Capturing Appearance Variation in Active Appearance Models

Laurens van der Maaten
Delft University of Technology / UC San Diego
lvdmaaten@gmail.com

Emile Hendriks
Delft University of Technology
E.A.Hendriks@tudelft.nl

Abstract

The paper presents an extension of active appearance models (AAMs) that is better capable of dealing with the large variation in face appearance that is encountered in large multi-person face data sets. Instead of the traditional PCA-based texture model, our extended AAM employs a mixture of probabilistic PCA to describe texture variation, leading to a richer model. The resulting extended AAM can be efficiently fitted to held-out test images using an adapted version of the inverse compositional algorithm: the computational complexity scales linearly with the number of components in the texture mixture. The results of our experiments on three face data sets illustrate the merits of our extended AAM.

1. Introduction

Active appearance models (AAMs) form a powerful approach to important computer vision problems such as deformable shape segmentation and appearance modeling of deformable objects [6, 8, 17]. AAMs do so by combining two separate models: (1) a model describing the shape deformations of the objects and (2) a model describing the object texture after shape normalization. As a result, AAMs can represent the complex interactions between object shape and object texture. Next to applications in face modeling, AAMs have also been successfully applied to, e.g., medical image analysis [1] and industrial vision problems [18]. Hitherto, studies on how to fit AAMs to images can roughly be subdivided into two main approaches.

The first approach, which is sometimes referred to as the *discriminative* approach, attempts to iteratively update the parameters of the AAM by training an update model on a data set of annotated images, in which the annotations are randomly perturbed [27]. The update model is trained as to learn a mapping from the feature space to the parameter update space. In the discriminative approach, both linear techniques [7, 8, 13] and non-linear techniques [15, 22] have been explored. The main advantage of discriminative approaches is that the fitting is relatively fast, because the function from the feature space to the parameter update space is fixed. However, the quality of the fits is hampered by the fixed, heuristically chosen, often linear parameter update scheme used in the discriminative approach.

The second approach, which is sometimes referred to as the *generative* approach, considers the fitting of AAMs as an image alignment problem that can be solved by minimizing the squared error between the observed image and the model fit [3, 17, 20], thereby assuming a Gaussian noise model. The generative approach does not require the heuristics of the discriminative approach. Instead, AAM fitting in the generative approach amounts to gradient-based maximization of a likelihood function using optimization techniques that are well understood. A disadvantage of the generative approach is that these gradient-based approaches are often relatively slow. A notable exception is a variant of the *inverse compositional* algorithm [2] that projects out texture variation [17]. The fitting scheme presented in [17] generally gives good results at low computational costs, however, its performance is often hampered by the presence of large texture variations in the data [10, 20].

To address this problem of the project-out inverse compositional algorithm, we present an extension of the standard AAM that is better at dealing with large texture variations in the data, but that can still be fitted efficiently. The extension entails the use of a mixture model to describe object texture variation. The results of experiments on three multi-person face data sets show that our extended model outperforms the standard AAM thanks to its richer texture model (both in terms of shape fit as in terms of texture fit).

The outline of the remainder of the paper is as follows. In section 2, we present a review of AAMs, and we discuss the main limitations of the standard model. Section 3 presents our extension of the AAM that aims to increase the model's ability to deal with large texture variations in the image data. In section 4, we describe how the extended model can be fitted efficiently to test images. Section 5 presents experiments with the extended AAMs on three multi-person face data sets. In section 6, we discuss the results of the experiments. Section 7 presents our conclusions, as well as directions for future work.

2. Active appearance models

AAMs simultaneously describe the shape and texture variation of objects [6, 17]. In the remainder of this paper, we assume that the objects are faces of various individuals exhibiting a range of facial expressions. In this case, AAMs are trained on a collection of face images, in which facial feature points are annotated. The feature points are required to be relatively dense, in such a way that, e.g., a Delaunay triangulation constructed on the feature point annotations approximately captures the geometry of the face. The feature point annotations are normalized for translation, rotation, and scale differences using Procrustes alignment, and subsequently, a model for the feature point locations is learned using PCA. The resulting shape model describes the variation in feature point locations across the faces in the training data. In addition, a base shape mesh $\boldsymbol{\nu}$ is constructed, typically, by computing the mean of the normalized feature point annotations.

The base shape mesh is used to normalize the face images in the data set. Every image in the training set is warped onto the base shape mesh $\boldsymbol{\nu}$ (e.g., using a piecewise linear or thin-plate spline warp) using the feature point annotations as control points. This results in a collection of aligned facial texture images that are all defined in the same coordinate frame. From this collection of texture images, a texture model is learned, again, using PCA.

In the instantiation of the AAM, the shape model and texture model are combined in three steps. First, the shape model is used to generate a face shape, i.e., to lay out the facial feature points. Second, the texture model is used to generate a facial texture image. Recall that this texture image is defined in the coordinate frame of the base mesh $\boldsymbol{\nu}$. Third, the texture image is warped onto the face shape using the constructed feature points as control points to construct the final face image.

