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

Learning Bilinear Models for Two-Factor Problems in Vision

W. T. Freeman, J. B. Tenenbaum

TR96-37 December 1996

Abstract

In many vision problems, we want to infer two (or more) hidden factors which interact to produce our observations. We may want to disentangle illuminant and object colors in color constancy; rendering conditions from surface shape in shape-from-shading; face identity and head pose in face recognition; or font and letter class in character recognition. We refer to these two factors generically as s̈tyleänd c̈ontent.̈ This paper received Outstanding Paper prize, CVPR ́97.

Proc. IEEE Computer Vision and Pattern Recognition (CVPR ́97), Puerto Rico

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

Learning bilinear models for two-factor problems in vision

W. T. Freeman and J. B. Tenenbaum

TR-96-37 May 1999

Abstract

In many vision problems, we want to infer two (or more) hidden factors which interact to produce our observations. We may want to disentangle illuminant and object colors in color constancy; rendering conditions from surface shape in shape-from-shading; face identity and head pose in face recognition; or font and letter class in character recognition. We refer to these two factors generically as "style" and "content". Bilinear models offer a powerful framework for extracting the two-factor structure of a set of observations, and are familiar in computational vision from several well-known lines of research. This paper shows how bilinear models can be used to learn the style-content structure of a pattern analysis or synthesis problem, which can then be generalized to solve related tasks using different styles and/or content. We focus on three kinds of tasks: extrapolating the style of data to unseen content classes, classifying data with known content under a novel style, and translating two sets of data, generated in different styles and with distinct content, into each other's styles. We show examples from color constancy, face pose estimation, shape-from-shading, typography and speech.

In Proceedings IEEE Computer Vision and Pattern Recognition, 1997, Puerto Rico.

1. First printing, TR96-37, March, 1997

Learning bilinear models for two-factor problems in vision

W. T. Freeman
MERL, a Mitsubishi Electric Research Lab
201 Broadway
Cambridge, MA 02139
freeman@merl.com

J. B. Tenenbaum
MIT Dept. of Brain and Cognitive Sciences
E10-210
Cambridge, MA 02139
jbt@psyche.mit.edu

MERL Technical Report 96-37
MERL, a Mitsubishi Electric Research Lab
201 Broadway
Cambridge, MA 02139
November, 1996

Abstract

In many vision problems, we want to infer two (or more) hidden factors which interact to produce our observations. We may want to disentangle illuminant and object colors in color constancy; rendering conditions from surface shape in shape-from-shading; face identity and head pose in face recognition; or font and letter class in character recognition. We refer to these two factors generically as "style" and "content".

Bilinear models offer a powerful framework for extracting the two-factor structure of a set of observations, and are familiar in computational vision from several well-known lines of research. This paper shows how bilinear models can be used to learn the style-content structure of a pattern analysis or synthesis problem, which can then be generalized to solve related tasks using different styles and/or content. We focus on three kinds of tasks: extrapolating the style of data to unseen content classes, classifying data with known content under a novel style, and translating two sets of data, generated in different styles and with distinct content, into each other's styles. We show examples from color constancy, face pose estimation, shape-from-shading, typography and speech.

1 Introduction

A set of observations is often influenced by two or more independent factors. For example, in typography, character and font combine to yield the rendered letter, Fig. 1. We may think of one factor as "content" (the character) and the other as "style" (the font).

Many estimation problems fit this form (see also [10, 15]). In speech recognition, the speaker's accent modulates words to produce sounds. In face recognition, a person's image is modulated by both their identity and by the orientation of their head. In shape-from-shading, both the shape of the object and the lighting conditions influence the image. In color perception, the unknown illumination color can be thought of as a style which modulates the unknown object surface reflectances to produce the observed eye cone responses. We will generically refer to the two factors as "style" and "content".

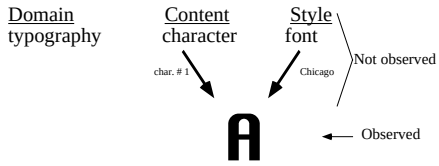

Figure 1: Example problem with two-factor structure. The observed letter is a function of two unseen factors, the character (content) and the font (style).

In these problems, and others, we want to make inferences about the factors underlying the observations. We want to perceive the true shapes, colors, or faces, independent of the rendering conditions, or want to recognize the speaker's words independent of the accent.

