

## Boosting of Image Denoising Algorithms\*

Yaniv Romano<sup>†</sup> and Michael Elad<sup>‡</sup>

**Abstract.** In this paper we propose a generic recursive algorithm for improving image denoising methods. Given the initial denoised image, we suggest repeating the following “SOS” procedure: (i) *Strengthen* the signal by adding the previous denoised image to the degraded input image, (ii) *Operate* the denoising method on the strengthened image, and (iii) *Subtract* the previous denoised image from the restored signal-strengthened outcome. The convergence of this process is studied for the K-SVD image denoising and related algorithms. Still in the context of K-SVD image denoising, we introduce an interesting interpretation of the SOS algorithm as a technique for closing the gap between the local patch-modeling and the global restoration task, thereby leading to improved performance. In a quest for the theoretical origin of the SOS algorithm, we provide a graph-based interpretation of our method, where the SOS recursive update effectively minimizes a penalty function that aims to denoise the image, while being regularized by the graph Laplacian. We demonstrate the SOS boosting algorithm for several leading denoising methods (K-SVD, NLM, BM3D, and EPLL), showing its tendency to further improve denoising performance.

**Key words.** image restoration, denoising, boosting, sparse representation, K-SVD, graph Laplacian, graph theory, regularization

**AMS subject classifications.** 68U10, 94A08, 62H35, 05C50, 47A52, 68R10

**DOI.** 10.1137/140990978

**1. Introduction.** Image denoising is a fundamental restoration problem. Consider a given measurement image  $\mathbf{y} \in \mathbb{R}^{r \times c}$ , obtained from the clean signal  $\mathbf{x} \in \mathbb{R}^{r \times c}$  by a contamination of the form

$$(1.1) \quad \mathbf{y} = \mathbf{x} + \mathbf{v},$$

where  $\mathbf{v} \in \mathbb{R}^{r \times c}$  is a zero-mean additive noise that is independent with respect to  $\mathbf{x}$ . Note that  $\mathbf{x}$  and  $\mathbf{y}$  are held in the above equation as column vectors after lexicographic ordering. A solution to this inverse problem is an approximation  $\hat{\mathbf{x}}$  of the unknown clean image  $\mathbf{x}$ .

Plenty of sophisticated algorithms have been developed in order to estimate the original image content; these include the NLM [6], K-SVD [15], BM3D [13], EPLL [53], and others [51, 9, 35, 26, 46, 36]. These algorithms rely on powerful image models/priors, where sparse representations [5, 14] and processing of local patches [27] have become two prominent ingredients.

\*Received by the editors October 10, 2014; accepted for publication (in revised form) March 13, 2015; published electronically May 26, 2015. This research was supported by the European Research Council under EU's 7th Framework Program, ERC grant agreement 320649, by the Intel Collaborative Research Institute for Computational Intelligence, and by the Google Faculty Research Award.

<http://www.siam.org/journals/siims/8-2/99097.html>

<sup>†</sup>Department of Electrical Engineering, Technion – Israel Institute of Technology, Technion City, Haifa 32000, Israel ([yromano@tx.technion.ac.il](mailto:yromano@tx.technion.ac.il)).

<sup>‡</sup>Department of Computer Science, Technion – Israel Institute of Technology, Technion City, Haifa 32000, Israel ([elad@cs.technion.ac.il](mailto:elad@cs.technion.ac.il)).

Despite the effectiveness of the above denoising algorithms, improved results can be obtained by applying a boosting technique (see [46, 10, 31] for more details). There are several such techniques that were proposed over the years, e.g., “twicing” [49], Bregman iterations [32],  $l_2$ -boosting [8], SAIF [46], and others (e.g., [36]). These algorithms are closely related and share in common the use of the residuals (also known as the “method-noise” [6]) in order to improve the estimates. The residual is defined as the difference between the noisy image and its denoised version. Naturally, the residual contains signal leftovers due to imperfect denoising (together with noise).

For example, motivated by this observation, the idea behind the twicing technique [49] is to extract these leftovers by denoising the residual and then add them back to the estimated image. This can be expressed as [10]

$$(1.2) \quad \hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^k + f(\mathbf{y} - \hat{\mathbf{x}}^k),$$

where the operator  $f(\cdot)$  represents the denoising algorithm and  $\hat{\mathbf{x}}^k$  is the  $k$ th iteration denoised image. The initialization is done by setting  $\hat{\mathbf{x}}^0 = \mathbf{0}$ .

Using the concept of Bregman distance [4] in the context of total-variation denoising [39], Osher et al. [32] suggest exploiting the residual by

$$(1.3) \quad \hat{\mathbf{x}}^{k+1} = f\left(\mathbf{y} + \sum_{i=1}^k (\mathbf{y} - \hat{\mathbf{x}}^i)\right),$$

where the recursive function is initialized by setting  $\hat{\mathbf{x}}^0 = \mathbf{0}$ . Note that if the denoising algorithm  $f(\cdot)$  can be represented as a linear (data-independent) matrix, (1.2) and (1.3) coincide [10]. Furthermore, for these two boosting techniques, it has been shown [46] that as  $k$  increases, the estimate  $\hat{\mathbf{x}}^k$  returns to the noisy image  $\mathbf{y}$ .

Motivated by the above-mentioned algorithms, our earlier work [36] improves the K-SVD [15], NLM [6], and the first-stage of the BM3D [13] by applying an iterative boosting algorithm that extracts the “stolen” image content from the method-noise image. The improvement is achieved by adding the extracted content back to the initial denoised result. The work in [36] suggests representing the signal leftovers of the method-noise patches using the same basis/support that was chosen for the representation of the corresponding clean patch in the initial denoising stage. As an example, in the context of the K-SVD, the supports are sets of atoms that participate in the representation of the noisy patches.

However, in addition to signal leftovers that reside in the residual image, there are noise leftovers that are found in the denoised image. Driven by this observation, SAIF [46] offers a general framework for improving spatial domain denoising algorithms. Their algorithm controls the denoising strength locally by iteratively filtering the image patches. For each patch, it automatically chooses the improvement mechanism, twicing or diffusion, and the number of iterations to apply. The diffusion [31] is a boosting technique that suggests repeating applications of the same denoising filter, thereby removing the noise leftovers that rely on the previous estimate (sometimes also sacrificing some of the high frequencies of the signal).

In this paper we propose a generic recursive function that treats the denoising method as a “black-box” and has the ability to push it forward to improve its performance. Unlike

the above methods, instead of adding the residual (which mostly contains noise) back to the noisy image, or filtering the previous estimate over and over again (which could lead to oversmoothing), we suggest *strengthening the signal* by leveraging on the availability of the denoised image. More specifically, given an initial estimation of the cleaned image, improved results can be achieved by iteratively repeating the following SOS procedure:

1. *Strengthen* the signal by adding the previous denoised image to the noisy input image.
2. *Operate* the denoising method on the strengthened image.
3. *Subtract* the previous denoised image from the restored signal-strengthened outcome.

The core equation that describes this procedure can be written in the following form:

$$(1.4) \quad \hat{\mathbf{x}}^{k+1} = f(\mathbf{y} + \hat{\mathbf{x}}^k) - \hat{\mathbf{x}}^k,$$

where  $\hat{\mathbf{x}}^0 = \mathbf{0}$ . As we show hereafter, a performance improvement is achieved since the signal-strengthened image can be denoised more effectively compared to the noisy input image, due to the improved signal-to-noise ratio (SNR).

The convergence of the proposed algorithm is studied in this paper by formulating the linear part of the denoising method and assessing the iterative system's matrix properties. In this work we put special emphasis on the K-SVD and describe the resulting denoising matrix and the corresponding convergence properties related to it. The work by Milanfar [31] shows that most existing denoising algorithms (e.g., NLM [6], bilateral filter [47], LARK [11]) can be represented as row-stochastic positive definite matrices. In this context, our analysis suggests that for most denoising algorithms, the proposed SOS boosting method is guaranteed to converge. Therefore, we get a straightforward stopping criterion.

In addition, we introduce an interesting interpretation of the SOS boosting algorithm, related to a major shortcoming of patch-based methods: the gap between the local patch-processing and the global need for a whole restored image. In general, patch-based methods (i) break the image into overlapping patches, (ii) restore each patch (local processing), and (iii) reconstruct the image by aggregating the overlapping patches (the global need). The aggregation is usually done by averaging the overlapping patches. The proposed SOS boosting is related to a different algorithm that aims to narrow the local-global gap mentioned above [37]. For each patch, this algorithm defines the difference between the local (intermediate) result and the patch from the global outcome as a "disagreement." Since each patch is processed independently, such disagreement naturally exists.

Interestingly, in the context of the K-SVD image denoising, the SOS algorithm is equivalent to repeating the following steps (see [37] and section 6.2 for more details): (i) compute the disagreement per patch, (ii) subtract the disagreement from the degraded input patches, (iii) apply the restoration algorithm on these patches, and (iv) reconstruct the image. Therefore, the proposed algorithm encourages the overlapping patches to share their local information, thus reducing the gap between the local patch-processing and the global restoration task.

The above should remind the reader of the EPLL framework [53], which also addresses the local-global gap. EPLL encourages the patches of the *final image* (i.e., after patch-averaging) to comply with the local prior. In EPLL, given a local patch model, the algorithm alternates between denoising the previous result according to the local prior and an image reconstruction step (patch-averaging). Several local priors can use this paradigm; the Gaussian mixture model

(GMM) is suggested in the original paper [53]. Similarly, EPLL with sparse and redundant representation modeling was recently proposed in [43]. EPLL bears some resemblance to diffusion methods [31], as it amounts to iterated denoising with a diminishing variance setup, in order to avoid an oversmoothed outcome. In practice, at each diffusion step, the authors of [53, 43] empirically estimate the noise that resides in  $\hat{\mathbf{x}}^k$  (which is neither Gaussian nor independent of  $\hat{\mathbf{x}}^k$ ). In contrast, in our scheme, setting this parameter is trivial; the noise level of  $\mathbf{y} + \hat{\mathbf{x}}^k$  is nearly  $\sigma$  regardless of the iteration number.

In the context of image denoising, several works (e.g., [16, 2, 23, 24, 21]) suggest representing an image as a weighted graph, where the weights measure the similarity between the pixels/patches. Since the graph Laplacian describes the structure of the underlying signal, it can be used as an adaptive regularizer, as is done in the above-mentioned methods. Put differently, the graph Laplacian preserves the geometry of the image by promoting similar pixels to remain similar, thus achieving an effective denoising performance. It turns out that the steady-state outcome of the SOS minimizes a cost function that involves the graph Laplacian as a regularizer, providing another justification for the success of our method. Furthermore, influenced by the SOS mechanism, we offer novel iterative algorithms that minimize the graph Laplacian cost functions that are defined in [16, 2, 23]. Similarly to the SOS, the proposed iterative algorithms treat the denoiser as a “black-box” and operate on the strengthened image, without an explicit construction of the weighted graph.

This paper is organized as follows: In section 2 we provide brief background material on sparse representation and dictionary learning, with special attention to the K-SVD denoising and its matrix form. In section 3 we introduce our novel SOS boosting algorithm, study its convergence, and generalize it by introducing two parameters that govern the steady-state outcome, the requirements for convergence and the rate of convergence. In section 4 we discuss the relation between the SOS boosting and the local-global gap. In section 5 we provide a graph-based analysis to the steady-state outcome of the SOS and offer novel recursive algorithms for related graph Laplacian methods. Experiments are discussed in section 6, showing a meaningful improvement of the K-SVD image denoising and similar boosting for other methods—the NLM, BM3D, and EPLL. Conclusions and future research directions are drawn in section 7.

**2. K-SVD image denoising revisited.** We provide the following discussion on sparse representations and specifically the K-SVD image denoising algorithm, because its matrix interpretation will serve hereafter as a benchmark in the convergence analysis.

**2.1. Sparse representation and K-SVD denoising.** The sparse-land modeling [5, 14] assumes that a given signal  $x \in \mathbb{R}^n$  (in this context, the signal  $x$  is not necessarily an image) can be well represented as  $x = \mathbf{D}\alpha$ , where  $\mathbf{D} \in \mathbb{R}^{n \times m}$  is a dictionary composed of  $m \geq n$  atoms as its columns, and  $\alpha \in \mathbb{R}^m$  is a sparse vector, i.e., has a few nonzero coefficients. For a noisy signal  $y = x + v$ , we seek a representation  $\hat{\alpha}$  that approximates  $x$  up to an error bound, which is proportional to the amount of noise in  $v$ . This is an NP-hard problem that can be expressed as

$$(2.1) \quad \hat{\alpha} = \min_{\alpha} \|\alpha\|_0 \quad \text{s.t.} \quad \|\mathbf{D}\alpha - y\|_2^2 \leq \epsilon^2,$$

where  $\|\alpha\|_0$  counts the nonzero coefficients in  $\alpha$ , and the constant  $\epsilon$  is an error bound. There are many efficient sparse-coding algorithms that approximate the solution of (2.1), such as OMP [33], BP [12], and others [14, 48].

The above discussion assumes that  $\mathbf{D}$  is known and fixed. A line of work (e.g., [17, 42, 1]) shows that adapting the dictionary to the input signal results in a sparser representation. In the case of denoising, under an error constraint, since the dictionary is adapted to the image content, the subspace that the noisy signal is projected onto is of smaller dimension compared to the case of a fixed dictionary. This leads to a stronger noise reduction, i.e., better restoration. Given a set of measurements  $\{\mathbf{y}_i\}_{i=1}^N$ , a typical dictionary learning process [1, 17] is formulated as

$$(2.2) \quad \left[ \hat{\mathbf{D}}, \{\hat{\alpha}_i\}_{i=1}^N \right] = \min_{\mathbf{D}, \{\alpha_i\}_{i=1}^N} \sum_{i=1}^N \gamma_i \|\alpha_i\|_0 + \|\mathbf{D}\alpha_i - \mathbf{y}_i\|_2^2,$$

where  $\hat{\mathbf{D}}$  and  $\{\hat{\alpha}_i\}_{i=1}^N$  are the resulting dictionary and representations, respectively. The scalars  $\gamma_i$  are signal-dependent so as to comply with a set of constraints of the form  $\|\mathbf{D}\alpha_i - \mathbf{y}_i\|_2^2 \leq \epsilon^2$ .

