Active Appearance Models with Rotation Invariant Kernels

Onur C. Hamsici and Aleix M. Martinez
Department of Electrical and Computer Engineering
Ohio State University, Columbus, OH 43210
{hamsicio,aleix}@ece.osu.edu

Abstract

2D Active Appearance Models (AAM) and 3D Morphable Models (3DMM) are widely used techniques. AAM provide a fast fitting process, but may represent unwanted 3D transformations unless strictly constrained not to do so. The reverse is true for 3DMM. The two approaches also require of a pre-alignment of their 2D or 3D shapes before the modeling can be carried out which may lead to errors. Furthermore, current models are insufficient to represent nonlinear shape and texture variations. In this paper, we derive a new approach that can model nonlinear changes in examples without the need of a pre-alignment step. In addition, we show how the proposed approach carries the above mentioned advantages of AAM and 3DMM. To achieve this goal, we take advantage of the inherent properties of complex spherical distributions, which provide invariance to translation, scale and rotation. To reduce the complexity of parameter estimation we take advantage of a recent result that shows how to estimate spherical distributions using their Euclidean counterpart, e.g., the Gaussians. This leads to the definition of Rotation Invariant Kernels (RIK) for modeling nonlinear shape changes. We show the superiority of our algorithm to AAM in several face datasets. We also show how the derived algorithm can be used to model complex 3D facial expression changes observed in American Sign Language (ASL).

1. Introduction

Active Appearance Models (AAM) [4, 3] and 3D Morphable Models (3DMM) [2] are commonly used to fit a shape and an appearance (texture) model to an image of a learned object. Both of these algorithms learn the statistics of the shape and the appearance of the object from examples. A primary advantage of these methods is that it is not necessary to create complex representations for each object, since their models are implicitly defined by the selection of the sample images. However, to achieve this, AAM and 3DMM require that the shape of *all* the sample images be *aligned* to each other. This is usually accomplished with a least-squares fit, which can be sensitive to outliers [5]. After alignment, one can use Principal Components Analysis (PCA) to describe each shape and appearance as a deviation from a mean shape/texture [7]. Thus, in general, object variations are modeled using a Gaussian (Normal) distribution, $N(\mu, \mathbf{\Sigma})$, where μ is the mean shape/appearance and $\mathbf{\Sigma}$ defines the possible (allowable) variations from the mean.

AAM are used to model 2D shape and appearance changes, while 3DMM model 3D variations. Modeling 3D variations is preferred in many applications, because it can be made more robust to illumination and pose changes. Moreover, 3DMM can easily incorporate the modeling of occlusions, whereas AAM need to be specifically trained to do that. However, one very important advantage of AAM is their fitting speed (from model to image), which can run at more than 100 frames per second. To date, several discriminative and generative approaches of AAM have been proposed. In [18] a discriminative approach with a boosted ensemble of classifiers based on Haar-like features is defined. In [21] three methods, a classification-regression-ranking based, are compared with the latter yielding superior results. Another successful application of ranking models is shown for face alignment in [19]. And [8] proposes a multi-level generative model that is robust to expressions, occlusions and image noises.

In this paper, we derive a 3D AAM based on a nonlinear model which combines the advantages of 3DMM and AAM and does not require of the common step of shape alignment. The 3D nature of the proposed approach provides the advantages of the 3DMM mentioned above. The nonlinear model allows us to represent the nonlinear changes in the shape and texture over the face images. The AAM roots of the formulation presented below permit a fast fitting process.

The 3D extension of AAM and the unnecessary alignment step are possible thanks to the realization that 3D shapes can be made translation, rotation, and scale invariant if they are modelled using distributions with the antipodally symmetric property, i.e., x and $-x$ have the same likelihood

[7]. One particular distribution that has been extensively used for this purpose in shape modelling is the complex Bingham. Unfortunately, the parameters of the Bingham distribution have been typically estimated using complex optimization procedures, which cannot guarantee convergence and make the training process very slow whenever convergence is achieved. A recent result [10] shows how, under mild conditions, the Bingham distribution can be substituted by a zero-mean Gaussian. Following this approach, the learning of the shape and appearance of our objects becomes a simple task with a closed-form solution of polynomial complexity lower than order 2. The resulting approach is called 3D AAM, because 3DMM generally use dense feature point representations, whereas the proposed model employs a lower number of these such as in 2D AAM.

2. Background Formulation

The goal in 2D AAM is to fit a shape model to an image using the appearance information from the image. The statistical shape model is learned from the shape vectors $\mathbf{s} = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$ defining the x and y image coordinates of n landmarks. The shape of the objects are obtained after aligning the translation, scale and rotation changes [7]. The aligned shape vectors can then be summarized by PCA, yielding a mean $\mathbf{s}_0$ and d basis (shape) vectors $\mathbf{s}_i$. Any shape can then be represented as a linear combination of this mean and basis shapes [3, 7], $\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{d} p_i \mathbf{s}_i$, where $\mathbf{s}_i$ is the i^{th} eigenvector and p_i is the corresponding coefficient.

