66]). It improves overall accuracy since the collection of PCs now aligns well with the non-linear axis, although each local subspace can be accurate only within a local parameter range.

1.3 Our Approach and Scope

The *Parametric Piecewise Linear Subspace* (PPLS) method proposed in this article (for a preliminary description, see [50]) combines the advantages of a parametric representation model and of the piecewise linear method. This combination provides a representation framework in which a variety of image variations can be covered in a compact (linear) model, and the model is able to accurately account for the complex non-linear data structure. Using localized subspaces, however, introduces redundant coverage of the parameter space because each local subspace covers the entire space. To realize a one-to-one mapping, outputs of local mapping functions are interpolated by a weighted linear combination. The one-to-one, continuous mappings of PPLS helps to improve the processing accuracy of the previous methods based on discrete treatment of the variations, as described in section 1.4. For validating the proposed method, we applied it to the problem of representing and processing human faces with head pose variations. The analysis mapping function of PPLS provides a means for *pose estimation*, while the synthesis function provides for *pose transformation* or *facial animation*. Many studies have been devoted to facial image recognition in a past few decades (for surveys, see [65, 77, 9]). Despite these efforts, the pose variation problem, a subproblem of the general facial image processing, has not yet been fully solved [58, 56, 57]. By choosing faces among other objects as our target, a separation of the implicitly encoded image variations become more complex. The intrinsic facial properties, which distinguish one face from another, do not vary greatly across individuals. Therefore, the magnitudes of variation of the intrinsic properties often become much smaller than the magnitudes of the extrinsic variations, such as head pose and illumination, which makes the separation harder. For reliable face recognition, it is therefore important to account accurately for pose and other variations.

1.4 Related Studies

There are a number of related previous studies that have attempted to solve the problem of pose estimation and pose transformation of human faces. For pose estimation, many provided solutions based on 3D geometrical information of faces (e.g., 3D model fitting [30, 26, 11, 69] and 3D model reconstruction [72, 21, 84]). Other approaches include: 1) template matching of facial features [4, 74, 33, 16, 42, 68]; 2) geometric transformation of facial parts [8, 10]; and 3) general statistical learning of image-to-pose mappings [29, 46, 44, 53, 48, 83]. For pose transformation, on the other hand, three types of method have been reported: 1) discrete transformation from one head pose to another (e.g., Poggio's linear class theory [76, 61, 1, 2, 79], Gabor jet transformation [39, 40], and RBF network [35]); 2) continuous transformation from head angles to images [3, 53, 48]; and 3) reconstruction of 3D structural models [17, 19, 85]. Typically, these studies treated the continuous pose variation only discretely, resulting in sub-optimal estimation accuracy. Moreover, they studied only a limited range and/or a limited number of degrees-of-freedom (DOF) of pose variation, failing to address an issue of generalization to unknown and arbitrary head poses. Only a few have reported quantitative analyses of estimation accuracy. The best reported accuracy of pose estimation was approximately 3 degrees within a limited range of only 1D or 2D pose variation [47].

The approach based on 3D geometric information is an attractive alternative to our approach. After 3D structure information is provided, the pose estimation can be done fairly accurately [11, 84] and the pose transformation becomes a simple image-rendering [19]. However, reconstructing 3D face model is not a trivial task. Approaches in computer graphics [54, 24, 59, 37, 45] have recently become more accurate, however they are still labor-intensive and time-consuming. Vision-based approaches (e.g., Vetter [7], Fua [17], and Belhumeur [19]) have recently made notable advancements, however their computational instability is not fully resolved yet. Above all, this type of approach has a danger to become a pose-specific solution by employing analytical knowledge of 3D rotation.

The idea of piecewise linear approach has been applied to the problem of linearly approximating non-linear data in various domains for decades (e.g., sensor-motor coordination of robot hands [66, 80], sample classification [27], texture morphing [78]). For faces, Pentland et al. [55] proposed the view-specific eigenface method which consists of a set of PC-based local linear subspaces for different head poses. This method was successfully applied to pose-insensitive face recognition. For generic objects, recent work by Wieghardt and von der Malsburg [81] and Tenenbaum et al. [71] demonstrated unsupervised learning methods for topological registration of local PC-subspaces using the multi-dimensional scaling. The piecewise approach in these studies are similar to ours, however they did not address the continuous parametric representation of our focus.

A number of studies have also addressed the continuous, parametric, representation model of 3D objects using explicit non-linear methods. Beymer et al. [3] demonstrated a parametric representation model based on regularization networks [60] that includes both analysis and synthesis systems. Murase and Nayar [44] proposed the parametric eigenspace method which continuously parameterizes non-linear manifolds of pose- and illumination-variation of generic 3D objects using cubic-spline interpolation. Graham and Allison [23, 22] extended the above method for facial images of arbitrary people using RBF networks. Their continuous treatment of pose variation is similar to ours, however their systems consist of explicit non-linearity whose disadvantages are described in section 2.3. Moreover, none of them investigated generalization capability of their systems, and their experiments only used a limited range and DOF of pose variation. Our previous studies [53, 48] proposed the LPCMAP model that utilizes linear methods for the same task, avoiding the non-linear pitfalls. However, its accuracy was reduced beyond a limited range of head poses due to its linearity. This shortcoming is not tolerable since it severely limits the system's practical usability as will be described in section 2.2.

The method proposed in this article employs facial representation based on 2D Gabor wavelet transform [34, 82] while majority of the previous studies used raw grey-scale image values. The Gabor wavelet-based representation has been inspired by its biological implication; receptive fields in the primary visual cortical areas of higher vertebrates are most appropriately described as two-dimensional Gabor functions [32, 13, 14]. It has also been successfully applied to the face recognition problem, evaluated highly in the FERET competition [49, 57].

1.5 Organization

This article consists of seven sections. The first section has introduced a framework and context in which the presented study is carried out. The next section discusses design criteria that we consider for deriving the presented method. In the third section, the problem of our focus is formally defined. The fourth section describes our PPLS method as a solution to the problem. The method is implemented as a part of FLAVOR [64], an in-house, C++ based, object library for computer vision applications. This software implementation is referred as the PPLS system. We empirically evaluated this system with two data sets. Results of numerical experiments with a toy data set and 2D samples derived from 3D facial models scanned by a Cyberware^{TM} are presented in the fifth and sixth sections, respectively. Finally, we conclude this article by discussing the experimental results and our future work.

2 General Design Criteria

Towards our goal of realizing a comprehensive representation model, it is helpful to have a philosophy that can guide the numerous design choices required for making an optimal system. The following discusses three general criteria that we consider for designing the presented method.

2.1 Extendibility

An extendible system is one that can account for multiple variation sources: not only pose variation but also other types of variation such as illumination and expression. Such extendibility would be greatly hampered by the usage of variation-specific knowledge. For the case of pose variation, explicit use of analytical knowledge of ridged object's 3D Euclidean rotation leads to a variation-specific solution. Many previous studies of pose variation, which assume availability of 3D structure information [26, 40, 69], fall into this category. A solution to avoid this pitfall is to utilize a general learning algorithm for constructing a system solely from sample-statistics instead of manually formulating the system functions from such variation-specific knowledge. Another problem of the extension is the difficulty of coping with high dimensionality of the variation parameter space. When considering more variations within a single system, the dimensionality will become much more than the three that are sufficient to fully describe the 3D pose variation. For learning a mapping, an enormous number of samples will be required for appropriately populating a product space of such mid- or high-dimensional parameter spaces. This problem, known as *curse of dimensionality problem* [6], often causes poor extendibility of function approximation solutions from low-dimensional cases to high-dimensional ones. *Generalization* is a fundamental mechanism to resolve this problem. When learned functions possess this capability, it alleviates the necessity to populate the entire dot-product space, which is practically impossible. Choosing a *linear* function form is one way to emphasize the generalization capability, although it often poses a risk of losing the function's accuracy.

2.2 Accuracy

Accuracy is another important criterion when considering the practical usefulness of a system. The discrete sampling of the continuous pose angles in most of the previous studies has failed to achieve high accuracy because it requires a prohibitively large number of templates or functions for smoothly covering a wide range of the continuous variation. Another negative property of these discrete methods is the requirement for samples with specific pre-determined head poses. This requirement makes the sample collection procedure labor-intensive and stands in the way of making on-line systems. In order to realize an accurate pose processing system that avoids these shortcomings, the pose variation needs to be treated continuously.

2.3 Simplicity

The simplicity criterion emphasizes avoiding complexity in the structure and the learning algorithm of a system, and plays a crucial role for balancing a trade-off between the above two criteria. For maximizing the system's accuracy, a non-linear learning method may be used for fitting a system to the pose variation's non-linear characteristics. Such a non-linear method, however, complicates its learning process, requiring time-consuming iterative processes, and faces overfitting, which compromises its generalization capability [6] (non-linear pitfalls). This tradeoff between generalization and accuracy in the function approximation problem is known as the *bias/variance dilemma* [18]. When the function's internal DOF precedes the intrinsic DOF of the problem at hand, an approximation often results in overfitting. In the opposite case, it results in oversmoothing, which compromises its accuracy. The simplicity criterion supports the linear design of our framework. This design choice puts emphasis on avoiding the complexity in the learning process and facilitates the generalization capability that leads to good extendibility to other types of variation. In order to avoid possible oversmoothing, however, we must carefully design and evaluate the system in order to maximize its accuracy.

3 Problem Definition

Suppose that we have M training samples, denoted by a set of M pairs $\{(\vec{v}^m, \vec{\theta}^m) | m = 1, .., M\}$. A pair of vectors $(\vec{v}^m, \vec{\theta}^m)$ denotes a training sample of our model, where $\vec{v}^m$ is the m-th vectorized facial image and $\vec{\theta}^m = (\theta_1^m, \theta_2^m, \theta_3^m)$ are the 3D head angles of a face presented in $\vec{v}^m$. A problem of our focus is to learn bidirectional mapping functions between $\vec{v}$ and $\vec{\theta}$ from the training samples,

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

Ω represents the data entities learned from the training samples and also symbolizes a learned model. We call $\mathcal{A}_\Omega$ an analysis mapping and $\mathcal{S}_\Omega$ a synthesis mapping. 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, 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. In this study, we assume that these functions are *personalized*: each function is learned from and tested by samples from the same specific individual. Therefore, the synthesis mapping output $\vec{v}(\Omega)$ exhibits personal appearance that solely depends on Ω.

4 Parametric Piecewise Linear Subspace Method

The parametric piecewise linear subspace (PPLS) method [50] consists of a set of local linear models, each of which realizes the continuous analysis and synthesis mappings. 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, this method pieces together a number of local models distributed over the pose parameter space. 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. In order to account for the local model's parameter-range limitation, each model is paired with a radius-basis weight function. The PPLS then performs a weighted linear combination of local model's outputs, realizing a continuous global function.

