

000  
001  
002054  
055  
056

# External Prior Guided Internal Prior Learning for Real Noisy Image Denoising

057  
058  
059003  
004  
005  
006  
007  
008  
009  
010  
011060  
061  
062  
063  
064  
065

Anonymous CVPR submission

012

066

013

067

014

068

015

069

016

070

017

071

018

072

019

073

020

074

021

075

022

076

023

077

024

078

025

079

026

080

027

081

028

082

029

083

030

084

031

085

032

086

033

087

034

088

035

089

036

090

037

091

038

092

039

Most of existing image denoising methods use some statistical models such as additive white Gaussian noise (AWGN) to model the noise, and learn image priors from either external data or the noisy image itself to remove noise. However, the noise in real-world noisy images is much more complex than AWGN, and it is hard to be modeled by simple analytical distributions. Therefore, many state-of-the-art denoising methods in literature become much less effective when applied to real noisy images. In this paper, we develop a robust denoiser for real noisy image denoising without explicit assumption on noise models. Specifically, we first learn external priors from a set of clean natural images, and then use the learned external priors to guide the learning of internal latent priors from the given noisy image. The proposed method is simple yet highly effective. Experiments on real noisy images demonstrate that it achieves much better denoising performance than state-of-the-art denoising methods, including those designed for real noisy images.

033

093

## 1. Introduction

Image denoising is a crucial and indispensable step to improve image quality in digital imaging systems. In particular, with the decrease of size of CMOS/CCD sensors, noise is more easily to be corrupted and hence denoising is becoming increasingly important for high resolution imaging. In literature of image denoising, the observed noisy image is usually modeled as  $\mathbf{y} = \mathbf{x} + \mathbf{n}$ , where  $\mathbf{x}$  is the latent clean image and  $\mathbf{n}$  is the corrupted noise. Numerous image denoising methods [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] have been proposed in the past decades, including sparse representation and dictionary learning based methods [1, 2, 3], nonlocal self-similarity based methods [4, 5, 6, 3, 7], low-rank based methods [8], neural network based methods [9], and discriminative learning based methods [10, 11].

Most of the existing denoising methods [1, 2, 4, 5, 6, 3, 7, 8, 9, 10, 11, 12, 13] mentioned above assume noise  $\mathbf{n}$  to be additive white Gaussian noise (AWGN). Unfortunately, this assumption is too ideal to be true for real-world noisy im-

ages, where the noise is much more complex than AWGN [14, 15] and varies by different cameras and camera settings (ISO, shutter speed, and aperture, etc.). According to [15], the noise corrupted in the imaging process [is signal dependent and comes from five main sources: photon shot, fixed pattern, dark current, readout, and quantization noise. As a result, many advanced denoising methods in literature becomes much less effective when applied to real-world noisy images. Fig. 1 shows an example, where we apply some representative and state-of-the-art denoising methods, including CBM3D [6], WNNM [8], MLP [9], CSF [10], and TRD [11], to a real noisy image (captured by a Nikon D800 camera with ISO is 3200) provided in [14]. One can see that these methods either remain the noise or over-smooth the image details on this real noisy image.

There have been a few methods [16, 17, 18, 14, 19, 20, 21] developed for real noisy image denoising. Almost all of these methods follow a two-stage framework: first estimate the parameters of the assumed noise model (usually Gaussian or mixture of Gaussians (MoG)), and then perform denoising with the estimated noise model. Again, the noise in real noisy images is very complex and hard to be modeled by explicit distributions such as Gaussian and MoG. Fig. 1 also shows the denoised results of two state-of-the-art real noisy image denoising methods, Noise Clinic [19, 20] and Neat Image [21]. One can see that these two methods do not perform well on this noisy image either.

This work aims to develop a robust solution for real noisy image denoising without explicitly assuming certain noise models. To achieve this goal, we propose to first learn image priors from external clean images, and then employ the learned external priors to guide the learning of internal latent priors from the given noisy image. The flowchart of the proposed method is illustrated in Fig. 3. We first extract millions of patch groups from a set of high quality natural images, with which a Gaussian Mixture Model (GMM) is learned as the external prior. The learned GMM prior model is used to cluster the patch groups extracted from the given noisy image, and then a hybrid orthogonal dictionary (HOD) is learned as the internal prior for image denoising. Our proposed denoising method is simple and ef-

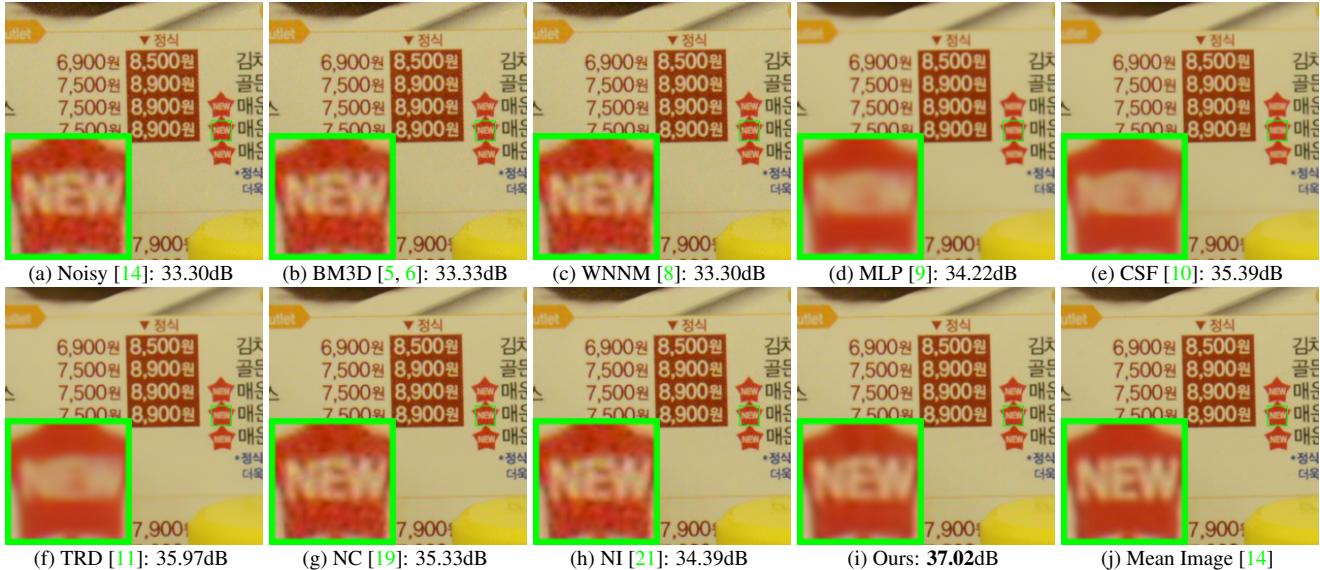


Figure 1. Denoised images of the real noisy image “Nikon D800 ISO 3200 A3” from [14] by different methods. The images are better viewed by zooming in on screen.

ficient, yet our extensive experiments on real noisy images clearly demonstrate its better denoising performance than the current state-of-the-arts.

