Learning Inhomogeneous FRAME Models for Object Patterns

Jianwen Xie[1], Wenze Hu[2], Song-Chun Zhu[1], and Ying Nian Wu[1]
[1]University of California, Los Angeles (UCLA), USA
jianwen@ucla.edu, {sczhu, ywu}@stat.ucla.edu
[2]Google Inc, USA
wenzehu@google.com

Abstract

We investigate an inhomogeneous version of the FRAME (Filters, Random field, And Maximum Entropy) model and apply it to modeling object patterns. The inhomogeneous FRAME is a non-stationary Markov random field model that reproduces the observed marginal distributions or statistics of filter responses at all the different locations, scales and orientations. Our experiments show that the inhomogeneous FRAME model is capable of generating a wide variety of object patterns in natural images. We then propose a sparsified version of the inhomogeneous FRAME model where the model reproduces observed statistical properties of filter responses at a small number of selected locations, scales and orientations. We propose to select these locations, scales and orientations by a shared sparse coding scheme, and we explore the connection between the sparse FRAME model and the linear additive sparse coding model. Our experiments show that it is possible to learn sparse FRAME models in unsupervised fashion and the learned models are useful for object classification.

1. Introduction

Generative models. Developing generative models for image patterns is one of the most fundamental problems in vision. Although the past decade has witnessed tremendous advance in developing discriminative methods for object recognition, the progress in developing generative models has been lagging behind. The goal of this paper is to develop generative models for object patterns and explore their connections with existing theories in image representation and modeling.

The foundation of our work is the FRAME (Filters, Random field, And Maximum Entropy) model that Zhu, Wu, and Mumford (1997) [24] proposed for texture patterns. Being a texture model, FRAME is a spatially stationary Markov random field model, and it is the maximum entropy distribution that reproduces the observed marginal histograms of responses from a band of filters, where for each filter tuned to a specific scale and orientation, the marginal histogram is spatially pooled over all the pixels in the image domain.

Inhomogeneous FRAME. In this article, we investigate an inhomogeneous version of the FRAME model for representing object patterns instead of texture patterns. The inhomogeneous FRAME model is a spatially non-stationary random field, and is the maximum entropy distribution that reproduces distributions or statistics of filter responses at individual locations, scales and orientations without spatial pooling. We call this model the *dense* FRAME model because it seeks to reproduce statistical properties of filter responses over *all* the locations, scales and orientations. An inhomogeneous model has also been proposed by Liu, Zhu and Shum [10] for face shape data.

Sparse FRAME. The dense FRAME model can be computationally and inferentially demanding with its large number of parameters. We then investigate a sparsified version of the inhomogenous FRAME model which we call the *sparse* FRAME model. Instead of reproducing statistical properties of filter responses at all the locations, scales and orientations, the model seeks to reproduce statistical properties of filter responses at a relatively small number of selected locations, scales and orientations. We explore the connection between the sparse FRAME model and the linear additive sparse coding model [14], and we propose to select the locations, scales and orientations of the filter responses within the linear additive framework using a shared sparse coding scheme. This connects two major frameworks for image representation and modeling, namely the sparse coding framework with its root in harmonic analysis and the Markov random field framework with its root in statistical physics.

Our experiments show that both the dense and sparse versions of the inhomogeneous FRAME model are capable of generating realistic object patterns observed in images of natural scenes. Our experiments also show that it is possible

to learn sparse FRAME models in unsupervised manner and the learned models can be useful for image classification.

Related work. (1) Energy-based model. The FRAME model is an energy-based model. Other examples include field of experts [16], product of experts [5], restricted Boltzmann machine and its various extensions [6]. Compared to these models, the sparse FRAME performs feature selection via a linear additive model and it can reconstruct the training images. (2) Sparse coding model. The sparse FRAME selects the features via a shared sparse coding scheme. Compared to existing methods based on sparse coding [18, 13, 23], the sparse FRAME defines an explicit probability distribution on image intensities and can synthesize images by sampling from the distribution.

In this paper, we assume that the bank of filters are given, such as Gabor filters and difference of Gaussian (DoG) filters as in the original FRAME model. They can be learned if the training data are abundant.

2. Inhomogeneous FRAME model

Figure 1: The inhomogeneous FRAME is a generative model that seeks to represent and generate object patterns shown above.

Notation. We start from modeling roughly aligned images of object patterns from the same category, such as images in each row of Fig. 1. Let $\{\mathbf{I}_m, m = 1, ..., M\}$ be a set of training images defined on a common image domain $\mathcal{D}$. We use the notation $B_{x,s,\alpha}$ to denote a basis function such as a Gabor wavelet centered at pixel x (which is a two-dimensional vector), and tuned to scale s and orientation α. We assume that s and α take values within a finite and properly discretized range. The inner product $\langle \mathbf{I}, B_{x,s,\alpha} \rangle$ can be considered the filter response of $\mathbf{I}$ at pixel x to a filter tuned to scale s and orientation α. Let us assume that the basis functions are all normalized to have unit ℓ_2 norm.

Model. The inhomogeneous FRAME model is a probability distribution defined on $\mathbf{I}$,

$$p(\mathbf{I}; \lambda) = \frac{1}{Z(\lambda)} \exp\left(\sum_{x,s,\alpha} \lambda_{x,s,\alpha}(\langle \mathbf{I}, B_{x,s,\alpha} \rangle) \right) q(\mathbf{I}), \quad (1)$$

where $q(\mathbf{I})$ is a known reference distribution such as a Gaussian white noise model, $\lambda_{x,s,\alpha}()$ are one-dimensional functions that depend on (x, s, α), $\lambda = \{\lambda_{x,s,\alpha}, \forall x, s, \alpha\}$, and $Z(\lambda) = \mathrm{E}_q \left[\exp\left(\sum_{x,s,\alpha} \lambda_{x,s,\alpha}(\langle \mathbf{I}, B_{x,s,\alpha} \rangle) \right) \right]$ is the normalizing constant. $p(\mathbf{I}; \lambda)$ is said to be an exponential tilting of $q(\mathbf{I})$.

