MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Separating style and content with bilinear models

Joshua B. Tenenbaum, William T. Freeman

TR99-04   December 1999

Abstract

PERCEPTUAL systems routinely separate ¨content¨from ¨style¸, classifying familiar words spoken in an unfamiliar accent, identifying a font or handwriting style across letters, or recognizing a familiar face or object seen under unfamiliar viewing conditions. Yet a general and tractable computational model of this ability to untangle the underlying factors of perceptual observations remains elusive. Existing factor models are either insufficiently rich to capture the complex interactions of perceptually meaningful factors such as phoneme and speaker accent or letter and font, or do not allow efficient learning algorithms. Here we show how perceptual systems may learn to solve these crucial tasks using surprisingly simple bilinear models. We report promising results in three realistic perceptual domains: spoken vowel classification with a benchmark multi-speaker database, extrapolation of fonts to unseen letters, and translation of faces to novel illuminants.

Neural Computation, 1999

MERL – A MITSUBISHI ELECTRIC RESEARCH LABORATORY
http://www.merl.com

Separating style and content with bilinear models

Joshua B. Tenenbaum
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139

William T. Freeman
MERL, a Mitsubishi Electric Research Lab
201 Broadway
Cambridge, MA 02139

TR-99-04 January 1999

Abstract

PERCEPTUAL systems routinely separate "content" from "style", classifying familiar words spoken in an unfamiliar accent, identifying a font or handwriting style across letters, or recognizing a familiar face or object seen under unfamiliar viewing conditions. Yet a general and tractable computational model of this ability to untangle the underlying factors of perceptual observations remains elusive. Existing factor models are either insufficiently rich to capture the complex interactions of perceptually meaningful factors such as phoneme and speaker accent or letter and font, or do not allow efficient learning algorithms. Here we show how perceptual systems may learn to solve these crucial tasks using surprisingly simple bilinear models. We report promising results in three realistic perceptual domains: spoken vowel classification with a benchmark multi-speaker database, extrapolation of fonts to unseen letters, and translation of faces to novel illuminants.

To appear in: Neural Computation*, 1999*

1. First printing, TR99-04, January, 1999

Separating style and content with bilinear models

Joshua B. Tenenbaum
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139

William T. Freeman
MERL, a Mitsubishi Electric Research Lab
201 Broadway
Cambridge, MA 02139

Address correspondence to:
Joshua B. Tenenbaum
Department of Brain and Cognitive Sciences
Massachussets Institute of Technology (E10-120)
Cambridge, MA 02139

TEL: (617) 258-7904
FAX: (617) 253-8335
E-Mail: jbt@psyche.mit.edu

Perceptual systems routinely separate "content" from "style", classifying familiar words spoken in an unfamiliar accent, identifying a font or handwriting style across letters, or recognizing a familiar face or object seen under unfamiliar viewing conditions. Yet a general and tractable computational model of this ability to untangle the underlying factors of perceptual observations remains elusive [Hofstadter, 1985]. Existing factor models [Mardia et al., 1979, Hinton and Zemel, 1994, Ghahramani, 1995, Bell and Sejnowski, 1995, Hinton et al., 1995, Dayan et al., 1995, Hinton and Ghahramani, 1997] are either insufficiently rich to capture the complex interactions of perceptually meaningful factors such as phoneme and speaker accent or letter and font, or do not allow efficient learning algorithms. We present a general framework for learning to solve two-factor tasks using bilinear models, which provide sufficiently expressive representations of factor interactions but can nonetheless be fit to data using efficient algorithms based on the singular value decomposition (SVD) and expectation-maximization (EM). We report promising results on three different tasks in three different perceptual domains: spoken vowel classification with a benchmark multi-speaker database, extrapolation of fonts to unseen letters, and translation of faces to novel illuminants.

1 Introduction

Perceptual systems routinely separate the "content" and "style" factors of their observations, classifying familiar words spoken in an unfamiliar accent, identifying a font or handwriting style across letters, or recognizing a familiar face or object seen under unfamiliar viewing conditions. These and many other basic perceptual tasks have in common the need to process separately two independent factors that underly a set of observations. This paper shows how perceptual systems may learn to solve these crucial two-factor tasks using simple and tractable bilinear models. By fitting such models to a training set of observations, the influences of style and content factors can be efficiently separated in a flexible representation that naturally supports generalization to unfamiliar styles or content classes.

Figure 1 illustrates three abstract tasks that fall under this framework: *classification*, *extrapolation* and *translation*. Examples of these abstract tasks in the domain of typography include *classifying* known characters in a novel font, *extrapolating* the missing characters of an incomplete novel font, or *translating* novel characters from a novel font into a familiar font (see Figure 1). The essential challenge in all of these tasks is the same. A perceptual system observes a training set of data in multiple styles and content classes, and is then presented with incomplete data in an unfamiliar style, missing either content labels (Figure 1a) or whole observations (Figure 1b) or both (Figure 1c). The system must generate the missing labels or observations using only the available data in the new style and what it can learn about the interacting roles of style and content from the training set of complete data.

We describe a unified approach to the learning problems of Figure 1 based on fitting models that discover explicit parameterized representations of *(i)* what the training data of each row have in common independent of column, *(ii)* what the data of each column have in common independent of row, and *(iii)* what all data have in common independent of row and column – the interaction of row and column factors. Such a modular representation naturally supports generalization to new styles or content. For example, we can extrapolate a new style to unobserved content classes (Figure 1b) by combining content and interaction parameters learned during training with style parameters estimated from available data in the new style.

A number of models for the underlying factors of observations have recently been proposed in the literature on unsupervised learning. These include essentially *additive* factor models, as used in principal component analysis [Mardia et al., 1979], independent component analysis [Bell and Sejnowski, 1995], and cooperative vector quantization [Hinton and Zemel, 1994, Ghahramani, 1995], and *hierarchical* factorial models, as used in the Helmholtz machine and its descendants [Hinton et al., 1995, Dayan et al., 1995, Hinton and Ghahramani, 1997].

We model the mapping from style and content parameters to observations as a bilinear mapping. Bilinear models are two-factor models with the mathematical property of *separability*: their outputs are linear in either factor when the other is held constant. Their combination of representational expressiveness and efficient learning procedures enables bilinear models to overcome two principal drawbacks of existing factor models which might be applied to learning the tasks in Figure 1 . In contrast to additive factor models, bilinear models provide for rich factor interactions by allowing factors to *multiplicatively* modulate each other's contributions (see Section 2). Model dimensionality can be adjusted to accomodate data that arise from arbitrarily complex interactions of style and content factors. In contrast to hierarchical factorial models, model fitting can be carried out by efficient techniques well-known from the study of linear models, such as the singular value decomposition (SVD) and the expectation-maximization (EM) algorithm, without having to invoke extensive sampling-based [Hinton et al., 1995] or variational [Dayan et al., 1995] approximations.

Our approach is also related to the "learning to learn" research program [Thrun and Pratt, 1998] – also known as "task transfer" or "multitask learning" [Caruana, 1998]. The central insight of "learning to learn" is that learning problems often come in clusters of related tasks, and thus learners may automatically acquire useful biases for a novel learning task by training on many related ones. Rather than families of related *tasks*, we focus on how learners can exploit the structure in families of related *observations*, bound together by their common styles, content classes, or style × content interaction, to acquire general biases useful for carrying out various tasks on novel observations from the same family. Thus our work is closest in spirit to the "family discovery" approach of [Omohundro, 1995], differing primarily in our focus on bilinear models to parameterize the style-content interaction.

The paper is structured as follows. Section 2 explains and motivates our bilinear modeling approach. Section 3 describes how these models are fit to a training set of observations. Sections 4, 5, and 6 present specific applications of these techniques to the three tasks of classification, extrapolation, and translation, using realistic data from several perceptual domains. Section 7 concludes with a general discussion.

A note on terminology. We will use the terms "style" and "content" generically to refer to any two independent factors underlying a set of perceptual observations. For tasks that require generalization to novel classes of only one factor (Figure 1a,b), we will refer to the variable factor (which changes during generalization) as "style" and the invariant factor (with a fixed set of classes) as “content”. For example, in a task of recognizing familiar words spoken in an unfamiliar accent, we would think of the words as “content” and the accent as “style”. For tasks that require generalization across both factors (Figure 1c), the labels “style” and “content” are arbitrary and we will use them as seems most natural.

2 Bilinear models

We have explored two bilinear models, closely related to each other, which we distinguish by the labels *symmetric* and *asymmetric*. The rest of this section describes these models and illustrates them on a simple data set of face images.

2.1 Symmetric model

In the symmetric model, we represent both style s and content c with vectors of parameters, denoted $\mathbf{a}^s$ and $\mathbf{b}^c$ and with dimensionalities I and J respectively. Let $\mathbf{y}^{sc}$ denote a K-dimensional observation vector in style s and content class c. We assume that $\mathbf{y}^{sc}$ is a bilinear function of $\mathbf{a}^s$ and $\mathbf{b}^c$ given most generally by the form

$$y_k^{sc} = \sum_{i=1}^{I} \sum_{j=1}^{J} w_{ijk} a_i^s b_j^c. \tag{1}$$

Here i, j, and k denote the components of style, content, and observation vectors respectively. [1] The w_{ijk} terms are independent of style and content and characterize the interaction of these two factors. Their meaning becomes clearer when we rewrite Equation 1 in vector form. Letting $\mathbf{W}_k$ denote the $I \times J$ matrix with entries $\{w_{ijk}\}$, Equation 1 can be written as

$$y_k^{sc} = \mathbf{a}^{s^{\mathrm{T}}} \mathbf{W}_k \mathbf{b}^c. \tag{2}$$

In Equation 2, the K matrices $\mathbf{W}_k$ describe a bilinear map from the style and content vector spaces to the K-dimensional observation space.

[1] The model in Equation 1 may appear trilinear, but we view the w_{ijk} terms as describing a fixed bilinear mapping from $\mathbf{a}^s$ and $\mathbf{b}^c$ to $\mathbf{y}^{sc}$.

The interaction terms have another interpretation which can be seen by writing the symmetric model in another vector form. Letting $\mathbf{w}_{ij}$ denote the K-dimensional vector with components $\{w_{ijk}\}$, Equation 1 can be written as

$$\mathbf{y}^{sc} = \sum_{i,j} \mathbf{w}_{ij} a_i^s b_j^c. \tag{3}$$

In Equation 3, the w_{ijk} terms represent $I \times J$ basis vectors of dimension K, and the observation $\mathbf{y}^{sc}$ is generated by mixing these basis vectors with coefficients given by the tensor product of $\mathbf{a}^s$ and $\mathbf{b}^c$.

Of course all of these interpretations are formally equivalent, but they suggest different intuitions which we will exploit later. As a concrete example, Figure 2 illustrates a symmetric model of face images of different people in different poses (sampled from the complete data in Figure 6). Here the basis vector interpretation of the w_{ijk} terms is most natural, by analogy to the well-known work on "eigenfaces" [Turk and Pentland, 1991]. Each pose is represented by a vector of I parameters, a_i^{pose}, and each person by a vector of J parameters, b_j^{person}. To render an image of a particular person in a particular pose, a set of $I \times J$ basis images $\mathbf{w}_{ij}$ is linearly mixed with coefficients given by the tensor product of these two parameter vectors (Equation 3). The symmetric model can exactly reproduce the observations when I and J equal the numbers of styles S and content classes C observed respectively, as is the case in Figure 2. The model provides coarser but more compact representations as these dimensionalities are decreased.

2.2 Asymmetric model

Sometimes linear combinations of a few basis styles learned during training may not describe new styles well. We can obtain more flexible, *asymmetric* models by letting the interaction terms w_{ijk} themselves vary with style. Then Equation 1 becomes $y_k^{sc} = \sum_{i,j} w_{ijk}^s a_i^s b_j^c$. Without loss of generality we can combine the style-specific terms of Equation 1 into

$$a_{jk}^s = \sum_i w_{ijk}^s a_i^s, \tag{4}$$

giving

$$y_k^{sc} = \sum_j a_{jk}^s b_j^c. \tag{5}$$

Again, there are two interpretations of the model corresponding to different vector forms of Equation 5. First, letting $\mathbf{A}^s$ denote the $K \times J$ matrix with entries $\{a_{jk}^s\}$, Equation 5 can be written as

$$\mathbf{y}^{sc} = \mathbf{A}^s \mathbf{b}^c. \tag{6}$$

Here, we can think of the a_{jk}^s terms as describing a style-specific linear map from content space to observation space. Alternatively, letting $\mathbf{a}_j^s$ denote the K-dimensional vector $\{a_{jk}^s\}$, Equation 5 can be written as

$$\mathbf{y}^{sc} = \sum_j \mathbf{a}_j^s b_j^c. \tag{7}$$

Now we can think of the a_{jk}^s terms as describing a set of J style-specific basis vectors which are mixed according to content-specific coefficients b_j^c (independent of style) to produce the observations.

