In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages I:761-768, Kauai, 2001.

Analysis and Synthesis of Human Faces with Pose Variations by a Parametric Piecewise Linear Subspace Method

Kazunori Okada †
†Computer Science Department
University of Southern California
Los Angeles, CA 90089-2520

Christoph von der Malsburg †‡
‡Institut für Neuroinformatik
Ruhr-Universität Bochum
Bochum, D-44801 Germany

Abstract

A framework for learning an accurate and general parametric facial model from 2D images is proposed and its application for analyzing and synthesizing facial images with pose variation is demonstrated. Our parametric piecewise linear subspace method covers a wide range of pose variation in a continuous manner through a weighted linear combination of local linear models distributed in a pose parameter space. The linear design helps to avoid typical non-linear pitfalls such as overfitting and time-consuming learning. Experimental results show sub-degree and sub-pixel accuracy within ±55 degree full 3D rotation and good generalization capability over unknown head poses when learned and tested for specific persons.

1. Introduction

2D images of faces change their appearance due to variations of both intrinsic property of faces and extrinsic condition of surrounding environment. These variations are entangled, and are encoded *implicitly* in the images. Only after disambiguating these variations and making them *explicit*, does the extrinsic variation source information become available for correct understanding and intuitive manipulation of the face (e.g., pose and expression) and the environment (e.g., illumination) in 2D images.

One solution is to parameterize facial images directly by explicit variation information. Such parameterization enables explicit *analysis* of variation source information in the images and *synthesis* of images with specific variation conditions. These processes can be formalized as bidirectional multivariate mapping functions that directly associate vector representations of facial images with their corresponding variation parameters. This study focuses on head pose variation, among other types of variation. Accordingly, the analysis mapping function provides a means for *pose estimation*; the synthesis function provides a procedure for *pose transformation* or *facial animation*.

This paper presents a novel framework for realizing this parameterization, which meets three design criteria: *extendibility*, *accuracy* and *simplicity*.

An extendible system is the one which can account for multiple variation sources. For pose variation, many previous studies [6, 11, 19] demonstrated variation-specific solutions utilizing analytical knowledge of 3D Euclidean rotation and assuming availability of 3D facial structure information. Our framework avoids this pitfall by *learning* the mapping functions solely from 2D sample-statistics instead of manually formulating the functions from variation-specific knowledge. This *data-driven* approach will, however, face the *curse of dimensionality problem* [3] as variation parameter dimensionality increases. A fundamental solution to this problem is *generalization* which alleviates the necessity to populate the entire dot-product space of the parameters. Our framework takes a *linear* functional form to emphasize this generalization capability.

Accuracy is another important criterion when considering the practical usefulness of a system. Most previous studies of facial pose variation [1, 8, 9, 12, 20] failed to achieve high accuracy because they treated the continuous variation *discretely*. Such representations require a prohibitively large number of samples to cover the variation smoothly. Our framework avoids this shortcoming by using *continuous* mapping functions, given by the framework's parameterization nature, instead of the discrete one.

The simplicity criterion is also supported by the linearity of our framework. While we choose a linear functional form for emphasizing the generalization, a non-linear form may be used to increase accuracy, given the intrinsic non-linearity of pose variations. Such a non-linear solution, however, complicates learning, re-

quiring time-consuming iterative processes, and faces *overfitting*, thus compromising its generalization capability. Our framework utilizes a linear form, which not only facilitates the extendibility but also simplifies the learning process and avoids overfitting. A shortcoming of this linear framework is *oversmoothing* [5], which decreases accuracy. Therefore, a system must be designed carefully to meet the accuracy criterion.

The *LPCMAP model* [14] is an implementation of the above framework for the pose variation problem. This data-driven, parametric, continuous, linear model has demonstrated good generalization capability over unknown head poses. However, the model was shown to be accurate only within a limited head pose range because of the oversmoothing caused by the framework's linearity. This shortcoming is not tolerable since it severely limits the model's overall accuracy and practical usability.