In the original FRAME model for stochastic texture patterns, $q(\mathbf{I})$ is assumed to be a uniform measure, and $\lambda_{x,s,\alpha}()$ is assumed to be independent of x (but dependent of s and α), so the model is spatially stationary. For modeling object patterns that are not spatially stationary, $\lambda_{x,s,\alpha}()$ must depend on x, in addition to s and α.

In the original homogeneous FRAME, the potential functions $\lambda_{s,\alpha}()$ are estimated non-parametrically as step functions. In the inhomogeneous FRAME model, we have to estimate $\lambda_{x,s,\alpha}()$ for each individual x. With small set of training images, we may not afford estimating $\lambda_{x,s,\alpha}()$ non-parametrically. We therefore have to parametrize $\lambda_{x,s,\alpha}$. In this article, we choose to use the parametrization

$$\lambda_{x,s,\alpha}(r) = \lambda_{x,s,\alpha}|r|, \quad (2)$$

where $r = \langle \mathbf{I}, B_{x,s,\alpha} \rangle$ is the filter response, and with slight abuse of notation, $\lambda_{x,s,\alpha}$ on the right hand side of (2) becomes a coefficient of the absolute value of the response.

Maximum likelihood learning. The FRAME model is a special case of the exponential family model, and the parameter $\lambda = (\lambda_{x,s,\alpha}, \forall x, s, \alpha)$ can be estimated from the training images $\{\mathbf{I}_m, m = 1, ..., M\}$ by MLE, which leads to the estimating equation

$$\mathrm{E}_\lambda \left(|\langle \mathbf{I}, B_{x,s,\alpha} \rangle| \right) = \frac{1}{M} \sum_{m=1}^{M} |\langle \mathbf{I}_m, B_{x,s,\alpha} \rangle|, \ \forall x, s, \alpha. \quad (3)$$

The MLE can be obtained by the stochastic gradient algorithm analyzed by Younes (1999) [22]. Let $\lambda^{(t)}$ be the current estimate of λ, and let $\{\tilde{\mathbf{I}}_m, m = 1, ..., \tilde{M}\}$ be a sample of synthesized images drawn from $p(\mathbf{I}; \lambda^{(t)})$. Then we can update λ by

$$\lambda^{(t+1)}_{x,s,\alpha} = \lambda^{(t)}_{x,s,\alpha} + \gamma_t \left(\frac{1}{M} \sum_{m=1}^{M} |\langle \mathbf{I}_m, B_{x,s,\alpha} \rangle| - \frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} |\langle \tilde{\mathbf{I}}_m, B_{x,s,\alpha} \rangle| \right),$$

where γ_t is the step size. The synthesized images $\{\tilde{\mathbf{I}}_m\}$ can be drawn from $p(\mathbf{I}; \lambda)$ by Hamiltonian Monte Carlo (HMC) [12], where a key step is to compute the gradient of the energy function, which is $\sum_{x,s,\alpha} \lambda_{x,s,\alpha} \mathrm{sign}(\langle \mathbf{I}, B_{x,s,\alpha} \rangle) B_{x,s,\alpha}$. The computation involves two rounds of convolutions (a bottom-up convolution followed by a top-down deconvolution), which can be efficiently implemented in Matlab by GPU. With HMC and warm start, $\{\tilde{\mathbf{I}}_m\}$ are produced by $\tilde{M}$ parallel chains.

The ratio of the normalizing constants

$$\frac{Z(\lambda^{(t+1)})}{Z(\lambda^{(t)})} = \mathrm{E}_{\lambda^{(t)}}\left[\exp\Big(\sum_{x,s,\alpha}(\lambda^{(t+1)}_{x,s,\alpha} - \lambda^{(t)}_{x,s,\alpha}) \times |\langle \mathbf{I}, B_{x,s,\alpha}\rangle|\Big)\right] \quad (4)$$

can be approximated by averaging over the sampled images $\{\tilde{\mathbf{I}}_m\}$. Starting from $\lambda^{(0)} = 0$ and $\log Z(\lambda^{(0)}) = 0$, we can compute $\log Z(\lambda^{(t)})$ along the learning process.

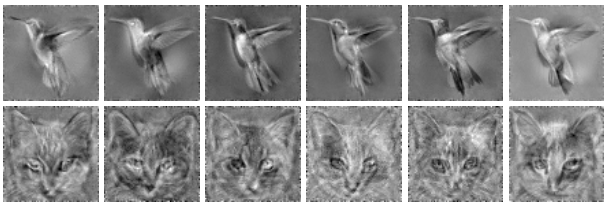

Figure 2: Synthesized images generated by the inhomogeneous FRAME model. The sizes of the images are 70×70. We work with grey-level images in this work.

Fig. 2 displays the synthesized images $\{\tilde{\mathbf{I}}_m\}$ generated by the models learned from training images in Fig. 1 (a separate model is learned from each training set).

