UW CSE Technical Report 03-06-01

Probabilistic Bilinear Models for Appearance-Based Vision

D.B. Grimes A.P. Shon R.P.N. Rao
Dept. of Computer Science and Engineering
University of Washington
Seattle, WA 98195

Abstract

We present a probabilistic approach to learning object representations based on the "content and style" bilinear generative model of Tenenbaum and Freeman. In contrast to their earlier SVD-based approach, our approach models images using particle filters. We maintain separate particle filters to represent the content and style spaces, allowing us to define arbitrary weighting functions over the particles to help estimate the content/style densities. We combine this approach with a new EM-based method for learning basis vectors that describe content-style mixing. Using a particle-based representation permits good reconstruction despite reduced dimensionality, and increases storage capacity and computational efficiency. We describe how learning the distributions using particle filters allows us to efficiently compute a probabilistic "novelty" term. Our example application considers a dataset of faces under different lighting conditions. The system classifies faces of people it has seen before, and can identify previously unseen faces as new content. Using a probabilistic definition of novelty in conjunction with learning content-style separability provides a crucial building block for designing real-world, real-time object recognition systems.

1. Introduction

Probabilistic methods for recognizing objects and the contexts in which they appear have produced encouraging results in recent years [12, 6, 7]. The bilinear generative model [13] provides one such framework for separating the features of an object (its "content") from the context in which it is presented (its "style"). The model describes an image as a multiplicative combination of an m-dimensional content vector $\mathbf{x}$ representative of a class of objects and an n-dimensional style vector $\mathbf{y}$ representative of the conditions under which the object is viewed (e.g. lighting or pose). A set of basis vectors $\mathbf{w}_{ij}$ describe how content and style representations mix to generate the image $\mathbf{z}$:

$$\mathbf{z} = f(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{w}_{ij} x_i y_j \tag{1}$$

Previous results using the bilinear model [13] used a sum squared error (SSE) criterion in developing learning and inference procedures. However, no previous work attempts to learn a probabilistic model of how $\mathbf{x}$ vectors are distributed given a certain content class, or similarly how $\mathbf{y}$ vectors are distributed given a certain style class. Modeling distributions rather than points is important if there is uncertainty in our representations, e.g. if we receive noisy images or the content in the image changes style over time. Many probabilistic approaches (e.g. Gaussian mixture models) seek to simplify probabilistic representations using analytically tractable closed form distributions. However, not all distributions of interest in the content and style spaces are necessarily Gaussian. Especially in the case of dimensionality reduction, where we want to maintain style-content seperability using low-dimensional spaces, nonparametric methods may be needed to describe these distributions.

1.1. Particle filters

Particle filters (also called bootstrap filters [4] or Condensation [5]) have emerged in recent years as an efficient method for approximating arbitrary probability densities using samples. The particle filter algorithm iteratively estimates regions of high density by representing them as discrete sample points. Each iteration of the algorithm assigns a likelihood to each particle that it matches the observed state of the system given the prior estimate (such likelihoods are often called *weights*). The weights are assigned by a weighting function that defines how well the particle reflects observed data. Next the algorithm randomly samples from the weighted set of particles; the probability of a particle being picked during sampling is proportional to its weight. After picking a new set of particles, the algorithm applies an update function to the new particles that reflects the state dynamics of the system being estimated. If no dynamics are known, for example, we might assume the update function is described by an identity function with zero-mean Gaussian noise. Our algorithm uses particle filters to represent densities in the content and style spaces.

1.2. Advantages of our algorithm

Our probabilistic bilinear algorithm incorporates four improvements over the previous approach [13]:

1. **Representational capacity:** Unlike the original "asymmetric" model proposed by Tenenbaum and Freeman, our model is able to perform dimensionality reduction in both the content and style spaces.

2. **Novelty:** The probabilistic framework lends itself to a definition of novelty for identifying new content and style classes.

3. **Computational complexity:** Complexity of the previous SVD-based approach is proportional to k^2, where k is the number of pixels in the image. Complexity of our algorithm is proportional to pk, where p is a fixed number of particles (generally proportional to the dimensionality of the content/style spaces), so the algorithm scales more favorably for large images. Importantly, p is a free parameter, enabling the algorithm to run more easily on systems with limited computational resources. Tuning p provides a tradeoff between reconstruction accuracy and computational parsimony.

4. **Dynamics and priors:** Using particle filters allows us to define arbitrary weighting functions for the particles. This in turn permits use of priors on content and style and the addition of dynamics, if prior information about the representations is known or the content and/or style change over time. Particle filters also allow the algorithm to represent non-parametric content and style densities.

