

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
068

Abstract

014

069

015

070
071

016

072
073

017

074
075

018

076
077

019

078
079

020

080
081

021

082
083

022

084
085

023

086
087

024

088
089

025

090
091

026

092
093

027

028

029

030

031

032

033

034

035

036

037

038

039

040

041

042

043

044

045

046

047

048

049

050

051

052

053

054

055

056

057

058

059

060

061

062

063

064

065

066

067

068

069

070

071

072

073

074

075

076

077

078

079

080

081

082

083

084

085

086

087

088

089

090

091

092

093

094

095

096

097

098

099

100

101

102

103

104

105

106

107

Anonymous CVPR submission

Paper ID 1047

012

013

014

015

016

017

018

019

020

021

022

023

024

025

026

027

028

029

030

031

032

033

034

035

036

037

038

039

040

041

042

043

044

045

046

047

048

049

050

051

052

053

054

055

056

057

058

059

060

061

062

063

064

065

066

067

068

069

070

071

072

073

074

075

076

077

078

079

080

081

082

083

084

085

086

087

088

089

090

091

092

093

094

095

096

097

098

099

100

101

102

103

104

105

106

107

External Prior Guided Internal Prior Learning for Real Noisy Image Denoising

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.

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. 2. 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-

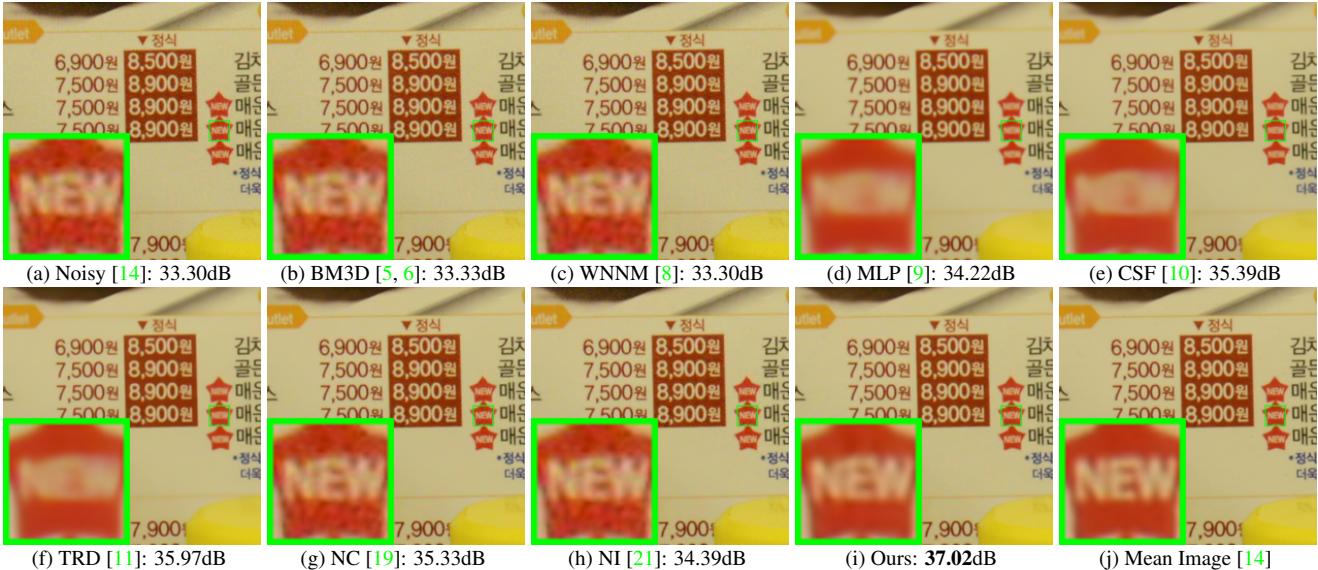


Figure 1. Denoised images of the real noisy image “Nikon D800 ISO 3200 A3” from [14] by different methods. The images are better viewed by zooming in on screen.

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

Recently, several denoising methods [16, 17, 19, 20, 18, 14] are proposed to remove unknown noise in noisy images.

