Manifold Guided Composite of Markov Random Fields for Image Modeling

Dahua Lin
CSAIL, MIT
dhlin@mit.edu

John Fisher
CSAIL, MIT
fisher@csail.mit.edu

Abstract

We present a new generative image model, integrating techniques arising from two different domains: manifold modeling and Markov random fields. First, we develop a probabilistic model with a mixture of hyperplanes to approximate the manifold of orientable image patches, and demonstrate that it is more effective than the field of experts in expressing local texture patterns. Next, we develop a construction that yields an MRF for coherent image generation, given a configuration of local patch models, and thereby establish a prior distribution over an MRF space. Taking advantage of the model structure, we derive a variational inference algorithm, and apply it to low-level vision. In contrast to previous methods that rely on a single MRF, the method infers an approximate posterior distribution of MRFs, and recovers the underlying images by combining the predictions in a Bayesian fashion. Experiments quantitatively demonstrate superior performance as compared to state-of-the-art methods on image denoising and inpainting.

1. Introduction

Generative image models instantiate prior knowledge crucial for solving a variety of low-level vision problems (*e.g*. image denoising and inpainting). Previous approaches are comprised of two main categories of models – *subspace and manifold models* and *Markov random fields*. The former, typically used for modeling specific object classes, are adept at capturing structured appearance, while the latter, often used to model natural images, have demonstrated better generalization properties. Here, we propose a new approach to image modeling, which, via a unified Bayesian framework, combines the strength of both manifold models and MRFs, and overcomes their respective weakness.

It has long been observed that real images concentrate about low-dimensional manifolds within the original representation space. Starting with the introduction of PCA for face recognition [24], there have been extensive efforts [3,5,6,11] to develop subspace analysis techniques for images. To break the linear assumption underlying these models, a variety of methods [1, 25, 26] have adapted subspaces to nonlinear manifolds. Nonetheless, successful application has been largely restricted to modeling specific object classes. The difficulty in application to generic images is partly ascribed to the shared common assumption of these methods, *i.e*. the object appearance has a similar global structure that can be effectively captured by a manifold via training. Natural images, however, often exhibit structural variability that exceeds the modeling capacity of existing manifold models.

Markov random fields, which emphasize local coherence rather than global structure, provide a general probabilistic formulation for low level vision problems, including image restoration [14, 16, 27], optical flow estimation [15], and super-resolution [8]. Early work with MRFs [9] utilized pairwise clique potentials, severely limiting the expressiveness of the model. Zhu *et al*. proposed FRAME [28], an MRF with clique potentials defined upon locally supported filter responses to overcome this limitation. Roth and Black [14] proposed the Field of Experts (FoEs), which extends this idea by formulating local potentials as products of experts, and subsequently [16] proposed Steerable Random Fields, with clique potentials defined on the responses of steerable filters, oriented towards local directions.

Current methods that utilize MRFs for natural image modeling are limited in two aspects. First, many methods rely on the distributions of filter responses to derive clique potentials, obscuring some aspects of the generative model. As we shall see (*e.g*. Figure 3) such models have limited capacity to describe local patterns. Second, non-Gaussian potentials usually lead to computational difficulties in both learning and inference. For example, contrastive divergence, which is known to converge slowly, is used for maximum likelihood estimation in [14]. Consequently, a variety of approximate formulations [17, 20, 27] have been proposed. Recent approaches [19, 21] suggest the use of conditional random fields (CRFs) that directly model the posterior instead of the prior. However, as articulated by Schmidt *et al*. [18], the gain in efficiency often comes with the loss of generality or probabilistic rigorousness.

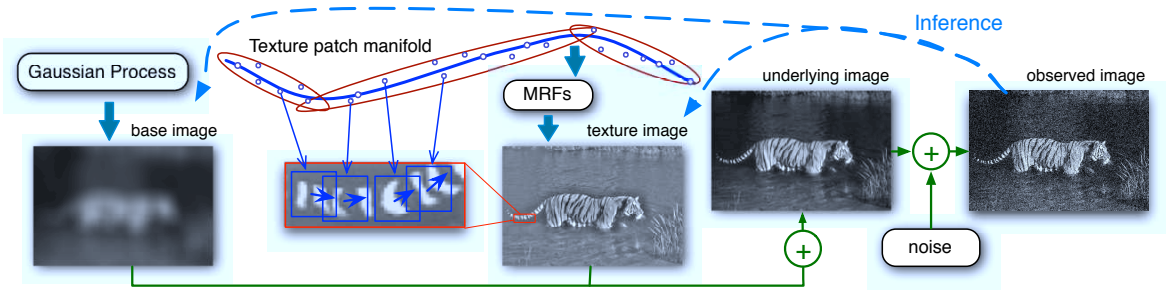

Figure 1. This is the overall framework of the generative image model. Each image is considered as a combination of a base image that roughly reflects the global appearance, and a texture image that captures the local texture patterns. The base image is generated from a GP prior; while the texture image is generated as a composite of oriented local patches from a patch manifold. An MRF is utilized to enforce coherence across patches. For denoising, the inference algorithm incorporates both the prior knowledge provided by this model and the observed information to derive a posterior distribution of the MRFs, and recover the underlying image in a Bayesian fashion.