3. Sparse FRAME model

Sparsification. In model (1), the (x, s, α) in $\sum_{x,s,\alpha}$ (as well as $\forall(x, s, \alpha)$ in (3)) is over all the pixels x and all the scales s and orientations α. We call such a model the dense FRAME. It is possible to sparsify the model by selecting only a small set of (x, s, α) so that $\sum_{x,s,\alpha}$ is restricted to this selected subset. More explicitly, we can write the sparsified model as

$$p(\mathbf{I}; \lambda) = \frac{1}{Z(\lambda)} \exp\left(\sum_{i=1}^{n} \lambda_i |\langle \mathbf{I}, B_{x_i,s_i,\alpha_i}\rangle|\right) q(\mathbf{I}), \quad (5)$$

where $\lambda = (\lambda_i, i = 1, ..., n)$. The model can still be trained by maximum likelihood as in the previous section, and properties such as maximum entropy still hold. The sparsification makes the computation faster and the inference more reliable.

In order to select the set of basis functions $(B_{x_i,s_i,\alpha_i}, i = 1, ..., n)$, we may use a sequential procedure such as filter pursuit [24] or projection pursuit [1]. In this work, we choose to follow a different strategy by exploiting the connection between sparse FRAME model and shared sparse coding model.

3.1. Connection with sparse coding

From sparse FRAME to shared sparse coding. Let us assume that the reference distribution $q(\mathbf{I})$ in the sparse FRAME model (5) is a Gaussian white noise model so that the pixel intensities follow $\mathrm{N}(0, \sigma^2)$ independently. For sparse FRAME, it is natural to assume that the number of selected basis functions n is much less than the number of pixels in $\mathbf{I}$, i.e., $n \ll |\mathcal{D}|$, where $\mathcal{D}$ is the image domain. For notational convenience, we can make $\mathbf{I}$ and $B_i = B_{x_i,s_i,\alpha_i}$, $i = 1, ..., n$ into $|\mathcal{D}|$-dimensional vectors, and let $\mathbf{B} = (B_1, ..., B_n)$ be the resulting $|\mathcal{D}| \times n$ matrix.

The connection between sparse FRAME and shared sparse coding is most evident if we temporarily assume that the selected basis functions $(B_i, i = 1, ..., n)$ are orthogonal (with unit ℓ_2 norm as assumed before). Extension to non-orthogonal $\mathbf{B}$ is straightforward but requires tedious notation (such as $(\mathbf{B}^T\mathbf{B})^{-1}$). For $\mathbf{B}$, we can construct $\bar{n} = |\mathcal{D}| - n$ basis vectors of unit norm $\bar{B}_1, ..., \bar{B}_{\bar{n}}$ that are orthogonal to each other and that are also orthogonal to $(B_i, i = 1, ..., n)$. Thus each image $\mathbf{I} = \sum_{i=1}^{n} r_i B_i + \sum_{i=1}^{\bar{n}} \bar{r}_i \bar{B}_i$, where $r_i = \langle \mathbf{I}, B_i\rangle$, and $\bar{r}_i = \langle \mathbf{I}, \bar{B}_i\rangle$. So we have the linear additive model $\mathbf{I} = \sum_{i=1}^{n} r_i B_i + \epsilon$, with $\epsilon = \sum_{i=1}^{\bar{n}} \bar{r}_i \bar{B}_i$ being the least squares residual image.

Under the Gaussian white noise $q(\mathbf{I})$, r_i and $\bar{r}_i$ are all independent $\mathrm{N}(0, \sigma^2)$ random variables because of the orthogonality of $(\mathbf{B}, \bar{\mathbf{B}})$. Let R be the column vector whose elements are r_i, and $\bar{R}$ be the column vector whose elements are $\bar{r}_i$. Then under the sparse FRAME model (5), only the distribution of R is modified by exponential tilting, which changes the distribution of R from Gaussian white noise $q(R)$ to $p(R; \lambda) = \exp\left(\sum_{i=1}^{n} \lambda_i |r_i|\right) q(R)/Z(\lambda)$, while the distribution of the residual coordinates $\bar{R}$ remains Gaussian white noise, and R and $\bar{R}$ remain independent. That is, $p(R, \bar{R}; \lambda) = p(R; \lambda) q(\bar{R})$.

Thus the sparse FRAME model implies a linear additive model $\mathbf{I} = \sum_{i=1}^{n} r_i B_i + \epsilon$, where $R \sim p(R; \lambda)$ and ϵ is a Gaussian white noise in the $\bar{n}$-dimensional residual space, and ϵ is independent of R. If we observe independent training images $\{\mathbf{I}_m, m = 1, ..., M\}$ from the model, then $\mathbf{I}_m = \sum_{i=1}^{n} r_{m,i} B_i + \epsilon_m$, i.e., $\{\mathbf{I}_m\}$ share a common set of basis functions $\mathbf{B} = (B_i, i = 1, ..., n)$ that provide sparse coding for multiple images simultaneously.

From shared sparse coding to sparse FRAME. Conversely, suppose we are given a shared sparse coding model of the form $\mathbf{I} = \sum_{i=1}^{n} c_i B_i + \epsilon = \mathbf{B}C + \epsilon$, where C is a column vector whose components are c_i. Assume $C \sim p(C)$ and $\epsilon \sim \mathrm{N}(0, I\sigma^2)$, where I is the $|\mathcal{D}|$-dimensional identity matrix, and ϵ and C are independent. Let $\delta = \mathbf{B}^T\epsilon$, each component of which $\delta_i = \langle \epsilon, B_i\rangle \sim \mathrm{N}(0, \sigma^2)$ independently. Then we can write $\mathbf{I} = \mathbf{B}R + \bar{\mathbf{B}}\bar{R}$, where $R = C + \delta$, and $\bar{\epsilon} = \bar{\mathbf{B}}\bar{R}$ is the projection of ϵ onto the space of $\bar{\mathbf{B}}$. Let $\tilde{p}(R)$ be the density of $R = C + \delta$, which is obtained by convolving $p(C)$ with Gaussian white noise density. Then $p(\mathbf{I}) = \tilde{p}(R) q(\bar{R}) = q(\mathbf{I})\tilde{p}(R)/q(R)$ since $q(\mathbf{I}) = q(R)q(\bar{R})$ under Gaussian white noise model ($d\mathbf{I} = dR d\bar{R}$ under orthogonality so there is no Jacobian term). If we

choose to model $\tilde{p}(R)/q(R) = \exp\left(\sum_{i=1}^{n} \lambda_i |r_i|\right)/Z(\lambda)$ by exponential tilting, we arrive at the sparse FRAME model.