Due to computational demands, adapting a dictionary to large signals (images in our case) is impractical. Therefore, a treatment of an image is done by breaking it into overlapping patches (e.g., of size  $8 \times 8$ ). Then, each patch is restored according to the sparsity-inspired prior. More specifically, the K-SVD image denoising algorithm [15] divides the noisy image into  $\sqrt{n} \times \sqrt{n}$  fully overlapping patches and then processes them *locally* by performing iterations of sparse-coding (using OMP) and dictionary learning as described in (2.2). Finally, the *global* denoised image is obtained by returning the cleaned patches to their original locations, followed by an averaging with the input noisy image. The above procedure approximates the solution of

$$(2.3) \quad \left[ \hat{\mathbf{x}}, \hat{\mathbf{D}}, \{\hat{\alpha}_i\}_{i=1}^N \right] = \min_{\mathbf{x}, \mathbf{D}, \{\alpha_i\}_{i=1}^N} \mu \|\mathbf{x} - \mathbf{y}\|_2^2 + \sum_{i=1}^N \gamma_i \|\alpha_i\|_0 + \|\mathbf{D}\alpha_i - \mathbf{R}_i \mathbf{x}\|_2^2,$$

where  $\hat{\mathbf{x}} \in \mathbb{R}^{rc}$  is the resulting denoised image,  $N$  is the number of patches, and  $\mathbf{R}_i \in \mathbb{R}^{n \times rc}$  is a matrix that extracts the  $i$ th patch from the image. The first term in (2.3) demands a proximity between the noisy and denoised images. The second term demands that each patch  $\mathbf{R}_i \mathbf{x}$  be represented sparsely up to an error bound, with respect to a dictionary  $\mathbf{D}$ . As to the coefficients  $\gamma_i$ , those are spatially dependent and are set as explained in (2.2).

**2.2. K-SVD image denoising: A matrix formulation.** The K-SVD image denoising can be divided into nonlinear and linear parts. The former is composed of preparation steps that include the support determination within the sparse-coding and the dictionary update, while the outcome of the latter is the actual image-adaptive filter that cleans the noisy image. The matrix formulation of the K-SVD denoising represents its linear part, assuming the availability of the nonlinear computations. At this stage we should note that *the following formulation is given as background to the theoretical analysis that follows, and it is not necessary when using the proposed SOS boosting in practice.*

Sparse-coding determines for each noisy patch  $\mathbf{R}_i \mathbf{y}$  a small set of atoms  $\mathbf{D}_{s_i}$  which participate in its representation. Following the last step of the OMP [33], given  $\mathbf{D}_{s_i}$ , the repre-

sentation<sup>1</sup>  $\alpha_i$  of the clean patch is obtained by solving

$$(2.4) \quad \alpha_i = \min_z \|\mathbf{D}_{s_i} z - \mathbf{R}_i \mathbf{y}\|_2^2,$$

which has a closed-form solution

$$(2.5) \quad \alpha_i = (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \mathbf{R}_i \mathbf{y}.$$

Given  $\alpha_i$ , the clean patch  $\hat{\mathbf{p}}_i$  is obtained by applying the inverse transform from the representation to the signal/patch space, i.e.,

$$(2.6) \quad \begin{aligned} \hat{\mathbf{p}}_i &= \mathbf{D}_{s_i} \alpha_i \\ &= \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \mathbf{R}_i \mathbf{y}. \end{aligned}$$

Notice that although the computation of  $s_i$  is nonlinear, the clean patch  $\hat{\mathbf{p}}_i$  is obtained by filtering its noisy version,  $\mathbf{R}_i \mathbf{y}$ , with a linear, image-adaptive, symmetric, and normalized filter.

Following (2.3) and given all  $\hat{\mathbf{p}}_i = \mathbf{D}_{s_i} \alpha_i$ , the globally denoised image  $\hat{\mathbf{x}}$  is obtained by minimizing

$$(2.7) \quad \hat{\mathbf{x}} = \min_{\mathbf{x}} \mu \|\mathbf{x} - \mathbf{y}\|_2^2 + \sum_{i=1}^N \|\hat{\mathbf{p}}_i - \mathbf{R}_i \mathbf{x}\|_2^2.$$

This is a quadratic expression that has a closed-form solution of the form

$$(2.8) \quad \begin{aligned} \hat{\mathbf{x}} &= \left( \mu \mathbf{I} + \sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i \right)^{-1} \left( \mu \mathbf{y} + \sum_{i=1}^N \mathbf{R}_i^T \hat{\mathbf{p}}_i \right) \\ &= \left( \mu \mathbf{I} + \sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i \right)^{-1} \left( \mu \mathbf{I} + \sum_{i=1}^N \mathbf{R}_i^T \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \mathbf{R}_i \right) \mathbf{y} \\ &= \mathbf{D}^{-1} \mathbf{K} \mathbf{y} \\ &= \mathbf{W} \mathbf{y}, \end{aligned}$$

where  $\mathbf{I} \in \mathbb{R}^{rc \times rc}$  is the identity matrix. The term  $\sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i$  is a diagonal matrix that counts the appearances of each pixel (e.g., 64 for patches of size  $8 \times 8$ ), and  $\mu \mathbf{I}$  originates from the averaging with the noisy image  $\mathbf{y}$ . The matrix  $\mathbf{R}_i^T$  returns a clean patch  $\hat{\mathbf{p}}_i$  to its original location in the global image. The matrix  $\mathbf{W} \in \mathbb{R}^{rc \times rc}$  is the resulting filter matrix formulation of the linear part of the K-SVD image denoising. In the context of graph theory,  $\mathbf{D}$  and  $\mathbf{K}$  are called the degree and similarity matrices, respectively (see section 5 for more information).

A series of works [46, 31, 45] studies the algebraic properties of such formulations for several image denoising algorithms (NLM [6], bilateral filter [47], kernel regression [11]), for

---

<sup>1</sup>We abuse notation here as  $\alpha_i$  refers hereafter only to the nonzero part of the representation, which is a vector of length  $|s_i| \ll m$ .

which the filter-matrix is a nonsymmetric and row-stochastic matrix. Thus, this matrix has real and positive eigenvalues in the range of  $[0, 1]$ , and the largest eigenvalue is unique and equals 1, with a corresponding eigenvector  $[1, 1, \dots, 1]^T$  [40, 22]. In the K-SVD case, and under the assumption of the periodic boundary condition,<sup>2</sup> the properties of the resulting matrix are somewhat different and are given in the following theorem.

**Theorem 2.1.** *The resulting matrix  $\mathbf{W}$  has the following properties:*

1. *Symmetric:  $\mathbf{W} = \mathbf{W}^T$ , and thus all eigenvalues are real.*
2. *Positive definite:  $\mathbf{W} \succ 0$ , and thus all eigenvalues are strictly positive.*
3. *The minimal eigenvalue of  $\mathbf{W}$  satisfies  $\lambda_{\min}(\mathbf{W}) \geq \frac{\mu}{\mu+n}$ , where  $n$  is the patch size.*
4. *Doubly stochastic in the sense of  $\mathbf{W}\mathbf{1} = \mathbf{W}^T\mathbf{1} = \mathbf{1}$ . Note that  $\mathbf{W}$  may have negative entries, which violates the classic definition of row- or column-stochasticity.*
5. *The above implies that 1 is an eigenvalue corresponding to the eigenvector  $\mathbf{1}$ .*
6. *The spectral radius of  $\mathbf{W}$  equals 1; i.e.,  $\|\mathbf{W}\|_2 = 1$ .*
7. *The above implies that the maximal eigenvalue satisfies  $\lambda_{\max}(\mathbf{W}) = 1$ .*
8. *The spectral radius  $\|\mathbf{W} - \mathbf{I}\|_2 \leq \frac{n}{\mu+n} < 1$ .*

Appendix B provides a proof of these claims.

For the denoising algorithms studied in [46, 31, 45], the matrix  $\mathbf{W}$  is neither symmetric nor positive definite; however, it can be approximated as such using the Sinkhorn procedure [31]. In the context of the K-SVD, as described in Appendix A,  $\mathbf{W}$  can become symmetric by a proper treatment of the boundaries (essentially performing cyclic processing of the patches).

To conclude, the discussion above shows that the K-SVD is a member of a large family of denoising algorithms that can be represented as matrices [31]. We will use this formulation in order to study the convergence of the proposed SOS boosting and for demonstrating the local-global interpretation.

**3. SOS boosting.** In this section we describe the proposed algorithm, study its convergence, and generalize it by introducing two parameters that govern its steady-state outcome, the requirements for convergence and its rate.

**3.1. SOS boosting: The core idea.** Leading image/patch priors are able to effectively distinguish the signal content from the noise. However, an emphasis of the signal over the noise could help the prior better identify the image content, thereby leading to better denoising performance. As an example, the sparsity-based K-SVD could choose atoms that better fit the underlying signal. Similarly, the NLM, which cleans a noisy patch by applying a weighted average with its spatial neighbors, could determine better weights. This is the key idea behind the proposed SOS boosting algorithm, which exploits the previous estimation in order to enhance the underlying signal. In addition, the proposed algorithm treats the denoiser as a black-box; thus it is easy to use and becomes applicable to a wide range of denoising methods.

As mentioned in section 1, the first class of boosting algorithms (twicing [49] or its variants [32, 8, 36]) suggests extracting the “stolen” content from the method-noise image, with the risk of returning noise back to the denoised image, together with the extracted information. On the other hand, the second class of boosting methods (diffusion [31] or EPLL [53, 43]) aims at removing the noise that resides in the estimated image, with the risk of obtaining

---

<sup>2</sup>See Appendix A for an explanation of this requirement.

an oversmoothed result (this depends on the number of iterations or the denoiser parameters at each iteration). As a consequence, these two classes of boosting algorithms are somewhat lacking as they address only one kind of leftovers [46]—that which resides in the method-noise, or the other, which is found in the denoised image. Also, these methods may result in an undersmoothed or oversmoothed version of the noisy image.

Adopting a different perspective, we suggest *strengthening* the signal by adding the clean image  $\hat{\mathbf{x}}^k$  to the noisy input  $\mathbf{y}$  and then *operating* the denoising algorithm on the strengthened result. Unlike diffusion filtering, as the estimated part of the signal is emphasized, there is no loss of signal content that has not been estimated correctly (due to the availability of  $\mathbf{y}$ ). Unlike twicing, we hardly increase the noise level (under the assumption that the energy of the noise which resides in the clean image is small). Finally, a *subtraction* of  $\hat{\mathbf{x}}^k$  from the outcome should be done in order to obtain a faithful denoised result. This procedure is formulated in (1.4):

$$\hat{\mathbf{x}}^{k+1} = f(\mathbf{y} + \hat{\mathbf{x}}^k) - \hat{\mathbf{x}}^k,$$

where  $\hat{\mathbf{x}}^0 = \mathbf{0}$ .

The SOS boosting obtains improved denoising performance due to higher SNR of the signal-strengthened image, compared to the noisy input. In order to demonstrate this, let us denote

$$(3.1) \quad \hat{\mathbf{x}} = \mathbf{x} + \mathbf{v}_r,$$

where  $\mathbf{v}_r$  is the error that resides in the outcome  $\hat{\mathbf{x}}$ , containing both noise residuals and signal errors. Assuming that the denoising algorithm is effective, and  $\hat{\mathbf{x}}$  has an improved SNR compared to  $\mathbf{y}$ , this means that

$$(3.2) \quad \frac{\|\mathbf{x}\|}{\|\mathbf{v}_r\|} \gg \frac{\|\mathbf{x}\|}{\|\mathbf{v}\|},$$

implying

$$(3.3) \quad \|\mathbf{v}_r\| = \delta \|\mathbf{v}\|, \quad \text{where } \delta \ll 1.$$

Thus, referring now to the addition  $\mathbf{y} + \hat{\mathbf{x}}$ , its SNR satisfies

$$(3.4) \quad \begin{aligned} \text{SNR}^2(\mathbf{y} + \hat{\mathbf{x}}) &= \frac{\|2\mathbf{x}\|^2}{\|\mathbf{v} + \mathbf{v}_r\|^2} \\ &\geq \frac{4\|\mathbf{x}\|^2}{\|\mathbf{v}\|^2 + 2\|\mathbf{v}\|\|\mathbf{v}_r\| + \|\mathbf{v}_r\|^2}. \end{aligned}$$

In the above we used the Cauchy–Schwarz inequality. Using (3.3), we get

$$(3.5) \quad \begin{aligned} \text{SNR}^2(\mathbf{y} + \hat{\mathbf{x}}) &\geq \frac{4\|\mathbf{x}\|^2}{(1 + \delta)^2 \|\mathbf{v}\|^2} \\ &= \frac{4}{(1 + \delta)^2} \text{SNR}^2(\mathbf{y}). \end{aligned}$$



Since  $\delta \ll 1$ , we have that

$$(3.6) \quad \text{SNR}(\mathbf{y} + \hat{\mathbf{x}}) > \text{SNR}(\mathbf{y}),$$

where in the ideal case ( $\delta = 0$ ) the relation becomes

$$(3.7) \quad \text{SNR}(\mathbf{y} + \hat{\mathbf{x}}) = 2 \cdot \text{SNR}(\mathbf{y}).$$