We develop a new image model with a goal of addressing these issues. It is motivated by the following rationale: a defining characteristic distinguishing natural images from other signals is the coherence of local patterns even while the global appearance and structure vary dramatically from image to image and scene to scene. The proposed framework comprises a patch model that leverages the expressive power of manifold modeling to capture the variations of local patterns, and a family of MRFs to enforce coherence across patches. Specifically, to produce an image, oriented local models derived from the manifold are selected to model local patches, and conditioned on these models, an MRF is constructed to enforce coherence across patches.

It is worth noting that in the generative model described above, we actually establish a prior over a space of Gaussian MRFs, in which each MRF is conditioned on a configuration of local patch models. When applying this model to image restoration, we infer an approximate posterior distribution of MRFs, adopting a Bayesian approach that combines predictions over the space of MRFs to recover the original image. This contrasts with previous work utilizing a single MRF or CRF (either hand-crafted or learned) for low-level vision tasks. Formulating the image prior as a distribution over MRFs brings forth several benefits: (1) a probabilistically consistent generative model (see section 2), (2) the capacity to model heavy tailed characteristics or other statistical properties that are not well described by Gaussian models (see section 2.1), and (3) the availability of efficient algorithms for learning and inference (see section 3).

2. Image Model

Figure 1 provides an illustration of the generative and inference framework that we develop in the sequel. An image I is the superposition of an overly blurred *base image* B that reflects the global appearance, and a *textured image* Y, which is a coherent composite of local patterns. The prior distribution over the base images is described by a Gaussian process with a covariance function as

$$cov(B(x), B(x')) = a_B \exp\left(-\|x - x'\|^2/(2\sigma_B^2)\right). \quad (1)$$

where x and x' are pixel locations and the parameters a_B and σ_B can be learned from training images. Here, we focus on the model of texture images comprised of two main components: (1) *a generative patch model* that describes local patterns based on a manifold, and (2) *a conditional MRF* that enforces coherence across patches.

2.1. The Patch Manifold Model

Each image contains a collection of local patches which we represent as vectors of pixels with dimension d_p. Our construction of a generative model is motivated by two observations: (1) In natural images, intensity values of neighboring pixels are highly correlated, consequently, patches may be well approximated by a manifold of dimension lower than d_p, and (2) a patch and its rotated versions are equally likely for a natural image. The local patch generative model is thus comprised of three steps:

1. Canonical patch selection: Given the equivalence class of patches that are rotated versions of each other, we designate the patch with horizontal orientation[1] as the *canonical patch*. Canonical patches are described by a manifold of dimension $d_m < d_p$, where we use a mixture of d_m-dimensional hyperplanes, denoted by $H_1, \ldots, H_K$, to approximate the manifold. Each constituent hyperplane H_k is characterized by an offset vector $\boldsymbol{\mu}_k \in \mathbb{R}^{d_p}$ and a basis matrix $\mathbf{W}_k \in \mathbb{R}^{d_p \times d_m}$, such that each patch vector thereon can be expressed in form of $\boldsymbol{\mu}_k + \mathbf{W}_k\mathbf{z}$. To generate a canonical patch, we first choose a specific hyperplane, by drawing an index $s \sim \boldsymbol{\pi}$, where $\boldsymbol{\pi}$ is a prior distribution over the K hyperplanes. Then, we draw the *latent representation* $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ and obtain a patch as $\mathbf{x} = \boldsymbol{\mu}_s + \mathbf{W}_s\mathbf{z}$.

[1] For our purposes, the orientation of a patch is determined by the leading eigenvector of the structural tensor [4].

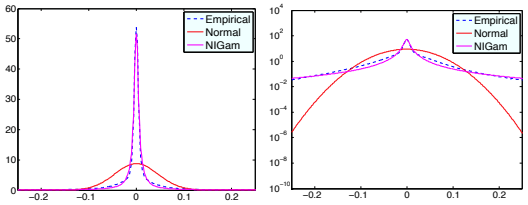

Figure 2. Comparison between normal distribution and normal-inverse-gamma distribution in modeling the residues. The left and right figures show the estimated models in linear and log scale.

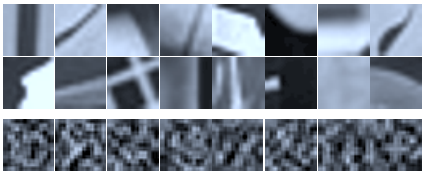

Figure 3. The first two rows show the samples from the texture patch manifold model (patch size is 13×13). The last row shows the samples from the FoEs [14] with 5×5 filter banks, which we obtained using a Gibbs sampler that runs on a 13×13 grid.

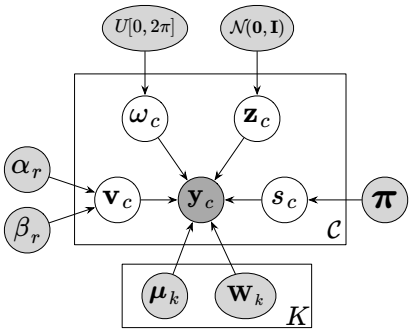

Figure 4. The graphical model for generating a patch vector $\mathbf{y}_c$, where c is a local clique in the image, and $\mathcal{C}$ is the set of all cliques.

2. Patch orientation. We model the orientation ω by a uniform distribution over $[0, 2\pi]$. The patch rotated clockwise relative to the canonical patch $\mathbf{x}$ is denoted by $R(\mathbf{x}, \omega)$.

