TURNING A DENOISER INTO A SUPER-RESOLVER USING PLUG AND PLAY PRIORS

Alon Brifman[⋆] *Yaniv Romano*[‡] *Michael Elad*[⋆]

[⋆]Department of Computer Science, Technion, Haifa 32000, Israel.
[‡]Department of Electrical Engineering, Technion, Haifa 32000, Israel.

ABSTRACT

Denoising and Super-Resolution are two inverse problems that have been extensively studied. Over the years, these two tasks were treated as two distinct problems that deserve a different algorithmic solution. In this paper we wish to exploit the recently introduced Plug-and-Play Prior (PPP) approach [1] to connect between the two. Using the PPP, we turn leading denoisers into super-resolution solvers. As a case-study we demonstrate this on the NCSR algorithm, which has two variants: one for denoising and one for super-resolution. We show that by using the NCSR denoiser, one can get equal or even better results when compared with the NCSR super-resolution.

Index Terms**—** Single image super-resolution, plug-and-play, NCSR, ADMM, image denoising.

1. INTRODUCTION

Single Image Super-Resolution (SISR) [2–9] is the process of recovering a High-Resolution (HR) image $\mathbf{x} \in \mathbf{R}^{\mathbf{sM \times sN}}$ from its blurred, decimated, and noisy Low-Resolution (LR) measurement $\mathbf{y} \in \mathbf{R}^{\mathbf{M \times N}}$. This linear degradation model can be expressed by

$$\mathbf{y} = \mathbf{SHx} + \eta, \tag{1}$$

where the matrix $\mathbf{H} \in \mathbf{R}^{\mathbf{s^2MN \times s^2MN}}$ blurs the original image $\mathbf{x}$, the down-sampling operator $\mathbf{S} \in \mathbf{R}^{\mathbf{MN \times s^2MN}}$ decimates the blurred image in a factor of $\mathbf{s}$ along the horizontal and vertical dimensions, and $\eta \sim N(0, \mathbf{I}\sigma^2) \in \mathbf{R}^{\mathbf{M \times N}}$ is an additive zero-mean white Gaussian noise. Note that $\mathbf{x}$, $\mathbf{y}$ and η are held as column vectors, after lexicographic ordering.

A naive solution to the SISR problem is based on the Maximum-Likelihood (ML), targeting the minimization of the log-likelihood expression,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{SHx} - \mathbf{y}\|_2^2, \tag{2}$$

where $\hat{\mathbf{x}} \in \mathbf{R}^{\mathbf{sM \times sN}}$ is an estimation of the unknown underlying image $\mathbf{x}$. Clearly, this approach is bound to fail since this problem has infinitely many possible solutions. In the last two decades, various regularizers/priors have been proposed (e.g. [10–15]) to fix this ill-posed inverse problem, aiming at producing an estimation that ”behaves like a natural image”. These priors are plugged into Equation (2) and form the following penalized ML (or MAP) minimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2} \|\mathbf{SHx} - \mathbf{y}\|_2^2 + \beta R(\mathbf{x}), \tag{3}$$

where σ is the standard-deviation of the noise and $R(\cdot)$ is an image prior, obtaining small values for ”well-behaved” images and large values for unlikely ones. The parameter β controls the importance of the prior compared to the data fidelity term.

A closely related problem to SISR is image denoising [2, 7, 9, 14, 16–22]. Its degradation model is a simpler version of Equation (1), given by

$$\mathbf{y} = \mathbf{x} + \eta, \tag{4}$$

i.e., neither blur nor decimation are applied on the original image. In the this case, the solution of the ML is trivial (the minimizer of Equation (2) when $\mathbf{SH} = \mathbf{I}$ is $\hat{\mathbf{x}} = \mathbf{y}$), while regularization-based methods promote non-trivial solutions, expressed by

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \beta R(\mathbf{x}). \tag{5}$$

Notice that the priors in Equations (3) and (5) can be the same. Furthermore, they are blind to the specific inverse problem that we are trying to solve. Yet, unfortunately, although these two inverse problems look very much alike, SISR methods usually modify the priors that are used for the denoising problem and take into consideration the different (and more complicated) degradation model (e.g. as done in NCSR [7]). Furthermore, most of the state-of-the-art denoising algorithms (e.g. [7, 16, 17]) are very complex and sophisticated, thus adapting their core denoising concepts to solve the SISR problem is far from trivial.