Denoting the warp of image $\mathbf{x}$ that maps the control points $\mathbf{s}$ to the target points $\boldsymbol{\nu}$ by $W_{\mathbf{s}\rightarrow\boldsymbol{\nu}}(\mathbf{x})$, the generative model of the standard AAM can be written as follows:

- Sample shape parameters $\mathbf{p} \sim \mathcal{N}(\mathbf{p}|0, \mathbf{I})$.
- Sample shape points $\mathbf{s} \sim \mathcal{N}(\mathbf{s}|\boldsymbol{\nu} + \mathbf{S}\mathbf{p}, \tau^2\mathbf{I})$.
- Sample texture parameters $\boldsymbol{\lambda} \sim \mathcal{N}(\boldsymbol{\lambda}|0, \mathbf{I})$.
- Sample texture image $\mathbf{x} \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu} + \mathbf{A}\boldsymbol{\lambda}, \sigma^2\mathbf{I})$.
- Generate face image $\mathbf{i} = W_{\mathbf{s}\rightarrow\boldsymbol{\nu}}(\mathbf{x})$.

In the above, $\mathbf{S}$ represent the shape basis, $\mathbf{A}$ represents the texture basis, $\boldsymbol{\nu}$ and $\boldsymbol{\mu}$ represent the corresponding shape and texture means, τ and σ represent the corresponding noise variances, and $\mathbf{I}$ represents the identity matrix. The marginal distribution over the texture $\mathbf{x}$ is a Gaussian with a low-rank covariance matrix [25], viz. by

$$p(\mathbf{x}|\boldsymbol{\mu}, \mathbf{A}, \sigma) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{A}\mathbf{A}^T + \sigma^2\mathbf{I}).$$

We argue that this marginal distribution is too simple to appropriately model the facial texture variations in large multi-person face data sets, in particular, when these faces exhibit a range of different characteristics or facial expressions. A low-rank Gaussian in pixel space is unlikely to have sufficient modeling power to capture the complex non-linear manifold on (or near) which facial textures lie. Also, it is very unlikely that the texture distribution of real-world faces is unimodal. For instance, faces of individuals with different ethnicities may well form different modes in the texture distribution. As another example, it is likely that persons with and persons without glasses, or persons with and without beards form separate modes in the texture space. In fact, even gender differences may lead to different modes in the texture space, though one may argue there exists a smooth manifold transition from male faces to female faces (in which case, the data likely constitutes a complex non-linear manifold). As a result, facial texture cannot be appropriately modeled by a single low-rank Gaussian.

3. Modeling texture variation

To address the limitations of the texture model in standard AAMs, we propose to use a mixture of probabilistic principal component analyzers [24] instead of standard PCA to model the variation in the facial textures. The mixture of PCAs has been shown to be capable of modeling the structure of complex non-linear manifolds [4, 21], and can be trained relatively easy using an EM-algorithm. Moreover, the use of a mixture of PCA-based texture model still allows for the use the inverse compositional algorithm during inference (i.e., fitting), because the mixture components are Gaussian models.

The probabilistic PCA mixture model entails a linear superposition of K components, in which each component is a separate probabilistic PCA model. A mixture of probabilistic PCA model with K mixture components is governed by parameters $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{A}, \boldsymbol{\sigma}\}$, where we use the notation $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ for the weights of the mixture components, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ for the component means, $\mathbf{A} = \{\mathbf{A}_1, \ldots, \mathbf{A}_K\}$ for the component bases, and $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_K\}$ for the component noise terms. Using the PCA mixture to model texture variation leads to an extended AAM, the generative model of which is illustrated in Figure 1.

Using a 1-of-K representation for the latent assignment variable $\mathbf{z}$, the conditional distributions that correspond to the texture part of the model can be denoted by

$$\begin{aligned} p(\mathbf{z}|\boldsymbol{\pi}) &= Discrete(\boldsymbol{\pi}), \\ p(\boldsymbol{\lambda}_k) &= \mathcal{N}(\boldsymbol{\lambda}_k|0, \mathbf{I}), \\ p(\mathbf{x}|\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{A}, \boldsymbol{\sigma}) &= \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k + \mathbf{A}_k\boldsymbol{\lambda}_k, \sigma_k^2\mathbf{I})^{z_k}. \end{aligned}$$

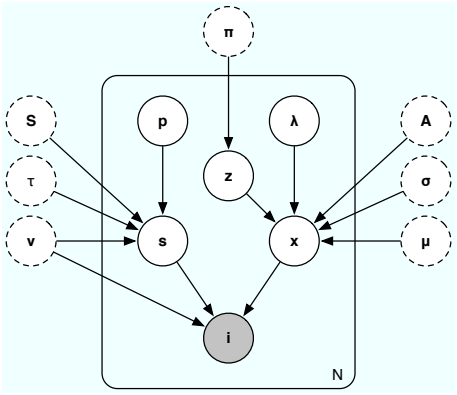


Figure 1. Generative model of the extended AAM.

The marginal distribution over the texture space can be obtained by marginalizing over $\mathbf{z}$ and over the latent spaces $\boldsymbol{\lambda}_k$. The resulting marginal is given by

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{A}_k\mathbf{A}_k^T + \sigma_k^2\mathbf{I}).$$

The mixture of PCAs can thus be thought of as a mixture of Gaussian model, in which the covariance matrices of the Gaussians are constrained to have a low (but not necessarily equal[1]) rank. As a result, the mixture of PCAs can represent more complicated distributions than the PCA-model used in the standard AAM. One can imagine that in a mixture of two PCAs, one component may be used to model the texture of faces with glasses, whereas the other component models the texture of faces without glasses. In this example, the weights of the mixture components would correspond to the ratio between faces with and faces without glasses in the image data. The mixture components may also be used to model different parts of the same complex, non-linear face manifold.