In this paper, we propose the *Parametric Piecewise Linear Subspace Method* (PPLS), which overcomes the accuracy range problem of the LPCMAP model while maintaining its positive properties. The method uses a set of LPCMAP models as *local linear models* which collectively cover the non-linear data variations. PPLS improves overall accuracy by continuously covering a wide range of pose variation, even though each local model can be accurate only within a small parameter-range. PPLS also maintains the model's linearity, thus avoiding the pitfalls of a non-linear function approximation and facilitating its simplicity and extendibility to other types of variation.

Our proposed method is related to a number of previous studies. Beymer [2] presented an analysis and synthesis mapping system using RBF networks. Murase and Nayar [13] proposed the parametric eigenspace method for generic objects with pose and illumination variations. Although both methods realize a continuous parameterization of facial and object images, their generalization capability is questionable due to their non-linearity. In the past, the piecewise linear approach has been used in various domains [18, 22]. Pentland's modular eigenface method [15] utilized local linear models in the form of a linear subspace spanned by principal components similar to the LPCMAP model. However, this work did not address the continuous parametric subspace of our focus. Overall, most studies did not conduct systematic accuracy evaluations, especially for generalization. Only few reported quantitative pose estimation accuracy, the best of which achieved 3 degree error on average [4]. Moreover, no studies to our knowledge demonstrated a system for full 3D rotation within a wide head pose range.

2. Parametric Piecewise Linear Subspace Method

The parametric piecewise linear subspace (PPLS) method consists of a set of local linear models, each of which realizes 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 continuously, 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.

2.1. Problem Definition

Let a pair of vectors $(\vec{v}^m, \vec{\theta}^m)$ denote 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 M given training samples $\{(\vec{v}^m, \vec{\theta}^m) | m = 1, .., M\}$,

$$\begin{array}{l} \mathcal{A} : \vec{v} \mapsto \vec{\theta} \\ \mathcal{S} : \vec{\theta} \mapsto \vec{v} \end{array} . \tag{1}$$

We call $\mathcal{A}$ an *analysis* mapping and $\mathcal{S}$ a *synthesis* mapping. Given an arbitrary facial image $\vec{v} \notin \{\vec{v}^1, .., \vec{v}^M\}$, $\mathcal{A}$ provides a 3D head angle estimate $\hat{\vec{\theta}} = \mathcal{A}(\vec{v})$ of a face in $\vec{v}$. On the other hand, given an arbitrary 3D head angle $\vec{\theta} \notin \{\vec{\theta}^1, .., \vec{\theta}^M\}$, $\mathcal{S}$ provides a synthesized sample or model view $\hat{\vec{v}} = \mathcal{S}(\vec{\theta})$ 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.

2.2. Local Linear Model

The local linear model is implemented by the LPCMAP model [14]. It provides bidirectional, continuous mapping functions between facial images and their corresponding 3D head angles. Each function consists of a combination of two linear systems: 1) *linear subspaces* spanned by principal components (PCs) derived from training samples and 2) *linear transfer matrices*, which associate projection coefficients of

training samples onto the subspaces and their corresponding 3D pose angles.

The model treats shape and texture information separately. After locating N landmarks in each facial image $\vec{v}^m$ by a landmark finder or other means, shape and texture representations are extracted from the image. The shape representation stands for object-centered 2D coordinates of the N landmarks while the texture representation stands for a set of N Gabor jets sampled at the N landmarks [21]. Let $2N$-component vector $\vec{x}^m$ and a set of L-component vectors $\{\vec{j}^{m,n}|n = 1,..,N\}$ denote the shape and texture representation of the $\vec{v}^m$, respectively. $\mathcal{D}_x$ and $\mathcal{D}_j$ denote operations of the shape and texture decomposition,

$$\begin{aligned} \mathcal{D}_x(\vec{v}^m) &= \vec{x}^m, \\ \mathcal{D}_j(\vec{v}^m) &= \vec{j}^{m,1},..,\vec{j}^{m,n},..,\vec{j}^{m,N}, \end{aligned} \tag{2}$$

and $\mathcal{R}$ denotes an operation of reconstructing an image $\vec{v}$ from shape and texture representations $\vec{x}, \{\vec{j}^n\}$ in the form of a Gabor jet graph representation [21] using an algorithm developed by Pötzsch [17],