This style/content factorization is an essential problem in vision, and many researchers have addressed related issues (e.g., [10, 22, 4, 13]). A key feature of the style-content factorization, which has not been addressed in these previous papers, is that it is well-suited to learning the structure of analysis or synthesis problems, as we describe below. We demonstrate competency in a vision task by being able to predict how observations would change were the style or content class changed, or in classifying by content observations in a new style. We explore the learning issues, and emphasize the problems that a learning approach lets you solve.

Learning the model parameters is analogous to learning from observations over the course of one's visual experience. We see how content-class data change when observed under different styles. We describe in the next section standard techniques for fitting our model parameters to obesrvation data in the complete matrix form.

We use a bi-linear model which explicitly represents the problem's factorial structure, and fit the model parameters to the observations. We then use the fitted models to solve a particular task. We identify and solve 3 example tasks for two-factor problems: extrapolation, classification, and translation.

In the next section, we describe those canonical tasks. Following that, we present our models, show how to learn the model parameters, and present solved examples for each task.

2 Tasks: extrapolation, classification, and translation

The first of our three problem tasks is extrapolation. Given some examples of previously observed content-classes in a new style, extrapolate to synthesize other content-classes in that new style. Referring to Fig. 2, this involves analyzing the style common to the letters of the bottom row, finding what the letters of a column have in common, and synthesizing that letter in that font. We apply our bilinear model to this example in Section 5.1.

The second task is classification. We observe examples of a new style, but with no content-class labels, Fig. 3. An example of this is listening to speech by a speaker with an unusual accent. We need to develop a model for how observations change within a style (accent), and how they change across styles for a given content-class (word). This information can improve classification performance, as we show in the examples of Section 6.

We call the third task translation, and it is the most difficult, although the most useful. We observe new content-classes in a new style, Fig. 4, and we want to translate those observations to a known style, or, holding style constant, to known content classes. To solve this problem, we have to build a model not only for style and content independent of each other, but for the problem itself, independent of style and content. We show two examples of this task in Section 7.

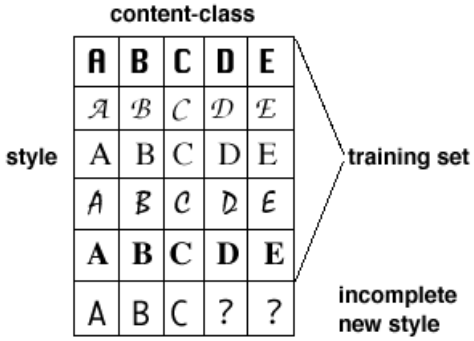

Figure 2: The extrapolation problem.

3 Bilinear models

We write an observation vector in the style s and content class c as y^{sc}, and let K be its dimension. We seek to fit these observations with some model $y^{sc} = f(a^s, b^c; W)$, where a^s and b^c are parameter vectors describing style s and content c, and W is a set of parameters for the rendering function f that determines the mapping from the style and content spaces to the observation space.

There are many reasons to choose a bilinear form for f:

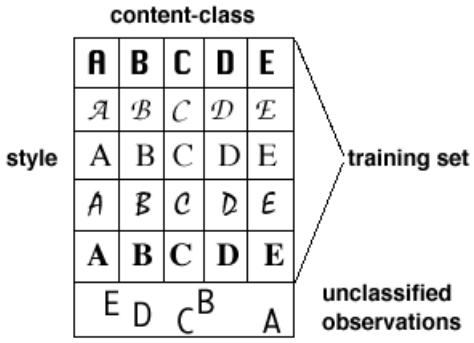

Figure 3: The classification problem.

- Linear models have been successfully applied to many vision problems [23, 7, 24, 2, 13, 14, 19, 16]. These studies suggest that many data sets produced by varying a single style or content factor (with all other factors held constant) can be represented exactly or approximately by linear models. Since a bilinear mapping from style and content to observations is linear in the style component for a fixed content class, and vice versa, bilinear models are thus naturally promising candidates for modeling data produced by the interaction of individually linear factors.

- Bilinear models inherit many convenient features of linear models: the are easy to fit (either with closed form solutions or efficient iterative solutions with guarantees of convergence [12]. They can be embedded in a probabilistic model with gaussian noise to yield tractable maximum likelihood estimation with missing information, as described in [21] and applied here in Section 4.

- The models extend easily to multiple factors, yielding multilinear models.