Statistical texture model is defined over shape-free representations which are obtained after affine warping the images to the mean shape. After normalizing their means and standard deviations, the texture can be represented as a linear combination of the mean appearance and the basis appearances obtained with PCA, $\mathbf{g} = \mathbf{g}_0 + \sum_{i=1}^{d} \lambda_i \mathbf{g}_i$, where λ_i are the associated coefficient to each of the eigenvectors $\mathbf{g}_i$.

3DMM are similarly defined [2]. In this case, 3D shape vectors are used with each $\mathbf{s}^3 = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_n, y_n, z_n)$ defining the vertices of a 3D mesh in Euclidean space. Then, a dense point-to-point correspondence is used, which is typically obtained with optical flow. Such a dense correspondence is employed to align all the shape feature vectors. The statistical shape changes are given by $\mathbf{s}^3 = \mathbf{s}^3_0 + \sum_{i=1}^{d} p^3_i \mathbf{s}^3_i$, with $\mathbf{s}^3_0$ the average shape, $\mathbf{s}^3_i$ the i^{th} eigenvector and p^3_i the corresponding coefficient. Appearance changes are again modeled as a linear combination of basis vectors extracted by PCA. In 3DMM, however, these are represented by the 2D (triangular) mesh defining the object.

To extend the uses of AAM and combine the advantages of these and 3DMM, [20] shows that the degrees of freedom of a AAM model is larger than those of the 3DMM. This suggests the use of 3D models to constraint the possible values of 2D AAM under the assumption of an affine projection between models, $\mathbf{s} = \mathbf{P}\mathbf{s}^3 + \mathbf{t}$, where $\mathbf{s}$ is the 2D shape generated by the projection of the 3D shape feature vector $\mathbf{s}^3$ using the affine projection matrix $\mathbf{P}$ and the translation vector $\mathbf{t}$. A main problem with this approach is that it too requires of a pre-alignment of the shapes (and texture if we want to be invariant to the intensity of the illumination source). This can lead to incorrect alignments caused by outliers in the least-squares fit [5].

3. 3D Active Appearance Models with Rotation Invariant Kernels

A classical statistical alternative to representing shapes is to model the feature vectors with spherical probability density functions (pdf) [7].

3.1. Spherical Representation

In the approach introduced above the 3D feature vectors are first normalized with respect to their mean, $\bar{\mathbf{s}}^3_{(j)} = \mathbf{s}^3_{(j)} - n^{-1} \sum_{i=1}^{n} \mathbf{s}^3_{(i)}$, where $\mathbf{s}^3_{(i)} = (x_i, y_i, z_i)$ are the 3D point coordinates of the i^{th} object landmark in $\mathbf{s}^3$. This makes our representation translation invariant. Then, we normalize the resulting vectors to have unit norm, $\hat{\mathbf{s}}^3 = \bar{\mathbf{s}}^3 / \|\bar{\mathbf{s}}^3\|$, where $\|.\|$ is the 2-norm length of the vector. This makes the representation invariant to scale changes.

The normalization steps described above map the original data onto the surface of a hypersphere of dimensionality $d-2$. One dimension is lost during the mean normalization step, which maps the data from a hyperplane of d dimensions to one of $d-1$. The second dimension is lost by the norm-normalization which is a mapping from the hyperplane $\mathbb{R}^{d-1}$ to the spherical representation S^{d-2}.

Now, to achieve rotation invariance, we need to describe the resulting feature vectors using a pdf that carries a rotation invariance property. If the shapes were two dimensional, this could be realized by representing the data using a distribution which has the following property $f(\mathbf{z}) = f(\mathbf{z}e^{i\theta}),\ \forall\theta \in [0, 2\pi]$ where $\mathbf{z} = (x_1 - iy_1, \ldots, x_p - iy_p)^*$ is the shape vector in the complex domain and $p = n$. In some cases, the shape distribution may in fact lie in a d-dimensional subspace, $d \leq n$. In any event, the rotation invariance is provided since multiplying a shape vector with an arbitrary rotation $e^{i\theta}$ does not change the shape representation in $f(.)$. This property is obtained when we have a complex antipodally symmetric pdf such as the complex Bingham [7] given by $f(\mathbf{z}|\mathbf{A}) = c_{CB}(p, \mathbf{A}) \exp\{\mathbf{z}^* \mathbf{A} \mathbf{z}\}$, where $c_{CB}(p, \mathbf{A})$ is the normalizing constant and $\mathbf{A}$ is a $p \times p$ symmetric matrix defining the parameters of the distribution.

Unfortunately, this approach requires that we estimate the normalizing constant of a complex Bingham distribution, which does not have a known solution. A recent result [10] shows how we can substitute the complex Bingham distribution by the zero-mean complex Gaussian, $N(\mathbf{\Sigma})$, with pdf $f(\mathbf{z}) = C_{CN}^{-1}(\mathbf{\Sigma})\exp(-\mathbf{z}^*\mathbf{\Sigma}^{-1}\mathbf{z}), \quad \mathbf{z} \in \mathbb{C}S^{p-2}$, where $\mathbf{\Sigma}$ is a $(p-1)\times(p-1)$ positive-definite complex Hermitian matrix and $C_{\mathbb{C}N}(\mathbf{\Sigma}) = \pi^{p-1}\det(\mathbf{\Sigma})$ is the normalizing constant.