3. Residue generation. The model allows small deviation from the manifold via a residue term. In order to select a suitable residue distribution, we fit a mixture of hyperplanes to a collection of patches from natural images and examine the marginal distribution of pixel-wise residues. Empirical analysis reveals heavy-tailed behavior. While a variety of models capture heavy-tailed behavior, here, we utilize a *normal inverse-gamma distribution*, denoted by $NIGam(\alpha_r, \beta_r)$, to model the residues. Such models can be viewed as *continuous Gaussian scale mixture*, where the variance of the Gaussian comes from an inverse-gamma distribution. Sampling $\xi \sim NIGam(\alpha_r, \beta_r)$ is as follows:

$$\sigma^2 \sim IGam(\alpha_r, \beta_r), \quad \xi \sim \mathcal{N}(0, \sigma^2). \tag{2}$$

Figure 2 shows that the inverse-gamma distribution yields much better fit to the empirical distribution. Furthermore (see section 3), the conjugacy between inverse-gamma and normal distributions with unknown variance leads to close-form updates of the residue variance in variational inference. Altogether, we obtain a graphical model to generate a patch vector $\mathbf{y}$, as in Figure 4, where $\mathbf{y}$ can be written as

$$\begin{aligned} &s \sim \boldsymbol{\pi}, \;\; \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \;\; \omega \sim U[0, 2\pi], \\ &\mathbf{y} = R(\boldsymbol{\mu}_s + \mathbf{W}_s \mathbf{z}; \omega) + \boldsymbol{\xi}, \text{ with } \boldsymbol{\xi} = (\xi^1, \ldots, \xi^{d_p}). \end{aligned} \tag{3}$$

Here, we have $\xi^j \sim \mathcal{N}(0, v^j)$ with $v^j \sim IGam(\alpha_r, \beta_r)$.

As an empirical comparison, we collect $100,000$ patches of size 13×13, and estimate both a manifold model and a field of experts model [14] over this set. Figure 3 shows the samples respectively generated from both models. Qualitatively, the proposed manifold model yields more structured local variation patterns.

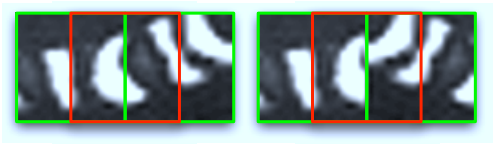

Figure 5. This figure, depicting three overlapping patches (green, red, and green from left to right), illustrates how inter-patch coherence is ensured. On the left is a detail of a natural image. By flipping the rightmost patch, we obtain the image on the right. Whereas the rightmost patch may be captured by the manifold, the innermost patch (red) has a discontinuity and as such is unlikely to be well explained by the manifold. Hence, by driving all patches towards the manifold, the MRF favors coherence across the left, middle and right patches.

2.2. Patch Coherence via Markov Random Fields

A critical element of the proposed model is to maintain coherent image structure across overlapping image patches. Simple methods such as *blending* yield noticeable artifacts when there is inconsistency between neighboring patches. Alternately, image quilting [7] addresses this issue via minimum error boundary cut. However, this requires solving a discrete optimization for all overlapping patches and is not easily captured in the context of a generative model.

The proposed framework uses a conditional MRF to enforce coherence across patches. Consider an image Y with a collection of overlapping patches, denoted by $\mathcal{C}$. For each patch $c \in \mathcal{C}$, we denote the vector of pixel values in c by $\mathbf{y}_c$. Note that $\mathbf{y}_c$ and $\mathbf{y}_{c'}$ may share part of the values when c and c' overlap. Given the patch model, we generate an image as follows. First, for each patch c, we draw $s_c \sim \boldsymbol{\pi}$ to choose a particular constituent hyperplane, and then draw the latent representation $\mathbf{z}_c \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the patch orientation $\omega_c \sim U[0, 2\pi]$. Next, we construct an MRF as below, conditioned on the configuration of local models, including

the hyperplane selectors $\mathbf{s} = (s_c)_{c\in\mathcal{C}}$, the latent representations $\mathbf{z} = (\mathbf{z}_c)_{c\in\mathcal{C}}$, and the local orientations $\boldsymbol{\omega} = (\omega_c)_{c\in\mathcal{C}}$.

$$p(Y|\mathbf{s},\mathbf{z},\boldsymbol{\omega}) \propto \exp\left(-\sum_{c\in\mathcal{C}} E_c(\mathbf{y}_c|s_c,z_c,\omega_c)\right). \quad (4)$$

For each patch c, denote the patch vector predicted by manifold model as $\bar{\mathbf{y}}_c = \boldsymbol{\mu}_{s_c} + \mathbf{W}_{s_c}\mathbf{z}_c$, and the energy term on the patch c is given by

$$E_c = \frac{1}{2}\sum_{j=1}^{d_m}(v_c^j)^{-1}\left(R(\mathbf{y}_c,-\omega_c)^{(j)} - \bar{\mathbf{y}}_c^{(j)}\right)^2. \quad (5)$$

Here, we rotate $\mathbf{y}_c$ by $-\omega_c$ to obtain $R(\mathbf{y},-\omega_c)$ rather than rotating the canonical patch $\bar{\mathbf{y}}_c$ and keeping $\mathbf{y}_c$ fixed. Though both are equivalent, the former simplifies inference, as $\bar{\mathbf{y}}_c$ involves a latent representation $\mathbf{z}$ that needs to be inferred. In addition, the variance $v_{c,j}$ is independently drawn from $IGam(\alpha_r,\beta_r)$. As the energy term is quadratic, the MRF constructed above is a Gaussian MRF. Marginalizing out the variances, we will end up with a continuous mixture of MRFs with heavy-tailed marginals on the residues. The resulting MRF drives local patches of Y towards the patch manifold, which, via clique overlapping, also encourages coherence across patches (see Figure 5).