2. Previous work

In the original model by Freeman and Tenenbaum, an iterative SVD-based procedure is used to estimate a content and a style for each image in the training set according to a least-squares error criterion. The original model is able to reconstruct images provided as part of the training set. It is also able to classify previously seen content and style using a Gaussian mixture model in the image space.

However, the original model suffers from two limitations: it lacks a framework for incorporating prior information about the images, and it makes the related assumption that content and style representations are distributed according to a Gaussian distribution in their respective spaces. Fig. 1(b) shows that dimensionality reduction (where content dimensionality $m < nc$ and style dimensionality $n < ns$ for nc distinct content classes and ns style classes) can generate non-Gaussian distributions in the content space. Dimensionality reduction is important for any real-world system to efficiently learn a large number of content and style classes.

3. Modeling content and style using particle filters

To overcome the limitations of the original model, we assume a probabilistic bilinear generative model (Fig. 1(a)). The model assumes that two hidden variables (vectors) $\mathbf{x}$ and $\mathbf{y}$ are respectively generated by some content class C^i and style class S^j. The hidden variables combine to form an image $\mathbf{z}$ according to some linear mixture matrix $\mathbf{W}$. Our task is therefore to estimate the distributions for $\mathbf{x}$ and $\mathbf{y}$, and to reestimate $\mathbf{W}$ as we adjust those densities. Since we wish to allow arbitrary content and style densities, we represent the densities using a nonparametric approach: particle filters. This also circumvents the problem of incorporating prior information: we can include prior knowledge by simply changing the weighting functions of the content and style particle filters. The weighting functions can take on arbitrary forms, not necessarily corresponding to any closed-form distribution.

Fig. 1(b) plots reconstruction likelihood surfaces for images in an example content class. The first three columns show surfaces for individual images from the same class; because each image has a different style, each of the Gaussian clouds representing that image's content lies in a different location. The final column shows the content representation for all images in the class taken together. This multimodal shape is not easily captured by the linear learning approaches used in the original model. While the "asymmetric" model of Tenenbaum and Freeman, which learns a separate linear model for each style, might seem applicable, it is of limited utility in dimensionality reduction: the model cannot simultaneously reduce the number of style and content dimensions used to represent an image set. The ability to represent content and style densities using sample sets (possibly without a reasonable parametric form) is key to our system's scalability.

Each image's content and style are represented by a cloud of discrete sample particles (Fig. 1(b)). By the *content sample set* we denote the union of all content particles for the images that make up that class across all styles; we denote the *style sample set* analogously for style classes. Our EM algorithm begins by weighting samples that represent hypotheses over the structures of the content and style spaces. After this E step, we perform an M step that consists of resampling the particles and reestimating the matrix

$\mathbf{W}$ that describes how content and style mix.

3.1. E step

We assign each particle a weight w proportional to its likelihood of having generated the input image $\mathbf{z}$, based on the particle's ability to reconstruct the image. For the experiments shown in Figs. 2 and 3, we also enforce the constraint that particles from a particular content or style class conform to a Gaussian prior; i.e., for a given content particle $\mathbf{x}$, its likelihood drops as it moves away from the mean $\bar{\mathbf{x}}$ of all particles in its cluster. In this case our model becomes similar to Tenenbaum and Freeman's original model, except that we can express how well a given particle fits the Gaussian cluster (since we have a covariance matrix). For particles $\mathbf{x}$ in the content space and $\mathbf{y}$ in the style space, we define the reconstruction image $\hat{\mathbf{z}}$ as:

$$\hat{\mathbf{z}} = \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{W}_{ij} x_i y_j \tag{2}$$

For some reconstruction covariance matrix Σ and prior covariance Γ, the weight of $\mathbf{x}$ is:

$$\begin{aligned} w_{\mathbf{x}} \propto \alpha \exp\left(-\frac{1}{2}(\mathbf{z}-\hat{\mathbf{z}})^{\mathbf{T}}\Sigma^{-1}(\mathbf{z}-\hat{\mathbf{z}})\right) + \\ (1-\alpha)\exp\left(-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^{\mathbf{T}}\Gamma^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right) \end{aligned} \tag{3}$$

Here α is a factor that allows us to trade off the importance of accurate reconstruction versus accuracy of the Gaussian clustering. By making Gaussian clustering contribute more significantly to the weights, we increase style-content separability, but we may also make it more difficult for the system to find accurate reconstructions. The reconstructions shown in Fig. 1 uses a value of $\alpha = 1$; the experiments shown in Figs. 2 and 3 use a value of $\alpha = 0.05$. Although we use the reconstruction error with a Gaussian prior to compute weights, we note that arbitrary weighting functions (e.g. to represent prior information about the distributions in the content and style spaces) are easily implemented using this technique by multiplying the likelihood with a prior probability. As an example, we might specify that content or style particles obey a sparseness constraint [1, 8] to learn local rather than global features.