The parameters $\boldsymbol{\theta}$ of the texture model can be learned from a collection of training data $\mathcal{D} = \{\mathbf{i}_n, \mathbf{s}_n\}$, $(n \in \{1, \ldots, N\})$, by warping all face images $\mathbf{i}_n$ to the base shape via $\mathbf{x}_n = W_{\mathbf{s}_n \to \boldsymbol{\nu}}(\mathbf{i}_n)$, and using the EM-algorithm described in [24] on the resulting data set $\{\mathbf{x}_n\}$.

4. Inference

The key inferential problem in AAMs is, given an unseen test image $\mathbf{i}_{N+1}$, to determine parameters $\mathbf{p}^*$, $\boldsymbol{\lambda}^*$, and $\mathbf{z}^*$ that are optimal in terms of likelihood[2], i.e., to find

$$\{\mathbf{p}^*, \boldsymbol{\lambda}^*, \mathbf{z}^*\} = \arg\max_{\mathbf{p},\boldsymbol{\lambda},\mathbf{z}} \log p(\mathbf{i}_{N+1}|\mathbf{p}, \boldsymbol{\lambda}, \mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\zeta}),$$

[1] We would like to emphasize that the probabilistic PCA mixture model constructs K *different* latent spaces $\boldsymbol{\lambda}_k$, which do not even necessarily have the same dimensionality.

[2] We could also perform maximum a posteriori estimation [20], but for the purpose of our experiments, maximum likelihood estimation suffices.

where we used the notation $\boldsymbol{\zeta}$ for the parameters of the shape model, $\boldsymbol{\zeta} = \{\boldsymbol{\nu}, \mathbf{S}, \tau\}$. First, we note that the maximization over $\mathbf{z}$ is tractable as the number of possible configurations of $\mathbf{z}$ is equal to K. Hence, we can maximize the likelihood with respect to $\mathbf{z}$ by selecting the maximum value of K separate likelihood maximizations over $\{\mathbf{p}, \boldsymbol{\lambda}_k\}$ (for all $k \in \{1, \ldots, K\}$) in which we assume that $z_k = 1$.

Analytically computing the maximum likelihood estimate $\{\mathbf{p}^*, \boldsymbol{\lambda}_k^*\}$ for a given $\mathbf{z}$ is intractable due to the coupling between $\mathbf{s}$ and $\mathbf{x}$. Following earlier studies [17, 20], we circumvent this problem by performing the likelihood maximization with respect to $\mathbf{p}$ first, whilst we assume that $\boldsymbol{\lambda}_k$ is set to the zero vector. Once an optimal $\mathbf{p}$ is found, the value of $\mathbf{p}$ is fixed and the maximization with respect to $\boldsymbol{\lambda}_k$ is performed. We discuss the maximizations with respect to $\mathbf{p}$ and $\boldsymbol{\lambda}_k$ separately.

4.1. Finding the shape parameters

Because the mapping between the texture $\mathbf{x}$ and the appearance $\mathbf{i}$ is a deterministic one, we can evaluate the likelihood either in the texture space or in the appearance space. The noise model is easier to evaluate in the texture space, so we opt to evaluate the likelihood there. Taking into account that $\boldsymbol{\lambda}_k = \mathbf{0}$, we can thus rewrite the maximum likelihood estimate as

$$\begin{aligned}
\mathbf{p}^* &= \arg\max_{\mathbf{p}} \log p(\mathbf{i}_{N+1}|\mathbf{p}, \boldsymbol{\lambda}_k = \mathbf{0}, z_k = 1, \boldsymbol{\mu}_k, \mathbf{A}_k, \sigma_k, \boldsymbol{\zeta}) \\
&= \arg\max_{\mathbf{p}} \log \mathcal{N}(W_{\mathbf{s}\to\boldsymbol{\nu}}(\mathbf{i}_{N+1}; \mathbf{p})|\boldsymbol{\mu}_k, \sigma_k^2\mathbf{I}) \\
&= \arg\min_{\mathbf{p}} \sum \left(W_{\mathbf{s}\to\boldsymbol{\nu}}(\mathbf{i}_{N+1}; \mathbf{p}) - W_{\boldsymbol{\nu}\to\mathbf{t}}(\boldsymbol{\mu}_k; \mathbf{0})\right)^2,
\end{aligned}$$

where $\mathbf{t}$ is the mean shape that corresponds to the shape parameter vector $\mathbf{p} = \mathbf{0}$ (i.e., $\mathbf{t}$ is equal to the base shape $\boldsymbol{\nu}$), and the summation is over all pixels in the texture space. Note that the factor $(\sigma_k\mathbf{I})^{-1}$ in the evaluation of the Gaussian falls out, because the noise model is isotropic.

