Image-Specific Prior Adaptation for Denoising

Xin Lu, Zhe Lin, Hailin Jin, Jianchao Yang, James. Z. Wang

***Abstract*—Image priors are essential to many image restoration applications, including denoising, deblurring, and inpainting. Existing methods use either priors from the given image (internal) or priors from a separate collection of images (external). We find through statistical analyses that unifying the internal and external patch priors may yield a better patch prior. We propose a novel prior learning algorithm that combines the strength of both internal and external priors. In particular, we first learn a generic Gaussian Mixture Model from a collection of training images and then adapt the model to the given image by simultaneously adding additional components and refining the component parameters. We apply this image-specific prior to image denoising. Experimental results show that our approach yields better or competitive denoising results in terms of both the peak signal-to-noise ratio and structural similarity.**

***Index Terms*—Image denoising, Internal and external denoising, Online-GMM, Patch-based denoising.**

I. Introduction

Noise is a fundamental problem in measuring light. No matter how good the sensors are, there is noise in images, especially in low-light conditions. Image denoising is the problem of reducing undesired noise in images. It has been studied extensively over the last half century because of its practical importance. The problem is mathematically ill-posed and image priors are used to regularize it such that meaningful solutions exist.

There are two kinds of image priors one can use: priors learned from the given image and priors learned from a separate set of images. We follow the common naming convention and refer to the former as internal priors and the latter as external priors (or generic priors). Smoothness and piecewise smoothness are probably the simplest form of internal priors. They led to many successes of PDE-based denoising algorithms in the nineties. Recently, more interesting internal priors, such as patch self-similarity, have been proposed. They have led to methods such as BM3D [1] which is still regarded as one of the state-of-the-art methods in image denoising. The obvious limitation of internal priors is that external images are completely ignored. For instance, for any given image, one can find similar images (at least in terms of parts) that contain significantly less noise and use these similar images to do a better job of denoising. It was a breakthrough in image denoising in going beyond internal priors to external priors. Allowing external images opens a wide range of possible priors. Popular priors that one learns from external images include sparse representations and nonlinear regression functions between clean and noisy patch pairs. One of the top-performed image denoising algorithms is based on external images [2]. There is one problem with external priors: It is well known that images form a heavy-tail distribution. No matter how large the external image set is, some images will fall in the heavy tail, i.e., will not be well-modeled by the learned priors. We believe this problem prevents methods that use external images from significantly outperforming those that do not.

There exists a natural question which is if there is a way to combine internal and external priors. It turns out that we are not the first to ask this question. For instance, [3] and [4] explored similar problems. They discovered that internal and external priors are good for different types of image patches and proposed methods to combine them. Strictly speaking, both [3] and [4] focus on combining internal and external denoising algorithms rather than internal and external priors. Instead, in this paper, we focus on priors rather than specific denoising algorithms. It is our belief that in addition to mathematical elegance, priors are more fundamental than specific algorithms. One can use different denoising algorithms with the same priors. The main idea in this work is that we can learn generic priors from external images and adapt them to a specific image using internal priors learned from that image. We overcome the heavy-tail problem in external priors by adapting generic priors to a specific image.

A. Related Work

State-of-the-art image denoising approaches leverage various types of patch priors for regularizing the ill-posed nature of the problem. In general, the priors can be categorized into internal patch priors and external patch priors.

Internal patch priors refer to patch statistics derived from the image itself. Typical examples include patch self-similarity [5], [1], sparsity prior [6], structural similarity [7], and patch recurrence across image scales [8]. The local patch self-similarity has been quite successful for denoising due to its effectiveness and simplicity. Non-Local Means [5] denoises each pixel of an image based on the weighted average of central pixels of similar patches. BM3D [1] exploits both patch self-similarity and 3d transform domain collaborative filtering, which achieves state-of-the-art performance in denoising. Mairal *et al.* [6] further advanced the patch similarity idea with sparse coding priors. To use patch recurrence across image scales, Zontak *et al.* [8] attempted to find clean versions of noisy patches by searching similar patches across multiple image scales. To leverage structural similarity, Dong *et al.* [8] examined the structural information of the entire image and considered the joint sparsity for noise removal.

External patch priors refer to patch statistics or denoising operators learned from external image sets, such as statistical distribution of image patches in natural images [9], sparse

X. Lu and J. Wang are with College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA, 16802, USA. E-mail: {xinlu,jwang}@psu.edu

Z. Lin, H.L. Jin, and J.C. Yang are with Adobe Research, Adobe Systems Inc., San Jose, CA, 95110, USA.
E-mail: {zlin,hljin,jiayang}@adobe.com

representation [10], global statistical prior [11], and regression functions from noisy patches to clean patches [2], [12], [13]. Roth and Black [11] used Markov Random Field to learn generic image priors. In [10], a dictionary learning-based method is introduced for compact patch representation, whereas in [9], a Gaussian Mixture Model (GMM) is learned from external patch databases and used as a prior for denoising. More recent work favors mapping functions learned through neural networks. Burger *et al.* [2] employed a plain multi-layer perceptron to learn the mapping functions between pairs of noisy patches and noise-free patches. Cho [12] applied the Boltzmann machines to map noisy images to clean images. In the same vein, Xie *et al.* [13] combined sparse coding and deep networks pre-trained with denoising auto-encoder for the training scheme to learn external priors.

Although internal and external priors have been widely used in previous work, little effort has been devoted to combining them in a principled way. In this work, we propose a novel, unified prior model combining internal and external priors. Most recent works, such as Mosseri *et al.* [4] and Burger *et al.* [3], are related to ours in that they also leverage both internal and external priors. Mosseri *et al.* [4] developed a Patch-SNR measure to decide whether a noisy patch is denoised using internal priors or external priors. Burger *et al.* [3] attempted to learn a non-linear regression function that can map two denoising results, one with the internal prior and the other with the external prior, to produce a better denoising result. Unlike these methods where the denoising is conducted separately with either the internal or external priors on image patches, we introduce a unified prior which is fundamentally different from those methods. The prior is general and thus can be applied to any image restoration application. Specifically, we derive the prior in the context of denoising. During the offline phase, a generic patch prior is learned from an external set of natural images; and during the online phase, we adapt the generic patch distribution to the test image by analyzing patch statistics in the test image. Our work is also related to EPLL [9] in that both use GMM to represent patch distributions. The difference is that our method adapts the generic prior to the test image while EPLL relies on the fixed prior model, additionally, our method does not require the reconstruction term and computationally expensive iterations as in EPLL.