To weight a content particle $\mathbf{x}$ by Eqns. 2 and 3, we need to use a style particle $\mathbf{y}$ to perform image reconstruction. Likewise, estimating the weight for a style particle requires a content particle. Since computing likelihoods over all possible pairs of content and style particles for an image would be prohibitively expensive, we consider two *canonical particles* $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ for each image, respectively denoting the content and style particles with the highest likelihood values for the image. Each content particle on an iteration of the algorithm is weighted using the reconstruction image $\hat{\mathbf{z}}$ computed by mixing the content particle with the previous iteration's canonical style particle according to the bilinear model, and each style particle's weight is similarly derived using the reconstruction derived from the previous iteration's canonical content particle.

3.2. M step

On the M step of our algorithm, we resample the particles. Particles are drawn using sampling-importance resampling (SIR) [11]. To ensure that the particles explore the space, we add zero-mean Gaussian noise to each particle (with covariance matrix $\lambda\mathbf{I}$, $\lambda = 0.025$) after resampling. Additionally, a fixed fraction of the particles (20% in the simulations shown here) are distributed uniformly over the space, allowing the system to find solutions far outside the original set of particles. Our use of zero-mean noise is based on a lack of priors over the dynamics of the content and style spaces. Because the images in our data set represent static snapshots, the current implementation of our algorithm uses the identity function to represent particle dynamics (with additive Gaussian noise). The approach easily generalizes to the case of time-varying image sequences by applying equations for describing content-style dynamics.

Each M step of the learning algorithm also reestimates $\mathbf{W}$ to maximize the posterior probability of generating the training images. $\mathbf{W}$ is determined by solving a system of linear equations in $\hat{\mathbf{x}}, \hat{\mathbf{y}}$. We begin by defining ω_k as the vector version of the matrix $\mathbf{W}_k$ that describes content-style mixing for the kth pixel, i.e. ω_k is an $mn \times 1$ vector rather than an $m \times n$ matrix. We refer to the ith element in ω_k as ω_{ik}. We further define the vector $\mathbf{b}$ as the $mn \times 1$ vector version of the outer product matrix $\hat{\mathbf{x}}\hat{\mathbf{y}}^{\mathbf{T}}$. Maximizing the log likelihood Q of the data given the bilinear model gives the equation:

$$\frac{\partial Q}{\partial \omega_k} = -\frac{1}{C}\sum_{i=1}^{l}\left(z_k^i - \mathbf{b}^{i^{\mathbf{T}}}\omega_k\right)\mathbf{b}^i = 0 \tag{4}$$

The summation gives us a vector $\mathbf{v}$ of mn elements, corresponding to the left-hand side of the system of linear equations. Rewriting the sum and expanding the dot product on the right-hand side, we obtain the form:

$$\begin{aligned} \mathbf{v} = \omega_{1k} \cdot b_1^1 \cdot \mathbf{b}^1 + \omega_{2k} \cdot b_2^1 \cdot \mathbf{b}^1 \ldots \\ \omega_{1k} \cdot b_1^l \cdot \mathbf{b}^l + \omega_{2k} \cdot b_1^l \cdot \mathbf{b}^l \ldots \end{aligned} \tag{5}$$

This system is solvable given that the system is not underconstrained (i.e. if enough training images are available).

4. Novelty detection

We define *novelty* of an image $\mathbf{z}$ with respect to a set of disjoint learned content classes labeled $C^1 \ldots C^i$, given that

