Separating Appearance from Deformation

Nebojsa Jojic[1], Patrice Simard[1], Brendan J. Frey[2], David Heckerman[1]

[1]Microsoft Research
Redmond, Washington

[2]Adaptive Algorithms Lab, ECE
University of Toronto

Abstract

By representing images and image prototypes by linear subspaces spanned by "tangent vectors" (derivatives of an image with respect to translation, rotation, etc.*), impressive invariance to known types of uniform distortion can be built into feedforward discriminators. We describe a new probability model that can jointly cluster data and learn mixtures of nonuniform, smooth deformation fields. Our fields are based on low-frequency wavelets, so they use very few parameters to model a wide range of smooth deformations (unlike,* e.g.*, factor analysis, which uses a large number of parameters to model deformations). In spirit, our ideas are most similar to the idea of separating content from style published by Tenenbaum and Freeman. However, our models do not need labeled data for training, and thus allow for unsupervised separation of appearance from deformation. We give results on handwritten digit recognition and face recognition.*

1 Introduction

Many computer vision and image processing tasks benefit from invariances to spatial deformations in the image. Examples include handwritten character recognition, face recognition and motion estimation in video sequences. When the input images are subjected to possibly large transformations from a *known* finite set of transformations (*e.g.,* translations in images), it is possible to model the transformations using a discrete latent variable and perform transformation-invariant clustering and dimensionality reduction using EM (Frey and Jojic 1999a; Jojic and Frey 2000). Although this method produces excellent results on practical problems, the amount of computation grows linearly with the total number of possible transformations in the input.

In many cases, we can assume the deformations are small, *e.g.,* due to dense temporal sampling of a video sequence, from blurring the input, or because of well-behaved handwriters. Suppose $(\boldsymbol{\delta}_x, \boldsymbol{\delta}_y)$ is a deformation field (a vector field that specifies where to shift pixel intensity), where $(\delta_{xi}, \delta_{yi})$ is the 2-D real vector associated with pixel i. Given a vector of pixel intensities $\mathbf{f}$ for an image, and assuming the deformation vectors are small, we can approximate the deformed image by

$$\tilde{\mathbf{f}} = \mathbf{f} + \frac{\partial \mathbf{f}}{\partial x} \circ \boldsymbol{\delta}_x + \frac{\partial \mathbf{f}}{\partial y} \circ \boldsymbol{\delta}_y, \tag{1}$$

where $\circ$ is element-wise product and $\partial \mathbf{f}/\partial x$ is a gradient image computed by shifting the original image to the right a small amount and then subtracting off the original image. Suppose $\boldsymbol{\delta}_y = \mathbf{0}$ and $\boldsymbol{\delta}_x = \alpha \mathbf{1}$, where α is a scalar. Then, (1) shifts the image to the right by an amount proportional to α. Fig. 1 shows some more complex examples of deformations computed in this way.

Simard *et al.* (1992, 1993) considered a deformation field that is a linear combination of the uniform fields for translation, rotation, scaling and shearing plus the nonuniform field for line thickness. When the deformation field is parameterized by a scalar α (*e.g.,* x-translation), $\frac{\partial \mathbf{f}}{\partial x} \circ \boldsymbol{\delta}_x + \frac{\partial \mathbf{f}}{\partial y} \circ \boldsymbol{\delta}_y$ can be viewed as the gradient of $\mathbf{f}$ with respect to α. Since the above approximation holds for small α, this gradient is tangent to the true 1-D deformation manifold of $\mathbf{f}$.

By processing the input from coarse to fine resolution, this tangent-based construction of a deformation field has also be used to model large deformations in an approximate manner (Vasconcelos and Lippman 1998).

The tangent approximation can also be included in generative models, including linear factor analyzer models (Hinton *et al.,* 1997) and nonlinear generative models (Jojic and Frey 2000).

Another approach to modeling small deformations is to jointly cluster the data and *learn* a locally linear deformation model for each cluster, *e.g.,* using EM in a factor analyzer (Ghahramani and Hinton 1997). An advantage of this approach over the tangent approach is that the types of deformation need not be specified beforehand. So, unknown, nonuniform types of deformation can be learned. However, a large amount of data is needed to accurately model the deformations, and learning is susceptible to local optima that confuse deformed data from one cluster with data from another cluster. (Some factors tend to "erase" parts of the image and "draw" new parts, instead of just perturbing the image.)