4.1 Local Linear Model

The local linear model is implemented by the LPCMAP model [48]. It realizes the continuous, but only locally valid, bidirectional mapping functions. Each function is derived by combining two linear systems: 1) **linear subspaces** spanned by principal components (PCs) learned from training samples and 2) **linear transfer matrices**, which associate projection coefficients of training samples onto the subspaces and their corresponding 3D head angles.

4.1.1 Shape and Texture Decomposition and Image Reconstruction

The LPCMAP model treats shape and texture information separately in order to utilize them for different purposes. Figure 2(a) 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 are extracted from the image. The shape representation $\vec{x}^m \in \mathbf{R}^{2N}$ stands for an array of object-centered 2D coordinates of the N landmarks while the texture representation $\{\vec{j}^{m,n} \in \mathbf{R}^L | n = 1, .., N\}$ stands for a set of N Gabor jets sampled at the N landmarks [34, 82]. $\mathcal{D}_x$ and $\mathcal{D}_j$ denote operations of the shape and texture decomposition, respectively,

$$\vec{x}^m = \mathcal{D}_x(\vec{v}^m), \quad \vec{j}^{m,1}, .., \vec{j}^{m,N} = \mathcal{D}_j(\vec{v}^m). \tag{2}$$

The model also provides a means to reconstruct a grey-level facial image from a pair of shape and texture representations $(\vec{x}, \{\vec{j}^n\})$ in the form of a Gabor jet graph representation [62]. $\mathcal{R}$ denotes this reconstruction operation,

$$\vec{v} = \mathcal{R}(\vec{x}, \vec{j}^1, .., \vec{j}^N). \tag{3}$$

4.1.2 Transformation between Head Angles and Pose Parameters

In order to account for the intrinsic non-linearity of the mapping functions between the representations and 3D head angles, the model transforms 3D head angles $\vec{\theta}^m$ to the **pose parameters** $\vec{\varphi}^m \in \mathbf{R}^{T \geq 3}$ with a trigonometric functional transformation $\mathcal{K}$,

$$\begin{gathered}\vec{\varphi}^m = \mathcal{K}(\vec{\theta}^m) = (\cos(\tilde{\theta}_1^m), \sin(\tilde{\theta}_1^m), \cos(\tilde{\theta}_2^m), \sin(\tilde{\theta}_2^m), \cos(\tilde{\theta}_3^m), \sin(\tilde{\theta}_3^m), \\ \tilde{\theta}_i^m = \theta_i^m - u_{\theta i}, \quad \vec{u}_\theta = (u_{\theta 1}, u_{\theta 2}, u_{\theta 3}) = \tfrac{1}{M} \textstyle\sum_{m=1}^M \vec{\theta}^m.\end{gathered} \tag{4}$$

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

$$\vec{\theta}^m = \mathcal{K}^{-1}(\vec{\varphi}^m) = \vec{u}_\theta + (\arctan(\frac{\varphi_2^m}{\varphi_1^m}), \arctan(\frac{\varphi_4^m}{\varphi_3^m}), \arctan(\frac{\varphi_6^m}{\varphi_5^m})). \tag{5}$$

The transformation is carried out to the 3D head angles before they are related to other parameters. This helps to linearly construct the mapping functions because the parameters derived from images are better correlated to the pose parameters than to the raw angles [47].

4.1.3 Learning the Subspace Models for Shape and Texture Representations

As the first step of the model's learning process, we extract a small number of significant statistical modes from training facial images, as illustrated in figure 2(b). As a preprocess, a set of shape representations $\{\vec{x}^m\}$ and a set of texture representations $\{\vec{j}^{m,n}\}$ are extracted from the training facial images $\{\vec{v}^m\}$ by using the method described in section 4.1.1. The shape set $\{\vec{x}^m\}$ is subjected to Principal Component Analysis (PCA) [63] which solves the eigen decomposition problem of the centered sample covariance matrix, $XX^t\vec{y}^p = \lambda_y^p\vec{y}^p$, where X is a $2N \times M$ column sample matrix. This results in an ordered set of $2N$ principal components $\{\vec{y}^p|p = 1,..,2N\}$ of the shape ensemble (**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 $\{\vec{b}^{s,n}|s = 1,..,L\}$ (**texture PCs**). Performing this procedure for all the N landmarks results in a set of local texture PC sets $\{\vec{b}^{s,n}|s = 1,..,L; n = 1,..,N\}$. The subspace model [70] is based on a vector space spanned by a subset of the PCs in decreasing order of their corresponding variances (see figure 1(a)). A **shape model**, a subspace model for shape representations, is constructed by the first $P_0 \leq 2N$ shape PCs, $Y = (\vec{y}^1,..,\vec{y}^{P_0})^t$. A **texture model**, a set of localized subspace models for texture representations, is constructed by the first $S_0 \leq L$ texture PCs at each landmark n, $\{B^n = (\vec{b}^{1,n},..,\vec{b}^{S_0,n})^t|n = 1,..,N\}$ (for the analyses of information coded in each shape and texture PC, see [47]). These subspace models are used to parameterize a centered input representation by orthographically projecting it onto the subspace,

$$\vec{q}^m = Y(\vec{x}^m - \vec{u}_x), \quad \vec{u}_x = \frac{1}{M}\sum_{m=1}^{M}\vec{x}^m, \tag{6}$$

$$\vec{r}^{m,n} = B^n(\vec{j}^{m,n} - \vec{u}_j^n), \quad \vec{u}_j^n = \frac{1}{M}\sum_{m=1}^{M}\vec{j}^{m,n}, \tag{7}$$

where $\vec{q}^m \in \mathbf{R}^{P_0}$ and $\vec{r}^{m,n} \in \mathbf{R}^{S_0}$ denote **shape parameters** and **texture parameters**, projection coefficients of an input shape and texture representation, respectively. Due to the orthonormality of the PCs, the best approximation of an original representation can be uniquely reconstructed by linearly combining the PCs weighted by the parameters,

$$\vec{x}^m \approx \vec{u}_x + Y^t\vec{q}^m, \tag{8}$$

$$\vec{j}^{m,n} \approx \vec{u}_j^n + (B^n)^t\vec{r}^{m,n}. \tag{9}$$

4.1.4 Learning the Linear Transfer Matrices between Different Parameters

As the second step of the learning process, the pose and model parameters are linearly associated with each others for realizing direct mappings between $\vec{v}$ and $\vec{\theta}$, as illustrated in figure 2(c). For both the analysis and synthesis mappings, the pose parameters $\vec{\varphi}^m$ are related only with the shape parameters $\vec{q}^m$,

$$\vec{\varphi}^m = F\vec{q}^m, \tag{10}$$

$$\vec{q}^m = G\vec{\varphi}^m. \tag{11}$$

A $T \times P_0$ transfer matrix F (denoted as **SP** in figure 2(c)) is learned by solving an overcomplete set of linear equations, $FQ = \Phi$, $Q = (\vec{q}^1, .., \vec{q}^M)$, $\Phi = (\vec{\varphi}^1, .., \vec{\varphi}^M)$, with the Singular Value Decomposition (SVD) [63]. A $P_0 \times T$ transfer matrix G (denoted as **PS** in figure 2(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,

$$\{\vec{r}^{m,n} = H^n \vec{q}^m | n = 1, .., N\}. \tag{12}$$

A set of $S_0 \times P_0$ transfer matrices $\{H^n\}$ (denoted as **ST** in figure 2(c)) is learned by solving, $H^n Q = R^n, R^n = (\vec{r}^{1,n}, .., \vec{r}^{M,n})$, by SVD for all the N landmarks.

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 the data entities that are statically stored for each model,

$$LM := \{\vec{u}_x, \{\vec{u}_j^n\}, \vec{u}_\theta, Y, \{B^n\}, F, G, \{H^n\}\}, \tag{13}$$

where $\vec{u}_x$ and $\vec{u}_j^1, .., \vec{u}_j^N$ are average shape and texture representations, $\vec{u}_\theta$ is an average 3D head angle vector, Y and $B^1, .., B^N$ are shape and texture models, F and G and $H^1, .., H^N$ are shape-to-pose, pose-to-shape, and shape-to-texture transfer matrices.

4.1.5 The Local Analysis and Synthesis Mapping Functions

The following describes a construction of the analysis and synthesis mappings as a function of the learned LPCMAP model LM, as illustrated in figure 3. The analysis mapping function $\mathcal{A}_{LM}(\vec{v})$ is given by combining formulae (2), (6), (10), and (5),

$$\hat{\vec{\theta}} = \mathcal{A}_{LM}(\vec{v}) = \vec{u}_\theta + \mathcal{K}^{-1}(F \cdot Y \cdot (\mathcal{D}_x(\vec{v}) - \vec{u}_x)). \tag{14}$$

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 (for the correlation analysis of different parameters, see [47]). The shape synthesis mapping function $\mathcal{SS}_{LM}(\vec{\theta})$ is given by combining formulae (4), (11), and (8), 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 (4), (11), (12), and (9), 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 synthesis functions to formula (3),

$$\begin{aligned} \hat{\vec{v}} &= \mathcal{S}_{LM}(\vec{\theta}) = \mathcal{R}(\mathcal{SS}_{LM}(\vec{\theta}), \mathcal{TS}_{LM}(\vec{\theta})), \\ \hat{\vec{x}} &= \mathcal{SS}_{LM}(\vec{\theta}) = \vec{u}_x + Y^t \cdot G \cdot \mathcal{K}(\vec{\theta} - \vec{u}_\theta), \\ \{\hat{\vec{j}}^n | n = 1,..,N\} &= \mathcal{TS}_{LM}(\vec{\theta}) = \{\vec{u}_j^n + B^n \cdot H^n \cdot G \cdot \mathcal{K}(\vec{\theta} - \vec{u}_\theta) | n = 1,..,N\}. \end{aligned} \tag{15}$$

4.2 Global Piecewise Model

The global piecewise model provides a piecewise linear solution (the PPLS method) of the pose problem by combining local linear models described above. The model PM consists of a set of K local linear models,

$$PM := \{LM_k | k = 1,..,K\}. \tag{16}$$

We define **3D angle space** as a 3D parameter space spanned by the head angles (for example, see figure 5(a,b)). Each local model LM_k is assumed to be learned with training data sampled from one of local regions that are appropriately distanced from each other in the 3D angle space. Each set of the local training samples is associated with a model center. The **model center** is given by the average 3D head angles $\vec{u}_\theta^{LM_k}$ of the samples and specifies the learned model's location in the 3D angle space. The global analysis mapping function $\mathcal{A}_{PM}(\vec{v})$ is given by linearly combining K local pose estimates with appropriate weights,

$$\hat{\vec{\theta}} = \mathcal{A}_{PM}(\vec{v}) = \sum_{k=1}^{K} w_k \hat{\vec{\theta}}_k = \sum_{k=1}^{K} w_k \mathcal{A}_{LM_k}(\vec{v}), \tag{17}$$

where $\hat{\vec{\theta}}_k$ denotes a local pose estimate by LM_k and w_k is a weight for the LM_k's local estimate. On the other hand, the global synthesis mapping function $\mathcal{S}_{PM}(\vec{\theta})$ is given by linearly combining

K locally synthesized samples with the same weights,

$$\begin{aligned} \hat{\vec{v}} &= \mathcal{S}_{PM}(\vec{\theta}) = \mathcal{R}(\hat{\vec{x}}, \{\hat{\vec{j}}^n\}), \\ \hat{\vec{x}} &= \mathcal{SS}_{PM}(\vec{\theta}) = \textstyle\sum_{k=1}^{K} w_k \hat{\vec{x}}_k = \sum_{k=1}^{K} w_k \mathcal{SS}_{LM_k}(\vec{\theta}), \\ \{\hat{\vec{j}}^n\} &= \mathcal{TS}_{PM}(\vec{\theta}) = \{\textstyle\sum_{k=1}^{K} w_k \hat{\vec{j}}_k^n\} = \sum_{k=1}^{K} w_k \mathcal{TS}_{LM_k}(\vec{\theta}), \end{aligned} \tag{18}$$

where $\hat{\vec{x}}_k$ and $\{\hat{\vec{j}}_k^n\}$ denote locally synthesized shape and texture representations by LM_k.