The likelihood maximization with respect to $\mathbf{p}$ can be performed using a standard gradient descent optimizer, but this is typically very slow. Instead, we opt to use the inverse compositional algorithm [2], which is an adaptation of the standard Lucas-Kanade algorithm that allows for most terms in the parameter update to be precomputed. The aim of the inverse compositional algorithm is to find a parameter update $\Delta\mathbf{p}^*$ that increases the likelihood by performing the following minimization

$$\min_{\Delta\mathbf{p}} \sum \left(W_{\mathbf{s}\to\boldsymbol{\nu}}(\mathbf{i}_{N+1}; \mathbf{p}) - W_{\boldsymbol{\nu}\to\mathbf{t}}(\boldsymbol{\mu}_k; \Delta\mathbf{p})\right)^2.$$

This minimization can be performed by evaluating the first-order Taylor series around $\mathbf{p} = \mathbf{0}$ of the likelihood at $\Delta\mathbf{p}$, leading to the approximation

$$\min_{\Delta\mathbf{p}} \sum \left(W_{\mathbf{s}\to\boldsymbol{\nu}}(\mathbf{i}_{N+1}; \mathbf{p}) - \boldsymbol{\mu}_k - \nabla\boldsymbol{\mu}_k \frac{\partial W}{\partial \mathbf{p}} \Delta\mathbf{p}\right)^2.$$

Herein, $\nabla \boldsymbol{\mu}_k$ represents the image gradient of the mean texture image $\boldsymbol{\mu}_k$, $\frac{\partial W}{\partial \mathbf{p}}$ is the Jacobian of the warp W around $\mathbf{0}$, and the summation is over the pixels in the texture space. The above minimization can be performed by setting the gradient of the first-order Taylor approximation to zero, which gives the solution

$$\Delta \mathbf{p}^* = \mathbf{H}^{-1} \sum \left[\nabla \boldsymbol{\mu}_k \frac{\partial W}{\partial \mathbf{p}} \right]^T \left[W_{\mathbf{s} \to \boldsymbol{\nu}}(\mathbf{i}_{N+1}; \mathbf{p}) - \boldsymbol{\mu}_k \right],$$

where $\mathbf{H} = \sum \left[\nabla \boldsymbol{\mu}_k \frac{\partial W}{\partial \mathbf{p}} \right]^T \left[\nabla \boldsymbol{\mu}_k \frac{\partial W}{\partial \mathbf{p}} \right]$ represents the Gauss-Newton approximation to the Hessian, and again, the summation is over the pixels in the texture space.

Due to the way the update objective is defined, the three main terms in the parameter update $\Delta \mathbf{p}^*$ can be precomputed. First, the warp Jacobian $\frac{\partial W}{\partial \mathbf{p}}$ can be precomputed because it is only evaluated at $\mathbf{p} = \mathbf{0}$. Second, the image derivative of the mean texture $\nabla \boldsymbol{\mu}_k$ can be precomputed because the mean textures $\boldsymbol{\mu}_k$ are fixed. Third, the inverse Hessian $\mathbf{H}^{-1}$ can also be precomputed, because it only contains precomputed terms.

The parameter update $\Delta \mathbf{p}^*$ is not the final update that is applied to the shape parameters, as it is an *inverse* parameter update: it warps $\boldsymbol{\mu}_k$ in the direction of $W_{\mathbf{s} \to \boldsymbol{\nu}}(\mathbf{i}_{N+1}; \mathbf{p})$ instead of warping $W_{\mathbf{s} \to \boldsymbol{\nu}}(\mathbf{i}_{N+1}; \mathbf{p})$ in the direction of $\boldsymbol{\mu}_k$ (hence the name inverse compositional algorithm). The computation of the final parameter update from $\Delta \mathbf{p}^*$ is straightforward, and is described in detail in [17].

4.2. Finding the texture parameters

Given the inferred shape parameters $\mathbf{p}^*$, the maximum likelihood estimate of the texture parameters $\boldsymbol{\lambda}_k$ can be found in closed form. The expression for the maximum likelihood estimate $\boldsymbol{\lambda}_k^*$ can be rewritten in a similar way as for the shape parameters, to find that $\boldsymbol{\lambda}_k^*$ can be obtained by performing the minimization

$$\boldsymbol{\lambda}_k^* = \arg\min_{\boldsymbol{\lambda}_k} \sum \left(W_{\mathbf{s} \to \boldsymbol{\nu}}(\mathbf{i}_{N+1}; \mathbf{p}^*) - \boldsymbol{\mu}_k - \mathbf{A}_k \boldsymbol{\lambda}_k \right)^2,$$

where $\mathbf{s}$ is the mean shape that corresponds to the shape parameters $\mathbf{p}^*$. The closed-form solution for the texture parameters is

$$\boldsymbol{\lambda}_k^* = \mathbf{A}_k^T \left(W_{\mathbf{s} \to \boldsymbol{\nu}}(\mathbf{i}_{N+1}; \mathbf{p}^*) - \boldsymbol{\mu}_k \right).$$

4.3. Improving approximate inference

Clearly, the inference procedure described in the previous two subsections is an approximate method. In particular, it does not take into account the variation in the texture of faces when it tries to find a maximum likelihood estimate for the shape parameters $\mathbf{p}$, but instead, it minimizes the squared error between the image $\mathbf{i}_{N+1}$ warped onto the base shape $\boldsymbol{\nu}$ and the texture mean $\boldsymbol{\mu}_k$. One may improve the situation by performing an alternating optimization[3] of $\mathbf{p}$ and $\boldsymbol{\lambda}_k$. The main disadvantage of such an alternating optimization scheme is that it leads to significant additional computational load, as each update of the texture parameters $\boldsymbol{\lambda}_k$ is relatively expensive.