## 2. Related Work

### 2.1. Internal vs. External Prior Learning

Image priors are playing a key role in image denoising [7, 13, 1, 22, 3, 23]. There are mainly two categories of prior learning methods. 1) External prior learning methods [12, 7, 13] learn priors (e.g., dictionaries) from a set of external clean images, and the learned priors are used to recover the latent clean image from noisy images. 2) Internal prior learning methods [1, 3, 22, 23] directly learn priors from the given noisy image, and the image denoising is often done simultaneously with the prior learning process. It has been demonstrated [7, 13] that the external priors learned from natural clean images are effective and efficient for image denoising problem, but they are not adaptive to the given noisy image so that some fine-scale image structures may not be well recovered. By contrast, the internal priors are adaptive to content of the given image, but the learning processing are usually slow. In addition, most of the internal prior learning methods [1, 3, 22, 23] assume AWGN noise, making the learned priors less robust for real noisy images. In this paper, we use external priors to guide the internal prior learning. Our method is not only much faster than the traditional internal learning methods, but also very effective to denoise real noisy images.

### 2.2. Real Noisy Image Denoising

In the last decade, there are many methods [16, 17, 19, 20, 18, 14] for blind image denoising problem. These meth-

ods can be applied to real noisy image denoising directly. Liu *et al.* [16] proposed to use “noise level function” to estimate the noise and then use Gaussian conditional random field to obtain the latent clean image. Gong et al. [17] models the noise by mixed  $\ell_1$  and  $\ell_2$  norms and remove the noise by sparsity prior in the wavelet transform domain. Recently, Zhu et al. proposed a Bayesian model [18] which approximates and removes the noise via low-rank mixture of Gaussians. The method of “Noise Clinic” [19, 20] and the software of Neat Image [21] are developed specifically for real noisy image denoising. “Noise Clinic” [19, 20] generalizes the NL-Bayes model [24] to deal with blind noise and achieves state-of-the-art performance. However, these methods largely depends on the modeling of noise in real noisy images which is hard to be modeled by explicit distributions. Besides, the parametric estimation of the Gaussian or MoG distribution is often time consuming.

## 3. External Prior Guided Internal Prior Learning

In this section, we first describe the learning of external prior, and then describe in detail the guided internal prior learning. Finally, the denoising algorithm with the learned priors is presented.

### 3.1. Learn External Patch Group Priors

The nonlocal self-similarity based patch group (PG) [7] has proved to be a very effective unit for image prior learning. In this work, we also extract PGs from natural clean images to learn priors. A PG is a group of similar patches to a local patch.

In our method, each local patch is extracted from a

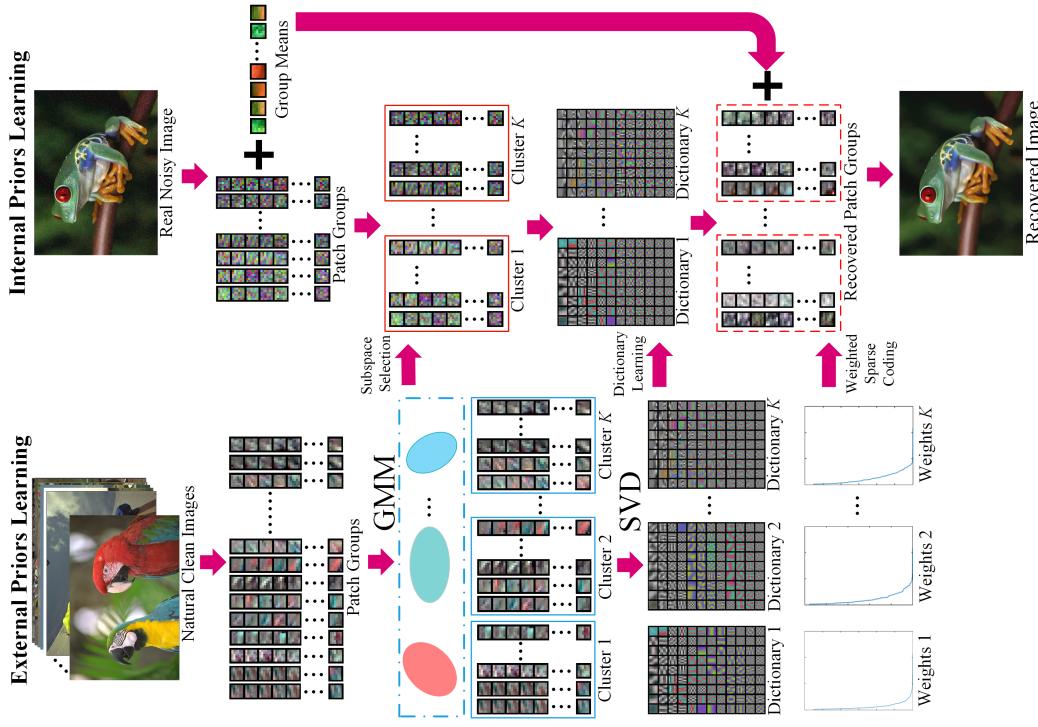


Figure 2. Flowchart of the proposed external prior guided internal prior learning and real noisy image denoising framework.

RGB image with patch size  $p \times p \times 3$ . We search the  $M$  most similar patches to this local patch (including the local patch itself) in a  $W \times W$  local region around it. Each patch is stretched to a patch vector  $\mathbf{x}_m \in \mathbb{R}^{3p^2 \times 1}$  to form the PG  $\{\mathbf{x}_m\}_{m=1}^M$ . The mean vector of this PG is  $\mu = \frac{1}{M} \sum_{m=1}^M \mathbf{x}_m$ , and the group mean subtracted PG is defined as  $\bar{\mathbf{X}} \triangleq \{\bar{\mathbf{x}}_m = \mathbf{x}_m - \mu\}$ .

Assume we extract a number of  $N$  PGs from a set of external natural images, and the  $n$ -th PG is  $\bar{\mathbf{X}}_n \triangleq \{\bar{\mathbf{x}}_{n,m}\}_{m=1}^M, n = 1, \dots, N$ . A Gaussian Mixture Model (GMM) is learned to model the PG prior. The overall log-likelihood function is

$$\ln \mathcal{L} = \sum_{n=1}^N \ln \left( \sum_{k=1}^K \pi_k \prod_{m=1}^M \mathcal{N}(\bar{\mathbf{x}}_{n,m} | \mu_k, \Sigma_k) \right). \quad (1)$$

The learning process is similar to the GMM learning in [7, 13]. Finally, a GMM model with  $K$  Gaussian components is learned, and the learned parameters include mixture weights  $\{\pi_k\}_{k=1}^K$ , mean vectors  $\{\mu_k\}_{k=1}^K$ , and covariance matrices  $\{\Sigma_k\}_{k=1}^K$ . Note that the mean vector of each cluster is naturally zero, i.e.,  $\mu_k = \mathbf{0}$ .