Selection of basis functions. For orthogonal $\mathbf{B}$, as shown above, the probability density $p(\mathbf{I};\lambda) = q(\bar{R})p(R;\lambda) = q(\bar{R})q(R)\exp\left(\sum_{i=1}^{n} \lambda_i |r_i|\right)/Z(\lambda)$. Given a set of training images $\{\mathbf{I}_m, m = 1, ..., M\}$, and for a candidate set of basis functions $\mathbf{B} = (B_i, i = 1, ..., n)$, we can estimate $\lambda = (\lambda_i, i = 1, ..., n)$ by MLE $\lambda^\star$, and the resulting log-likelihood is

$$\sum_{m=1}^{M} \log p(\mathbf{I}_m;\lambda^\star) = \sum_{m=1}^{M} \left[\log q(\bar{R}_m) + \log p(R_m;\lambda^\star)\right]$$

$$= -\frac{1}{2\sigma^2}\sum_{m=1}^{M} ||\mathbf{I}_m - \mathbf{B}R_m||^2 - \frac{M\bar{n}}{2}\log(2\pi\sigma^2) \quad (6)$$

$$+ \sum_{m=1}^{M} \log p(R_m;\lambda^\star). \quad (7)$$

Suppose we are to choose an optimal $\mathbf{B}$, ideally we should maximize the sum of (6) and (7). We may interpret (6) as the negative coding length of the residual image ϵ by the Gaussian white noise model, and interpret (7) as the negative coding length of the coefficients R by the fitted model $p(R;\lambda^\star)$. If σ^2 is small, (6) can be more important, while the coding length of R for different $\mathbf{B}$ may not differ too much in comparison. So we choose to seek a $\mathbf{B}$ to maximize only (6) or equivalently minimize the overall reconstruction error $\sum_{m=1}^{M} ||\mathbf{I}_m - \mathbf{B}R_m||^2$. This reflects a two-step strategy in modeling $\{\mathbf{I}_m\}$. First, we find a set of basis functions $\mathbf{B}$ to reconstruct $\{\mathbf{I}_m\}$ as best as possible. Then we fit a statistical model for the reconstruction coefficients.

Non-orthogonality. If $\mathbf{B}$ is not orthogonal, which is the case in our work, the connection between the sparse FRAME and shared sparse coding still holds nonetheless. The responses $R = \mathbf{B}^T\mathbf{I}$, but the reconstruction coefficients $C = (\mathbf{B}^T\mathbf{B})^{-1}R$. The projection of $\mathbf{I}$ onto the subspace spanned by $\mathbf{B}$ is $\mathbf{B}C$. We can continue to assume the implicit $\bar{\mathbf{B}} = (\bar{B}_i, i = 1, ..., \bar{n})$ to be orthonormal, and that they are orthogonal to the columns of $\mathbf{B}$. We can also continue to let $\bar{R} = \bar{\mathbf{B}}^T\mathbf{I}$. In this setting, R and $\bar{R}$ are still independent under the Gaussian white noise model $q(\mathbf{I})$ because $\mathbf{B}$ and $\bar{\mathbf{B}}$ are still orthogonal to each other. Under the sparse FRAME model (5), it is still the case that only the distribution of R is modified by exponential tilting, while the distribution of $\bar{R}$ remains white noise and is independent of R. The distribution of R implies a distribution of the reconstruction coefficients C because they are linked by a linear transformation. We choose to model R instead of C by exponential tilting because the former is more convenient. Now the distributions of R and C involve the Jacobian terms such that $dRd\bar{R} = |\det(\mathbf{B}^T\mathbf{B})|^{1/2}d\mathbf{I} = |\det(\mathbf{B}^T\mathbf{B})|dCd\bar{R}$. By the same logic as in (6) and (7), we still want to find $\mathbf{B}$ to minimize the overall reconstruction error $\sum_{m=1}^{M} \|\mathbf{I}_m - \mathbf{B}C_m\|^2$.

3.2. The learning algorithm

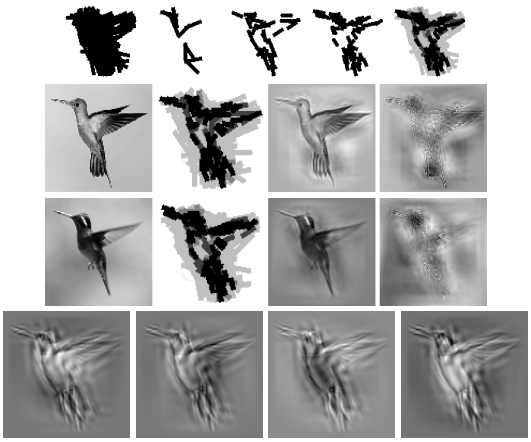

Figure 3: Reconstruction and synthesis. The images are 100×100. The number of selected wavelets is 300. The first row are symbolic sketches of selected Gabor wavelets at different scales, where each selected Gabor wavelet is symbolized by a bar. The first 4 sketches correspond to 4 different scales. The last one is the superposition of the 4 scales, where smaller scales appear darker. The second and third rows display two examples of the observed images, the deformed sketches, the reconstructed images, and the residual images. The last row displays examples of synthesized images generated by the learned model.