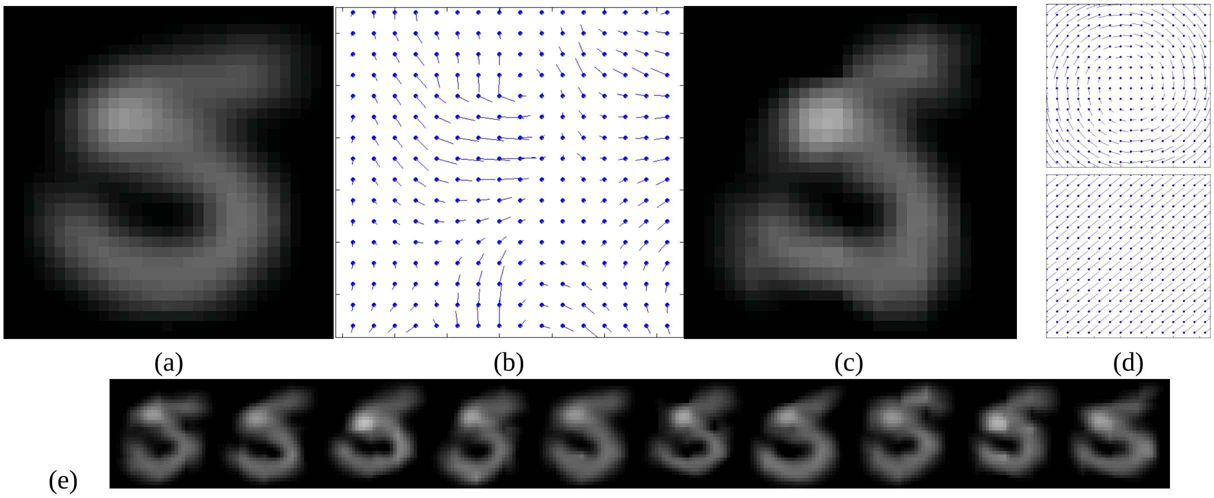


Figure 1. (a) An image of a hand-written digit. (b) A smooth, non-uniform deformation field. (c) The resulting deformed image. (d) Rotation and translation deformation fields. (e) Examples of deformed images produced by learned distributions over wavelet-based fields.

We describe a new probability model that can jointly cluster data and learn mixtures of nonuniform, smooth deformation fields. In contrast to the tangent approach, where the deformation field is a linear combination of prespecified uniform deformation fields (such as translation), in our model the deformation field is a linear combination of low-frequency wavelets. A mixture model of these wavelet coefficients is learned from the data, so our model can capture multiple types of nonuniform, smooth image deformations. In contrast to factor analysis, using a low-frequency wavelet basis allows our model to use significantly fewer parameters to represent a wide range of realistic deformations. For example, our model is much less likely to use a deformation field to "erase" part of an image and "draw" a new part, since the necessary field is usually not smooth.

In contrast with usual probabilistic deformation models, our generative model also incorporates the idea of "symmetric tangent distance" (Simard *et al*, 1993) by including deformations of the observed image. This leads to the idea of maximizing the likelihood of matching an observed image, instead of the likelihood of generating the observed image. However, under easily satisfiable conditions, the model reduces to the pure generative model. This symmetry allows the linear model for deformations to hold for larger transformations, as the prototype image and the observed image are both deformed to achieve a match.

In the computer vision community, the linear models of image deformations are often used in motion estimation and thus in this respect our model has some similarities with (Wu et al, 1997; Black et al, 1997; Hager and Belhumeur, 1998; Fleet et al, 2000).

While the linear subspace is usually fixed (for instance by computing image derivatives with respect to certain transformations of interest), some authors developed algorithms that can learn the characteristics of the more general classes of deformation fields. For example, (Black et al, 1997; Fleet et al, 2000) proposed a framework for learning deformation models from optical flow. They do not learn appearance models, though, as they are interested in motion from frame to frame.

Among many deformation models published in the computer vision and medical imaging communities, our model shares most similarities with the idea of separating style and content (Tenenbaum and Freeman, 1997), as our deformation linearization also leads to a bilinear model. Our model, however, is specialized for modeling deformations and does not need labeled data for training. The key aspect of our model is that it allows joint learning of appearance and deformation models.

A need for such a capability was noted in a number of papers on nonlinear face modeling that use linear models of shape and appearance and combine them using warping. In (Walker et al, 2000) manual labeling of landmarks for active appearance models was avoided by salient feature selection based on a heuristic. In their face models, Jones and Poggio used a bootstrapping technique of (Vetter et al, 1997) which iteratively warps images using the model and optical flow and updates the parameters of the shape and appearance subspaces. Ultimately, these models rely on having the deformation information for images in the training set from some source outside the pure data and the model itself (either from optical flow with respect to a reference image or from point correspondences based on local template matching).

We are particularly interested in solving the chicken and egg problem: knowing appearance model seems to be needed in order to find the deformations in images, while knowing the deformations in training images is necessary for learning the appearance. We train our model with no additional information then the data itself using the EM algorithm that guarantees the increase in the likelihood of the data in each iteration. Our experimental results indicate that jointly learning the deformation models and appearance models from the data without landmarks is possible.

2 Smooth, wavelet-based deformation fields

We ensure the deformation field $(\boldsymbol{\delta}_x, \boldsymbol{\delta}_y)$ is smooth by constructing it from low-frequency wavelets,

$$\boldsymbol{\delta}_x = \mathbf{R}\mathbf{a}_x, \qquad \boldsymbol{\delta}_y = \mathbf{R}\mathbf{a}_y, \tag{2}$$

where the columns of $\mathbf{R}$ contain low-frequency wavelet basis vectors, and $\mathbf{a} = \begin{bmatrix} \mathbf{a}_x \\ \mathbf{a}_y \end{bmatrix}$ are the deformation coefficients. We use a number of deformation coefficients that is a small fraction of the number of pixels in the image. (In contrast, each factor in factor analysis has a number of coefficients that is *equal* to the number of pixels.)

An advantage of wavelets is their space/frequency localization. The global trends in the image can be captured in the low-frequency coefficients while at the same time, the deformations localized in smaller regions of the image can be expressed by more spatially localized wavelets.

The deformed image can be expressed as

$$\tilde{\mathbf{f}} = \mathbf{f} + (\mathbf{G}_x\mathbf{f}) \circ (\mathbf{R}\mathbf{a}_x) + (\mathbf{G}_y\mathbf{f}) \circ (\mathbf{R}\mathbf{a}_y), \tag{3}$$

where the derivatives in (1) are approximated by sparse matrices $\mathbf{G}_x$ and $\mathbf{G}_y$ that operate on $\mathbf{f}$ to compute finite differences.