**3.2. Convergence analysis.** Studying the convergence of the SOS boosting is done by leveraging the linear matrix formulation of the denoising algorithm. The error of the SOS recursive function

$$(3.8) \quad e_k = \hat{\mathbf{x}}^k - \hat{\mathbf{x}}^*$$

is defined as the difference between the  $k$ th estimate,

$$(3.9) \quad \hat{\mathbf{x}}^k = \mathbf{W}_k (\mathbf{y} + \hat{\mathbf{x}}^{k-1}) - \hat{\mathbf{x}}^{k-1},$$

and the outcome that is obtained after a large number iterations,

$$(3.10) \quad \hat{\mathbf{x}}^* = \mathbf{W}_* (\mathbf{y} + \hat{\mathbf{x}}^*) - \hat{\mathbf{x}}^*,$$

where  $\mathbf{W}_k$  is a filter matrix, which is equivalent to applying  $f(\cdot)$  on the signal-strengthened image. Substituting (3.9) and (3.10) into (3.8) leads to

$$(3.11) \quad \begin{aligned} e_k &= \mathbf{W}_k (\mathbf{y} + \hat{\mathbf{x}}^{k-1}) - \hat{\mathbf{x}}^{k-1} - (\mathbf{W}_* (\mathbf{y} + \hat{\mathbf{x}}^*) - \hat{\mathbf{x}}^*) \\ &= (\mathbf{W}_k - \mathbf{W}_*) \mathbf{y} + \mathbf{W}_k \hat{\mathbf{x}}^{k-1} - \mathbf{W}_k \hat{\mathbf{x}}^* + \mathbf{W}_k \hat{\mathbf{x}}^* - \mathbf{W}_* \hat{\mathbf{x}}^* - (\hat{\mathbf{x}}^{k-1} - \hat{\mathbf{x}}^*) \\ &= (\mathbf{W}_k - \mathbf{W}_*) \mathbf{y} + \mathbf{W}_k e_{k-1} + (\mathbf{W}_k - \mathbf{W}_*) \hat{\mathbf{x}}^* - e_{k-1} \\ &= (\mathbf{W}_k - \mathbf{I}) e_{k-1} + (\mathbf{W}_k - \mathbf{W}_*) (\mathbf{y} + \hat{\mathbf{x}}^*), \end{aligned}$$

where we use the recursive connection  $e_{k-1} = \hat{\mathbf{x}}^{k-1} - \hat{\mathbf{x}}^*$ . We should note that the nonlinearity of  $f(\mathbf{y} + \hat{\mathbf{x}}^{k-1}) = \mathbf{W}_k (\mathbf{y} + \hat{\mathbf{x}}^{k-1})$  is neglected in the above derivation by allowing an operation of the form  $\mathbf{W}_k (\mathbf{y} + \hat{\mathbf{x}}^{k-1}) = \mathbf{W}_k \mathbf{y} + \mathbf{W}_k \hat{\mathbf{x}}^{k-1}$ .

In the following convergence analysis, we shall assume a fixed filter-matrix  $\mathbf{W}$  that operates on the signal-strengthened image throughout all of the SOS steps, i.e.,  $\mathbf{W} = \mathbf{W}_k = \mathbf{W}_*$ . This comes up in practice after applying the SOS boosting for a large number of iterations (as explained in the context of Figure 1). In this case, the above-mentioned abuse of the nonlinearity becomes correct, and thus the convergence analysis is valid.

**Theorem 3.1.** *Assume that  $\mathbf{W} = \mathbf{W}_k = \mathbf{W}_*$  and that the spectral radius of the transition matrix  $\|\mathbf{W} - \mathbf{I}\|_2 = \gamma < 1$ . The error  $e_k$  converges exponentially, i.e.,  $\|e_k\|_2 \leq \|e_0\|_2 \cdot \gamma^k \rightarrow 0$  for  $k \rightarrow \infty$ . Thus, the SOS recursive function is guaranteed to converge.*

*Proof.* By assigning  $\mathbf{W}_k = \mathbf{W}_*$ , the second term  $(\mathbf{W}_k - \mathbf{W}_*) (\mathbf{y} + \hat{\mathbf{x}}^*)$  in (3.11) vanishes; thus

$$(3.12) \quad \begin{aligned} e_k &= (\mathbf{W} - \mathbf{I}) e_{k-1} \\ &= (\mathbf{W} - \mathbf{I})^k e_0, \end{aligned}$$

where  $e_0 = \hat{\mathbf{x}}^0 - \hat{\mathbf{x}}^* = -\hat{\mathbf{x}}^*$  is a constant vector. Using matrix-norm inequalities, we get

$$(3.13) \quad \begin{aligned} \|e_k\|_2 &\leq \|\mathbf{W} - \mathbf{I}\|_2^k \cdot \|e_0\|_2 \\ &= \gamma^k \cdot \|e_0\|_2, \end{aligned}$$

where we use  $\|\mathbf{W} - \mathbf{I}\|_2 = \gamma$ . As a result,  $\|e_k\|_2$  is bounded by  $\gamma^k \cdot \|e_0\|_2$  and approaches zero for  $k \rightarrow \infty$  when  $\gamma < 1$ . ■

As such, the SOS boosting is guaranteed to converge for a wide range of denoising algorithms—those that can be formulated/approximated such that  $\mathbf{W} - \mathbf{I}$  is convergent, e.g., the K-SVD [15], NLM [6], bilateral filter [47], and LARK [11]. In the next subsection we soften the convergence requirements and intensify its properties, along with a practical demonstration.

**3.3. Parametrization.** We generalize the SOS boosting algorithm by introducing two parameters that modify the steady-state outcome, the requirements for convergence (the eigenvalues range) and its rate. Starting with the first parameter,  $\rho$ , which controls the signal emphasis, the proposed formulation is

$$(3.14) \quad \hat{\mathbf{x}}^{k+1} = f(\mathbf{y} + \rho\hat{\mathbf{x}}^k) - \rho\hat{\mathbf{x}}^k,$$

where a large value of  $\rho$  implies a strong emphasis of the underlying signal. Assigning  $\hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^k = \hat{\mathbf{x}}^*$  and replacing  $f(\cdot)$  with a fixed filter-matrix  $\mathbf{W}_*$  lead to

$$(3.15) \quad \hat{\mathbf{x}}^* = \mathbf{W}_*(\mathbf{y} + \rho\hat{\mathbf{x}}^*) - \rho\hat{\mathbf{x}}^*,$$

which implies a steady-state result

$$(3.16) \quad \hat{\mathbf{x}}^* = (\mathbf{I} + \rho(\mathbf{I} - \mathbf{W}_*))^{-1} \mathbf{W}_*\mathbf{y}.$$

This is the new steady-state outcome, obtained only if the SOS boosting converges. We should note that this outcome also minimizes a cost function that involves the graph Laplacian as a regularizer (see section 5 for further details). The conditions for convergence are studied hereafter.

The second parameter,  $\tau$ , modifies the eigenvalues of the error's transition matrix, thereby leading to a faster convergence and relaxing the requirement that only  $f(\cdot)$  with eigenvalues between 0 to 1 is guaranteed to converge. We introduce this parameter in such a way that it will not affect that steady-state outcome (at least as far as the linear approximation is concerned). We start with the steady-state relation

$$(3.17) \quad \hat{\mathbf{x}}^* = f(\mathbf{y} + \rho\hat{\mathbf{x}}^*) - \rho\hat{\mathbf{x}}^*.$$

We multiply both sides by  $\tau$  and add the term  $\hat{\mathbf{x}}^* - \hat{\mathbf{x}}^*$  to the right-hand side (RHS):

$$(3.18) \quad \tau\hat{\mathbf{x}}^* = \tau f(\mathbf{y} + \rho\hat{\mathbf{x}}^*) - \tau\rho\hat{\mathbf{x}}^* + \hat{\mathbf{x}}^* - \hat{\mathbf{x}}^*.$$

Thus, the same  $\hat{\mathbf{x}}^*$  solving (3.17) will also solve (3.18), and thus the steady-state is not affected. Rearranging this equality leads to

$$(3.19) \quad \hat{\mathbf{x}}^* = \tau f(\mathbf{y} + \rho\hat{\mathbf{x}}^*) - (\tau\rho + \tau - 1)\hat{\mathbf{x}}^*.$$

As a result, the proposed generalized SOS boosting is given by

$$(3.20) \quad \hat{\mathbf{x}}^{k+1} = \tau f(\mathbf{y} + \rho \hat{\mathbf{x}}^k) - (\tau\rho + \tau - 1)\hat{\mathbf{x}}^k.$$

It is important to note that although  $\tau$  does not affect  $\hat{\mathbf{x}}^*$  explicitly, it may modify the estimates  $\hat{\mathbf{x}}^k$  over the iterations. Due to the adaptivity of  $f(\cdot)$  to its input, such modifications may eventually affect the steady-state outcome.

We study the convergence of (3.20) the same way we did in section 3.2. We start with the error computation, expressed by

$$(3.21) \quad \begin{aligned} e_k &= \hat{\mathbf{x}}^k - \hat{\mathbf{x}}^* \\ &= \tau \mathbf{W}_k (\mathbf{y} + \rho \hat{\mathbf{x}}^{k-1}) - (\tau\rho + \tau - 1)\hat{\mathbf{x}}^{k-1} - (\tau \mathbf{W}_* (\mathbf{y} + \rho \hat{\mathbf{x}}^*) - (\tau\rho + \tau - 1)\hat{\mathbf{x}}^*) \\ &= \tau (\mathbf{W}_k - \mathbf{W}_*) \mathbf{y} + \tau\rho \mathbf{W}_k \hat{\mathbf{x}}^{k-1} - \tau\rho \mathbf{W}_k \hat{\mathbf{x}}^* + \tau\rho \mathbf{W}_k \hat{\mathbf{x}}^* - \tau\rho \mathbf{W}_* \hat{\mathbf{x}}^* - (\tau\rho + \tau - 1)e_{k-1} \\ &= (\tau\rho \mathbf{W}_k - (\tau\rho + \tau - 1)\mathbf{I}) e_{k-1} + \tau (\mathbf{W}_k - \mathbf{W}_*) (\mathbf{y} + \rho \hat{\mathbf{x}}^*). \end{aligned}$$

Next, following Theorem 3.1, and assuming  $\mathbf{W} = \mathbf{W}_k = \mathbf{W}_*$ , we get that the condition for convergence is

$$(3.22) \quad \forall i \quad \phi(\tau, \rho, \lambda_i) = |\tau\rho\lambda_i - (\tau\rho + \tau) + 1| < 1,$$

where  $\{\lambda_i\}_{i=1}^N$  and  $\{\phi(\tau, \rho, \lambda_i)\}_{i=1}^N$  are the eigenvalues of  $\mathbf{W}$  and the error's transition matrix, respectively. In order to achieve the fastest convergence, we seek the parameter  $\tau^*$  that minimizes

$$(3.23) \quad \tau^* = \min_{\tau} \max_{1 \leq i \leq N} \phi(\tau, \rho, \lambda_i) \quad \text{s.t.} \quad \forall i \quad \phi(\tau, \rho, \lambda_i) < 1.$$

Given  $\tau^*$ , the rate of convergence is governed by

$$(3.24) \quad \gamma^* = \max_{1 \leq i \leq N} \phi(\tau^*, \rho, \lambda_i).$$

Appendix D provides the following closed-form solution to (3.23):

$$(3.25) \quad \tau^* = \frac{2}{2(\rho + 1) - \rho(\lambda_{min} + \lambda_{max})},$$

along with optimal convergence rate

$$(3.26) \quad \gamma^* = \frac{\rho(\lambda_{max} - \lambda_{min})}{2(\rho + 1) - \rho(\lambda_{min} + \lambda_{max})}.$$

In the context of the K-SVD image denoising [15], Figure 1 demonstrates the properties of the generalized SOS recursive function for the image House, corrupted by zero-mean Gaussian noise with  $\sigma = 25$ . Each K-SVD operation includes five iterations of sparse-coding and dictionary-update with noise level of  $\hat{\sigma} = 1.05\sigma$ . We repeat these operations for 100 SOS

steps and set  $\hat{\mathbf{x}}^* = \hat{\mathbf{x}}^{100}$ . In the following experiment, the denoised images are the outcome of  $\hat{\mathbf{x}}^{k+1} = \mathbf{W}(\mathbf{y} + \hat{\mathbf{x}}^k) - \hat{\mathbf{x}}^k$ , where  $\mathbf{W}$  is held fixed as  $\mathbf{W} = \mathbf{W}_{100}$ , initializing with  $\hat{\mathbf{x}}_0 = \mathbf{0}$ .

According to Theorem 2.1 and based on the original K-SVD parameters ( $n = 8$ ,  $m = 256$ ,  $\mu = 1.02$ ), we get  $\lambda_{\min} \geq 0.015$  and  $\lambda_{\max} = 1$ . These lead to  $\tau^* = 0.67$  (see (3.25)). Figure 1(a) plots the logarithm of  $\|e_k\|_2$  for  $[\tau_1, \tau_2, \tau_3] = [\frac{1}{2}\tau^*, \tau^*, 1]$ . As can be seen, the error norm decreases linearly, and bounded by  $c \cdot \gamma_i^k$ , where  $[\gamma_1, \gamma_2, \gamma_3] = [0.66, 0.33, 0.98]$ . The fastest convergence is obtained for  $\tau_2 = \tau^* = 0.67$  with  $\gamma_2 = \gamma^* = 0.33$ , while the slowest is obtained for  $\tau_3 = 1$  with  $\gamma_3 = 0.98$ .

Figure 1(b) demonstrates the peak SNR (PSNR) improvement (the higher the better) as a function of the SOS step. As can be seen, faster convergence of  $\|e_k\|_2$  translates well into faster improvement of the final image. The SOS boosting achieves a PSNR of 32.78dB, offering an impressive improvement over the original K-SVD algorithm, which obtains 31.8dB.