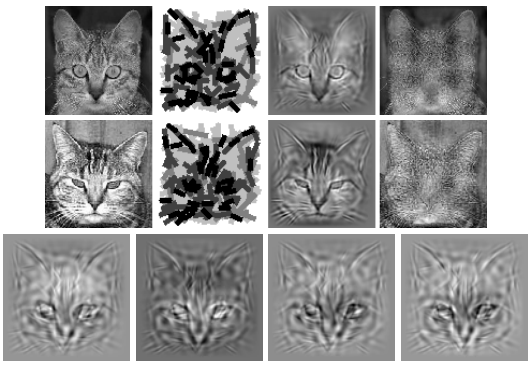

Figure 4: The images are 100×100. The number of selected Gabor wavelets is 400.

This subsection describes the learning algorithm, which consists of two stages. (1) Selecting $\mathbf{B} = (B_{x_i,s_i,\alpha_i}, i = 1, ..., n)$ by shared sparse coding. (2) Estimating $\lambda = (\lambda_i, i = 1, ..., n)$ given selected $\mathbf{B}$.

Deformable shared sparse coding. For training images $\{\mathbf{I}_m, m = 1, ..., M\}$, the shared sparse coding is of the form

$\mathbf{I}_m = \sum_{i=1}^{n} c_{m,i} B_{x_i,s_i,\alpha_i} + \epsilon_m$. We may allow these shared basis functions to perturb their locations and orientations to account for shape deformations, so we may extend the representation to deformable shared sparse coding first proposed by [20]

$$\mathbf{I}_m = \sum_{i=1}^{n} c_{m,i} B_{x_i+\Delta x_{m,i},s_i,\alpha_i+\Delta\alpha_{m,i}} + \epsilon_m, \tag{8}$$

where $(\Delta x_{m,i}, \Delta\alpha_{m,i})$ are the perturbations of the location and orientation of the i-th basis function. Both $\Delta x_{m,i}$ and $\Delta\alpha_{m,i}$ are assumed to vary within limited ranges (default setting: $\Delta x_{m,i} \in [-3,3]$ pixels alone the normal direction of the Gabor wavelet, and $\Delta\alpha_{m,i} \in \{-1,0,1\} \times \pi/16$). We want to select the basis functions $\{B_{x_i s_i,\alpha_i}, i = 1,...,n\}$ by minimizing $\sum_{m=1}^{M} \|\mathbf{I}_m - \sum_{i=1}^{n} c_{m,i} B_{x_i+\Delta x_{m,i},s_i,\alpha_i+\Delta\alpha_{m,i}}\|^2$.

Deformable shared matching pursuit algorithm . We can extend the matching pursuit algorithm [11] to select basis functions to code multiple images simultaneously, while inferring local perturbations by local max pooling [15]:

[0] Initialize $i \leftarrow 0$. For $m = 1,...,M$, initialize the residual image $\epsilon_m \leftarrow \mathbf{I}_m$.

[1] Let $i \leftarrow i+1$. Then we select $(x_i, s_i, \alpha_i) = \arg\max_{x,s,\alpha} \sum_{m=1}^{M} \max_{\Delta x,\Delta\alpha} |\langle \epsilon_m, B_{x+\Delta x,s,\alpha+\Delta\alpha}\rangle|^2$, where $\max_{\Delta x,\Delta\alpha}$ is local maximum pooling within the small ranges of $\Delta x_{m,i}$ and $\Delta\alpha_{m,i}$.

[2] For each m, given (x_i, s_i, α_i), infer the perturbations in location and orientation by retrieving the arg-max in the local maximum pooling of step [1]: $(\Delta x_{m,i}, \Delta\alpha_{m,i}) = \arg\max_{\Delta x,\Delta\alpha} |\langle \epsilon_m, B_{x_i+\Delta x,s_i,\alpha_i+\Delta\alpha}\rangle|^2$. Let the coefficient $c_{m,i} \leftarrow \langle \epsilon_m, B_{x_i+\Delta x_{m,i},s_i,\alpha_i+\Delta\alpha_{m,i}}\rangle$, and update the residual image by explaining away: $\epsilon_m \leftarrow \epsilon_m - c_{m,i} B_{x_i+\Delta x_{m,i},s_i,\alpha_i+\Delta\alpha_{m,i}}$.

[3] Stop if $i = n$, else go back to step [1].

Such a deformable shared matching pursuit algorithm was first proposed by [20], but it implemented a modified version that enforces approximated orthogonality of the selected basis functions.

Sparse FRAME as deformable template. After selecting $\mathbf{B} = \{B_i = B_{x_i,s_i,\alpha_i}, i = 1,...,n\}$, we can then model $\{\mathbf{I}_m\}$ by the sparse FRAME model (5), by estimating λ at MLE. $p(\mathbf{I};\lambda)$ in (5) now serves as the deformable template in that the log-likelihood ratio of the $\mathbf{I}_m$ is

$$\sum_{i=1}^{n} \lambda_i \max_{\Delta x,\Delta\alpha} |\langle \mathbf{I}_m, B_{x_i+\Delta x,s_i,\alpha_i+\Delta\alpha}\rangle| - \log Z(\lambda), \tag{9}$$

which serves as the template matching score, where we allow each selected B_i to perturb its location and orientation in view of (8), with the perturbations inferred by local max pooling. In the learning algorithm, again, let $\lambda^{(t)}$ be the current estimate of λ, let $\{\tilde{\mathbf{I}}_m, m = 1,...,\tilde{M}\}$ be the synthesized images drawn from $p(\mathbf{I};\lambda^{(t)})$ by $\tilde{M}$ parallel chains. Then we update λ by

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} + \gamma_t \left(\frac{1}{M} \sum_{m=1}^{M} \max_{\Delta x,\Delta\alpha} |\langle \mathbf{I}_m, B_{x_i+\Delta x,s_i,\alpha_i+\Delta\alpha}\rangle| - \frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} |\langle \tilde{\mathbf{I}}_m, B_i\rangle| \right).$$

The learned $p(\mathbf{I};\lambda)$ models the appearance of the undeformed template. So there is an explicit separation between appearance and shape variations. For HMC computation, the gradient of the energy function is of the form $\sum_{i=1}^{n} \lambda_i \text{sign}(\langle \mathbf{I}, B_i\rangle) B_i$, so HMC is like a generative process based on linear superpositions of $(B_i, i = 1,...,n)$.