- Bilinear models are simple, yet seem capable of modeling real image data, so we want to take this approach as far as we can.

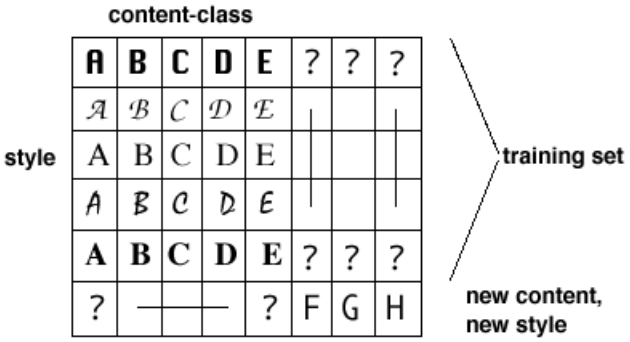

Figure 4: The translation problem.

We assume f is a *bilinear* mapping, given as follows:

$$y_k^{sc} = \sum_{ij} W_{ijk} a_i^s b_j^c \tag{1}$$

The W_{ijk} parameters represent a set of basis functions independent of style and content, which characterize the interaction between these two factors. Observations in style s and content c are generated by mixing these basis functions with coefficients given by the (tensor) product of a_i^s and b_j^c vectors. These represent (respectively) style independently of content and content independently of style. In general, the dimensionalities of the style and content vectors, denoted I and J, should be less than or equal to the numbers of styles N_s and content classes N_c respectively; the model is capable of perfectly reproducing the observations when $I = N_s$ and $J = N_c$, and finds increasingly compact (and hence more generalizable) representations of style and content as the dimensionality is decreased. We call this the *symmetric* model, because it treats the two factors symmetrically.

Note that the symmetric bilinear model can reduce the dimensionality of both style and content, so that assuming the basis functions W_{ijk} have been learned from experience, there may be fewer unknown degrees of freedom in style and content than there are constraints given by one high-dimensional image. Thus it is possible to do translation problems such as shape-from-shading and color constancy, where both the rendering conditions and the content classes of the new observation are unknown[1].

In many situations, it may be impractical or unnecessary to represent both style and content factors with low-dimensional parameter vectors. For instance, we may observe only a small sample of widely varying styles, which could in principle be expressed as linear combinations of some basis styles given a much larger sample of styles from which to learn this basis, but which cannot be so compactly represented given only the limited sample available. We can obtain a more flexible bilinear model which still preserves separable style and content representations by letting the basis functions depend on style or content, instead of being independent of both factors. For example, if the basis functions are allowed to depend on style, the bilinear model from Eq. (1) becomes $y_k^{sc} = \sum_{ij} W_{ijk}{}^s a_i^s b_j^c$, which simplifies, without loss of generality, to the following form by summing out the i index,

$$y_k^{sc} = \sum_j A_{jk}^s b_j^c. \tag{2}$$

Style s is now completely represented by the $J \times K$ matrix A_{jk}^s. Alternatively, if the basis functions depend on content, then we have content class c represented as an $I \times K$ matrix B_{ik}^c, according to

$$y_k^{sc} = \sum_i B_{ik}^c a_i^s. \tag{3}$$

Often, there is a natural intuitive interpretation of these bilinear models, which we call *asymmetric*. For example, in our typography example, Eq. (2) provides font-specific basis functions A_{jk}^s (look ahead to Fig. 7 for an illustration), which are combined according to letter-specific coefficients b_j^c. In other situations, such as speech recognition, it may be natural to see the style of an observation as a modulation of its underlying content, which is represented as a linear transformation of the content vector in Eq. (2). The disadvantages of the asymmetric models over the symmetric model of Eq. (1) are: (1) there is no explicit parameterization of the rendering function f independent of style and content, so a model trained on observations from a fixed set of styles and content classes may be of no use in analyzing new data generated from novel settings of both factors; (2) the matrix style model in Eq. (2) or the matrix content model in Eq. (3) may be *too* flexible and overfit the training data. Note that of the three tasks described in Section 2, only translation requires an explicit parameterization of the abstract rendering function independent of both style and content; classification and extrapolation do not. So for these tasks, if overfitting can be controlled by some appropriate kind of regularization, an asymmetric bilinear model of style and content can be just as useful as the symmetric version, and can often be applied when much less data is available for training, and/or when one of the factors cannot be reduced to a linear combination of basis elements.