These methods can be directly applied to denoise real noisy images. Among them, the “Noise Clinic” [19, 20] estimated the noise by a multivariate Gaussian and remove the noise by a generalized version of nonlocal Bayesian model [24]. Zhu *et al.* proposed a Bayesian method [18] to approximate and remove the noise via a low-rank MoG model. There are also several methods developed specifically for real noisy image denoising [14, 21]. The [14] modeled the cross-channel noise in real noisy image as a multivariate Gaussian and the noise is removed by Bayesian nonlocal means. The commercial software Neat Image [21] estimated the noise parameters from a flat region of the given noisy image and filtered the noise correspondingly. Almost all these methods [16, 17, 19, 20, 18, 14] use Gaussian or mixture of Gaussians (MoG) to model the noise in real noisy images. However, the noise in real noisy images is very complex and hard to be modeled by explicit distributions.

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

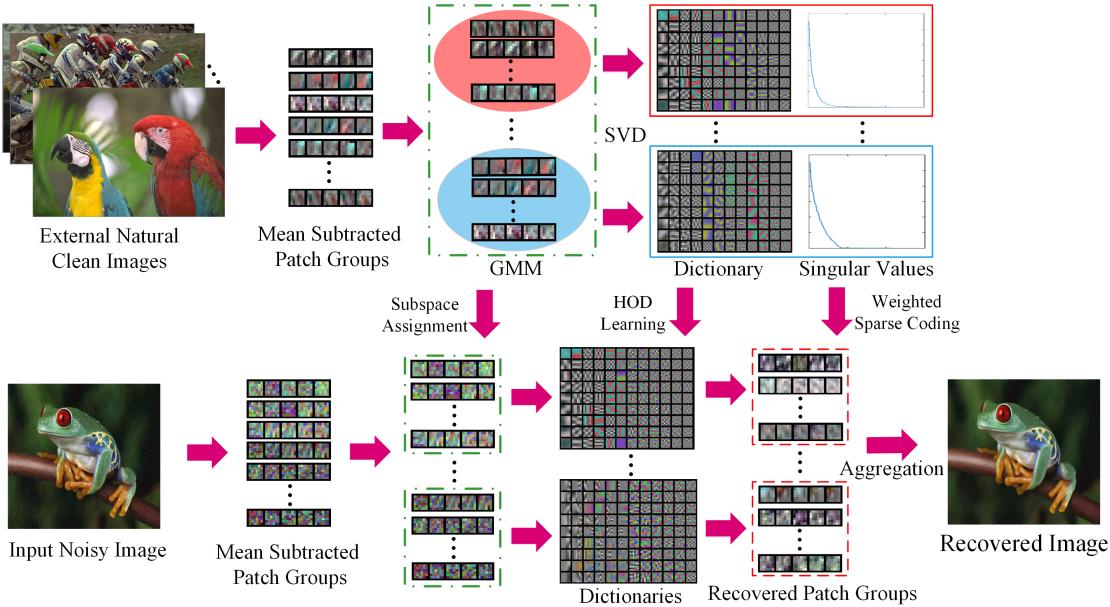


Figure 2. Flowchart of the proposed external prior guided internal prior learning and real noisy image denoising framework.

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 $\boldsymbol{\mu} = \frac{1}{M} \sum_{m=1}^M \mathbf{x}_m$, and the group mean subtracted PG is defined as $\bar{\mathbf{X}} \triangleq \{\bar{\mathbf{x}}_m = \mathbf{x}_m - \boldsymbol{\mu}\}$.

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

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

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 $\{\boldsymbol{\mu}_k\}_{k=1}^K$, and covariance matrices $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$. Note that the mean vector of each cluster is naturally zero, i.e., $\boldsymbol{\mu}_k = \mathbf{0}$.

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

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

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. 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 $\boldsymbol{\mu}_n$, is subtracted from each patch by $\bar{\mathbf{y}}_{n,m} = \mathbf{y}_{n,m} - \boldsymbol{\mu}_n$, 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 $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$ basically characterize the subspaces of natural high quality PGs. Therefore, we project the noisy PG $\bar{\mathbf{Y}}_n$ into the subspaces of $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$ and assign it to the most suitable subspace based on the posterior probability:

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

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

324

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:

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

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 $\alpha_{n,m}$. λ_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 ε 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

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

The orthogonality of internal dictionary $\mathbf{D}_i^{(t+1)}$ can be checked by $(\mathbf{D}_i^{(t+1)})^\top (\mathbf{D}_i^{(t+1)}) = \mathbf{V}_i \mathbf{U}_i^\top \mathbf{U}_i \mathbf{V}_i^\top = \mathbf{I}_r$.