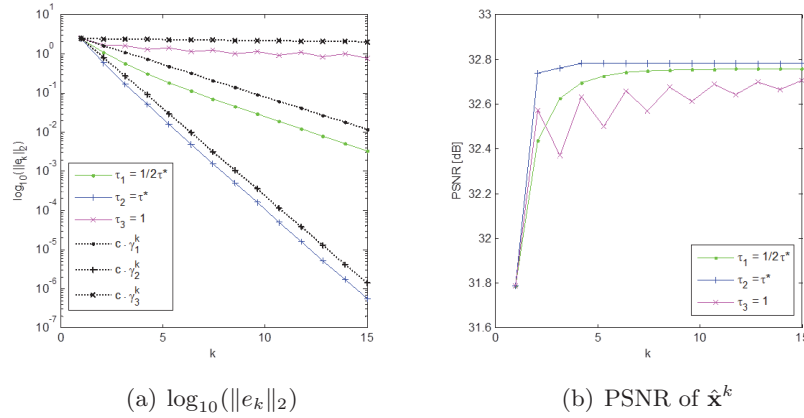
**4. Local-global interpretation.** As described in section 1, there is a stubborn gap between the local processing of image patches and the global need (creating a final image by aggregating the patches). Consider a denoising scenario based on overlapping patches (e.g., [15, 51]): At the local processing stage, each patch is denoised independently<sup>3</sup> without any influence from neighboring patches. Then, a global stage merges these outcomes by plainly averaging the local denoising results.

Inspired by game-theory ideas, in particular the “consensus and sharing” optimization problem [3], we introduce an interesting local-global interpretation of the above-proposed SOS boosting algorithm. A game theoretical terminology of a patch-based processing can be viewed as the following: There are several agents, and each one of them adjusts its local variable to minimize its individual cost (in our case, representing the noisy patch sparsely). In addition, there is a shared objective term (the global image) that describes the overall goal. Imitating this concept, we call the following SOS interpretation “sharing the disagreement.” This approach, reported in [37], reduces the local-global gap by encouraging the overlapping patches to reach an agreement before they merge their forces by the averaging.

The proposed boosting algorithm reduces the local-global gap in the following way. For each patch, we define the difference between the local (intermediate) result and the patch from the global outcome as a “disagreement.” Since each patch is denoised independently, such disagreement is almost always nonzero and even substantial. The information between the overlapping patches is shared by subtracting the disagreement from the noisy image patches, i.e., seeking an agreement between them. These modified patches are the new inputs to the denoising algorithm. In this way we push the overlapping patches to share their local results, influence each other, and reduce the local-global gap.

More specifically, given an initial denoised version of  $\mathbf{y}$  and its intermediate patch results, we suggest repeating the following procedure: (i) compute the disagreement for each patch, (ii) subtract the result from the noisy input patches, (iii) apply the denoising algorithm to these modified patches, and (iv) reconstruct the image by averaging on the overlaps. Focusing on the K-SVD image denoising, this procedure is detailed in Algorithm 1.

<sup>3</sup>Note that in our terminology, even methods like BM3D [13] are considered local, even though they share information between groups of patches. Indeed, our discussion covers this and related methods as well. In a way, the approach taken in [35] offers some sort of remedy to the BM3D method.



**Figure 1.** Illustration of the generalized SOS recursive function properties: (a) Convergence and (b) PSNR improvement. These graphs are generated by operating the K-SVD denoising [15] on noisy ( $\sigma = 25$ ) House image.

---

**Algorithm 1 :** Sharing the disagreement approach [37].

---

**Initialization:**

- 1:  $\mathbf{D}^0 \in \mathbb{R}^{n \times m}$  – initial dictionary.
- 2: Set  $k = 0$ .
- 3: Set  $\mathbf{q}_i^0 = 0$ , where  $\mathbf{q}_i \in \mathbb{R}^n$  is a “disagreement” patch, corresponding to the  $i$ th patch in the image.

**Repeat**

- 1: **Sparse-coding and dictionary update:** Solve

$$(4.1) \quad [\mathbf{D}^{k+1}, \{\hat{\alpha}_i^{k+1}\}_{i=1}^N] = \min_{\mathbf{D}, \{\alpha_i\}_{i=1}^N} \sum_{i=1}^N \gamma_i \|\alpha_i\|_0 + \|\mathbf{D}\alpha_i - (\mathbf{R}_i \mathbf{y} - \mathbf{q}_i^k)\|_2^2.$$

In practice we approximate the representation of  $\mathbf{R}_i \mathbf{y} - \mathbf{q}_i^k$  using the OMP [33] and update the previous dictionary  $\mathbf{D}^k$  using the K-SVD algorithm [1].

- 2: **Image reconstruction:** Solve

$$(4.2) \quad \hat{\mathbf{x}}^{k+1} = \min_z \sum_i \|\mathbf{D}^{k+1} \alpha_i^{k+1} - \mathbf{R}_i z\|_2^2.$$

This term leads to a simple averaging of the denoised patches  $\mathbf{D}^{k+1} \alpha_i^{k+1}$  on the overlaps.

- 3: **Disagreement computation:** Update

$$(4.3) \quad \mathbf{q}_i^{k+1} = \mathbf{D}^{k+1} \alpha_i^{k+1} - \mathbf{R}_i \hat{\mathbf{x}}^{k+1},$$

where  $\mathbf{q}_i^{k+1}$  is the “disagreement” between the independent estimation  $\mathbf{D}^{k+1} \alpha_i^{k+1}$  and the corresponding patch from the global outcome  $\mathbf{R}_i \hat{\mathbf{x}}^{k+1}$ .

**Until**

Maximum denoising quality has been reached, else increment  $k$  and return to “Sparse-coding and dictionary update.”

**Output**

$\hat{\mathbf{x}}^*$  – the last iteration result.

---

The modified input patches contain their neighbors' information, thus encouraging the locally denoised patches to agree on the global result. Substituting  $\mathbf{q}_i^k = \mathbf{D}^k \alpha_i^k - \mathbf{R}_i \hat{\mathbf{x}}^k$  into (4.1) leads to

$$(4.4) \quad \left[ \mathbf{D}^{k+1}, \{\hat{\alpha}_i^{k+1}\}_{i=1}^N \right] = \min_{\mathbf{D}, \{\alpha_i\}_{i=1}^N} \sum_{i=1}^N \gamma_i \|\alpha_i\|_0 + \|\mathbf{D}\alpha_i - (\mathbf{R}_i \mathbf{y} - \mathbf{D}^k \alpha_i^k + \mathbf{R}_i \hat{\mathbf{x}}^k)\|_2^2.$$

Now, by denoting the *local* residual (method-noise) as  $\mathbf{r}_i = \mathbf{R}_i \mathbf{y} - \mathbf{D}^k \alpha_i^k$ , we get

$$(4.5) \quad \left[ \mathbf{D}^{k+1}, \{\hat{\alpha}_i^{k+1}\}_{i=1}^N \right] = \min_{\mathbf{D}, \{\alpha_i\}_{i=1}^N} \sum_{i=1}^N \gamma_i \|\alpha_i\|_0 + \|\mathbf{D}\alpha_i - (\mathbf{R}_i \hat{\mathbf{x}}^k + \mathbf{r}_i)\|_2^2,$$

where the representation  $\mathbf{D}\alpha_i$  is the denoised version of the patch  $\mathbf{R}_i \hat{\mathbf{x}}^k + \mathbf{r}_i$ . In this formulation, the input to the K-SVD is a patch from the global (previous iteration) cleaned image  $\mathbf{R}_i \hat{\mathbf{x}}^k$ , contaminated by its own local method-noise  $\mathbf{r}_i$ . Notice the major differences between (1.2), which denoises the method-noise, (1.3), which adds the method-noise to the noisy image and then denoises the result, and our *local* approach, which aims at recovering the previous global estimation, thereby leading to an agreement between the patches. Our algorithm is also different from the EPLL [53], which denoises the previous cleaned image without considering its method-noise.

Still in the context of the K-SVD, Appendix C shows, under some assumptions, an *equivalence* between the SOS recursive function (equation (1.4)) and the above “sharing the disagreement” algorithm. It is important to emphasize that the former treats the K-SVD as a black-box and is thereby blind to the K-SVD intermediate results (the independent denoised patches, before the patch averaging step). On the contrary, in the case of the disagreement approach, these intermediate results are crucial; they are central in the algorithm. Therefore, the connection between the SOS and the disagreement algorithms is far from trivial.

**5. Graph Laplacian interpretation.** In this section we present a graph-based analysis to the SOS boosting. We start by providing a brief background on graph representation of an image in the context of denoising. Second, we explore the graph Laplacian regularization in general and, in the context of (3.16), the steady-state outcome of the SOS boosting. Finally, we suggest novel recursive algorithms (that treat the denoiser as a black-box) to the graph Laplacian regularizers that are described in [16, 2, 23, 24].

Recent works [19, 20, 16, 2, 41, 29, 21, 23, 24] suggest representing an image as a weighted graph  $\mathcal{G} = (\mathbf{V}, \mathbf{E}, \mathbf{K})$ , where the vertices  $\mathbf{V}$  represent the image pixels, and the edges  $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$  represent the connection/similarity between pairs of pixels, with a corresponding weight  $\mathbf{K}(i, j)$ .

A constructive approach for composing a graph Laplacian for an image is via image denoising algorithms. Given a denoising process for an image, which can be represented as a matrix multiplication,  $\hat{\mathbf{x}} = \mathbf{W}\mathbf{y}$ , one can refer to the entry  $(i, j)$  as revealing information about the proximity between the  $i$ th and  $j$ th pixels. We note that the existence of the matrix  $\mathbf{W}$  does not imply that the denoising process is linear. Rather, the nonlinearity is hidden within the construction of the entries of  $\mathbf{W}$ . For example, in the case of the NLM [6], bilateral [47],

and LARK [11] filters, the entries of  $\mathbf{K}$  can be expressed by

$$(5.1) \quad \mathbf{K}(i, j) = \exp\left(-\frac{d^2(i, j)}{h^2}\right),$$

where  $d(i, j)$  measures the distance between the  $(i, j)$  pixels (or patches), and  $h$  is a smoothing parameter. Notice that in the case of the sparsity-based K-SVD denoising [15], the weights  $\mathbf{K}(i, j)$ , as defined in (2.8), measure the similarity between the  $(i, j)$  pixels through the dictionary  $\mathbf{D}$ . Dealing with an undirected graph  $\mathcal{G}$ , the degree  $d_i$  of the vertex  $\mathbf{V}_i$  can be defined by

$$(5.2) \quad \mathbf{D}(i, i) = d_i = \sum_j \mathbf{K}(i, j),$$

where  $d_i$  is a sum over the weights on the edges that are connected to  $\mathbf{V}_i$ , and  $\mathbf{D}$  is a diagonal matrix (called the degree matrix) containing the values of  $\{d_i\}_{i=1}^N$  in its diagonal.<sup>4</sup>

The graph Laplacian has an important role in describing functions on a graph [50] and, in the case of image denoising, in representing the structure of the underlying signal [16, 2, 29, 21, 24]. There are several definitions of the graph Laplacian. In the context of the proposed SOS boosting, we shall use a normalized Laplacian, defined as

$$(5.3) \quad \mathcal{L} = \mathbf{I} - \mathbf{W},$$

where  $\mathbf{W}$  is a filter matrix representing the denoiser  $f(\cdot)$  (see (2.8)). Note that  $\mathbf{W}$  is a normalized version of the similarity matrix  $\mathbf{K}$  and thus has eigenvalues in a range of 0 to 1. There are several ways to obtain  $\mathbf{W}$  from  $\mathbf{K}$ . For example,  $\mathbf{W} = \mathbf{D}^{-1}\mathbf{K}$  is used in [44] and in this work (leading to a random walk Laplacian); another way is  $\mathbf{W} = \mathbf{D}^{-1/2}\mathbf{K}\mathbf{D}^{-1/2}$ , as used in [30]. Recently, Kheradmand and Milanfar [24] suggest  $\mathbf{W} = \mathbf{C}^{-1/2}\mathbf{K}\mathbf{C}^{-1/2}$ , where  $\mathbf{C}$  is the outcome of Sinkhorn algorithm [25]. Notice that different versions of  $\mathbf{W}$  result in different properties of  $\mathcal{L}$  (refer to [24] for more information).

In general, the spectrum of a graph is defined by the eigenvectors and eigenvalues of  $\mathcal{L}$ . In the context of image denoising, as argued in [18, 30, 21, 24], the eigenvectors that correspond to the small eigenvalues of  $\mathcal{L}$  encapsulate the underlying structure of the image. On the other hand, the eigenvectors that correspond to the large eigenvalues mostly represent the noise. Meyer and Shen [30] showed that the small eigenvalues are stable even for high noise scenarios. As a result, the graph Laplacian can be used as a regularizer, preserving the geometry of the image by encouraging similar pixels to remain similar in the final estimate [16, 2].

What can we do with  $\mathcal{L}$ ? The obvious usage of it is as an image adaptive regularizer in inverse problems. There are several ways to integrate  $\mathcal{L}$  in a cost function; for example, [16, 2] suggest solving the minimization problem<sup>5</sup>

$$(5.4) \quad \hat{\mathbf{x}} = \min_{\mathbf{x}} \|\mathbf{x} - \mathbf{y}\|_2^2 + \rho \mathbf{x}^T \mathcal{L} \mathbf{x},$$

<sup>4</sup>The K-SVD degree matrix, as defined in (2.8), also holds the relation described in (5.2). According to Theorem 2.1,  $\mathbf{1}$  is an eigenvector of  $\mathbf{W} = \mathbf{D}^{-1}\mathbf{K}$ , corresponding to eigenvalue  $\lambda = 1$ , leading to  $\mathbf{D}^{-1}\mathbf{K}\mathbf{1} = \mathbf{1}$ . Multiplying both sides by  $\mathbf{D}$  results in the desired relation  $\mathbf{K}\mathbf{1} = \mathbf{D}\mathbf{1}$ , i.e.,  $\mathbf{D}(i, i) = \sum_j \mathbf{K}(i, j)$ .