Luckily, as will be shown hereafter, using the Plug-and-Play scheme [1], one may recast the SISR problem (i.e. Equation (3)) as a series of denoising problems, using the very same prior without any modifications. Put differently, PPP enables to choose any denoiser that solves Equation (5) and use it as a ”black box” in order to solve Equation (3), leading to similar and even better estimation of the HR image.

This paper is organized as follows: In Section 2 we derive and describe the proposed SISR scheme, along with the description of the PPP scheme. Experiments are brought in Section 3, demonstrating the effectiveness of our method. Conclusions are given in Section 4.

2. THE PROPOSED ALGORITHM

Inspired by the PPP scheme [1], in this section we describe how to translate Equation (3) to a sequence of denoising problems, which minimize a much simpler function – Equation (5). We follow very closely the developments of the PPP [1] and ADMM [23] algorithms, while emphasizing the relation to the SISR, both for noisy and noiseless scenarios.

2.1. Plug and Play Denoisers for Super-Resolution

Using the variable splitting concept, we can separate the l_2 data fidelity term in Equation (3) from the prior by splitting the variable $\mathbf{x}$ in Equation (3) into two variables $\mathbf{x}$ and $\mathbf{v}$, and adding a new constraint $\mathbf{x} = \mathbf{v}$, leading to

$$\begin{aligned} \hat{\mathbf{x}} = &\arg\min_{\mathbf{x}} \frac{1}{2\sigma^2}\|\mathbf{SHx} - \mathbf{y}\|_2^2 + \beta R(\mathbf{v}) \\ &\text{s.t.} \quad \mathbf{x} = \mathbf{v}. \end{aligned} \tag{6}$$

Solving the above constrained optimization problem can be done by formulating it as an Augmented Lagrangian function, minimized in an unconstrained manner. The Lagrangian is given by

$$L_{\rho,\lambda}(\mathbf{x},\mathbf{v}) = \frac{1}{2\sigma^2}\|\mathbf{SHx} - \mathbf{y}\|_2^2 + \beta R(\mathbf{v}) + g_{\rho,\lambda}(\mathbf{x} - \mathbf{v}), \tag{7}$$

where $g_{\rho,\lambda}$ is a penalty function that becomes the ideal penalty function for $\rho \to \infty$ and obeys $g'_{\rho,\lambda}(0) = \lambda \in \mathbf{R}^{\mathbf{s}^2\mathbf{MN}}$. We choose the commonly used penalty

$$g_{\rho,\lambda}(t) = \frac{\rho}{2}\|t\|_2^2 + \lambda^T t, \tag{8}$$

implying that the larger the distance between $\mathbf{x}$ and $\mathbf{v}$, the higher the penalty.

As a consequence, Equation (7) can be iteratively minimized by repeating the following steps:

$$\mathbf{x}^{k+1}, \mathbf{v}^{k+1} = \arg\min_{\mathbf{x},\mathbf{v}} L_{\rho^k,\lambda^k}(\mathbf{x},\mathbf{v}) \tag{9}$$

$$\lambda^{k+1} = g'_{\rho^k,\lambda^k}(0) \tag{10}$$

$$\rho^{k+1} = \alpha\rho^k \tag{11}$$

First, for fixed ρ and λ we minimize $L_{\rho,\lambda}(\mathbf{x},\mathbf{v})$ by alternating between the minimization of $\mathbf{x}$ and $\mathbf{v}$, leading to the updated images $\mathbf{x}^{k+1}$ and $\mathbf{v}^{k+1}$. Then, in order to comply with the constraint $\mathbf{x} = \mathbf{v}$, we tighten the penalty function $g_{\rho,\lambda}$ by updating its parameters ρ and λ according to Equations (10) and (11), respectively.