Figure 3 illustrates an asymmetric bilinear model applied to the face database, with head pose as the "style" factor. Now each pose is represented by a set of J basis images $\mathbf{A}^{pose}$ and each person represented by a vector of J parameters $\mathbf{b}^{person}$. To render an image of a particular person in a particular pose, the pose-specific basis images are linearly mixed with coefficients given by the person-specific parameter vector.

Note that the basis images for each pose look like eigenfaces [Turk and Pentland, 1991] in the appropriate style of each pose. However, they do not provide a true orthogonal basis for any one pose, as in [Moghaddam and Pentland, 1997] where a distinct set of eigenfaces is computed for each of several poses. Instead, the factorized structure of the model ensures that corresponding basis vectors play corresponding roles across poses (e.g. the first vector holds (roughly) the mean face for that pose, the second seems to modulate hair distribution, the third seems to modulate head size), which is crucial for adapting to new styles. Familiar content can be easily translated across to a new style by just mixing the new style-specific basis functions with the old content-specific coefficients.

Figure 4 shows the same data represented by an asymmetric model, but with the roles of "style" and "content" switched. Now the $\mathbf{A}^s$ parameters provide a basis set of poses for each person's face. Again, corresponding basis vectors play corresponding roles across styles (e.g. for each person's face, the first vector holds (roughly) the mean pose, the second modulates head orientation, the third modulates amount of hair showing, the fourth adds in facial detail), allowing ready stylistic translation.

Finally, we note that because the asymmetric model can be obtained by summing out redundant degrees of freedom in the symmetric model (Equation 4), [2] the three sets of basis images in Figures 2 - 4 are not at all independent. Both the pose- and person-specific basis images in Figures 3 and 4 can be expressed as linear combinations of the symmetric model basis images in Figure 2, mixed according to the pose- or person-specific coefficients (respectively) from Figure 2.

The asymmetric model's high-dimensional matrix representation of style may be *too* flexible in adapting to data in new styles, and cannot support translation tasks (Figure 1c) because it does not explicitly model the structure of observations that is independent of both style and content (represented by w_{ijk} in Equation 1). However, if overfitting can be controlled by limiting the model dimensionality J or imposing some additional constraint, asymmetric models may solve classification and extrapolation tasks (Figure 1a,b) that could not be solved using symmetric models with a realistic number of training styles.

3 Model fitting

In conventional supervised learning situations, the data are divided into complete training patterns and incomplete (e.g. unlabeled) test patterns, which are assumed to be sampled randomly from the same distribution [Bishop, 1995]. Learning then consists of fitting a model to the training data which allows the missing aspects of the test patterns (e.g. class labels

[2] This only holds exactly when the dimensionalities of style and content vectors are equal to the number of observed styles and content classes, respectively.

in a classification task) to be filled in given the available information. The tasks in Figure 1, however, require that the learner generalize from training data sampled according to one distribution (i.e. in one set of styles and content classes) to test data drawn from a *different* but related distribution (i.e. in a different set of styles and/or content classes).

Because of the need to adapt to a different but related distribution of data during testing, our approach to these tasks involves model fitting during both training and testing phases. In the training phase, we learn about the interaction of style and content factors by fitting a bilinear model to a complete array of observations of C content classes in S styles. In the testing or generalization phase, we adapt the same model to new observations which have something in common with the training set, either in content or style, or in their interaction. The model parameters corresponding to the assumed commonalities are clamped to the values learned during training, and new parameters are estimated for the new styles and/or content using algorithms similar to those used in training. New and old parameters are then combined to accomplish the desired classification, extrapolation, or translation task.

This section presents the basic algorithms for model fitting during training. The algorithms for model fitting during testing are essentially variants of these training algorithms, but they depend on the particular task and thus will be presented in the appropriate sections below.

The goal of model fitting during training is to minimize the total squared error over the training set for the symmetric or asymmetric models. This goal is equivalent to maximum likelihood estimation of the style and content parameters given the training data, under the assumption that the data were generated from the models plus i.i.d. gaussian noise.

3.1 Asymmetric model

Becuase the procedure for fitting the asymmetric model is simpler, we discuss it first. Let $\mathbf{y}(t)$ denote the tth training observation ($t = 1, \ldots, T$). Let the indicator variable $h^{sc}(t) = 1$ if $\mathbf{y}(t)$ is in style s and content class c, and 0 otherwise. Then the total squared error E over the training set for the asymmetric model (in the form of Equation 6) can be written as

$$E = \sum_{t=1}^{T} \sum_{s=1}^{S} \sum_{c=1}^{C} h^{sc}(t) ||\mathbf{y}(t) - \mathbf{A}^s \mathbf{b}^c||^2. \tag{8}$$

If the training set contains equal numbers of observations in each style and in each content class, there exists a closed-form procedure to fit the asymmetric model using the SVD. While we are the first to use this procedure as the basis for a learning algorithm, it is mathematically equivalent to a family of computer vision algorithms [Koenderink and Doorn, 1997] best known in the context of recovering structure from motion of tracked points under orthographic projection [Tomasi and Kanade, 1992].

Let $\bar{\mathbf{y}}^{sc} = \frac{\sum_t h^{sc}(t)\mathbf{y}(t)}{\sum_t h^{sc}(t)}$, the mean observation in style s and content class c. These observations are most naturally represented in a three-way array, but in order to work with standard matrix algorithms, we must stack these SC K-dimensional (column) vectors into a single $(SK) \times C$ observation matrix

$$\bar{\mathbf{Y}} = \begin{bmatrix} \bar{\mathbf{y}}^{11} & \cdots & \bar{\mathbf{y}}^{1C} \\ \vdots & \ddots & \\ \bar{\mathbf{y}}^{S1} & & \bar{\mathbf{y}}^{SC} \end{bmatrix}. \tag{9}$$

We can then write the asymmetric model (see Equation 6) in compact matrix form,[3]

$$\bar{\mathbf{Y}} = \mathbf{AB}, \tag{10}$$

identifying the $(SK) \times J$ matrix $\mathbf{A}$ and $J \times C$ matrix $\mathbf{B}$ as the stacked style and content parameters respectively,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^1 \\ \vdots \\ \mathbf{A}^S \end{bmatrix}, \tag{11}$$

$$\mathbf{B} = \left[\mathbf{b}^1 \cdots \mathbf{b}^C\right]. \tag{12}$$

To find the least-squares optimal style and content parameters for Equation 10, we simply compute the SVD of $\bar{\mathbf{Y}} = \mathbf{USV}^T$. (By convention, we always take the diagonal elements of S

[3] Strictly speaking, Equation 10 is a model of the *mean* observations. However, when the data are evenly distributed across styles and content classes, the parameter values which minimize the total squared error for Equation 10 will also minimize E in Equation 8.

to be ordered by decreasing eigenvalue). We then define the style parameter matrix $\mathbf{A}$ to be the first J columns of $\mathbf{US}$, and the content parameter matrix $\mathbf{B}$ to be the first J rows of $\mathbf{V}^T$. The model dimensionality J can be chosen in various ways: from prior knowledge, by requiring a sufficiently good approximation to the data, or by looking for a "knee" in the singular value spectrum.

If the training data are not distributed equally across the different styles and content classes, we must minimize Equation 8 directly. There are many ways to do this. We use a quasi-newton method (BFGS; [Press et al., 1992]) with initial parameter estimates determined by the SVD of the mean observation matrix $\bar{\mathbf{Y}}$, as described above. If there happen to be no observations in a particular style s and content class c, $\bar{\mathbf{Y}}$ will have some indeterminate (0/0) entries. Before taking the SVD of $\bar{\mathbf{Y}}$, we replace any indeterminate entries by the mean of the observations in the appropriate style s (across all content classes) and/or content class c (across all styles). In our experience, this method has yielded satisfactory results, although it is at least an order of magnitude slower than the closed-form SVD solution. Note that if the training data are *almost* equally distributed across styles and content classes, then the closed-form SVD solution found by assuming the data are exactly balanced will almost minimize Equation 8, and improving this solution via quasi-newton optimization will probably not be worth the much greater effort involved. Because all of the examples presented below have equally distributed training observations, we will need only the closed-form procedure for the remainder of this paper.

3.2 Symmetric model

The total squared error E over the training set for the symmetric model (in the form of Equation 2) can be written as

$$E = \sum_{t=1}^{N}\sum_{s=1}^{S}\sum_{c=1}^{C}\sum_{k=1}^{K} h^{sc}(t)||y_k(t) - \mathbf{a}^{s^{\mathrm{T}}}\mathbf{W}_k\mathbf{b}^c||^2. \tag{13}$$

Again, if we assume the training set consists of an equal number of observations in each style and content class, there are efficient matix algorithms for minimizing E. The algorithm we use was described for scalar observations by [Magnus and Neudecker, 1988] and adapted to vector observations by [Marimont and Wandell, 1992], in the context of characterizing color surface and illuminant spectra. Essentially, we repeatedly apply the above SVD algorithm for fitting the asymmetric model, alternating the role of style and content factors within each iteration until convergence.

First we need a few matrix definitions. Recall that $\bar{\mathbf{Y}}$ consists of the SC K-dimensional mean observation vectors $\bar{\mathbf{y}}^{sc}$ stacked into a single $SK \times C$ matrix (Equation 9). In general, for any $AK \times B$ matrix $\mathbf{X}$ constructed by stacking AB K-dimensional vectors A down and B across, we can define its "vector-transpose" $\mathbf{X}^{\mathrm{VT}}$ to be the $BK \times A$ matrix consisting of the same K-dimensional vectors stacked B down and A across, where the vector a across and b down in $\mathbf{X}$ becomes the vector b across and a down of $\mathbf{X}^{\mathrm{VT}}$. See the illustration in Figure 5. In particular, $\bar{\mathbf{Y}}^{\mathrm{VT}}$ consists of the means $\bar{\mathbf{y}}^{sc}$ stacked into a single $(KC) \times S$ matrix:

$$\bar{\mathbf{Y}}^{\mathrm{VT}} = \begin{bmatrix} \bar{\mathbf{y}}^{11} & \cdots & \bar{\mathbf{y}}^{1S} \\ \vdots & \ddots & \\ \bar{\mathbf{y}}^{C1} & & \bar{\mathbf{y}}^{CS} \end{bmatrix}. \tag{14}$$

Finally, we define the $IK \times J$ stacked weight matrix $\mathbf{W}$, consisting of the IJ K-dimensional basis functions $\mathbf{w}_{ij}$ (see Equation 3) in the form,

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}^{11} & \cdots & \mathbf{w}^{1I} \\ \vdots & \ddots & \\ \mathbf{w}^{J1} & & \mathbf{y}^{IJ} \end{bmatrix}. \tag{15}$$

Its vector-transpose $\mathbf{W}^{\mathrm{VT}}$ is also defined accordingly.