To better describe the subspace of each Gaussian component, we perform singular value decomposition (SVD) on the covariance matrix:

$$\Sigma_k = \mathbf{U}_k \mathbf{S}_k \mathbf{U}_k^\top. \quad (2)$$

The eigenvector matrices  $\{\mathbf{U}_k\}_{k=1}^K$  will be employed as the external orthogonal dictionary to guide the internal dictionary learning in next sub-section. In Fig. 4 (a) and (b), we

illustrate an external clean image and one orthogonal dictionary learned via GMM on PGs of the external clean image. The singular values in  $\mathbf{S}_k$  reflect the significance of the singular vectors in  $\mathbf{U}_k$ . They will also be utilized as prior weights for weighted sparse coding in our denoising algorithm.

### 3.2. Guided Internal Prior Learning

After the external PG prior is learned, we employ it to guide the internal PG prior learning for a given real noisy image. The guidance lies in two aspects. One is that the external prior can guide the subspace assignment of internal noisy PGs, while the other is that the external prior could guide the orthogonal dictionary learning of internal noisy PGs.

#### 3.2.1 Internal Subspace Assignment

Given a real noisy image, we extract  $N$  (overlapped) local patches from it. Similar to the external prior learning stage, for the  $n$ -th local patch we search its  $M$  most similar patches around it to form a noisy PG, denoted by  $\mathbf{Y}_n = \{\mathbf{y}_{n,1}, \dots, \mathbf{y}_{n,M}\}$ . Then the group mean of  $\mathbf{Y}_n$ , denoted by  $\mu_n$ , is subtracted from each patch by  $\bar{\mathbf{y}}_{n,m} = \mathbf{y}_{n,m} - \mu_n$ , leading to the mean subtracted noisy PG  $\bar{\mathbf{Y}}_n \triangleq \{\bar{\mathbf{y}}_{n,m}\}_{m=1}^M$ .

The external GMM prior models  $\{\Sigma_k\}_{k=1}^K$  basically characterize the subspaces of natural high quality PGs. Therefore, we project the noisy PG  $\bar{\mathbf{Y}}_n$  into the subspaces of  $\{\Sigma_k\}_{k=1}^K$  and assign it to the most suitable subspace

216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323

324

based on the posterior probability:

$$P(k|\bar{\mathbf{Y}}_n) = \frac{\prod_{m=1}^M \mathcal{N}(\bar{\mathbf{y}}_{n,m}|\mathbf{0}, \Sigma_k)}{\sum_{l=1}^K \prod_{m=1}^M \mathcal{N}(\bar{\mathbf{y}}_{n,m}|\mathbf{0}, \Sigma_l)} \quad (3)$$

for  $k = 1, \dots, K$ . Then  $\bar{\mathbf{Y}}_n$  is assigned to the component with the maximum A-posteriori (MAP) probability  $\max_k P(k|\bar{\mathbf{Y}}_n)$ .

### 3.2.2 Guided Orthogonal Dictionary Learning

Assume we have assigned all the internal noisy PGs  $\{\bar{\mathbf{Y}}_n\}_{n=1}^N$  to their corresponding most suitable subspaces in  $\{\mathcal{N}(\mathbf{0}, \Sigma_k)\}_{k=1}^K$ . For the  $k$ -th subspace, the noisy PGs assigned to it are  $\{\bar{\mathbf{Y}}_{k,n}\}_{n=1}^{N_k}$  where  $\bar{\mathbf{Y}}_{k,n} = [\bar{\mathbf{y}}_{k,n,1}, \dots, \bar{\mathbf{y}}_{k,n,M}]$  and  $\sum_{k=1}^K N_k = N$ . We propose to learn an orthogonal dictionary  $\mathbf{D}_k$  from each set of PGs  $\bar{\mathbf{Y}}_{k,n}$  with the guidance of the corresponding external orthogonal dictionary  $\mathbf{U}_k$  (Eq. (2)) to characterize the internal PG prior. The reasons that we learn orthogonal dictionaries are two-fold. Firstly, the PGs  $\bar{\mathbf{Y}}_{k,n}$  are in a subspace of the whole space of all PGs, therefore, there is no necessary to learn a redundant over-complete dictionary to characterize it, while an orthonormal dictionary has naturally zero *mutual incoherence* [25]. Secondly, the orthogonality of dictionary can make the encoding in the testing stage very efficient, leading to an efficient denoising algorithm (please refer to sub-section 3.3 for details).

We let the orthogonal dictionary  $\mathbf{D}_k$  be  $\mathbf{D}_k \triangleq [\mathbf{D}_{k,E} \mathbf{D}_{k,I}] \in \mathbb{R}^{3p^2 \times 3p^2}$ , where  $\mathbf{D}_{k,E} = \mathbf{U}_k(:, 1:r) \in \mathbb{R}^{3p^2 \times r}$  is the external sub-dictionary and it includes the first  $r$  most important eigenvectors of  $\mathbf{U}_k$ , and the internal sub-dictionary  $\mathbf{D}_{k,I}$  is to be adaptively learned from the noisy PGs  $\{\bar{\mathbf{Y}}_{k,n}\}_{n=1}^{N_k}$ . The rationale to design  $\mathbf{D}_k$  as a hybrid dictionary is as follows. The external sub-dictionary  $\mathbf{D}_{k,E}$  is pre-trained from external clean data, and it represents the  $k$ -th latent subspace of natural images, which is helpful to reconstruct the common latent structures of images. However,  $\mathbf{D}_{k,E}$  is general to all images and it is not adaptive to the given noisy image. Some fine-scale details specific to the given image may not be well characterized by  $\mathbf{D}_{k,E}$ . Therefore, we learn an internal sub-dictionary  $\mathbf{D}_{k,I}$  to supplement  $\mathbf{D}_{k,E}$ . In other words,  $\mathbf{D}_{k,I}$  is to reveal the latent subspace adaptive to the input noisy image, which cannot be effectively represented by  $\mathbf{D}_{k,E}$ .