We note however that this approach still requires that we describe the shape vector $\mathbf{s}$ in the complex domain, $\mathbb{C}S^{d-2}$. To avoid this, [9] demonstrates that by using a rotation invariant kernel[1] (RIK) one can bypass the requirement of using the complex domain, because rotation invariance is already given by the kernel. Hence, the property $f(\mathbf{z}) = f(\mathbf{z}e^{i\theta})$ can be dropped. For two 2-dimensional shapes $\mathbf{z}_j$ and $\mathbf{z}_k$, the rotation invariant kernel is defined as,

$$k(\mathbf{z}_j, \mathbf{z}_k) = \exp\left(-\frac{\|\mathbf{z}_j - \mathbf{z}_k \exp^{-i\theta_{\mathbf{z}_j\mathbf{z}_k}}\|^2}{2\sigma^2}\right). \qquad (1)$$

This can be rewritten as $\exp\left(-\frac{2-2\|\mathbf{z}_j^*\mathbf{z}_k\|}{2\sigma^2}\right)$, since $\|\mathbf{z}_j^*\mathbf{z}_j\| = 1$. Hence, *RIK* give us an easy way to calculate the similarity between the shapes.

The above kernel is defined for 2D shapes [9]. In this paper, we derive a new RIK which can be applied to 3D shapes. This RIK depends on the linearly independent 2D projections of the 3D shape. These projections are obtained in the subspace defined by some selected feature points. We call these points *pivot* points, since the rotation is defined with respect to these [11]. This is shown in Fig. 1. In (a) we see the mean and norm-normalized 3D shape $\hat{\mathbf{s}}^3$, the pivot points marked with red circles, and the basis vectors defined by the pivot points $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$. Fig. 1(b) shows one of the 2D projections obtained in this coordinate system. We now represent this 2D projection in the complex domain, i.e. $\mathbf{z}^1 = \hat{\mathbf{s}}^3\mathbf{v}_1 + i\hat{\mathbf{s}}^3\mathbf{v}_2$. Since this projection eliminates the depth information, we need at least one more 2D projection. This can be selected from any of the other two basis vectors or from some other basis vectors obtained from other pivot points. The important point in this process is the selection of the pivot points. These points should correspond to the same landmarks across different shapes (e.g. in our application we selected the pivot points from the nose landmarks of the face shapes) and they should be selected from rigid points.

By using the complex vector domain in the 2D subspaces defined by the pivot points, we obtain the complex shape representations $\mathbf{z}^1$ as described above and $\mathbf{z}^2$ in the subspace spanned by $\mathbf{v}_2$ and $\mathbf{v}_3$. Then, the RIK $k(\hat{\mathbf{s}}_j^3, \hat{\mathbf{s}}_k^3)$ defined between shapes $\hat{\mathbf{s}}_j^3$ and $\hat{\mathbf{s}}_k^3$ can be calculated as

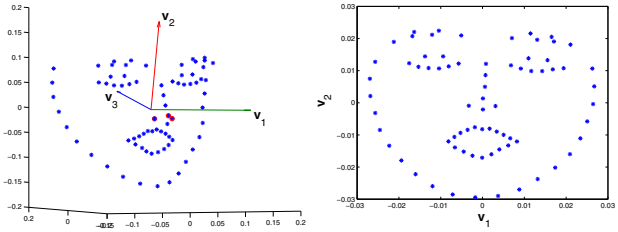
$$\exp\left(-\frac{\|\mathbf{z}_j^1 - \mathbf{z}_k^1\exp^{-i\theta_{\mathbf{z}_j^1\mathbf{z}_k^1}}\|^2 + \|\mathbf{z}_j^2 - \mathbf{z}_k^2\exp^{-i\theta_{\mathbf{z}_j^2\mathbf{z}_k^2}}\|^2}{2\sigma^2}\right), \qquad (2)$$

where $\exp^{-i\theta_{\mathbf{z}_j^l\mathbf{z}_k^l}}$ is the complex scalar rotation between the l^{th} subspace projections $\mathbf{z}_j^l$ and $\mathbf{z}_k^l$.

Furthermore, since $\|\mathbf{z}_j^l - \mathbf{z}_k^l\exp^{-i\theta_{\mathbf{z}_j^l\mathbf{z}_k^l}}\|^2 = \mathbf{z}_j^{l\,*}\mathbf{z}_j^l + \mathbf{z}_k^{l\,*}\mathbf{z}_k^l - 2\|\mathbf{z}_j^{l\,*}\mathbf{z}_k^l\|$, the 3D RIK can easily be calculated using only inner products between the 2D shapes.

Note that shapes projected into the kernel space become rotation invariant. We also note that this kernel carries an inherent mapping resulting in a kernel space that is still spherical, since $k(\mathbf{x}, \mathbf{x}) = 1$. Hence, while the two 2-dimensional shapes that are generated by the projection of three dimensional shapes are no longer spherical in the original space, they are spherical in the RIK space and we will require to use a zero-mean Gaussian fit.

The main difference between our approach and that of [17] is that in the latter the authors extend 2D active shape models (ASM) [3] to multi-view scenes using kernel PCA, rather than using the properties of spherical distributions and rotation invariant kernels. Similar problems to those observed in 2D ASM/AAM remain when using kernel PCA, because this approach also employs the least-squares alignment procedure. Furthermore, since their shape model is 2D, it has an over-representation of the shape, which can generate impossible 2D projections of the underlying 3D model [20].