3.3. The Denoising Algorithm

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

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

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

434 **Input:** Noisy image y , external PG prior GMM model

435 **Output:** The denoised image \hat{x} .

436 **Initialization:** $\hat{x}^{(0)} = y$;

437 **for** $Ite = 1 : IteNum$ **do**

438 1. Extracting internal PGs from $\hat{x}^{(Ite-1)}$;

439 **for** each PG \mathbf{Y}_n **do**

440 2. Calculate group mean vector μ_n and form
 mean subtracted PG $\bar{\mathbf{Y}}_n$;

441 3. Subspace selection via Eq. (3);

442 **end for**

443 **for** the PGs in each Subspace **do**

444 4. External PG prior Guided Internal Orthogonal
 Dictionary Learning by solving (4);

445 5. Recover each patch in all PGs via Eq. (11);

446 **end for**

447 6. Aggregate the recovered PGs of all subspaces to form
 the recovered image $\hat{x}^{(Ite)}$;

448 **end for**

449 **end for**

450 vectors $\{\hat{\alpha}_{n,m}^{(T-1)}\}$ in Eq. (8) and the orthogonal dictionary
 451 $\mathbf{D}_{(T)} = [\mathbf{D}_E \mathbf{D}_I^{(T)}]$ in Eq. (9), the latent clean patch of a
 452 noisy patch $\hat{y}_{n,m}$ in PG \mathbf{Y}_n is reconstructed as

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

453 where μ_n is the group mean of \mathbf{Y}_n . The latent clean image
 454 is then reconstructed by aggregating all the reconstructed
 455 patches in all PGs. We perform the above denoising pro-
 456 cedures for several iterations for better denoising outputs.
 457 The proposed denoising algorithm is summarized in Alg. 1.
 458 The latent clean image \hat{x} is reconstructed by aggregating all
 459 the estimated PGs. Similar to [7], we perform the above de-
 460 noising procedures for several iterations for better denoising
 461 outputs. The proposed denoising algorithm is summarized in
 462 Alg. 1.

4. Experiments

470 We evaluate the performance of the proposed algorithm
 471 on real-world noisy images [14, 20] in comparison with
 472 state-of-the-art denoising methods [5, 6, 9, 8, 10, 11, 14,
 473 19, 20, 21].

4.1. Implementation Details

478 Our proposed method has two stages: the external prior
 479 learning stage and the external prior guided internal prior
 480 learning stage. In the first stage, we set $p = 6$ (the
 481 patch size), $M = 10$ (the number of similar patches in
 482 a PG), $W = 31$ (the window size for PG searching) and
 483 $K = 32$ (the number of Gaussian components in GMM).
 484 We learn the external GMM prior with 3.6 million PGs ex-
 485 tracted from the Kodak PhotoCD Dataset (<http://r0k.us/>).



Figure 3. Some samples cropped from real noisy images of [14].

us/graphics/kodak/), which includes 24 high quality
 color images.

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

4.2. The Testing Datasets

We evaluate the proposed method on two real noisy im-
 age datasets, where the images were captured under indoor
 or outdoor lighting conditions by different types of cameras
 and camera settings.

The first dataset is provided in [20], which includes 20
 real noisy images collected under uncontrolled outdoor en-
 vironment. Since there is no “ground truth” of the noisy
 images, the objective measures such as PSNR cannot be
 computed on this dataset.

The second dataset is provided in [14], which includes
 noisy images of 17 static scenes. The noisy images were
 collected under controlled indoor environment. Each scene
 was shot 500 times under the same camera and camera set-
 ting. The mean image of the 500 shots is roughly taken as
 the “ground truth”, with which the PSNR can be computed.
 Since the image size is very large (about 7000×5000) and
 the 17 scenes share repetitive contents, the authors of [14]
 cropped 15 smaller images (of size 512×512) to perform
 experiments. To more comprehensively evaluate the pro-
 posed methods, we cropped 60 images of size 500×500
 from the dataset for experiments. Some samples are shown
 in Fig. 3. Note that the noise in our cropped 60 images
 is different from the noise in the 15 images cropped by the
 authors of [14] since they are from different shots.