More specifically, following the ADMM approach [23], we introduce the *scaled* dual variable $\mathbf{u} = \frac{1}{\rho}\lambda$ and obtain the following steps to repeat:

$$\mathbf{u} = \frac{1}{\rho}\lambda^k \tag{12}$$

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2}\|\mathbf{SHx} - \mathbf{y}\|_2^2 + \frac{\rho^k}{2}\|\mathbf{x} - \mathbf{v}^k + \mathbf{u}^k\|_2^2 \tag{13}$$

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \frac{\rho^k}{2}\|\mathbf{x}^{k+1} - \mathbf{v} + \mathbf{u}^k\|_2^2 + \beta R(\mathbf{v}) \tag{14}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{x}^{k+1} - \mathbf{v}^{k+1} \tag{15}$$

$$\lambda^{k+1} = \rho^k \mathbf{u}^{k+1} \tag{16}$$

$$\rho^{k+1} = \alpha\rho^k \tag{17}$$

Notice that Equation (13) is Quadratic in $\mathbf{x}$ and thereby can be solved analytically, and Equation (14) can be re-written as

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \frac{1}{2\left(\frac{1}{\sqrt{\rho^k}}\right)^2}\|\mathbf{v} - \widetilde{\mathbf{v}}\|_2^2 + \beta R(\mathbf{v}), \tag{18}$$

where $\widetilde{\mathbf{v}} = \mathbf{x}^{k+1} + \mathbf{u}^k$. As such, Equation (18) is nothing but a denoising problem, aiming at cleaning the noisy image $\widetilde{\mathbf{v}}$, contaminated with noise-level $\sigma = 1/\sqrt{\rho}$. Since most of the denoisers expect only σ as an input parameter, we suggest normalizing Equation (18), results in $\sigma = \sqrt{\beta/\rho}$. Algorithm 1 summarizes the proposed SR scheme.

2.2. The (Nearly-) Noiseless Case

As the above scheme deals with the noisy case, handling the noiseless case can be done by setting $\sigma \to 0$. However, in this setting, Equation (13) may suffer from numerical errors and instabilities. Instead, it is more natural to formulate the data fidelity term as a constraint by

$$\begin{aligned} \mathbf{x}^{k+1} = &\arg\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{v}^k + \mathbf{u}^k\|_2^2 \\ &\text{s.t.} \quad \mathbf{SHx} = \mathbf{y}, \end{aligned} \tag{19}$$

which can be solved using Lagrange multipliers:

$$L(\mathbf{x}, d) = \frac{1}{2}\|\mathbf{x} - \mathbf{v}^k + \mathbf{u}^k\|_2^2 + d^T(\mathbf{SHx} - \mathbf{y}), \tag{20}$$

where $d \in \mathbf{R}^{\mathbf{M}\times\mathbf{N}}$. Taking the derivative of this Lagrangian w.r.t. $\mathbf{x}$ and demanding it to be zero leads to the following system:

$$\begin{aligned} (\mathbf{x} - \mathbf{v}^k + \mathbf{u}^k) + (\mathbf{SH})^T d = 0 \\ \mathbf{SHx} = \mathbf{y}, \end{aligned} \tag{21}$$

```
Algorithm 1: The proposed SISR solver
  Input: $\mathbf{y}$ – LR image;
         $D(\mathbf{z}, \sigma_{\mathbf{z}})$ – A denoiser, receiving an image $\mathbf{z}$
         and its noise-level $\sigma_z$;
         $\sigma$ – The s.t.d. of the noise in $\mathbf{y}$;
         $\mathbf{S}$ – A decimation matrix;
         $\mathbf{H}$ – A blur matrix;
         $K$ – Number of iterations.
  Init: $\lambda^0 = 0$;
        $\mathbf{x}^0 = \mathbf{v}^0$ a bicubic interpolation of the image $\mathbf{y}$;
        $\mathbf{L} = \frac{1}{\sigma^2}(\mathbf{SH})^T\mathbf{SH}$
  for $k = 0 : K$ do
      $u = \frac{\lambda^k}{\rho^k}$
      $\mathbf{x}^{k+1} = \mathbf{L}^{-1}\left(\frac{1}{\sigma^2}(\mathbf{SH})^T\mathbf{y} + \rho^k(\mathbf{v}^k - \mathbf{u})\right)$
      $\mathbf{v}^{k+1} = D\left(\mathbf{x}^{k+1} + \mathbf{u}^k, \sqrt{\frac{\beta}{\rho^k}}\right)$
      $\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{x}^{k+1} - \mathbf{v}^{k+1}$
      $\lambda^{k+1} = \rho^k\mathbf{u}$
      $\rho_{k+1} = \alpha\rho^k$
  end
  Output: $\mathbf{v}^K$, the super-resolved version of $\mathbf{y}$.
```

where the second equation is the constraint. Separating $\mathbf{x}$ from the first equation results in

$$\mathbf{x} = \mathbf{v}^k - \mathbf{u}^k - (\mathbf{SH})^T d, \tag{22}$$

and assigning it back into the constraint leads to

$$\mathbf{SH}(\mathbf{SH})^T d = \mathbf{SH}(\mathbf{v}^k - \mathbf{u}) - \mathbf{y}. \tag{23}$$