<sup>5</sup>The work in [21] is closely related, but their regularization term is  $\|\mathbf{L}\mathbf{x}\|_2^2$ , and thus it leads to  $\mathbf{L}^T\mathbf{L}$  in the steady-state formula, where  $\mathbf{L} = \mathbf{D} - \mathbf{K}$  is an unnormalized graph Laplacian. Thus, we omit it from the next discussion.

leading to a closed-form expression for  $\hat{\mathbf{x}}$ ,

$$(5.5) \quad \hat{\mathbf{x}} = (\mathbf{I} + \rho\mathcal{L})^{-1}\mathbf{y}.$$

The authors of [24] suggest an iterative graph-based framework for image restoration. Specifically, in the case of image denoising [23], they suggest a variant of (5.4),

$$(5.6) \quad \hat{\mathbf{x}} = \min_{\mathbf{x}} (\mathbf{x} - \mathbf{y})^T \mathbf{W}(\mathbf{x} - \mathbf{y}) + \rho \mathbf{x}^T \mathcal{L} \mathbf{x}.$$

Unlike (5.4), the above expression offers a weighted data fidelity term, resulting in the following closed-form expression to the final estimate:

$$(5.7) \quad \hat{\mathbf{x}} = (\mathbf{W} + \rho\mathcal{L})^{-1}\mathbf{W}\mathbf{y}.$$

It turns out that (3.16), the steady-state result of the SOS boosting, i.e.,

$$(5.8) \quad \begin{aligned} \hat{\mathbf{x}}^* &= (\mathbf{I} + \rho(\mathbf{I} - \mathbf{W}_*))^{-1} \mathbf{W}_* \mathbf{y} \\ &= (\mathbf{I} + \rho\mathcal{L}_*)^{-1} \mathbf{W}_* \mathbf{y}, \end{aligned}$$

can be also treated as emerging from a graph Laplacian regularizer, being the outcome of the following cost function:

$$(5.9) \quad \hat{\mathbf{x}}^* = \min_{\mathbf{x}} \|\mathbf{x} - \mathbf{W}_* \mathbf{y}\|_2^2 + \rho \mathbf{x}^T \mathcal{L}_* \mathbf{x}.$$

Notice the differences between (5.4), (5.6), and (5.9). The last expression suggests that SOS aims to find an image that is close to the estimated image  $\mathbf{W}_* \mathbf{y}$  rather than the noisy  $\mathbf{y}$  itself. In the spirit of the SOS boosting,

$$\hat{\mathbf{x}}^{k+1} = f(\mathbf{y} + \rho\hat{\mathbf{x}}^k) - \rho\hat{\mathbf{x}}^k,$$

we can suggest expressing the above-mentioned graph Laplacian regularization methods, i.e., (5.5) and (5.7), as recursive, providing novel black-box iterative algorithms that minimize their corresponding penalty functions without explicitly building the matrix  $\mathbf{W}$ . Starting with (5.5), the steady-state outcome should satisfy

$$(5.10) \quad (\mathbf{I} + \rho(\mathbf{I} - \mathbf{W})) \hat{\mathbf{x}} = \mathbf{y}.$$

There are many ways to rearrange this expression, using the fixed point strategy, in order to get a recursive update formula. We shall adopt a path that leads to an iterative process that operates on the strengthened image,  $\mathbf{y} + \hat{\mathbf{x}}^k$ , in order to expose the similarities to and differences from our scheme. Therefore, we suggest adding  $\mathbf{W}\mathbf{y} - \mathbf{W}\hat{\mathbf{x}}$  to the RHS, i.e.,

$$(5.11) \quad \hat{\mathbf{x}} + \rho\hat{\mathbf{x}} - \rho\mathbf{W}\hat{\mathbf{x}} = \mathbf{y} + \mathbf{W}\mathbf{y} - \mathbf{W}\mathbf{y}.$$

Rearranging the above expression results in

$$(5.12) \quad \hat{\mathbf{x}} = \frac{1}{(1 + \rho)} [\mathbf{W}(\mathbf{y} + \rho\hat{\mathbf{x}}) + (\mathbf{y} - \mathbf{W}\mathbf{y})].$$



As a consequence, the obtained iterative black-box formulation to the conventional graph Laplacian regularization [16, 2] is given by

$$(5.13) \quad \hat{\mathbf{x}}^{k+1} = \frac{1}{(1 + \rho)} \left[ f(\mathbf{y} + \rho \hat{\mathbf{x}}^k) + (\mathbf{y} - f(\mathbf{y})) \right].$$

As can be seen, we got an iterative algorithm that, similar to SOS, operates on the strengthened image. However, rather than simply subtracting  $\rho \hat{\mathbf{x}}^k$  from the outcome, we add the method noise and then normalize.

In a similar way, (5.7), which is formulated as

$$(5.14) \quad (\mathbf{W} + \rho(\mathbf{I} - \mathbf{W})) \hat{\mathbf{x}} = \mathbf{W}\mathbf{y},$$

can be expressed by

$$(5.15) \quad \hat{\mathbf{x}} = \frac{1}{\rho} \mathbf{W}(\rho \hat{\mathbf{x}} + \mathbf{y} - \hat{\mathbf{x}}),$$

and in the general case, the black-box version of [23] is formulated by

$$(5.16) \quad \hat{\mathbf{x}}^{k+1} = \frac{1}{\rho} f(\rho \hat{\mathbf{x}}^k + \mathbf{y} - \hat{\mathbf{x}}^k) = \frac{1}{\rho' + 1} f(\mathbf{y} + \rho' \hat{\mathbf{x}}^k),$$

where  $\rho' = \rho - 1$ . Again, we see a close resemblance to our SOS method. However, instead of subtracting  $\rho \hat{\mathbf{x}}^k$  from the denoised strengthened image, we simply normalize accordingly.

Equations (5.13) and (5.16) offer two iterative algorithms that are essentially minimizing the penalty functions (5.4) and (5.6), respectively. However, these algorithms offer far more; both can be applied with the denoiser as a black-box, implying that no explicit matrix construction of  $\mathbf{W}$  (or  $\mathcal{L}$ ) is required. Furthermore, these schemes, given in the form of denoising on the strengthened image, imply that parameter setting is trivial; the noise level is nearly  $\sigma$  regardless of the iteration number. Finally, an update of  $\mathbf{W}$  within the iterations of these recursive formulas seems most natural.

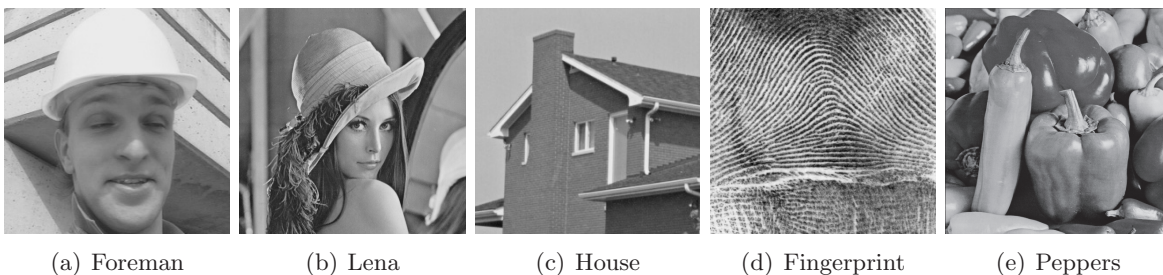


Figure 2. Visualization of the test images.

**6. Experimental results.** In this section, we provide detailed results of the SOS boosting and its local-global variant, “sharing the disagreement.” The results are presented for the images Foreman, Lena, House, Fingerprint, and Peppers (see Figure 2). These images are

extensively tested in earlier work, thus enabling a convenient and fair demonstration of the potential of the proposed boosting. The images are corrupted by an additive zero-mean Gaussian noise with a standard deviation  $\sigma$ . The denoising performance is evaluated using the PSNR, defined as  $20 \log_{10}(\frac{255}{\sqrt{\text{MSE}}})$ , where MSE is the mean squared error between the original image and its denoised version.

**6.1. SOS boosting with state-of-the-art algorithms.** The proposed SOS boosting is applicable to a wide range of denoising algorithms. We demonstrate its abilities by improving several state-of-the-art methods: (i) K-SVD [15], (ii) NLM [6, 7], (iii) BM3D [13], and (iv) EPLL [53]. The K-SVD [15], which was discussed in detail in this paper, is based on an adaptive sparsity model. The NLM [6] leverages the “self-similarity” property of natural images, i.e., the assumption that each patch may have similar patches within the image. The BM3D [13] combines the self-similarity property with a sparsity model, achieving the best restoration, and even touches some recently developed image denoising bounds [28]. The EPLL [53], which was described in section 1, represents the image patches using the Gaussian mixture model (GMM) and encourages the global result to comply with the prior local patches. As can be inferred, these algorithms are diverse and build upon different models and forces. Furthermore, the EPLL itself can be considered as a boosting method designed to improve a GMM denoising algorithm. The diversity of the above algorithms emphasizes the potential of the SOS boosting.

The improved denoising performance is gained simply by applying the authors’ original software as a black-box, *without any internal algorithmic modifications or parameters settings*.<sup>6</sup> Such modifications may lead to better results, and we leave these for future study. In order to apply SOS boosting we need to set the parameters  $\rho$ ,  $\tau$  and a modified noise-level  $\hat{\sigma}$  (although  $\sigma$  is known). The parameter  $\hat{\sigma}$ , which might be a little higher than  $\sigma$ , represents the noise-level of  $\mathbf{y} + \rho \hat{\mathbf{x}}^k$ . We can estimate  $\hat{\sigma}$  automatically (e.g., using [52]) or tune a fixed value manually. In the following experiments we choose the second option. We set  $\tau = 1$  (the effect of  $\tau^*$  is demonstrated later) and run several tests to tune  $\rho$  and  $\hat{\sigma}$  for each noise-level and denoising algorithm, as detailed in Table 1 under the “SOS params” column.

In the case of the EPLL and BM3D, the authors’ software is designed to denoise an input image in the range of 0 to 1. As such, we apply the SOS boosting ( $\tau = 1$ ) in the following formulation:

$$(6.1) \quad \hat{\mathbf{x}}^{k+1} = \frac{1}{1 - \tilde{\rho}} \cdot f \left( (1 - \tilde{\rho})\mathbf{y} + \tilde{\rho} \hat{\mathbf{x}}^k \right) - \frac{\tilde{\rho}}{1 - \tilde{\rho}} \cdot \hat{\mathbf{x}}^k,$$

with a corresponding  $\tilde{\sigma}$ . In order to remain consistent with the SOS parameters of the K-SVD and NLM, which apply (3.14), we provide hereafter the parameters  $\rho = \frac{\tilde{\rho}}{1 - \tilde{\rho}}$  and  $\hat{\sigma} = \frac{\tilde{\sigma}}{1 - \tilde{\rho}}$  for the EPLL and BM3D.

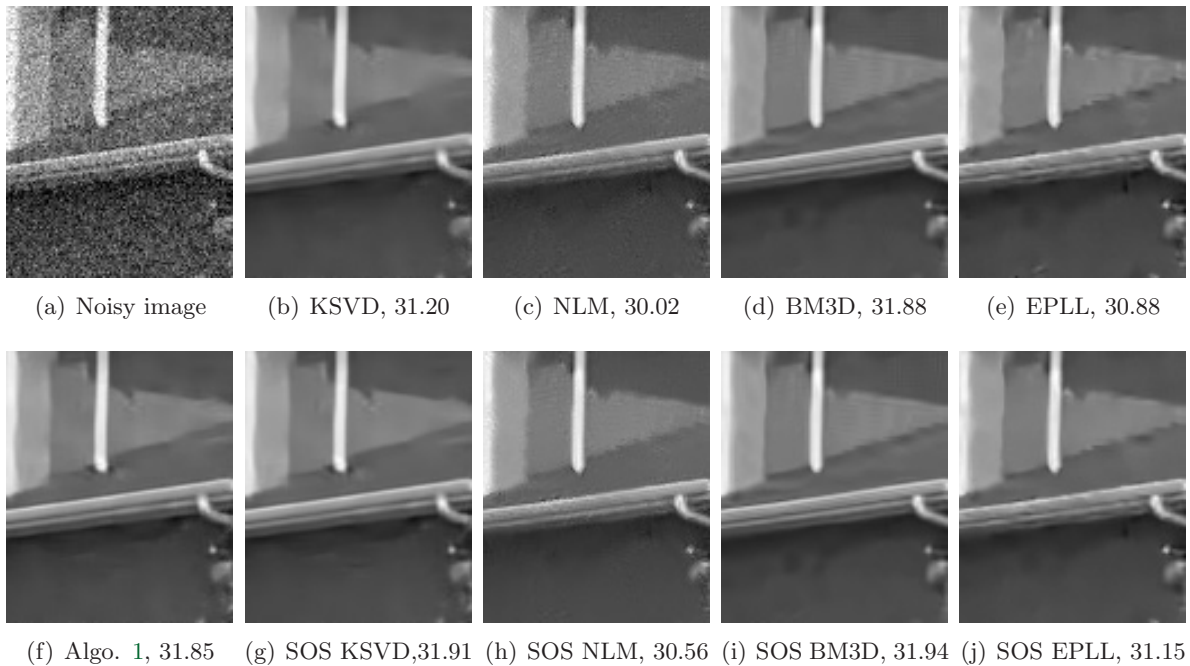
Table 1 lists the restoration results of various denoising algorithms and their SOS versions. The PSNR values that appear in the “Orig” column are obtained by applying the denoising algorithm on  $\mathbf{y}$  using the input noise level  $\sigma$  (and not  $\hat{\sigma}$  as done at the consecutive SOS steps).

<sup>6</sup>The original K-SVD uses  $8 \times 8$  patches, but our experiments show that  $9 \times 9$  yields nearly the same results for the core algorithm, while enabling better improvement with the SOS boosting. As a consequence, in the following experiments we demonstrate the results of the  $9 \times 9$  version.

Table 1

Comparison between the denoising results (PSNR) of various algorithms (K-SVD [15], NLM [7], BM3D [13], and EPLL [53]) and their SOS boosting outcomes. For each denoising algorithm, we apply the authors' original software with the SOS formulation (using  $\tau = 1$ , with the appropriate  $\rho$  and  $\hat{\sigma}$ ). The best results for each denoising algorithm, image, and noise level are highlighted.