4.3. Comparison among external, internal and guided internal priors

To demonstrate the advantages of external prior guided
 internal priors, in this section we perform real noisy image
 denoising by using external priors (denoted by “External”),
 internal priors (denoted by “Internal”), and the proposed

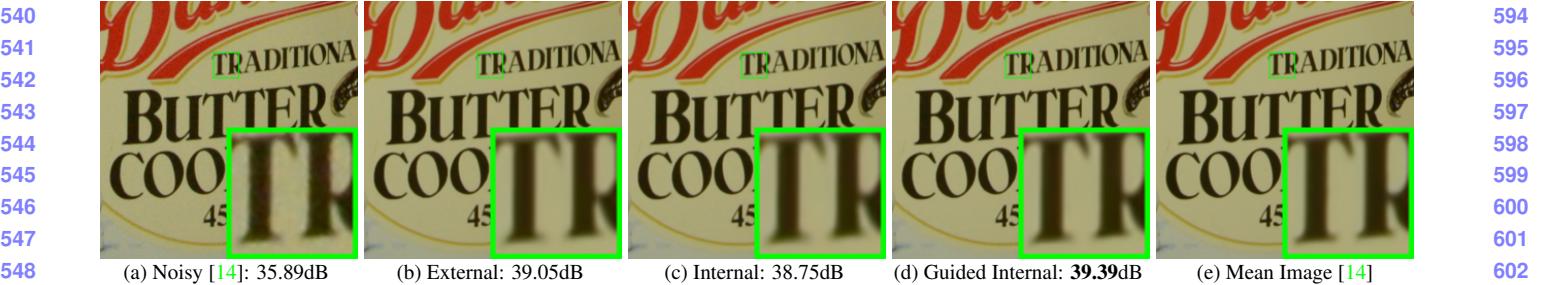


Figure 4. Denoised images of the 96-th cropped image from “Nikon D600 ISO 3200 C1” [14] by different methods. The images are better to be zoomed in on screen.

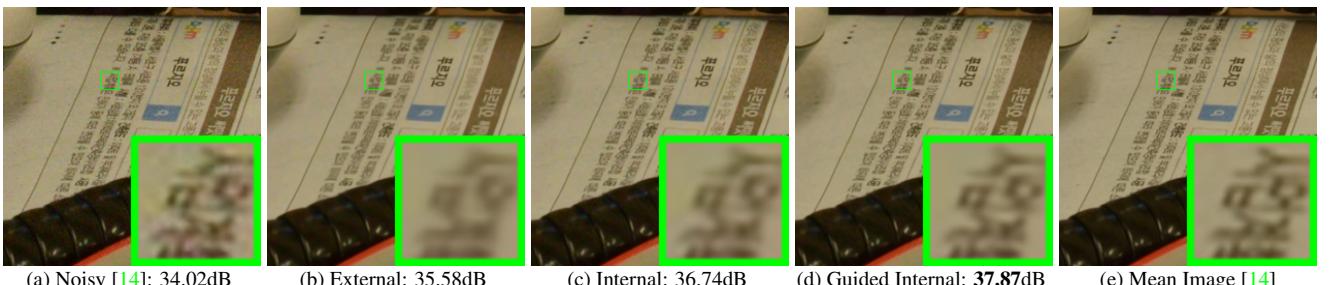


Figure 5. Denoised images of the 111-th cropped image from “Nikon D800 ISO 3200 A1” [14] by different methods. The images are better to be zoomed in on screen.

562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647

guided internal priors (denoted by “Guided Internal”), respectively. For the “External” method, we utilize the full external dictionaries (i.e., $r = 108$ in Eq. (4)) for denoising. For the “Internal” method, the overall framework is similar to the method of [3]. A GMM model (with $K = 32$ Gaussians) is directly learned from the PGs extracted from the given noisy image without using any external data, and then the internal orthogonal dictionary is obtained via Eqn. (2) to perform denoising. All parameters of the “External” and “Internal” methods are tuned to achieve their best performance.

We compare the three methods mentioned above on the 60 cropped images from [14]. The average PSNR and run time are listed in Table 1. It can be seen that “Guided Internal” prior achieves better PSNR than both “External” and “Internal” priors. In addition, the “Internal” method is very slow because it involves online GMM learning, while the “Guided Internal” method is only a little slower than the “External” method. Fig. 4 and Fig. 5 show the denoised images of two noisy image by the three methods. We can see that the “External” method is good at recovering large-scale structures (see Fig. 4) while the “Internal” method is good at recovering fine-scale textures (see Fig. 5). By utilizing external priors to guide the internal prior learning, our proposed method can effectively recover both the large-scale structures and fine-scale textures.