Note that outputs of each local model cover the whole 3D angle space because of the model's continuous nature. In formulae (17) and (18), therefore, a weight vector $\vec{w} = (w_1, .., w_K)$ must be responsible for the localization of the model's output space. In order to meet this requirement, we use a normalized Gaussian function of distance between an input pose and each model center,

$$w_k(\vec{\theta}) = \frac{\rho_k(\vec{\theta} - \vec{u}_\theta^{LM_k})}{\sum_{k=1}^{K} \rho_k(\vec{\theta} - \vec{u}_\theta^{LM_k})}, \quad \rho_k(\vec{\theta}) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{\|\vec{\theta}\|^2}{2\sigma_k^2}), \tag{19}$$

where σ_k denotes the width of the Gaussian associated with the k-th local model. The Gaussian width determines the extent to which each local model influences the global output $\hat{\vec{\theta}}$ and $\hat{\vec{v}}$. The weight takes the maximum value when the input pose coincides with one of the model centers and its value decays as the distance increases. We set σ_k by the standard deviation of the 3D angle vectors of the LM_K's training samples. This setting is supported by our experimental results given in section 5.4.

Figure 4 illustrates the global piecewise model. Note that the local model views become more distorted as their model centers deviate further from an input pose, illustrating the pose range limitation of the LPCMAP model [48]. However, these largely distorted local outputs do not greatly influence a global output because their contribution is strongly inhibited by low weight values.

4.2.1 Gradient Descent-based Pose Estimation

The global analysis mapping function (17) cannot be solved by evaluating its right-hand-side because the weights are computed as a function of an unknown $\vec{\theta}$. To overcome this problem, we formulate a gradient descent-based iterative solution of the formula.

Let $\vec{x}$, $\vec{x}_i$, and $\vec{\theta}_i$ denote an input shape vector to this iterative algorithm and the shape and angle estimates by the i-th iteration, respectively. In order to set an initial condition $\vec{x}_0$ and $\vec{\theta}_0$, we first find a local model whose average shape $\vec{u}_x^{LM_k}$ is most similar to $\vec{x}$. Then, $\vec{x}_0$ and $\vec{\theta}_0$ are set by,

$$\vec{x}_0 = \vec{u}_x^{LM_{k_{min}}}, \quad \vec{\theta}_0 = \vec{u}_\theta^{LM_{k_{min}}}, \quad k_{min} = index(\min_{k=1}^{K} \|\vec{x} - \vec{u}_x^{LM_k}\|^2). \tag{20}$$

The following formulae are iterated until $\|\Delta\vec{x}_i\|^2$ becomes sufficiently small,

$$\begin{aligned} \Delta\vec{x}_i &= \vec{x} - \vec{x}_i, \\ \Delta\vec{\theta}_i &= \textstyle\sum_{k=1}^{K} w_k(\vec{\theta}_i)\mathcal{A}'_{LM_k}(\Delta\vec{x}_i), \\ \vec{\theta}_{i+1} &= \vec{\theta}_i + \eta\Delta\vec{\theta}_i, \\ \vec{x}_{i+1} &= \textstyle\sum_{k=1}^{K} w_k(\vec{\theta}_{i+1})\mathcal{SS}_{LM_k}(\vec{\theta}_{i+1}), \end{aligned} \tag{21}$$

where η is the learning rate that is set to a very small value, and $\mathcal{A}'_{LM_k}$ is a slight modification of formula (14) that has a shape vector interface. Note that the weighted sum of the analysis mappings in (21) 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 our global piecewise model, such gradients are only available at the locations of the K discrete model centers. The second formula in (21), 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 [48] supports the validity of this approximation. Moreover, our choice of the initial condition should decrease the chance of being trapped at a local minimum during the iterations as long as a sufficient number of local models are allocated in the 3D angle space. 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 (18) remains valid.

4.2.2 Self-Occlusion Handling

As a head rotates, some landmarks will become hidden behind other facial parts. This problem is called **landmark self-occlusion**. Our system must handle this problem because it is designed to cover a wide range of head poses, in which such occlusion occurs naturally. This problem suffers PCA used for learning the shape model because PCA requires a data set with constant dimensionality. Landmark self-occlusion introduces uncertainties in shape vectors, resulting in missing values for certain vector components. This causes an erroneous bias to resulting PCs because sample moments, such as data mean and variance, cannot be computed correctly from such incomplete data in a straight-forward manner. This problem is known as the **missing data problem** [38]. We handled this problem by applying the **mean imputation method** [38] which fills in each missing component by a mean computed from all available data at the component dimension. As a result, the value of each missing component becomes zero when the data set is centered, which cancels the influence of the missing components to the value of component-wise variances. This method has been shown to perform well when the number of missing components is relatively small. Because it makes the data complete, the straight-forward procedure of PCA becomes feasible. However, it causes an underestimation of sample covariance, which introduces a bias that is not related to the true nature of the data. Because of this, the method does not usually perform well when there are a large number of missing components.

5 Toy Data Experiments

In this section, we evaluate the PPLS system with a toy data set that is created artificially under strict control. By using such data, we seek an experimental proof of our method's correctness and investigate the optimal parameter settings for maximizing its performance.

5.1 Toy Data

We created artificial shape representations, each of which consists of a 2D orthographic projection of 25 3D landmark points. These 3D landmarks are located on a 5 by 5 square grid pasted onto a surface of a rotating 3D unit sphere. 2D coordinates of the projected points are scaled and translated for fitting them into a 128 by 128 image coordinate space. 3D rotation angles for each shape representation are given by explicit rotation angles of the sphere. Texture representations are not considered in this experiment. These shape samples differ from realistic facial data in that their depth profile is much more regular than that of faces. Furthermore, there are no measurement errors of landmark locations and rotation angles.

As training samples, we created 7 local training sample sets . These local sets are distributed over the 3D angle space and centered at one of the 7 model centers, (0,0,0), (±40,0,0), (0,±40,0), and (0,0,±40), as illustrated in figures 5(a,b). For each local set, we created 403 samples by rotating the sphere and projecting it to a 2D plane in one degree interval within a ±15 degree range from the center. We used two types of rotation: one is a rotation along only one rotation axis at a time and the other is a rotation along two axes simultaneously. As a total, there are 2831 training samples which cover a range of ±55 degree 3D rotation.

As test samples, we created 804 samples whose 3D rotation angles are different from those of the training samples. Figure 5(b) illustrates a 2D projection of 3D angle distributions of the test samples. The test sample set covers the range of ±50 degree 3D rotation. It includes two types of angle distribution: one falls between several local training sets (four crosses in the figure) and the other is within a sparsely populated region of a local set (four long horizontal and vertical lines). The former poses a more difficult testing situation than the latter, requiring a smooth interpolation between more than two neighboring local models.

The landmark self-occlusion is simulated by introducing an occluding plane, $z = c$ (c: constant, $\|c\| \leq 1$), which is parallel to an image plane. A landmark point is considered as occluded when it goes below the occluding plane.

5.2 Test Formats

We utilize two types of test for evaluating the PPLS system's performance. An **accuracy test** evaluates the system's accuracy by testing a learned system with the training samples (known poses). On the other hand, a **generalization test** evaluates the system's generalization by testing the system with the test samples described above (unknown poses). Since the texture representations are not available in the toy data, the following evaluates the pose estimation and shape synthesis processes only.

5.3 Evaluation of the System's Correctness

First, we studied the average processing errors of the one-shot pose estimation process (17) and shape synthesis process (18) in the most controlled conditions. The conditions are that 1) all landmarks are considered to be visible ($c = -1$), and 2) a trigonometric functional transformation $\mathcal{K}$ includes pairwise products of the trigonometric functions. We compare two data-precision settings: float and integer accuracy of the shape representations. The former gives the most accurate landmark position information possible, while the latter provides a more realistic situation. Figure 6(a) shows the results of the accuracy test for the two processes in the above-described conditions. The average errors are plotted against the number of PCs included in a shape model. For both pose estimation and shape synthesis, the average error of the float-accuracy system became approximately zero after including the first 6 shape PCs (12% of the total PCs). This result strongly supports our system's correctness. The difference of the errors between the float and integer systems was small, indicating the system's robustness against small measurement errors in landmark locations. This system setting with the increased dimensionality of the pose parameters, however, resulted in overfitting with poor performance for the generalization test. $\mathcal{K}$, in the form of formula (4) without the pairwise products, provided the best balance between the system's accuracy and generalization.

5.4 Evaluation of the Gaussian Weight Function

Next, we investigated the influence of different widths of the Gaussian weight function on our system's performance. σ_k of the weight function (19) controls the range over which each local model is responsible in the 3D angle space. This experiment treats σ_k as a function of the sample standard deviation and investigates an appropriate value of σ_k which aligns the Gaussian width to the effective pose range of our local linear models, which was shown to be about ±15-20 degrees [48]. For this experiment, we use $\mathcal{K}$ as defined in formula (4), the integer accuracy of the landmark positions, and all landmarks are again considered visible. We conducted the generalization test only with a fixed number of 8 shape PCs. Figure 6(b) shows average errors plotted against the different values of p, a positive scaling factor of the sample standard deviation, $\sigma_k = p \times \sqrt{\frac{1}{M_k-1}\sum_{m=1}^{M_k}(\vec{\theta}^m - \vec{u}_\theta^{LM_k})^2}$. Results showed that the minimum error was reached when σ_k was near the sample standard deviation ($p = 1$) for both pose estimation and shape synthesis. This suggests that the optimal setting of σ_k is given by the sample standard deviation itself. The error curves for both processes were smooth, suggesting that a slight variation of the σ_k value does not greatly influence the system's performance.