We can then write the symmetric model (see Equation 6) in either of these two equivalent matrix forms,[4]

$$\begin{aligned} \bar{\mathbf{Y}} &= \left[\mathbf{W}^{\mathrm{VT}}\mathbf{A}\right]^{\mathrm{VT}}\mathbf{B}, & (16) \\ \bar{\mathbf{Y}}^{\mathrm{VT}} &= \left[\mathbf{W}\mathbf{B}\right]^{\mathrm{VT}}\mathbf{A}, & (17) \end{aligned}$$

[4] As with the asymmetric model, when the data are evenly distributed across styles and content classes, the parameter values that minimize the total squared error for the mean observations in Equations 17 or 17 will also minimize E in Equation 13.

identifying the $I \times S$ matrix $\mathbf{A}$ and the $J \times C$ matrix $\mathbf{B}$ as the stacked style and content parameter vectors respectively,

$$\mathbf{A} = \left[\mathbf{a}^1 \cdots \mathbf{a}^S\right], \mathbf{B} = \left[\mathbf{b}^1 \cdots \mathbf{b}^C\right]. \tag{18}$$

The iterative procedure for estimating least-squares optimal values of $\mathbf{A}$ and $\mathbf{B}$ proceeds as follows. We initialize $\mathbf{B}$ using the closed-form SVD procedure described above for the asymmetric model (i.e. Equation 12). Note that this initial $\mathbf{B}$ is an orthogonal matrix (i.e. $\mathbf{B}\mathbf{B}^{\mathrm{T}}$ is the $J \times J$ identity matrix), so that $[\bar{\mathbf{Y}}\mathbf{B}^{\mathrm{T}}]^{\mathrm{VT}} = \mathbf{W}^{\mathrm{VT}}\mathbf{A}$ (from Equation 16). Thus, given this initial estimate for $\mathbf{B}$, we can compute the SVD of $[\bar{\mathbf{Y}}\mathbf{B}^{\mathrm{T}}]^{\mathrm{VT}} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ and update our estimate of $\mathbf{A}$ to be the first I rows of $\mathbf{V}^T$. This $\mathbf{A}$ is also orthogonal, so that $\left[\bar{\mathbf{Y}}^{\mathrm{VT}}\mathbf{A}^{\mathrm{T}}\right]^{\mathrm{VT}} = \mathbf{W}\mathbf{B}$ (from Equation 17). Thus, given this estimate of $\mathbf{A}$, we can compute the SVD of $[\bar{\mathbf{Y}}^{\mathrm{VT}}\mathbf{A}^{\mathrm{T}}]^{\mathrm{VT}} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ and update our estimate of $\mathbf{B}$ to be the first J rows of $\mathbf{V}^T$. This completes one iteration of the algorithm. Typically, convergence occurs within 5 iterations (i.e. around 10 SVD operations). Convergence is also guaranteed; see [Magnus and Neudecker, 1988] for a proof for the scalar case ($K = 1$) which can easily be extended to the vector case considered here. Upon convergence, we solve for $\mathbf{W} = [[\bar{\mathbf{Y}}\mathbf{B}^{\mathrm{T}}]^{\mathrm{VT}}\mathbf{A}^{\mathrm{T}}]^{\mathrm{VT}}$ to obtain the basis vectors independent of both style and content. As with the asymmetric model, if the training data are not distributed equally across the different styles and content classes, we minimize Equation 8 starting from the same initial estimates for $\mathbf{A}$ and $\mathbf{B}$ but using a more costly quasi-newton method.

4 Classification

Many common classification problems involve multiple observations likely to be in one style, for example, recognizing the handwritten characters on an envelope or the accented speech of a telephone voice. People are significantly better at recognizing a familiar word spoken in an unfamiliar voice or a familiar letter character written in an unfamiliar font when it is embedded in the context of other words or letters in the same novel style [Bergem et al., 1988, Sanocki, 1992, Nygaard and Pisoni, 1998], presumably because the added context allows the perceptual system to build a model of the new style and factor out its influence.

In this section, we show how a perceptual system may use bilinear models and assumptions of style consistency to factor out the effects of style in content classification, and thereby significantly improve classification performance on data in novel styles. We first describe the two concrete tasks investigated. We then present the general classification algorithm and the results of several experiments comparing this algorithm to standard techniques from the pattern recognition literature, such as nearest neighbor classification, which do not explicitly model the effects of style on content classification.

4.1 Task specifics

We report experiments with two data sets: a benchmark speech data set and a face data set which we collected ourselves. The speech data consist of 6 samples of each of 11 vowels (content classes) uttered by 15 speakers (styles) of British English (originally collected by David Deterding and available from the UC Irvine machine learning repository [5].) Each data vector consists of $K = 10$ log area parameters, a standard vocal tract representation computed from a linear predictive analysis of the digitized speech. The specific task we must learn to perform is classification of spoken vowels (content) for new speakers (styles).

The face data were introduced in Section 2 (Figures 2-4). Figure 6 shows the complete data set, consisting of images of 11 people's faces (styles) viewed under 15 different poses (content classes). The poses span a grid of three vertical positions (up, level, down) and five horizontal positions (far-left, left, straight-ahead, right, far-right). The pictures were shifted to align the nose tip position, found manually. The images were then blurred and cropped to 22×32 pixels, and represented simply as vectors of $K = 704$ pixel values each. The specific task we must learn to perform is classification of head pose (content) for new people's faces (styles).

[5] http://www.ics.uci.edu/AI/ML/Machine-Learning.html

4.2 Algorithm

For both speech and face data sets, we train on observations in all content classes but in a subset of the available styles (the "training" styles). We fit asymmetric bilinear models (Equation 10) to this training data using the closed-form SVD procedure described in Section 3.1. This yields a $K \times J$ matrix $\mathbf{A}^s$ representing each style s and a J-dimensional vector $\mathbf{b}^c$ representing each content class c. The model dimensionality J is a free parameter, which we discuss in depth at the end of this section.

The generalization task is then to classify observations in the remaining styles (the "test" styles), i.e. to fill in the "missing" content labels for these novel observations using the style-invariant content vectors $\mathbf{b}^c$ learned during training (see Figure 6). Observe that trying to estimate both content labels as well as style parameters for the new data presents a classic "chicken-and-egg" problem, very much like the problems of k-means clustering or mixture modeling [Duda and Hart, 1973]. If the content class assignments were known, then it would be easy to estimate parameters for a new style $\tilde{s}$ by simply inserting into the basic asymmetric bilinear model (i.e. Equation 6) all the observation vectors in style $\tilde{s}$, along with the appropriate content vectors $\mathbf{b}^c$ and solving for the style matrix $\mathbf{A}^{\tilde{s}}$. Similarly, if we had a model $\mathbf{A}^{\tilde{s}}$ of new style $\tilde{s}$, then we could classify any test observation from this new style based simply on its distance to each of the known content vectors $\mathbf{b}^c$ multiplied by $\mathbf{A}^{\tilde{s}}$. [6] Initially, however, both style models and content class assignments are unknown for the test data.

To handle this uncertainty, we embed the bilinear model within a gaussian mixture model to yield a *separable mixture model* (SMM) [Tenenbaum and Freeman, 1997] which can then be fit efficiently to new data using the EM algorithm [Dempster et al., 1977]. The mixture model has $S \times C$ gaussian components, one for each pair of S styles and C content classes, with means given by the predictions of the bilinear model. However, only $\mathcal{O}(S + C)$ parameters are needed to represent the means of these $S \times C$ gaussians, because of the bilinear model's separable structure. To simultaneously classify known content in new styles and estimate new

[6] This is equivalent to maximum likelihood classification, assuming gaussian likelihood functions centered on the predictions of the bilinear model $\mathbf{A}^{\tilde{s}}\mathbf{b}^c$.

style parameters, the EM algorithm alternates between estimating the most likely content labels given current style parameter estimates (E-step) and estimating the most likely style parameters given current content label estimates (M-step), with likelihood determined by the gaussian mixture model. If in addition the test data are not segmented according to style, style labels can be estimated simultaneously as part of the E-step.

More formally, after training on labeled data from S styles and C content classes, we are given test data from the same C content classes and $\tilde{S}$ new styles, with labels for content (and possibly also style) missing. We assume that the probability of a new unlabeled observation $\mathbf{y}$ being generated in new style $\tilde{s}$ and old content c is given by a gaussian distribution of variance σ^2 centered at the prediction of the bilinear model: $p(\mathbf{y}|\tilde{s}, c) \propto \exp\{-\|\mathbf{y} - \mathbf{A}^{\tilde{s}}\mathbf{b}^c\|^2/(2\sigma^2)\}$. The total probability of $\mathbf{y}$ is then given by the mixture distribution $p(\mathbf{y}) = \sum_{\tilde{s},c} p(\mathbf{y}|\tilde{s}, c)p(\tilde{s}, c)$. Here we assume equal prior probabilities $p(\tilde{s}, c)$, unless the observations are otherwise labeled. [7] The content vectors $\mathbf{b}^c$ are known from training. The EM algorithm [Dempster et al., 1977] alternates between two steps in order to find new style matrices $\mathbf{A}^{\tilde{s}}$ and style-content labels $p(\tilde{s}, c|\mathbf{y})$ that best explain the test data. In the E-step, we compute the probabilities $p(\tilde{s}, c|\mathbf{y}) = p(\mathbf{y}|\tilde{s}, c)p(\tilde{s}, c)/p(\mathbf{y})$ that each test vector $\mathbf{y}$ belongs to style $\tilde{s}$ and content class c, given the current style matrix estimates. In the M-step, we estimate new style matrices by setting $\mathbf{A}^{\tilde{s}}$ to maximize the total loglikelihood of the test data, $L^* = \sum_{\mathbf{y}} \log p(\mathbf{y})$. The M-step can be computed in closed form by solving the equations $\partial L^*/\partial \mathbf{A}^{\tilde{s}} = \mathbf{0}$, which are linear in $\mathbf{A}^{\tilde{s}}$ given the probability estimates from the E-step and the quantities $\mathbf{m}^{\tilde{s}c} = \sum_{\mathbf{y}} p(\tilde{s}, c|\mathbf{y})\mathbf{y}$ and $n^{\tilde{s}c} = \sum_{\mathbf{y}} p(\tilde{s}, c|\mathbf{y})$:

$$\mathbf{A}^{\tilde{s}} = \left[\sum_c \mathbf{m}^{\tilde{s}c}\mathbf{b}^{c^{\mathrm{T}}}\right]\left[\sum_c n^{\tilde{s}c}\mathbf{b}^c\mathbf{b}^{c^{\mathrm{T}}}\right]^{-1}. \tag{19}$$

The EM algorithm is guaranteed to converge to a local maximum of L^* and this typically takes around 20 iterations for our problems. After each E-step, test vectors in new styles can be

[7] That is, if the style identity s^* of $\mathbf{y}$ is labeled but the content class is unknown, we let $p(\tilde{s}, c) = 1/C$ if $\tilde{s} = s^*$ and 0 if $\tilde{s} \neq s^*$. If both the style and content identities of $\mathbf{y}$ are unlabeled, we let $p(\tilde{s}, c) = 1/(\tilde{S}C$ for all $\tilde{s}$ and c.

classified by selecting the content class c that maximizes $p(c|\mathbf{y}) = \sum_{\tilde{s}} p(\tilde{s}, c|\mathbf{y})$. Classification performance is determined by the percentage of test data for which the probability of content class c, as given by EM, is greatest for the actual content class. Note that because content classification is determined as a "by-product" of the E-step, the standard practice of running EM until convergence to a local maximum in likelihood will not necessarily lead to optimal classification performance. In fact, we have often observed "overfitting" behavior, in which optimal classification is obtained after only two or three iterations of EM, but the likelihood continues to increase in subsequent iterations as observations are assigned to incorrect content classes.

This classification algorithm thus has three free parameters for which good values must somehow be determined. In addition to the model dimensionality J mentioned above, these include the model variance σ^2 and the maximum number of iterations for which EM is run, t_{max}. In general, we set these parameters using a leave-one-style-out cross-validation procedure with the training data. That is, given S complete training styles, we train separate bilinear models on each of the S subsets of $S-1$ styles and evaluate these models' classification performance on the one style that each was not trained on. For each of these S training set splits, we try a range of values for the parameters J, σ^2, and t_{max}. Those values which yield the best average performance over the S training set splits are then used in fitting a new bilinear model to the full training data and generalizing to the designated test set.

Initialization is an important factor in determining the success of the EM algorithm. As we are primarily interested in good classification, we initialize EM in the E-step, using the results of a simple 1-nearest neighbor classifier to set the content-class assignments $p(c|\mathbf{x})$. That is, we initially assign each test observation to the content class of the most similar training observation (for which the content labels are known).

4.3 Results

We conducted four different experiments, three using the benchmark speech data to investigate different aspects of the algorithm's behavior and one using the face data to provide evidence from a separate domain. In each case, we report results with all three free parameters set using the cross-validation procedure described above, as well as for two conditions in which EM was run until convergence (i.e. $t_{max} = \infty$): J, σ^2 determined by cross-validation, and J, σ^2 set to their optimal values (as an indicator of best in-principle performance for the maximum likelihood solution).

4.3.1 Train 8, test 7 on speech data – speakers labeled

The first experiment with the speech data was the standard benchmark task described in [Robinson, 1989]. Robinson compared many learning algorithms trained to categorize vowels from the first 8 speakers (4 male and 4 female) and tested on samples from the remaining 7 speakers (4 male and 3 female). The variety and the small number of styles make this a difficult task. Table 1 shows the best results we know of for standard approaches that do not adapt to new speakers. Of the many techniques that Robinson [Robinson, 1989] tested, 1-nearest neighbor (1-NN) performs the best with 56.3% correct; chance is approximately 9% correct. Hastie & Tibshirani's [Hastie and Tibshirani, 1996] discriminant adaptive nearest neighbor (DANN) classifier slightly outperforms 1-NN, obtaining 59.7% correct with its generic parameter settings and 61.7% correct for optimal parameter settings.