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

A LPCMAP model LM includes the following data entities learned from training samples,

$$LM = \{\vec{u}_x, \{\vec{u}_j^n\}, \vec{u}_\theta, Y, \{B^n\}, F, G, \{H^n\}\}, \tag{4}$$

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 is a *shape model* represented as a row matrix of the first $P_0 \leq 2N$ shape PCs, $B^1,..,B^N$ are *texture models* represented as row matrices of the first $S_0 \leq L$ texture PCs, and $F, G, H^1,..,H^N$ are shape-to-pose, pose-to-shape and shape-to-texture transfer matrices, respectively.

Relating the 3D head angles only to the shape representations, the analysis mapping function $\mathcal{A}$ is given by,

$$\hat{\vec{\theta}} = \mathcal{A}(\vec{v}) = \mathcal{K}^{-1}(F \cdot Y \cdot (\mathcal{D}_x(\vec{v}) - \vec{u}_x)), \tag{5}$$

where $\mathcal{K}^{-1}$ extracts 3D angles from their trigonometric functions, and the shape synthesis mapping function $\mathcal{SS}$ is given by,

$$\hat{\vec{x}} = \mathcal{SS}(\vec{\theta}) = \vec{u}_x + Y^t \cdot G \cdot \mathcal{K}(\vec{\theta}), \tag{6}$$

where $\mathcal{K}$ transforms 3D angles to *pose parameters*, a vector of trigonometric functions of the angles. The texture synthesis mapping function $\mathcal{TS}$ is given by synthesizing texture from the synthesized shape,

$$\begin{aligned} \{\hat{\vec{j}}^n|n = 1,..,N\} = \mathcal{TS}(\vec{\theta}) = \\ \{\vec{u}_j^n + B^n \cdot H^n \cdot G \cdot \mathcal{K}(\vec{\theta})|n = 1,..,N\}. \end{aligned} \tag{7}$$

Finally, the synthesis mapping function $\mathcal{S}$ is given by,

$$\hat{\vec{v}} = \mathcal{S}(\vec{\theta}) = \mathcal{R}(\mathcal{SS}(\vec{\theta}), \mathcal{TS}(\vec{\theta})). \tag{8}$$

2.3. Global Piecewise System

2.3.1. Weighted Linear Combination

Suppose K local linear models $\{LM_1,..,LM_k,..,LM_K\}$ are learned from local training sample sets, each of which includes samples within a limited pose range. We assume that average 3D head angles of each local training set $\vec{u}_\theta^{LM_k}$ are appropriately distanced from each other so that the local models cover a wide range of a 3D parameter space spanned by the head angles. We call this parameter space *3D angle space* and the $\vec{u}_\theta^{LM_k}$, the *model center* of LM_k.

The global analysis mapping function is given by averaging K local pose estimates with appropriate weights,

$$\hat{\vec{\theta}} = \sum_{k=1}^{K} w_k \hat{\vec{\theta}}_k = \sum_{k=1}^{K} w_k \mathcal{A}_{LM_k}(\vec{v}). \tag{9}$$

Similarly, the global synthesis mapping function is given by averaging K locally synthesized samples with the same weights,

$$\begin{aligned} \hat{\vec{v}} &= \mathcal{R}(\hat{\vec{x}}, \{\hat{\vec{j}}^n\}) \\ \hat{\vec{x}} &= \textstyle\sum_{k=1}^{K} w_k \hat{\vec{x}}_k = \sum_{k=1}^{K} w_k \mathcal{SS}_{LM_k}(\vec{\theta}), \\ \{\hat{\vec{j}}^n\} &= \{\textstyle\sum_{k=1}^{K} w_k \hat{\vec{j}}_k^n\} = \sum_{k=1}^{K} w_k \mathcal{TS}_{LM_k}(\vec{\theta}). \end{aligned} \tag{10}$$