K-SVD [15]															
$\sigma$	SOS params		Foreman		Lena		House		Fingerprint		Peppers		Average		
	$\rho$	$\hat{\sigma}$	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Imprv.
10	0.30	$1.00\sigma$	36.92	<b>37.13</b>	35.47	<b>35.58</b>	36.25	<b>36.49</b>	32.27	<b>32.35</b>	34.68	<b>34.71</b>	35.12	<b>35.25</b>	0.13
20	0.60	$1.00\sigma$	33.81	<b>34.11</b>	32.43	<b>32.67</b>	33.34	<b>33.62</b>	28.31	<b>28.54</b>	32.29	<b>32.35</b>	32.04	<b>32.26</b>	0.22
25	1.00	$1.00\sigma$	32.83	<b>33.12</b>	31.32	<b>31.62</b>	32.39	<b>32.72</b>	27.13	<b>27.44</b>	31.43	<b>31.49</b>	31.02	<b>31.28</b>	0.26
50	1.00	$1.00\sigma$	28.88	<b>29.85</b>	27.75	<b>28.37</b>	28.01	<b>28.98</b>	23.20	<b>23.98</b>	28.16	<b>28.66</b>	27.20	<b>27.97</b>	0.77
75	1.00	$1.00\sigma$	26.24	<b>27.32</b>	25.74	<b>26.40</b>	25.23	<b>26.85</b>	19.93	<b>21.88</b>	25.73	<b>26.72</b>	24.57	<b>25.83</b>	1.26
100	1.00	$1.00\sigma$	25.21	<b>25.39</b>	24.50	<b>24.99</b>	23.69	<b>24.59</b>	17.98	<b>19.61</b>	24.17	<b>25.03</b>	23.11	<b>23.92</b>	0.81
NLM [7]															
$\sigma$	SOS params		Foreman		Lena		House		Fingerprint		Peppers		Average		
	$\rho$	$\hat{\sigma}$	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Imprv.
10	0.10	$1.20\sigma$	35.55	<b>36.13</b>	34.32	<b>34.72</b>	34.93	<b>35.39</b>	31.04	<b>31.45</b>	34.02	<b>34.37</b>	33.97	<b>34.41</b>	0.44
20	0.10	$1.10\sigma$	32.78	<b>33.15</b>	31.59	<b>31.84</b>	32.40	<b>32.86</b>	27.26	<b>27.55</b>	31.49	<b>31.78</b>	31.10	<b>31.44</b>	0.34
25	0.40	$1.10\sigma$	31.26	<b>31.88</b>	30.51	<b>30.88</b>	31.22	<b>31.87</b>	26.20	<b>26.22</b>	30.47	<b>30.85</b>	29.93	<b>30.34</b>	0.41
50	0.50	$1.05\sigma$	27.62	<b>28.05</b>	27.31	<b>27.57</b>	27.42	<b>28.00</b>	23.00	<b>23.06</b>	26.79	<b>26.97</b>	26.43	<b>26.73</b>	0.30
75	0.60	$1.05\sigma$	25.38	<b>26.06</b>	25.12	<b>25.75</b>	24.59	<b>25.49</b>	20.84	<b>21.13</b>	24.63	<b>24.94</b>	24.11	<b>24.67</b>	0.56
100	0.60	$1.05\sigma$	23.82	<b>24.21</b>	23.71	<b>24.17</b>	23.07	<b>23.45</b>	19.50	<b>19.67</b>	23.27	<b>23.65</b>	22.67	<b>23.03</b>	0.36
BM3D [13]															
$\sigma$	SOS params		Foreman		Lena		House		Fingerprint		Peppers		Average		
	$\rho$	$\hat{\sigma}$	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Imprv.
10	0.05	$1.02\sigma$	37.23	<b>37.24</b>	35.84	<b>35.85</b>	36.54	<b>36.55</b>	32.46	<b>32.47</b>	34.96	34.96	35.40	<b>35.41</b>	0.01
20	0.11	$1.03\sigma$	34.50	<b>34.55</b>	33.00	<b>33.02</b>	33.81	33.81	28.82	<b>28.83</b>	32.67	<b>32.68</b>	32.56	<b>32.58</b>	0.02
25	0.18	$1.04\sigma$	33.41	<b>33.48</b>	32.02	<b>32.04</b>	32.90	32.90	27.72	27.72	31.87	<b>31.89</b>	31.58	<b>31.61</b>	0.03
50	0.25	$1.04\sigma$	30.22	<b>30.36</b>	28.98	<b>29.00</b>	29.68	<b>29.80</b>	24.57	<b>24.59</b>	29.09	<b>29.14</b>	28.51	<b>28.58</b>	0.07
75	0.43	$1.04\sigma$	28.09	<b>28.30</b>	27.15	<b>27.21</b>	27.73	<b>27.95</b>	22.84	<b>22.88</b>	27.09	<b>27.11</b>	26.58	<b>26.69</b>	0.11
100	0.43	$1.11\sigma$	26.16	<b>26.42</b>	25.77	<b>25.82</b>	25.74	<b>25.93</b>	21.56	<b>21.67</b>	25.72	<b>25.81</b>	24.99	<b>25.13</b>	0.14
EPLL [53]															
$\sigma$	SOS params		Foreman		Lena		House		Fingerprint		Peppers		Average		
	$\rho$	$\hat{\sigma}$	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Imprv.
10	0.09	$1.11\sigma$	36.98	<b>37.09</b>	35.53	<b>35.66</b>	35.67	<b>35.73</b>	32.12	<b>32.33</b>	34.82	<b>34.93</b>	35.02	<b>35.15</b>	0.13
20	0.09	$1.11\sigma$	33.70	<b>34.03</b>	32.57	<b>32.77</b>	33.06	<b>33.33</b>	28.26	<b>28.49</b>	32.48	<b>32.69</b>	32.01	<b>32.26</b>	0.25
25	0.18	$1.11\sigma$	32.44	<b>32.78</b>	31.62	<b>31.84</b>	32.07	<b>32.38</b>	27.14	<b>27.30</b>	31.59	<b>31.87</b>	30.97	<b>31.23</b>	0.26
50	0.43	$1.11\sigma$	29.24	<b>29.60</b>	28.39	<b>28.66</b>	28.78	<b>29.24</b>	23.63	<b>23.69</b>	28.67	<b>29.00</b>	27.74	<b>28.04</b>	0.30
75	0.43	$1.11\sigma$	27.17	<b>27.55</b>	26.53	<b>26.85</b>	26.78	<b>27.28</b>	21.51	<b>21.54</b>	26.73	<b>27.10</b>	25.74	<b>26.06</b>	0.32
100	0.43	$1.11\sigma$	25.58	<b>25.91</b>	25.23	<b>25.49</b>	25.08	<b>25.47</b>	19.77	19.77	25.36	<b>25.73</b>	24.20	<b>24.47</b>	0.27

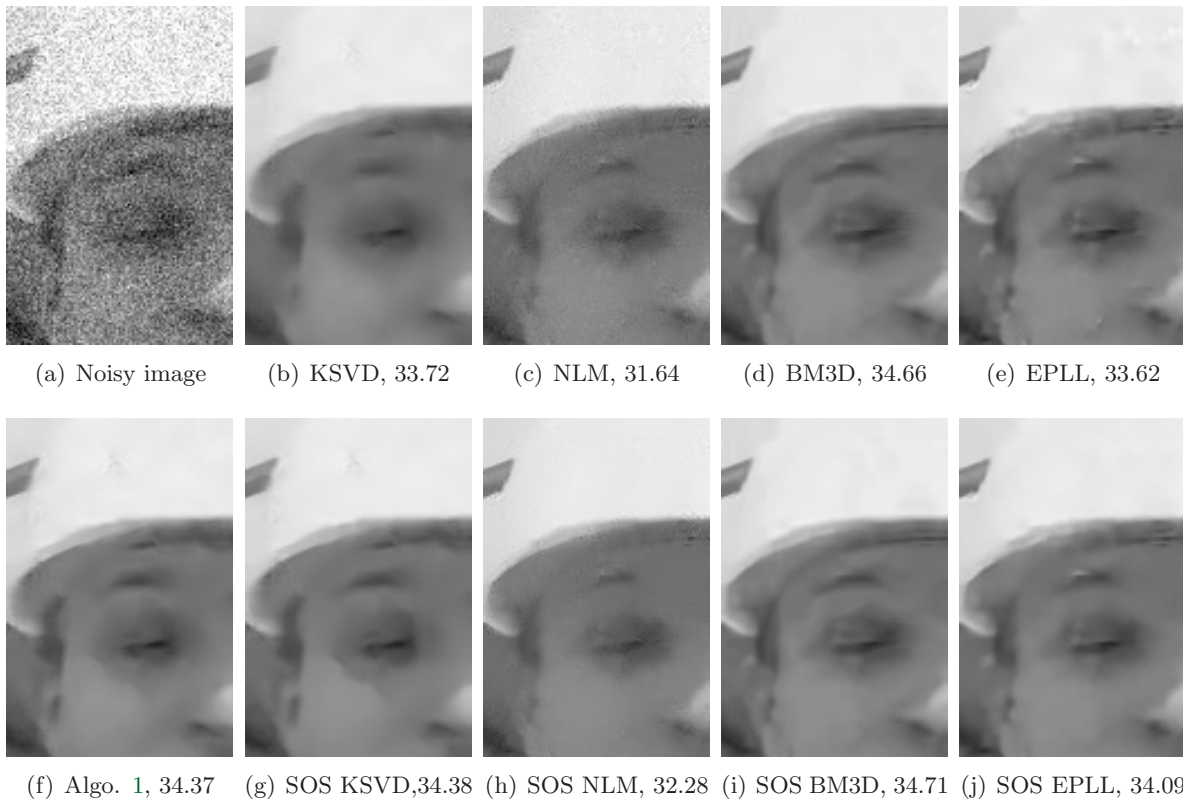


**Figure 3.** Visual and PSNR comparisons between standard denoising and boosting outcomes of a  $100 \times 120$  cropped region from noisy image House ( $\sigma = 25$ ).

These are also the first estimates of the SOS boosting (i.e.,  $\hat{\mathbf{x}}^1$ ). In the case of the K-SVD denoising [15], at the first SOS step we apply 20 iterations of sparse-coding and dictionary update, while at the remaining SOS steps we apply only two such iterations (we found this to be a convenient compromise between runtime and performance). We operate the K-SVD [15], NLM [7], BM3D [13], and EPLL [53] for 30, 2, 3, and 4 SOS steps, respectively.

The ‘Average Imprv.’ column in Table 1 indicates that the SOS boosting achieves an improvement over the original denoising algorithms. More specifically, for all denoising algorithms, images, and noise levels, the SOS outcomes are at least as good as the original results and, more importantly, usually better (in terms of PSNR). A clear improvement over the whole range of noise-levels is achieved for the K-SVD [15], NLM [7, 6], and EPLL [53]. While in the case of the BM3D [13], we succeed in improving it slightly, mainly for high noise energy. The fact that the BM3D performance is very close to the denoising bound posed in [28] explains the difficulties in improving it. A visual comparison is given in Figures 3, 4, and 5, illustrating the effectiveness of the SOS boosting. Compared to the original results, the SOS offers better restoration of edges (in the case of the K-SVD, focus on the house’s roof, the foreman’s eye and ear, and Lena’s hat). In addition, the SOS obtains cleaner estimations (when using the NLM) and fewer artifacts (for the EPLL and somewhat also for the BM3D).

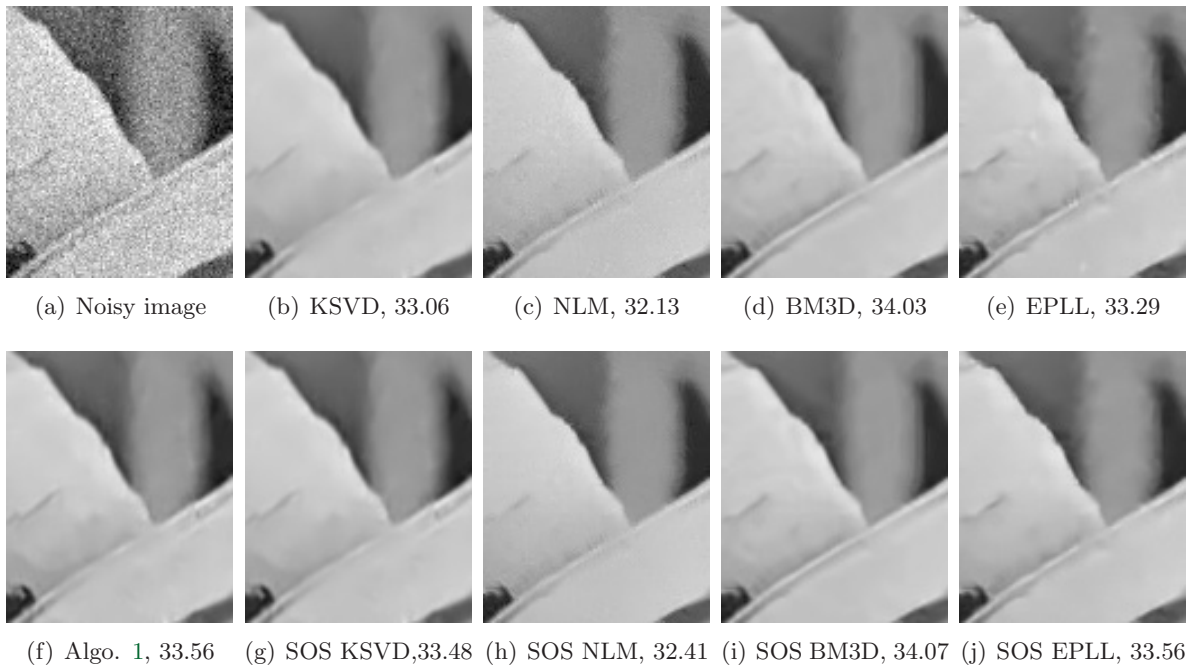
In the context of the K-SVD denoising, based on (3.25), we demonstrate the effect of  $\tau^*$  on the SOS recursive function. Note that we do not test its influence on the other denoising algorithms because the information about their eigenvalues’ range, which is required in (3.25), is not derived. Figure 6 plots the average PSNR over the test images ( $\sigma = 50$ ), as a function