5.5 Evaluation of the Occlusion Handling

Next, we investigated the influence of landmark occlusion on our system's performance. We compared average errors of the two one-shot processes with ($c = 0.1$) and without ($c = -1.0$) occlusion. At most, 10% of the total landmarks in a local set were occluded in the occlusion data set. The same settings in the previous section were used ($\mathcal{K}$ without the pairwise products, integer precision, $p = 1$). Figure 6(c) shows results of the generalization test. They showed that the error difference of the two data sets was very small (0.2 degrees for pose estimation and 0.1 pixels for shape synthesis, with the first 8 shape PCs). This supports the effectiveness of our missing data handling by the mean imputation method. The average errors were approximately 0.7 degrees and 1.1 pixels for the two processes.

5.6 Evaluation of the Gradient Descent-based System

Lastly, we evaluated the gradient descent-based system in section 4.2.1 in the most realistic conditions with integer precision and 10% landmark occlusion. We iterated the gradient descent loop 500 times and set the learning rate η to 0.01. We compared average errors of the gradient descent-based system with those of the one-shot system. Figure 6(d) shows results of the generalization test. They showed that the error difference of the two systems was again very small (0.1 degrees and 0.1 pixels). This supports the feasibility of our complete system with the gradient descent-based solution. The average errors were approximately 0.8 degrees and 1.0 pixels, indicating good accuracy and generalization. Within these experimental settings, we did not observe trappings into local minima that were significantly distanced from the global minima.

6 Cyberware-scanned Face Data Experiments

In order to assess our system's feasibility in more realistic scenarios, we evaluate the PPLS system with samples derived from actual faces. For rigorous analyses, however, we must collect a large number of samples with specific head poses for many people, which is not an easy task. To mitigate this difficulty, we use 3D face models pre-recorded by a Cyberware^{TM} scanner. Given such data, relatively faithful facial images with arbitrary, but precise, head poses can easily be created by image rendering [31, 7].

6.1 Cyberware-scanned Data

In this experiment, we used 20 face models randomly picked from the ATR-Database [31] as shown in figure 7(a). The same pose distributions used for the toy data experiments, shown in figures 5(a,b), are also used for these experiments. As a result, for each individual, we have 804 test samples and 2831 training samples constituting 7 local training sets, each of which includes 403 samples. The test set, the total training set, and each local training set, covers a pose range of ±50, ±55 and ±15 degrees along each rotation axis, respectively. Figure 7(b) shows the definition of the 20 facial landmarks. These landmarks were manually placed on the surface of the 3D model. For each 2D sample, 2D landmark locations are then derived by rotating the 3D landmark coordinates and projecting them onto an image plane. 3D head angles are also given by the explicit rotation angles of the models. The self-occlusion information is provided from the rendering system. 5 to 10% of the total landmarks were self-occluded in each local training set.

6.2 Test Formats

In order to assess our method's improvement in performance from our previous studies [53, 48], we compare the PPLS and LPCMAP systems learned from the same training samples. The former consists of 7 local linear models, each of which is learned from one of the local training sets; the latter is a single local model learned from the total 2831 samples. Both systems use $\mathcal{K}$ without the pairwise products and integer shape precision. The PPLS system use σ_k set to the sample standard deviation and the gradient descent-based system with 500 iterations and η set to 0.01.

6.3 Average Error and Similarity Analysis

Figure 8(a) compares average pose estimation errors of the PPLS and LPCMAP systems in both accuracy and generalization tests. In the accuracy test, the average angular error with the first 8 PCs was 0.8 ± 0.6 and 3.0 ± 2.4 degrees and the worst error was 5.6 and 18.9 degrees for the PPLS and LPCMAP systems, respectively. In the generalization test, the average error was 0.9 ± 0.6 and 2.4 ± 1.4 degrees, and the worst error was 4.5 and 10.2 degrees for the two systems. Figure 8(b) compares average shape synthesis errors of the two systems in the two test cases. In the accuracy test, the average landmark position error with the first 8 PCs was 0.8 ± 0.4 and 2.2 ± 1.2 pixels, and the worst error was 3.0 and 7.6 pixels for the PPLS and LPCMAP systems, respectively. In the generalization test, the average error was 0.9 ± 0.4 and 2.4 ± 0.7

pixels, and the worst error was 2.7 and 5.6 pixels for the two systems. Figure 8(c) compares average similarities of synthesized and ground-truth textures for the two systems 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 ± 0.03 and 0.91 ± 0.04, and the worst similarity was 0.81 and 0.73 for the PPLS and LPCMAP systems, respectively. In the generalization test, the average similarity was 0.945 ± 0.03 and 0.88 ± 0.03, and the worst similarity was 0.82 and 0.77 for the two systems.

For all three tasks, the PPLS system greatly improved performance over the LPCMAP system 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. The errors in these experiments were also similar to those with the toy data shown in figure 6(d), suggesting our system's robustness against irregular depth variation of faces. As a reference for our texture similarity analysis, we computed average texture similarities over 450 people from the FERET database [58, 57]. The average similarity was 0.94 ± 0.03 for the same person pairs and 0.86 ± 0.02 for the most similar, but different, person pairs. The average similarity of the PPLS system was higher than that of the large FERET database, which validates the results of our texture similarity analysis.

6.3.1 Synthesized Samples

Figure 9 illustrates model views: images reconstructed from samples synthesized by formula (18) of the PPLS system. Note that facial images reconstructed by the Pötzsch algorithm [62] do not retain original picture quality. This is because a transformation $\mathcal{D}_j$ from images to our Gabor jet-based representations is lossy due to coarse sampling in both image and frequency spaces. Nonetheless, these images still capture characteristics of faces fairly well. Figure 9(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 9(b) shows test samples whose pose is close to a model center, but with a large rotation along one dimension. In contrast, figure 9(c) shows samples whose pose is in-between several model centers. For all three cases, the original and synthesized model views were very similar, indicating our system's good accuracy and successful generalization to unknown head poses even for a wide head pose range. Figure 9(d) shows model views synthesized by the PPLS and LPCMAP systems for comparison.

7 Discussions

This article presented the parametric piecewise linear subspace method: a novel framework for parameterizing 2D images of 3D objects by their physical variations. Using this framework, we demonstrated a simultaneously general, accurate and simple solution to the problem of head pose estimation and facial image synthesis as a function of head poses. An implementation of the method was empirically evaluated. The results showed that our system possesses 1) high accuracy (sub-degree and sub-pixel); 2) good generalization capability over unknown head poses; and 3) a coverage of a wide range and a full number of DOF of pose variation (±55 degree 3D rotation). The continuous nature of our method helps us to reach high accuracy by smoothly interpolating the discrete local models. It also provides the basis for an on-line visual learning system which simplifies an otherwise labor-intensive data collection procedure. The explicit variation parameters provide a common reference frame which may be used to interface different functional modules in multi-modal systems. The generalization capability not only facilitates the extendibility to other types of image variations, but also enables learning from few samples. Our subspace-based method facilitates compactness of the representation model. In our experimental setting, the system achieved a data compression with a factor of twenty; the size of a PPLS model learned from 2800 training samples was equivalent to approximately 140 samples. In the scope of the on-line visual learning, a PPLS model captures only limited views of a face, if only such limited views are given as training samples. This is analogous to the fact that it is hard to recognize someone if you have seen the person only from a certain view-point and you are presented with a previously unseen view. Psychophysical studies by Biederman and Kalocsai [5] and Troje and Bülthoff [73] reported this effect in human face recognition tasks. One of the applications of our method is pose-insensitive face identification (for our reports on this task using extensions of the proposed method, see [47, 53, 52, 51]). Furthermore, the presented framework can be extended for accommodating interpersonal variations [47]. Our method should also be applicable to classes of objects other than faces. Although we did not investigate its applicability to non-face objects, it does not impose any constraints that limit its usage only to faces. This advantage extends our proposed method to much wider application scenarios.

As future work, the next step is to extend our method in the context of other types of image variations (for a related work on facial expression, see [28]). Although the experimental results showed good generalization capability of our system, they do not prove the method's extendibility. A challenge will be to avoid the curse of dimensionality problem when realizing a single system which accounts for the pose, illumination, and expression variations all together. Furthermore, our framework may be used to realize an automatic pose-insensitive landmark finder based only on information derived from a single static view, using a continuous bidirectional mapping between shape and texture. For locating facial landmarks, our previous study [47] employed a Gabor jet-based automatic landmark tracking system [41], however this method relies on temporal continuity of an object in a view frame, requiring information from previous frames. Such a single view-based landmark finder will help to further automate our system. Lastly, although our experiments showed sufficiency of the simple sample manipulation-based handling of the missing data problem, our method may produce non-negligible errors when we consider much wider range than the ±55 degree poses. More sophisticated solutions (e.g., EM algorithm [38, 20]) may be required to address this problem properly.