4 Fitting bilinear models

Our applications involve two phases: an initial training phase and a subsequent testing phase. In the training phase, we fit a bilinear model to a complete matrix of observations in N_s styles and N_c content classes. This gives explicit representations of style independent of content, content independent of style, and, for the symmetric bilinear model, the interaction of style and content. In the testing phase (Figs. 2, 3 and 4), the same model is fit to a new sample of observations which is assumed to have something in common with the training set, in content, in style, or at least in the interaction between content and style. During testing, the parameters (a^s, b^c, or W) corresponding to the assumed commonalities are clamped to their values learned during training. This enables better performance on a classification, extrapolation, or translation task with the test data than would have been possible without the initial training on the set of related observations.

Singular value decomposition (SVD) can be used to fit the parameters of the asymmetric model [12, 13, 10, 21]. This technique works the same regardless of whether we have one or more observations for each style-content pair, as long as we have the same number of observations for each pair.

[1] It can be shown that if a symmetric bilinear model fits the observations, then, given the appropriate W, there is a unique solution for a^s and b^c as long as $K \geq I + J$.

Let the K-dimensional column vector $\bar{y}^{sc}$ denote the mean of the observed data generated by the style-content pair $\{A^s, b^c\}$, and stack these vectors into a single $(K \times N_s) \times N_c$-dimensional measurement matrix

$$Y = \begin{bmatrix} \bar{y}^{11} & \cdots & \bar{y}^{1N_c} \\ \vdots & \ddots & \\ \bar{y}^{N_s 1} & & \bar{y}^{N_s N_c} \end{bmatrix}. \quad (4)$$

We compute the SVD of $Y = USV^T$, and define the $(K \times N_s) \times J$-dimensional matrix $\hat{A}$ to be the first J columns of U, and the $J \times N_c$-dimensional matrix $\hat{B}$ to be the first J rows of SV^T. The model dimensionality J can be chosen in various ways: by a priori considerations, by requiring a sufficiently good approximation to the data (as measured by mean squared error or some more subjective metric), or by looking for a gap in the singular value spectrum. Finally, we can identify $\hat{A}$ and $\hat{B}$ as the desired parameter estimates in stacked form,

$$A = \begin{bmatrix} A^1 \\ \vdots \\ A^{N_s} \end{bmatrix}, \quad B = \begin{bmatrix} b^1 \cdots b^{N_c} \end{bmatrix}, \quad (5)$$

where each $A^s, 1 \leq s \leq N^s$, corresponds to the style matrix $A^s_{jk}$and each $b^c, 1 \leq c \leq N^c$, corresponds to the content vector b^c_jin Eq. (2).

Note that this is formally identical to the Tomasi and Kanade's use of the SVD to solve the structure-from-motion problem under orthographic projection, but instead of camera motion and shape matrices replaced by style and content matrices respectively.

See [21] for how to fit model to more complicated data patterns (e.g. varying numbers of observations per style and content, uncertain assignment of observations to styles or content classes).

For the symmetric model, there is an iterative procedure for fitting model parameters to the data which converges to the least squares model fit [12, 13]. The algorithm iteratively applies SVD as in the asymmetric case, repeatedly re-ordering the observation matrix elements to alternate the roles of the style and content factors. We refer the reader to the pseudo-code in Marimont and Wandell [13].

5 Extrapolation

5.1 Typography

To show that the bilinear models can analyze and synthesize something like style even for a very non-linear problem, we apply these models to typography. The extrapolation task is to synthesize letters in a new font, given a set of examples of letters in that font.

The letter representation is important. We need to represent shapes of different topologies in comparable forms. This motivates using a particle model to represent shape [20]. Further, we 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 [3] advocate a dense warp map for related problems. Combining the above, we chose to represent shape by the set of displacements that a set of particles would have to undergo from a reference shape to form the target shape.

With identical particles, there are many possible such warp maps. For our models to work well, we want similar warps to represent similarly shaped letters. To accomplish this, we use a physical model (in the spirit of [17, 18], but with the goal of dense correspondence, rather than modal representation). 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 it would intersect the plane of the target letter, set to be in front of and parallel to the reference shape. The force lines therefore 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 a "Coulomb warp" representation.