(3) is bilinear in the deformation coefficients $\mathbf{a}$ and the original image $\mathbf{f}$, *i.e.,* it is linear in $\mathbf{f}$ given $\mathbf{a}$ and it is linear in $\mathbf{a}$ given $\mathbf{f}$. To rewrite the element-wise product as a matrix product, we convert either the vector $\mathbf{Gf}$ or the vector $\mathbf{Ra}$ to a diagonal matrix using the $\mathrm{diag}()$ function:

$$\begin{aligned} \tilde{\mathbf{f}} &= \mathbf{f} + \mathbf{D}(\mathbf{f})\mathbf{a}, \qquad &(4)\\ &\text{where } \mathbf{D}(\mathbf{f}) = [\mathrm{diag}(\mathbf{G}_x\mathbf{f})\mathbf{R} \quad \mathrm{diag}(\mathbf{G}_y\mathbf{f})\mathbf{R}] \\ \tilde{\mathbf{f}} &= \mathbf{T}(\mathbf{a})\mathbf{f}, &(5)\\ &\text{where } \mathbf{T}(\mathbf{a}) = [\mathbf{I} + \mathrm{diag}(\mathbf{R}\mathbf{a}_x)\mathbf{G}_x + \mathrm{diag}(\mathbf{R}\mathbf{a}_y)\mathbf{G}_y]\,. \end{aligned}$$

The first equation shows by applying a simple pseudo inverse, we can estimate the coefficients of the image deformation that transforms $\mathbf{f}$ into $\tilde{\mathbf{f}}$: $\mathbf{a} = \mathbf{D}(\mathbf{f})^{-1}(\tilde{\mathbf{f}} - \mathbf{f})$. This low-dimensional vector of coefficients minimizes the distance $||\mathbf{f} - \tilde{\mathbf{f}}||$. Under easily satisfied conditions on the differencing matrices $\mathbf{G}_x$ and $\mathbf{G}_y$, $\mathbf{T}(\mathbf{a})$ in (5) can be made invertible regardless of the image $\mathbf{f}$, so that $\mathbf{f} = \mathbf{T}(\mathbf{a})^{-1}\tilde{\mathbf{f}}$.

Given a test image $\mathbf{g}$, we could match $\mathbf{f}$ to $\mathbf{g}$ by computing the deformation coefficients, $\mathbf{a} = \mathbf{D}(\mathbf{f})^{-1}(\mathbf{g} - \mathbf{f})$, that minimize $||\mathbf{f} - \mathbf{g}||$. However, more extreme deformations can be successfully matched by deforming $\mathbf{g}$ as well:

$$\tilde{\mathbf{g}} = \mathbf{g} + (\mathbf{G}_x\mathbf{g}) \circ (\mathbf{R}\mathbf{b}_x) + (\mathbf{G}_y\mathbf{g}) \circ (\mathbf{R}\mathbf{b}_y), \tag{6}$$

where $\mathbf{b}$ are the deformation coefficients for $\mathbf{g}$. The difference between the two deformed images is

$$\tilde{\mathbf{f}} - \tilde{\mathbf{g}} = \mathbf{f} - \mathbf{g} + [\mathbf{D}(\mathbf{f}) - \mathbf{D}(\mathbf{g})] \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}. \tag{7}$$

Again, minimizing $||\tilde{\mathbf{f}} - \tilde{\mathbf{g}}||$ is a simple quadratic optimization with respect to the deformation coefficients $\mathbf{a}$, $\mathbf{b}$. To favor some deformation fields over others, we can include a cost term that depends on the deformation coefficients.

Finally, a versatile image distance can be defined as:

$$d(\mathbf{f},\mathbf{g}) = \min_{\mathbf{a},\mathbf{b}} \left\{ (\tilde{\mathbf{f}} - \tilde{\mathbf{g}})'\boldsymbol{\Psi}^{-1}(\tilde{\mathbf{f}} - \tilde{\mathbf{g}}) + [\mathbf{a}' \quad \mathbf{b}']\boldsymbol{\Gamma}^{-1} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \right\} \tag{8}$$

Matrix $\boldsymbol{\Psi}$ is a diagonal matrix whose non-zero elements contain variances of appropriate pixels. This distance allows different pixels to have different importance. For example, if we are matching two images of a tree in the wind, the deformation coefficients should be capable of aligning the trunk and large branches, while the variability in the appearance of the leaves would be captured in $\boldsymbol{\Psi}$. $\boldsymbol{\Gamma}$ captures the covariance structure of the wavelet coefficients of the allowed deformations. This distance can be used in the same applications as tangent distance. However, in the next section we present also a probabilistic model that can be used to jointly learn the appearance and deformation models from training data.

3 Bayes net for deformable image matching

In Fig. 2a we show a Bayes net that can be used to compute the likelihood that the input image matches the images modeled by the network. For classification, we learn one of these networks for each class of data.

The generative matching process begins by clamping the test image $\mathbf{g}$. Then, an image cluster index c is drawn from $P(c)$ and given c, a latent image $\mathbf{f}$ is drawn from a Gaussian, $\mathcal{N}(\mathbf{f}; \boldsymbol{\mu}_c, \boldsymbol{\Phi}_c)$. In this paper, we assume $\boldsymbol{\Phi}_c = \mathbf{0}$, so $p(\mathbf{f}|c) = \delta(\mathbf{f} - \boldsymbol{\mu}_c)$. This allows us to use exact EM to learn the parameters of the model. We are investigating techniques which would allow us to learn $\boldsymbol{\Phi}_c$ as well.