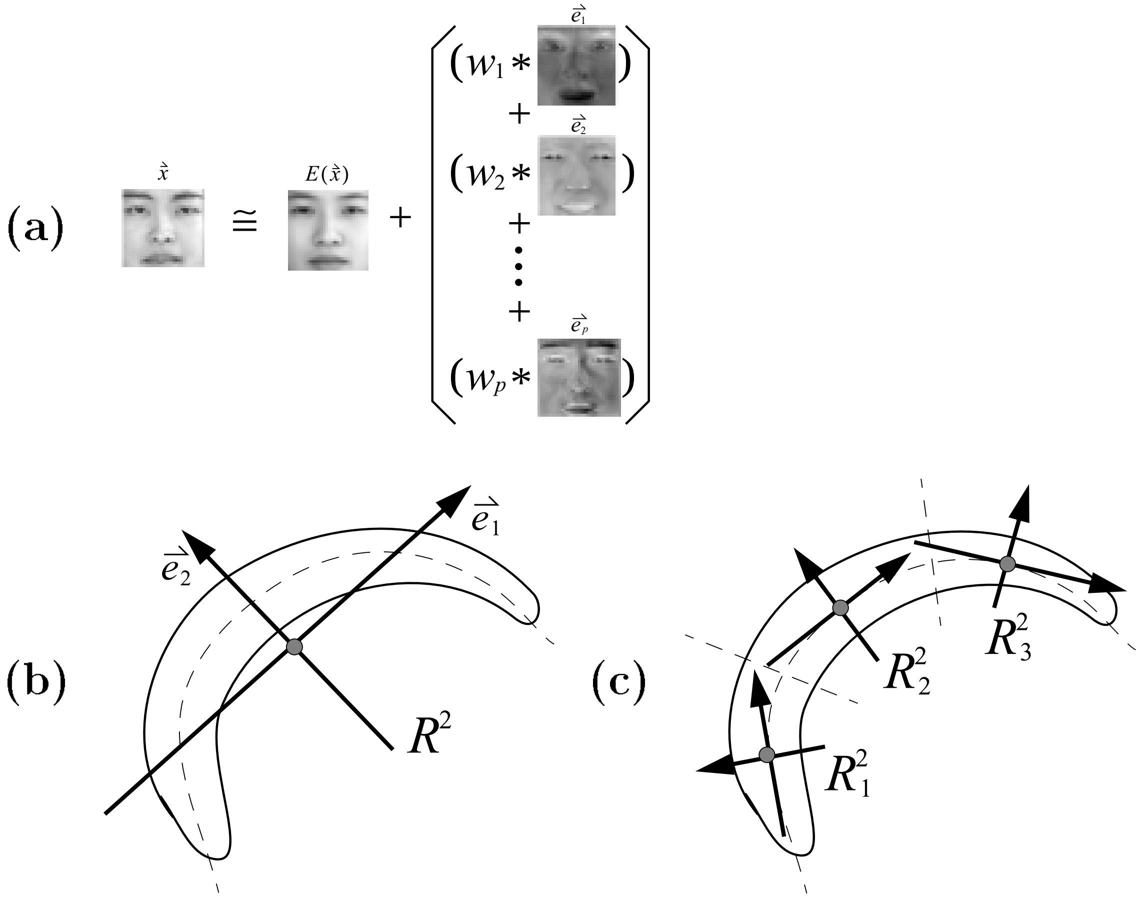

Figure 1. Linear subspace-based data encoding. (a): PC-based subspace model by Sirovich and Kirby [70]; (b,c): comparison of the global subspace (b) and the piecewise linear subspaces (c) for a simplified 2D non-linear data ensemble. In figure (a), an image $\vec{x}$ is approximated as the sum of the average image $E(\vec{x})$ and a weighted sum of PCs $(\vec{e}_1, .., \vec{e}_p)$. The weight vector $(w_1, .., w_p)$ is a compact representation of the image $\vec{x}$. Due to orthonormality of PCs, the weight vector is simply derived by orthographically projecting the input $\vec{x}$ to the vector space spanned by the PCs. In figures (b,c), a bent ellipse symbolizes a 2D data cloud and the curved interrupted line its non-linear axis. When components are aligned well with the main axis, their weights (parameters) can describe the ensemble's non-linear structure more accurately.

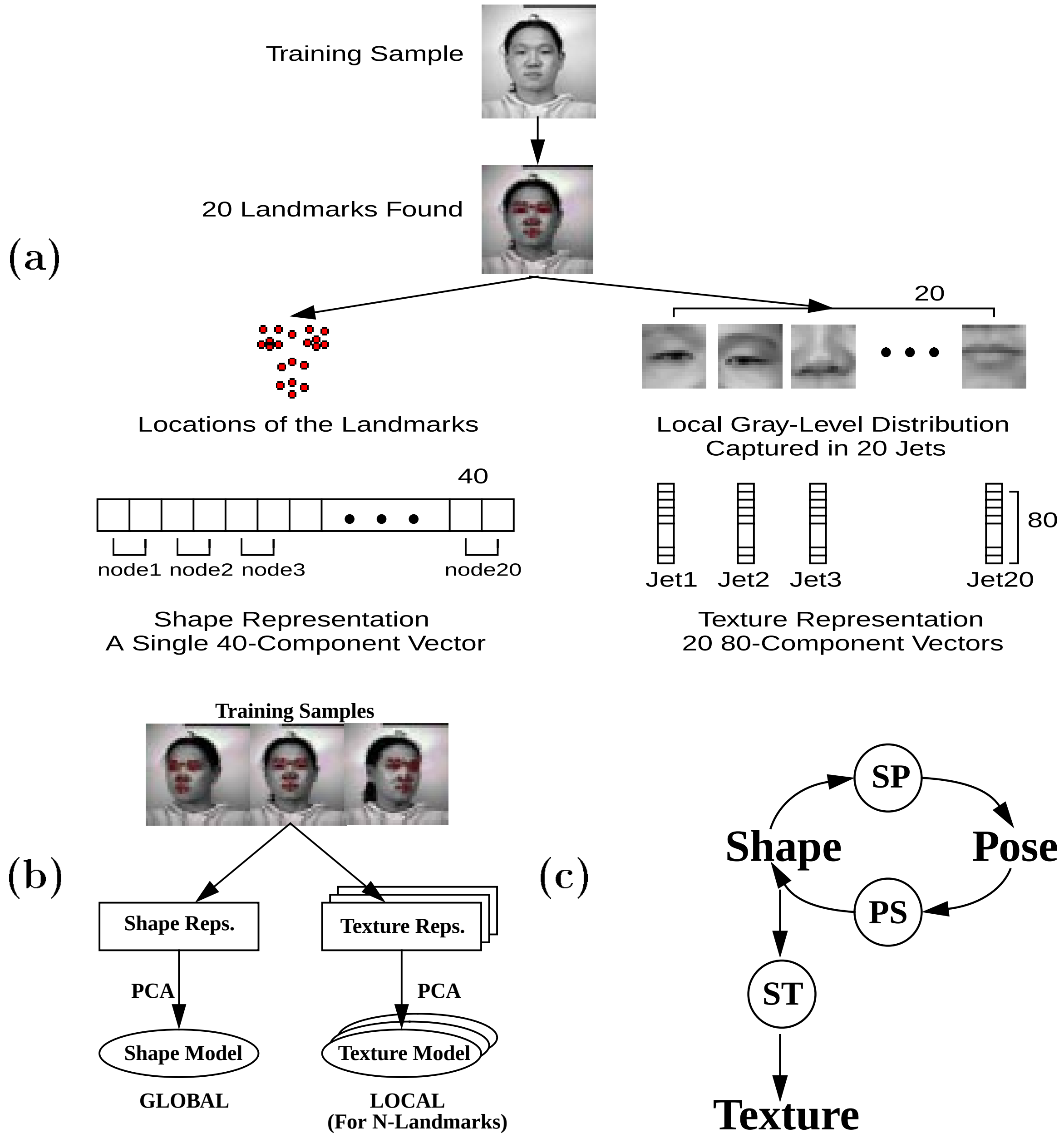

Figure 2. Learning processes of the local model. (a): shape and texture decomposition process; (b): subspace models for shape and texture representations learned by principal component analysis; and (c): transfer matrices associating different parameters. Figure (a) describes the decomposition process with parameter settings used for our experiments in section 6; the number of landmarks N is set to 20; the length of a texture vector L is set to 80, as coefficients of a bank of 5-level, 8-orientation, 2D, complex Gabor filters. In figure (b), a rectangle denotes a set of training samples; an ellipse denotes a PC-based subspace model. In figure (c), SP denotes the shape-to-pose transfer matrix (a linear map from shape to pose parameters); PS: the pose-to-shape matrix; and ST: shape-to-texture matrices. Note that ST symbolizes 20 different matrices because of our localized texture representation.

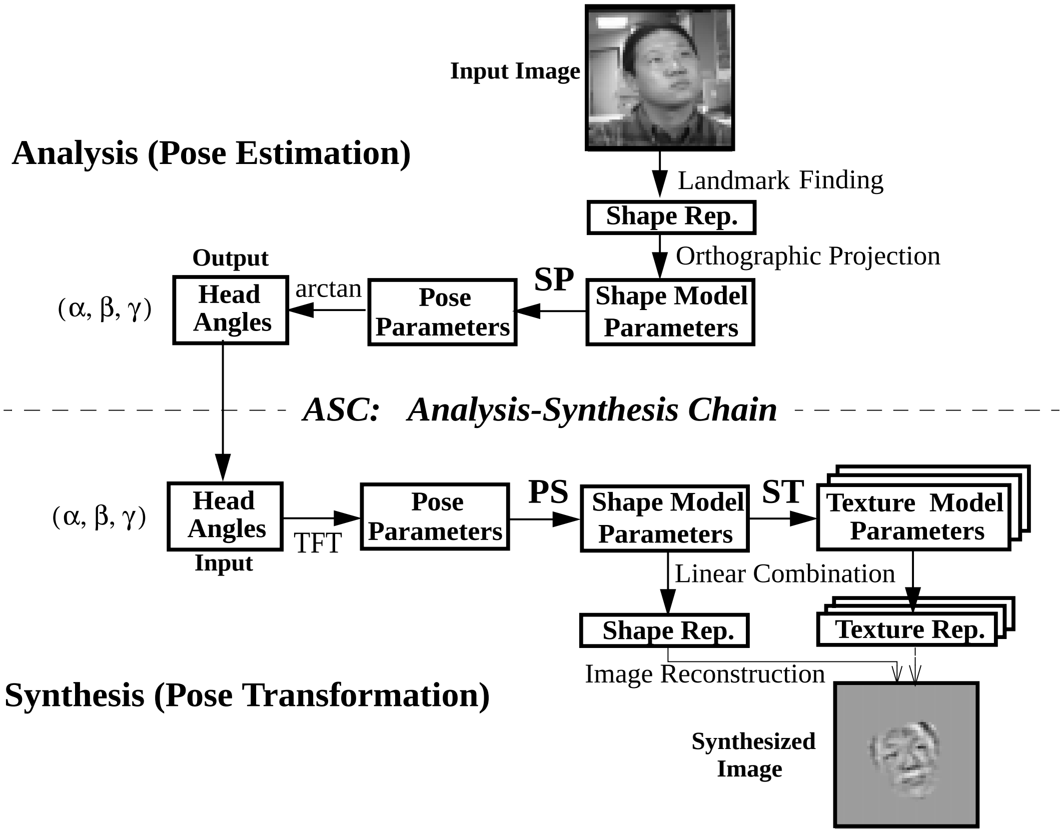

Figure 3. Local analysis and synthesis mapping functions. TFT denotes the trigonometric functional transformation $\mathcal{K}$; $\arctan$ denotes $\mathcal{K}^{-1}$. SP, PS and ST denote the transfer matrices shown in figure 2(c). The analysis and synthesis processes can be concatenated by using the analysis outputs as the synthesis inputs. This concatenated process is called the analysis-synthesis chain, and it can be used to fit a learned model to a face with arbitrary head poses, resulting in a pose-aligned model view at the bottom-right.