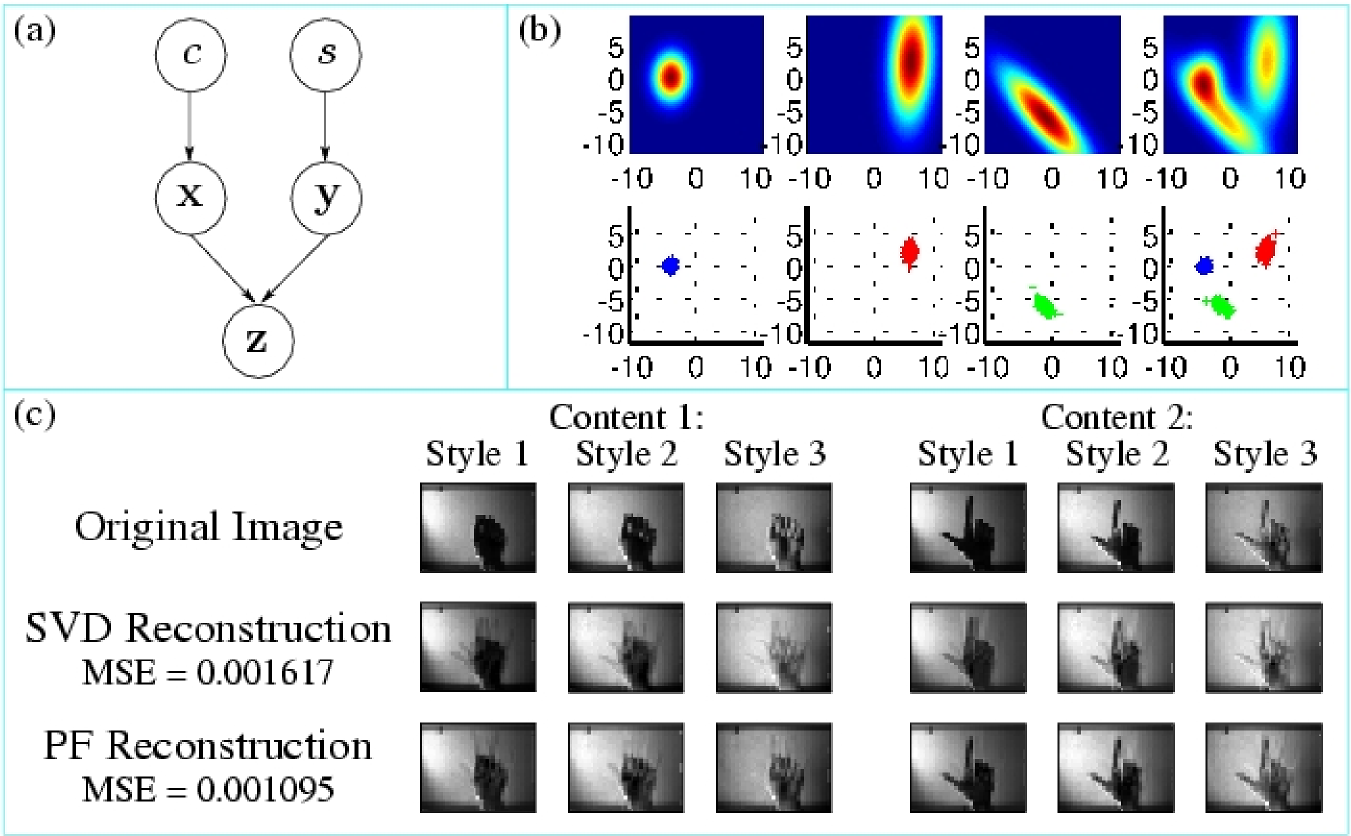

Figure 1: **Probabilistic bilinear models based on particle filters.** (a) Graphical model for our probabilistic framework. Distributions C and S generate content and style vectors $\mathbf{x}$ and $\mathbf{y}$, which in turn generate image $\mathbf{z}$. (b) Reduced dimensionalities in the content and style spaces produce non-Gaussian likelihood surfaces. A particle filter (lower row) is able to capture the non-parametric shapes of the content likelihood surfaces shown here (upper row). The first 3 columns show likelihood surfaces in the content space for 3 individual images composing a single content class. The final column shows the likelihood surface for the entire class. (c) Image reconstructions under Freeman and Tenenbaum's original asymmetric model and our model. First row: original images; second row: images reconstructed by the asymmetric model; third row: images reconstructed by our particle filter algorithm (after 50 iterations). Note our model's lower mean squared error on image reconstruction.

the image has style S, as:

$$p_{\text{novel}}(\mathbf{z}|S) = \prod_{i=1}^{nc} \left(1 - P(C^i|\mathbf{z}, S)\right) \qquad (6)$$

That is, novelty is the probability that the image was not generated using style S by any of the classes $C^1 \ldots C^i$. Note this definition can also be extended to cover novelty with respect to a style, a class, or some combination of styles and classes.

Our sample-based representation for content and style presents a problem when calculating novelty. Ideally, given the maximum likelihood particles $\mathbf{x}$ and $\mathbf{y}$ that represent a test image, we could determine the probability that the content (or style) sample for the test image was generated by each content (style) class provided during training. Unfortunately, with probability 1, no two samples are identical between the particles from the test image's sample set and the particles from each training class' sample set.