4.4. Comparison with State-of-the-Art Denoising Methods

We compare the proposed method with state-of-the-art image denoising methods, including CBM3D [5, 6],

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

	Noisy	External	Internal	Guided Internal
PSNR	34.51	38.21	38.07	38.75
Time	—	39.57	667.36	41.89

WNNM [8], MLP [9], CSF [10], TNRD [11], Noise Clinic (NC) [19], Cross-Channel (CC) [14], and Neat Image (NI) [21]. Among them, CBM3D, WNNM, MLP, CSF and TNRD are designed based on Gaussian noise model, and they need to know the noise level for denoising. We use the method in [27] to estimate the noise level for them. All the other parameters in these methods are set as the default ones. Since WNNM, MLP, CSF and TNRD are designed for grayscale images, we use them to denoise the R, G, B channels separately for color noisy images.

Like our method, the NC is a blind image denoising method which does not need any noise prior. The NI is a commercial software for image denoising, which has been embedded into Photoshop and Corel PaintShop [21]. The code of CC is not released but its results on the 15 cropped images are available at [14]. Therefore, we only compare with it on the 15 cropped images from [14].

4.4.1 Results on Dataset [20]

Since there is no “ground truth” for the real noisy images in dataset [20], we only compare the visual quality of the denoised images by different methods. (Note that method CC



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

[14] is not compared since its code is not available.) Fig. 7 shows the denoised images of “Dog”. It can be seen that CBM3D and WNNM tend to over-smooth much the image while remaining some noise caused color artifacts. MLP and TNRD are likely to remain many noise-caused color artifacts across the whole image. These results demonstrate that the methods designed with Gaussian noise model are not effective for real noise removal. Though NC and NI methods are specifically developed for real noisy images, their performance on noise removal is not very satisfactory. In comparison, our proposed method recovers much better the structures and textures (such as the eye area) than the other competing methods. More visual comparisons on dataset [20] can be found in the supplementary file.

4.4.2 Comparison on Dataset [14]

As described in section 4.2, there is a mean image for each of the 17 scenes used in dataset [14], and those mean images can be roughly taken as “ground truth” images for quantitative evaluation of denoising algorithms. We firstly perform quantitative comparison on the 15 cropped images used in [14]. The PSNR results of CBM3D [5], WNNM [8], MLP [9], CSF [10], TNRD [11], NC [19] and CC [14] are listed in Table 3. (The results of CC are copied from the original paper [14].) The best PSNR result of each image is highlighted in bold. One can see that on 9 out of the 15 images, our method achieves the best PSNR values. CC achieves the best PSNR on 3 of the 15 images. It should be noted

Table 2. 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	37.75	36.53	37.57	38.75

that in the CC method, a specific model is trained for each camera and camera setting, while our method uses the same model for all images. On average, our proposed method has 0.28dB PSNR improvements over [14] and much higher PSNR gains over other competing methods. Fig. ?? shows the denoised images of a scene captured by Canon 5D Mark 3 at ISO = 3200. We can see that CBM3D, WNNM, NC, NI, and CC would either remain noise or generate artifacts, while MLP, TNRD over-smooth much the image. By using the external prior guided internal priors, our proposed method preserves edges and textures better than other methods, leading to visually pleasant outputs. More visual comparisons can be found in the supplementary file.

We then perform denoising experiments on the 60 images we cropped from [14]. The average PSNR results are listed in Table 2 (CC is not compared since the code of [14] is not available). Again, our proposed method achieves much better PSNR results than the other methods. The improvement of our method over the second best method (TNRD) is about 1dB. Due to the space limitations, the visual comparisons are provided in the supplementary file.