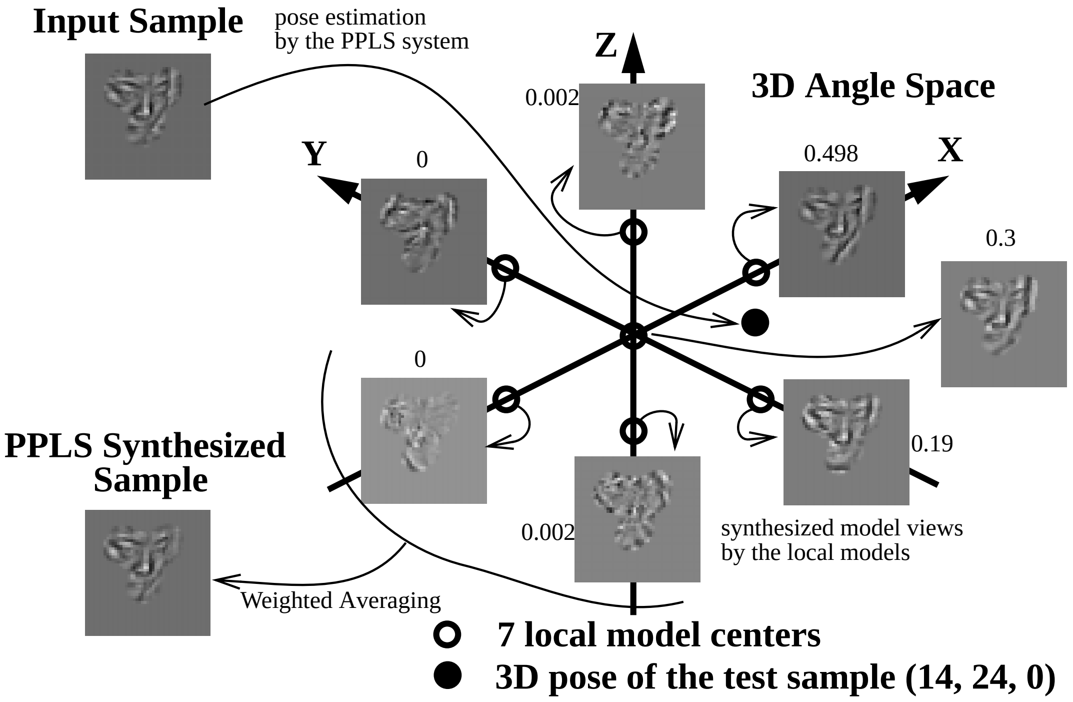

Figure 4. A sketch of the global piecewise model (the PPLS system). For pose estimation, an input sample at the top-left corner is first subjected to the analysis process of each local model whose model centers are denoted by circles. This results in multiple, local, pose estimates. As a global analysis process, these local estimates are averaged with the Gaussian weights, resulting in the global pose estimate denoted by a black dot. For pose transformation, each local model takes a 3D head pose as input and synthesizes shape and texture representations whose model views are shown near each model center. A global synthesized sample is given by a weighted linear combination of these locally synthesized samples. Weights, shown next to the local views, are computed based on the distance of the input and model centers.

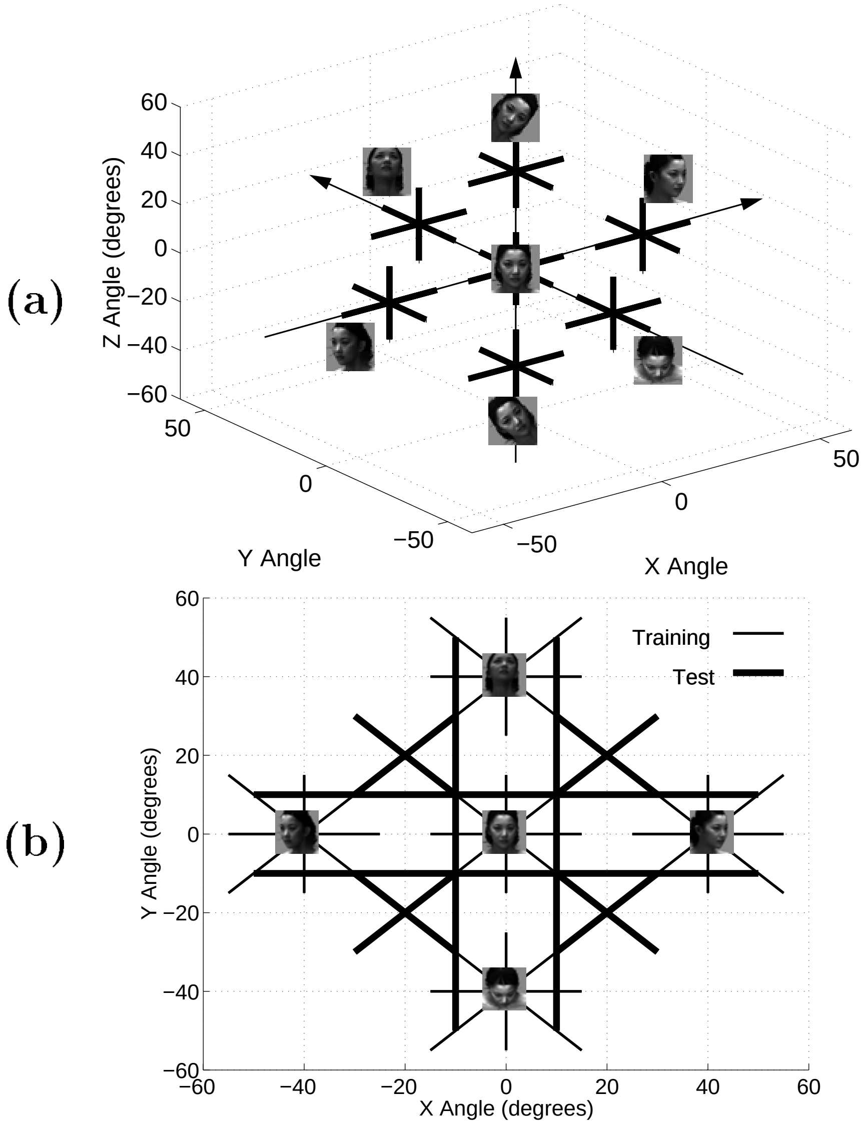

Figure 5. 3D angle distribution of training and test samples for our experiments. (a): training samples in 3D angle space; (b): both training (thin lines) and test (thick lines) samples projected onto a 2D x-y plane. Facial images in the figures are used for describing the different rotation angles of the model centers. These sample distributions are used for both toy data experiments in section 5 and Cyberware-scanned face data experiments in section 6.

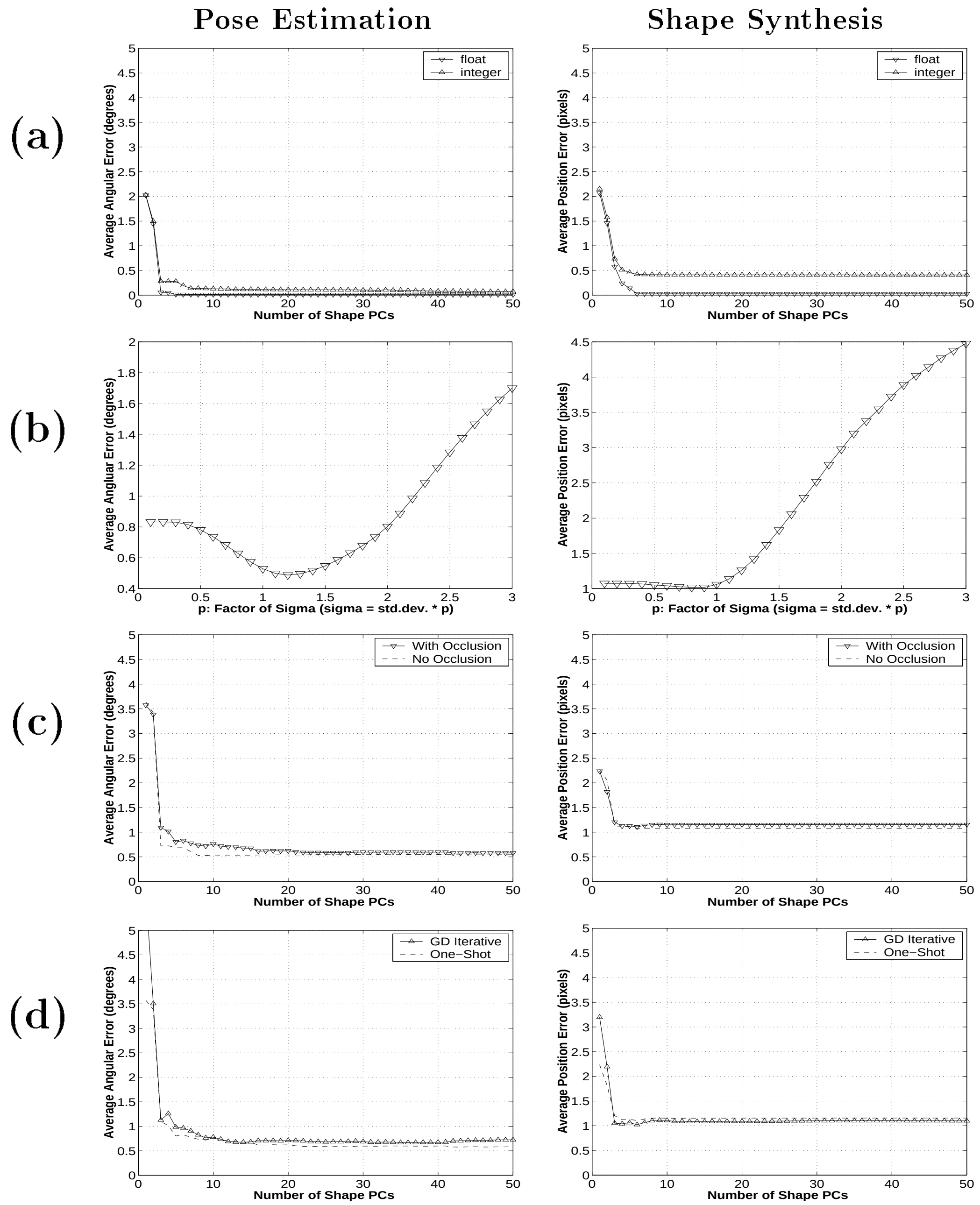

Figure 6. Results of the toy data experiment in various settings. Experiment (a) conducted the accuracy test; experiments (b,c,d) conducted the generalization test. For pose estimation, we evaluated angular error in degrees averaged over 3 rotation dimensions and 2831 samples (for the accuracy test) or 804 samples (for the generalization test). For shape synthesis, we evaluated landmark position error in pixels averaged over 25 landmarks and 2831 training or 804 test samples. (a): systems in the most controlled condition; (b): with different Gaussian width; (c): with and without the landmark self-occlusion; and (d): with and without the gradient descent-based pose estimation.