After we learn λ and compute $Z(\lambda)$ as in (4), we can use the learned model as a deformable template to be matched to the testing image, where the template matching score at each location can be computed like (9).

Fig. 3 illustrates the basic idea. The training images are scaled to 100×100. The number of selected basis functions (Gabor and big DoG wavelets) n is set at 300. In principle it can be automatically determined by criteria like BIC. We normalize the images to have zero mean and unit variance, and we fix σ^2 of $q(\mathbf{I})$ to be .1. The first row displays the selected Gabor wavelets B_i, where each B_i is symbolized by a bar. The first four plots in the first row display the selected B_i at 4 different scales, from the largest to the smallest. The last plot in the first row is the superposition of the 4 scales, with smaller scales appear darker. The second and third rows display two training images $\mathbf{I}_m$ for two different m, the symbolic sketches of the deformed templates $(B_{x_i+\Delta x_{m,i},s_i,\alpha_i+\Delta\alpha_{m,i}}, i = 1,...,n)$, the reconstructed images $\sum_{i=1}^{n} c_{m,i} B_{x_i+\Delta x_{m,i},s_i,\alpha_i+\Delta\alpha_{m,i}}$, and the residual image ϵ_m. For the synthesized images $\tilde{\mathbf{I}}_m$ generated from the learned $p(\mathbf{I};\lambda)$, we project them onto the subspace spanned by $\mathbf{B}$. The last row displays projections of the four synthesized images. Fig. 4 shows another example.

4. Experiments

Project page: The code and more results and details can be found at http://www.stat.ucla.edu/~jxie/iFRAME.html.

1. Dense FRAME. Fig. 5 displays some images generated by the dense models learned from roughly aligned training images. We run a single chain in the learning process, i.e., $\tilde{M} = 1$ in this experiment.

2. Sparse FRAME. Fig. 6 displays some images generated by the sparse models learned from roughly aligned images. Experiment setting is the same as that in Fig. 3 except that the image sizes are typically 80×80, and the allowed displacement of a Gabor wavelet is up to 2 pixels. Number of wavelets is 300. We run $\tilde{M} = 36$ parallel chains in the learning algorithm.

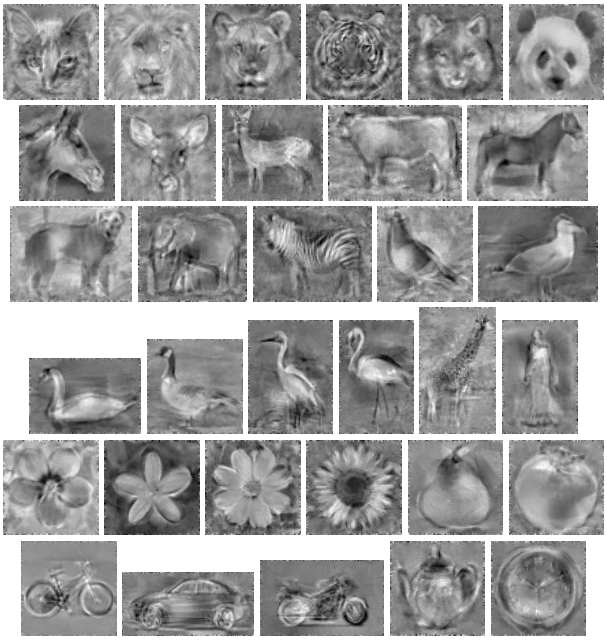

Figure 5: Images generated by the learned dense FRAME model. Typical sizes of the images are 70×70.

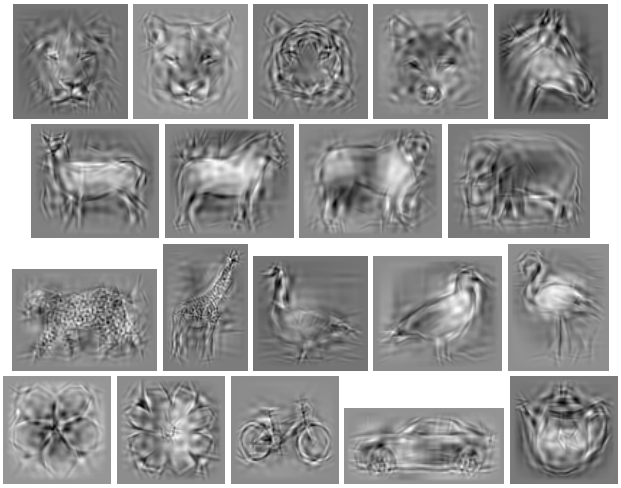

Figure 6: Images generated by the learned sparse FRAME models. Typical sizes of the images are 80×80. Number of selected wavelets is 300.

3. Detection by template matching. The learned model can be used for detection. Fig. 7 shows one example. The template is 100×100. The first image is a synthesized image generated by the model trained on 6 roughly aligned images with 250 wavelets. The other two images are testing images where the objects are located by the bounding boxes. We allow the change of the overall scale and orientation of the template. $\tilde{M} = 36$.