Unlike prior work that uses MRF for image modeling, where the parameters of the random field are fixed, either through empirical validation or learning, the Gaussian MRF that we use here actually depends on the configuration of local patches. As we shall see, such a configuration would be inferred when this model is applied to image recovery, resulting in an observation-dependent MRF. The formulation enables modeling properties that are difficult to be captured by a fixed Gaussian MRF, like the heavy-tailed property.

2.3. The Joint Likelihood w.r.t. Prior

Overall, the model has the following parameters: (1) the hyperplanes of the manifold: $H_1,\ldots,H_K$ with $H_k = (\boldsymbol{\mu}_k,\mathbf{W}_k)$, (2) the prior $\boldsymbol{\pi}$ over these hyperplanes, and (3) the parameters of the residue distribution α_r and β_r. These parameters together are denoted by $\boldsymbol{\theta}$. In addition, each texture image Y is associated with several hidden variables: the hyperplane selectors $\mathbf{s}$, the latent representations $\mathbf{z}$, the orientations $\boldsymbol{\omega}$, and the residue variances $\mathbf{v}$. Given $\boldsymbol{\theta}$, the joint likelihood of Y and these hidden variables is

$$p(Y|G_Y)\prod_{c\in\mathcal{C}} p(s_c|\boldsymbol{\pi})p(\mathbf{z}_c)p(\omega_c)p(\mathbf{v}_c|\alpha_r,\beta_r). \quad (6)$$

Here, $p(s_c|\boldsymbol{\pi})$ is a categorical distribution, $p(\mathbf{z}_c)$ is a standard Gaussian distribution, $p(\omega_c)$ is a uniform distribution over $[0,2\pi]$, and $p(\mathbf{v}_c|\alpha_r,\beta_r)$ is a multivariate inverse-gamma distribution. G_Y denotes the conditional MRF to generate Y, given by Eq.(4) and Eq.(5).

2.4. Discussions

First, this model focuses on local characteristics. This is sufficient for low level vision tasks where recovery of local patterns is the main objective. Consider an image corrupted by white noises, its overall appearance structure is largely intact. Denoising such an image mainly requires prior knowledges on local textures.

Second, though assumed independent a priori, the selection of patches at different cliques are actually coupled given the observations, provided that the patches are overlapping. The inference procedure will take the information from the observed image to guide the choices of latent values, encouraging the generation of locally coherent images that have similar appearance structure as the observation.

Third, the local model over each patch is similar to a mixture of PPCA that has been employed for digit recognition and image compression [22]. The novelty here consists in the maintenance of coherence across patches via conditional MRFs, and the use of dominant orientations and heavy-tailed residue distribution. Furthermore, using manifold model to derive clique potentials distinguishes it from previous work on natural image modeling, where use of derivative filters in constructing MRF is prevalent.

3. Learning and Inference Algorithms

Based on the generative image model, we develop a learning algorithm that estimates model parameters from a given training set, and inference algorithms that apply the model to solve low-level vision problems.

3.1. Learning

Given a set of training images, denoted by $I_1,\ldots,I_n$, we decompose each image I_i into a base image B_i and a texture image Y_i via preprocessing. Specifically, for each image I_i, we obtain B_i by filtering I_i with a Gaussian kernel of large radius (25 pixels), and let $Y_i = I_i - B_i$. We then learn the GP prior for the base images using the GPML toolbox [13], and the generative model for texture images using a variational E-M algorithm as described below.

Direct maximum likelihood estimation of the model parameter $\boldsymbol{\theta}$ is intractable, as it requires integration over all hidden variables $\mathbf{s}$, $\mathbf{z}$, $\boldsymbol{\omega}$, and $\mathbf{v}$. Hence, we utilize the variational E-M algorithm. Particularly, we factorize the posterior distribution of these hidden variables into a product as $\prod_{c\in\mathcal{C}} q_c(s_c,\mathbf{z}_c,\omega_c,\mathbf{v}_c)$, where we approximate q_c by

$$q_c = \delta_{\tilde{\omega}_c}(\omega_c)\prod_{j=1}^{d_p} q_{v_c}(v_c^j|\tilde{\alpha}_c^j,\tilde{\beta}_c^j)\sum_{k=1}^{K}\tilde{\pi}_c(k)\delta_k(s_c)\delta_{\tilde{z}_{c,k}}(\mathbf{z}). \quad (7)$$