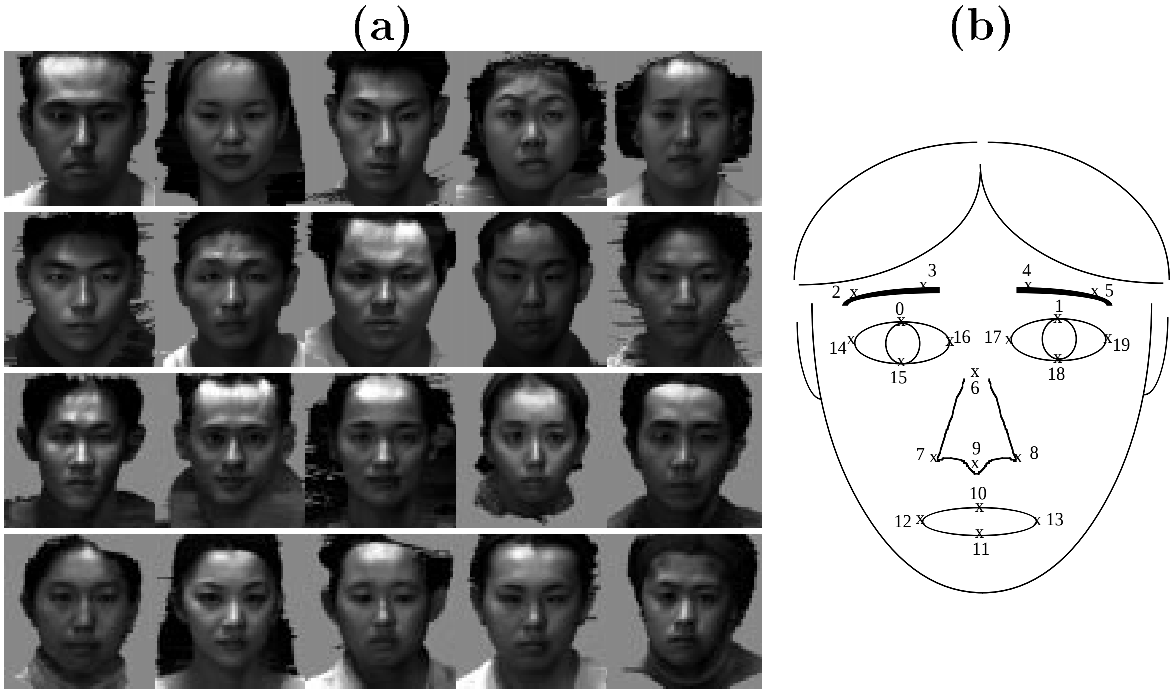

Figure 7. Face data for our experiments. (a): frontal views of 20 3D face models from the ATR database, consisting of the faces of 10 female and 10 male Japanese people; (b): definition of facial landmarks, 20 distinctive feature locations within the inner region of faces.

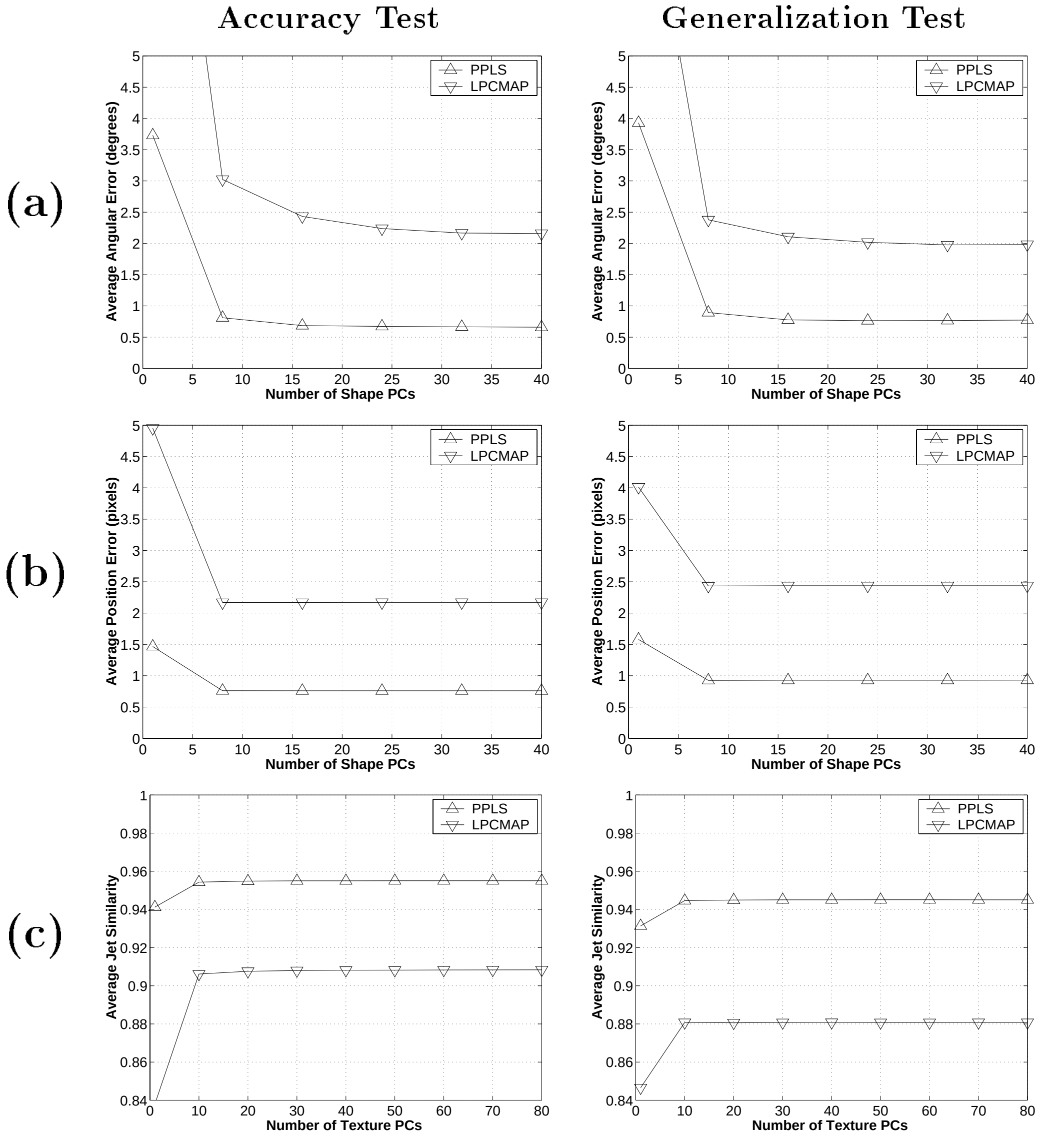

Figure 8. Results of the average error/similarity analysis with our face data, comparing the PPLS and LPCMAP systems. The average errors/similarities are plotted against the number of shape or texture PCs included in each local model of the systems. (a): pose estimation errors in degrees, averaged over 3 rotation dimensions, 2831 training (accuracy test) or 804 test (generalization test) samples, and 20 people; (b): shape synthesis errors in pixels, averaged over 20 landmarks, 2831 training or 804 test samples, and 20 people; and (c): texture synthesis accuracy in Gabor jet similarity values, averaged over 20 landmarks, 2831 training or 804 test samples, and 20 people.

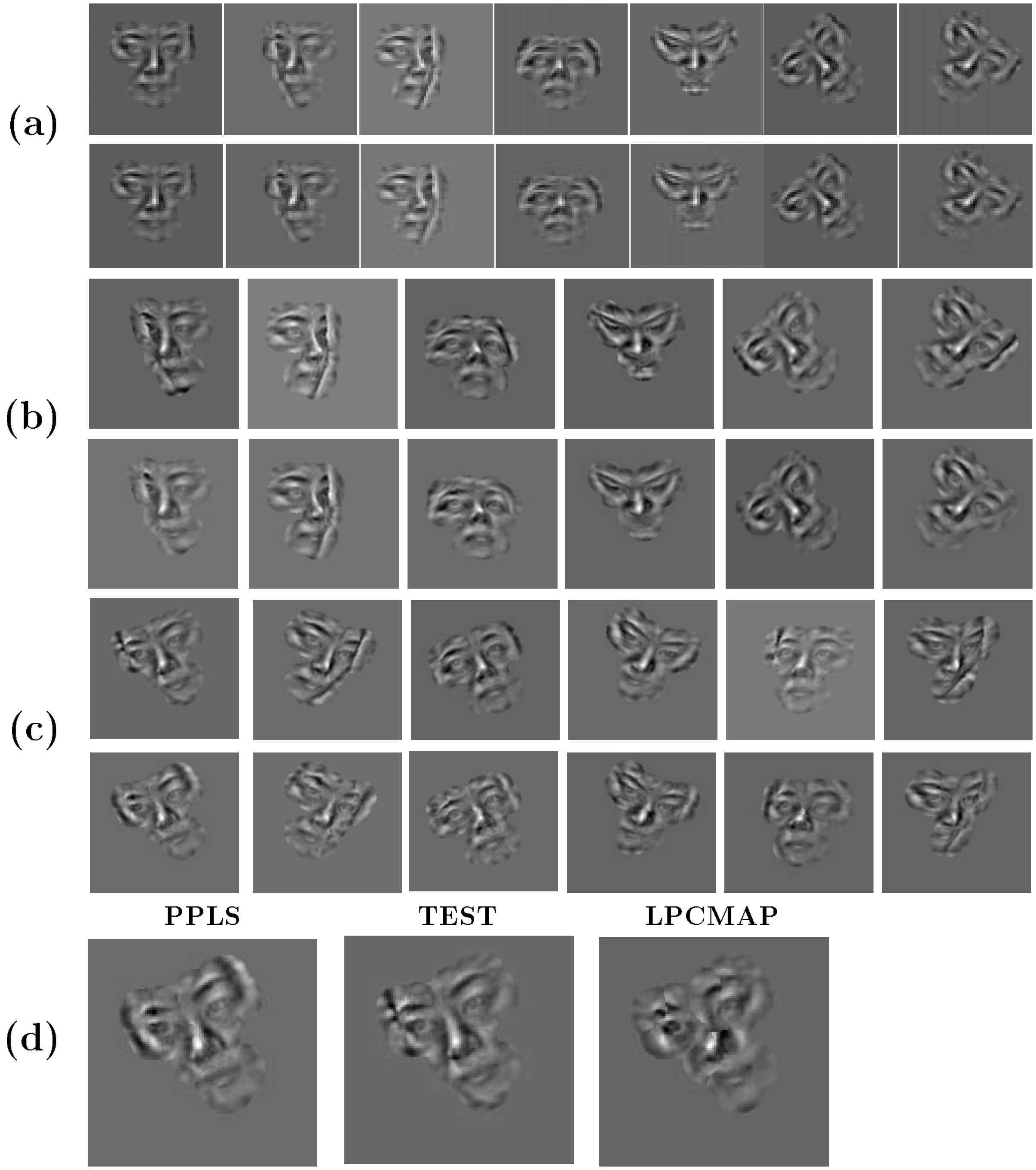

Figure 9. Examples of sample synthesis by the PPLS system. In figures (a,b,c), images in the first row are directly reconstructed from recorded training or test samples as a reference; those in the second row are corresponding (pose-aligned) model views synthesized by the PPLS system. (a): training samples with known head pose (accuracy test case); (b): test samples with unknown head poses with large rotation along one dimension (generalization test case); (c): test samples with unknown head pose far from any of the model centers; and (d): an example of corresponding model views synthesized by the PPLS and LPCMAP systems in the condition of (c).