(a) (b)

Figure 1. In (a) we show a 3D shape with 3 pivot points selected as the tip of the nose and sides of the nose. A coordinate system defined with respect to these pivot points is given by the basis vectors $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$. In (b) we show the projection of the 3D shape onto the subspace defined by $\mathbf{v}_1$ and $\mathbf{v}_2$.

3.2. Fitting Procedure

Since our major concern is the normalization and the model selection of the AAM, we fit the learned AAM to a new test image using a simple gradient descent approach.

As described in the preceding section, after the mean-norm-normalization step, we fit the zero-mean Gaussian distribution to the shape changes of the model in the RIK

[1] A kernel is rotation invariant if the mapping of a shape feature vector and a rotated version of it are identical in the kernel space.

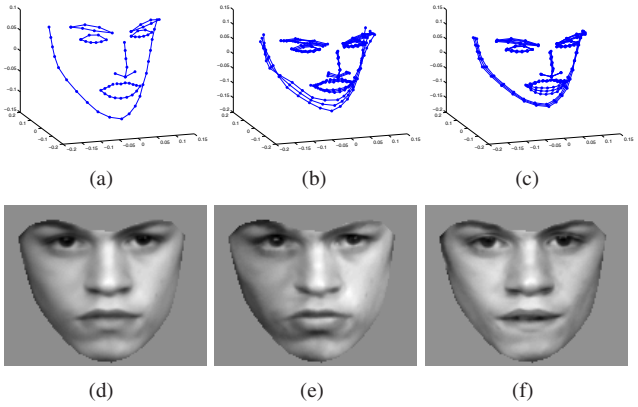


Figure 2. In (a-c) we show the pre-images of the first three eigenvectors of the zero-mean Gaussian distribution fitted to the normalized shape vectors in the kernel space given by (2). (a) The mean direction vector (mean shape). (b-c) The allowable deviations (represented here as deformations from the mean). (d) The mean appearance. ± 2 standard deviations describing the texture change about the second (e-f) eigenvectors.

space of (1) and (2) for $2D$ and $3D$ shapes, respectively. Note that since these distributions are defined in the kernel space, reconstruction of a shape (pre-image) during the fitting procedure can be problematic. To resolve this, we can approximate the pre-image by minimizing

$$\arg\min_{\mathbf{z}} \|\phi(\mathbf{z}) - P_n\phi(\mathbf{x})\|^2, \tag{3}$$

where $P_n\phi(\mathbf{x})$ is a vector in the functional space or the projection of a new sample in the functional space for denoising purposes. This minimization problem can be solved using the iterative algorithm of [15]. Because of the computational complexity and local minima problems, many alternatives to this approach have been proposed. We used the pre-image algorithm proposed in [1] because of its efficiency and accuracy.

As mentioned above, in the present paper, we use the kernel trick, which directly provides the mapping function $\phi(.)$, $k(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x})^T\phi(\mathbf{y})$, with $k(.)$ the kernel. When working with shape vectors we will use the kernel defined in (1) or (2). To represent the 2D shape-free appearance we do not require of a rotation-invariant kernel. In that case, we will employ the RBF kernel.

Solving for (3) provides the eigenvectors of the shape representation. The first 3 eigenvectors, ± 2 standard deviations as given by their corresponding eigenvalues, are shown in Fig. 2(a-c). The first eigenvector of the resulting covariance matrix is the (Procrustes) shape mean. Additional eigenvectors describe the nonrigid shape change/motion. After obtaining the mean shape vector, every image is warped to a shape free patch. The warping function is defined from the current shape to the mean shape using the affine warp over the triangles obtained with delaunay triangulation.

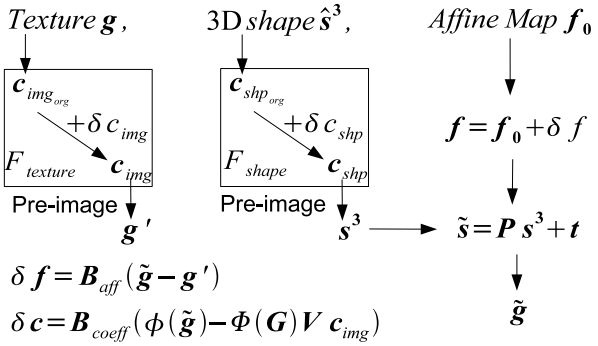


Figure 3. The 3D AAM learning procedure. The gradient direction of affine projection parameters f, $\mathbf{B}_{aff}$, is estimated in the Euclidean space. The gradient direction of the texture and shape changes $\mathbf{B}_{coeff}$ are computed in the kernel space. F_{shape} corresponds to the kernel space given by Eq. (2). $F_{texture}$ is the kernel space defined by the RBF kernel used to model shape-free patches.

Once the shape-free graylevel images are obtained, we require of a normalization step to eliminate the intensity changes across the images. As above, this can be achieved by means of a mean-norm-normalization. As in shape analysis this will map the images onto the surface of an hypersphere. Again, we need to use the zero-mean Gaussian distributions to appropriately describe the data. To obtain a good fit, we first use the RBF kernel to map the shape-free image patches to a high-dimensional space where the data best adapts to this Gaussian model [10].