Here, $\delta_{\tilde{\omega}_c}$ is a delta-distribution that assigns probability 1 to $\tilde{\omega}_c$, q_{v_c} is an inverse-gamma distribution. In addition, we use $\tilde{\pi}_c$ to capture the posterior distribution of s_c, and given each value of s_c, we use a separate delta-distribution

to approximate the conditional distribution of $\mathbf{z}$. With this approximation, we derive the **E-steps**, as follows

$$\tilde{\pi}_c(k) \propto \boldsymbol{\pi}_k \exp(-(\|\mathbf{y}'_c, -\bar{\mathbf{y}}_{c,k}\|^2 + \|\mathbf{z}_{c,k}\|^2)/2); \quad (8)$$

$$\tilde{\mathbf{z}}_{c,k} = (\mathbf{I} + \mathbf{W}_k^T \tilde{\boldsymbol{\Lambda}}_c \mathbf{W}_k)^{-1}(\mathbf{W}_k^T \tilde{\boldsymbol{\Lambda}}_c(\mathbf{y}_c - \boldsymbol{\mu}_k)); \quad (9)$$

$$\tilde{\alpha}_c^j = \alpha_r + 1/2; \quad (10)$$

$$\tilde{\beta}_c^j = \beta_r + \sum_{k=1}^{K} \tilde{\pi}_c(k)(\mathbf{y'}_c^{(j)} - \bar{\mathbf{y}}_{c,k}^{(j)})^2/2. \quad (11)$$

Here, $\mathbf{y}'_c = R(\mathbf{y}, -\omega_c)$, $\bar{\mathbf{y}}_{c,k} \triangleq \boldsymbol{\mu}_k + \mathbf{W}_k \mathbf{z}_{c,k}$, and $\tilde{\boldsymbol{\Lambda}}_c$ is a diagonal matrix as $\mathrm{diag}(\tilde{\alpha}_c^1/\tilde{\beta}_c^1, \ldots, \tilde{\alpha}_c^{d_p}/\tilde{\beta}_c^{d_p})$. The local orientation $\tilde{\omega}_c$ can be initialized using structure tensor [4], and updated in the E-steps via gradient descent. The **M-steps**, which update the model parameters, are given by

$$\hat{\boldsymbol{\pi}}(k) = \langle \tilde{\pi}_c^{(k)} \rangle_e; \quad (12)$$

$$\hat{\boldsymbol{\mu}}_k = \langle \tilde{\pi}_c^{(k)} \tilde{\boldsymbol{\Lambda}}_c \rangle_e^{-1} \langle \tilde{\pi}_c^{(k)} \tilde{\boldsymbol{\Lambda}}_c (\mathbf{y}'_c - \hat{\mathbf{W}}_k \mathbf{z}_{c,k}) \rangle_e; \quad (13)$$

$$\hat{\mathbf{W}}_k = \langle \tilde{\pi}_c^{(k)} \tilde{\boldsymbol{\Lambda}}_c (\mathbf{y}'_c - \hat{\boldsymbol{\mu}}_k) \tilde{\mathbf{z}}_{c,k} \rangle_e \langle \tilde{\pi}_c^{(k)} \tilde{\boldsymbol{\Lambda}}_c \mathbf{z}_c \mathbf{z}_c^T \rangle_e^{-1}. \quad (14)$$

Here, $\langle \cdot \rangle_e$ indicates the empirical mean over all patches on all images. In addition, the scalar parameters α_r and β_r of the inverse gamma distribution can be obtained via MLE over the approximate distribution given by q_{v_c}. More details of the derivation are provided in the supplemental. To initialize the manifold model, we group all patches from all images by K-means into K clusters, where K is empirically set. For each cluster, we apply probabilistic PCA [23] to estimate $\boldsymbol{\mu}_k$ and $\mathbf{W}_k$. After that, we set $\boldsymbol{\pi}$ to be the relative weights of these clusters, and obtain α_r and β_r by performing MLE on the residues. This completes the initialization.

3.2. Inference

We apply the image model to solve low level vision problems, including image denoising and inpainting. Generally, an *observed image* O is given, which is assumed to be generated from an *underlying image* I by a measurement process. Inference of I can be formulated as MAP estimation:

$$\hat{I} = \mathrm{argmax}_I \; p(I|\boldsymbol{\theta}) p(O|I; \boldsymbol{\eta}). \quad (15)$$

Here, $\boldsymbol{\theta}$ is the parameter of the image prior, and $\boldsymbol{\eta}$ is the parameter of the measurement model. Different low level vision tasks have different measurement processes, which, nonetheless, can be solved with the same image model. This is one significant advantage of the generative approach.

Image Denoising. We consider a measurement process, where the image is corrupted by white Gaussian noise, as

$$O(x) = I(x) + \varepsilon_x, \quad \text{with } \varepsilon_x \sim \mathcal{N}(0, \sigma_\varepsilon^2). \quad (16)$$

Directly solving Eq.(15) involves the intractable integration over the hidden variables. Again, we resort to variational E-M, based on the mean field approximation given in (7). While the E-steps are identical to the learning algorithm, but the M-steps differ during the inference process. Here, we estimate I given the model parameters $\boldsymbol{\theta}$, while for learning, it is the other way round. Given q, the approximate posterior of the hidden variables, we have

$$\mathbb{E}_q \left[\log(Y|\tilde{\mathbf{h}}, \boldsymbol{\theta}) \right] = -\sum_{c \in \mathcal{C}} \sum_{k=1}^{K} \tilde{\pi}_c(k) \tilde{E}_{c,k}. \quad (17)$$