For notation simplicity, in the following development we ignore the subspace index  $k$  for  $\bar{\mathbf{Y}}_{k,n}$  and  $\mathbf{D}_k$ , etc. The learning of hybrid orthogonal dictionary  $\mathbf{D}$  is performed under the following weighted sparse coding framework:

$$\begin{aligned} & \min_{\mathbf{D}, \{\alpha_{n,m}\}} \sum_{n=1}^N \sum_{m=1}^M (\|\bar{\mathbf{y}}_{n,m} - \mathbf{D}\alpha_{n,m}\|_2^2 + \sum_{j=1}^{3p^2} \lambda_j |\alpha_{n,m,j}|) \\ & \text{s.t. } \mathbf{D} = [\mathbf{D}_e \mathbf{D}_i], \mathbf{D}_i^\top \mathbf{D}_i = \mathbf{I}_r, \mathbf{D}_e^\top \mathbf{D}_i = \mathbf{0}, \end{aligned} \quad (4)$$

where  $\alpha_{n,m}$  is the sparse coding vector of the  $m$ -th patch  $\bar{\mathbf{y}}_{n,m}$  in the  $n$ -th PG  $\bar{\mathbf{Y}}_n$  and  $\alpha_{n,m,j}$  is the  $j$ -th element of  $\alpha_{n,m}$ .  $\lambda_j$  is the  $j$ -th regularization parameter defined as

$$\lambda_j = \lambda / (\sqrt{\mathbf{S}_k(j)} + \varepsilon), \quad (5)$$

where  $\mathbf{S}_k(j)$  is the  $j$ -th singular value of diagonal singular value matrix  $\mathbf{S}_k$  (please refer to Eq. (2)) and  $\varepsilon$  is a small positive number to avoid zero denominator. Noted that  $\mathbf{D}_E = \mathbf{U}_k$  if  $r = 3p^2$  and  $\mathbf{D}_E = \emptyset$  if  $r = 0$ . The dictionary  $\mathbf{D} = [\mathbf{D}_E \mathbf{D}_I]$  is orthogonal by checking that:

$$\mathbf{D}^\top \mathbf{D} = \begin{bmatrix} \mathbf{D}_e^\top \\ \mathbf{D}_i^\top \end{bmatrix} [\mathbf{D}_e \mathbf{D}_i] = \begin{bmatrix} \mathbf{D}_e^\top \mathbf{D}_e & \mathbf{D}_e^\top \mathbf{D}_i \\ \mathbf{D}_i^\top \mathbf{D}_e & \mathbf{D}_i^\top \mathbf{D}_i \end{bmatrix} = \mathbf{I} \quad (6)$$

We employ an alternating iterative approach to solve the optimization problem (4). Specifically, we initialize the orthogonal dictionary as  $\mathbf{D}^{(0)} = \mathbf{U}_k$  and for  $t = 0, 1, \dots, T-1$ , we alternatively update  $\alpha_{n,m}$  and  $\mathbf{D}$  as follows:

**Updating Sparse Coefficient:** Given the orthogonal dictionary  $\mathbf{D}^{(t)}$ , we update each sparse coding vector  $\alpha_{n,m}$  by solving

$$\alpha_{n,m}^{(t)} := \arg \min_{\alpha_{n,m}} \|\bar{\mathbf{y}}_{n,m} - \mathbf{D}^{(t)} \alpha_{n,m}\|_2^2 + \sum_{j=1}^{3p^2} \lambda_j |\alpha_{n,m,j}| \quad (7)$$

Since dictionary  $\mathbf{D}^{(t)}$  is orthogonal, the problems (7) has a closed-form solution

$$\alpha_{n,m}^{(t)} = \text{sgn}((\mathbf{D}^{(t)})^\top \bar{\mathbf{y}}_{n,m}) \odot \max(|(\mathbf{D}^{(t)})^\top \bar{\mathbf{y}}_{n,m}| - \lambda, 0), \quad (8)$$

where  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_{3p^2}]$  is the vector of regularization parameter and  $\text{sgn}(\bullet)$  is the sign function,  $\odot$  means element-wise multiplication. The detailed derivation of Eq. (8) can be found in the supplementary file.

**Updating Internal Sub-dictionary:** Given the sparse coding vectors  $\alpha_{n,m}^{(t)}$ , we update the internal sub-dictionary by solving

$$\begin{aligned} \mathbf{D}_I^{(t+1)} &:= \arg \min_{\mathbf{D}_I} \sum_{n=1}^N \sum_{m=1}^M (\|\bar{\mathbf{y}}_{n,m} - \mathbf{D} \alpha_{n,m}^{(t)}\|_2^2) \\ &= \arg \min_{\mathbf{D}_I} \|\mathbf{Y} - \mathbf{D} \mathbf{A}^{(t)}\|_F^2 \end{aligned} \quad (9)$$

$$\text{s.t. } \mathbf{D} = [\mathbf{D}_E \mathbf{D}_I], \mathbf{D}_I^\top \mathbf{D}_I = \mathbf{I}_r, \mathbf{D}_E^\top \mathbf{D}_I = \mathbf{0},$$

where  $\mathbf{A}^{(t)} = [\alpha_{1,1}^{(t)}, \dots, \alpha_{1,M}^{(t)}, \dots, \alpha_{N,1}^{(t)}, \dots, \alpha_{N,M}^{(t)}]$ . The sparse coefficient matrix can be written as  $\mathbf{A}^{(t)} = [(\mathbf{A}_E^{(t)})^\top (\mathbf{A}_I^{(t)})^\top]^\top$  where the external part  $\mathbf{A}_E^{(t)} \in \mathbb{R}^{(3p^2-r) \times NM}$  and the internal part  $\mathbf{A}_I^{(t)} \in \mathbb{R}^{r \times NM}$  represent the coding coefficients of  $\mathbf{Y}$  over external sub-dictionary  $\mathbf{D}_E$  and internal sub-dictionary  $\mathbf{D}_I$ , respectively. According to the Theorem 4 in [26], the problem (9) has a closed-form solution  $\mathbf{D}_I^{(t+1)} = \mathbf{U}_i \mathbf{V}_i^\top$ , where  $\mathbf{U}_i \in \mathbb{R}^{3p^2 \times r}$  and  $\mathbf{V}_i \in \mathbb{R}^{r \times r}$  are the orthogonal matrices obtained by the following SVD

432   **Alg. 1:** External Prior Guided Internal Prior Learning  
 433    for Real Noisy Image Denoising

434   **Input:** Noisy image  $\mathbf{y}$ , external PG prior GMM model  
 435   **Output:** The denoised image  $\hat{\mathbf{x}}$ .  
 436   **Initialization:**  $\hat{\mathbf{x}}^{(0)} = \mathbf{y}$ ;  
 437   **for**  $Ite = 1 : IteNum$  **do**  
 438     1. Extracting internal PGs from  $\hat{\mathbf{x}}^{(Ite-1)}$ ;  
 439       **for** each PG  $\mathbf{Y}_n$  **do**  
 440         2. Calculate group mean vector  $\mu_n$  and form  
 441           mean subtracted PG  $\bar{\mathbf{Y}}_n$ ;  
 442         3. Subspace selection via Eq. (3);  
 443       **end for**  
 444       **for** the PGs in each Subspace **do**  
 445         4. External PG prior Guided Internal Orthogonal  
 446           Dictionary Learning by solving (4);  
 447         5. Recover each patch in all PGs via Eq. (11);  
 448       **end for**  
 449       6. Aggregate the recovered PGs of all subspaces to form  
 450           the recovered image  $\hat{\mathbf{x}}^{(Ite)}$ ;  
 451   **end for**

$$(\mathbf{I} - \mathbf{D}_e \mathbf{D}_e^\top) \mathbf{Y} (\mathbf{A}_i^{(t)})^\top = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^\top. \quad (10)$$