After fitting an asymmetric bilinear model to the training data, we tested classification performance using our separable mixture model (SMM) on the 7 new speakers' data. We first assumed style labels were available for the test data (indicating only a change of speaker, but no information about the new speaker's style). Running EM until convergence ($t_{max} = \infty$), our best results of 77.3% correct were obtained with $J = 4$ and $\sigma^2 = 1/16$. Almost comparable results of 75.8% correct were obtained using cross-validation to set all parameters (J, σ^2, t_{max}) automatically (see Table 1). The SMM clearly outperforms the many nonadaptive techniques tested, by exploiting extra information available in the speaker labels which nonadaptive techniques make no use of.

4.3.2 Train 8, test 7 on speech data – speakers not labeled

We next repeated the benchmark experiment without the assumption that any style labels were available during testing. Thus our SMM algorithm and the nonadaptive techniques had exactly the same information available for each test observation (although the SMM had style labels available during training). We used EM to figure out both the speaker assignments as well as the vowel class assignments for the test data. EM requires that the number of style components S in the mixture model be set in advance; we choose $S = 7$ (the actual number of distinct speakers) for consistency with the previous experiment. In initializing EM in the E-step, we assigned each test observation equally to each new style component, plus or minus ten percent random noise to break symmetry and allow each style component to adapt to a distinct subset of the new data. Using cross-validation to set J, σ^2 and t_{max} automatically, we obtained $69.8\% \pm .3\%$ correct. Table 1 presents results for other parameter settings. These scores reflect average performance over ten different random initializations. Not surprisingly, SMM performance here was worse than in the previous section, where the correct style labels were assumed to be known. Nonetheless, the SMM still provided a significant improvement over the best nonadaptive approaches tested.

4.3.3 Train 14, test 1 on speech data

We noticed that the performance of the SMM on the speech data varied widely across different test speakers, and also depended significantly on the particular speakers chosen for training and testing. In some cases, the SMM did not perform much better than 1-NN, and in other cases the SMM actually did worse. Thus, we decided to conduct a more systematic study of the effects of individual speaker style on generalization. Specifically, we tested the SMM's ability to classify the speech of each of the 15 speakers in the database individually, when the other 14 speakers were used for training. Because only one speaker was presented during testing, we used only a single style model in EM and thus there was no distinction between the "speakers labeled" and "speakers not labeled" conditions investigated in the previous two sections.

Averaged over all 15 possible test speakers, 1-NN obtained 63.9% ± 3.4% correct. Using cross-validation to set J, σ^2 and t_{max} automatically, our SMM obtained 74.3% ± 4.2% correct. Running EM until convergence ($t_{max} = \infty$), we obtained 75.6%±4.0% using the best parameter settings of $J = 3$, $\sigma^2 = 1/16$ and 73.5% ± 4.4% correct using cross-validation to select J and σ^2 automatically. These SMM scores are not significantly different from each other, but are all significantly higher than 1-NN as measured by paired t-tests ($p < .01$ in all cases). The results suggest that the superior performance of SMM over nonadaptive classifiers such as nearest neighbor will hold in general over a range of different test styles.

4.3.4 Train 10, test 1 on face data

To provide further support for the generality of SMM over nonadaptive approaches, we replicated the previous experiment using the face database instead of the speech data. While the two databases are of roughly comparable size, the nature of the observations are quite different: 704-dimensional vectors of pixel values vs. 10-dimensional vectors of vocal tract log area coefficients. Specifically, we tested the SMM's ability to classify head pose for each of the 11 faces in the database individually, when the other 10 people's faces were used for training. There were 15 different possible poses, with one image of each face in each pose.

Averaged over all 11 possible test faces, 1-NN obtained 53.9% ± 4.3% correct. Using cross-validation to set J, σ^2 and t_{max} automatically, our SMM obtained 73.9% ± 6.7% correct. Running EM until convergence ($t_{max} = \infty$), we obtained 80.6% ± 7.5% using the best parameter settings of $J = 6$, $\sigma^2 = 10^5$ and 75.8% ± 6.4% correct using cross-validation to select J and σ^2 automatically. As on the speech data, these SMM scores are not significantly different from each other, but do represent significant improvements over 1-NN as measured by paired t-tests ($p < .05$ in all cases).

4.4 Discussion

Our approach to style-adaptive content classification involves two significant modeling choices: first, the use of a bilinear model of the mean observations, and second, the use of a gaussian mixture model – centered on the predictions of the bilinear model – for observations whose content and/or style assignments are unknown. The mixture model provides a principled probabilistic framework, allowing us to use the EM algorithm to solve the chicken-and-egg problem of simultaneously estimating style paramters for the new data and labeling the data according to content class (and possibly style as well). The bilinear structure of the model allows the M-step to be computed in closed form, by solving systems of linear equations. In this sense, bilinear models represent the content of observations independent of their style in a form that can be generalized easily to model data in new styles. To summarize our results in this section, we found that separating style and content with bilinear models improves content classification in new styles substantially over the best nonadaptive approaches to classification, even when no style information is available during testing, and dramatically so when style demarkation is available. Although our SMM classifier has several free parameters which must be chosen a priori, we showed that near optimal values can be determined automatically, using a cross-validation training procedure. We obtained good results on two very different data sets, low-dimensional speech data and high-dimensional face image data, suggesting that our approach may be widely applicable to many two-factor classification tasks that can be thought of in terms of recognizing invariant content elements under variable style.

5 Extrapolation

The ability to draw analogies across observations in different contexts is a hallmark of human perception and cognition [Hofstadter, 1995, Holyoak and Barnden, 1994]. In particular, the ability to *produce* analogous content in a novel style – and not just *recognize* it as in the previous section – has been taken as a severe test of perceptual abstraction [Hofstadter, 1995, Grebert et al., 1992]. The domain of typography provides a natural place to explore these issues of analogy and production. Indeed, Hofstadter has argued that the question of "What is the letter 'a'?" may be "the central problem of AI" ([Hofstadter, 1985], p. 633). Following Hofstadter, we study the task of extrapolating a novel font from an incomplete set of letter observations in that font to the remaining unobserved letters. We first describe the task specifics and our shape representation for letter observations. We then present our algorithm for stylistic extrapolation based on bilinear models and show results on extrapolating a natural font.

5.1 Task specifics

Given a training set of $C = 62$ characters (content) in $S = 5$ standard fonts (style), the task is to generate characters that are stylistically consistent with letters in a novel sixth font. The initial data were obtained by digitizing the uppercase letters, lowercase letters, and digits 0-9 of the six fonts at 38×38 pixels using Adobe Photoshop. Successful shape modeling often depends on having an image representation that makes explicit the appropriate structure which is only implicit in raw pixel values. Specifically, the need to represent shapes of different topologies in comparable forms motivates using a particle-based representation [Szeliski and Tonnesen, 1992]. We also want the letters in our representation to behave like a linear vector space, where linear combinations of letters also look like letters. [Beymer and Poggio, 1996] advocate a dense warp map for related problems. Combining these two ideas, we chose to represent each letter shape by a $2 \times 38 \times 38 = 2888$-dimensional vector of (horizontal and vertical) displacements that a set of $38 \times 38 = 1444$ ink particles must undergo to form the target shape from a reference grid.

With identical particles, there are many possible such warp maps. To ensure that similarly shaped letters are represented by similar warps, we use a physical model. We give each particle of the reference shape (taken to be the full rectangular bitmap) unit positive charge, and each pixel of the target letter negative charge proportional to its grey level intensity. The total charge of the target letter is set equal to the total charge of the reference shape. We track the electrostatic force lines from each particle of the reference shape to where they intersect the plane of the target letter, positioned opposite to the reference shape. The force lines land in a uniform density over the target letter, resulting in a smooth, dense warp map from each pixel of the reference shape to the letter. The electrostatic forces are easily calculated from Coulomb's law. We call this a "Coulomb warp" representation. To render a warp map representation of a shape, we first translate each particle of the reference shape by its warp map value, using a grid at four times the linear pixel resolution. We then blur and sub-sample to the original font resolution. By allowing non-integer charge values and sub-pixel translations, we can preserve font anti-aliasing information.

Figure 7 shows three pairs of shapes of different topologies, and the average of each pair in a pixel representation and in a Coulomb warp representation. Averaging the shapes in a pixel representation simply yields a "double-exposure" of the two images; averaging in a Coulomb warp representation results in a shape intermediate to the two being averaged.

5.2 Algorithm

During training, we fit the asymmetric bilinear model (Equation 10) to five full fonts using the closed-form SVD procedure described in Section 3.1. This yields a $K \times J$ matrix $\mathbf{A}^s$ representing each font s and a J-dimensional vector $\mathbf{b}^c$ representing each letter class c, with the observation dimensionality $K = 2888$ (see above). Adapting the model to an incomplete new style $\tilde{s}$ can be carried out in closed form, using the content vectors $\mathbf{b}^c$ learned during training. Suppose we observe M samples of style $\tilde{s}$, in content classes $C = \{c_1, \ldots, c_M\}$. We find the style matrix $\mathbf{A}^{\tilde{s}}$ that minimizes the total squared error over the test data,

$$E^* = \sum_{c \in C} ||\mathbf{y}^{\tilde{s}c} - \mathbf{A}^{\tilde{s}}\mathbf{b}^c||^2. \tag{20}$$

The minimum of E^* is found by solving the linear system $\partial E^* / \partial \mathbf{A}^{\tilde{s}} = \mathbf{0}$. Missing observations in the test style $\tilde{s}$ and known content class c can then be synthesized from $\mathbf{y}^{\tilde{s}c} = \mathbf{A}^{\tilde{s}}\mathbf{b}^c$.

In order to allow the model sufficient expressive range to produce natural-looking letter shapes, we set the model dimensionality J as high as possible. However, such a flexible model led to overfitting on the available letters of the test font and consequently poor synthesis of the missing letters in that font. To regularize the style fit to the test data and thereby avoid overfitting, we add a prior term to the squared-error cost of Equation 20 which encourages $A^{\tilde{s}}$ to be close to the linear combination of training style parameters $A^1, \ldots, A^S$, which best fits the test font. Specifically, let A^{OLC} be the value of $A^{\tilde{s}}$ which minimizes Equation 20 subject to the constraint that A^{OLC} is a linear combination of the training style parameters A^s, i.e., $A^{OLC} = \sum_{s=1}^{S} \alpha_s A^s$ for some values of α_s. ("OLC" stands for "optimal linear combination".) Without loss of generality, we can think of the α_s coefficients as the best fitting style parameters of a *symmetric* bilinear model with dimensionality I equal to the number of styles S. We then define E^* to include this new cost,

$$E^* = \sum_{c \in C} ||\mathbf{y}^{\tilde{s}c} - \mathbf{A}^{\tilde{s}}\mathbf{b}^c||^2 + \lambda ||\mathbf{A}^{\tilde{s}} - \mathbf{A}^{OLC}||^2, \tag{21}$$

and again minimize E^* by solving the linear system $\partial E^* / \partial \mathbf{A}^{\tilde{s}} = \mathbf{0}$. The tradeoff between these two costs is determined by the free parameter λ, which we set by eye to yield results with the best appearance. For this example we used a model dimensionality of 60 and $\lambda = 2 \times 10^4$.

5.3 Results

Figure 8 shows the results of extrapolating the unseen letters "A"-"I" of a new font, Monaco, using the asymmetric model with a symmetric model (i.e. OLC) prior as described above. All characters in the Monaco font except the upper case letters "A" through "I" were used to estimate its style parameters (via Equation 20). In contrast to the objective performance scores on the classification tasks reported in the previous section, here the evaluation of our results is necessarily subjective. Given examples of "A" through "I" in only the five training fonts, the model nonetheless has succeeded in rendering these letters in the test font, with approximately correct shapes for each letter class and with the distinctive stylistic features of Monaco: strict upright posture and uniformly thin strokes. Note that each of these stylistic features appears separately in one or more of the training fonts, but they do not appear *together* in any one training font.