The approach proposed by Wang and Morel [14] is very close to ours; they also adapt the generic prior, which is a mixture model, to the test images. The difference is that their approach conducts an "in-place" modification to the prior (i.e., components of the mixture model are shifted and deformed according to image-specific information) while ours augments the prior model by image-specific component addition (see Section III.A for details). Lebrun *et al.* [15] proposes a patch-based denoising method where a noisy patch is restored by nearby similar patches assuming from a Gaussian model.

B. Contributions

The main contributions of the paper are the following: 1. We propose a unified algorithm that learns a generic prior from external images and adapts the prior to a specific image. This is fundamentally different from combining different denoising algorithms as done in [3], [4]. 2. We show that by adapting a generic prior into an image-specific one, we do not need global image reconstruction[1] as done in [9]. This significantly speeds up the algorithm.

II. Modeling Patch Statistics using Gaussian Mixture Model

This section introduces learning patch prior using Gaussian Mixture Model. We learn the GMM model using an external patch database (external patch statistics) and using patches in one image (internal patch statistics), respectively. We empirically analyze the internal and external GMM patch priors, and then discuss ideas of leveraging both priors into a unified natural image patch prior.

A. Gaussian Mixture Model

We learn finite Gaussian Mixture Model (GMM) over natural image patches $\{x_i\}$ as patch priors. Using the GMM model, the log likelihood of a given patch x_i is:

$$p(x_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \tag{1}$$

where x_i is a D-dimensional vector, K is the total number of mixture components chosen for the GMM, N is the total number of patches in the training set, and the GMM model is parameterized by mean vectors $\{\mu_k\}$, covariance matrices $\{\Sigma_k\}$, and mixture weights of mixture components $\{\pi_k\}$. We collectively represent these parameters by $\Theta = \{\mu_k, \Sigma_k, \pi_k\}_{k=1}^{K}$, and Θ is learned using Expectation Maximization algorithm (EM).

In the **E-Step**, we calculate the posterior probability for the component k as:

$$\Pr(k|x_i, \Theta) = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}, \tag{2}$$

$$n_k = \sum_{i=1}^{N} Pr(k|x_i, \Theta). \tag{3}$$

In the **M-Step**, we update the model parameters as follows:

$$\pi_k = n_k/N \tag{4}$$

$$\mu_k = \frac{\sum_{i=1}^{N} \pi_k x_i}{\sum_{i=1}^{N} \pi_k} \tag{5}$$

$$\Sigma_k = \frac{\sum_{i=1}^{N} \Pr(k|x_i, \Theta) x_i x_i^t}{n_k} \tag{6}$$

We iterate over the E-Step and M-Step until convergence.

B. Building External and Internal GMM

[1]The "global image reconstruction" refers to using a cost function on overlapping patches extracted from the noisy image for image denoise, rather than merely applying denoising operators on each individual noisy patch, e.g., the algorithm presented in [9], Section 3.1.

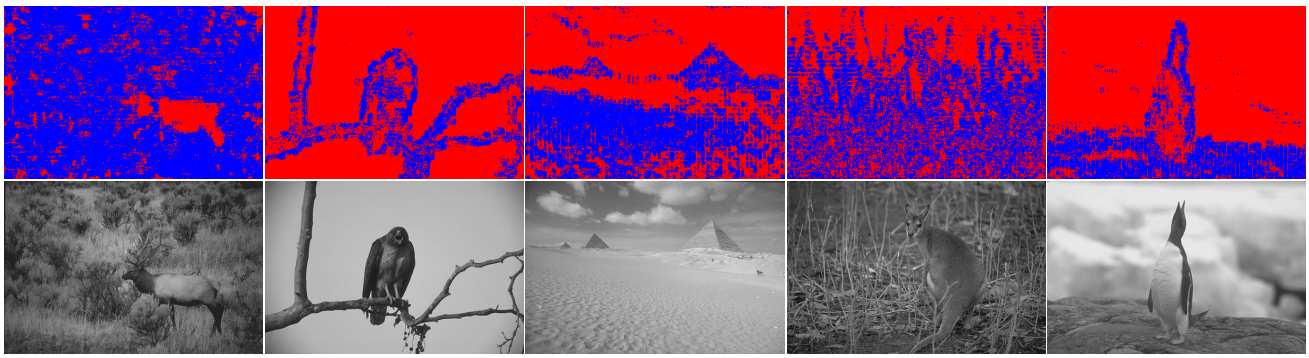


Fig. 1. Internal and external patch statistics. Given an image patch, the posterior probability of patches are computed using both the internal and external GMM patch priors. A blue pixel indicates the patch that centered at the pixel has low probability in the external GMM model; a red pixel indicates the patch has low probability in the internal GMM model. The figure is better viewed in color.

1) External/Generic GMM: We built a generic, external GMM (denoted as GMM_{ext} with parameters Θ_{ext}) following the settings in [9]. We used the same image collection as used in [9], i.e., the BSD training dataset[2] (200 images in total). We densely sampled $50K$ 8×8 zero-mean patches for GMM model training as discussed in Section II-A[3]. We set $K = 200$ for the external GMM model learning, and all images are converted into gray scale in this work.

2) Internal GMM: We took the BSD test set (100 images) for the analysis purpose, and none of the images in the BSD test set was used for external GMM training. We built an internal GMM model of each individual image in the BSD test set. For each image, we extracted all 8×8 overlapping patches, generated zero-mean patches, and trained the GMM model as discussed in Section II-A. We let all internal GMM models (denoted as $\{\text{GMM}_{\text{int}}\}$ with parameters $\{\Theta_{\text{int}}\}$) to have the same number of components with the external GMM model, i.e., $K = 200$.

C. Internal Statistics vs. External Statistics

For each image in the BSD test set, we extracted all 8×8 overlapping patches. For each patch x, we computed $Pr_{\text{ext}} = \max_k Pr(k|x, \Theta_{\text{ext}})$ and $Pr_{\text{int}} = \max_k Pr(k|x, \Theta_{\text{int}})$ presented in Equation 3. For some patches that have low probabilities in the GMM_{ext} (i.e., these patches are not very frequent across all images) but higher probabilities in GMM_{int} (i.e., many self-similar examples), we might be able to model them better. There are some patches that cannot be modeled well with either GMM_{ext} or GMM_{int} (i.e., not self-repeating patterns). We present example images in Figure 1, where each pixel refers to a patch's center pixel ($(4, 4)$ in a 8×8 patch). We annotate the pixel as blue if the corresponding patch has a low probability in GMM_{ext}, and otherwise we annotate the pixel in red. These results indicate that we may get a better patch prior by unifying the GMM_{int} and GMM_{ext} priors.

[2]http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

[3]We represent a patch by a D-1 vector, and zero-mean patches refers to the mean-subtracted patches.