The variable d can be obtained by solving Equation (23), e.g., using conjugate-gradient. Then, the image $\mathbf{x}^{k+1}$ is computed by assigning d in Equation (22).

2.3. Penalty Parameter Update

In general, by incrementing the penalty parameter ρ through the iterations we force the variables/images $\mathbf{x}$ and $\mathbf{v}$ to be identical. The parameter ρ also affects the noise-level since $\sigma = \sqrt{\beta/\rho}$, i.e., it controls the strength of the denoiser – the larger the ρ the more conservative the denoiser. In our implementation, we update ρ in an adaptive manner, according to the value of $\|\rho(\mathbf{v}^k - \mathbf{v}^{k-1})\|_2^2$, which can be considered as an estimation to the dual feasibility [23]. Since the dual feasibility should decrease during the iterations, our strategy is to increase ρ by a factor of α (in our tests $\alpha = 1.2$) as long as this measure decreases. If the dual feasibility increases for several iterations, we decrease ρ and update the dual variable accordingly.

3. EXPERIMENTAL RESULTS

In this section, detailed results of the proposed algorithm are presented for various images, scaling factors and noise levels.

3.1. Testing Setup

We tested our algorithm with the state-of-the-art NCSR method [7], which has variants both for SISR and for denoising. NCSR combines the sparse-coding and dictionary learning [24] with the nonlocal self-similarity assumption [14]. These two priors are very powerful, leading to efficient reconstruction of the underlying image. We find it interesting to evaluate the effectiveness of our approach by plugging the NCSR denoiser in Algorithm 1 and compare the results with the original NCSR-SISR solver.

In order to have a fair comparison, we tested these two algorithms on the same dataset, supplied by the authors of NCSR. The LR images are generated by blurring the HR database images with Gaussian kernel with standard-deviation of 1.6, followed by decimation in a factor of s in each axis. In the noisy cases, we contaminated the LR images by an additive Gaussian noise with standard-deviation σ.

The restoration performance is evaluated based on the Peak Signal to Noise Ratio (PSNR) between the luminance channel of the original and the restored images. Note that a LR color image is upscaled by (i) converting it to YCbCr color space, (ii) upscaling the luminance channel using our method, while the chromatic channels are upscaled using the bicubic interpolator, and (iii) converting the restored HR channels back to the RGB color space.

3.2. Choice of Parameters

We conducted several tests in order to tune the parameters of the proposed algorithm, leading to the following setting:

$$\rho_0 = 0.0001, \quad \beta = 2^{11}\rho_0, \quad \alpha = 1.2, \quad K = 35.$$