Figure 5 shows two simple shapes of different topologies, and the average of the two shapes in a pixel representation and in a Coulomb warp representation. Averaging the shapes in a pixel representation yields a simple "double-exposure" of the two images; averaging in a Coulomb warp representation results in a shape that looks like a shape in between the two.

To render a warp map representation of a shape, we first translate each particle of the reference shape, using a grid at four times the linear pixel resolution. We blur that, then sub-sample to the original font resolution. By allowing non-integer charge values and sub-pixel translations, we can preserve font anti-aliasing information.

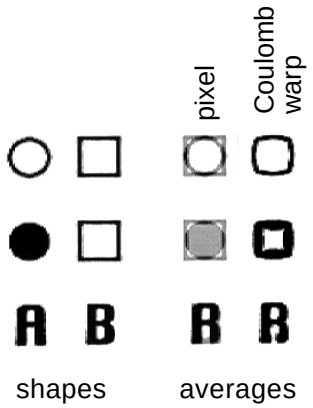

Figure 5: Behavior of the Coulomb warp representation under averaging, compared with a pixel representation.

We applied the asymmetric bilinear model, in the representation above, to the extrapolation task of Fig. 2. Our shape representation allowed us to work with familiar and natural fonts, in contrast to earlier work on extrapolation in typography [6, 8] We digitized 62 letters (uppercase letters, lowercase letters, digits 0-9) of six fonts at 38 × 38 pixels using Adobe Photoshop (uppercase letters shown in the first 5 columns and last column of Fig. 6). We trained the asymmetric model to learn separate A_i models for the first 5 fonts (Chicago, Zaph, Times, Mistral and TimesBold) and B_j models for all 62 letters. Fig. 7) shows the style-specific basis functions for each font. To generate a letter in two different fonts, we use the same linear combination of basis warps, applied to the appropriate basis for each font. Note that the character of the warp basis functions reflects the character of the font: slanted, italic, upright, etc. For the testing phase, we are presented with the letters of a 6th font, Monaco, with the first third of the upper case letters omitted. The extrapolation task is to synthesize the missing letters in the Monaco font.

For this very non-linear task, we chose a high model dimensionality of 60. To control such a high dimensional model and avoid over-fitting, we added a prior model for the font style. The target prior was the optimum style from the symmetric model: the linear combination of training styles A^i which fit the Monaco font the best, $A^s_{jk_{OLC}}$. We then used a gradient descent procedure to find the font-specific style matrix A which optimized

$$(1-\lambda)(y^{sc}_k - \sum_j A^s_{jk} b^c_j)^2 + \lambda(A^s_{jk} - A^s_{jk_{OLC}})^2. \quad (6)$$

We set $\lambda = 0.99995$, which controlled overfitting to give the best results.

The next to last column of Fig. 6 shows the results. The model has succeeded in learning the style of the new font. Each letter looks more like Monaco than any of the other fonts. Even partially succeeds in generating the extended crossbars of the "I", characteristic of the Monaco font, even though none of the example "I"s in the training set have such a stylistic feature.

6 Classification

6.1 Face Pose Estimation

We collected a face database of 11 subjects viewed in 15 different poses, Fig. 10. The poses span a grid of three vertical positions (up, level, down) and five horizontal positions (extreme-left, left, straight-ahead, right, extreme-right). The pictures were shifted to align the nose tip position, found manually. The images were blurred and cropped to 22 x 32 pixels.

We tested the model's usefulness for classification by training on the face images of 10 subjects viewed in all 15 poses, and then using the learned pose models to classify the 15 images of the remaining subject's face into the appropriate pose category. For training, we took as input the face database with each face labeled according to pose and subject identity, and with one subject removed for later testing. We then fit the asymmetric bilinear model in Eq. (3) to this training data using the SVD procedure described in Section 4, yielding models of subject c's head, B^c_{ik}, and pose s, a^s_i. The model dimensionality $I = 6$ was chosen as the minimum dimensionality giving an adequate fit to the training data.

For testing, we take as input the 15 images of the remaining subject c^* in different poses, but without pose labels. The goal is now to classify these images into the appropriate pose category, using the pose models a^s_ilearned during the training phase. Now, if we had a model $B^{c^*}_{ik}$ of the new subject's head, we could simply combine this head model with the known pose models and then assign each test image to the pose that gives it the highest likelihood, assuming a gaussian likelihood function with mean given by the model predictions from Eq. (3). But we have been given neither $B^{c^*}_{ik}$ nor the appropriate pose labels for the test subject, so we use the EM algorithm to solve this clustering problem. EM iterates between estimating the parameters of a new head model $B^{c^*}_{ik}$ (M-step), and (soft) assigning each image to the appropriate pose categories (E-step). For details of the EM algorithm applied to classification problems with a new style or content, see [21].