Next, a deformation type index ℓ is picked according to $P(\ell|c)$. This index determines the covariance $\boldsymbol{\Gamma}_l$ of the deformation coefficients for both the latent image $\mathbf{f}$ and the test image $\mathbf{g}$:

$$p(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} | \ell) = \mathcal{N}(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}; \mathbf{0}, \boldsymbol{\Gamma}_\ell). \tag{9}$$

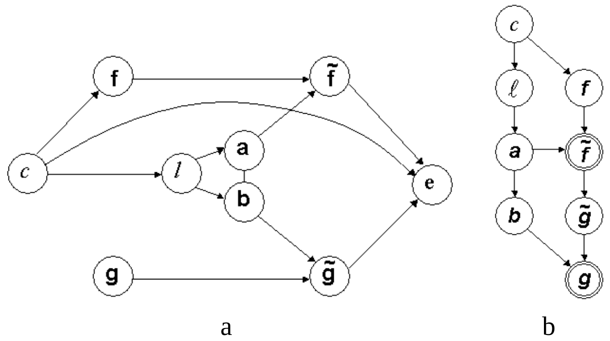


Figure 2. (a) A Bayes net for deformable image matching. (b) A generative version of the net conditioned on $\mathbf{e} = \mathbf{0}$.

$\mathbf{\Gamma}_\ell$ could be a diagonal matrix with larger elements corresponding to lower-frequency basis functions, to capture a wide range of smooth non-uniform deformations. However, $\mathbf{\Gamma}_\ell$ could also capture correlations among deformations in different parts of the image. The deformation coefficients for the latent image $\mathbf{a}$ and for the observed image $\mathbf{b}$ should be strongly correlated, so we model the joint distribution instead of modeling $\mathbf{a}$ and $\mathbf{b}$ separately.

Once the deformation coefficients $\mathbf{a}$, $\mathbf{b}$ have been generated, the deformed latent image $\tilde{\mathbf{f}}$ and the deformed test image $\tilde{\mathbf{g}}$ are produced from $\mathbf{f}$ and $\mathbf{g}$ according to (3) and (6). Using the functions $\mathbf{D}()$ and $\mathbf{T}()$ introduced above, we have

$$p(\tilde{\mathbf{f}}|\mathbf{f},\mathbf{a}) = \delta(\tilde{\mathbf{f}} - \mathbf{f} - \mathbf{D}(\mathbf{f})\mathbf{a}) = \delta(\tilde{\mathbf{f}} - \mathbf{T}(\mathbf{a})\mathbf{f}), \quad (10)$$

$$p(\tilde{\mathbf{g}}|\mathbf{g},\mathbf{b}) = \delta(\tilde{\mathbf{g}} - \mathbf{g} - \mathbf{D}(\mathbf{g})\mathbf{b}) = \delta(\tilde{\mathbf{g}} - \mathbf{T}(\mathbf{b})\mathbf{g}). \quad (11)$$

As an illustration of the generative process up to this point, in Fig. 1 we show several images produced by randomly selecting 8 deformation coefficients from a unit-covariance Gaussian and applying the resulting deformation field to an image.

The last random variable in the model is an error image $\mathbf{e}$ (called a "reference signal" in control theory), which is formed by adding a small amount of diagonal Gaussian noise to the difference between the deformed images $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{g}}$:

$$p(\mathbf{e}|\tilde{\mathbf{f}},\tilde{\mathbf{g}},c) = \mathcal{N}(\mathbf{e};\tilde{\mathbf{f}} - \tilde{\mathbf{g}},\mathbf{\Psi}_c). \quad (12)$$

For good model parameters, it is likely that one of the cluster means can be slightly deformed to match a slightly deformed observed image. However, due to the constrained nature of these deformations, an exact match may not be achievable. Thus, to allow an exact match, the model helps the image difference with a small amount of non-uniform, cluster dependent noise. $\mathbf{\Psi}_c$ is diagonal and the non-zero elements contain the pixel variances. A natural place to include cluster dependence is in fact in the cluster noise $\mathbf{\Phi}_c$. Since we have chosen to collapse this noise model to zero, it is helpful to add cluster dependence into $\mathbf{\Psi}_c$.

This model can now be used to evaluate how likely it is to achieve a zero error image $\mathbf{e}$ by randomly selecting hidden variables conditioned on their parents in the fashion described above. If the model has the right cluster means, right noise levels and the right variability in the deformation coefficients, then the likelihood $p(\mathbf{e} = 0|\mathbf{g})$ will be high. Thus, this likelihood can be used for classification of images when the parameters of the models for different classes are known. Also, we can use the EM algorithm to estimate the parameters of the model that will maximize this likelihood for all observed images $\mathbf{g}_t$ in a training data set (see the Appendix).

By conditioning on $\mathbf{e} = \mathbf{0}$, we can transform the network into the generative network shown in Fig. 2b.[1]

After collapsing the deterministic nodes in the network, the joint distribution conditioned on the input $\mathbf{g}$ is

$$p(c,l,\mathbf{a},\mathbf{b},\mathbf{e}|\mathbf{g}) = P_{c,l}\mathcal{N}(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix};0,\mathbf{\Gamma}_\ell)\cdot$$
$$\cdot\,\mathcal{N}(\mathbf{e};\boldsymbol{\mu}_c + \mathbf{D}(\boldsymbol{\mu}_c)\mathbf{a} - \mathbf{g} - \mathbf{D}(\mathbf{g})\mathbf{b},\mathbf{\Psi}_c) \quad (13)$$