Instead, we opt to use the faster (but less accurate) 'project-out' method [17]. This method uses the fact that the likelihood maximization w.r.t. to the shape parameters can be decomposed into two parts: one that operates in the span of $\mathbf{A}_k$ and another that operates in the space orthogonal to the span of $\mathbf{A}_k$. The minimal squared error in the first part is always equal to 0, as a result of which the optimal value of $\mathbf{p}$ can be found by only taking into account the second part. The texture variation in the span of $\mathbf{A}_k$ can thus be projected out[4]. The project-out method may be hampered by large texture variations in the test image [10, 20], but our use of the mixture of PCA-based texture model has already addressed this problem.

5. Experiments

In order to evaluate the performance of our extended AAM, we performed experiments on three large multi-person data sets of annotated face images. The setup of these experiments is described in subsection 5.1. In subsection 5.2, we present the results of the experiments on the three data sets.

5.1. Experimental setup

We performed experiments with our extended AAMs on three annotated face data sets: (1) an annotated subset of the AR face data set [16], (2) the IMM data set [19], and (3) the Cohn-Kanade data set [14]. The annotated subset of the AR data set that we used contains 504 color images of 126 individuals with various facial expressions of size 576×768 pixels [16]. The images are annotated with 22 facial feature points. The IMM data set contains grayscale and color images (all of which were converted to grayscale in our experiments) of 40 individuals, annotated by 58 feature points [19]. In total, the data set comprises 240 images of size 640×480 pixels with out-of-plane face rotations of up to 30 degrees. The Cohn-Kanade data set contains 496 gray-scale movies, showing 128 individuals producing various facial expressions [14]. The total number of frames in the Cohn-Kanade data set is $8,795$ frames, all of which are annotated with 59 facial feature points.

On all three data sets, we assess the performance of the AAMs by computing mean point-to-point errors and mean

[3] If the algorithm alternates after every parameter update, the *simultaneous* algorithm of [10] is obtained.

[4] The project-out method requires the bases $\boldsymbol{A}_k$ to be orthonormal. We achieve this by running Gram-Schmidt orthonormalization on the bases $\mathbf{A}_k$ after the training of the mixture of probabilistic PCA.

squared appearance errors from model fits on held-out test images. The point-to-point error is the mean Euclidean distance (in pixels) between the point annotations that resulted from the inference and the ground truth point annotations. The squared appearance error is the squared error between the shape-normalized image and the inferred texture (note that this measure is proportional to the likelihood that AAMs aim to maximize).

In all experiments on the three face data sets, we used 10-fold cross-validation to measure the performances, i.e., we randomly selected 90% of face images to train the AAMs, and we used the remaining 10% of the images to determine the generalization performance of the trained models (and we repeated this process 10 times). As we randomly split the face data, the test set may contain images of individuals that were not in the training set. In all experiments, we used AAMs with the settings described below.

As shape model, we use a standard PCA model with 10 dimensions [6]. We manually add 4 components to the linear basis that can capture differences in scale, translation, and rotation of the face (see [17] for details). In the shape model, 4 shape parameters thus represent the size, location, and orientation of the face, whereas the remaining 10 shape parameters represent the shape variation. Note that the shape basis is orthogonal by construction, because the shapes were normalized using Procrustes alignment before the PCA model was trained.

As texture model, we use the mixture of probabilistic PCA model with 30-dimensional latent spaces $\boldsymbol{\lambda}_k$ and K mixture components. The mixture of probabilistic PCA is trained by running the EM-algorithm for 100 iterations. For computational reasons, we first perform PCA to reduce the dimensionality of the face textures to 300 before training the mixture of probabilistic PCA model. In order to account for camera gain and offset, we manually add the mean texture image and an all-white image to the texture bases, and orthonormalize. We use color information in the experiments on the AR data set (the other two data sets only provide gray-scale images).

In all experiments, we perform inference by running the likelihood maximization for 30 iterations. In our inference procedure, the initialization of the shape parameters $\mathbf{p}$ is performed by running the facial feature point detector developed in [9]. The detector combines local appearance information with a top-down model in order to find the corners of the eyes, nose, and mouth in the detected face. Given the location of these facial feature points, the initial values of the translation, scale, and rotation parameters can be computed in closed form. In our implementation, the warping function W is implemented by a piecewise linear warp. We provide Matlab code to reproduce the results of our experiments on http://www.anonymized.com.

5.2. Results

Below, we present the results of our experiments on the three face data sets in three separate subsections.

5.2.1 AR data set

Table 1 gives an overview of mean point-to-point errors and mean *per-pixel* squared appearance errors of fits measured on the AR data set[5]. Both errors were computed using 10-fold cross validation. The best performance in terms of point-to-point error and in terms of squared appearance error is typeset in boldface. In addition, the table presents the mean time required to perform full inference on a single face image (measured in a simple Matlab-implementation). The reported computation times include the time required for face detection and parameter initialization. Please note that the setting $K = 1$ corresponds to fitting a standard PCA-based AAM using the project-out inverse compositional algorithm, as described in [17].