Here, $\tilde{\mathbf{h}}$ denotes all hidden variables. According to Eq.(5), we derive the expected energy $\tilde{E}_{c,k}$:

$$\tilde{E}_{c,k} = \frac{1}{2} \|R(\mathbf{y}_c, -\omega_c) - (\boldsymbol{\mu}_k + \mathbf{W}_k \mathbf{z}_{c,k})\|^2. \quad (18)$$

Eq.(17) and (18) together leads to a prior energy function over Y that contains only linear and quadratic terms. This is equivalent to imposing a "mean Gaussian MRF" over Y, conditioned on the approximate posterior of hidden variables, which we denote by $\tilde{G}_Y$. Therefore, the inferential M-steps maximize the following function w.r.t. Y and B:

$$p(Y|\tilde{G}_Y) p(B|G_B) p(O|Y + B), \quad (19)$$

Here, $p(B|G_B)$ is the GP-prior of the base image, and $p(O|Y + B)$ is the model given in Eq.(16). Both are Gaussian models. This reduces the problem to the inference over a Gaussian MRF, which we can readily solve.

Image Inpainting. The task of inpainting is to recover missing portions of a partially observed image. Likewise, we use variational E-M to solve this problem. Let $\mathcal{O}$ and $\mathcal{U}$ respectively denote the set of the observed and missing pixels. Given the hidden variables derived in an E-step, which leads to a mean Gaussian MRF $\tilde{G}_Y$, the M-step maximizes the following objective w.r.t. $Y(\mathcal{U})$ and $B(\mathcal{U})$, as

$$p(Y(\mathcal{U}) \cup Y(\mathcal{O})|\tilde{G}_Y) \cdot p(B(\mathcal{U}) \cup B(\mathcal{O})|G_B). \quad (20)$$

To bootstrap the E-M procedure, we initialize the missing part as follows. First, we obtain $B(\mathcal{O})$ by overly blurring the observed region, and solve $B(\mathcal{U})$ via GP-based inference conditioned on $B(\mathcal{O})$. Then we let $Y(\mathcal{O}) = I(\mathcal{O}) - B(\mathcal{O})$, and derive the missing part of Y by greedily filling in the missing pixels, from boundary towards the center. At each iteration, we pick a partially observed patch with the least missing pixels, and evaluate the marginal likelihood of the observed part w.r.t. all hyperplanes, choosing the one that yields highest value to explain that patch. Then, we set the initial values of the missing pixels using the mean that can be inferred conditioned on the chosen model. The process continues until all missing pixels are filled, which provides a reasonably good initialization.

Figure 7. The clean image underlying the inputs in Figure 6.

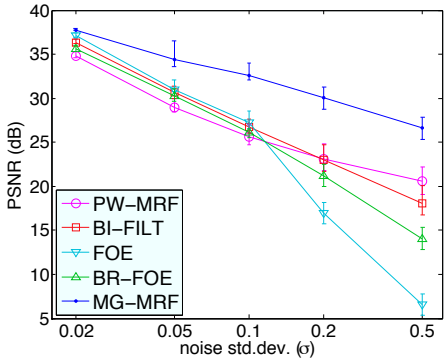

Figure 8. Each curve shows the median of the PSNR values on all testing images. The bars below and above each data point are respectively the 25% and 75% quantiles.

4. Experiments

To test the proposed model, we apply it to image denoising and inpainting. The tests are conducted on the Berkeley segmentation database, which has been widely used to assess denoising and inpainting methods [10, 14, 16, 27]. Note that we use the BSDS500 [2], a recently released extension including 200 new test images.

Training. We train the model on the training set of BSDS500, which contains 200 images. In this paper, we focus on gray-scale images. For each image I, we obtain the base image B by blurring I with a Gaussian kernel of radius 25, and let the texture image be $Y = I - B$. We estimate the parameters of the GP prior using the base images derived from the training set. To learn the texture manifold, we group all images into five categories: *nature*, *animals*, *people*, *buildings*, and *shores*, and respectively learn a patch manifold model for each category. The design parameters are set empirically to balance accuracy and model complexity. In particular, we set the number of mixture hyperplanes to $K = 160$ for each category, and fix their dimension to be $q = 12$. After the individual model for each category is learned, we combine them into a unified model, which is then used in image recovery. This separate training strategy parallelizes the training procedure and reduces memory demands.

Denoising. In this task, we consider images corrupted by additive (white) Gaussian noise. We examine the robustness of the method to a range of noise variance. We also compare the proposed method (MG-MRF) with four other methods on image denoising, which include the classic pairwise MRF (PW-MRF), bilateral filtering (BI-FILT) [12], field of experts (FOE) [14], and Weiss's variant of FoE (BR-FOE) [27]. When using MG-MRF for denoising, the MRFs are built upon overlapping patches of size 13×13 with 3-pixel interval. Under this setting, each pixel is covered by 16 to 20 patches, which provides a balance between coherence, robustness, and computational efficiency. The inference algorithm takes 5 to 30 iterations to converge. In general, more iterations are required under higher noise levels. We implement the algorithms for PW-MRF and BI-FILT, and use the code published by the authors of the corresponding papers for FOE and BR-FOE. Here, the FoE model is constructed with 5×5 cliques and 24 filters. We seek the best settings of design parameters via cross validation for all comparison methods, and evaluate the performance in terms of peak signal-to-noise ratio (PSNR) in dB.