By integrating out the deformation coefficients we obtain $p(c,\ell,\mathbf{e}|\mathbf{g}) = P_{c,\ell}\mathcal{N}\left(\mathbf{e};\boldsymbol{\mu}_c - \mathbf{g},[\mathbf{\Psi}_c^{-1} - \mathbf{\Psi}_c^{-1}\mathbf{M}_c\mathbf{\Omega}_{c,\ell}\mathbf{M}_c'\mathbf{\Psi}_c^{-1}]^{-1}\right)$, where $\mathbf{M}_c = \left[\mathbf{D}(\boldsymbol{\mu}_c) \quad - \mathbf{D}(\mathbf{g})\right]$ and $\mathbf{\Omega}_{c,\ell} = (\mathbf{\Gamma}_\ell^{-1} + \mathbf{M}_c'\mathbf{\Psi}_c^{-1}\mathbf{M}_c)^{-1}$. This density function can be normalized over c, ℓ to obtain $P(c,\ell|\mathbf{e},\mathbf{g})$. The likelihood can be computed by summing over the class and transformation indices:

$$p(\mathbf{e}|\mathbf{g}) = \quad (14)$$
$$\sum_{c,l} P_{c,l}\mathcal{N}\left(\mathbf{e};\boldsymbol{\mu}_c - \mathbf{g},[\mathbf{\Psi}_c^{-1} - \mathbf{\Psi}_c^{-1}\mathbf{M}_c\mathbf{\Omega}_{c,\ell}\mathbf{M}_c'\mathbf{\Psi}_c^{-1}]^{-1}\right)$$

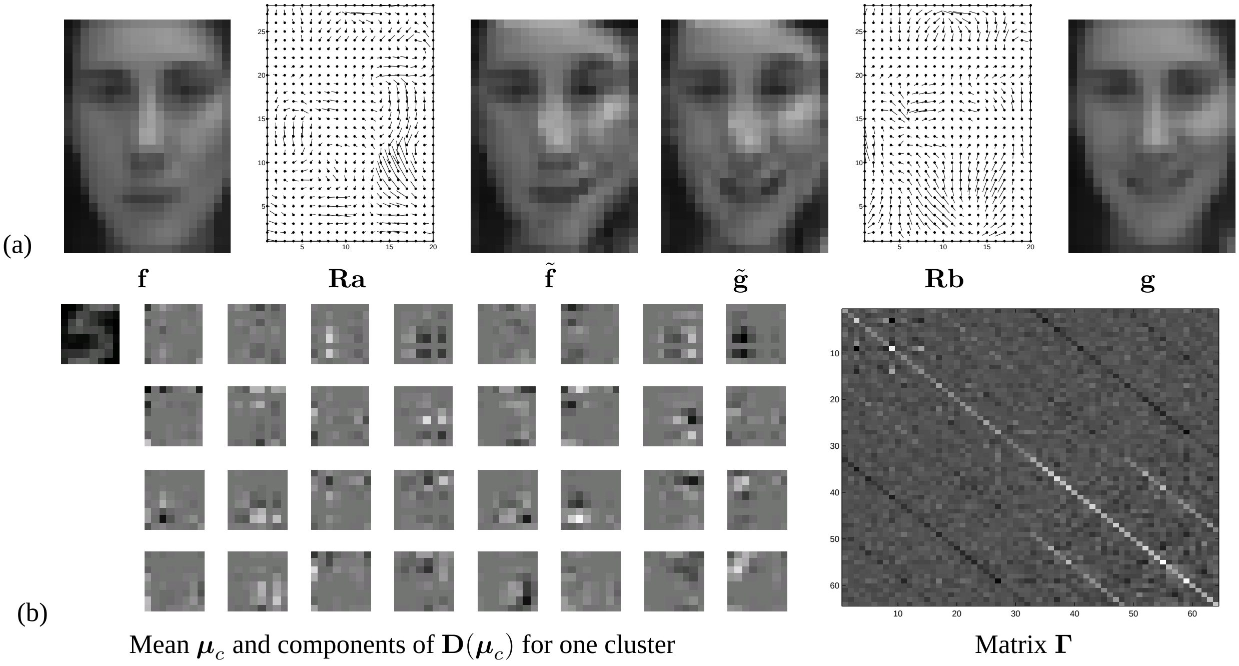
By using this likelihood instead of the distance measure in (8), we are integrating over all possible deformations instead of finding the optimal deformation (which is given by (18) in the Appendix).

4 Experiments

We tested our algorithm on 20x28 greyscale images of people with different facial expressions and 8x8 greyscale images of handwritten digits from the CEDAR CDROM (Hull, 1994).

Deformable image matching. In Fig. 3a we estimate the optimal deformation fields necessary to match two images of a face of the same person but with different facial expression. We set the $\mathbf{\Psi}$ matrix to identity and we set $\mathbf{\Gamma}$ by hand to allow a couple of pixels of deformations. See Section 2 for nomenclature. In short, the two images $\mathbf{f}$ and $\mathbf{g}$ are shown left and right and the estimated flow fields that bring them together are shown next to each of them. In the middle are

[1] To do so in a straightforward fashion, we assume that $|\mathbf{T}(\mathbf{b})| = 1$.