Functions (9) and (10) require a weight vector $\vec{w} = (w_1,..,w_k...,w_K)$ to be set with appropriate values. These weights must be responsible for localizing the model's outputs because the LPCMAP model covers the whole 3D angle space despite the fact that it is accurate only within a limited pose range. For this purpose, we use a normalized Gaussian weight function in the 3D angle space,

$$\begin{aligned} w_k(\vec{\theta}) &= \frac{\rho_k(\vec{\theta} - \vec{u}_\theta^{LM_k})}{\sum_{k=1}^{K} \rho_k(\vec{\theta} - \vec{u}_\theta^{LM_k})}, \\ \rho_k(\vec{\theta}) &= \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{\|\vec{\theta}\|^2}{2\sigma_k^2}), \end{aligned} \tag{11}$$

where σ_k denotes the k-th Gaussian width. Function (11) computes the weights as a function of distance between an input pose and each model center. A weight value reaches its maximum when the input pose coincides with one of the model centers; it decays as the distance increases. The Gaussian width σ_k is set by the standard deviation of 3D head angle vectors in training samples for LM_k and determines the extent to which each local model influences the global outputs $\hat{\vec{\theta}}$ and $\hat{\vec{v}}$.

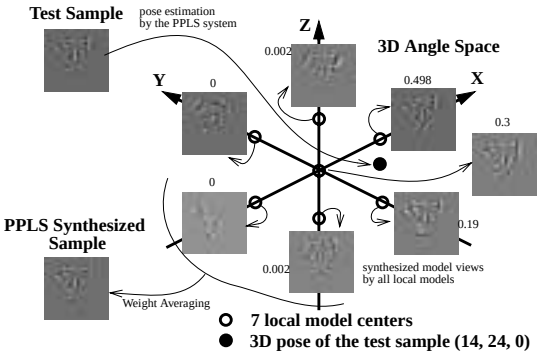

Figure 1. Sketch of the global piecewise system.

We formulate an *analysis-synthesis-chain* function by connecting an analysis output to a synthesis input,

$$\begin{aligned} \hat{\vec{v}} &= \mathcal{R}(\hat{\vec{x}}, \{\hat{\vec{j}}^n\}) \\ \hat{\vec{x}} &= \textstyle\sum_{k=1}^{K} w_k \mathcal{SS}_{LM_k}(\sum_{k=1}^{K} w_k \mathcal{A}_{LM_k}(\vec{v})), \\ \{\hat{\vec{j}}^n\} &= \textstyle\sum_{k=1}^{K} w_k \mathcal{TS}_{LM_k}(\sum_{k=1}^{K} w_k \mathcal{A}_{LM_k}(\vec{v})). \end{aligned} \tag{12}$$

This auto-associative mapping realizes a process of fitting a learned model to an arbitrary input face, resulting in a facial image whose pose is aligned to the input and whose appearance is derived from the model.

Figure 1 illustrates the global piecewise system. The numbers next to local model views are weight values for the test input. Note that the local views become more distorted as their model centers deviate further from an input pose, illustrating the pose range limitation of our local model. However, these largely distorted local outputs do not greatly influence a global output because their contribution is strongly inhibited by relatively low weight values.

2.3.2. Gradient Descent System

The global analysis mapping function (9) cannot provide a pose estimate by evaluating its r.h.s. because the weights are computed as a function of an unknown $\vec{\theta}$. Next, we formulate a gradient descent-based algorithm which solves this problem by using a pose estimate from a previous iteration step.