III. Denoising with Image-Specific Prior

Based on our analysis of the patch statistics in Section II, we propose a novel method to learn image-specific prior by unifying internal and external GMM image patch priors. Figure 2 gives an overview of our approach. We first train a generic GMM model on a collection of patches randomly sampled from a set of clean natural images (as presented in Section II-B), which serves as our generic GMM patch prior. The training is off-line and only needs to be performed once. Given a noisy image, we conduct a two-step adaptation to generate an image-specific patch prior inspired by the Kolmogorov-Smirno (KS) Test. The first step consists in constructing new Gaussian components for individual or image-specific patches that the generic GMM prior fails to model, and the second step performs GMM adaptation to better fit the distribution of both generic patches and image-specific patches. Equipped with the image-specific prior, we can conduct more effective denoising on the input. In the following, we first detail the two steps for learning an image-specific prior for a clean image, and then we describe how to adapt the procedure to noisy inputs for denoising.

A. Image-Specific Components Addition

A generic GMM may not be able to model every image patch well, considering the large space of natural image patches (e.g., 8×8 patches). The image-specific patches of a given image will be outliers in the view of a generic GMM prior, and thus we need to construct new components to model these outlier patches. This process is essentially similar to the online clustering or online GMM in the statistics literature. We build the new Gaussian components and add them through online GMM. Specifically, given an image represented by a collection of overlapping patches $Y = \{y_i\}_{i=1}^{M}$, where M is the total number of patches in the image. We first generate an image-specific(internal) GMM model as presented in Section II-B, where we learn parameters $\Theta_y = \{\mu_{in}^k, \Sigma_{in}^k, \pi_{in}^k\}_{k=1}^{K_y}$, where K_y is the total number of components. We let $K_y = 200$, the same number of Gaussian components as in the generic GMM. To accelerate the process, we first do a k-means clustering on the internal patches and do one EM iteration to generate a

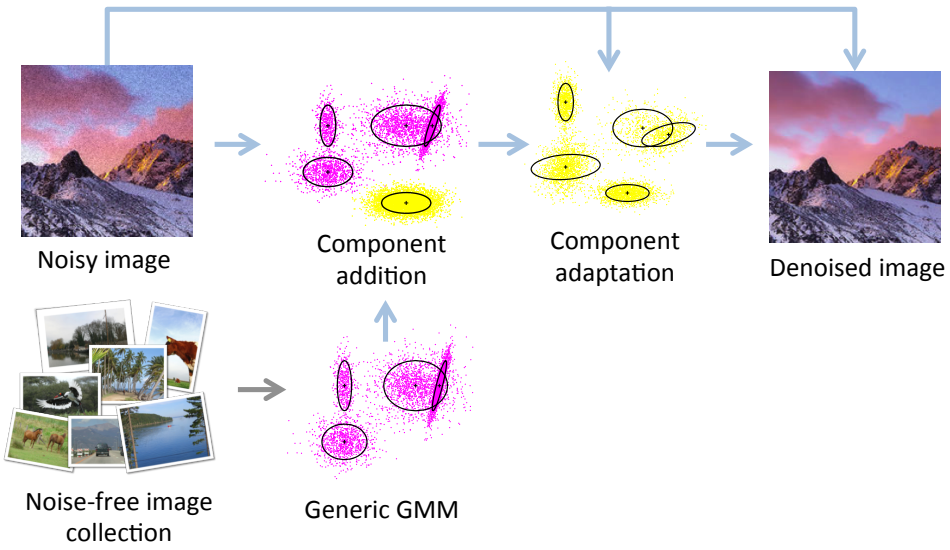


Fig. 2. Approach overview. We first trained a generic GMM patch prior using a collection of natural image patches as introduced in Section II. We represent the generic GMM model in magenta. Given a noisy image, we conduct a two-step adaptation to generate an image-specific patch prior. In the component addition, new Gaussian components (in yellow) are constructed for individual or image-specific patches that the generic GMM prior fails to model. In component adaptation, GMM adaptation is performed to better fit the distribution of both generic and image-specific patches of the given image. Equipped with the image-specific prior, we can conduct more effective denoising on the input.

coarse image-specific GMM model. Given the generic GMM model $\Theta_g = \{\mu_g^k, \Sigma_g^k, \pi_g^k\}_{k=1}^{K_g}$, we discover representative components and add them to the generic GMM prior following Algorithm 1, where $D_{KL}(\Theta_y^k, \Theta_g^k)$ is the Kullback-Leibler (KL) divergence[4] defined as follows:

$$\begin{aligned} & D_{KL}(\Theta_y^k, \Theta_g^k) \\ & = \int [\log(\Theta_y^k(t)) - \log(\Theta_g^k(t))]\Theta_y^k(t)dt \\ & = \frac{1}{2}[\log \frac{|\Sigma_y^k|}{|\Sigma_g^k|} - d + tr((\Sigma_y^k)^{-1}\Sigma_g^k) \\ & + (\mu_y^k - \mu_g^k)^T(\Sigma_y^k)^{-1}(\mu_y^k - \mu_g^k)] \end{aligned} \tag{7}$$

where d is the dimension of the patch vector. δ is the threshold for detecting new Gaussian components: if the divergence between the current component and the nearest generic component is larger than δ, a new component is added. We empirically determine the value of δ for denoising in the experimental section. Note that the image-specific GMM Θ_y is coarsely constructed for efficiency. We discuss how to refine the new GMM to fit the patch distribution in the following subsection.

Algorithm 1: Component Addition

Input: Generic GMM model with parameters of Θ_g , a coarse image-specific GMM model Θ_{x^y}

$\Theta_{output} = \Theta_g$, $K_{mod} = K_g$
for $\Theta_y^k \in \Theta_{x^y}$, $k \in [1, K_y]$
  $flag = true$
  for $\Theta_g^k \in \Theta_g$, $k \in [1, K_g]$
    if $D_{KL}(\Theta_y^k, \Theta_g^k) < \delta$
      $flag = false$; break;
    end if
  end for
  if $flag == true$
    $\Theta_{output}^{K_{mod}+1} = \Theta_y^k$
    $K_{mod} = K_{mod} + 1$
  end if
end for

B. Gaussian Mixture Model Adaptation

Equipped with the added new Gaussian components capturing the image-specific patches, we are ready to adapt our complete image specific GMM to the patch distribution of the given image. To avoid notation clutter, we use $\Theta_0 = \{\mu_k, \Sigma_k, \pi_k\}_{k=1}^{K}$ (K is the number of mixtures) to denote the current GMM after component addition. Given the patches extracted from the input image $Y = \{y_i\}_{i=1...M}$, we want to estimate the image-specific GMM parameters $\hat{\Theta} = \{\hat{\mu}_k, \hat{\Sigma}_k, \hat{\pi}_k\}_{k=1}^{K}$, which can be solved via maximizing a posteriori (MAP) estimation:

$$\begin{aligned} \hat{\Theta}_{MAP} &= \arg\max_{\Theta} p(\Theta|Y) \\ &= \arg\max_{\Theta} f(Y|\Theta)g(\Theta) \end{aligned} \tag{8}$$

where $g(\Theta)$ denotes the prior information of the unknown GMM parameters.