Mean $\boldsymbol{\mu}_c$ and components of $\mathbf{D}(\boldsymbol{\mu}_c)$ for one cluster

Matrix $\boldsymbol{\Gamma}$

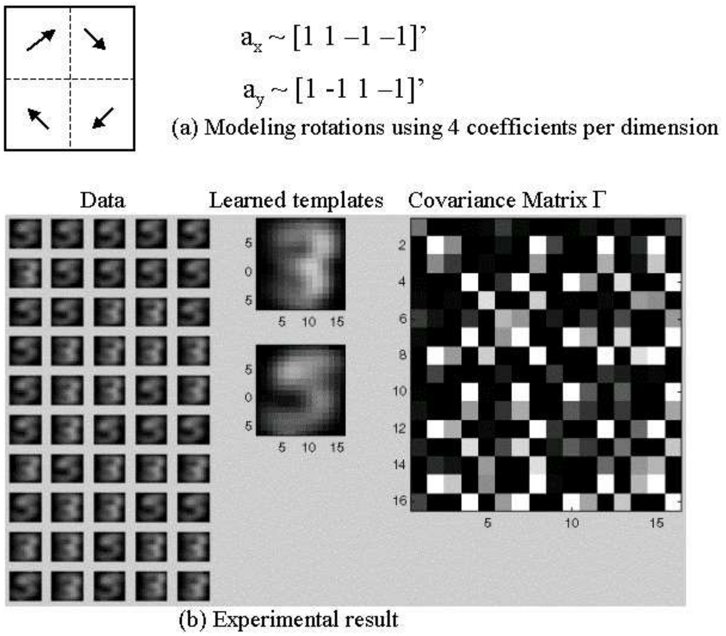
Figure 3. Estimating the image deformation due to a change in facial expression and a subset of the learned parameters for the model of handwritten digits

the deformed versions of the two images that together with the motion fields illustrate that the estimated deformations act to stretch the face on the right and make the face on the left smile, thus ending in similar images.

Comparison with the mixture of diagonal Gaussians (MDG). To compare our method with other generative models, we used a training set of 2000 images to learn 10 digit models using the EM algorithm and tested the algorithms on a test set of 1000 digit images. MDG needs 10-20 classes per digit to achieve the optimal error rate of only about 8% (Frey and Jojic 1999a) on the handwritten digit recognition task. Note that our network reduces to MDG when $\boldsymbol{\Gamma}_\ell$ is set to zero. To demonstrate the effectiveness of adding a deformation model to MDG, we trained our model with 15 classes per digit and only a single transformation model ($L = 1$) for all digits, with a total of 64 deformation coefficients (8 for each dimension in the latent and the observed images). In Fig. 3b we show one of the learned cluster means, the components in the corresponding deformation matrix $\mathbf{D}$ and the learned covariance matrix $\boldsymbol{\Gamma}$. $\boldsymbol{\Gamma}$ shows anticorrelation among the deformation coefficients for the latent and the observed image, as the network usually applies opposite deformations on these two images to achieve the match. However, there is also strong correlation between $\mathbf{b}_x$ and $\mathbf{b}_y$ and less correlation between $\mathbf{a}_x$ and $\mathbf{a}_y$ as the network uses mostly a rotational adjustment on the input image, while the latent image is more freely deformed (Fig. 1e). Our model achieved the error rate of **3.6%**. Even if we keep only the diagonal elements in $\boldsymbol{\Gamma}$, the model achieves a 5% error rate.

Comparison with factor analysis. In factor analysis (FA) or in a mixture of factor analyzers (MFA), the deformation matrix $\mathbf{D}$ is called factor loading matrix and is not tied to the mean $\boldsymbol{\mu}_c$ as in our model (Fig. 3b). The factor covariance matrix is set to the identity matrix, as the extra freedom in the choice of the factor variances can be captured in the factor loading matrix. So, while FA/MFA try to capture the variability in the data by learning the components in the factor loading matrix and keeping the distribution over the factors fixed, our model does the opposite by tying the factor loading matrix to the mean image and learning the distribution over the factors (deformation coefficients). By doing this, we we are able to expand other images using the same deformation model. This allows us to share the deformation model across clusters and also to deform the input images. The comparable error rate in classification of handwritten digits for FA/MFA (3.3%) and our model (3.6%) indicates that most of the variability in images of handwriten digits can be captured by modeling smooth, non-uniform deformations without allowing full FA learning.

Learning deformation fields. To better illustrate the capability of the algorithm to separate the effects of the pattern's appearance and deformation, we tested our algorithm on a problem of learning two classes of digits, digit 3 and digit 5 from 50 examples in 16X16 resolution, each rotated by a random angle. To allow easier visualization of the result, we used a deformation model with only 4 wavelet



(a) Modeling rotations using 4 coefficients per dimension

(b) Experimental result

Figure 4. Joint appearance and deformation learning in case of rotated training images

coefficients per dimension, which essentially means that the image was divided in 4 quadrants, and in each quadrant the major motion was modeled by a single vector. As illustrated in Fig. 4 , in case of rotation these vectors are highly correlated. For instance, the upper two quadrants have the same drift in the x direction, which is opposite from the x motion in the lower two quadrants. Similarly, y motion is correlated not only among quadrants, but also with the x motion in all quadrants. Thus, the deformation coefficients for rotation should exhibit proportionality as indicated in the figure. Also, in bidirectional deformation, $\mathbf{a}$ and $\mathbf{b}$ coefficients should be correlated, as the best way to match images is to rotate them in opposite directions. Note that $\mathbf{a}'\mathbf{a}$ is a chess-board-like pattern (though not a perfect, uniform chess-board) and thus this type of correlation would produce chess-board-like pattern in the covariance matrix $\mathbf{\Gamma}$. We did not give this information to our algorithm. We only gave it the data in Fig. 4b and set appropriately the wavelet decomposition matrix to the simplest case of 4 coefficients per dimension, and the number of deformation models to one. As shown in the figure, the algorithm separated the two classes and learned proper templates. Also, the estimated deformation correlation matrix $\mathbf{\Gamma}$ exhibits the chess-board-like pattern indicating that the model really learned to favor rotations. Note that the this covariance structure can now be used to rotate *other, previously unseen* patterns by an arbitrary small angle. Thus, by simply taking derivatives and computing the appropriate $\mathbf{R}$ matrices for a new template, the model would be rotation-invariant. This example illustrates the capability of the model to separate the appearance from deformation, and reuse the learned deformation models on new images, something that factor analysios and PCA cannot do, and (Tenenbaum and Freeman, 1997) model would be able to do only if the data were labeled, which is not necessary in our case. While (Black et al, 1997) also propose a technique for learning deformation fields, their technique is geared towards video sequences and is thus based on the analysis of optical flow. It is not capable of jointly estimating the appearance and deformation and is suboptimal if applied to pattern recognition problems. On the other hand, if our technique is applied to a video sequence, in addition to deformation models, our model would learn optimal templates with respect to which the motion should be computed.