Table 3. 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	38.76	38.37	40.50
	33.88	33.94	33.93	33.24	34.03	34.35	34.87	35.69	35.37	37.05
	33.83	33.88	33.90	32.37	32.63	33.10	34.77	35.54	34.91	36.11
Nikon D600 ISO = 3200	33.28	33.33	33.34	31.93	31.78	32.28	34.12	35.57	34.98	34.88
	33.77	33.85	33.79	34.15	35.16	35.34	35.36	36.70	35.95	36.31
	34.93	35.02	34.95	37.89	39.98	40.51	38.68	39.28	41.15	39.23
Nikon D800 ISO = 1600	35.47	35.54	35.57	33.77	34.84	35.09	37.34	38.01	37.99	38.40
	35.71	35.79	35.77	35.89	38.42	38.65	38.57	39.05	40.36	40.92
	34.81	34.92	34.95	34.25	35.79	35.85	37.87	38.20	38.30	38.97
Nikon D800 ISO = 3200	33.26	33.34	33.31	37.42	38.36	38.56	36.95	38.07	39.01	38.66
	32.89	32.95	32.96	34.88	35.53	35.76	35.09	35.72	36.75	37.07
	32.91	32.98	32.96	38.54	40.05	40.59	36.91	36.76	39.06	38.52
Nikon D800 ISO = 6400	29.63	29.66	29.71	33.59	34.08	34.25	31.28	33.49	34.61	33.76
	29.97	30.01	29.98	31.55	32.13	32.38	31.38	32.79	33.21	33.43
	29.87	29.90	29.95	31.42	31.52	31.76	31.40	32.86	33.22	33.58
Average	33.41	33.48	33.48	34.32	35.33	35.65	35.49	36.43	36.88	37.16

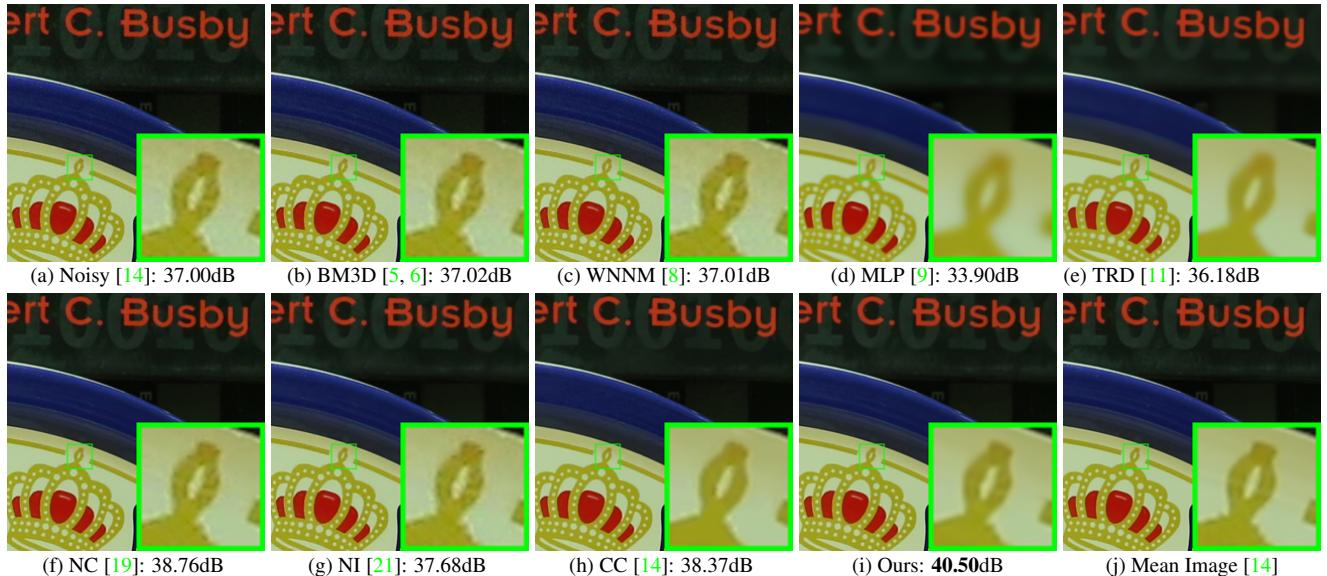


Figure 7. Denoised images of the image “Canon 5D Mark 3 ISO 3200 1” by different methods. The images are better to be zoomed in on screen.

5. Conclusion

The noise in real-world noisy images is much more complex than additive white Gaussian noise and hard to be modeled by simple analytical distributions such as Gaussian or mixture of Gaussians. In this paper, we proposed a simple yet effective solution for real noisy image denoising without explicitly assuming certain noise models. Specifically, we firstly learned image priors of Gausian Mixture Model 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 learned hybrid orthogonal dictionary are simutaneously utilized in real noisy

image denoising. Experiments on two real noisy image datasets, where the images were captured under indoor or outdoor lighting conditions by different types of cameras and camera settings, demonstrate that our proposed method achieved much better performance than state-of-the-art image denoising methods.