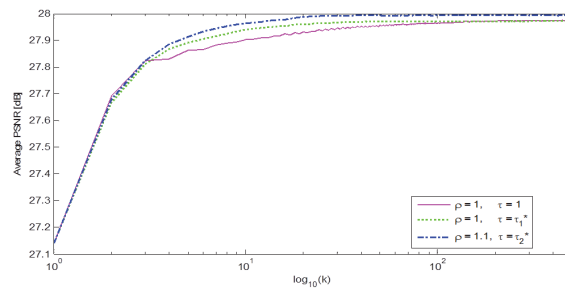
**Figure 4.** Visual and PSNR comparisons between standard denoising and boosting outcomes of a  $100 \times 150$  cropped region from noisy image Foreman ( $\sigma = 25$ ).

of the SOS-step, for three different parameter settings: First, as a baseline, we apply the SOS with  $\rho = 1$  and  $\tau = 1$  (without using the closed-form expression for  $\tau^*$ ). Second, we improve the convergence rate by using  $\tau^*$  with the same signal-emphasis factor ( $\rho = 1$ ). Third, we plot the PSNR obtained by the couple that leads to the best restoration ( $\rho = 1.1$  with the corresponding  $\tau^*$ ). As a reminder, according to section 3.3 and Appendix D, the parameters  $\rho$  and  $\tau$  affect the conditions for convergence and its rate. More specifically, a modification of  $\rho$  without an adjustment of  $\tau$  may violate the condition for convergence (e.g., according to condition (3.22), the couple  $\rho = 1.1$  and  $\tau = 1$  results in  $\gamma > 1$ ). Therefore, using the parameter  $\tau^*$  enables modification of  $\rho$  without violating the requirements for convergence, even leading to the fastest rate of convergence. These results are consistent with Table 2, which lists the achieved PSNR when applying the SOS for 30 steps using the best  $\rho$  and  $\tau^*$  (per noise level). As can be seen,  $\tau^*$  not only results in a faster convergence but also allows a stronger emphasis of the estimated signal, thus leading to better restoration.

**6.2. Sharing the disagreement.** We demonstrate the effectiveness of the local-global interpretation of the SOS boosting, which was described in section 4. The denoising results of Table 3 are obtained by applying Algorithm 1 for 30 steps, where each step includes two



**Figure 5.** Visual and PSNR comparisons between standard denoising and boosting outcomes of a  $100 \times 120$  cropped region from noisy image Lena ( $\sigma = 20$ ).



**Figure 6.** Demonstration of the effect of  $\tau^*$  on the SOS boosting outcome for the K-SVD denoising ( $\sigma = 50$ ).

sparse-coding and dictionary update iterations. The initial dictionary is obtained by applying 20 iterations of the K-SVD algorithm. Similarly to the SOS boosting, we tune the parameter  $\hat{\sigma}$  for each input  $\sigma$  (this variant is limited to  $\rho = 1$  and  $\tau = 1$ ).

According to Table 3, for all images and noise levels, “sharing the disagreement” boosting achieves a clear improvement over the original K-SVD algorithm [15]. Notice the resemblance and the differences in the PSNR values between Table 3 and the K-SVD part in Table 1. In general, the differences originate from the nonlinearity of the denoising algorithm; the input patch to the sparse-coding step differs between the SOS and its local-global variant. As a reminder, the equivalence between these two approaches is valid under the assumption of a fixed filter-matrix (see Appendix C for more details). Furthermore, in the case of SOS

Table 2

Denoising results (PSNR) of the K-SVD [15] and its SOS boosting outcomes, where we use the parameter  $\tau^*$  (according to (3.25)) along with the appropriate  $\rho$  and  $\hat{\sigma}$ . The best results for each image and noise level are highlighted.

$\sigma$	SOS params		Foreman		Lena		House		Fingerprint		Peppers		Average		
	$\rho$	$\hat{\sigma}$	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Orig	SOS	Imprv.
10	0.30	$1.00\sigma$	36.92	<b>37.14</b>	35.47	<b>35.58</b>	36.25	<b>36.49</b>	32.27	<b>32.35</b>	34.68	<b>34.72</b>	35.12	<b>35.26</b>	0.14
20	0.60	$1.00\sigma$	33.81	<b>34.11</b>	32.43	<b>32.68</b>	33.34	<b>33.62</b>	28.31	<b>28.54</b>	32.29	<b>32.35</b>	32.04	<b>32.26</b>	0.22
25	1.00	$1.00\sigma$	32.83	<b>33.12</b>	31.32	<b>31.65</b>	32.39	<b>32.74</b>	27.13	<b>27.46</b>	31.43	<b>31.53</b>	31.02	<b>31.30</b>	0.28
50	1.10	$1.00\sigma$	28.88	<b>29.86</b>	27.75	<b>28.42</b>	28.01	<b>29.05</b>	23.20	<b>24.03</b>	28.16	<b>28.68</b>	27.20	<b>28.01</b>	0.81
75	1.20	$1.00\sigma$	26.24	<b>27.36</b>	25.74	<b>26.50</b>	25.23	<b>27.08</b>	19.93	<b>22.02</b>	25.73	<b>26.80</b>	24.57	<b>25.95</b>	1.38
100	1.20	$1.00\sigma$	25.21	<b>25.46</b>	24.50	<b>25.09</b>	23.69	<b>24.70</b>	17.98	<b>19.93</b>	24.17	<b>25.17</b>	23.11	<b>24.07</b>	0.96

Table 3

Comparison between the denoising results (PSNR) of the original K-SVD algorithm [15] and its “sharing the disagreement” boosting outcome (Algorithm 1). The best results for each image and noise level are highlighted.

$\sigma$	$\hat{\sigma}$	Foreman		Lena		House		Fingerprint		Peppers		Average		
		Orig	Boost	Orig	Boost	Orig	Boost	Orig	Boost	Orig	Boost	Orig	Boost	Imprv.
10	$1.08\sigma$	36.92	<b>37.13</b>	35.47	<b>35.58</b>	36.25	<b>36.34</b>	32.27	<b>32.35</b>	34.68	<b>34.70</b>	35.12	<b>35.22</b>	0.10
20	$1.02\sigma$	33.81	<b>34.11</b>	32.43	<b>32.68</b>	33.34	<b>33.56</b>	28.31	<b>28.59</b>	32.29	<b>32.37</b>	32.04	<b>32.26</b>	0.22
25	$1.02\sigma$	32.83	<b>33.17</b>	31.32	<b>31.64</b>	32.39	<b>32.71</b>	27.13	<b>27.47</b>	31.43	<b>31.60</b>	31.02	<b>31.32</b>	0.30
50	$1.00\sigma$	28.88	<b>29.37</b>	27.75	<b>28.28</b>	28.01	<b>28.67</b>	23.20	<b>24.04</b>	28.16	<b>28.55</b>	27.20	<b>27.78</b>	0.58
75	$1.00\sigma$	26.24	<b>27.04</b>	25.74	<b>26.28</b>	25.23	<b>26.54</b>	19.93	<b>21.76</b>	25.73	<b>26.52</b>	24.57	<b>25.63</b>	1.06
100	$1.00\sigma$	25.21	<b>25.28</b>	24.50	<b>24.91</b>	23.69	<b>24.43</b>	17.98	<b>19.82</b>	24.17	<b>24.92</b>	23.11	<b>23.87</b>	0.76

boosting, more freedom is obtained by tuning the parameters  $\rho$  and  $\tau$ , which may lead to better utilization of the prior (as shown in Figure 6 and Table 2). However, visually, according to Figures 3, 4, and 5, the outcomes of the SOS and its local-global variant are very similar; both of them effectively improve the restoration of the underlying signal.

To conclude, we demonstrate the potential of the SOS boosting and its local-global interpretation. The proposed algorithm achieves a clear and meaningful improvement over the examined state-of-the-art denoising algorithms, both visually and in terms of PSNR.

**7. Conclusions and future directions.** We have presented the SOS boosting, a generic method for improving various image denoising algorithms. The improvement is achieved by treating the denoiser as a black-box and repeating three simple SOS steps: (i) *Strengthening* the signal, (ii) *Operating* the denoising algorithm, and (iii) *Subtracting* the previous denoised image from the result. In addition, we provided an interesting local-global interpretation, called “sharing the disagreement” boosting, indicating that the SOS boosting not only leverages the improved SNR of the estimates but also reduces the gap between the local patch processing and the global need for a whole denoised image, all in the context of the K-SVD denoising algorithm. We also constructed the matrix-formulation of the K-SVD (and similar algorithms), showing that its eigenvalues are in the range of 0 to 1. Under these conditions, we have studied the convergence of the SOS boosting recursive function, leading to the conclu-

sion that for various known denoising algorithms, the SOS boosting is guaranteed to converge. Moreover, a generalization of the SOS function has been obtained by introducing two parameters that govern the steady-state result and soften the requirements for convergence (the eigenvalues' range) and the rate of convergence. We also provided a closed-form expression for the parameter that leads to the fastest convergence.

Finally, we have introduced a graph-based interpretation, showing that the SOS boosting acts as a graph Laplacian regularization method, thus effectively estimating the structure of the underlying signal. Inspired by the SOS scheme, we suggested novel recursive algorithms that treat the denoiser as a black-box in order to minimize related graph Laplacian objective functions, without explicitly constructing the weighted graph.

The proposed algorithm is easy to use, it reduces the local-global gap, acting as a graph Laplacian regularizer, it is applicable to a wide range of denoising algorithms, and it converges. These attributes make it a powerful and convenient tool for improving various denoising algorithms, as demonstrated in the experiments.

It is intriguing to study the proposed iterative algorithms that are defined in section 5 (which minimize different graph Laplacian cost functions). They may lead to better results than the original methods due to the nonlinearity of the denoiser and its adaptivity to the signal-strengthened image. We hope that other restoration problems, such as superresolution [34] and interpolation/inpainting [38], could also benefit from a similar concept.

**Appendix A. Periodic boundary condition.** Following (2.8), the periodic boundary condition affects the term  $\sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i$ , which is a diagonal matrix that counts the number of appearances per pixel in the final patch-averaging. Due to boundary effects, the values along the diagonal in this matrix are different (since the number of overlapping patches in the image borders is smaller than in other areas). As shown in Appendix B, the numerator of Equation (2.8) is a symmetric and positive definite matrix. Each row of this matrix is normalized by the number of overlapping patches, i.e., by the corresponding diagonal element from  $(\mu\mathbf{I} + \sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i)$ . As a result, the rows and columns are normalized by different constants which ruin the symmetric property. By assuming the periodic boundary condition, all the pixels have the same number of representations, which equals to the patch size. In this case, we get  $\sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i = n\mathbf{I}$ , where  $n$  is the patch size, and thus the rows and columns are normalized by the same constant, which preserves the symmetric property of  $\mathbf{W}$ .

**Appendix B. Properties of the K-SVD filter-matrix.**

*Proof of property 1: Symmetric  $\mathbf{W} = \mathbf{W}^T$ .* Following (2.8) and based on the assumption of the periodic boundary condition (see Appendix A), the matrix  $\mathbf{W}$  can be expressed as

$$\begin{aligned}
 \text{(B.1)} \quad \mathbf{W} &= \left( \mu\mathbf{I} + \sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i \right)^{-1} \left( \mu\mathbf{I} + \sum_{i=1}^N \mathbf{R}_i^T \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \mathbf{R}_i \right) \\
 &= (\mu\mathbf{I} + n\mathbf{I})^{-1} \left( \mu\mathbf{I} + \sum_{i=1}^N \mathbf{R}_i^T \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \mathbf{R}_i \right) \\
 &= \frac{1}{\mu + n} \left( \mu\mathbf{I} + \sum_{i=1}^N \mathbf{R}_i^T \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \mathbf{R}_i \right).
 \end{aligned}$$



Notice that the term  $\mathbf{Z}_i = \mathbf{R}_i^T \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \mathbf{R}_i$  is symmetric (in fact, it is also positive semidefinite (PSD), as it is built of  $\mathbf{R}_i^T \hat{\mathbf{Z}}_i \mathbf{R}_i$ , where  $\hat{\mathbf{Z}}_i$  is a projection matrix [22]). Thus,  $\mathbf{W}$  is built as a sum of  $N + 1$  matrices, each of them symmetric, which leads to the claimed symmetry,  $\mathbf{W} = \mathbf{W}^T$ . ■

*Proof of properties 2 and 3: Positive definite  $\mathbf{W} \succ 0$  and  $\lambda_{\min}(\mathbf{W}) \geq \frac{\mu}{\mu+n}$ .* As we have seen above,  $\mathbf{W}$  can be written as

$$\begin{aligned} \text{(B.2)} \quad \mathbf{W} &= \frac{\mu}{\mu+n} \mathbf{I} + \frac{1}{\mu+n} \sum_{i=1}^N \left( \mathbf{R}_i^T \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \mathbf{R}_i \right) \\ &= \frac{\mu}{\mu+n} \mathbf{I} + \frac{1}{\mu+n} \sum_{i=1}^N \mathbf{Z}_i \\ &= \mathbf{bI} + \mathbf{A}, \end{aligned}$$

where  $\mathbf{A} = \frac{1}{\mu+n} \sum_{i=1}^N \mathbf{Z}_i$  and  $\mathbf{b} = \frac{\mu}{\mu+n} > 0$ . As mentioned above,  $\mathbf{Z}_i$  is PSD, and therefore, according to (B.2),  $\mathbf{A}$  is a linear combination of  $\mathbf{Z}_i \succeq \mathbf{0}$ ; thus it is PSD as well [22]. Finally, based on the fact that the eigenvalues of  $\mathbf{A} + \mathbf{bI}$  are lower-bounded by  $\mathbf{b} > 0$ , we get that  $\mathbf{W} \succ \mathbf{0}$ , with minimal eigenvalue satisfying

$$\text{(B.3)} \quad \lambda_{\min}(\mathbf{W}) \geq \mathbf{b} = \frac{\mu}{\mu+n}. \quad \blacksquare$$