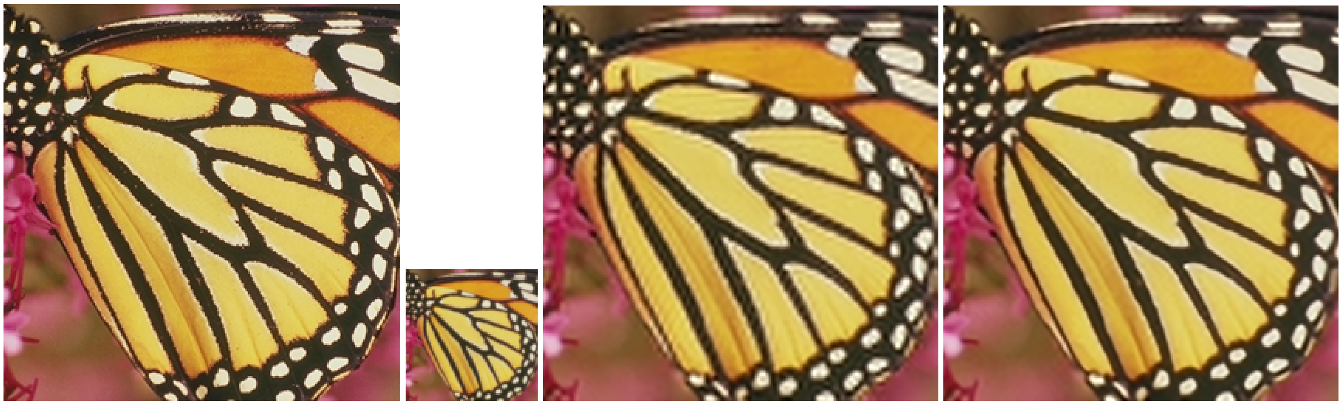
Also, for $\sigma < 0.5$ we use the noiseless modification depicted in Section 2.2. As for the denoiser itself, we treat it as a ”black-box” and use the very same parameters as in the original NCSR software. In order to reduce computations, since the NCSR denoiser is an iterative algorithm, we apply it only for $1/3$ of the number of iterations that are set by the authors. Generally speaking, the proposed approach is about 10 times slower than the SISR-NCSR method. Note that we did not put emphasis on runtime when choosing the parameters.

3.3. Results

In Table 1 we provide a comparison (in terms of PSNR) between the bicubic, NCSR-SISR, and the proposed algorithm. As can be seen, in most cases we achieve superior results than the NCSR-SISR, and in all cases we outperform the bicubic. Quantitatively, compared to NCSR-SISR, for the noiseless ($\sigma = 0$) and small noise-levels of 0.5, 1, and 2, we gain an average improvement of 0.12dB, 1.05dB, 0.72dB, and 0.23dB, respectively. For higher noise-levels ($\sigma = 5$ and $\sigma = 10$), the proposed approach and NCSR-SISR performs quite the same. A visual comparison is given in Fig.1 for Butterfly

Table 1: Comparison between the denoising results [PSNR] of the bicubic, original NCSR-SISR algorithm [7] and the proposed approach. The best results per each image, noise-level and scaling factor are highlighted.

Images	Parrots	Raccoon	Flower	Hat	Bike	Parthen.	Butterfly	Plants	Girl	**Average**
$s = 4, \sigma = 0$										
Bicubic	24.36	25.34	23.48	26.05	19.71	23.00	19.27	26.27	28.49	23.99
NCSR	27.66	27.74	**26.99**	29.58	22.77	25.72	25.16	31.02	32.28	27.66
Ours	**27.88**	**27.86**	26.85	**29.64**	**22.86**	**25.84**	**25.55**	**31.17**	**32.32**	**27.78**
$s = 8, \sigma = 0.5$										
Bicubic	21.43	22.35	20.72	23.14	17.09	19.93	15.64	22.90	24.87	20.90
NCSR	22.64	23.73	21.71	24.25	17.93	21.01	16.63	24.05	26.49	22.05
Ours	**23.33**	**25.17**	**22.56**	**24.99**	**18.83**	**21.38**	**17.71**	**25.47**	**28.48**	**23.10**
$s = 6, \sigma = 1$										
Bicubic	22.45	23.66	21.65	24.36	18.02	21.35	16.99	24.24	26.32	22.11
NCSR	24.67	25.88	23.62	26.70	19.62	23.38	19.17	26.80	29.49	24.37
Ours	**25.12**	**25.97**	**24.08**	**27.60**	**20.32**	**23.58**	**20.84**	**27.96**	**30.35**	**25.09**
$s = 8, \sigma = 2$										
Bicubic	25.53	26.38	24.78	27.15	20.78	24.09	20.76	27.79	29.87	25.24
NCSR	29.97	28.82	28.89	30.75	24.22	26.87	27.12	32.94	**33.02**	29.18
Ours	**30.37**	**28.93**	**29.04**	**30.85**	**24.44**	**27.03**	**27.77**	**33.19**	33.01	**29.40**
$s = 4, \sigma = 5$										
Bicubic	24.27	25.22	23.37	25.89	19.66	22.92	19.23	26.12	28.27	23.88
NCSR	27.47	27.22	**26.49**	28.89	**22.60**	25.45	24.85	**30.07**	**31.22**	27.14
Ours	**27.52**	**27.29**	26.45	**28.93**	22.57	**25.50**	**24.91**	**30.07**	31.21	**27.16**
$s = 3, \sigma = 10$										
Bicubic	25.05	25.81	24.38	26.54	20.62	23.75	20.59	27.03	28.75	24.72
NCSR	28.06	27.20	**26.85**	**28.73**	**23.02**	25.67	**25.63**	**29.98**	30.67	**27.31**
Ours	**28.08**	**27.23**	26.79	28.68	22.83	**25.71**	25.32	29.94	**30.72**	27.25