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

- 864 [2] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman.
865 Non-local sparse models for image restoration. *IEEE International Conference on Computer Vision (ICCV)*, pages
866 2272–2279, 2009. 1
- 867 [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
- 868 [4] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm
869 for image denoising. *IEEE Conference on Computer Vision
870 and Pattern Recognition (CVPR)*, pages 60–65, 2005. 1
- 871 [5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image
872 denoising by sparse 3-D transform-domain collaborative
873 filtering. *IEEE Transactions on Image Processing*,
874 16(8):2080–2095, 2007. 1, 2, 5, 6, 7, 8
- 875 [6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Color
876 image denoising via sparse 3D collaborative filtering with
877 grouping constraint in luminance-chrominance space. *IEEE
878 International Conference on Image Processing (ICIP)*, pages
879 313–316, 2007. 1, 2, 5, 6, 7, 8
- 880 [7] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng. Patch
881 group based nonlocal self-similarity prior learning for image
882 denoising. *IEEE International Conference on Computer Vision
883 (ICCV)*, pages 244–252, 2015. 1, 2, 3, 5
- 884 [8] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear
885 norm minimization with application to image denoising.
886 *IEEE Conference on Computer Vision and Pattern Recognition
887 (CVPR)*, pages 2862–2869, 2014. 1, 2, 5, 6,
888 7, 8
- 889 [9] H. C. Burger, C. J. Schuler, and S. Harmeling. Image
890 denoising: Can plain neural networks compete with BM3D?
891 *IEEE Conference on Computer Vision and Pattern Recognition
892 (CVPR)*, pages 2392–2399, 2012. 1, 2, 5, 6, 7, 8
- 893 [10] U. Schmidt and S. Roth. Shrinkage fields for effective image
894 restoration. *IEEE Conference on Computer Vision and Pattern
895 Recognition (CVPR)*, pages 2774–2781, June 2014. 1,
896 2, 5, 6, 7
- 897 [11] Y. Chen, W. Yu, and T. Pock. On learning optimized re-
898 action diffusion processes for effective image restoration.
899 *IEEE Conference on Computer Vision and Pattern Recognition
900 (CVPR)*, pages 5261–5269, 2015. 1, 2, 5, 6, 7, 8
- 901 [12] S. Roth and M. J. Black. Fields of experts. *International
902 Journal of Computer Vision*, 82(2):205–229, 2009. 1, 2
- 903 [13] D. Zoran and Y. Weiss. From learning models of natural
904 image patches to whole image restoration. *IEEE International
905 Conference on Computer Vision (ICCV)*, pages 479–
906 486, 2011. 1, 2, 3
- 907 [14] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim. A holistic
908 approach to cross-channel image noise modeling and its ap-
909 plication to image denoising. *IEEE Conference on Computer
910 Vision and Pattern Recognition (CVPR)*, pages 1683–1691,
911 2016. 1, 2, 5, 6, 7, 8
- 912 [15] G. E Healey and R. Kondepudy. Radiometric CCD cam-
913 era calibration and noise estimation. *IEEE Transactions on
914 Pattern Analysis and Machine Intelligence*, 16(3):267–276,
915 1994. 1
- 916 [16] C. Liu, R. Szeliski, S. Bing Kang, C. L. Zitnick, and W. T.
917 Freeman. Automatic estimation and removal of noise from
918 a single image. *IEEE Transactions on Pattern Analysis and
919 Machine Intelligence*, 30(2):299–314, 2008. 1, 2
- 920 [17] Z. Gong, Z. Shen, and K.-C. Toh. Image restoration with
921 mixed or unknown noises. *Multiscale Modeling & Simula-
922 tion*, 12(2):458–487, 2014. 1, 2
- 923 [18] F. Zhu, G. Chen, and P.-A. Heng. From noise modeling to
924 blind image denoising. *IEEE Conference on Computer Vi-
925 sion and Pattern Recognition (CVPR)*, June 2016. 1, 2
- 926 [19] M. Lebrun, M. Colom, and J.-M. Morel. Multiscale image
927 blind denoising. *IEEE Transactions on Image Processing*,
928 24(10):3149–3161, 2015. 1, 2, 5, 6, 7, 8
- 929 [20] M. Lebrun, M. Colom, and J. M. Morel. The noise clinic:
930 a blind image denoising algorithm. [http://www.ipol.