5 Conclusions

Our deformable image matching network could be used for a variety of computer vision tasks such as optical flow estimation, deformation invariant recognition and modeling correlations in deformations. For example, our learning algorithm could learn to jointly deform the mouth and eyes when modeling facial expressions.

The key feature of our model is that an exact EM algorithm can be used to jointly learn a mixture of appearances and deformation models from an *unlabeled* dataset. This is possible because the optimal templates that summarize the data are treated as parameters in the model. So, while the deformation model includes a noise model that controls the variability in deformations, our appearance model is at this point limited to a set of templates (although it could be easily extended to a PCA model, as well). In order to add a noise model $\mathbf{\Phi}_c$ that would control the variability in the appearance of each template, we are developing a variational learning technique in which we approximate the non-Gaussian posterior distribution over the deformation and appearance coefficients.

References

A. P. Dempster, N. M. Laird and D. B. Rubin 1977. Maximum likelihood from incomplete data via the EM algorithm. *Proceedings of the Royal Statistical Society* B-39, 1–38.

B. J. Frey and N. Jojic 1999a. Estimating mixture models of images and inferring spatial transformations using the EM algorithm. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Ft. Collins, CO. IEEE Computer Society Press, Los Alamitos, CA.

Z. Ghahramani and G. E. Hinton 1997. The EM algorithm for mixtures of factor analyzers. University of Toronto Technical Report CRG-TR-96-1. Available at www.gatsby.ucl.ac.uk/∼zoubin.

G. E. Hinton, P. Dayan and M. Revow 1997. Modeling the manifolds of images of handwritten digits. *IEEE Trans. on Neural Networks* **8**, 65–74.

N. Jojic and B. J. Frey 2000. Topographic transformation as a discrete latent variable. In S.A. Solla, T. K. Leen, and K.-R. Müller (eds) *Advances in Neural Information Processing Systems 12*, MIT Press, Cambridge, MA.

P. Y. Simard, Y. Le Cun and J. Denker 1993. Efficient pattern recognition using a new transformation distance. In S. J. Hanson, J. D. Cowan and C. L. Giles, *Advances in Neural Information Processing Systems 5*, Morgan Kaufmann, San Mateo, CA.

N. Vasconcelos and A. Lippman 1998. Multiresolution tangent distance for affine invariant classification. In M. I. Jordan and M. I. Kearns and S. A. Solla (eds) *Advances in Neural Information Processing Systems 10*, MIT Press, Cambridge, MA.

J. B. Tenenbaum and W. T. Freeman 1997. Separating style from content. *In Adv. in Neural Info. Proc. Systems*, volume 9, MIT Press, 1997.

M. J. Black, Y. Yacoob, A. D. Jepson, D. J. Fleet 1997. Learning parameterized models of image motion. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 561-567.

G. D. Hager and P. N. Belhumeur 1998. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10), 1025 -1039.

Y.-T. Wu, T. Kanade, J. Cohn and L. Ching-Chung 1998. Optical flow estimation using wavelet motion model. *In Proc. of Intl. Conf on Computer Vision*, pp. 992-998.

D. J. Fleet, M. J. Black, Y. Yacoob, and A. D. Jepson 2000. Design and use of linear models for image motion analysis. *International Journal of Computer Vision*, 36(3), pp. 169–191.

K. N. Walker and T. F. Cootes and C. J. Taylor 2000. Determining correspondences for statistical models of facial appearance. *In Proc. IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, pp. 271-276.

T. Vetter, M. J. Jones and T. Poggio 1997. A bootstrapping algorithm for learning linear models of object classes. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 40 -46.

Appendix: EM for deformable image matching network

To fit the network to a set of training data, we assume that the error images for the training cases are zero ($\mathbf{e}_t = \mathbf{0}$) and estimate the maximum likelihood parameters using EM (Dempster *et al.* 1977). In deriving the M-step, both forms of the deformation equations (4) and (5) are useful, depending on which parameters are being optimized. Using $\langle\rangle$ to denote an average over the training set, the update equations are:

$$
\begin{aligned}
P_{c,\ell} =& \langle P(c,\ell|\mathbf{g}_t)\rangle \qquad (15)\\
\hat{\boldsymbol{\mu}}_c =& \langle \sum_\ell P(c,\ell|\mathbf{g}_t)\mathrm{E}[\mathbf{T}(\mathbf{a})'\mathbf{\Psi}_c^{-1}\mathbf{T}(\mathbf{a})|c,\ell,\mathbf{g}_t]\rangle^{-1}\\
&\cdot \langle \sum_\ell P(c,\ell|\mathbf{g}_t)\mathrm{E}[\mathbf{T}(\mathbf{a})'\mathbf{\Psi}_c^{-1}\mathbf{T}(\mathbf{b})\mathbf{g}_t|c,\ell,\mathbf{g}_t]\rangle, \qquad (16)\\
\hat{\mathbf{\Gamma}}_\ell =& \frac{\left\langle \sum_c P(c,\ell|\mathbf{g}_t)\mathrm{E}\left\{ \left[\begin{array}{c}\mathbf{a}\\ \mathbf{b}\end{array}\right] [\mathbf{a}' \quad \mathbf{b}'] \middle| c,\ell,\mathbf{g}_t \right\}\right\rangle}{\langle \sum_c P(c,\ell|\mathbf{e}_t=\mathbf{0},\mathbf{g}_t)\rangle} \qquad (17)\\
\hat{\mathbf{\Psi}}_c =& \mathrm{diag}\left(\frac{\langle \sum_\ell P(c,l|\mathbf{g}_t)\mathrm{E}[(\tilde{\mathbf{f}}-\tilde{\mathbf{g}}_t)\circ(\tilde{\mathbf{f}}-\tilde{\mathbf{g}}_t)|c,l,\mathbf{g}_t]\rangle}{\langle \sum_\ell P(c,l|\mathbf{g}_t)\rangle} \right)
\end{aligned}
$$