452  
 453  
 454  
 455  
 456   The orthogonality of internal dictionary  $\mathbf{D}_i^{(t+1)}$  can be  
 457   checked by  $(\mathbf{D}_i^{(t+1)})^\top (\mathbf{D}_i^{(t+1)}) = \mathbf{V}_i \mathbf{U}_i^\top \mathbf{U}_i \mathbf{V}_i^\top = \mathbf{I}_r$ . In Figure 4 (c) and (d), we illustrate a denoised image by  
 458   our proposed method and one internal orthogonal dictionary  
 459   learned from PGs of the given noisy image.  
 460

### 3.3. The Denoising Algorithm

461  
 462   The denoising of the given noisy image can be simultaneously done with the guided internal dictionary learning process. Once we obtain the solutions of sparse coding  
 463   vectors  $\{\hat{\alpha}_{n,m}^{(T-1)}\}$  in Eq. (8) and the orthogonal dictionary  
 464    $\mathbf{D}_{(T)} = [\mathbf{D}_E \mathbf{D}_I^{(T)}]$  in Eq. (9), the latent clean patch of a  
 465   noisy patch  $\hat{\mathbf{y}}_{n,m}$  in PG  $\mathbf{Y}_n$  is reconstructed as  
 466

$$\hat{\mathbf{y}}_{n,m} = \mathbf{D}_{(T)} \hat{\alpha}_{n,m} + \mu_n, \quad (11)$$

467  
 468   where  $\mu_n$  is the group mean of  $\mathbf{Y}_n$ . The latent clean image  
 469   is then reconstructed by aggregating all the reconstructed  
 470   patches in all PGs. We perform the above denoising pro-  
 471   cedures for several iterations for better denoising outputs.  
 472   The proposed denoising algorithm is summarized in Alg. 1.  
 473   The latent clean image  $\hat{\mathbf{x}}$  is reconstructed by aggregating all  
 474   the estimated PGs. Similar to [7], we perform the above  
 475   denoising procedures for several iterations for better denoising  
 476   outputs. The proposed denoising algorithm is summarized  
 477   in Alg. 1.

## 4. Experiments

482  
 483   We evaluate the performance of the proposed algorithm  
 484   on real-world noisy images [14, 20] in comparison with

485   state-of-the-art denoising methods [5, 6, 9, 8, 10, 11, 14,  
 486   19, 20, 21].

### 4.1. Implementation Details

487  
 488   Our proposed method has two stages: the external prior  
 489   learning stage and the external prior guided internal prior  
 490   learning stage. In the first stage, we set  $p = 6$  (the  
 491   patch size),  $M = 10$  (the number of similar patches in  
 492   a PG),  $W = 31$  (the window size for PG searching) and  
 493    $K = 32$  (the number of Gaussian components in GMM). We  
 494   learn the external GMM prior with 3.6 million PGs ex-  
 495   tracted from the Kodak PhotoCD Dataset (<http://r0k.us/graphics/kodak/>), which includes 24 high quality  
 496   color images.  
 497

498  
 499   In the second stage, we set  $r = 54$  (the number of atoms  
 500   in the external sub-dictionaries); that is, we let the external  
 501   sub-dictionary have the same number of atoms as the internal  
 502   sub-dictionary to be learned. Our experiments show that  
 503   setting  $r$  between 27 and 81 will lead to very similar results.  
 504   For other parameters, we set  $\lambda = 0.001$  (the sparse regu-  
 505   larization parameter),  $T = 2$  (the number of iterations for  
 506   solving problem (4)), and  $IteNum = 4$  (the number of iter-  
 507   ations for Alg.1). All parameters of our method are fixed to  
 508   all experiments, which are run under the Matlab2014b en-  
 509   vironment on a machine with Intel(R) Core(TM) i7-5930K  
 510   CPU of 3.5GHz and 32GB RAM.  
 511

### 4.2. The Testing Datasets

512  
 513   The comparisons are performed on two standard datasets  
 514   in which the images were captured under indoor or out-  
 515   door lighting conditions by different types of cameras and  
 516   camera settings. The first dataset provided in [20] includes  
 517   20 real noisy images collected under uncontrolled outdoor  
 518   environment. This dataset does not have “ground truth”  
 519   images and hence the objective measurements can not be  
 520   performed. In order to evaluate the compared methods on  
 521   quantitative measures, we perform experiments on the sec-  
 522   ond dataset provided in [14]. It includes 17 real noisy im-  
 523   ages and corresponding mean images. The noisy images  
 524   were collected under controlled indoor environment. Some  
 525   samples can be found in [14]. For each image, the same  
 526   scene was shot 500 times under the same camera and cam-  
 527   era setting. The mean image of the 500 shots is roughly  
 528   taken as the “ground truth”, with which the PSNR can be  
 529   computed. Since the 17 images are too large (of size about  
 530    $7000 \times 5000 \times 3$ ) and share repetitive contents, the authors in  
 531   [14] performed comparison on 15 cropped images (of size  
 532    $512 \times 512 \times 3$ ). To evaluate the compared methods on more  
 533   samples, we cropped the 17 large images from [14] into 60  
 534   smaller images (of size  $500 \times 500 \times 3$ ) including different  
 535   contents. Some samples are shown in Figure 5. Note that  
 536   the noise in our cropped 60 images used in [14] are different  
 537   from the noise in the 15 images cropped by the authors of  
 538

540 Table 1. Average PSNR (dB) results and Run Time (seconds) of  
 541 the External, the Internal, and our proposed methods on 60 real  
 542 noisy images (of size  $500 \times 500 \times 3$ ) cropped from [14].

	Noisy	External	Internal	Ours
PSNR	34.51	38.21	38.07	<b>38.75</b>
Time	—	<b>39.57</b>	667.36	41.89

543 [14] since they are taken in different shots.

### 544 4.3. Comparison among external, internal and ex- 545 ternal guided internal priors

546 In this section, we compare our proposed method on real  
 547 image denoising with external prior based method (denoted  
 548 as “External”) and internal prior based method (denoted as  
 549 “Internal”). For the “External” method, we utilize the ex-  
 550 ternal dictionaries (i.e.,  $r = 0$  in Eq. (5)) for denoising.  
 551 For the given noisy image, we extract the PGs and then do  
 552 internal subspace selection via Eq. 3. The denoising is per-  
 553 formed via the weighted sparse coding framework proposed  
 554 in [7]. For the “Internal” method, the overall framework is  
 555 similar to the method of [3]. We employ the GMM model  
 556 (also with  $K = 32$  Gaussians) to cluster the noisy PGs ex-  
 557 tracted from given noisy image into multiple subspaces, and  
 558 for each subspace, we utilize the internal orthogonal dictio-  
 559 nary obtained via Eq. (2) by weighted sparse coding frame-  
 560 work in [7]. All parameters of the three methods are tuned  
 561 to achieve best performance.