Let a shape vector $\vec{x}$ be an input to this iterative algorithm. Let a shape vector $\vec{x}_i$ and a 3D angle vector $\vec{\theta}_i$ denote the shape and angle estimates by the i-th iteration. In order to find an initial condition $\vec{x}_0$ and $\vec{\theta}_0$, we first find a local model whose center shape $\vec{u}_x^{LM_k}$ is most similar to $\vec{x}$. Then, $\vec{x}_0$ and $\vec{\theta}_0$ are set by,

$$\begin{aligned} \vec{x}_0 &= \vec{u}_x^{LM_{k_{min}}}, \\ \vec{\theta}_0 &= \vec{u}_\theta^{LM_{k_{min}}}, \\ k_{min} &= index(\min_{k=1}^{K} \|\vec{x} - \vec{u}_x^{LM_k}\|^2). \end{aligned} \tag{13}$$

The following defines iteration rules of the algorithm,

$$\Delta\vec{x}_i = \vec{x} - \vec{x}_i, \tag{14}$$

$$\Delta\vec{\theta}_i = \sum_{k=1}^{K} w_k(\vec{\theta}_i)\mathcal{A}'_{LM_k}(\Delta\vec{x}_i), \tag{15}$$

$$\vec{\theta}_{i+1} = \vec{\theta}_i + \eta\Delta\vec{\theta}_i, \tag{16}$$

$$\vec{x}_{i+1} = \sum_{k=1}^{K} w_k(\vec{\theta}_{i+1})\mathcal{SS}_{LM_k}(\vec{\theta}_{i+1}), \tag{17}$$

where η is the learning rate and $\mathcal{A}'$ is a slight modification of (5) that has a shape vector interface. This algorithm iterates through the loop of equations from (14) to (17) until the mean-square error $\|\Delta\vec{x}_i\|^2$ becomes sufficiently small.

Note that the shape-to-pose analysis mapping $\mathcal{A}'$ in (15) is used as an approximation of the gradients of $\vec{\theta}$ with respect to $\vec{x}_i$ at a current pose estimate $\vec{\theta}_i$. In PPLS, such gradients $\frac{\Delta\vec{\theta}}{\Delta\vec{x}}$ are only available at the locations of K discrete model centers. Therefore, (15) interpolates the K local gradient matrices to compute gradients at an arbitrary point. Note also that this algorithm performs pose estimation and shape synthesis simultaneously since it iterates between pose and shape in each loop. This gives an alternative way for the shape synthesis although the global synthesis mapping in (10) remains valid.

2.3.3. Self-Occlusion Handling

As a head rotates, some landmarks 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 pose range, in which such occlusion occurs naturally.

This problem suffers principal component analysis (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 sample mean and variance, cannot be computed correctly from such incomplete data. This problem is known as the *missing data problem* [10].

We handled this problem by applying the *mean-imputation method* [10] which fills in each missing component by a mean computed from all available data at the component dimension. 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

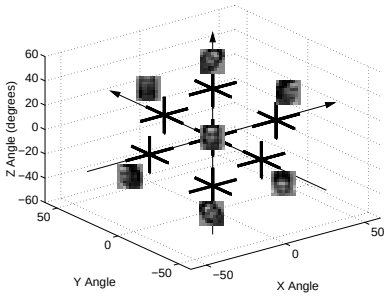

Figure 2. Seven local training sample sets.

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.

3. Toy Data Experiments

In order to assess our method's correctness and to investigate its optimal setting, an implementation of the PPLS, which we call the *PPLS system*, is evaluated with a toy data set.

3.1. Data Set

We created artificial shape representations, each of which consists of 2D orthographic projections of 25 3D landmark points on a 5 by 5 square grid, pasted onto the surface of a rotating 3D unit sphere. 2D coordinates of the projected points are scaled and translated for fitting 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. The toy data differ from realistic facial data in that their depth profile is much more regular than that of faces and that there are no measurement errors of landmark locations and rotation angles.