Fig. 2(d-f) shows the eigenvectors obtained for the graylevel images using (3). As in shape modeling, the first eigenvector corresponds to the mean appearance while the others represent allowable (orthogonal) changes.

The process described thus far gives two independent representations, one to describe the shape and the other the appearance. These two representations utilize the zero-mean Gaussians in the kernel spaces $F_{texture}$ and F_{shape} as shown in Fig. 3. These two models can now be employed to build our 3D AAM as follows.

For each image and the corresponding 3D model, 25 randomly perturbed shape changes i.e., perturbations along the shape $\delta\mathbf{c}_{shp}$ and appearance $\delta\mathbf{c}_{img}$ coefficients of the eigenvectors, and affine projection parameters scale, rotation and translation $\delta\mathbf{f}$ as shown in Fig. 3 are generated. Each image in the training set is associated with a 3D shape $\hat{\mathbf{s}}^3$ and a shape-free patch $\mathbf{g}$. The 3D shape and appearance are mapped onto the subspaces given by the zero-mean Gaussian distributions in the kernel spaces $F_{texture}$ and F_{shape}. This allows to obtain their corresponding coefficients, which in turn provides the necessary perturbation, yielding the coefficients $\mathbf{c}_{shp}$ and $\mathbf{c}_{img}$ and the pre-images $\mathbf{s}^3$ and $\mathbf{g}'$ that we need.

The pre-image of the shape coefficients defines a 3D

shape. This shape is further perturbed using an affine projection matrix $\mathbf{P}$. For each 3D shape, the affine projection is given by $\tilde{\mathbf{s}} = \mathbf{P}\mathbf{s}^3 + \mathbf{t}$. In this model, the perturbed affine projection matrix $\mathbf{P}$ is generated from the first two rows of $(1+s)\mathbf{R}_1\mathbf{R}_2\mathbf{R}_3\mathbf{D}$. Here, s defines the scale, which is a random variable set to .1 standard deviation and thus provides a 10% change in scale. $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3$ are the three rotation angles (roll, yaw and pitch), which are sampled from a Normal distribution with 2.5 angular degrees of standard deviation. $\mathbf{D}$ is the original affine projection of the 3D shape, which we learn using a structure from motion algorithm [12]. And, $\mathbf{t}$ corresponds to the translation parameter, which is the original translation with an added random perturbation sampled from a Normal distribution with zero mean and $36\mathbf{I}$ covariance matrix, i.e., a translation noise of 6 pixels. As defined, the model allows variations along each of the shape and appearance eigenvector (herein referred to as $\mathbf{c}_{shp}$ and $\mathbf{c}_{img}$), the six degrees of freedom of the affine transformation, plus the two variables defining the translation (referred to as $\mathbf{f}$).

3.3. Fitting the model to an image

To fit the model to a new observation (image), one needs to learn the relation between the appearance change and the shape change. This can be done by means of a linear fitting. If all the parameters were defined in the Euclidean space as in 2D AAM, one could concatenate them and define a gradient descent procedure. However, in our case, the eigenvector coefficients are defined in the kernel space, whereas the affine projection parameters are defined in the Euclidean space. To work with this diversity of representations we need to define two gradient descent procedures These are defined as follows.

Assume that the perturbations from the original parameter vector related to the eigenvector coefficients $\mathbf{c}_{org}$ (referred to as $\mathbf{c}_{img_{org}}$ and $\mathbf{c}_{shp_{org}}$ in Fig. 3) are $\delta\mathbf{c}$. It then follows that $\mathbf{c}_{org} + \delta\mathbf{c}$ defines the coefficients generating a new pre-image for the shape and the appearance. We call the pre-image generated for the appearance $\mathbf{g}'$. On the other hand, projection of the 3D shape $\mathbf{s}^3$ by the affine projection matrix perturbed with $\delta\mathbf{f}$ defines a 2D perturbed shape $\mathbf{s}$ over the image. This will correspond to a different shape-free patch $\tilde{\mathbf{g}}$.

We can now fit a linear model between these two changes, $\delta\mathbf{f} = \mathbf{B}_{aff}\delta\mathbf{g}$, where $\mathbf{B}_{aff}$ is the least-squares linear fit, i.e., $\Delta\mathbf{F}\Delta\mathbf{G}^T\left(\Delta\mathbf{G}\Delta\mathbf{G}^T\right)^{-1}$, and $\Delta\mathbf{F}$ and $\Delta\mathbf{G}$ are the matrices whose columns correspond to a perturbation of the affine model parameters $\delta\mathbf{f}$ and the corresponding shape-free image patch change $\delta\mathbf{g}$.