Figure 6 shows the denoising results obtained on a test image. The corresponding uncorrupted image are shown in Figure 7. Generally, when the noise is moderate ($\sigma = 0.1$), PW-MRF, as expected, tends to slightly blur edges; while other methods preserve edge sharpness. Close examination reveals that the image generated by MG-MRF is qualitatively better than the others. As the noise level increases, MG-MRF continues to perform robustly except for minor blurring of boundaries between different texture patterns; while other methods degrade noticeably. Interestingly, when $\sigma = 0.5$, PW-MRF performs significantly better than both FOE and BR-FOE. This observation is consistent with the dependence of FoE methods on derivative filter responses, which are sensitive to high noise levels.

Figure 8 summarizes the performance statistics obtained over the images in the test set, under different noise conditions (*i.e.* $\sigma_\varepsilon = 0.02, 0.05, 0.1, 0.2$ and 0.5). In general, the methods based on pairwise links (PW-MRF and BI-FILT) degrade more gracefully than the FoE-based methods (FOE and BR-FOE) as the noise level increases. MG-MRF consistently outperforms other methods. The experimental results demonstrate that MG-MRF is superior to other methods in two aspects: preservation of texture details and robustness to high noise levels. This is a consequence of its distinctive mechanism in which the oriented templates derived from the learned patch manifold generate local patterns, and are combined with an MRF to ensure coherence between them. This is in contrast to prior methods using MRFs which impose coherence at the pixel level. When the noise variance is large, the direct influence of the observed pixel values becomes insignificant. The inference algorithm uses the observed image mainly for choosing templates from the manifold. Note that each choice is conditioned on all 169 pixels in a patch, making it much more

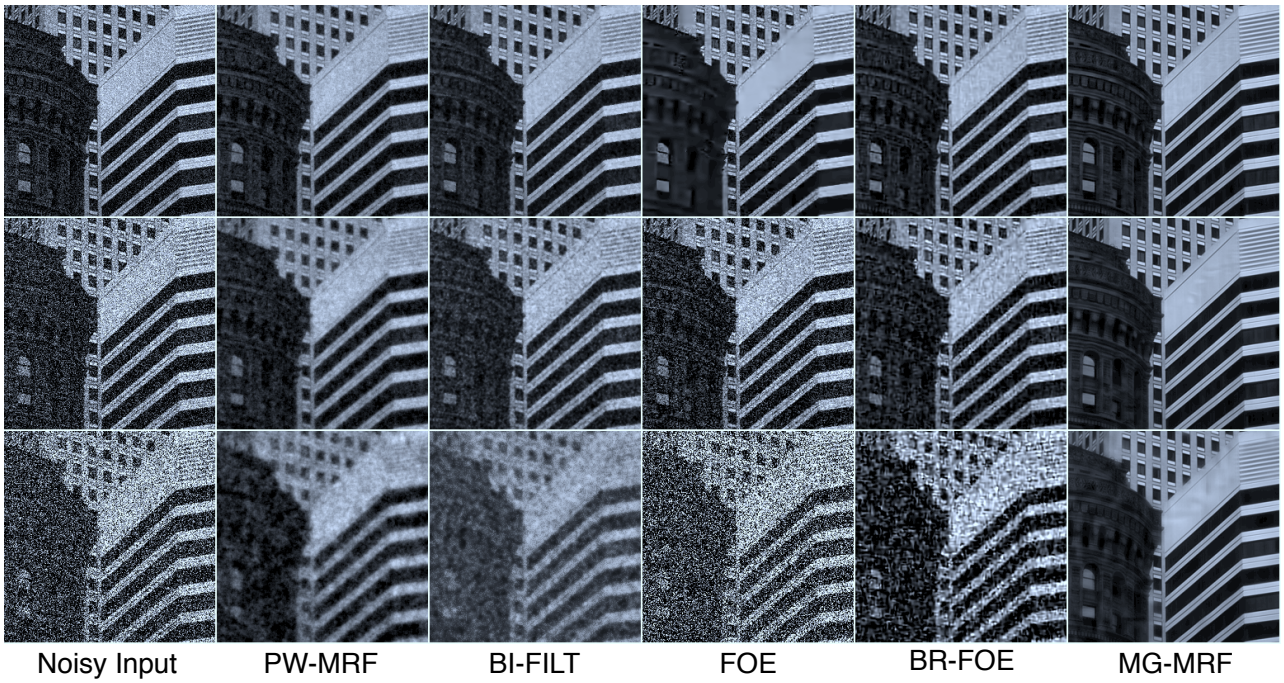

Figure 6. The input noisy images (the first column) with the recovered images obtained with different methods. Only part of the images are shown to highlight the differences between methods (see the full clean image in Figure 7). The inputs at different rows are subject to different levels of noise ($\sigma = 0.1, 0.2, 0.5$).

Figure 9. The results of inpainting on a partially observed image with a mask of width at 10 pixels.

robust than the methods that rely on a much smaller neighborhoods. The Bayesian formulation utilizing a distribution of models instead of a single model also contributes to the reliability.

Inpainting. This task is to infer the missing part given a partially observed image. To test the algorithm under different conditions, we generate occlusion masks of different widths. Specifically, we draw a free-form curve as a skeleton, and dilate it to a specific width to generate the mask.