Classification performance is determined by the percentage of test images for which the probability of pose s, as given by EM, is greatest for the actual pose class. We repeated the experiment 11 times, leaving each of the 11 subjects out of the training set in turn. The SMM achieved an average performance of 81% correct. In fact, the performance distribution is somewhat skewed, with two or fewer errors on 8 out of 11 subjects, and six or more errors on 3 out of 11 subjects, indicating that the bilinear model learns an essentially faithful representation for most subjects' heads. For comparison, a simple 1-nearest neighbor matching algorithm yielded 53% correct, with no fewer than 5 errors on any test subject.

In this example, we found it most natural to think of subject identity as the content factor, and pose as the style factor, so this task becomes an example of style classification under varying content. Of course, exactly the same techniques can be applied to classifying content under varying styles, and [21] reports one example in the domain of speech recognition. A bilinear model was trained on data from 8 speakers (4 male, 4 female) uttering 11 different vowels, and then used to classify the utterances of 7 new speakers (4 male, 3 female) into the correct vowel category, using the same EM procedure described above. The bilinear model with EM achieved 77% correct performance, compared to 51% for a multi-layer perceptron and 56% for 1-nearest neighbor, the best of many statistical techniques tested on the same dataset.

7 Translation

7.1 Color Constancy

Translation problems occur frequently in vision. One example is the "color constancy" problem [4, 13, 10, 5]. The spectrum of light reflecting off an object is a function of both the unknown illumination spec-

trum and the object's unknown reflectance spectrum. The color of the reflected light may vary wildly under different illumination conditions, yet human observers perceive colors to be relatively constant across illuminants. They can effectively translate the observations of unknown colors, viewed under an unknown illuminant, to how they would appear under a canonical illuminant. This is the translation problem of Fig. 4. References [4, 13, 10] analyze the case of full training information, but do not address the translation problem.

Our approach is to learn a W matrix for the training set. We then hold that fixed, and fit for both the a and b parameters for the new observations. Then we can translate across style or content, as desired.

In general, the color constancy problem is underdetermined [5]. Researchers have proposed using additional constraints or visual cues, such as interreflections or specular reflections. A nice property of the bilinear models is that these or other visual constaints can be learned.

We show a synthetic example to illustrate this. Methods which exploit specularities to achieve color constancy typically require specularity levels large enough to dominate image chromaticity histograms [11, 9]. Here we show how small, random variations in specularity may also be used to achieve color constancy.

We assume that from a localized patch, we observe three samples of light reflected from the same object, each with an unknown fractional specular reflection, uniformly distributed between 0 and 10%. (In a natural image, observations could consist of image values and local filter outputs. We have not yet undertaken the calibrated studies to determine how well this technique works for natural scenes.)

Using a set of programs developed by Brainard [5], we drew 30 random samples from a 3-dimensional linear model for surface reflectances, and 9 random samples from another 3-dimensional linear model for illuminants. We rendered these, using the randomly drawn specularity values, creating the full training observation matrix corresponding. We fit this with the symmetric bilinear model, using a 3-dimensional model for illuminants and a 5-dimensional model for surface color and the random fractional specularity. This fitting stage also sets the W matrix.

We then drew 30 new color surfaces, viewed under a new random draw of illumination, with small random specularities added to each color observation. We use the W matrix trained above, and used 10000 iterations of the linear fitting to fit a and b values to the observations to describe the illuminant, surfaces, and specularities. There is an ambiguity of scale between a and b, so this approach can't fit absolute intensities, but it finds the chromaticities correctly. With the symmetric bilinear model, we are able to predict the chromaticities of the new surfaces under a canonical illuminant, with an rms error of 0.07%. For comparison, the rms difference in observation matrix chromaticities under the two illuminants is 54%.

7.2 Face and lighting translation

We worked with the Weizmann database of face images under controlled variations in illumination (provided by Yael Moses). Images of 24 male faces under four different conditions of illumination were blurred and cropped to 40 x 24 pixels, removing most nonfacial features (hair, clothing, etc.) from the images.