Density trees [9, 2] provide a convenient mechanism for turning a sample-based representation into a continuous representation over a space. Density trees are essentially binary search trees that partition a space. Each leaf node i is annotated with a density value d_i describing the *density* (summed sample weights $w_{mathbfx}$) of the samples $\mathbf{x}$ contained within the leaf:

$$d_i = \frac{\sum_{\mathbf{x} \text{ in } i} w_{\mathbf{x}}}{|V|} \qquad (7)$$

where $|V|$ denotes the volume of leaf node i. We normalize over the density values in the leaf nodes to produce a probability measure. Given that a sample is drawn from the sample set corresponding to a particular density tree, the probability of the sample being generated by a particular leaf node in the tree is thus proportional to the density at that leaf. We define a density tree for each content sample set and style sample set, giving us a continuous novelty measure over the content and style spaces. Fig. 3(a) shows an example density tree, plotted with the underlying samples from which it is composed. The tree performs a binary search on the space to find regions of high sample density, stopping recursion when a lower limit of square root of the number of samples is reached [14] or when the recursion depth exceeds 10. We compute novelty of an image by fitting a group of samples to the image, then determining the probability that the sample set was not generated by any of the trees representing learned content classes. Each sample in the sample set has a probability that it is novel with respect to all the learned content trees. The joint distribution $p(\mathbf{x}_0, \mathbf{x}_1, \ldots \mathbf{x}_n)$ over all the n samples in the set represents the probability that the entire sample set is novel. We approximate the joint density by assuming each particle is independent of the others in the set. Thus we add the log likelihoods over all the particles to determine the probability that the whole sample set lies outside the learned content classes encoded by the trees.

5. Results

We have compared our algorithm to the original model of Tenenbaum and Freeman [13] where appropriate. Fig. 1(c) shows the relative performances of our model and the original asymmetric model on a dataset of hands performing ASL finger spelling gestures (27 × 37 greyscale images, with 3 content classes and 3 style classes). Here $\alpha = 1$. Our algorithm produces lower sum-squared errors on the training set than the original SVD-based approach, and the resulting images appear qualitatively closer to the training set. Face data shown in Figs. 2 and 3 come from the Harvard face database[1].

5.1. Dimensionality reduction

Reducing the number of dimensions used to represent content and style is critical to implementing bilinear models on realistic input data. Storage capacity required for any given image increases linearly in the number of dimensions, and for particle filters in particular the time required to converge will tend to increase exponentially in the number of dimensions. Further, having too many unconstrained dimensions (e.g. when a real-time object recognition system is initialized and hasn't seen very many training images) results in singular or near-singular matrices when we recompute the basis vectors $\mathbf{W}$ on the M step of our algorithm.

Dimensionality reduction in the original asymmetric model is equivalent to maintaining several linear models (one for each style), and performing PCA for each model to learn global features for each content class [3, 10, 15]. The particle filter appears highly robust to reduced dimensionality. Fig. 2(a) shows some sample images from our training set of 450 images (10 different content classes under 45 different styles, resolution 24×21). Fig. 2(b) shows reconstruction quality for the same images with dimensionalities $m = 8$, $n = 20$ (reduced from $m = 10, n = 45$), while Fig. 2(c) shows reconstruction of the same images with reduced dimensionalities $m = 8, n = 4$. Note the similar quality of reconstruction despite the difference in dimensionalities. Fig. 2(d) plots a graph that shows how reduced dimensionality affects reconstruction accuracy. Holding content dimensionality fixed at $m = 8$, the graph shows how reconstruction improves with increasing style dimensionality n. Past $n = 4$, increased dimensionality does not significantly lower the MSE of the reconstructions. Points on the graph represent the mean MSE from 3 different runs of our algorithm, each with a different initial random seed. Error bars represent standard deviation.

5.2. Novelty detection and classification

The training algorithms developed in the original bilinear model, and the extensions proposed here, assume that all training images are provided with content and style labels. Novelty detection is critical for an unsupervised system to learn to differentiate categories of objects and contexts of presentation when such labels are not available. We tested our system using a set of 6 content classes seen during training and 2 novel content classes, with the goal of identifying the test images as displaying novel content. A simple threshold classifier is able to differentiate between the 6 faces that were part of the training set and the 2 novel faces not in the training set. Fig. 3(c) demonstrates the ability of our system to learn novelty. The algorithm marks each face with a novelty value calculated using a density tree according to Eqn. 6. Here we provide the algorithm with a style label S to assist detection of novel content. Starting from a uniform distribution, the content particles (shown in red) coding for the non-novel image collapse into a Gaussian cloud over approximately the same region as the particles for that content that were learned during training (shown in blue). For an image with novel content, the red particles converge to a spot outside any of the sample sets learned during training. This causes the algorithm to assign the image's content a high novelty value.