5.4 Discussion

We have shown that it is possible to learn the style of a font from observations and extrapolate that style to unseen letters, using a hybrid of asymmetric and symmetric bilinear models. Note that the asymmetric model uses 173280 parameters (the 2888×60 matrix $A^{\tilde{s}}$) to describe the test style, while the optimal linear combination style model A^{OLC} uses only 5 (i.e. the number of training styles) parameters in the α_i. Results using only the high-dimensional asymmetric model without the low-dimensional OLC prior are far too unconstrained and fail to look like recognizable letters (Figure 9, second column). Results using only the low-dimensional prior without the high-dimensional asymmetric model are clearly recognizable as the correct letters, but fail to capture the distinctive style of Monaco (Figure 9, third column). The combination of these two terms, with a flexible high-dimensional model constrained to lie near the subspace of known style parameters, is capable of successful stylistic extrapolation on this example (Figure 8 and Figure 9, fourth column). It is an interesting question why this hybrid modeling strategy was necessary here, but not in the classification tasks investigated above. We think this is due at least in part to a general feature of extrapolation tasks which makes them objectively harder than classification tasks. Successful classification requires only that the outputs of the bilinear model using the correct content classes be closer in mean squared error to the test data than are the model outputs using incorrect content classes. Extrapolation tasks require that the model outputs be close to the true data not only in mean squared error, but also in the metric of visual appearance, which is far more subtle than mean squared error [Teo and Heeger, 1994].

Using an appropriate representation, such as our Coulomb warp, was also important in obtaining visually satisfying results. Applying the same modeling methodology to a pixel space representation of letters resulted in significantly less appealing output (Figure 9, first column). Previous models of extrapolation and abstraction in typography have been restricted to artificial grid-based fonts, for which the grid elements provide a reasonable distributed representation [Hofstadter, 1995, Grebert et al., 1992], or even simpler "grandmother-cell" representations of each letter [Polk and Farah, 1997]. In contrast, our shape representation allowed us to work directly with natural fonts.

Although the choice of model and representation turned out to be essential in this example, our results were obtained without any detailed knowledge or processing specific to the domain of typography. Hofstadter [Hofstadter, 1995] has been critical of approaches to stylistic extrapolation which minimize the role of domain-specific knowledge and processing, in particular the connectionist model of [Grebert et al., 1992], arguing that models which "don't know anything about what they are doing" (p. 408) cannot hope to capture the subtleties and richness of a human font designer's productions. We agree with Hofstatder's general diagnosis. There are many aspects of typographical competence, and we model only a subset of those. In particular, we have not tried to model the higher-level creative processes of an expert font designer, who draws on an elaborate knowledge base, reflects on the results of his work, and engages in multiple revisions of each synthesized character. However, we do think that our approach captures two essential aspects of human competence in font extrapolation: (1) our representations of letter and font characteristics are modular and independent of each other; (2) our knowledge of letters and fonts is abstracted from the ability to perform the particular behavior of character synthesis. Generic connectionist approaches to font extrapolation such as [Grebert et al., 1992] do not satisfy these constraints. Letter and font information is mixed together inextricably in the extrapolation network's weights, and an entirely different network would be needed to perform recognition or classification tasks with the same stimuli. Our bilinear modeling approach, in contrast, captures the perceptual modularity of style and content in terms of the mathematical property of separability that characterizes Equations 1 - 7. Knowledge of style s, in Equation 6 for example, is localized to the matrix parameter $\mathbf{A}^s$ while knowledge of content class c is localized to the vector parameter $\mathbf{b}^c$, and both can be freely combined with other content or style parameters respectively. Moreover, during training, our models acquire knowledge about the interaction of style and content factors that is truly abstracted from any particular behavior, and thus can support not only extrapolation of a novel style, but also a range of other synthesis and recognition tasks as shown in Figure 1.

6 Translation

Many important perceptual tasks require the perceiver to recover simultaneously two unknown pieces of information from a single stimulus in which these variables are confounded. A canonical example is the problem of separating the intrinsic shape and texture characteristics of a face from the extrinsic lighting conditions, which are confounded in any one image of that face. In this section, we show how a perceptual system may, using a bilinear model, learn to solve this problem from a training set of faces labeled according to identity and lighting condition. The bilinear model allows a novel face viewed under a novel illuminant to be "translated" to its appearance under known lighting conditions, and the known faces to be translated to the new lighting condition. Such translation tasks are the most difficult kind of two-factor learning task, because they require generalzation across both factors at once. That is, what is common across both training and test data sets is not any particular style nor any particular content class, but only the manner in which these two factors interact. Thus only a symmetric bilinear model (Equation 1-3) is appropriate, because only it represents explicitly the interaction between style and content factors, in the $\mathbf{W}_k$ parameters.

6.1 Task specifics

Given a training set of $S = 23$ faces (content) viewed under $C = 3$ different lighting conditions (style) and a novel face viewed under a novel light source, the task is to translate the new face to known lighting conditions, and the known faces to the new lighting condition. The face images, provided by Y. Moses of the Weizmann Institute, were cropped to remove non-facial features and blurred and subsampled to 48×80 pixels. Because these faces were aligned and lacked sharp edge features (unlike the typed characters of the previous section), we could represent the images directly as 3840-dimensional vectors of pixel brightness values.

6.2 Algorithm

We first fit the symmetric model (Equation 2) to the training data using the iterated SVD procedure described in Section 3.2. This yields vector representations $\mathbf{a}^s$ and $\mathbf{b}^c$ of each face c and illuminant s, and a matrix of interaction parameters $\mathbf{W}$ (defined in Equation 15). The dimensionalities for $\mathbf{a}^s$ and $\mathbf{b}^c$ were set equal to S and C respectively, allowing the bilinear model maximum expressivity while still ensuring a unique solution.

For generalization from a single test image $\tilde{\mathbf{y}}$, we adapt the model simultaneously to both the new face identity $\tilde{c}$ and the new illuminant $\tilde{s}$, while holding fixed the face-illuminant interaction term $\mathbf{W}$ learned during training. Specifically, we first make an initial guess for the new face identity vector $\mathbf{b}^{\tilde{c}}$ (e.g. the mean of the training set style vectors) and solve for the least-squares optimal estimate of the illuminant vector $\mathbf{a}^{\tilde{s}}$:

$$\mathbf{a}^{\tilde{s}} = \left[\left[\mathbf{W}\mathbf{b}^{\tilde{c}}\right]^{\mathrm{VT}}\right]^{-1}\tilde{\mathbf{y}}. \tag{22}$$

Here $[\ldots]^{-1}$ denotes the pseudoinverse. Given this new value for $\mathbf{a}^{\tilde{s}}$, we then re-estimate $\mathbf{b}^{\tilde{c}}$ from

$$\mathbf{b}^{\tilde{c}} = \left[\left[\mathbf{W}^{\mathrm{VT}}\mathbf{a}^{\tilde{s}}\right]^{\mathrm{VT}}\right]^{-1}\tilde{\mathbf{y}}, \tag{23}$$

and iterate Equations 22-23 until both $\mathbf{a}^{\tilde{s}}$ and $\mathbf{b}^{\tilde{c}}$ converge. We can then generate images of the new face under known illuminant s (from $y_k^{s\tilde{c}} = \mathbf{a}^{s^{\mathrm{T}}}\mathbf{W}_k\mathbf{b}^{\tilde{c}}$), and of known face c under the new illuminant (from $y_k^{\tilde{s}c} = \mathbf{a}^{\tilde{s}^{\mathrm{T}}}\mathbf{W}_k\mathbf{b}^{c}$).

6.3 Results

Figure 10 shows results, with both the old faces translated to the new illuminant and the new face translated to the old illuminants. For comparison, the true images are shown next to the synthetic ones. Again, evaluation of these results must necessarily be subjective. The lighting and shadows for each synthesized image appear approximately correct, as do the facial features of the old faces translated to the new illuminant. The facial features of the new face translated to the old illuminants appear slightly blurred, but otherwise resemble the new face more than any of the old faces. One reason the synthesized images of the new face are not as sharp as the synthesized images of the old faces is that the latter are produced by averaging images of a single face under several lighting conditions – across which all the facial features are precisely aligned – while the former are produced by averaging images of many faces under a single lighting condition – across which the facial features vary significantly in their positions.

6.4 Discussion

A history of applying linear models to face images motivates our bilinear modeling approach. The original work on eigenfaces [Kirby and Sirovich, 1990, Turk and Pentland, 1991] established that images of many different faces taken under identical lighting and viewpoint conditions occupy a low-dimensional linear subspace of pixel space. Subsequent work [Hallinan, 1994] showed that images of a single face taken under many different lighting conditions also occupy a low-dimensional linear subspace. Thus the factors of facial identity and illumination have already been shown to satisfy approximately the definition of bilinearity – the effects of one factor are linear when the other is held constant – so it is natural to integrate them into a bilinear model.

While the general problem of separating shape, texture, and illumination features in an image is underdetermined [Barrow and Tenenbaum, 1978], here the bilinear model learned during training provides sufficient constraint to approximately recover both face and illumination parameters from a single novel image. [Atick et al., 1996] proposed a related approach to learning shape-from-shading for face images, based on a linear model of head shape in three dimensions and a physical model of the image formation process; in contrast, our bilinear model is completely two-dimensional (i.e. image-based) and requires no prior knowledge about the physics of image formation. Of course, we have not solved the general shape-from-shading recovery problem for arbitrary objects under arbitrary illumination. Our solution (as well as that of [Atick et al., 1996]) depends critically on the assumption that the new image, like the images in the training set, depicts an upright face under reasonable lighting conditions. In fact, there is evidence that the brain does not solve the shape-from-shading problem in its most general form, but rather has learned (or evolved) solutions to important special cases such as face images [Cavanagh, 1991]. So-called “Mooney faces” – brightness-thresholded face images in which shading is the only cue to shape – can be easily recognized as images of three-dimensional surfaces when viewed in upright position, but cannot be discriminated from two-dimensional ink blotches when viewed upside-down so that the shading conventions are atypical [Shepard, 1990], or when the underlying 3D structure has been distorted away from a globally face-like shape [Moore and Cavanagh, 1998]. More generally, the ability to learn constrained solutions to *a priori* underconstrained inference problems may turn out to be essential for perception [Poggio and Hurlbert, 1994, Nayar and Poggio, 1996]. Bilinear models offer one simple and general framework for how biological and aritifical perceptual systems may learn to solve a wide range of such tasks.

7 Directions for future work

The most obvious extension of our work is to observations and tasks with more than two underlying factors, via multilinear models [Magnus and Neudecker, 1988]. For example, a symmetric trilinear model in three factors q, r, and s would take the form:

$$y_l^{qrs} = \sum_{i,j,k} w_{ijkl} a_i^q b_j^r c_k^s. \tag{24}$$

The procedures for fitting these models are direct generalizations of the learning algorithms for two-factor models that we describe in this paper. As in the two-factor case, we iteratively apply linear matrix techniques to solve for the parameters of each factor given parameter estimates for the other factors, until all parameter estimates converge [Magnus and Neudecker, 1988].

As with any learning framework, the success of bilinear models depends on having a suitable input representation. Further research is needed to determine what kinds of representations will endow specific kinds of observations with the most nearly bilinear structure. In particular, the font extrapolation task might benefit from a representation that is better tailored to the important features of letter shapes. Also, it would be of interest to develop general procedures for incorporating available domain-specific knowledge into bilinear models, e.g. via a priori constraints on the model parameters [Simard et al., 1993].

Finally, we would like to explore the relevance of bilinear models for research in neuroscience and psychophysics. The essential computational procedures for learning in bilinear models, SVD and EM, can be implemented naturally in neural networks, using Hebb-like learning rules [Sanger, 1994] and soft competitive mechanisms [Nowlan and Senjowski, 1995] respectively. What would a biologically plausible instantiation of our bilinear models look like? The essential representational feature of bilinear models is their multiplicative factor interactions. At the physiological level, multiplicative neuronal interactions [Andersen et al., 1985, Olshausen et al., 1993, Riesenhuber and Dayan, 1997], arising from nonlinear synaptic [Koch, 1997] or population-level [Salinas and Abbott, 1993] mechanisms, have been proposed for visual computations that require the synergistic combination of two inputs, such as modulating spatial attention [Andersen et al., 1985, Olshausen et al., 1993, Riesenhuber and Dayan, 1997, Salinas and Abbott, 1993] or estimating motion [Koch, 1997]. May these same kinds of circuits be co-opted to solve some of the tasks we study here? At the level of psychophysics, many studies have demonstrated that the ability of the human visual and auditory systems to factor out contextually irrelevant dimensions of variation, such as lighting conditions or speaker accent, is neither perfect nor instantaneous. Observations in unusual styles are generally more difficult or more time-consuming to process. Do bilinear models have difficulty on the same kinds of stimuli that people do? Do the dynamics of adaptation in bilinear models (e.g. EM for classification, or the iterative SVD-based procedure for translation) take longer to converge on stimuli that people are slower to process? These are just a few of the empirical questions motivated by a bilinear modeling approach to studying perceptual inference.