The adaptive GMM model is derived by following the GMM

[4]While the Kullback-Leibler (KL) divergence is not symmetrical, the same generic GMM is used for all images. In our implementation, we computed the KL divergence on pairs of an internal GMM component and an external GMM component, where we fixed the order of internal and external GMM components in each of the pairs. We thus avoid the problem caused by the unsymmetrical KL divergence.

adaptation algorithm proposed in [17]. In the **E-Step**, we calculate the posterior probability for the component k as:

$$\Pr(k|y_i,\Theta) = \frac{\pi_k \mathcal{N}(y_i|\mu_k,\Sigma_k)}{\sum_{k=1}^{K}\pi_k \mathcal{N}(y_i|\mu_k,\Sigma_k)}, \quad (9)$$

$$n_k = \sum_{i=1}^{N} Pr(k|y_i,\Theta). \quad (10)$$

In the **M-Step**, we update the model parameters as follows:

$$\hat{\pi}_k = [\alpha^{\pi} n_k/N + (1-\alpha^{\pi})\pi_k]\gamma \quad (11)$$

$$\hat{\mu}_k = \alpha^{\mu}\frac{\sum_{i=1}^{N}\pi_k y_i}{\sum_{i=1}^{N}\pi_k} + (1-\alpha^{\mu})\mu_k \quad (12)$$

$$\hat{\Sigma}_k = \alpha^{\Sigma}\frac{\sum_{i=1}^{N}\Pr(k|y_i,\Theta) y_i y_i^t}{n_k} + (1-\alpha^{\Sigma})\Sigma_k \quad (13)$$

where γ is the scale factor to ensure $\sum_{k=1}^{K}\hat{\pi}_k = 1$. We iterate over the E-Step and M-Step until convergence. In practice, we find that it is good enough to just run one iteration of EM by setting the parameter adaptation rates α^{π}, α^{μ} and α^{Σ} to be 1, which means we fully trust the new input patches. We detail the parameter selection of α^{π}, α^{μ} and α^{Σ} in the experimental results.

C. Learning Image-Specific Prior for Denoising

The above image-specific prior learning assumes clean input images. To learn an image-specific prior from noisy images for denoising, we have two options for the model construction:

1) Assuming zero mean for the noise, the model parameters affected are mainly the covariance matrices. Also assuming a diagonal covariance matrix Σ_n for the noise, we can estimate the coarse image-specific GMM parameters with $\Sigma_{in}^k - \Sigma_n$ for component addition[5]. For GMM adaptation, we can similarly subtract Σ_n from the estimation in Equation 13. We found that this scheme works well for small noise cases (noise with low variance).
2) We first apply the generic GMM prior for denoising to get an initial result, from which we can learn an image-specific prior, which is more accurate for the final denoising. In practice, we find this scheme works well for strong noise cases.

After we have learned an image-specific prior for the noisy input image, we first extracted overlapping noisy patches from the input image. For each noisy patch y, the Bayes Least Square solution of its denoised version $\hat{y}$ is $\hat{y} = E[x|y] = \sum_{k=1}^{K}[Pr(k|y)E[x|y,\Theta_k]]$, where Θ_k represents the parameters of k-th component of the mixture. Considering the large number of overlapping patches in an image (more than $60K$ patches in a 256×256 image), this solution is costly if we utilize all Gaussian components to denoise each patch. In practice, we took an approximate solution as presented in [9] and found that this simple solution worked well in general. We discuss this approximation in the experimental section.

Specifically, for each noisy patch y, we select the Gaussian component with the maximum posterior probability:

$$k^* = \arg\max_k Pr(k|y,\Theta). \quad (14)$$

We then apply the Wiener filter for noise removal as discussed in [9]:

$$\hat{y} = (\hat{\Sigma}_k + \Sigma_n)^{-1}(\hat{\Sigma}_k y + \Sigma_n \hat{\mu}_k) \quad (15)$$

where $\Sigma_n = \sigma^2\mathbf{I}$. With all overlapping denoised $\hat{y}$, we generate the final denoised image by averaging the overlapping pixels.

D. Discussions

There are alternative ways of learning an image-specific patch prior. One could be to adapt the generic GMM to the given image without constructing any new components (adaptive GMM), and the other is to directly learn a GMM based on the internal examples only (internal GMM). Compared with these two methods, our method is superior in terms of speed and potentially performance:

1) As we mentioned before, the image-specific patches are outliers in the view of the generic GMM. As a result, adapting the generic GMM takes more iterations in order to model those outlier patches well.
2) Building an internal GMM based on the internal examples alone is prone to overfitting to noise. It is also very slow for the algorithm to converge to a reasonably good model for denoising. Note that in our component addition step, we also train a coarse image-specific GMM for finding new components based on fast kmeans for efficiency. The coarse image-specific GMM is different from the internal GMM, which needs to be well trained and thus is much slower.

We demonstrate the superior performance of image-specific prior generated by the online GMM approach compared to the internal GMM and adaptive GMM approaches in the experimental results.

IV. Experimental Results

We evaluate the effectiveness of our approach by comparing the image-specific prior generated by the proposed approach with priors generated by internal GMM and adaptive GMM (discussed in Section III-C) as well as state-of-the-art denoising approaches such as BM3D [1], EPLL [9], MLP [2], NLB [15], SPLE [14], combined internal and external method (CBIE)[4][6], and combined BM3D and MLP (CBBM)[3][7]. Our approach achieves better or competitive image denoising results in terms of PSNR (i.e., Peak Signal-to-Noise Ratio) and SSIM [18] (i.e., Structural Similarity). In the following, we refer to our approach as "OL-GMM", internal GMM as "ING-GMM", and adaptive GMM as "ADA-GMM". We also use the

[5]Through out the paper, we assume the noise is additive and zero-mean and the variance of signal is far larger than the variance of noise to apply $\Sigma_{in}^k - \Sigma_n$. Practically, to avoid singularity of covariance matrix in GMM computation, we add very small value (e.g., 1e-5) to its diagonal elements.

[6]The results of [4] are based on our implementation of [4]. NLM[5] is used for both internal denoising and external denoising.

[7]The results of BM3D [1], EPLL [9], MLP [2], NLB [15], SPLE [14], and CBBM[3] are based on authors' source codes released on the Web.