The algorithm is also able to recognize content classes it has seen before. Unlike the method proposed by Tenenbaum and Freeman [13], which reconstructs images and then fits a mixture model to the results, our EM algorithm classifies images within the content and style spaces themselves. We examine the density tree values for the content (or style) samples after the algorithm has converged. If the

[1] http://cvc.yale.edu/people/faculty/belhumeur.html

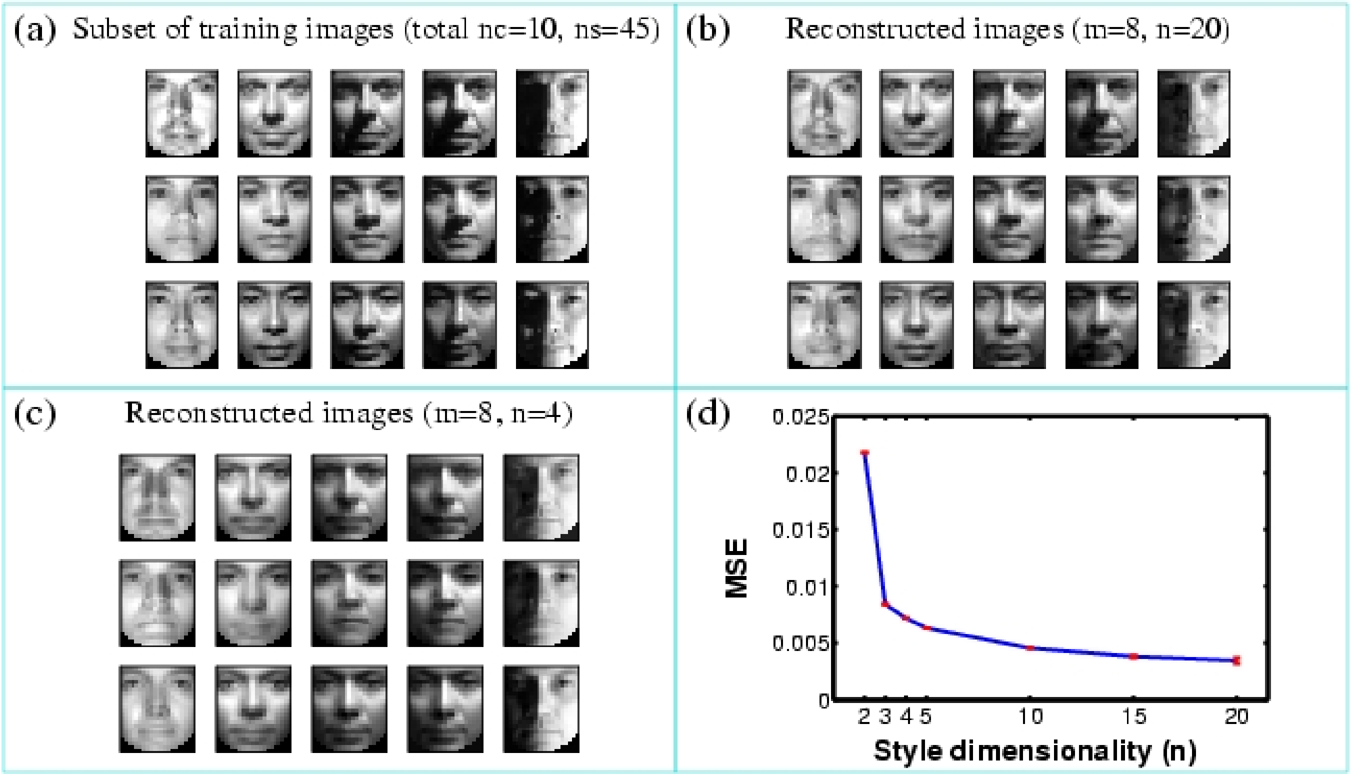

Figure 2: **Particle filters allow substantial dimensionality reduction.** (a) A subset of our training set of 450 images (10 contents, 45 styles). (b) Reconstructions when content dimensionality is reduced from $m = 10$ to $m = 8$ and style dimensionality is reduced from $n = 45$ to $n = 20$. (c) Reconstructions when $m = 8$ and $n = 4$. (d) MSE decreases as we increase the style dimensionality. The plot shows mean MSE over 3 different initial random conditions; errorbars indicate standard deviation. The EM algorithm runs for 100 iterations, ensuring convergence. Past $n = 4$, increasing the style dimensionality does not substantially increase reconstruction accuracy.

converged samples lie in regions of high density for one of the trees (see Fig. 3(a)), they are likely to represent the content or style stored by that tree, and will therefore be marked by the tree as having a high probability. The tree with the highest joint density $p(\mathbf{x}_0, \mathbf{x}_1, \ldots \mathbf{x}_n)$ over all the n samples in the set is the most likely tree for the converged sample set of particles. Again, we make the simplifying assumption that we can simply multiply the samples together to get an estimate of the likelihood that the sample set belongs to each content class. Thus, we add the log likelihoods assigned by a given tree over all the samples to obtain the log likelihood for the entire sample set. The tree with the highest log likelihood determines the class to which the sample set is assigned.