Now, let the projection of $\tilde{\mathbf{g}}$ to the kernel space be $\phi(\tilde{\mathbf{g}})$ and the perturbed image defined in the nonlinear subspace of the zero-mean Gaussian (image) distribution be $\Phi(\mathbf{G})\mathbf{V}\mathbf{c}_{img}$, with $\Phi(\mathbf{G})$ the data matrix of the shape-free images mapped into the kernel space defined by $\Phi(.)$. Here, the i^{th} column of $\Phi(\mathbf{G})$ corresponds to the mapping of i^{th} shape-free training image $\phi(\mathbf{g}_i)$, and $\mathbf{V}$ is the $m \times n$ eigenvector coefficient matrix for the zero-mean Gaussian distribution defined in $\Phi(.)$ (m is the dimensionality of the subspace, n is the number of images). We also note that $\mathbf{c}_{img} = \mathbf{c}_{img_{org}} + \delta\mathbf{c}$ and $\mathbf{c}_{img_{org}} = \mathbf{V}^T\Phi(\mathbf{G})^T\phi(\mathbf{g}) = \sum_{j=1}^{m}\sum_{i=1}^{n} v_{ji}\phi(\mathbf{g}_i)^T\phi(\mathbf{g}) = \sum_{j=1}^{m}\sum_{i=1}^{n} v_{ji}k(\mathbf{g}_i, \mathbf{g})$ are the coefficients obtained in the kernel space defined by $k(.,.)$.

Following the notation given above, the linear fitting between coefficients (i.e., the nonlinear fitting with regard to the original space) is given by

$$\delta\mathbf{c} = \mathbf{B}_{coeff}\left(\phi(\tilde{\mathbf{g}}) - \Phi(\mathbf{G})\mathbf{V}\mathbf{c}_{img}\right),$$

where $\mathbf{B}_{coeff} = \Gamma^T\Phi(\mathbf{G})^T$, $\Gamma = \Delta\mathbf{C}\Delta\Phi(\mathbf{G})^{-1}$, with $\Delta\mathbf{C}$ and $\Delta\Phi(\mathbf{G})$ the matrices where each column is the change in the eigenvector coefficients $\delta\mathbf{c}$ and the corresponding shape-free image change in the kernel space $\Phi(\mathbf{G})^T\delta\phi(\mathbf{g}) = \Phi(\mathbf{G})^T\left(\phi(\tilde{\mathbf{g}}) - \Phi(\mathbf{G})\mathbf{V}\mathbf{c}_{img}\right)$.

We can now fit the AAM to a test image using the following gradient descent algorithm. We start with an initial parameter vector $\mathbf{c}_{ini}$ and an affine mapping $\mathbf{f}_{ini}$. From these initial estimates, one can extract the 3D shape, affine projection and the initial appearance $\mathbf{g}_{ini}$. We can also extract the shape-free patch that is given by the warping function from the current 2D shape (obtained from affine projection of the 3D shape) to the mean shape, i.e., $\tilde{\mathbf{g}}_{ini}$. Then, $\mathbf{B}_{aff}\delta\mathbf{g}$ gives the gradient direction that we need to follow. That is, $\mathbf{f}_{next} = \mathbf{f}_{ini} - k\delta\mathbf{f}$, where k is the learning rate.

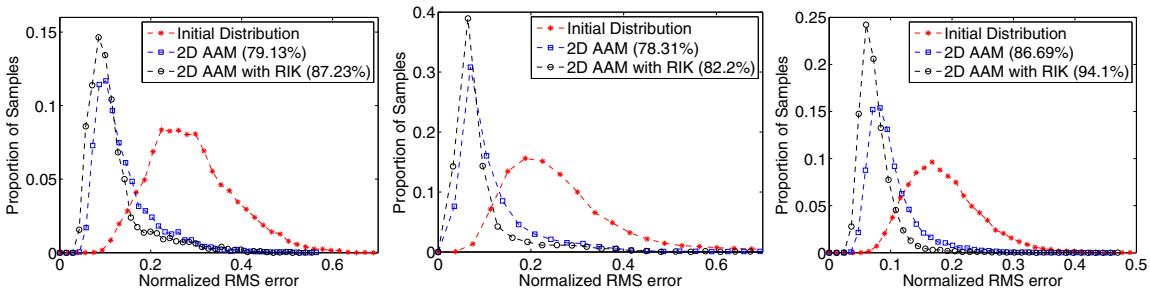
A similar iterative procedure can be used to fit the coefficient parameters $\mathbf{c}$ defining the kernel spaces. In this case, the perturbation in the current coefficients imply the definition of a new image $\tilde{\mathbf{g}}_{ini}$ and, hence, a new projection of the image to the kernel space. This change of the images in the kernel space, $\phi(\tilde{\mathbf{g}}_{ini}) - \Phi(\mathbf{G})\mathbf{V}\mathbf{c}_{img_{ini}}$, further implies a gradient direction, $\delta\mathbf{c}$. As above, we can use this result to write our iterative approach as $\mathbf{c}_{next} = \mathbf{c}_{ini} - k\delta\mathbf{c}$.

With the computed parameters, we can calculate a shape-free patch, a 3D shape and an affine mapping. This gives $\delta\mathbf{g}$. If $\delta\mathbf{g}$ is smaller than a specified threshold, we can stop searching for the best parameter. Otherwise, the new $\delta\mathbf{f}$ and $\delta\mathbf{c}_{img}$ are calculated and the iteration continues until the threshold is reached.

4. Experimental Results

We first compare the performance of 2D AAM to the proposed 2D AAM with RIK algorithm using three face datasets IMM [16], BIOID [13] and XM2VTS [14].