562 We compare the above mentioned methods on the 60  
 563 cropped images (of size  $500 \times 500 \times 3$ ) from [14]. The  
 564 average PSNR and speed of these methods are listed in Ta-  
 565 ble 1. It can be seen that our proposed method achieves  
 566 better PSNR results than the methods of “External” and “In-  
 567 ternal”. The speed of our proposed method is much faster  
 568 than the “Internal” method while only a little slower than the  
 569 “External” method. We also compare the visual quality  
 570 of the denoised images by these methods. From the results  
 571 listed in Figure 3 and Figure 4, we can see that the “Ex-  
 572 ternal” method is good at recovering structures (Figure 3)  
 573 while the “Internal” method is good at recovering internal  
 574 complex textures (Figure 4). And by utilizing both the ex-  
 575 ternal and internal priors, our proposed method can recover  
 576 well both the structures and textures. Noted that the noisy  
 577 images in Figures 3 and 4 are cropped from the same image  
 578 captured by Nikon D600 at ISO = 3200 in [14]. Hence, the  
 579 differences on PSNR and visual quality among these meth-  
 580 ods only depends on the contents of the cropped images.

### 581 4.4. Comparison with Other Denoising Methods

582 In this section, we compare the proposed method with  
 583 other state-of-the-art image denoising methods such as  
 584 BM3D [5], WNNM [8], MLP [9], CSF [10], TRD [11],  
 585 Noise Clinic (NC) [19], Cross-Channel (CC) [14], and Neat  
 586 Image (NI) [21]. The methods of BM3D [5], WNNM [8],

587 MLP [9], CSF [10], and TRD [11] are designed for remov-  
 588 ing Gaussian noise. For BM3D and WNNM, the level  $\sigma$   
 589 of Gaussian noise is very important and is estimated by  
 590 the method [27]. The other parameters are set as default.  
 591 For the methods of MLP, CSF, and TRD, we employ their  
 592 default parameters settings. Since these methods are de-  
 593 signed for grayscale images, we utilize them to denoise the  
 594 R, G, B channels separately for color noisy images. The  
 595 Noise Clinic (NC) [19] is a blind image denoising method  
 596 which does not need any noise prior. We also compare with  
 597 Neat Image (NI), a commercial software for image denois-  
 598 ing. Due to its excellent performance, Neat Image (NI) is  
 599 embedded into Photoshop and Corel PaintShop [21]. The  
 600 comparisons are performed on the real noisy images from  
 601 [20] and [14].

#### 602 4.4.1 Comparison on the First Dataset [20]

603 The real noisy images in the dataset [20] do not have  
 604 “ground truth” images. On this dataset, we compare the  
 605 proposed method with the methods of BM3D [5], WNNM  
 606 [8], MLP [9], TRD [11], Noise Clinic (NC) [19], and Neat  
 607 Image (NI) [21]. We only compare the visual quality of  
 608 the denoised images. Figure 6 shows the denoised images  
 609 of “Dog” by the competing methods. More visual compari-  
 610 sons can be found in the supplementary file. It can be seen  
 611 that the methods of BM3D, WNNM tend to globally over-  
 612 smooth the image while locally remain some noise, while  
 613 the methods of MLP, TRD are likely to remain noise in the  
 614 whole image. This demonstrates that the methods designed  
 615 for Gaussian noise are not effective for removing the com-  
 616 plex noise in real noisy images. Though Noise Clinic and  
 617 Neat Image are specifically developed for removing com-  
 618 plex noise, they would sometimes fail to recover real noisy  
 619 images. However, our proposed method recoveries more  
 620 faithfully the structures and textures (such as the eye area)  
 621 than the other competing methods.

#### 622 4.4.2 Comparison on the Second Dataset [14]

623 The real noisy images in the second dataset [14] have cor-  
 624 responding “ground truth” images. On this dataset, we  
 625 firstly perform comparison on the 15 cropped images used  
 626 in [14]. The compared method are BM3D [5], WNNM [8],  
 627 MLP [9], CSF [10], TRD [11], Noise Clinic (NC) [19], and  
 628 Cross-Channel (CC) [14]. The PSNR values are listed in  
 629 Table 2. As we can see, on most (9 out of the 15) images  
 630 captured by different cameras and camera settings, our pro-  
 631 posed method obtains better PSNR values than the other  
 632 methods. Noted that, though in [14] a specific model is  
 633 trained for each camera and camera setting, our proposed  
 634 general method still gains 0.28dB improvements on PNSR  
 635 over [14]. We also compare the visual quality of the de-  
 636 noised images by the competing methods. Figure 7 shows

648  
649  
650  
651  
652  
653  
654  
655  
656

Figure 3. Denoised images of the 96-th cropped image from “Nikon D600 ISO 3200 C1” [14] by different methods. The images are better to be zoomed in on screen.

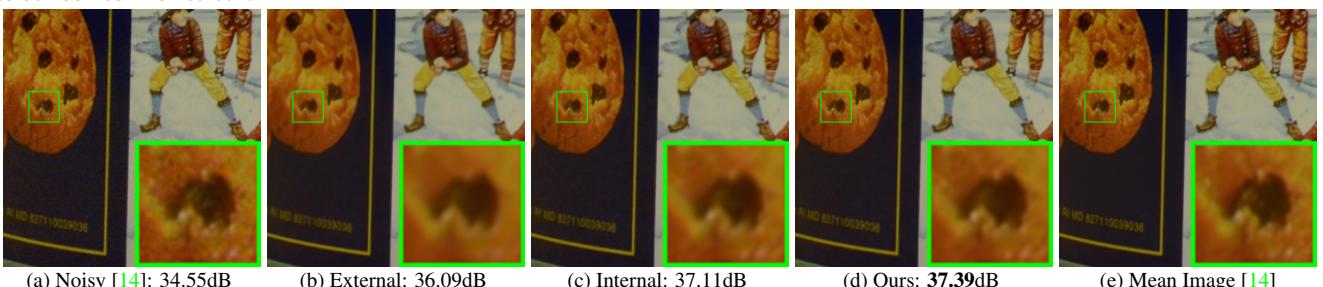
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667

Figure 4. Denoised images of the 94-th cropped image from “Nikon D600 ISO 3200 C1” [14] by different methods. The images are better to be zoomed in on screen.

670



Figure 5. Some samples cropped from real noisy images of [14].

the denoised images of a scene captured by Canon 5D Mark 3 at ISO = 3200 by the competing methods. More visual comparisons can be found in the supplementary file. We can see that BM3D, WNNM, NC, NI, and CC would either remain noise or generate artifacts, while MLP, TRD are likely to over-smooth the image. By combining the external and internal priors, our proposed method preserves edges and textures better than other methods.

To evaluate the compared methods on more samples, we then perform denoising experiments on the 60 smaller images cropped from the 17 images provided in [14]. The average PSNR results are listed in Table 3 (the code of [14] is not available so that it is not compared). The numbers in red color and blue color are the best and second best results, respectively. It can be seen that our proposed method achieves much better PSNR results than the other methods. The improvement of our method over the second best method (TRD) is 1dB. Due to the spacial limitations, the visual comparisons are provided in the supplementary file.