We tested the model's effectiveness for style and content translation by training the symmetric bilinear model on 23 of 24 faces viewed under 3 of 4 illuminants, leaving out one subject under all illuminants and one illuminant for all subjects. We then applied the learned model to render the novel 24th face under the 3 known illuminants and the 23 known faces under the new illuminant. Thus this is another example of style-content translation.

For training, we took as input the face database of 24 faces under 4 illuminants, minus one subject under all illuminants and all subjects under illuminant. We then fit the symmetric bilinear model $y^{s_i c_j} = a^{s_i} W b^{c_j}$ to this training data using the iterated SVD procedure described above, yielding models of the ith illuminant, a^{s_i}, the jth face, b^{c_j}, and the interaction between illuminant and face, W. The dimensionalities for face and illuminant models were set equal to the maximum values that still allow a unique solution (the number of different faces and illuminants respectively), in order to give the models of face space and illumination space the largest range possible.

For testing, input one image: the single unseen face under the single unseen illuminant. The goal is now to apply the illuminant-face interaction model W learned during training to recover appropriate face and illuminant models a^{s*} and b^{c*} for this new image which are commensurable with the face and illuminant models learned for the training set.

Figure 10 shows the results of rendering the novel face under the three old illuminants, and rendering the 7 of the 23 known faces under the new illuminant. The quality of the synthesized results demonstrates that the style-content interaction model W learned during the training phase is sufficient to allow good recovery of both face shape and illuminant from a single image during the test phase.

Atick [1] also showed that shape-from-shading from a single image could be solved by assuming a low-dimensional shape space. An explicit shape space is learned directly from 3-D range maps and built into a physical model of shading. In contrast, we learn an implicit shape space, as well as a space of illuminants and a model of the interaction between illuminant and shape, only from the a set of 2-D images.

8 Summary

We address learning two-factor problems in vision. We use both symmetric and asymmetric bilinear models to learn parameters describing both the style and content-classes of a set of observations. We identify and study the problems of extrapolation, classification, and translation, illustrating these with examples from color constancy, face pose estimation, shape-from-shading, typography, and speech.

References

[1] J. J. Atick, P. A. Griffin, and A. N. Redlich. Statistical approach to shape from shading: Reconstruction of 3d face surfaces from single 2d images. *Neural Computation*, 1995. in press.

[2] P. N. Belhumeur and D. J. Kriegman. What is the set of images of an object under all possible lighting conditions. In *Proc. IEEE CVPR*, pages 270–277, 1996.

[3] D. Beymer and T. Poggio. Image representations for visual learning. *Science*, 272:1905–1909, June 28 1996.

[4] M. D'Zmura. Color constancy: surface color from changing illumination. *J. Opt. Soc. Am. A*, 9:490–493, 1992.

[5] W. T. Freeman and D. H. Brainard. Bayesian decision theory, the maximum local mass estimate, and color constancy. In *Proc. 5th Intl. Conf. on Computer Vision*, pages 210–217. IEEE, 1995.

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

[7] P. W. Hallinan. A low-dimensional representation of human faces for arbitrary lighting conditions. In *Proc. IEEE CVPR*, pages 995–999, 1994.

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

[9] G. J. Klinker, S. A. Shafer, and T. Kanade. The measurement of highlights in color images. *Intl. J. Comp. Vis.*, 2(1):7–32, 1988.

[10] J. J. Koenderink and A. J. van Doorn. The generic bilinear calibration–estimation problem. *Intl. J. Comp. Vis.*, 1996. in press.

[11] H. Lee. Method for computing the scene-illuminant chromaticity from specular highlights. *J. Opt. Soc. Am. A*, 3(10):1694–1699, 1986.

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

[13] D. H. Marimont and B. A. Wandell. Linear models of surface and illuminant spectra. *J. Opt. Soc. Am. A*, 9(11):1905–1913, 1992.

[14] H. Murase and S. Nayar. Visual learning and recognition of 3-d objects from appearance. *Intl. J. Comp. Vis.*, 14:5–24, 1995.

[15] S. M. Omohundro. Family discovery. In *Advances in Neural Information Processing Systems*, 1995.

[16] A. P. Pentland. Linear shape from shading. *Intl. J. Comp. Vis.*, 1(4):153–162, 1990.