As training samples, we created 7 local training sample sets distributed over the 3D angle space as illustrated in figure 2. Around each of 7 model centers, (0,0,0),($\pm$40,0,0),(0,$\pm$40,0), and (0,0,$\pm$40), we rotate the sphere along each rotation axis at a time and two axes simultaneously within $\pm$15 degrees from the center. 403 samples are recorded in one degree intervals for each local set. In total, there are 2831 training samples, which cover a range of $\pm$55 degree 3D rotation. Note that figure 2 uses facial images for describing the model centers, instead of the actual toy data, for facilitating an intuitive understanding of the different rotation angles. As test samples, we created 804 samples whose rotation angles are different from those of the training samples. This test sample set covers a range of $\pm$50 degree 3D rotation. It includes two types of pose distribution: one is between several local training sets and the other is within a sparsely populated region of a local set. The former poses a more difficult testing situation than the latter, which requires a smooth interpolation between neighboring local models. Landmark self-occlusion is simulated by introducing an occluding plane, $z = c$ (c: constant, $\|c\| \leq 1$), which is parallel to the image plane. A landmark point is considered as occluded when it goes below the occluding plane.

3.2. Results

Two types of test are used for evaluating the PPLS system. An *accuracy test* evaluates the system's accuracy by testing a learned system with the training samples; a *generalization test* evaluates the system's generalization by testing the system with the test samples described above.

First, we studied the average errors of the one-shot pose estimation (9) and shape synthesis (10) in the most controlled condition, in which all landmarks are considered to be visible ($c = -1$), shape vectors have float precision, and the trigonometric transformation $\mathcal{K}$ includes pairwise products of the trigonometric functions. Results of the accuracy test showed that average pose estimation error in degrees over 3 rotation dimensions and 2831 samples, and average shape synthesis error in pixels over 25 landmarks and 2831 samples became approximately zero after including the first 6 shape PCs (12%). This result strongly supports our system's correctness. The difference of the errors between the float and integer precision was small, indicating the system's robustness against small measurement errors in landmark locations. This system setting, however, resulted in overfitting with poor performance for the generalization test. $\mathcal{K}$, without the pairwise products, provided the best balance between the system's accuracy and generalization.

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 ($c = -1$) occlusion. At most, 10% of the total landmarks in a local set were occluded in the occlusion data set. Results of the generalization test with integer precision showed that the error difference of the two data sets was very small (0.2 degrees and 0.1 pixels with the first 8 PCs), supporting the effectiveness of our missing data handling by the mean-imputation method. The average errors were roughly 0.7 degrees and 1.1 pixels for pose estimation and shape synthesis.

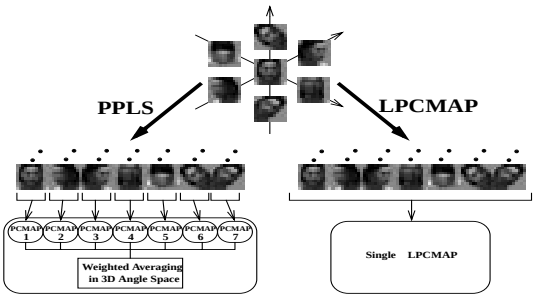

Figure 3. PPLS and LPCMAP systems.

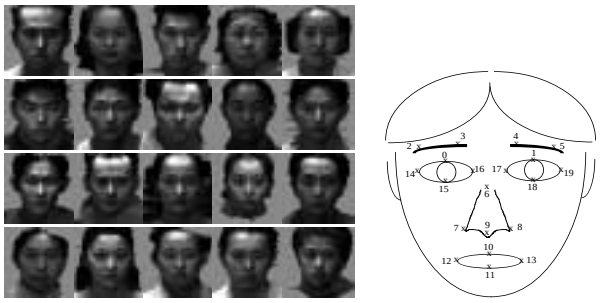

Figure 4. 3D face models and facial landmarks.

Lastly, we evaluated the gradient descent iterative algorithm in section 2.3.2 in the most realistic conditions with integer accuracy and 10% landmark self-occlusion. We iterated the gradient descent loop 500 times and set the learning rate η to 0.01. Results of the generalization test were compared with those of the one-shot system. They showed that the error difference of the two systems was again very small (0.1 degrees and 0.1 pixels), supporting the feasibility of our system. The average errors were roughly 0.8 degrees and 1.0 pixels, indicating good accuracy and generalization.