Acknowledgments

The authors wish to thank Shigeru Amakatsu and Katsunori Isono for making their 3D face database and rendering system available for this study. We also thank Jan Wieghardt and Junmei Zhu for helpful discussions, Lawrence Kite for proof-read, and developers of the FLAVOR libraries for providing this work's software platform. This work was partially supported by ONR grant N00014-98-1-0242.

References

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

[2] D. Beymer and T. Poggio. Face recognition from one example view. Technical Report A.I. Memo, No. 1536, Artificial Intelligence Laboratory, M.I.T., 1995.

[3] D. Beymer, A. Shashua, and T. Poggio. Example based image analysis and synthesis. Technical Report A.I. Memo, No. 1431, Artificial Intelligence Laboratory, M.I.T., 1993.

[4] M. Bichsel and A. Pentland. Automatic interpretation of human head movements. Technical Report Technical Report No. 186, MIT Media Laboratory, Vision and Modeling Group, 1993.

[5] I. Biederman and P. Kalocsai. Neurocomputational bases of object and face recognition. *Philosophical Transactions of the Royal Society: Biological Sciences*, 352, 1997. 1203–1219.

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

[7] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of Siggraph*, pages 187–194, 1999.

[8] R. Brunelli. Estimation of pose and illuminant direction for face processing. Technical Report A.I. Memo, No. 1499, Artificial Intelligence Laboratory, M.I.T., 1994.

[9] 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.

[10] Q. Chen, H. Wu, T. Fukumoto, and M. Yachida. 3D head pose estimation without feature tracking. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 88–93, Nara, 1998.

[11] K. N. Choi, M. Carcassoni, and E. R. Hancock. Estimating 3D facial pose using the EM algorithm. In *Face Recognition: From Theory to Applications*, pages 412–423. Springer-Verlag, 1998.

[12] I. Craw, N. Costen, T. Kato, G. Robertson, and S. Akamatsu. Automatic face recognition: Combining configuration and texture. In *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, pages 53–58, Zurich, 1995.

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

[14] J. G. Daugman. Non-orthogonal wavelet representations in relaxation networks: image encoding and analysis with biological visual primitives. In *New Developments in Neural Computing*, pages 233–250. Institute of Physics Press, 1989.

[15] P. Demartines and J. Herault. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8:148–154, 1997.

[16] E. Elagin, J. Steffens, and H. Neven. Bunch graph matching technology. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 136–141, Nara, Japan, 1998.

[17] P. Fua. Regularized bundle-adjustment to model heads from images sequences without calibration data. *International Journal of Computer Vision*, 38(2):153–172, 2000.

[18] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[19] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Generative models for recognition under variable pose and illumination. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 277–284, Grenoble, France, 2000.

[20] Z. Ghahramani and M. I. Jordan. Learning from incomplete data. Technical Report A.I. Memo, No. 1509, Artificial Intelligence Laboratory, M.I.T., 1994.

[21] G. G. Gordon. 3D pose estimation of the face from video. In *Face Recognition: From Theory to Applications*, pages 433–455. Springer-Verlag, 1998.

[22] 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.

[23] D. B. Graham and N. M. Allinson. Face recognition from unfamiliar views: Subspace methods and pose dependency. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 348–353, 1998.

[24] B. Günter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *Proceedings of Siggraph*, pages 55–66, 1998.

[25] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84:502–516, 1989.

[26] J. Heinzmann and A. Zelinsky. 3D facial pose and gaze point estimation using a robust real-time tracking paradigm. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 142–147, Nara, 1998.

[27] G. T. Herman and K. T. D. Yeung. On piecewise-linear classification. *IEEE Transaction of Pattern Analysis and Machine Intelligence*, 14(7):782–786, July 1992.

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

[29] J. Huang, D. Ii, X. Shao, and H. Wechsler. Pose discrimination and eye detection using support vector machines (SVM). In *Face Recognition: From Theory to Applications*, pages 528–536. Springer-Verlag, 1998.

[30] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5:195–212, 1990.

[31] K. Isono and S. Akamatsu. A representation for 3D faces with better feature correspondence for image generation using PCA. Technical Report HIP96-17, The Institute of Electronics, Information and Communication Engineers, 1996.

[32] J. P. Jones and L. A. Palmer. An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58:1233–1258, 1987.

[33] 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.

[34] 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.

[35] M. Lando and S. Edelman. Generalization from a single view in face recognition. In *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, pages 80–85, Zurich, 1995.

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

[37] R. Lengagne, O. Monga, and P. Fua. Using differential constraints to generate a 3D face model from stereo. In *Face Recognition: From Theory to Applications*, pages 556–567. Springer-Verlag, 1998.

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

[39] T. Maurer and C. von der Malsburg. Learning features transformations to recognize faces rotated in depth. In *Proceedings of the International Conference on Artificial Neural Networks*, volume 1, pages 353–359, Paris, 1995.

[40] T. Maurer and C. von der Malsburg. Single-view based recognition of faces rotated in depth. In *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, pages 248–253, Zurich, 1995.

[41] T. Maurer and C. von der Malsburg. Tracking and learning graphs and pose on image sequences. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 176–181, Vermont, 1996.

[42] S. J. McKenna and S. Gong. Real-time face pose estimation. *Real-Time Imaging*, 4:333–347, 1998.

[43] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. In *Proceedings of Neural Information Processing Systems*, pages 39–43, 1998.

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

[45] U. Neumann, J. Li, R. Enciso, J.-Y. Noh, D. Fidaleo, and T.-Y. Kim. Constructing a realistic head animation mesh for a specific person. Technical Report 99-691, Computer Science Department, University of Southern California, 1999.

[46] J. Ng and S. Gong. Multi-view face detection and pose estimation using a composite support vector machine across the view sphere. In *Proceedings of the International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, pages 14–21, 1999.

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

[48] K. Okada, S. Akamatsu, and C. von der Malsburg. Analysis and synthesis of pose variations of human faces by a linear pcmap model and its application for pose-invariant face recognition system. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 142–149, Grenoble, France, 2000.

[49] 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.

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

[51] K. Okada and C. von der Malsburg. Pose-invariant face recognition: Representing known persons by view-based statistical models. Submitted to *Computer Vision and Image Understanding*, 2002.

[52] K. Okada and C. von der Malsburg. Pose-invariant face recognition with parametric linear subspaces. To appear in Proceedings of *the IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

[53] K. Okada, C. von der Malsburg, and S. Akamatsu. A pose-invariant face recognition system using linear pcmap model. In *Proceedings of IEICE Workshop of Human Information Processing*, pages 7–12, Okinawa, Japan, November 1999.

[54] F. I. Parke. Computer generated animation of faces. In *Proceedings of the ACM Annual Conference*, 1972.

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

[56] P. J. Phillips, H. Moon, S. Rizvi, and P. Rauss. The FERET evaluation. In *Face Recognition: From Theory to Applications*, pages 244–261. Springer-Verlag, 1998.

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

[58] P. J. Phillips, P. Rauss, and S. Z. Der. FERET (face recognition technology) recognition algorithm development and test results. Technical report, US Army Research Laboratory, 1996.

[59] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and H. Salesin. Synthesizing realistic facial expressions from photographs. In *Proceedings of Siggraph*, pages 75–84, 1998.

[60] T. Poggio and F. Girosi. A theory of networks for approximation and learning. Technical Report A.I. Memo, No. 1140, Artificial Intelligence Laboratory, M.I.T., 1989.

[61] T. Poggio and T. Vetter. Recognition and structure from one 2D model view: Observations on prototypes, object classes and symmetries. Technical Report A.I. Memo, No. 1347, Artificial Intelligence Laboratory, M.I.T., 1992.

[62] M. Pötzsch, T. Maurer, L. Wiskott, and C. von der Malsburg. Reconstruction from graphs labeled with responses of Gabor filters. In *Proceedings of the International Conference of Artificial Neural Networks*, pages 845–850, Bochum, 1996.

[63] 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.

[64] M. Rinne, M. Pötzsch, C. Eckes, and C. von der Malsburg. Designing objects for computer vision: The backbone of the library FLAVOR. Technical Report IRINI 99/08, Institut fur Neuroinformatik, Ruhr-Universität Bochum, 1999.

[65] A. Samal and P. A. Iyengar. Automatic recognition and analysis of human faces and facial expression: a survey. *Pattern Recognition*, 25:65–77, 1992.

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

[67] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Technical Report TR44, Max-Plank-Institut fur Biologische Kybernetik, 1996.

[68] J. Sherrah and S. Gong. Fusion of 2D face alignment and 3D head pose estimation for robust and real-time performance. In *Proceedings of the International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, pages 26–27, 1999.

[69] I. Shimizu, Z. Zhang, S. Akamatsu, and K. Deguchi. Head pose determination from one image using a generic model. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 100–105, Nara, 1998.

[70] 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.

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

[72] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

[73] N. F. Troje and H. H. Bülthoff. Face recognition under varying poses: The role of texture and shape. *Vision Research*, 36:1761–1771, 1996.

[74] A. Tsukamoto, C.-W. Lee, and S. Tsuji. Detection and pose estimation of human face with synthesized image models. In *Proceedings of the International Conference on Pattern Recognition*, pages 754–757, 1994.

[75] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

[76] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:992–1006, 1991.

[77] 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.

[78] K. Venkataraman and T. Poston. Piece-wise linear morphing and rendering with 3D textures. *Computer Networks and ISDN Systems*, 29:1625–1633, 1997.

[79] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:733–742, 1997.

[80] J. Walter. *Rapid Learning in Robotics*. PhD thesis, University of Bielefeld, 1996.

[81] J. Wieghardt and C. von der Malsburg. Pose-independent object representation by 2-D views. In *Proceedings of the IEEE International Workshop on Biologically Motivated Computer Vision*, May 2000. submitted.

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

[83] Y. Wu and K. Toyama. Wide-range, person- and illumination-insensitive head orientation estimation. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 183–188, Grenoble, France, 2000.

[84] M. Xu and T. Akatsuka. Detecting head pose from stereo image sequence for active face recognition. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 82–87, Nara, 1998.

[85] W. Y. Zhao and R. Chellappa. SFS based view synthesis for robust face recognition. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 285–292, Grenoble, France, 2000.