[17] S. E. Sclaroff. *Modal matching: a method for describing, comparing, and manipulating digital signals*. PhD thesis, MIT, 1995.

[18] L. Shapiro and J. Brady. Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, 10(5):283–288, June 1992.

[19] A. Shashua. *Geometry and photomnetry in 3D visual recognition*. PhD thesis, MIT, 1992.

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

[21] J. B. Tenenbaum and W. T. Freeman. Separable mixture models: Separating style and content. In *Advances in Neural Information Processing Systems*, 1996.

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

[23] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1), 1991.

[24] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Pat. Anal. Mach. Intell.*, 13(10):992–1006, 1991.

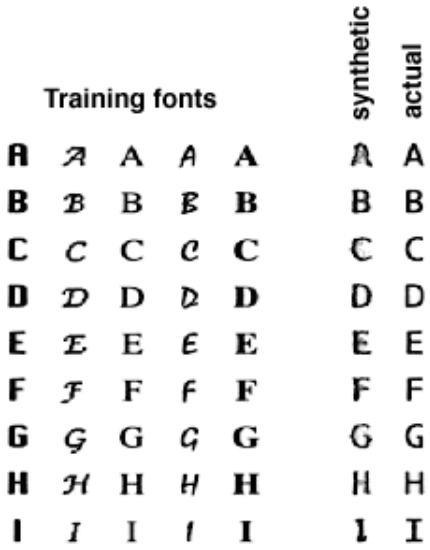

Figure 6: Example of style extrapolation in the domain of typography. Training data included upper and lower case alphabets, and digits 0-9.

Figure 7: Warp bases.

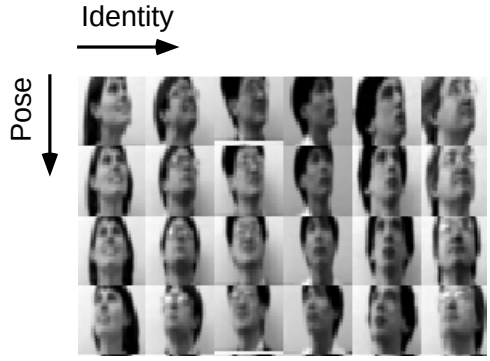

Figure 8: A subset of the face images used for pose classification.

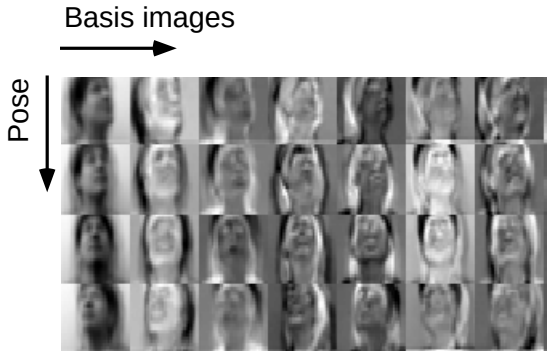

Figure 9: A subset of the pose-specific basis faces. Note that in the bilinear model, each basis face plays the same role across poses.

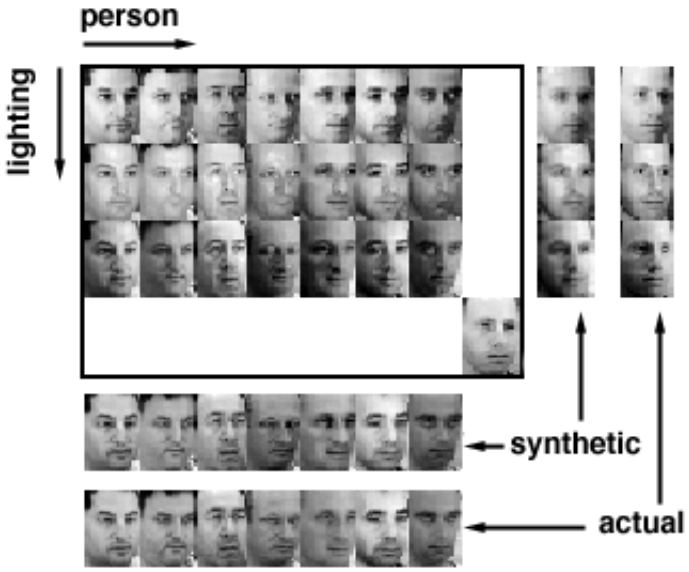

Figure 10: An example of translation.