K	*Pt.-to-pt. err.*	*Appear. err.*	*Time per im.*
1	6.30	2.13×10^{-2}	1.89 sec.
2	6.25	1.60×10^{-2}	2.90 sec.
3	5.95	1.34×10^{-2}	3.95 sec.
4	**5.46**	1.23×10^{-2}	4.83 sec.
5	5.75	$\mathbf{1.20 \times 10^{-2}}$	6.04 sec.

Table 1. Point-to-point and per-pixel squared appearance errors on the AR data set using various settings of K.

The results presented in Table 1 show that using a mixture of PCA to model texture variation leads to better results when fitting AAMs on new faces, both in terms of the shape fit as in terms of the appearance fit. In particular, using a mixture appearance model with 4 mixture components leads to an improvement of more than 10% in terms of the mean point-to-point error, and an improvement of more than 40% in terms of the mean squared appearance error.

In Figure 2, we present a plot that shows the cumulative appearance error distribution for the model with $K = 1$ and the model with $K = 5$. To construct this plot, we averaged the test results on the AR data set over all 10 folds. The plot shows that using a mixture model as appearance model has a significant positive effect on the expected squared appearance error of a fit.

Some examples of shape and appearance fits on the AR data set (obtained using the model with $K = 2$) are presented in Figure 3. The visualizations reveal that our extended AAMs pick up on quite a lot of facial features, such as the presence of (minor) facial hair, color of eyes and lips, position of the mouth, etc. Admittedly, the generated faces

[5] In the experiments on the AR data set, we did not encounter any images for which the inference procedure diverged.

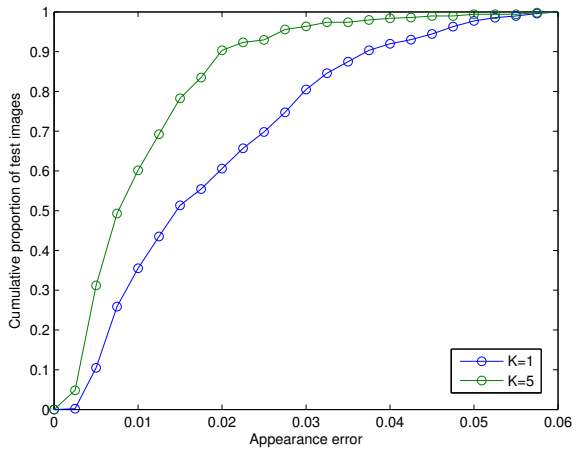


Figure 2. Cumulative distribution of appearance errors on the AR data set (averaged over 10 folds).

are slightly blurred compared to their original counterparts. This blurring is the result of (1) small errors in the feature point locations, leading to, e.g., blurry glasses, (2) the fact that the model was trained on images of more than 100 different individuals with a variety of facial expressions, and (3) the fact that the texture is governed by only 30 parameters.

5.2.2 IMM data set

In Table 2, we present the results of our experiments on the IMM data set. As for the AR data set, we report mean point-to-point and per-pixel squared appearance errors measured using 10-fold cross-validation. We did not encounter images on which the inference procedure diverged in our experiments on the IMM data set.

The results show an improvement of approximately 20% in terms of appearance error for the AAM with $K = 5$. In terms of point-to-point errors, the model with $K = 2$ performs best, although the differences between the various models are very small. Most likely, this is due to one of the following two reasons: (1) the number of different individuals that are depicted in the IMM data set is relatively small, giving the mixture model a limited advantage, and (2) the results may be influenced by a ceiling effect that is due to small errors in the manual annotations.

K	*Pt.-to-pt. err.*	*Appear. err.*	*Time per im.*
1	6.43	9.22×10^{-3}	1.21 sec.
2	**6.37**	8.57×10^{-3}	1.55 sec.
3	6.66	8.17×10^{-3}	1.87 sec.
4	6.58	7.90×10^{-3}	2.22 sec.
5	6.55	$\mathbf{7.72 \times 10^{-3}}$	2.55 sec.

Table 2. Point-to-point and per-pixel squared appearance errors on the IMM data set using various settings of K.

5.2.3 Cohn-Kanade data set

In Table 3, we present the results of our experiments on the Cohn-Kanade data set. The results are presented in the same way as the results on the AR and IMM data sets, but as the inference procedure sometimes did not converge on images in the Cohn-Kanade data set, we also present the relative number of times that the inference diverged.