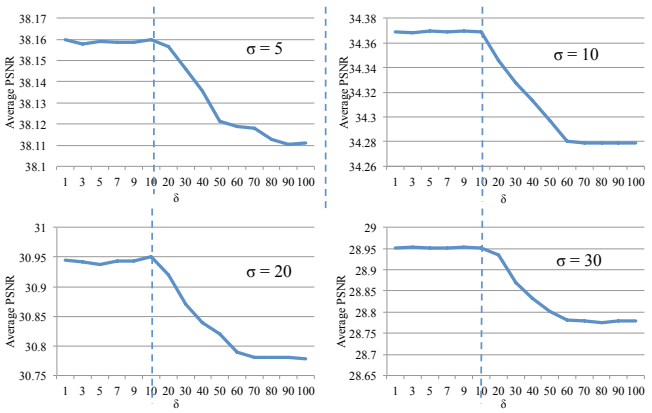


Fig. 3. Denoising Results (average PSNR) on standard images with different δ. The x-axis denotes δ, and the y-axis refers to average PSNR.

global reconstruction term as in [9] to refine the performance of OL-GMM, and we denoted it as "OL-GMM-RT". We made a single EM iteration for OL-GMM and ADA-GMM. In OL-GMM-RT, we iterate once using the reconstruction term defined in [9] with the OL-GMM result.

To evaluate our performance quantitatively, we corrupted clean images with the white Gaussian noise and used the same noisy test images to generate the results of all the methods for fair comparison. All our experiments reported in this paper are conducted on gray-scale images.

The training patches used in our experiments were sampled from the Berkeley segmentation dataset (training images)[8]. The performance of our approach is evaluated on the four benchmark datasets including standard test images [1], Berkeley segmentation dataset (testing images)[8], Pascal VOC2007 dataset[9], and McGill dataset[10].

A. Evaluation on the standard test images

We empirically determined the value of δ for image denoising when noise is of different variance. We evaluated average PSNR on the standard images as δ changes, and we present results in Figure 3, where the x-axis denotes δ, and the y-axis refers to average PSNR. As shown in the figure, the PSNR drops when δ is larger than 10 for all the four noise levels. In particular, we found that setting δ to 1 provided the highest PSNR for denoising when the σ is 5 or 10, and setting δ to 10 provided the highest PSNR for denoising when σ is 20 or 30. We thus set δ as 1 for all the remaining evaluation when σ is 5 or 10 and set δ as 10 when σ is 20 or 30.

Similarly, we empirically determined the value of α (i.e., α^{π}, α^{μ} and α^{Σ}). We evaluated denoising results in terms of PSNR on standard images by setting α as 0.2, 0.4, 0.6, 0.8, and 1, respectively. In all these experiments, we ran one iteration of EM for GMM adaptation, and present results in Table I. As shown in the table, for all the four noise levels, the highest PSNR is achieved when α is 1. We thus set α as 1 for all the remaining evaluation.

[8]http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/
[9]http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
[10]http://pirsquared.org/research/mcgilldb/

TABLE I
AVERAGE PSNR ON STANDARD IMAGES WITH DIFFERENT α

	0.2	0.4	0.6	0.8	1.0
$\sigma = 5$	38.07	38.08	38.09	38.12	**38.16**
$\sigma = 10$	34.34	34.35	34.36	34.36	**34.37**
$\sigma = 20$	30.9	30.92	30.94	30.94	**30.95**
$\sigma = 30$	28.88	28.9	28.93	28.94	**28.95**

We present our evaluation results (average PSNR and SSIM) on the standard test images in Figure 4, with different noise levels ($\sigma = 5, 10, 20, 30$). The test image dataset includes 14 images: pepper, house, boat, couple, man, lena, fingerprint (denoted as "fprint"), barbara, montage, monarch, hill, straw, cameraman (denoted as "caman"), and hex.

As shown in Figure 4, compared with image priors generated with ADA-GMM and INT-GMM, OL-GMM produces significantly better denoising performance at all noise levels. Also, the OL-GMM approach achieves competitive results to both [4] and [3] that leverage internal and external patch priors, respectively. As shown in Figure 4, OL-GMM outperforms all other competing methods at $\sigma = 5$ in terms of average performance on standard images. In particular, our method outperforms BM3D by 0.15db in average PSNR, which is a remarkable improvement considering that BM3D is still the state of the art for image denoising with low variance (small noise) cases.

By comparing OL-GMM with OL-GMM-RT, we identify that with an image-specific prior, we do not need global image reconstruction as in [9]. Compared with EPLL [9], we achieved a better denoising result with a single EM iteration while the former requires multiple iterations. For large σ, our approach can be further improved if we use complimentary internal self-similarity-based methods such as BM3D to initialize our internal GMM model. We leave this fusion method as our future work.

Individual results of $\sigma = 5, 10, 20, 30$ are presented in Figure 5, where the proposed OL-GMM approach was compared with EPLL [9] and BM3D [1]. The advantage of our method at small noise levels can be further validated from the comparisons for individual test images, shown in Figure 5, which demonstrate that the proposed OL-GMM approach achieves the best result on 12 out of the 14 benchmark images. Small noise levels ($\sigma < 5$) has practical importance since real-world images are mostly corrupted by such noises.

To examine the accuracy of the adopted approximated solution in Section III-C, we empirically evaluated denoising results in terms of PSNR using 1 to 10 largest GMM posterior probability components, respectively, and present the results in Figure 6. As shown in the figure, when σ is 5, the highest average PSNR is achieved when the two largest-posterior-probability components are used; and when σ is 10, 20, or 30, the highest average PSNR is achieved by only using the component with the maximum posterior probability. These results demonstrate the accuracy of the approximated solution of only using the component with the maximum posterior probability for noising. The phenomenon that the average PSNR decreases when using more GMM components may