Over a set of 32 faces drawn from the training set, averaged over 3 different initial random conditions, the system is able to classify 82% ± 3% of the content correctly by checking values in the density tree. Over a set of 8 out-of-sample faces drawn from contents the system has seen before, but in novel styles, and averaged over 3 different initial random conditions, the system classifies correctly 71% ± 11% of the time. We do not provide the system with any hints as to the content or style.

It is possible to train the system as a classifier when the particles are completely unconstrained (i.e., when $\alpha = 1$). However, the resulting multimodal representation also makes classification much more difficult since the algorithm must now sample over a wider range of the style space to find a good canonical style particle $\hat{\mathbf{y}}$. Sample-based classification without a Gaussian prior does not currently perform robustly: after training on a small subset of 12 faces, the algorithm is able to classify 6 of them (again without providing any prior information about the content or style being shown). We are continuing to investigate improved methods for performing classification when content and style representations are multimodal.

6. Summary and conclusions

We have presented a new method for learning bilinear appearance-based models of objects based on particle filters. The system robustly reconstructs training images despite appreciable dimensionality reduction, outperforming the original asymmetric model in many cases. The system is computationally efficient, able to tune the number of particles to adapt to available computing resources. Using particle filters also provides a principled method for including prior information or dynamics in the content and style spaces. We have used our probabilistic model to define novelty with respect to content and style classes. Novelty detection provides a building block for future systems which will need to determine when to form new representational classes for objects.

Development of a novelty measure motivates the ability to learn new content and style classes. However, the new classes must not be learned at the expense of previously formed class definitions. To perform relearning, we could reconstruct a canonical image for each content-style combination learned thus far by sampling from the appropriate density trees and iterating the algorithm until it converges to a good reconstruction. We would then combine the resulting images with images defining novel classes the system may have acquired, and rerunning our EM algorithm. Assuming that the algorithm learned a good representation for the original training set, the relearned data should not disturb the original set of canonical particles. Alternatively, we could simply remember the canonical particles for each image in the original training set (possibly requiring a large amount of data storage). We are currently investigating how parameterization of the system affects its ability to perform stable relearning.

One future direction would be to extend our EM algorithm to cover sparseness priors, allowing us to learn local features rather than global ones. Another possible extension of our work would cover the case where dynamics in the style space are important, e.g. if objects are moving in a scene or if lighting conditions are changing relatively quickly over time. Over the long term, we anticipate incorporating our algorithm into a larger vision system for context-invariant appearance-based recognition of objects, capable of identifying and representing new object types as it encounters them.

References

[1] A. J. Bell and T. J. Sejnowski. The ‘independent components’ of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.

[2] J. L. Bentley. Multidimensional divide and conquer. *Comm. ACM*, 23(4):214–219, 1980.

[3] F. De la Torre and M. Black. Robust principal component analysis for computer vision. In *ICCV ’01*, volume I, pages 362–369, 2001.

[4] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140:107–113, 1993.

[5] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[6] N. Jojic and B. Frey. Topographic transformation as a discrete latent variable. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. Cambridge, MA: MIT Press, 2000.

[7] S. K. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[8] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

[9] S. M. Omohundro. Bumptrees for efficient function, constraint, and classification learning. In R. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 693–699. 1991.

[10] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *CVPR ’94*, pages 84–90, 1994.

[11] D. B. Rubin. Using the SIR algorithm to simulate posterior distributions. In M. Bernardo, K. DeGroot, D. Lindley, and A. Smith, editors, *Bayesian Statistics 3*. Oxford, UK: Oxford University Press, 1988.

[12] B. Schiele and A. Pentland. Probabilistic object recognition and localization. In *ICCV ’99*, volume I, pages 177–182, 1999.

[13] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[14] S. Thrun, J. C. Langford, and D. Fox. Monte Carlo hidden Markov models: Learning non-parametric models of partially observable stochastic processes. In *Proc. 16th International Conf. on Machine Learning*, pages 415–424. Morgan Kaufmann, San Francisco, CA, 1999.