8 Conclusions

In one sense, bilinear models are not new to perceptual research. It has previously been shown that several core vision problems, such as the recovery of structure from motion under orthographic projection [Tomasi and Kanade, 1992] or color constancy under multiple illuminants [Brainard and Wandell, 1991, Marimont and Wandell, 1992, D'Zmura, 1992], are solvable efficiently because they are fundamentally bilinear at the level of geometry or physics

[Koenderink and Doorn, 1997]. These results are important but of limited usefulness, because most two-factor problems in perception do not have this true bilinear character. More commonly, perceptual inferences based on accurate physical models of factor interactions are either very complex (as in speech recognition), underdetermined (as in shape-from-shading), or simply inappropriate (as in typography).

Here we have proposed that perceptual systems may often solve such challenging two-factor tasks without detailed domain knowledge, using bilinear models to learn approximate solutions rather than to describe explicitly the intrinsic geometry or physics of the problem. We presented a suite of simple and efficient learning algorithms for bilinear models, based on the familiar techniques of SVD and EM. We then demonstrated the scope of this approach with applications to three different two-factor tasks – classsification, extrapolation, and translation – using three different kinds of signals – speech, typography, and face images. With their combination of broad applicability and ready learning algorithms, bilinear models may prove to be a generally useful component in the toolkits of engineers and brains alike.

Acknowledgements

We thank E. Adelson, B. Anderson, M. Bernstein, D. Brainard, P. Dayan, W. Richards, and Y. Weiss for helpful discussions. Joshua Tenenbaum was a Howard Hughes Medical Institute Predoctoral Fellow. Preliminary reports of this material appeared in [Tenenbaum and Freeman, 1997, Freeman and Tenenbaum, 1997].

References

[Andersen et al., 1985] Andersen, R., Essick, G., and Siegel, R. (1985). The encoding of spatial location by posterior parietal neurons. *Science*, 230:456–458.

[Atick et al., 1996] Atick, J. J., Griffin, P. A., and Redlich, A. N. (1996). Statistical approach to shape from shading: Reconstruction of 3d face surfaces from single 2d images. *Neural Computation*, 8:1321–1340.

[Barrow and Tenenbaum, 1978] Barrow, H. G. and Tenenbaum, J. M. (1978). Recovering intrinsic scene characteristics from images. In Hanson, A. R. and Riseman, E. M., editors, *Computer Vision Systems*, pages 3–26. Academic Press, New York.

[Bell and Sejnowski, 1995] Bell, A. and Sejnowski, T. (1995). An information–maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159.

[Bergem et al., 1988] Bergem, D. R. V., Pols, L. C., and Beinum, F. J. K.-V. (1988). Perceptual normalization of the vowels of a man and a child in various contexts. *Speech Communication*, 7(1):1–20.

[Beymer and Poggio, 1996] Beymer, D. and Poggio, T. (1996). Image representations for visual learning. *Science*, 272:1905–1909.

[Bishop, 1995] Bishop, C. M. (1995). *Neural networks for pattern recognition.* Oxford.

[Brainard and Wandell, 1991] Brainard, D. and Wandell, B. (1991). A bilinear model of the illuminant's effect on color appearance. In Landy, M. S. and Movshon, J. A., editors, *Computational Models of Visual Processing*, chapter 13. MIT Press, Cambridge, MA.

[Caruana, 1998] Caruana, R. (1998). Multitask learning. In Thrun, S. and Pratt, L., editors, *Learning to Learn*, pages 95–134. Kluwer, Norwell, MA.

[Cavanagh, 1991] Cavanagh, P. (1991). What's up in top-down processing? In Gorea, A., editor, *Representations of Vision: Trends and Tacit Assumptions in Vision Research*, pages 295–304. Cambridge University Press, Cambridge, UK.

[Dayan et al., 1995] Dayan, P., Hinton, G., Neal, R., and Zemel, R. (1995). The helmholtz machine. *Neural Computation*, 7(5):889–904.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:1–38.

[Duda and Hart, 1973] Duda, R. O. and Hart, P. E. (1973). *Pattern classification and scene analysis*. Wiley-Interscience.

[D'Zmura, 1992] D'Zmura, M. (1992). Color constancy: surface color from changing illumination. *Journal of the Optical Society of America A*, 9:490–493.

[Freeman and Tenenbaum, 1997] Freeman, W. T. and Tenenbaum, J. B. (1997). Learning bilinear models for two-factor problems in vision. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 554–560, San Juan, PR.

[Ghahramani, 1995] Ghahramani, Z. (1995). Factorial learning and the EM algorithm. In Tesauro, G., Touretzky, D., and Leen, T., editors, *Adv. in Neural Information Processing Systems*, volume 7, pages 617–624, Cambridge, MA. MIT Press.

[Grebert et al., 1992] Grebert, I., Stork, D. G., Keesing, R., and Mims, S. (1992). Connectionist generalization for production: An example from gridfont. *Neural Networks*, 5:699–710.

[Hallinan, 1994] Hallinan, P. W. (1994). A low-dimensional representation of human faces for arbitrary lighting conditions. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 995–999.

[Hastie and Tibshirani, 1996] Hastie, T. and Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification. *IEEE Pattern Analysis and Machine Intelligence*, (18):607–616.

[Hinton et al., 1995] Hinton, G., Dayan, P., Frey, B., and Neal, R. (1995). The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161.

[Hinton and Ghahramani, 1997] Hinton, G. and Ghahramani, Z. (1997). Generative models for discovering sparse distributed representations. *Phil. Trans. Royal Soc. B*, 352:1177–1190.

[Hinton and Zemel, 1994] Hinton, G. E. and Zemel, R. (1994). Autoencoders, minimum description length, and Helmholtz free energy. In Cowan, J., Tesauro, G., and Alspector, J., editors, *Adv. in Neural Information Processing Systems*, volume 6, pages 3–10, San Mateo, CA. Kauffman.

[Hofstadter, 1985] Hofstadter, D. (1985). *Metamagical Themas.* Basic Books, New York.

[Hofstadter, 1995] Hofstadter, D. (1995). *Fluid Concepts and Creative Analogies.* Basic Books.

[Holyoak and Barnden, 1994] Holyoak, K. and Barnden, J., editors (1994). *Advances in Connectionist and Neural Computation Theory.* Ablex, Norwood, NJ.

[Kirby and Sirovich, 1990] Kirby, M. and Sirovich, L. (1990). Application of the karhunen-loeve procedure for the characterization of human faces. *PAMI*, 12(1):103–108.

[Koch, 1997] Koch, C. (1997). Computation and the single neuron. *Nature*, 385:207–211.

[Koenderink and Doorn, 1997] Koenderink, J. J. and Doorn, A. J. V. (1997). The generic bilinear calibration–estimation problem. *International Journal of Computer Vision*, 23(3):217–234.

[Magnus and Neudecker, 1988] Magnus, J. R. and Neudecker, H. (1988). *Matrix differential calculus with applications in statistics and econometrics.* Wiley.

[Mardia et al., 1979] Mardia, K., Kent, J., and Bibby, J. (1979). *Multivariate Analysis*. Academic Press, London.

[Marimont and Wandell, 1992] Marimont, D. H. and Wandell, B. A. (1992). Linear models of surface and illuminant spectra. *Journal of the Optical Society of America A*, 9(11):1905–1913.

[Moghaddam and Pentland, 1997] Moghaddam, B. and Pentland, A. P. (1997). Probabilistic visual learning for object representation. *IEEE Pattern Analysis and Machine Intelligence*, 19(7):696–710.

[Moore and Cavanagh, 1998] Moore, C. and Cavanagh, P. (1998). Recovery of 3d volume from 2-tone images of novel objects. *Cognition*, 67(1,2):45–71.

[Nayar and Poggio, 1996] Nayar, S. and Poggio, T., editors (1996). *Early Visual Learning*. Oxford University Press, New York.

[Nowlan and Senjowski, 1995] Nowlan, S. and Senjowski, T. J. (1995). A selection model for motion processing in area mt of primates. *J. Neuroscience*, 15:1195–1214.

[Nygaard and Pisoni, 1998] Nygaard, L. C. and Pisoni, D. B. (1998). Talker-specific learning in speech perception. *Perception & Psychophysics*, 60(3):355–376.

[Olshausen et al., 1993] Olshausen, B., Anderson, C., and Essen, D. V. (1993). A neural model of visual attention and invarient pattern recognition. *J. Neuroscience*, 13:4700–4719.

[Omohundro, 1995] Omohundro, S. M. (1995). Family discovery. In *Adv. in Neural Information Processing Systems*, volume 8, pages 402–408.

[Poggio and Hurlbert, 1994] Poggio, T. and Hurlbert, A. (1994). Observations on cortical mechanisms for object recognition and learning. In Koch, C. and Davis, J., editors, *Large Scale Neuronal Theories of the Brain*, pages 152–182. MIT Press, Cambridge, MA.

[Polk and Farah, 1997] Polk, T. A. and Farah, M. J. (1997). A simple common contexts explanation for the development of abstract letter identities. *Neural Computation*, 9(6):1277–1289.

[Press et al., 1992] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C.* Cambridge Univ. Press.

[Riesenhuber and Dayan, 1997] Riesenhuber, M. and Dayan, P. (1997). Neural models for part-whole hierarchies. In Mozer, M., Jordan, M., and Petsche, T., editors, *Adv. in Neural Information Processing Systems*, volume 9, pages 17–23, Cambridge, MA. MIT Press.

[Robinson, 1989] Robinson, A. (1989). *Dynamic error propagation networks.* PhD thesis, Cambridge University Engineering Dept.

[Salinas and Abbott, 1993] Salinas, E. and Abbott, L. (1993). A model of multiplicative neural responses in parietal cortex. *Proc. Nat. Acad. Sci. USA*, pages 11956–11961.

[Sanger, 1994] Sanger, T. (1994). Two algorithms for iterative computation of the singular value decomposition from input/output samples. In Cowan, J., Tesauro, G., and Alspector, J., editors, *Adv. in Neural Information Processing Systems*, volume 6, pages 144–151, San Mateo, CA. Kauffman.

[Sanocki, 1992] Sanocki, T. (1992). Effects of font- and letter-specific experience on the perceptual processing of letters. *American Journal of Psychology*, 105(3):435–458.

[Shepard, 1990] Shepard, R. N. (1990). *Mind Sights.* Freeman, New York.

[Simard et al., 1993] Simard, P. Y., LeCun, Y., and Denker, J. (1993). Efficient pattern recognition using a new transformation distance. In Hanson, S., Cowan, J., and Giles, L., editors, *Adv. in Neural Information Processing Systems*, volume 5. Morgan Kaufman.

[Szeliski and Tonnesen, 1992] Szeliski, R. and Tonnesen, D. (1992). Surface modeling with oriented particle systems. In *Proc. SIGGRAPH 92*, volume 26, pages 185–194. In *Computer Graphics*, Annual Conference Series.

[Tenenbaum and Freeman, 1997] Tenenbaum, J. B. and Freeman, W. T. (1997). Separating style and content. In Mozer, M., Jordan, M., and Petsche, T., editors, *Adv. in Neural Information Processing Systems*, volume 9, pages 662–668. MIT Press.

[Teo and Heeger, 1994] Teo, P. and Heeger, D. (1994). Perceptual image distortion. In *First IEEE International Conference on Image Processing*, volume 2, pages 982–986. IEEE.

[Thrun and Pratt, 1998] Thrun, S. and Pratt, L., editors (1998). *Learning to Learn.* Kluwer, Norwell, MA.

[Tomasi and Kanade, 1992] Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *Intl. J. Comp. Vis.*, 9(2):137–154.

[Turk and Pentland, 1991] Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *J. of Cognitive Neuroscience*, 3(1).

Classifier	Percent correct on test data
Multi-layer perceptron (MLP)	51%
Radial basis function network (RBF)	53%
1-Nearest neighbor (1-NN)	56%
Discriminant adaptive nearest neighbor (DANN):	
generic parameter settings	59.7%
optimal parameter settings	61.7%
Separable mixture models (SMM) – test speakers labeled:	
all parameters set by CV ($J = 3, \sigma^2 = 1/64, t_{max} = 2$)	75.8%
$t_{max} = \infty$; J, σ^2 set by CV ($J = 3, \sigma^2 = 1/32$)	68.2%
$t_{max} = \infty$; optimal J, σ^2 ($J = 4, \sigma^2 = 1/16$)	77.3%
Separable mixture models (SMM) – test speakers *not* labeled:	
all parameters set by CV ($J = 3, \sigma^2 = 1/64, t_{max} = 2$)	69.8% ± .3%
$t_{max} = \infty$; J, σ^2 set by CV ($J = 3, \sigma^2 = 1/32$)	59.9% ± 1.1%
$t_{max} = \infty$; optimal J, σ^2 ($J = 3, \sigma^2 = 1/64$)	63.2% ± 1.5%

Table 1: Accuracy of classifying spoken vowels from a benchmark multi-speaker database. MLP, RBF, and 1-NN results were obtained by [Robinson, 1989]. Hastie and Tibshirani's DANN classifier [Hastie and Tibshirani, 1996] achieves the best performance we know of for an approach that does not adapt to new speakers. The SMM classifiers perform significantly better vowel classification by simultaneously modeling speaker style. "CV" denotes the cross-validation procedure for parameter setting described in the text.