4. Cyberware Scanned Face Data Experiments

In order to assess our system's feasibility in more realistic scenarios, the PPLS system is evaluated 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 scanner. Given such data, relatively faithful image samples with arbitrary, but precise, head poses can easily be created by image rendering [7]. In order to assess our method's improvement in performance, we compare the PPLS and LPCMAP systems learned from the same training samples. The former consists of 7 local models, while the latter is a single local model learned from the total training samples, as shown in figure 3.

4.1. Data Set

We used 20 face models randomly picked from the ATR-Database [7], as shown in figure 4. The same pose distributions used for the toy data experiments in the previous section are also used for creating training and test sample sets of these experiments. As a result, for each individual, we have 804 test samples and 2821 training samples consisting of 7 local training sets, each of which includes 403 samples. The test set, total training set and each local training set covers a pose range of ± 50, ± 55 and ± 15 degrees along each rotation axis, respectively. Figure 4 also shows the definition of the 20 facial landmarks. These landmarks were manually placed on the surface of the 3D model. For each 2D sample, the 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.

4.2. Results

For the following experiments, both PPLS and LPCMAP systems use integer shape precision, $\mathcal{K}$ without the pairwise products, and the gradient descent system with 500 iterations and η set to 0.01.

4.2.1. Average Error and Similarity Analyses

Figure 5 compares average pose estimation errors of the PPLS and LPCMAP systems in both accuracy and generalization tests. The errors were averaged over 3 pose dimensions, 804 samples and 20 persons for 6 different shape model sizes. The up- and down-triangles denote the errors of the PPLS and LPCMAP systems, respectively. In the accuracy test, the average error with the first 8 shape PCs was 0.8 ± 0.6 and 3.0 ± 2.4 degrees for the PPLS and LPCMAP systems, respectively. In the generalization test, the average was 0.9 ± 0.6 and 2.4 ± 1.4 degrees for the two systems.

Figure 6 compares average shape synthesis errors of the two systems in the two test cases. In the accuracy test, the average error with the first 8 PCs was 0.8 ± 0.4 and 2.2 ± 1.2 pixels for the PPLS and LPCMAP systems, respectively. In the generalization test, the average was 0.9 ± 0.4 and 2.4 ± 0.7 pixels for the two systems.

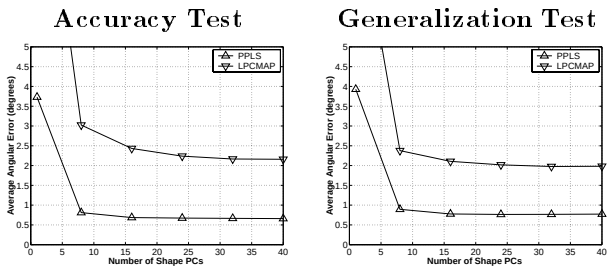

Figure 5. Pose estimation errors in degrees.

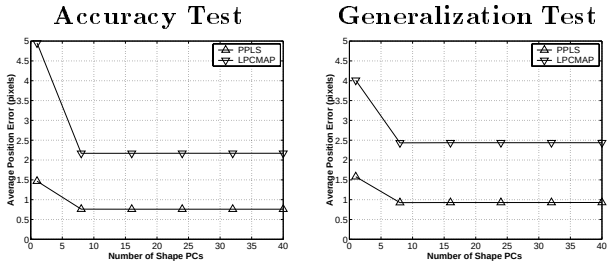

Figure 6. Shape synthesis errors in pixels.

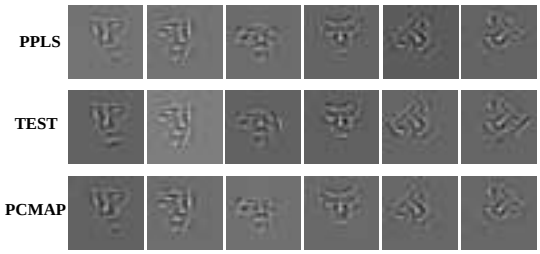

Figure 7. Synthesized test samples (large angle).