The IMM face dataset includes 240 images of 40 subjects, each with 6 different orientation and expressions. We keep 60 images of 10 randomly selected subjects for testing. A random shape from the training set is perturbed with



(a) (b) (c)

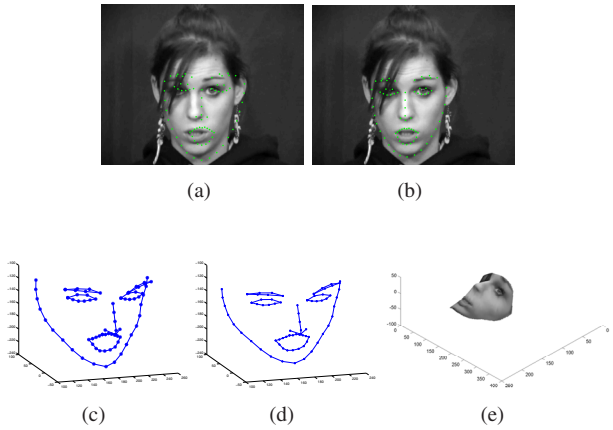
Figure 4. Distribution of the normalized RMS error is shown for IMM, BIOID and XM2VTS datasets respectively in (a),(b), and (c). Proposed 2D AAM with RIK achieve larger convergence rate (shown in parenthesis) with smaller RMS error in all datasets.

2.5 degrees of rotation changes, scale changes randomized with 5% standard deviation, and translation changes with standard deviation of 10 pixels. This random initialization is done 100 times for each test image. After we run the algorithms, the fitting procedure that has point-to-point RMS error smaller than the initial RMS is considered a *converged* solution. Fig. 4(a) shows the histogram of the proportion of the samples with respect to the converged point-to-point normalized RMS error obtained by 2D AAM and 2D AAM with RIK. Normalization is done by dividing each RMS error to the corresponding eye distance in the face image. The distribution of the normalized RMS error for the initial shapes is also plotted. While traditional AAM converges 79% of the time, our algorithm obtained 87% convergence with smaller RMS errors.

The BIOID face dataset has 1,521 face images for a total of 23 subjects. We randomly selected 152 ($\sim$ 10%) of these images for training. The rest are used for testing. 20 (provided) feature points are used to train the models. The trained AAMs are fitted to the remaining 1,369 images, with an random initialization. Fig. 4(b) shows the results. As seen in the figure, the proposed algorithm boosts both the convergence rate and the accuracy of the model.

The XM2VTS dataset consists of 2,360 images of 295 subjects, with 8 images per subject. We use 240 images for a total of 30 subjects ($\sim$ 10%) for training. Testing is done on the other subjects with random initializations. Fig. 4(c) shows our results. Again, AAM with RIK results in smaller RMS error and better convergence.

3D AAM have a large number of applications, especially in face recognition. Here, we have used video sequences of face close-ups of American Sign Language (ASL) sentences [6]. Facial articulations in ASL are extremely rich, performing semantic, prosodic, pragmatic, and syntactic functions in addition to social interaction and conversational regulation functions. In our experiment, we used two video sequences for a total of seven signers for training. Each sequence consists of $\sim$ 60 frames. In each frame, the internal and external facial components are outlined using 74 landmarks. 21 of these landmarks describe the jaw line, 6 are used to define each of the eyebrows, 8 for each eye, 7 for the nose, and 18 for the mouth.



(a) (b) (c) (d) (e)

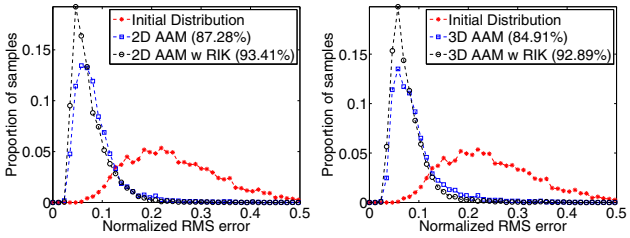
Figure 5. Shown here are the initial position and composition of the shape (a) and that obtained after five iterations (b). (c-d) These 3D models correspond to the initial composite shown in (a) and the final results obtained in (b). In (e) we plot the final 3D structure and shape recovered with the proposed algorithm.

We trained 2D AAM, 2D AAM with RIK, 3D AAM and 3D AAM with RIK using these sequences. Testing of the learned models is done over four new sequences of ASL signed by seven different subjects. The sequences used to test the AAM in this second test, correspond to the following sentences: "Grill start who?," "Grill Mary start?," "Fork in kitchen?," "Fork where?," which are different from those used for training. For each frame, we randomly initialized the model 20 times with a translation, scale and rotation perturbed 2D or 3D shape from the training set. This leads to 5,360 instances of testing.

In Fig. 5, we show a sample run of the 3D AAM with RIK fitting algorithm. The proposed approach converges very fast, usually after just five iterations. Fig. 6 shows additional fits.

The distribution of the normalized RMS error obtained

Figure 6. Shown here are the position of the affine 2D shape at the fifth iteration. The images include partial occlusions and large pose and expression changes.