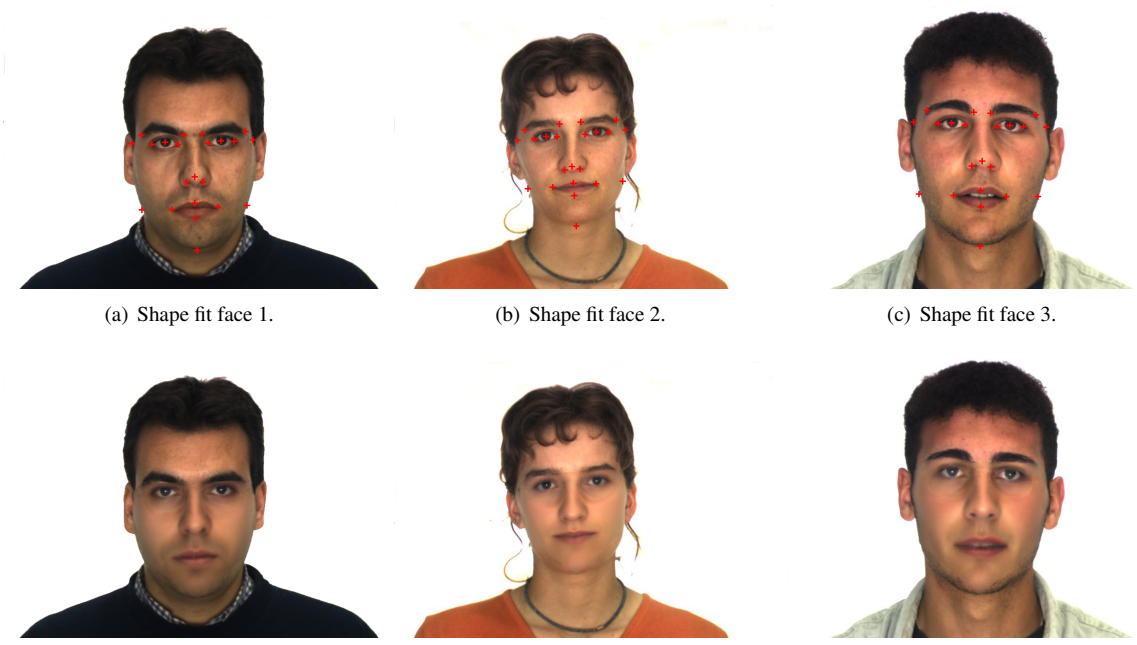
K	*Diverg.*	*P2P err.*	*Appear. err.*	*Time per im.*
1	0.29%	5.02	0.33×10^{-1}	0.56 sec.
2	0.23%	**4.69**	0.33×10^{-1}	1.09 sec.
3	0.20%	5.51	0.33×10^{-1}	1.63 sec.
4	0.18%	4.99	0.33×10^{-1}	2.18 sec.
5	**0.17%**	5.20	0.33×10^{-1}	2.70 sec.

Table 3. Point-to-point and (squared) appearance errors on the Cohn-Kanade data set using various settings of K.

The results reveal that using the extended AAM reduces the number of times that the inference diverges: the model with $K = 1$ diverges approximately 70% more often than the model with $K = 5$. In terms of the mean appearance error, using a model with $K > 1$ does not appear to lead to a better performance, but the reader should note that this result is biased: the model with $K = 5$ converges on more of the images, so the reported appearance error for the $K = 5$ model is an average over more images than the reported appearance error for the $K = 1$ model. As the images on which the $K = 1$ model diverges, but the $K = 5$ model converges, are typically 'difficult' test images, the average appearance error for the $K = 5$ model is negatively influenced by the inclusion of these images. The same holds for the mean point-to-point error. Despite the bias in the error estimates, the point-to-point error is the lowest for the model with $K = 2$.

6. Discussion

From the results presented in the previous section, we observe that our extended AAMs often produce better results, for instance, in terms of the mean squared error of the appearance fit (which is the criterion that AAMs aim to minimize). Although these improved results come at additional computational costs, the computational costs only grow linearly in the number of mixture components K. Moreover, parallelizing the inference procedure described in section 4 is straightforward: the optimization for each mixture component can be performed on a separate processor. Moreover, one could imagine an hierarchical approach that first performs rapid fitting using a simple PCA-based texture model, and subsequently, refines the fit using the mixture components in the extended AAM. Such an approach may exploit the learned mixing proportions to determine the ordering of



(a) Shape fit face 1. (b) Shape fit face 2. (c) Shape fit face 3.

(d) Appearance fit face 1. (e) Appearance fit face 2. (f) Appearance fit face 3.

Figure 3. Examples of shape fits (top row) and appearance fits (bottom row) on the AR data set by an extended AAM with $K = 2$, trained on images of 126 different individuals (using 10-dimensional shape bases and 30-dimensional texture bases).

the mixture components, as these proportions indicate the probability that a mixture component leads to the best fit.

We showed that, apart from sometimes leading to better feature point localization, our extended model achieves lower appearance errors on the AR and the IMM data sets. The appearance error reduction of our extended models is particularly interesting for face synthesis tasks such as expression cloning [23]. Currently, face synthesis is only performed with AAMs that are trained on image of a single individual, but our extended model may pave the way for the construction of a single AAM that can synthesize faces from lots of different individuals.

Like the standard formulation of AAMs, our extended AAMs implement the idea of selecting texture subspaces based on an objective that tries to maximizes the data variance in these subspaces. However, we note that for tasks such as facial expression recognition or face identification based on using the inferred AAM parameters as features, these subspaces are likely to be suboptimal. In many data sets, the main sources of variance are sources such as illumination changes, under which we would like such features to be invariant. In contrast, minor components may capture information that is essential to expression or identity, such as the presence of small wrinkles or speckles. Hence, we surmise the value of AAMs in facial expression or identity recognition may be increased by adding minor components to the texture bases, or by learning which components to use in the texture bases [5].

A remaining question is whether it would be beneficial to implement the shape model using a mixture model as well. Presumably, the shape variation distribution is not a multimodal distribution, but it forms a smooth non-linear manifold. The mixture of PCA model has previously been successfully applied to such non-linear manifolds [4, 21]. It thus seems likely that using a mixture model to describe shape variation may be beneficial, in particular, when out-of-plane rotations, occlusions, or extreme facial expressions are present in the face data [11]. However, using mixture models to describe both shape and texture variations does lead to a significant additiona computational burden.