For inpainting, we compare our method with two other MRF-based approaches: the FoE-based method [14] and TV-MRF regularized recovery. The number of iterations needed to recover an image increases as the width of masking curve increases. Figure 9 shows results for an example image. The results yielded by both FOE and TV-MRF contain noticeable artifacts, especially at places where the masking curve passes through complex patterns, while MG-

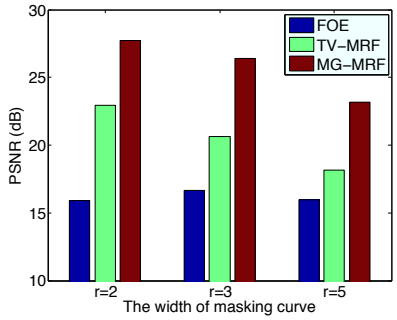

Figure 10. The PSNR of inpainting results within masked region.

MRF performs better in recovering such patterns, as they are effectively captured by the texture manifold. We also perform quantitative evaluation, in terms of PSNR within the masked region. The results shown in Figure 10 show that MG-MRF works better than the comparison methods for all three different mask widths.

Efficiency. The variational EM algorithm requires 5 to 30 iterations to converge, depending on the difficulty of the task, *e.g*. the noise variance. With our MATLAB implementation, each iteration on an image in the Berkeley dataset, whose size is 481×321 pixels, takes about 10 seconds.

5. Conclusion

We developed a generative image model for low level vision, which incorporates a patch manifold to model the local texture patterns, and a conditional MRF to ensure coherence between patches. With a mean field approximation, we derived efficient algorithms for both learning and inference, which we apply to image denoising and inpainting. The experimental results demonstrate that our method performs substantially better than other methods in recovering complex texture patterns, and shows superior robustness against severe noise corruption. Such improvement is ascribed to the patch model that is more effective than an MRF based on derivative filters in capturing local structures, as well as the Bayesian approach that adaptively combines the MRF predictions in posterior inference.

Acknowledgement

This research was partially supported by the Office of Naval Research Multidisciplinary Research Initiative (MURI) program, award N000141110688.

References

[1] S. Ali and M. Shah. A supervised learning framework for generic object detection in images. In *Proc. of ICCV'05*, 2005. 1

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Countour detection and hierarchical image segmentation. *IEEE Trans. on PAMI*, 33(5), 2011. 6

[3] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on PAMI*, 19(7), 1997. 1

[4] J. Bigun and G. Granlund. Optimal orientation detection of linear symmetry. In *Proc. of 1st ICCV*, 1987. 2, 5

[5] M. Black and A. Jepson. EigenTracking : Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *International Journal of Computer Vision*, 26(1):63–84, 1998. 1

[6] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001. 1

[7] A. Efros and W. Freeman. Image quilting for texture synthesis and transfer. In *SIGGraph'01*, 2001. 3

[8] W. Freeman, E. Pasztor, and O. Carmicheal. Learning Low-level Vision. *International Journal of Computer Vision*, 40(1):25–47, 2000. 1

[9] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. PAMI*, 14, 1992. 1

[10] V. Jain and H. Seung. Natural image denoising with convolutional networks. In *Adv. of NIPS'08*, 2008. 6

[11] F. D. la Torre and M. J. Black. Robust Principal Component Analysis for Computer Vision. In *Proc. of CVPR'01*, volume 1, 2001. 1

[12] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand. Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision*, 4(1), 2008. 6

[13] C. E. Rasmussen and H. Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11, 2010. 4

[14] S. Roth and M. J. Black. Fields of Experts: A Framework for Learning Image Priors. In *Proc. of CVPR'05*, 2005. 1, 3, 6, 7

[15] S. Roth and M. J. Black. On the spatial statistics of optical flow. In *Proc. of ICCV'05*, 2005. 1

[16] S. Roth and M. J. Black. Steerable Random Fields. In *Proc. of ICCV'07*, 2007. 1, 6

[17] K. G. Samuel and M. Tappen. Learning optimized map estimates in continuously-valued mrf models. In *Proc. of CVPR'09*, 2009. 1

[18] U. Schmidt, Q. Gao, and S. Roth. A Generative Perspective on MRFs in Low-Level Vision. In *Proc. of CVPR*, 2010. 1

[19] M. Tanaka and M. Okutomi. Locally Adaptive Learning for Translation-Variant MRF Image Priors. In *Proc. of CVPR'08*, 2008. 1

[20] M. F. Tappen. Utilizing Variational Optimization to Learn Markov Random Fields. In *Proc. of ICCV'07*, 2007. 1

[21] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman. Learning Gaussian Conditional Random Fields for Low-Level Vision. In *Proc. of CVPR'07*, 2007. 1

[22] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analysis. *Neural Comp.*, 11(2), 1999. 4

[23] M. Tipping and C. Bishop. Probabilistic principal component analysis. *J. R. Statist. Soc. B*, 61, 1999. 5

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

[25] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proc. of ECCV'02*, pages 447–460, 2002. 1

[26] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *IEEE Trans. on PAMI*, 27(12), 2005. 1

[27] Y. Weiss and W. T. Freeman. What Makes a Good Model of Natural Images? In *Proc. of CVPR'07*, 2007. 1, 6

[28] S.-C. Zhu, Y.-N. Wu, and D. Mumford. Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998. 1