(a) (b)

Figure 7. Shown here is normalized RMS error distribution and the convergence rate in parenthesis for ASL sequences obtained by 2D AAM and 2D AAM with RIK in (a), and 3D AAM and 3D AAM with RIK in (b).We also show the normalized RMS error distribution of the initial shapes.

for the converged fittings with 2D and 3D AAMs and 2D and 3D AAM with RIK are shown in Fig. 7(a-b). The distribution of the normalized RMS error of the initialization shapes is also plotted. As seen in these figures RIK based algorithms improves the convergence rate while reducing the RMS error. Examples of the fitting over two of these independent testing sequences are in the first and third rows of Fig. 8. The second and fourth row in the figure illustrates the 3D reconstruction of the face provided by the algorithm.

5. Conclusions

This paper introduced 2D and 3D AAM algorithms for training and fitting that does not require of any pre-alignment of shapes. This was achieved by using the properties of complex spherical distributions, which are typically employed to describe shape vectors invariant to translation, scale and rotation. We have shown how the complexity of this approach, which is germane in the estimation of the spherical representation, can be bypassed by using a rotation invariant kernel and a simple Gaussian model. Experimental results using three datasets and several ASL video sequences show that traditional AAM results are improved in terms of convergence rate and convergence accuracy.

Acknowledgments

Supported in part by the National Science Foundation (0713055) and the National Institutes of Health (R01-DC005241).

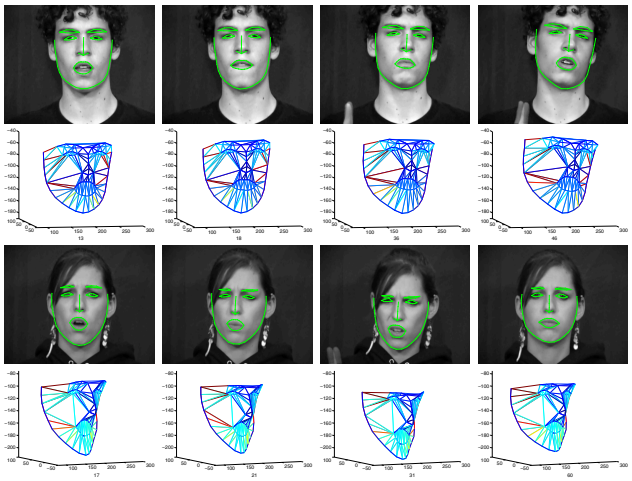


Figure 8. Examples of fitted 3D AAM to frames of unseen ASL sentences and accompanying 3D reconstruction of the face.

References

[1] P. Arias, G. Randall, and G. Sapiro, "Connecting the out-of-sample and pre-image problems in kernel methods," CVPR, 2007.

[2] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," SIGGRAPH, pp. 187–194, 1999.

[3] T.F. Cootes, C.J. Taylor, D.H. Cooper and J. Graham, "Active Shape Models – Their Training and Application," CVIU, pp. 38-59, 1995.

[4] T.F. Cootes, G.J. Edwards and C.J. Taylor, "Active Appearance Models," PAMI, 23(6):681–685, 2001.

[5] F. De la Torre, A. Collet Romea, J. Cohn, and T. Kanade, "Filtered Component Analysis to Increase Robustness to Local Minima in Appearance Models," CVPR, 2007.

[6] L. Ding and A.M. Martinez, "Precise Detailed Detection of Faces and Facial Features," CVPR, 2008.

[7] I.L. Dryden and K.V. Mardia, "Statistical Shape Analysis," John Wiley & Sons, 1998.

[8] L. Gu and T. Kanade, "A Generative Shape Regularization Model for Robust Face Alignment," ECCV, 2008.

[9] O.C. Hamsici and A.M. Martinez, "Spherical-homoscedastic shapes," ICCV, 2007.

[10] O.C. Hamsici and A.M. Martinez, "Spherical-homoscedastic distributions: The equivalency of spherical and Normal distributions in classification," J. Mach. Learn. Res. 8:1583–1623, 2007.

[11] O.C. Hamsici and A.M. Martinez, "Rotation Invariant Kernels and Their Application to Shape Analysis," PAMI, accepted.

[12] H. Jia and A.M. Martinez, "Low-Rank Matrix Fitting Based on Subspace Perturbation Analysis with Applications to Structure from Motion," PAMI, 31(5):841–854, 2009.

[13] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the hausdorff distance," AVBPA, 2001.

[14] K. Messer, J. Matas, J. Kittler, and K. Jonsson, "Xm2vtsdb: The extended m2vts database," AVBPA, 1999.

[15] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz and G. Rätsch, "Kernel PCA and de-noising in feature spaces," NIPS 11, 1999.

[16] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann, "The IMM face database - an annotated dataset of 240 face images," Tech. Rep., Tech. Univ. Denmark, 2004.

[17] S. Romdhani, S. Gong and A. Psarrou, "A Multi-View Nonlinear Active Shape Model Using Kernel PCA," BMVC, 1999.

[18] J. Saragih and R. Goecke, "A Nonlinear Discriminative Approach to AAM Fitting," ICCV, 2007.

[19] H. Wu, X. Liu and G. Doretto, "Face alignment via boosted ranking model," CVPR, 2008.

[20] J. Xiao, S. Baker, I. Matthews and T. Kanade, "Real-time combined 2d+3d active appearance models," CVPR, pp. 535–542, 2004.

[21] J. Zhang, S. K. Zhou, D. Comaniciu and L. Mcmillan, "Discriminative Learning for Deformable Shape Segmentation: A Comparative Study," ECCV, 2008