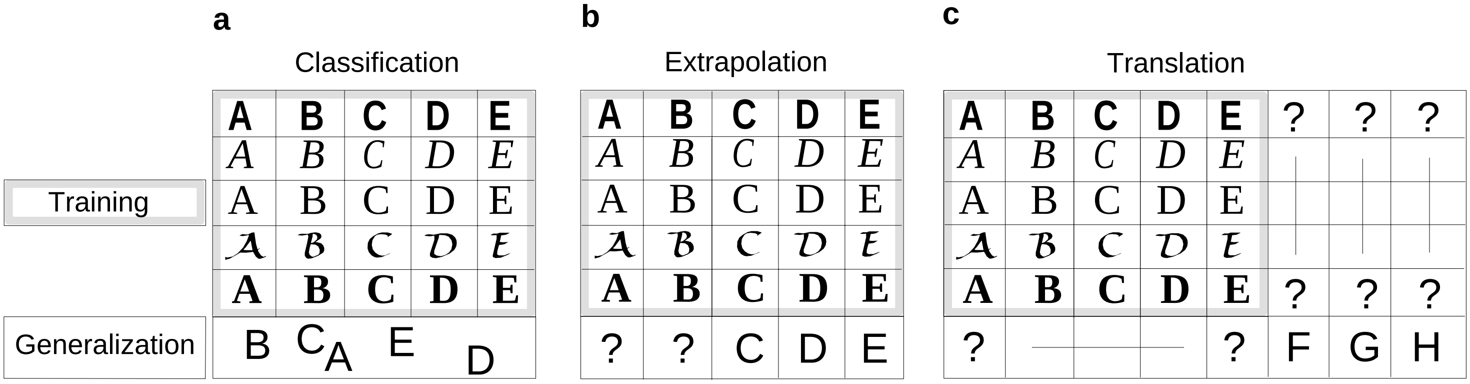

Figure 1: Given a labeled training set of observations in multiple styles (e.g. fonts) and content classes (e.g. letters), we want to (**a**) *classify* content observed in a new style, (**b**) *extrapolate* a new style to unobserved content classes, and (**c**) *translate* from new content observed only in new styles into known styles or content classes.

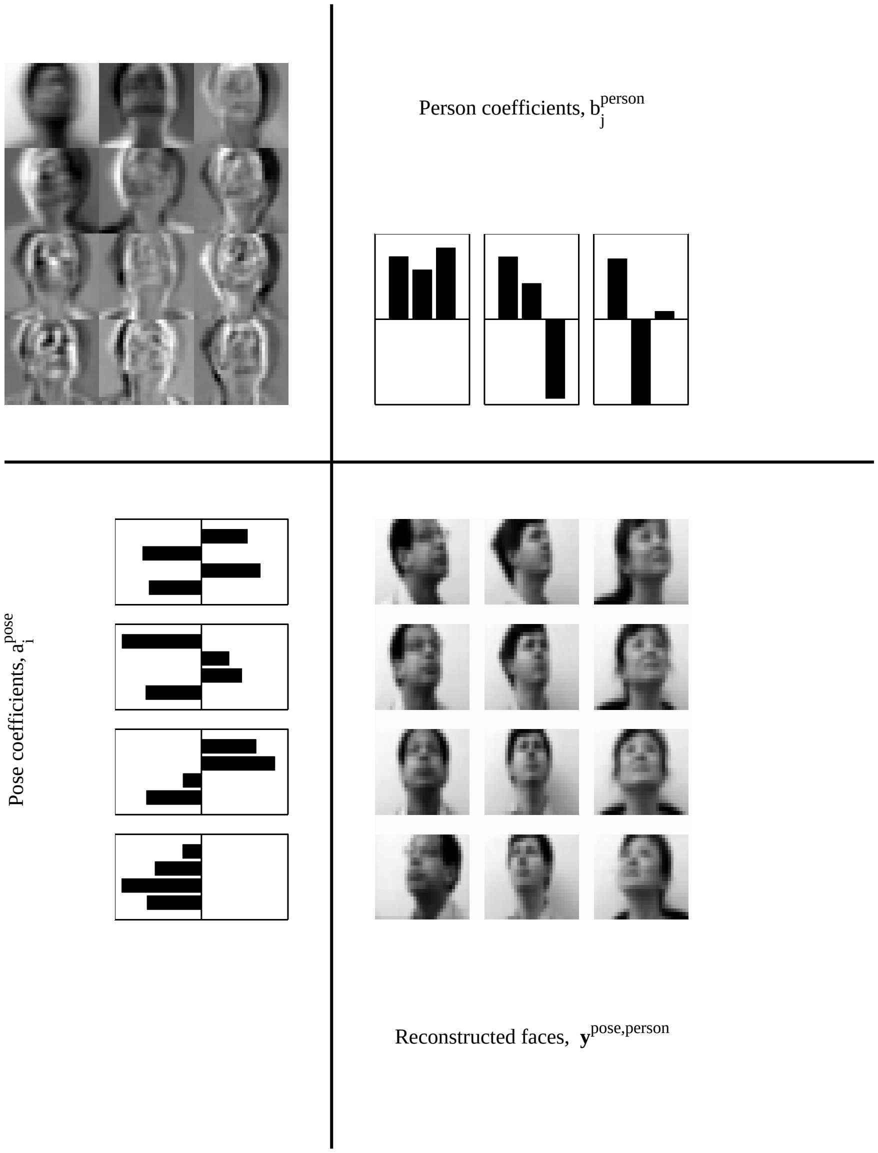
Person coefficients, b_j^{person}
Pose coefficients, a_i^{pose}
Reconstructed faces, $\mathbf{y}^{pose,person}$

Figure 2: Illustration of symmetric bilinear model for a small set of faces (a subset of Figure 6). The two factors for this example are identity and pose. A vector of coefficients, a_i^{pose}, describes the pose, and a second vector, b_j^{person}, describes the person. To render a particular person under a particular pose, the vectors a_i^{pose} and b_j^{person} multiply along the 4 rows and 3 columns of the array of basis images w_{ijk}. The weighted sum of basis functions yields the reconstructed faces $y_k^{pose,\ person}$.

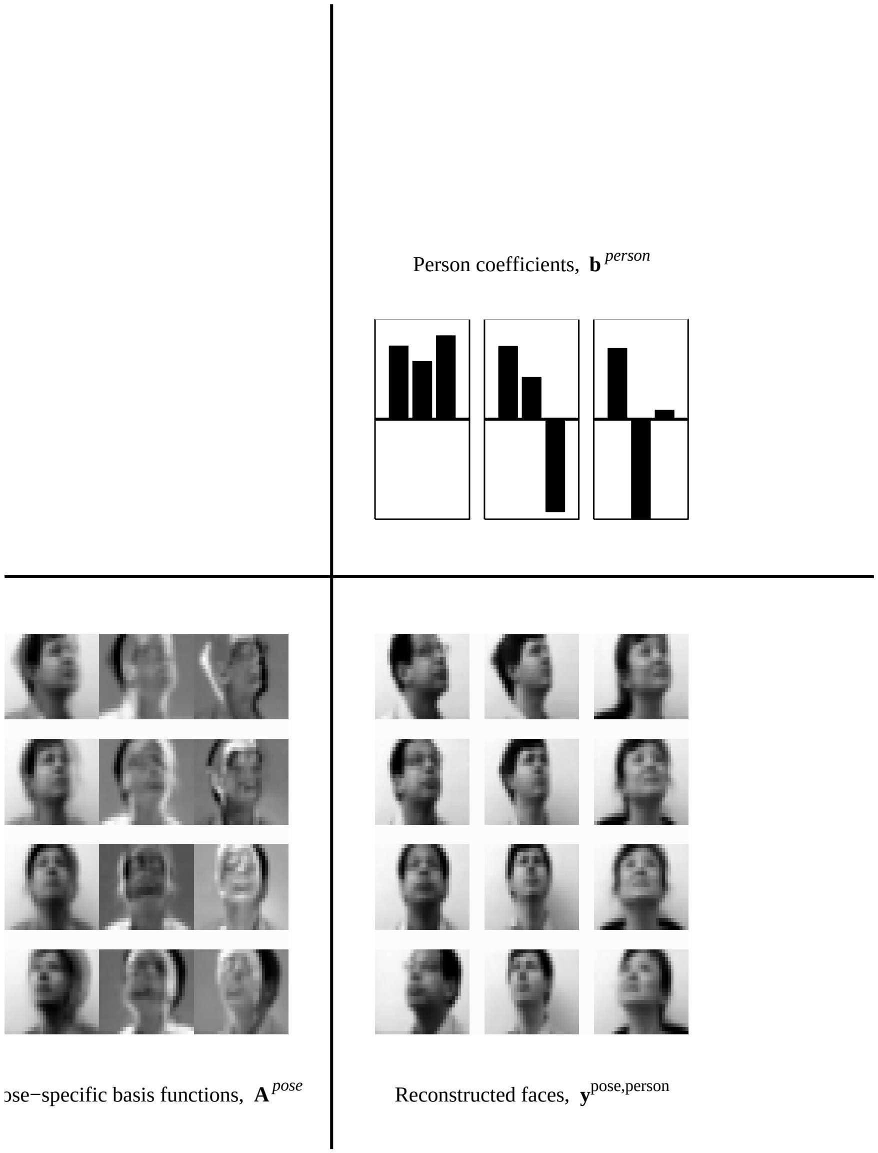
Person coefficients, $\mathbf{b}^{person}$
ose−specific basis functions, $\mathbf{A}^{pose}$
Reconstructed faces, $\mathbf{y}^{\mathrm{pose,person}}$

Figure 3: The images of Figure 2, represented by an asymmetric bilinear model, with head pose as the style factor. Person-specific basis vectors, $\mathbf{b}^{person}$, multiply pose-specific basis images, $\mathbf{A}^{pose}$; the sum reconstructs a given person in a given pose. The basis images are similar to an eigenface representation within a given pose [Moghaddam and Pentland, 1997], except that in this model the different basis images are constrained to allow one set of person coefficients to reconstruct the same face across different poses.

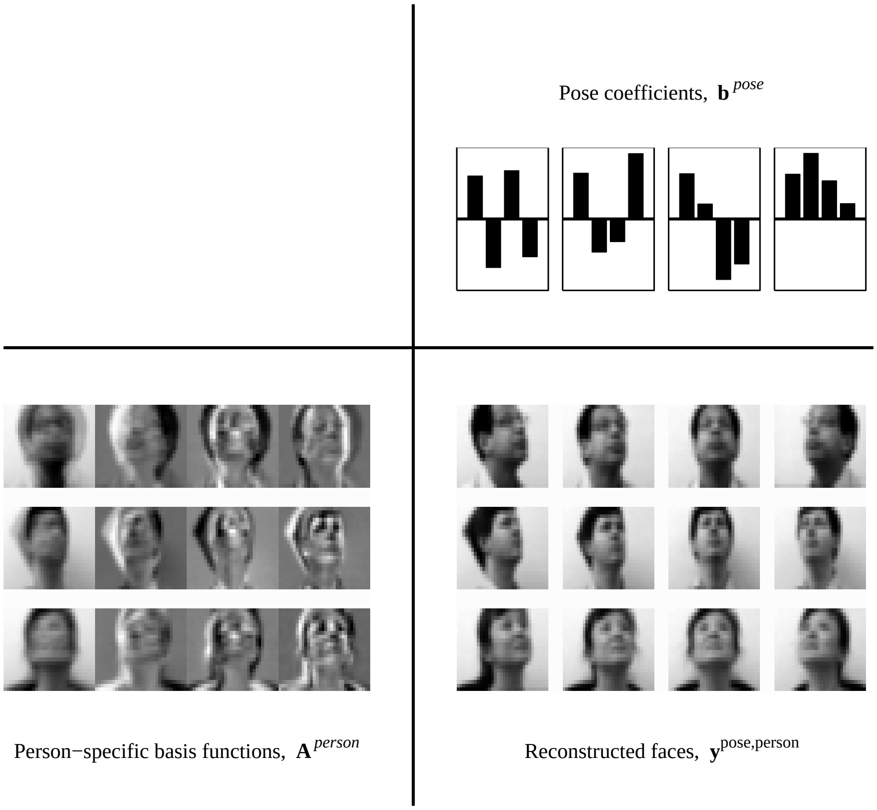
Pose coefficients, $\mathbf{b}^{pose}$
Person−specific basis functions, $\mathbf{A}^{person}$
Reconstructed faces, $\mathbf{y}^{\text{pose,person}}$

Figure 4: Asymmetric bilinear model applied to the data of Figs. 2 and 3, treating identity is the "style" factor. The basis functions are person-specific basis functions, $\mathbf{A}^{person}$, and the content vectors are pose-specific coefficients, $\mathbf{b}^{pose}$. Each image of the person-specific basis functions plays the same role in rotating head position, independent of the face being rotated.