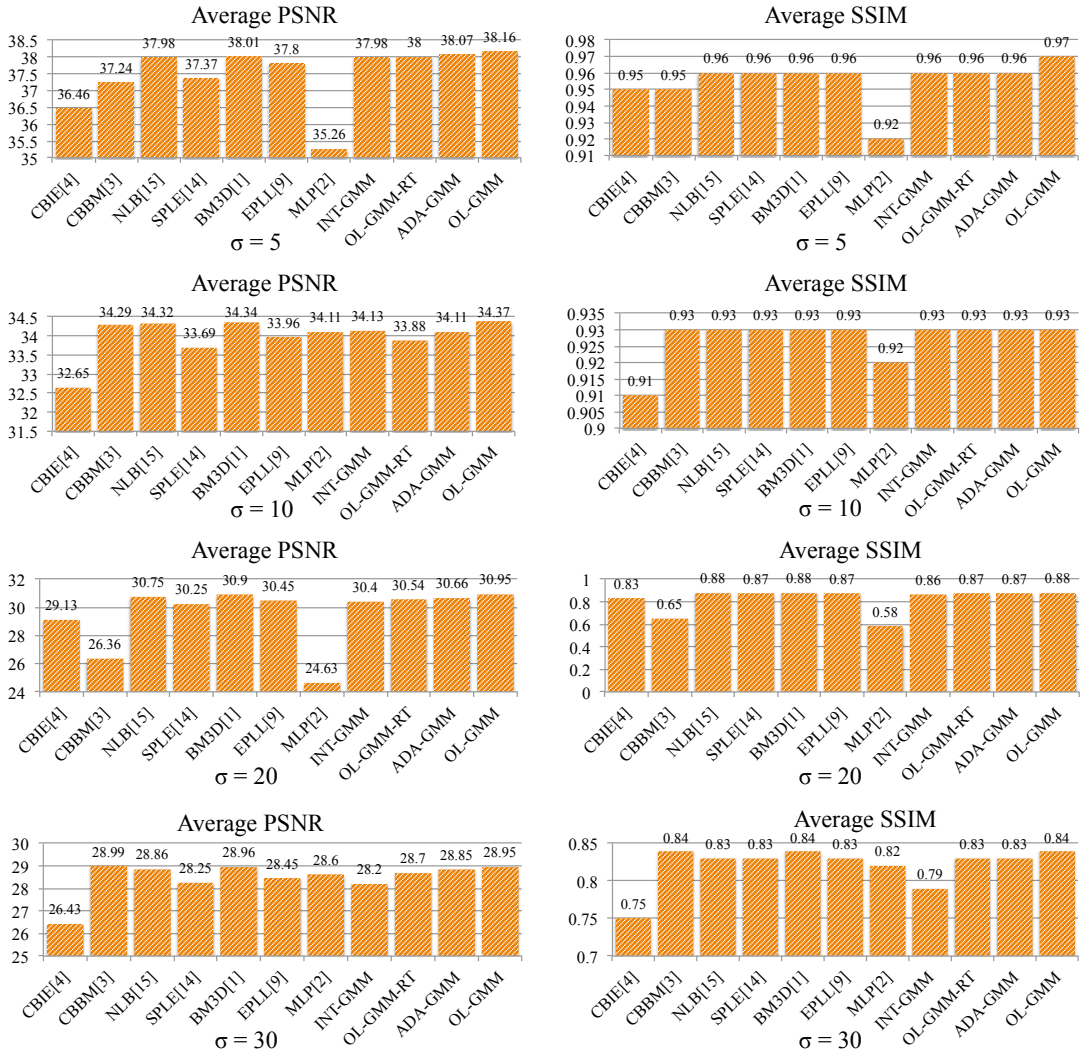


Fig. 4. Average PSNR and SSIM on standard images.

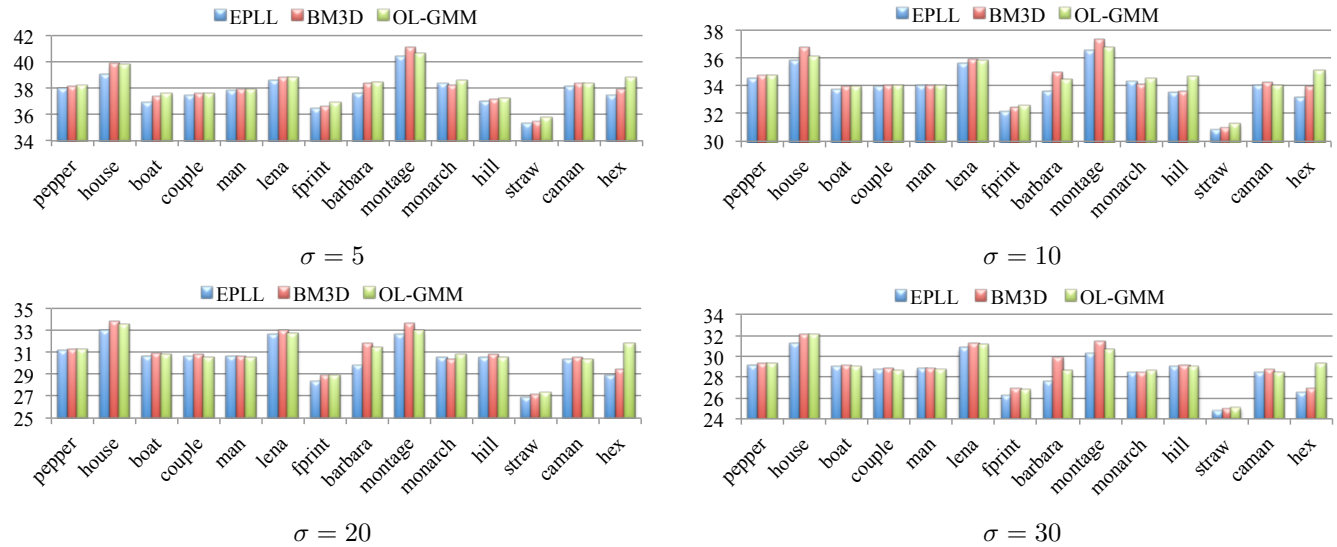


Fig. 5. Denoising performance on standard images. The proposed online GMM approach (OL-GMM) (in green) was compared with EPLL [9] (in blue) and BM3D [1] (in red) for $\sigma = 5, 10, 20, 30$. The figure is better viewed in color.

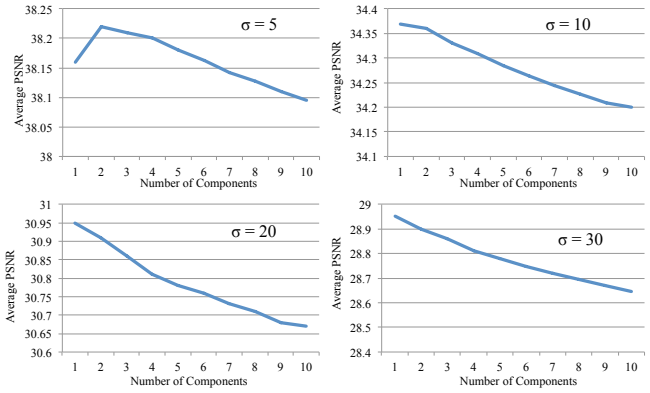


Fig. 6. Denoising Results (average PSNR) on standard images as the number of GMM components increases. The x-axis denotes the number of components used for denoising, and the y-axis refers to average PSNR.

be caused by the imperfect statistical patch model (i.e., the GMM model).

With Matlab implementation using Intel(R) Xeon(R) CPU X5550 @ 2.67GHz, it took about 2 minutes to denoise a 256×256 image (all overlapping patches were used in denoising). The computational bottlenecks lie in two parts: 1) clustering internal patch and generating a coarse image-specific GMM model and 2) one iteration GMM adaptation. The running time could be significantly improved by code optimization.