(a) Original (b) LR (c) NCSR-SISR (d) Ours

Fig. 1: Visual comparison for Butterfly image, when $s = 3$ and $\sigma = 0.5$.

image, showing that our approach results in less aliasing and ringing artifacts than the NCSR-SISR method, supporting the gain in PSNR.

4. CONCLUSIONS

In this paper, using the brilliant Plug-and-Play Priors framework [1], we presented an algebraic method which turns any ”black-box” denoiser into a SISR solver. The proposed algorithm is tested on various images, scaling factors, and noise levels. The effectiveness of the proposed algorithm is demonstrated on NCSR [7], which has two variants – one for denoising and the other for SISR. More specifically, we have shown that plugging the NCSR denoiser in our SR scheme outperforms its own variant for SISR, leading to state-of-the-art results.

5. REFERENCES

[1] S. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *IEEE Global Conference on Signal and Information Processing*, 2013, pp. 945–948.

[2] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. on Image Processing*, vol. 16, no. 2, pp. 349–366, 2007.

[3] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Int. Conf. on Computer Vision*. IEEE, 2009, pp. 349–356.

[4] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–10, 2010.

[5] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: from Gaussian mixture models to structured sparsity," *IEEE Trans. on Image Processing*, vol. 21, no. 5, pp. 2481–2499, 2012.

[6] L. Sun and J. Hays, "Super-resolution from internet-scale scene matching," in *Int. Conf. on Computational Photography*. IEEE, 2012, pp. 1–12.

[7] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, April 2013.

[8] Y. Romano, M. Protter, and M. Elad, "Single image interpolation via adaptive nonlocal sparsity-based modeling," *IEEE Trans. on Image Processing*, vol. 23, no. 7, pp. 3085–3098, 2014.

[9] V. Papyan and M. Elad, "Multi-scale patch-based image restoration," *IEEE Trans. on Image Processing*, vol. 25, no. 1, pp. 249–261, 2016.

[10] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, pp. 259–268, Nov. 1992.

[11] R. Coifman and M. Wickerhauser, "Adapted waveform analysis as a tool for modeling, feature extraction, and denoising," *Optical Engineering*, vol. 33, no. 7, pp. 2170 – 2174, July 1994.

[12] M. Do and M. Vetterli, "The finite ridgelet transform for image representation," *IEEE Trans. on Image Processing*, vol. 12, no. 1, pp. 16–28, 2003.

[13] M. Aharon, M. Elad, and A. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[14] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2005, pp. 60–65.

[15] D. Zoran and Y. Weiss, "Natural images, gaussian mixtures and dead leaves," in *Advances in Neural Information Processing Systems*, 2012, pp. 1736–1744.

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

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

[18] Y. Romano and M. Elad, "Improving K-SVD denoising by post-processing its method-noise," in *IEEE Int. Conf. on Image Processing*. IEEE, 2013, pp. 435–439.

[19] A. Kheradmand and P. Milanfar, "A general framework for regularized, similarity-based image restoration," *IEEE Trans. on Image Processing*, vol. 23, no. 12, pp. 5136–5151, 2014.

[20] Y. Romano and M. Elad, "Patch-disagreement as away to improve K-SVD denoising," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, April 2015, pp. 1280–1284.

[21] J. Sulam and M. Elad, "Expected patch log likelihood with a sparse prior," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer, 2015, pp. 99–111.

[22] M. Moeller, J. Diebold, G. Gilboa, and D. Cremers, "Learning nonlinear spectral filters for color image reconstruction," in *IEEE Int. Conf. on Computer Vision*, 2015, pp. 289–297.

[23] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–24, 2004.

[24] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, Feb. 2011.