## 5. Conclusion and Future Work

Image priors are important for solving image denoising problems. The external priors learned from external clean

images are generally effective to most images, while the internal priors learned directly from the noisy image are adaptive to the given image but would be biased by the complex noise in real noisy images. In this paper, we demonstrate that, once unifying both the priors in external clean images and internal noisy images, we can achieve much better while still efficient performance on real image denoising problem. Specifically, the external patch group (PG) priors learned on natural clean images can be used to guide the subspace selection and orthogonal dictionary learning of internal noisy PGs from given noisy images. The experiments on real image denoising problem have demonstrated the powerful ability of the proposed method. In the future, we will speed up the proposed algorithm and evaluate the proposed method on other computer vision tasks such as image super-resolution.

702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755

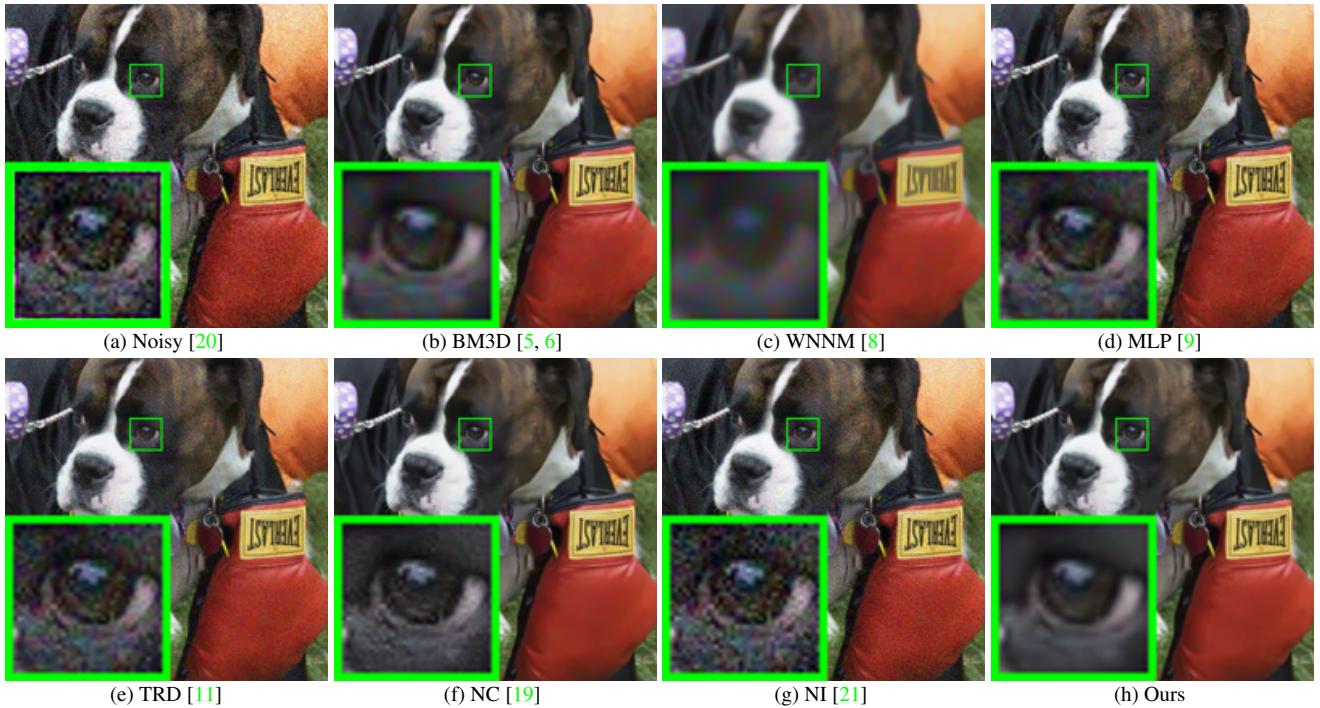


Figure 6. Denoised images of the image “Dog” by different methods. The images are better to be zoomed in on screen.

Table 2. Average PSNR(dB) results of different methods on 15 cropped real noisy images used in [14].

Camera Settings	Noisy	BM3D	WNNM	MLP	CSF	TRD	NI	NC	CC	Ours
Canon 5D Mark III ISO = 3200	37.00	37.08	37.09	33.92	35.68	36.20	37.68	<b>38.76</b>	38.37	<b>40.50</b>
	33.88	33.94	33.93	33.24	34.03	34.35	34.87	<b>35.69</b>	35.37	<b>37.05</b>
	33.83	33.88	33.90	32.37	32.63	33.10	34.77	<b>35.54</b>	34.91	<b>36.11</b>
Nikon D600 ISO = 3200	33.28	33.33	33.34	31.93	31.78	32.28	34.12	<b>35.57</b>	<b>34.98</b>	34.88
	33.77	33.85	33.79	34.15	35.16	35.34	35.36	<b>36.70</b>	35.95	<b>36.31</b>
	34.93	35.02	34.95	37.89	39.98	<b>40.51</b>	38.68	39.28	<b>41.15</b>	39.23
Nikon D800 ISO = 1600	35.47	35.54	35.57	33.77	34.84	35.09	37.34	<b>38.01</b>	37.99	<b>38.40</b>
	35.71	35.79	35.77	35.89	38.42	38.65	38.57	39.05	<b>40.36</b>	<b>40.92</b>
	34.81	34.92	34.95	34.25	35.79	35.85	37.87	38.20	<b>38.30</b>	<b>38.97</b>
Nikon D800 ISO = 3200	33.26	33.34	33.31	37.42	38.36	38.56	36.95	38.07	<b>39.01</b>	<b>38.66</b>
	32.89	32.95	32.96	34.88	35.53	35.76	35.09	35.72	<b>36.75</b>	<b>37.07</b>
	32.91	32.98	32.96	38.54	<b>40.05</b>	<b>40.59</b>	36.91	36.76	39.06	38.52
Nikon D800 ISO = 6400	29.63	29.66	29.71	33.59	34.08	<b>34.25</b>	31.28	33.49	<b>34.61</b>	33.76
	29.97	30.01	29.98	31.55	32.13	32.38	31.38	32.79	<b>33.21</b>	<b>33.43</b>
	29.87	29.90	29.95	31.42	31.52	31.76	31.40	32.86	<b>33.22</b>	<b>33.58</b>
Average	33.41	33.48	33.48	34.32	35.33	35.65	35.49	36.43	<b>36.88</b>	<b>37.16</b>

Table 3. Average PSNR(dB) results of different methods on 60 real noisy images cropped from [14].