Figure 7: Detection. The first image (100×100) is generated by the learned model (with 250 wavelets). The rest are the testing images.

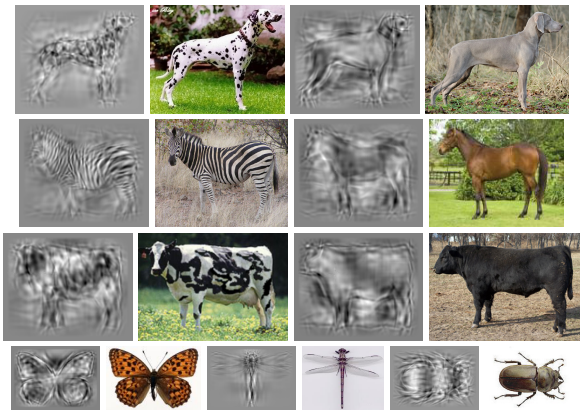

Figure 8: Clustering. Each row illustrates one clustering experiment by displaying a synthesized image (100×100) and a training example from each cluster. Number of wavelets in each model is 300. Each cluster has 15 images.

4. Clustering by mixture models. Model-based clustering can be accomplished by EM-type algorithm that fits mixtures of sparse FRAME models. Fig. 8 illustrates 4 experiments. The EM-type algorithm usually converges within 3-5 iterations. For each cluster, we generate $\tilde{M} = 144$ parallel chains in learning.

To evaluate the clustering accuracies, we use two measures: conditional purity and conditional entropy [19]. For a random training image, let x be its true category (unknown to the algorithm) and y be the inferred category. The conditional purity is defined as $\sum_y p(y) \max_x p(x|y)$ (the larger the better), and the conditional entropy is defined as $\sum_y p(y) \sum_x p(x|y) \log(1/p(x|y))$ (the smaller the better), where both $p(y)$ and $p(x|y)$ can be estimated from the training data. We perform 7 clustering experiments. The numbers of clusters vary from 2 to 5 and are assumed known in these experiments. The number of images in each cluster is typically 15 except in one experiment. We compare the performance of the sparse FRAME with that of k-mean based on HoG features. Table 1 displays the clustering accuracies and standard errors based on 10 repetitions of each experiment.

5. Unsupervised learning of codebooks. We can learn a codebook of sparse FRAME models from non-aligned im-

Table 1: Comparison of conditional purity (the first two rows) and conditional entropy (the last two rows) between sparse FRAME and k-mean for clustering

	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7
k-mean (purity)	0.623±0.016	0.870±0.043	0.933±0.141	0.825±0.121	0.911±0.086	0.888±0.091	0.687±0.110
FRAME (purity)	0.943±0.063	0.990±0.016	0.938±0.131	0.895±0.132	1.000±0.000	0.879±0.141	0.741±0.111
k-mean (entropy)	0.652±0.009	0.376±0.086	0.092±0.195	0.243±0.167	0.226±0.084	0.199±0.126	0.639±0.161
FRAME (entropy)	0.145±0.157	0.037±0.060	0.090±0.191	0.155±0.189	0.000±0.000	0.179±0.208	0.497±0.192

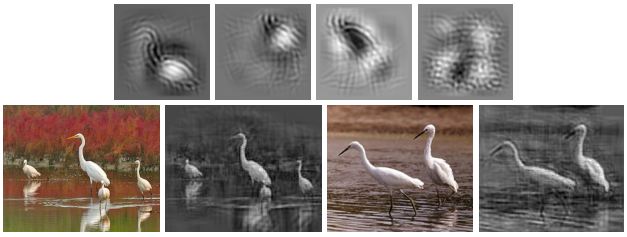

Figure 9: Codebook learning. A codebook of 4 models (each has 250 wavelets) are learned from 20 images. The first row displays the synthesized images (100 × 100) from the 4 models. The second row displays two training images and their reconstructions by the 4 models.

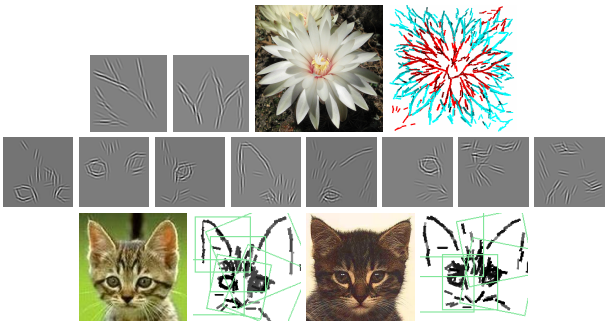

Figure 10: In each experiment, synthesized images (100 × 100) from the models of the learned codebook are displayed together with the training images and their sketches by the learned models, where each Gabor wavelet is illustrated by a bar. Lotus experiment: each model has 30 wavelets, learned from 7 images. Cat experiment: each model has 40 wavelets, learned from 20 images.

ages without annotation, by adopting the method of [7]. The learning algorithm iterates the following two steps: (1) Image encoding: given the current codebook, encode the training images by spatially translated, rotated, scaled versions of the models (templates) in the codebook. (2) Codebook re-learning: re-learn each model in the codebook from the image patches currently covered by this template. Figs. 9 and 10 illustrate experiments of codebook learning. For the experiments in Fig. 10, we select a small number (30 and 40) wavelets of a single scale, so the synthesized images mainly capture the edge patterns. We choose the number of codewords by hand, although it can in principle be chosen by BIC-like criterion [7].