7. Conclusions

In this paper, we have presented an extension of the standard AAM that uses a mixture model to capture the large variations in facial appearances in large multi-person data sets. The results of our experiments on three multi-person face data sets revealed that this extension leads to performance improvements, in particular, in terms of the mean squared error of the appearance fits, and that the computational burden of our extension is relatively small. We mention four potential directions for future work.

As a first direction for future work, we aim to include priors over the shape parameters $\mathbf{p}$ and texture parameters $\boldsymbol{\lambda}$ in the inference procedure, i.e., to perform maximum a posteriori estimation to prevent the optimization proce-

dure from diverging [20]. Second, we intend to extend our model to simultaneously fit a 2D and a 3D AAM to obtain better performance under the presence of out-of-plane rotations [12, 26]. Third, we aim to investigate whether the use of minor components in the texture bases [5] can improve the value of AAMs in, e.g., identity recognition. Fourth, we intend to investigate discriminative fitting schemes to fit the extended AAM. Hitherto, we did not address discriminative fitting schemes, because in this study, we are mainly interested in comparing *models*, not in comparing fitting schemes.

8. Acknowledgements

LvdM is supported by the EU-FP7 NoE on Social Signal Processing (SSPNet) and by the Netherlands Organisation for Scientic Research (grant 680.50.0908). The authors thank David Tax and Josh Susskind for helpful discussions.

References

[1] N. Baka, J. Milles, E. Hendriks, A. Suinesiaputra, M. Herold, J. Reiber, and B. Lelieveldt. Segmentation of myocardial perfusion MR sequences with multi-band active appearance models driven by spatial and temporal features. In *SPIE Medical Imaging*, 2008.

[2] S. Baker and I. Matthews. Equivalence and efficiency of image alignment problems. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, 2001.

[3] A. Batur and M. Hayes. Adaptive active appearance models. *IEEE Transactions of Image Processing*, 14(11):1707–1721, 2005.

[4] M. Brand. Charting a manifold. In *Advances in Neural Information Processing Systems*, volume 15, pages 985–992, Cambridge, MA, 2002. The MIT Press.

[5] Y. Chen and M. Welling. Bayesian Extreme Components Analysis. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1022–1027, 2009.

[6] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 484–498, 1998.

[7] R. Donner, M. Reiter, G. Langs, P. Peloshek, and H. Bishof. Fast active appearance model search using canonical correlation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1690–1694, 2006.

[8] G. Edwards, C. Taylor, and T. Cootes. Interpreting face image using active appearance models. In *Proceedings of IEEE Conference on Automatic Face and Gesture Recognition*, pages 300–305, 1998.

[9] M. Everingham, J. Sivic, and A. Zisserman. 'Hello! My name is... Buffy' - Automatic naming of characters in TV video. In *Proceedings of the* 17^{th} *British Machine Vision Conference*, pages 889–908, 2006.

[10] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23:1080–1093, 2005.

[11] L. Gu and T. Kanade. A generative shape regularization model for robust face alignment. In *Proceedings of the European Conference on Computer Vision*, pages 413–426, 2008.

[12] O. Hamsici and A. Martinez. Active appearance models with rotation invariant kernels. In *Proceedings of the International Conference on Computer Vision*, pages 1003–1009, 2009.

[13] X. Hou, S. Li, H. Zhang, and Q.Cheng. Direct appearance models. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 828–833, 2001.

[14] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 46–53, 2000.

[15] X. Liu. Generic face alignment using boosted appearance model. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[16] A. Martinez and R. Benavente. The AR Face database. Technical Report 24, CVC, 1998.

[17] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[18] P. Mittrapiyanuruk, G. DeSouza, and A. Kak. Accurate 3D tracking of rigid objects with occlusion using active appearance models. In *WACV/MOTION*, pages 90–95, 2005.

[19] M. Nordstrøm, M. Larsen, J. Sierakowski, and M. Stegmann. The IMM face database - An annotated dataset of 240 face images. Technical report, Techn. University of Denmark, 2004.

[20] G. Papandreou and P. Maragos. Adaptive and constrained algorithms for inverse compositional active appearance model fitting. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[21] S. Roweis, L. Saul, and G. Hinton. Global coordination of local linear models. In *Advances in Neural Information Processing Systems*, volume 14, pages 889–896, 2001.

[22] J. Saragih and R. Goecke. A nonlinear discriminative approach to AAM fitting. In *Proceedings of the International Conference on Computer Vision*, pages 1–8, 2007.

[23] B.-J. Theobald, I. Matthews, J. Cohn, and S. Baker. Real-time expression cloning using appearance models. In *Proceedings of the* 9^{th} *International Conference on Multimodal Interfaces*, pages 134–139, 2007.

[24] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.

[25] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3):611–622, 1999.

[26] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2D+3D active appearance models. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 535–542, 2004.

[27] J. Zhang, S. Zhou, D. Comaniciu, and L. McMillan. Discriminative learning for deformable shape segmentation: A comparative study. In *Proceedings of the European Conference on Computer Vision*, pages 711–724, 2008.