B. Evaluation on VOC2007, McGill, and BSD-Test

To further verify the effectiveness of the proposed approach, we compared our results produced by OL-GMM to those produced by BM3D [1] and EPLL [9] on the BSD-test, Pascal VOC2007, and McGill datasets. We used all of the 100 images in the BSD-test set, and we randomly sampled 100 images from both the Pascal VOC2007 set and the McGill set. We present the average PSNR and SSIM results on different noise levels ($\sigma = 5, 10, 15, 20, 25, 30$). The comparison of denoising performance is presented in Figure 7. As shown in the figure, with the image-specific prior, we achieve competitive denoising results on all noise levels. In particular, for small noise levels (i.e., σ = 5 and σ = 10), the OL-GMM consistently generate better denoising results than do the EPLL [9] approach on the test images in all three datasets, and it approaches BM3D in terms of quality.

C. Qualitatively Analysis of Image-Specific Patches

In addition to performing quantitative analyses, we also qualitatively analyze patches that are the referred to as "image-specific patches". We take the image of "barbara" as an example, and visualize examples of added components given the noisy image ($\sigma = 5$). As shown in Figure 8, frequently recurring patches in the test image tend to form new components, which indicates that those patches are the referred "image-specific patches". Those patches have higher density among patches in the same image than patches sampled from a diverse collection of images have. For instance, the texture of stripe frequently recurred in the "barbara" image, and patches with those patterns may not recur frequently in natural image patches. As shown in Figure 8, when we conduct denoising on the corrupted "barbara" image, patches with stripe texture tend to form new Gaussian components, and the new Gaussian components are used to denoise the image. Patches with low-contrast patterns also tend to be "image-specific patches" in case they appear frequently in the test image, as shown in the third component in Figure 8.

We found that image-specific patches are spatially clustered together, i.e., "image-specific patches" tend to cluster in some region of the image and forms new Gaussian components. By forming individual GMM components for these "image-specific patches", we adapt the generic GMM prior to an image-specific prior, which leverages the local patch similarity for image denoising.

V. Conclusions and Future Work

We present a unified algorithm for learning an image-specific patch prior and applied the prior to the image denoising problem. Rather than combining internal and external denoising results, we unified the two types of priors in a principled way in order to use more fully the internal and external information. We demonstrate the effectiveness of our approach by comparing with image-specific priors generated by alternative approaches. Experimental results demonstrate that with the novel prior, we achieved a better or competitive denoising performance in terms of the peak signal-to-noise ratio and structural similarity.

As a future work, the proposed image-specific patch prior could be used for other restoration tasks, such as image deblurring and image super-resolution. Since our method provides an improved prior over EPLL, it can be applied to deblurring by following the deblurring algorithm in Zoran and Weiss, ICCV2011 [19]. The image specific prior can be first learned from any existing deblurring algorithm and then the blurred input image can be restored based on the learned prior, with which the restoration result could be iteratively improved. For super-resolution, we can learn the image-specific prior from the low-resolution input and then use it to constrain reconstructed patches in high resolution in an MAP framework.

References

[1] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing (TIP)*, vol. 16, no. 8, pp. 2080–2095, 2007.

[2] H. Burger, C. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2392 – 2399.

[3] H. C. Burger, C. J. Schuler, and S. Harmeling, "Learning how to combine internal and external denoising methods," in *German Conference on Pattern Recognition*, 2013, pp. 121–130.

[4] I. Mosseri, M. Zontak, and M. Irani, "Combining the power of internal and external denoising," in *IEEE International Conference on Computational Photography (ICCP)*, 2013, pp. 1–9.

[5] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[6] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 2272 –2279.

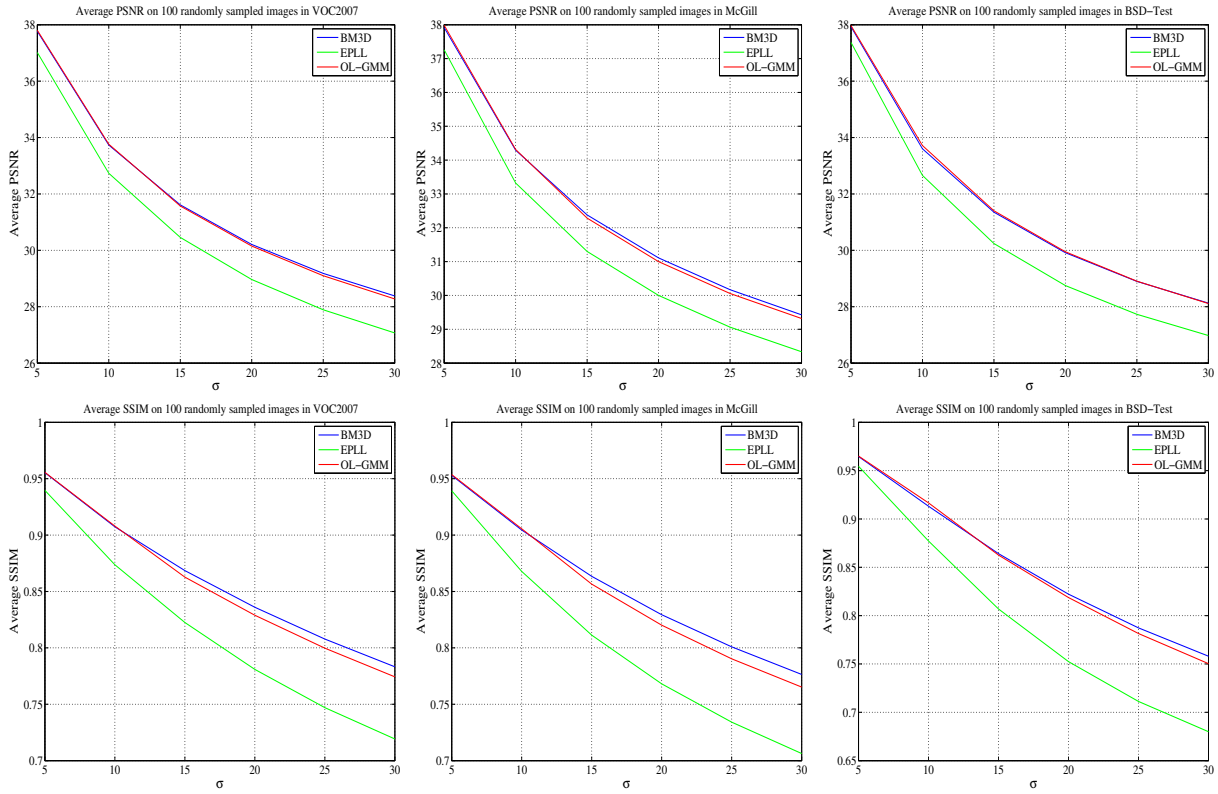


Fig. 7. Denoising performance on VOC2007, McGill, and BSD-Test dataset.