6. Object classification on domain adaptation data sets. We test the sparse FRAME model by image classification on domain adaptation tasks, and compare with published results [17, 3, 2, 21, 8]. The 4 datasets are: Amazon, Webcam, DSLR and Caltech-256 [4]. Each dataset is regarded as a domain. For the experiment with single source training, 10 classes common to all 4 datasets are extracted. For the experiment with multiple sources training, all 31 classes in Amazon, Webcam and DSLR are used. We use the evaluation protocol in [2]. We randomly sample labeled data in the source domain as training examples, and unlabeled data in the target domain as testing examples. We learn a codebook of 3 sparse FRAME models for each category in unsupervised way, under the same setting as experiments in Fig. 10. We then combine the codebooks of all the categories. The maps of the template matching scores from the models in the combined codebook are computed for each image, and they are then fed into spatial pyramid matching [9], which equally divides an image into 1, 4, 16 areas, and concatenates the maximum scores within different image areas into a feature vector. We use multi-class SVM to train image classifiers based on the feature vectors, and then evaluate the classification accuracies of these classifiers on the testing domain. For each pair of source and target domains, we report averaged accuracies on target domains as well as standard errors. Table 2 show the comparisons of recognition accuracies on target domains for single source training and multiple source training. It can be seen that the learned codebooks perform well even though we do not make use of any domain adaptation techniques.

5. Conclusion

The sparse FRAME model has the following properties. (1) It can reconstruct the training images. (2) It can synthesize new images. (3) It separates shape deformations and appearance variations. (4) It gives interpretable sketches. (5) Dictionaries or codebooks of models can be learned in unsupervised manner. (6) It combines rich traditions of harmonic analysis and Markov random field models.

Table 2: Classification accuracies on single source four domains benchmark (the first table, C: caltech, A: amazon, D: DSLR, W: webcam) and multiple sources three domains benchmark (the second table)

Method	C→A	C→D	A→C	A→W	W→C	W→A	D→A	D→W
Metric [17]	33.7±0.8	35.0±1.1	27.3±0.7	36.0±1.0	21.7±0.5	32.3±0.8	30.3±0.8	55.6±0.7
SGF [3]	40.2±0.7	36.6±0.8	37.7±0.5	37.9±0.7	29.2±0.7	38.2±0.6	39.2±0.7	69.5±0.9
GFK [2]	46.1±0.6	55.0±0.9	39.6±0.4	56.9±1.0	32.8±0.7	46.2±0.7	46.2±0.6	80.2±0.4
FDDL [21]	39.3±2.9	55.0±2.8	24.3±2.2	50.4±3.5	22.9±2.6	41.1±2.6	36.7±2.5	65.9±4.9
Our method	62.2±1.6	52.2±4.0	46.7±2.5	53.2±4.9	39.1±3.0	53.2±4.4	55.3±2.9	72.4±3.1

Source	Target	SGF [3]	RDALR [8]	FDDL [21]	Our method
DLSR, amazon	webcam	52±2.5	36.9±1.1	41.0±2.4	52.2±1.4
amazon, webcam	DSLR	39±1.1	31.2±1.3	38.4±3.4	54.5±3.3
webcam, DSLR	amazon	28±0.8	20.9±0.9	19.0±1.2	32.1±1.6

Acknowledgments. The work is supported by NSF DMS 1310391, ONR MURI N00014-10-1-0933, DARPA MSEE FA8650-11-1-7149.

References

[1] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, **82**, 249-266, 1987. 3

[2] B. Gong, Y. Shi, F. Sha and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. *CVPR*, 2066–2073, 2012. 7, 8

[3] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: an unsupervised approach. *ICCV*, 999-1006, 2011. 7, 8

[4] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, Caltech, 2007. 7

[5] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, **14**, 1771–1800, 2002. 2

[6] G. E. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, **18**, 1527-1554, 2006. 2

[7] Y. Hong, Z. Si, W. Hu, S.-C. Zhu, and Y. N. Wu, Unsupervised learning of compositional sparse code for natural image representation. *Quarterly of Applied Mathematics*, in press, 2013. 7

[8] I. Jhou, D. Liu, D. T. Lee, and S. Chang. Robust visual domain adaptation with low-rank reconstruction. *CVPR*, 2168-2175, 2012. 7, 8

[9] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006. 7

[10] C. Liu, S.-C. Zhu, and H.-Y. Shum. Learning Inhomogeneous Gibbs Model of Faces by Minimax Entropy. *ICCV*, 281-287, 2001. 1

[11] S. Mallat and Z. Zhang. Matching pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing*, **41**, 3397-3415, 1993. 5

[12] R. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2011. 2

[13] G. Obozinski, M. J. Wainwright, and M. I. Jordan, Support union recovery in high-dimensional multivariate regression, *Annals of Statistics*, **39**, 1–47, 2011. 2

[14] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, **381**, 607-609, 1996. 1

[15] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, **2**, 1019–1025, 1999. 5

[16] S. Roth and M. Black. Fields of experts. *IJCV*, **82**, 205-229, 2009. 2

[17] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *ECCV*, 213-226, 2010. 7, 8

[18] J. Tropp, A. Gilbert, and M. Strauss. Algorithms for simultaneous sparse approximation. part I: Greedy pursuit. *Journal of Signal Processing*, **86**, 572–588, 2006. 2

[19] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: a comparison. *IJCV*, 2009 6

[20] Y. N. Wu, Z. Si, H. Gong, and S.-C. Zhu. Learning active basis model for object detection and recognition. *IJCV*, **90**, 198-235, 2010. 5

[21] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. *ICCV*, 543-550, 2011. 7, 8

[22] L. Younes. On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics and Stochastic Reports*, **65**, 177-228, 1999. 2

[23] M. Zeiler, G. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. *ICCV*, 2011. 2

[24] S.-C. Zhu, Y. N. Wu, and D. B. Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, **9**, 1627-1660, 1998. 1, 3