Methods	BM3D	WNNM	MLP	CSF
PSNR	34.58	34.52	36.19	37.40
Methods	TRD	NI	NC	Ours
PSNR	<b>37.75</b>	36.53	37.57	<b>38.75</b>

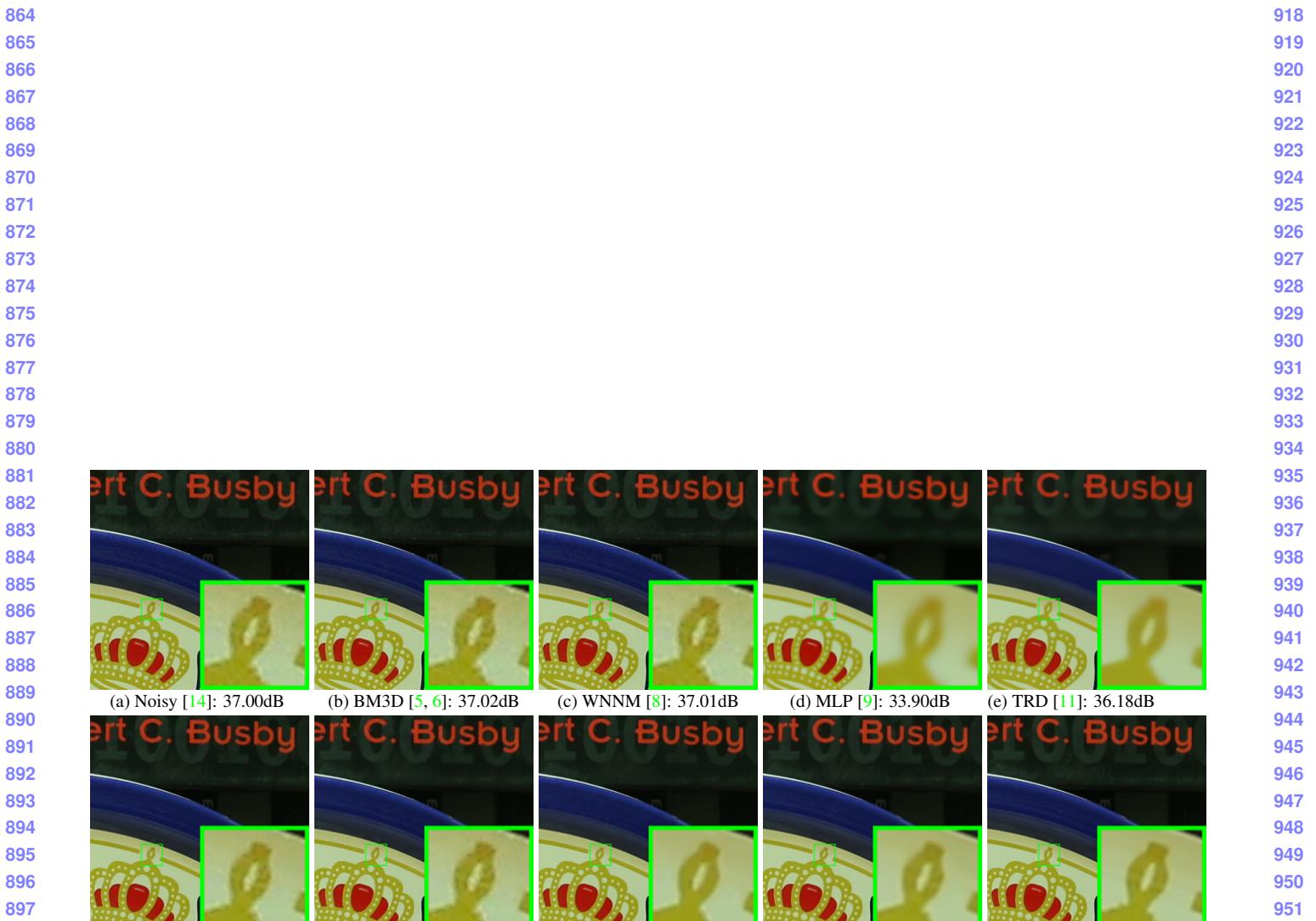


Figure 7. Denoised images of the image “Canon 5D Mark 3 ISO 3200 1” by different methods. The images are better to be zoomed in on screen.

972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025

## References

- [1] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. 1, 2
- [2] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. *IEEE International Conference on Computer Vision (ICCV)*, pages 2272–2279, 2009. 1
- [3] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013. 1, 2, 6
- [4] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 60–65, 2005. 1
- [5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 1, 2, 5, 6, 8, 9
- [6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. *IEEE International Conference on Image Processing (ICIP)*, pages 313–316, 2007. 1, 2, 5, 8, 9
- [7] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng. Patch group based nonlocal self-similarity prior learning for image denoising. *IEEE International Conference on Computer Vision (ICCV)*, pages 244–252, 2015. 1, 2, 3, 5, 6
- [8] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2862–2869, 2014. 1, 2, 5, 6, 8, 9
- [9] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2392–2399, 2012. 1, 2, 5, 6, 8, 9
- [10] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2774–2781, June 2014. 1, 2, 5, 6
- [11] Y. Chen, W. Yu, and T. Pock. On learning optimized reaction diffusion processes for effective image restoration. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5261–5269, 2015. 1, 2, 5, 6, 8, 9
- [12] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009. 1, 2
- [13] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. *IEEE International Conference on Computer Vision (ICCV)*, pages 479–486, 2011. 1, 2, 3
- [14] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1683–1691, 2016. 1, 2, 5, 6, 7, 8, 9
- [15] G. E Healey and R. Kondepudy. Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, 1994. 1
- [16] C. Liu, R. Szeliski, S. Bing Kang, C. L. Zitnick, and W. T. Freeman. Automatic estimation and removal of noise from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):299–314, 2008. 1, 2
- [17] Z. Gong, Z. Shen, and K.-C. Toh. Image restoration with mixed or unknown noises. *Multiscale Modeling & Simulation*, 12(2):458–487, 2014. 1, 2
- [18] F. Zhu, G. Chen, and P.-A. Heng. From noise modeling to blind image denoising. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2
- [19] M. Lebrun, M. Colom, and J.-M. Morel. Multiscale image blind denoising. *IEEE Transactions on Image Processing*, 24(10):3149–3161, 2015. 1, 2, 5, 6, 8, 9
- [20] M. Lebrun, M. Colom, and J. M. Morel. The noise clinic: a blind image denoising algorithm. <http://www.ipol.im/pub/art/2015/125/>. Accessed 01 28, 2015. 1, 2, 5, 6, 8
- [21] Neatlab ABSoft. Neat Image. <https://ni.neatvideo.com/home>. 1, 2, 5, 6, 8, 9
- [22] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity. *IEEE Transactions on Image Processing*, 21(5):2481–2499, 2012. 2
- [23] M. Zontak and M. Irani. Internal statistics of a single natural image. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 977–984, 2011. 2
- [24] M. Lebrun, A. Buades, and J. M. Morel. A nonlocal Bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013. 2
- [25] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862, 2001. 4
- [26] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006. 4
- [27] X. Liu, M. Tanaka, and M. Okutomi. Single-image noise level estimation for blind denoising. *IEEE transactions on Image Processing*, 22(12):5226–5237, 2013. 6
- 1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079