The expectations needed to evaluate the above are given by:

$$
\begin{aligned}
\mathbf{\Omega}_{c,\ell} = \mathrm{cov}\left\{ \left[\begin{array}{c}\mathbf{a}\\ \mathbf{b}\end{array}\right] \middle| c,\ell,\mathbf{g}_t \right\} =& (\mathbf{\Gamma}_\ell^{-1} + \mathbf{M}_c'\mathbf{\Psi}_c^{-1}\mathbf{M}_c)^{-1}\\
\boldsymbol{\gamma}_{c,\ell} = \mathrm{E}\left\{ \left[\begin{array}{c}\mathbf{a}\\ \mathbf{b}\end{array}\right] \middle| c,l,\mathbf{g}_t \right\} =& \mathbf{\Omega}_{c,\ell}^{-1}\mathbf{M}_c'\mathbf{\Psi}_c^{-1}(\boldsymbol{\mu}_c - \mathbf{g}_t) \qquad (18)\\
\mathrm{E}\left\{ \left[\begin{array}{c}\mathbf{a}\\ \mathbf{b}\end{array}\right] [\mathbf{a}' \quad \mathbf{b}'] \middle| c,\ell,\mathbf{g}_t \right\} =& \mathbf{\Omega}_{c,\ell} + \boldsymbol{\gamma}_{c,\ell}\boldsymbol{\gamma}_{c,\ell}' \qquad (19)\\
\mathrm{E}[(\tilde{\mathbf{f}}-\tilde{\mathbf{g}}_t)\circ(\tilde{\mathbf{f}}-\tilde{\mathbf{g}}_t)|c,l,\mathbf{g}_t] =& \mathrm{diag}(\mathbf{M}_c(\mathbf{\Omega}_{c,\ell})\mathbf{M}_c')\\
&+ \left(\boldsymbol{\mu}_c - \mathbf{g}_t + \mathbf{M}_c\boldsymbol{\gamma}_{c,\ell}\right) \circ \left(\boldsymbol{\mu}_c - \mathbf{g}_t + \mathbf{M}_c\boldsymbol{\gamma}_{c,\ell}\right) \qquad (20)
\end{aligned}
$$

Expectations in (16) are computed using

$$
\begin{aligned}
\mathbf{T}(\mathbf{a})'\mathbf{\Psi}_c^{-1}\mathbf{T}(\mathbf{a}) =& \mathbf{\Psi}_c^{-1} + \sum_{d\in\{x,y\}} \mathbf{G}_d'\mathrm{diag}(\mathbf{R}\mathbf{a}_d)\mathbf{\Psi}_c^{-1} \qquad (21)\\
&+ \sum_{d\in\{x,y\}} \mathbf{\Psi}_c^{-1}\mathrm{diag}(\mathbf{R}\mathbf{a}_d)\mathbf{G}_d\\
&+ \sum_{d_1,d_2\in\{x,y\}} \mathbf{G}_{d_1}'\mathbf{\Psi}_c^{-1}\mathrm{diag}(\mathbf{R}\mathbf{a}_{d_1}\mathbf{a}_{d_2}'\mathbf{R}')\mathbf{G}_{d_2}\\
\mathbf{T}(\mathbf{a})'\mathbf{\Psi}_c^{-1}\mathbf{T}(\mathbf{b})\mathbf{g}_t =& \mathbf{\Psi}_c^{-1}\mathbf{g}_t + \sum_{d\in\{x,y\}} \mathbf{G}_d'\mathrm{diag}(\mathbf{R}\mathbf{a}_d)\mathbf{\Psi}_c^{-1}\mathbf{g}_t \qquad (22)\\
&+ \sum_{d\in\{x,y\}} \mathbf{\Psi}_c^{-1}\mathrm{diag}(\mathbf{R}\mathbf{b}_d)\mathbf{G}_d\mathbf{g}_t\\
&+ \sum_{d_1,d_2\in\{x,y\}} \mathbf{G}_{d_1}'\mathbf{\Psi}_c^{-1}\mathrm{diag}(\mathbf{R}\mathbf{a}_{d_1}\mathbf{b}_{d_2}'\mathbf{R}')\mathbf{G}_{d_2}\mathbf{g}_t.
\end{aligned}
$$

Then, the expectations $\mathrm{E}[\mathbf{a}]$ and $\mathrm{E}[\mathbf{b}]$ are the two halves of the vector $\boldsymbol{\gamma}_{c,\ell}$, while $\mathrm{E}[\mathbf{a}_{d_1}\mathbf{a}_{d_2}']$ and $\mathrm{E}[\mathbf{a}_{d_1}\mathbf{b}_{d_2}']$, for $d_1, d_2 \in \{x, y\}$, are square blocks of the matrix in (19).