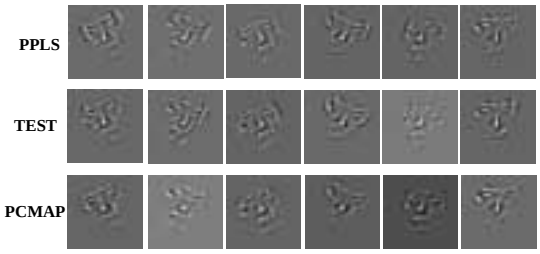

Figure 8. Synthesized test samples (far from center).

Average similarities between synthesized and ground-truth textures are also studied for the two systems in the two test cases. Texture similarity is computed as a normalized dot-product 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. These similarities are averaged over the number of landmarks, test samples, and persons. In the accuracy test, the average similarity with the first 20 texture PCs was 0.955 ± 0.03 and 0.91 ± 0.04 for the PPLS and LPCMAP systems, respectively. In the generalization test, the average was 0.945 ± 0.03 and 0.88 ± 0.03 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 average errors between the two test cases were similar, indicating good generalization to unknown poses. The errors with the toy data and the facial samples were also similar, suggesting our system's robustness against irregular depth variation of faces. As a reference, we computed average texture similarities over 450 people from the FERET database [16]. The 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 FERET database, which validates the results of our texture similarity analysis.

4.2.2. Synthesized Samples

Figures 7 and 8 illustrate model views of test samples whose head poses are not known to the learned system. The analysis-synthesis chain (12) was used to derive these model views. Their corresponding ground-truth is shown in the middle rows. Figure 7 shows test samples whose pose is close to a model center, but with a large angle along one dimension. In contrast, figure 8 shows samples whose pose is in-between several model centers. For both cases, the ground-truth and the PPLS's model views were very similar, indicating our system's successful generalization capability. The LPCMAP's model views were only slightly more distorted than those of the PPLS system. However, human visual perception may be insensitive to the difference, which is clearly shown in the similarity difference described in section 4.2.1.

5. Conclusion

We present a novel framework for parameterizing facial images continuously by their 3D head poses. Our method uses piecewise linear PC-based subspaces for realizing an accurate, general and simple solution to the problem of head pose estimation and facial image synthesis as a function of pose. An implementation of our framework has shown sub-degree and sub-pixel accuracy within ± 55 degree 3D head rotation,

while displaying good generalization to unknown poses. The data-driven and continuous nature of the proposed method 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 of multi-modal systems. We have recently extended this framework to accommodate interpersonal variations and to realize a pose-insensitive face identification system. This work, however, is outside the scope of this paper. The proposed system utilizes pixel-wise landmark locations for representing facial shape. In reality, finding landmark locations in facial images with arbitrary head pose is an ill-posed problem. We plan to apply the proposed framework to solve this problem. We also plan to extend our method to include other types of variation, such as illumination variation, in the future.

Acknowledgments

The authors thank Shigeru Akamatsu and Katsunori Isono for making their 3D face database available, and Chad Jenkins and Larry Kite for helpful comments. This study was partially supported by ONR grant N00014-98-1-0242.

References

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

[2] D. Beymer, A. Shashua, and T. Poggio. Example based image analysis and synthesis. Technical Report 1431, Artificial Intelligence Laboratory, M.I.T., 1993.

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

[4] 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.

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

[6] 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.

[7] K. Isono and S. Akamatsu. A representation for 3D faces with better feature correspondence for image generation using PCA. Technical Report HIP96-17, IEICE, 1996.

[8] 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.

[9] 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.

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

[11] 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.

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

[13] 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.

[14] K. Okada, S. Akamatsu, and C. von der Malsburg. Analysis and synthesis of pose variations of human faces by a linear pcmap model. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 142–149, Grenoble, 2000.

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

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

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] M.-H. Yang, N. Ahuja, and D. Kriegman. Face detection using mixtures of linear subspaces. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 70–76, Grenoble, 2000.