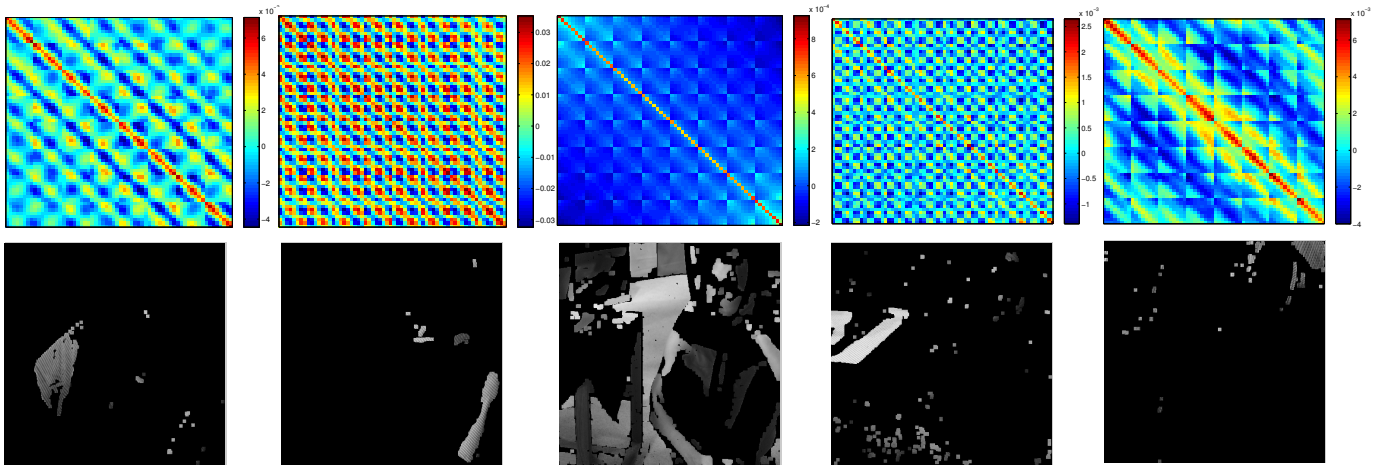


Fig. 8. Examples of added components and associated denoised patches. The first row shows the covariance matrix of the components, and the second row presents the patches being assigned to the components.

[7] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 457–464.

[8] M. Zontak, I. Mosseri, and M. Irani, "Separating signal from noise using patch recurrence across scales," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1195–2102.

[9] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 479–486.

[10] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing (TIP)*, vol. 15, no. 12, pp. 3736–3745, 2006.

[11] S. Roth and M. Black, "Fields of experts: a framework for learning image priors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 860–867.

[12] K. Cho, "Boltzmann machines and denoising autoencoders for image denoising," in *arXiv:1301.3468v5*, 2013.

[13] J. Y. Xie, L. Xu, and E. H. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 350–358.

[14] Y. Wang and J. Morel, "SURE guided gaussian mixture image denoising," *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 999–1034, 2013.

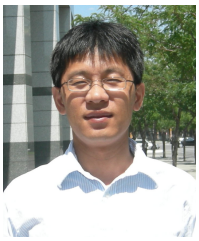
[15] M. Lebrun, A. Buades, and J. Morel, "A nonlocal bayesian image denoising algorithm," *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, p. 16651688, 2013.

[16] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Proceeding*, vol. 2, no. 2, pp. 291–298, 1994.

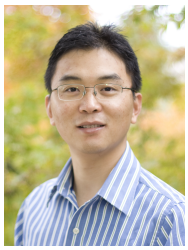
[17] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing (TIP)*, vol. 13, no. 4, pp. 600–612, 2004.

Xin Lu Xin Lu received the B.E. and B.A. degree in Electronic Engineering and English, the M.E. Degree in Signal and Information Processing, all from Tianjin University, China, in 2008 and 2010 respectively. She is currently a Ph.D. candidate at College of Information Sciences and Technology, The Pennsylvania State University, University Park. She was a research intern at Adobe Research in the summer 2012, 2013, and 2014. Her research interests include computer vision, deep learning, image processing, Web search and mining, and user-generated content analysis.

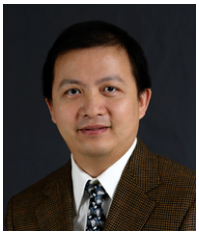


Zhe Lin Zhe Lin received the BEng degree in automatic control from the University of Science and Technology of China in 2002, the MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology in 2004, and the PhD degree in electrical and computer engineering from the University of Maryland, College Park, in 2009. He has been a research intern at Microsoft Live Labs Research. He is currently a senior research scientist at Adobe Research, San Jose, California. His research interests include object detection and recognition, image classification, content-based image and video retrieval, human motion tracking, and activity analysis. He is a member of the IEEE.



Hailin Jin Dr. Hailin Jin received his Bachelor's degree in Automation from Tsinghua University, Beijing, China in 1998. He then received his Master of Science and Doctor of Science degrees in Electrical Engineering from Washington University in Saint Louis in 2000 and 2003 respectively. Between fall 2003 and fall 2004, he was a postdoctoral researcher at the Computer Science Department, University of California at Los Angeles. Since October 2004, he has been with Adobe Systems Incorporated where he is currently a Principal Scientist. He received the best student paper award (with J. Andrews and C. Sequin) at the 2012 International CAD conference for work on interactive inverse 3D modeling. He is a member of the IEEE and the IEEE Computer Society.

Jianchao Yang Jianchao Yang (S08, M12) received the B.E. degree from the Department of Electronics Engineering and Information Science, University of Science and Technology of China, Hefei, China, in 2006, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Urbana, in 2011. He is currently a Research Scientist with the Advanced Technology Laboratory, Adobe Systems Inc., San Jose, CA. His research interests include object recognition, deep learning, sparse coding, compressive sensing, image and video enhancement, denoising, and deblurring.



James Z. Wang James Z. Wang (S96-M00-SM06) is a Professor and the Chair of Faculty Council at the College of Information Sciences and Technology, The Pennsylvania State University. He received a Summa Cum Laude Bachelors degree in Mathematics and Computer Science from University of Minnesota, an M.S. in Mathematics and an M.S. in Computer Science, both from Stanford University, and a Ph.D. degree in Medical Information Sciences from Stanford University. His main research interests are automatic image tagging, aesthetics and emotions, computerized analysis of paintings, and image retrieval.