[15] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

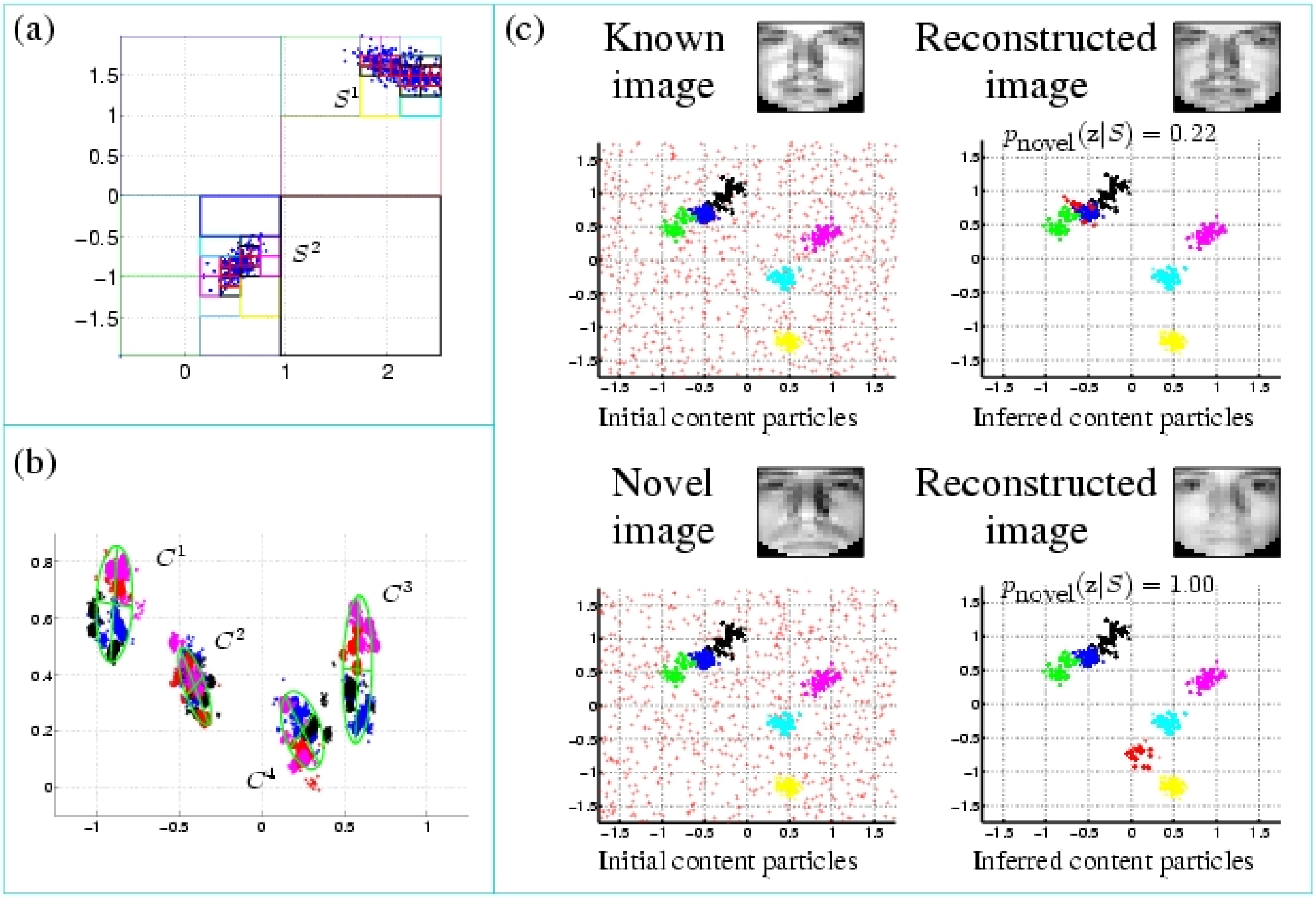

Figure 3: **Using a probabilistic model allows novelty detection.** (a) Example of a 2-D style sample set partitioned by a density tree. (b) Content representations under a Gaussian prior. Each Gaussian represents a separate content class; ellipses denote principal eigenvectors of the covariance matrices. Individual images (shown as different colored clouds of particles) can cluster into non-parametric shapes, but weights for the particles include a term to pull images with similar content toward one another, maintaining style-content invariance. (c) Contents in the algorithm's training set (top row) are marked as non-novel, while contents that are previously unseen (bottom row) are marked as novel. Evaluation of each image starts with a uniformly distributed set of particles, shown in red (first column). After running our EM algorithm for 5 iterations, the content particles (in red, top right) coding for the non-novel image have collapsed into the same region of the content space as the particles that coded for that content class during training. This means the converged content particles lie within regions of high density for this class' density tree, leading to a low novelty score $p_{\text{novel}}(\mathbf{z}|S)$. In contrast, the novel image's particles (in red, bottom right) converge to a Gaussian cloud outside of any other learned sample sets. Thus, the image's content representation lies outside the regions of high density for all learned density trees. This causes the algorithm to assign high novelty to the image.