931 im/pub/art/2015/125/](http://www.ipol.im/pub/art/2015/125/). Accessed 01 28, 2015. 1, 2,
932 5, 6, 7
- 933 [21] Neatlab ABSsoft. Neat Image. [https://ni.
934 neatvideo.com/home](https://ni.neatvideo.com/home). 1, 2, 5, 6, 7, 8
- 935 [22] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems
936 with piecewise linear estimators: From Gaussian mixture
937 models to structured sparsity. *IEEE Transactions on Image
938 Processing*, 21(5):2481–2499, 2012. 2
- 939 [23] M. Zontak and M. Irani. Internal statistics of a single natural
940 image. *IEEE Conference on Computer Vision and Pattern
941 Recognition (CVPR)*, pages 977–984, 2011. 2
- 942 [24] M. Lebrun, A. Buades, and J. M. Morel. A nonlocal
943 Bayesian image denoising algorithm. *SIAM Journal on
944 Imaging Sciences*, 6(3):1665–1688, 2013. 2
- 945 [25] D. L. Donoho and X. Huo. Uncertainty principles and ideal
946 atomic decomposition. *IEEE Transactions on Information
947 Theory*, 47(7):2845–2862, 2001. 4
- 948 [26] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal com-
949 ponent analysis. *Journal of Computational and Graphical
950 Statistics*, 15(2):265–286, 2006. 4
- 951 [27] X. Liu, M. Tanaka, and M. Okutomi. Single-image noise
952 level estimation for blind denoising. *IEEE transactions on
953 Image Processing*, 22(12):5226–5237, 2013. 6
- 954 [28] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman.
955 Non-local sparse models for image restoration. *IEEE International Conference on Computer Vision (ICCV)*, pages
956 2272–2279, 2009. 1
- 957 [29] C. Liu, R. Szeliski, S. Bing Kang, C. L. Zitnick, and W. T.
958 Freeman. Automatic estimation and removal of noise from
959 a single image. *IEEE Transactions on Pattern Analysis and
960 Machine Intelligence*, 30(2):299–314, 2008. 1, 2
- 961 [30] M. Lebrun, M. Colom, and J.-M. Morel. Multiscale image
962 blind denoising. *IEEE Transactions on Image Processing*,
963 24(10):3149–3161, 2015. 1, 2, 5, 6, 7, 8
- 964 [31] M. Lebrun, M. Colom, and J. M. Morel. The noise clinic:
965 a blind image denoising algorithm. [http://www.ipol.
967 im/pub/art/2015/125/](http://www.ipol.
966 im/pub/art/2015/125/). Accessed 01 28, 2015. 1, 2,
968 5, 6, 7
- 969 [32] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems
970 with piecewise linear estimators: From Gaussian mixture
971 models to structured sparsity. *IEEE Transactions on Image
972 Processing*, 21(5):2481–2499, 2012. 2
- 973 [33] M. Zontak and M. Irani. Internal statistics of a single natural
974 image. *IEEE Conference on Computer Vision and Pattern
975 Recognition (CVPR)*, pages 977–984, 2011. 2
- 976 [34] M. Lebrun, A. Buades, and J. M. Morel. A nonlocal
977 Bayesian image denoising algorithm. *SIAM Journal on
978 Imaging Sciences*, 6(3):1665–1688, 2013. 2
- 979 [35] D. L. Donoho and X. Huo. Uncertainty principles and ideal
980 atomic decomposition. *IEEE Transactions on Information
981 Theory*, 47(7):2845–2862, 2001. 4
- 982 [36] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal com-
983 ponent analysis. *Journal of Computational and Graphical
984 Statistics*, 15(2):265–286, 2006. 4
- 985 [37] X. Liu, M. Tanaka, and M. Okutomi. Single-image noise
986 level estimation for blind denoising. *IEEE transactions on
987 Image Processing*, 22(12):5226–5237, 2013. 6
- 988 [38] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman.
989 Non-local sparse models for image restoration. *IEEE International Conference on Computer Vision (ICCV)*, pages
990 2272–2279, 2009. 1
- 991 [39] C. Liu, R. Szeliski, S. Bing Kang, C. L. Zitnick, and W. T.
992 Freeman. Automatic estimation and removal of noise from
993 a single image. *IEEE Transactions on Pattern Analysis and
994 Machine Intelligence*, 30(2):299–314, 2008. 1, 2
- 995 [40] M. Lebrun, M. Colom, and J.-M. Morel. Multiscale image
996 blind denoising. *IEEE Transactions on Image Processing*,
997 24(10):3149–3161, 2015. 1, 2, 5, 6, 7, 8