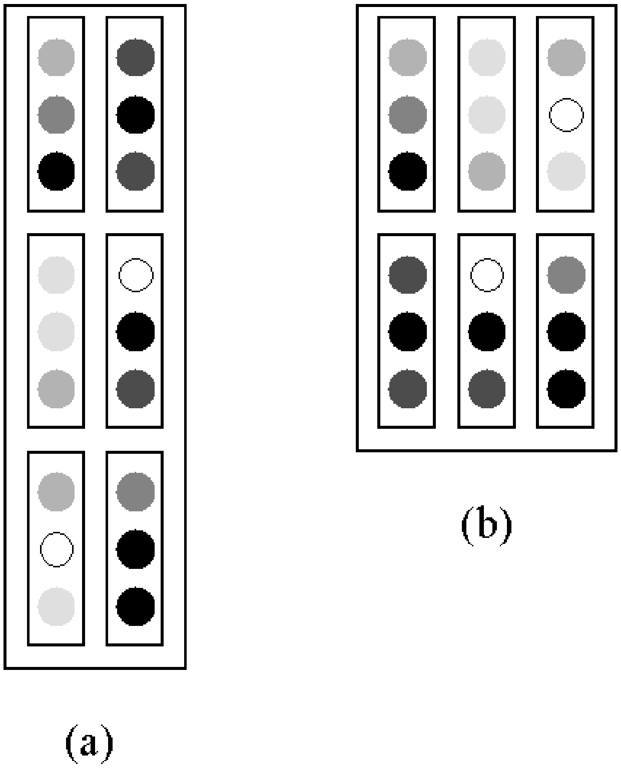

Figure 5: Schematic illustration of the vector tranpose (following [Marimont and Wandell, 1992]). A matrix is considered to be an array of stacked vectors, as in (a). The vector transpose, (b), is then the matrix of stacked vectors with their positions in the matrix transposed.

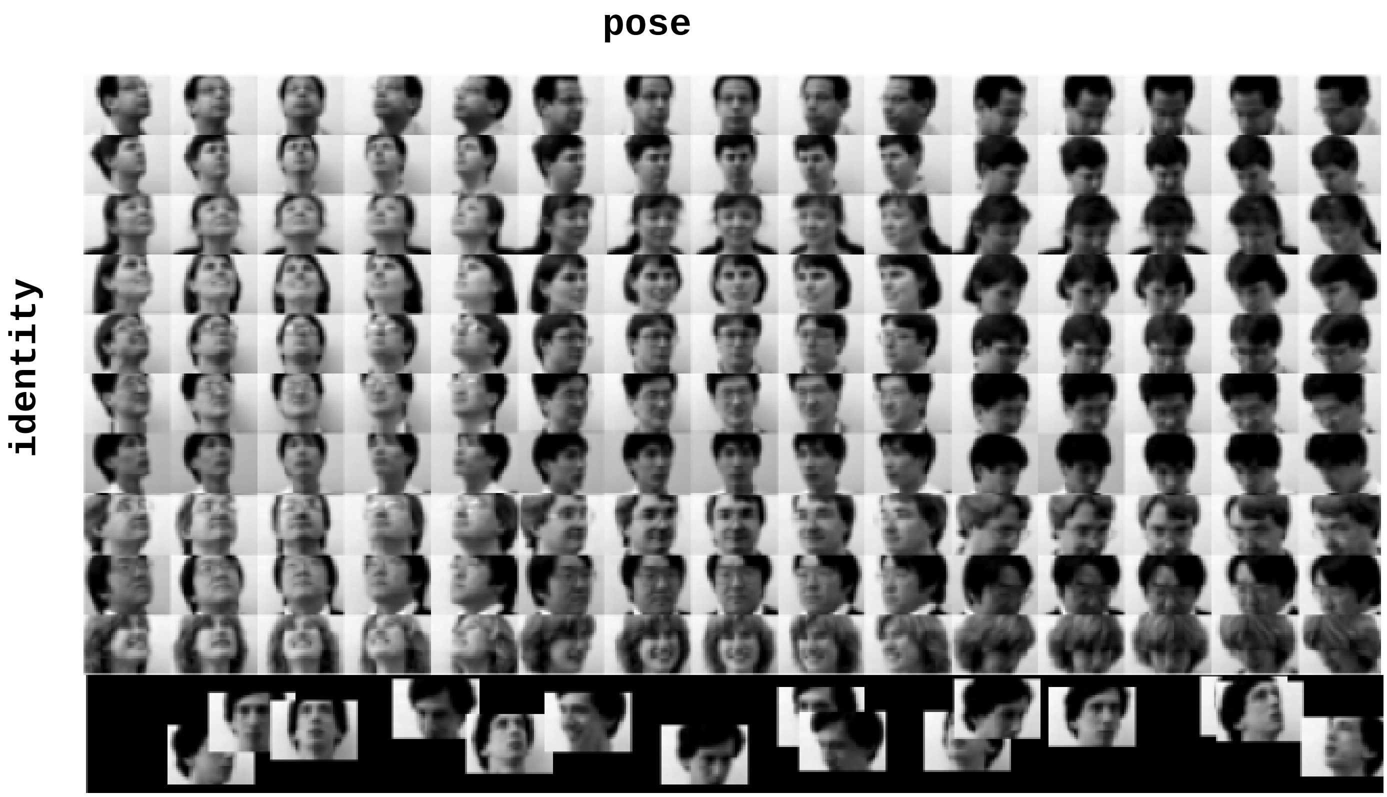

Figure 6: Classification of familiar content in a new style, illustrated on a data set of face images. During training, we observe the faces of different people (style) in different poses (content), resulting in the matrix of faces shown. During generalization, we observe a new person's face in each of these familiar poses, and we seek to classify the pose of each new image. The separable mixture model allows us to build up a model for the new face at the same time as we classify the new poses, thereby improving classification performance.

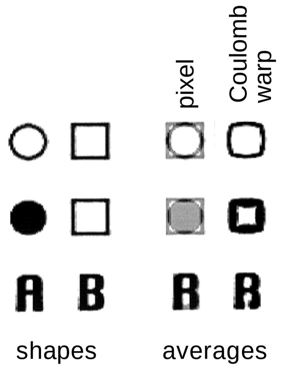

Figure 7: The result of averaging shapes together under different representations. Averaging in a pixel representation always gives a “double exposure” of the two shapes. Under the Coulomb warp representation, a circle averaged with a square gives a rounded square; a filled circle averaged with a square gives a very thick, rounded square. The letter A averaged with the letter B yields a shape that is arguably intermediate to the shapes of the two letters. This property of the representation makes it well suited to linear algebraic manipulations with bilinear models.

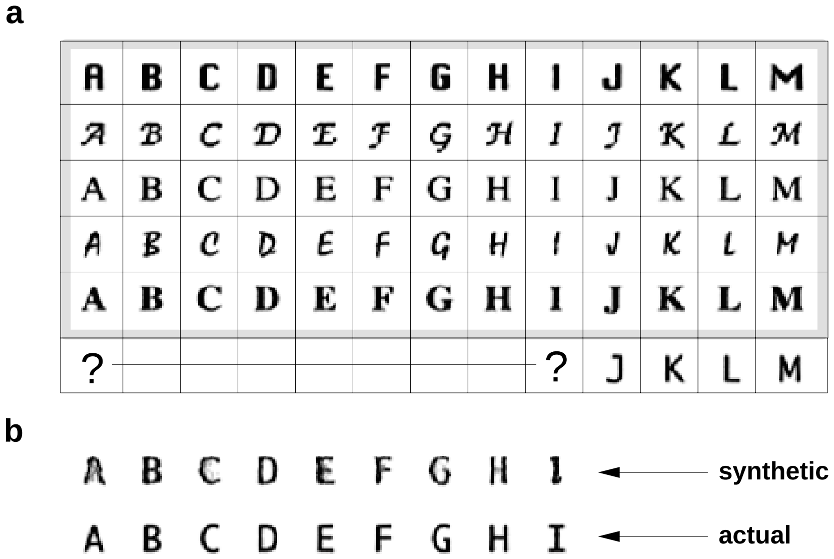

Figure 8: Style extrapolation in typography. (**a**) Rows 1-5: The first 13 (of 62) letters of the training fonts. Row 6: The novel test font, with A-I unseen by the model. (**b**) Row 1: Font extrapolation results. Row 2: The actual unseen letters of the test font.

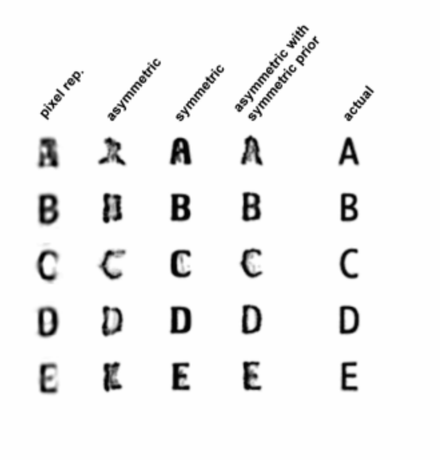

Figure 9: Result of different methods applied to font extrapolation problem (Figure 8), where unseen letters in a new font are synthesized. The asymmetric bilinear model has too many parameters to fit, and generalization to new letters is poor (second column). The symmetric bilinear model has only 5 degrees of freedom for our data, and fails to represent the characteristics of the new font (third column). We used the symmetric model result as a prior to constrain the flexibility of the asymmetric model, yielding the result shown here (fourth column) and in (Figure 8). All these methods used the Coulomb warp representation for shape. Performing the same calculations in a pixel representation requires blurring the letters so that linear combinations can modify shape, and yields barely passable results (first column). The far right column shows the actual letters of the new font.

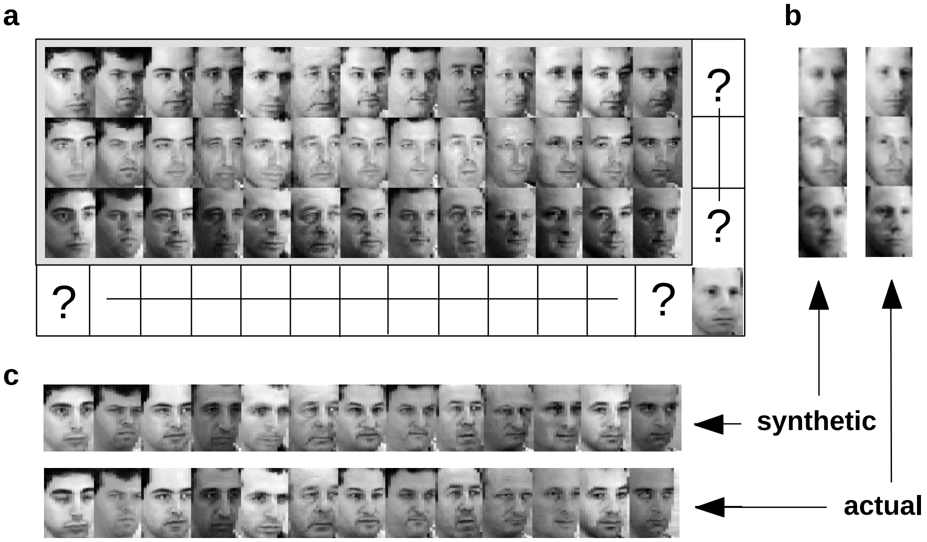

Figure 10: Translation across style and content in shape-from-shading. (**a**) Row 1-3: The first 13 (of 24) faces viewed under the three illuminants used for training. Row 4: The single test image of a new face viewed under a new light source. (**b**) Column 1: Translation of the new face to known illuminants. Column 2: The actual (unseen) images. (**c**) Row 1: Translation of known faces to the new illuminant. Row 2: The actual (unseen) images.