*Proof of property 4 (and 5):  $\mathbf{W}\mathbf{1} = \mathbf{W}^T\mathbf{1} = \mathbf{1}$ .* This property originates directly from the K-SVD denoising algorithm, which preserves the DC component of the image. In general, the trained dictionary is adapted to the image patches after their DC is removed. Once trained, the DC is returned as an additional atom  $d_0$ . Thus, this DC atom is necessarily orthogonal to the rest of the dictionary atoms. Each patch is represented by

$$\text{(B.4)} \quad \mathbf{D}_{s_i} = [d_0, d_1, d_2, \dots] = [d_0, \tilde{\mathbf{D}}_{s_i}],$$

where  $d_0 \in \mathbb{R}^n$  is the DC atom (the DC atom is included if the mean of the patch is not zero), and for the rest of the atoms (if any),  $d_i^T d_0 = 0$ , i.e.,  $d_0^T \tilde{\mathbf{D}}_{s_i} = \mathbf{0}$ . Note that the Gram matrix in this case is block-diagonal:

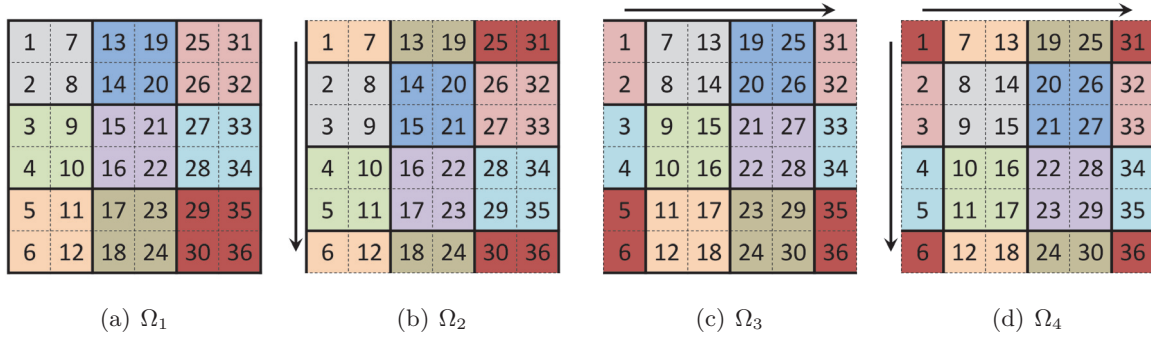
$$\text{(B.5)} \quad \mathbf{D}_{s_i}^T \mathbf{D}_{s_i} = \begin{bmatrix} 1 & 0 \\ \mathbf{0} & \tilde{\mathbf{D}}_{s_i}^T \tilde{\mathbf{D}}_{s_i} \end{bmatrix}.$$

Following (B.2), when multiplying  $\mathbf{W}$  by a constant image,  $\mathbf{1}$ , we get

$$\text{(B.6)} \quad \mathbf{W}\mathbf{1} = \frac{\mu}{\mu+n} \mathbf{1} + \frac{1}{\mu+n} \sum_{i=1}^N \mathbf{Z}_i \mathbf{1}.$$

Let us look at the term  $\mathbf{Z}_i \mathbf{1}$ :

$$\text{(B.7)} \quad \mathbf{Z}_i \mathbf{1} = \mathbf{R}_i^T \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \mathbf{R}_i \mathbf{1}.$$



**Figure 7.** Dividing  $r \times c = 6 \times 6$  input image (its pixels are numbered in  $1, 2, \dots, 36$ ) into  $\sqrt{n} \times \sqrt{n} = 2 \times 2$  overlapping patches (solid squares in different colors). We assume a cycling processing of the patches (periodic boundary condition); thus there are  $N = 36$  such patches, which can be divided into  $\{\Omega_j\}_{j=1}^4$  possible distinct groups of nonoverlapping patches: (a)  $\Omega_1$ , without any shift; (b)  $\Omega_2$ , down shift; (c)  $\Omega_3$ , right shift; (d)  $\Omega_4$ , down and right shift.

$\mathbf{R}_i \underline{1} = \underline{1}$ ; this is a shorter constant vector of length  $n$ .  $\mathbf{D}_{s_i}^T \underline{1} = [n, 0, 0, \dots]^T$  due to the orthogonality of the rest of the atoms to the DC. Multiplication of the inverse of  $\mathbf{D}_{s_i}^T \mathbf{D}_{s_i}$  results in  $[n, 0, 0, \dots]^T$ . The outcome of  $\mathbf{D}_{s_i} [n, 0, 0, \dots]^T = n \mathbf{d}_0 = \underline{1}$  is the desired DC patch of length  $n$ . Finally,  $\mathbf{R}_i^T$  returns the resulting constant patch back to its original location in the image.

Returning to (B.6), the input image  $\underline{1}$  is divided into  $N$  overlapping patches (one per pixel) of length  $n$ , where each DC patch is represented perfectly by  $\mathbf{Z}_i$ . Since each pixel appears in  $n$  patches, we get that  $\sum_{i=1}^N \mathbf{Z}_i \underline{1} = n \underline{1}$ . As a result,  $\mathbf{W} \underline{1} = \frac{\mu}{\mu+n} \underline{1} + \frac{n}{\mu+n} \underline{1} = \underline{1}$  preserves the DC of the image. Based on the symmetric property of  $\mathbf{W}$ , we get that  $\mathbf{W}^T \underline{1} = \mathbf{W} \underline{1} = \underline{1}$ . ■

*Proof of property 6 (and 7):*  $\|\mathbf{W}\|_2 = 1$ . In order to denoise the image, we break it into  $\sqrt{n} \times \sqrt{n}$  overlapping patches. The following proof relies on the observation that we can divide the  $N$  overlapping patches into  $\{\Omega_j\}_{j=1}^n$  distinct groups of nonoverlapping patches, as demonstrated in Figure 7. As a consequence, the matrix  $\mathbf{W}$ , which is a sum over  $N$  projection matrices (one for each patch), can be expressed as a sum over  $n$  distinct groups:

$$\begin{aligned}
 \text{(B.8)} \quad \mathbf{W} &= \frac{\mu}{\mu+n} \mathbf{I} + \frac{1}{\mu+n} \sum_{i=1}^N \left( \mathbf{R}_i^T \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \mathbf{R}_i \right) \\
 &= \frac{\mu}{\mu+n} \mathbf{I} + \frac{1}{\mu+n} \sum_{i=1}^N \left( \mathbf{R}_i^T \hat{\mathbf{Z}}_i \mathbf{R}_i \right) \\
 &= \frac{\mu}{\mu+n} \mathbf{I} + \frac{1}{\mu+n} \sum_{j=1}^n \left[ \sum_{k \in \Omega_j} \left( \mathbf{R}_k^T \hat{\mathbf{Z}}_k \mathbf{R}_k \right) \right] \\
 &= \frac{\mu}{\mu+n} \mathbf{I} + \frac{1}{\mu+n} \sum_{j=1}^n \tilde{\mathbf{W}}_j.
 \end{aligned}$$

Based on the property that  $\mathbf{R}_k \mathbf{R}_l^T = \mathbf{0}$  for all  $(k \neq l) \in \Omega_j$ , the following shows that  $\tilde{\mathbf{W}}_j$  is

an idempotent matrix:

$$\begin{aligned}
 \text{(B.9)} \quad (\tilde{\mathbf{W}}_j)^2 &= \left[ \sum_{k \in \Omega_j} (\mathbf{R}_k^T \hat{\mathbf{Z}}_k \mathbf{R}_k) \right] \left[ \sum_{l \in \Omega_j} (\mathbf{R}_l^T \hat{\mathbf{Z}}_l \mathbf{R}_l) \right] \\
 &= \sum_{k \in \Omega_j} \left( \mathbf{R}_k^T \hat{\mathbf{Z}}_k \mathbf{R}_k \right) \left( \mathbf{R}_k^T \hat{\mathbf{Z}}_k \mathbf{R}_k \right) \\
 &= \sum_{k \in \Omega_j} \left( \mathbf{R}_k^T \hat{\mathbf{Z}}_k \hat{\mathbf{Z}}_k \mathbf{R}_k \right) \\
 &= \sum_{k \in \Omega_j} \left( \mathbf{R}_k^T \hat{\mathbf{Z}}_k \mathbf{R}_k \right) \\
 &= \tilde{\mathbf{W}}_j,
 \end{aligned}$$

where we have used the equality  $\mathbf{R}_k \mathbf{R}_k^T = \mathbf{I} \in \mathbb{R}^{n \times n}$ , and

$$\begin{aligned}
 \text{(B.10)} \quad (\hat{\mathbf{Z}}_k)^2 &= \left( \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \right) \left( \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \right) \\
 &= \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \\
 &= \hat{\mathbf{Z}}_k.
 \end{aligned}$$

As a result, following (B.9), we can infer that  $\|\tilde{\mathbf{W}}_j\|_2 = 1$  [22]. Finally, using the matrix-norm inequalities, we get

$$\begin{aligned}
 \text{(B.11)} \quad \|\mathbf{W}\|_2 &= \left\| \frac{\mu}{\mu+n} \mathbf{I} + \frac{1}{\mu+n} \sum_{j=1}^n \tilde{\mathbf{W}}_j \right\|_2 \\
 &\leq \frac{\mu}{\mu+n} \cdot \|\mathbf{I}\|_2 + \frac{1}{\mu+n} \cdot \sum_{j=1}^n \|\tilde{\mathbf{W}}_j\|_2 \\
 &= \frac{\mu}{\mu+n} + \frac{n}{\mu+n} \\
 &= 1.
 \end{aligned}$$

To conclude, based on the above and by relying on the property that 1 is an eigenvalue of  $\mathbf{W}$ , we get that  $\lambda_{max}(\mathbf{W}) = 1$ , i.e.,  $\|\mathbf{W}\|_2 = 1$ . ■

*Proof of property 8:*  $\|\mathbf{W} - \mathbf{I}\|_2 \leq \frac{n}{\mu+n}$ . In general, the eigenvalues of  $\mathbf{A} + b\mathbf{I}$  are bigger than the eigenvalues of  $\mathbf{A}$  by the constant  $b$  [22]. Therefore, the eigenvalues of  $(\mathbf{W} - \mathbf{I})$  are equal to  $\lambda(\mathbf{W}) - 1$ . Based on  $\frac{\mu}{\mu+n} \leq \lambda(\mathbf{W}) \leq 1$ , we get that

$$\begin{aligned}
 \text{(B.12)} \quad \|\mathbf{W} - \mathbf{I}\|_2 &\leq |\lambda_{min}(\mathbf{W}) - 1| \\
 &= \left| \frac{\mu}{\mu+n} - 1 \right| \\
 &= \frac{n}{\mu+n}. \quad \blacksquare
 \end{aligned}$$

**Appendix C. Equivalence between the SOS boosting and sharing the disagreement procedure.** In the context of the K-SVD image denoising [15], we show an equivalence between the SOS boosting recursive function (1.4) and the disagreement and sharing approach (Algorithm 1). The following study assumes fixed supports and dictionary during the iterations, i.e., the projection matrix  $\mathbf{D}_{s_i}$  of the  $i$ th patch is known and fixed. In addition, a periodic boundary is considered (see Appendix A), and we use the K-SVD matrix form (see (B.1)) with  $\mu = 0$ , i.e.,

$$\begin{aligned} \mathbf{W} &= \frac{1}{n} \sum_{i=1}^N \mathbf{R}_i^T \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T \mathbf{R}_i \\ &= \frac{1}{n} \sum_{i=1}^N \mathbf{R}_i^T \hat{\mathbf{Z}}_i \mathbf{R}_i. \end{aligned} \quad (\text{C.1})$$

Following Algorithm 1, we denote by  $\mathbf{p}_i^k$  the  $k$ th iteration input patch to the denoising algorithm, which is influenced by the neighbors' information. Using the projection matrix  $\hat{\mathbf{Z}}_i = \mathbf{D}_{s_i} (\mathbf{D}_{s_i}^T \mathbf{D}_{s_i})^{-1} \mathbf{D}_{s_i}^T$ , let us compute the disagreement patch—the difference between an independent denoised patch, as defined in (2.6),

$$\hat{\mathbf{p}}_i^k = \hat{\mathbf{Z}}_i \mathbf{p}_i^k, \quad (\text{C.2})$$

and its corresponding patch from the global outcome (after patch-averaging),  $\mathbf{R}_i \hat{\mathbf{x}}^k$ , thus expressed by

$$\mathbf{q}_i^k = \hat{\mathbf{p}}_i^k - \mathbf{R}_i \hat{\mathbf{x}}^k. \quad (\text{C.3})$$

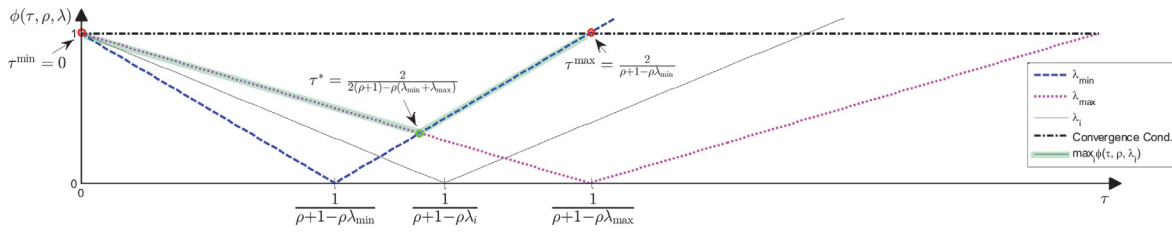
Next, we subtract the disagreement patch from the corresponding noisy one, i.e.,

$$\begin{aligned} \mathbf{p}_i^{k+1} &= \mathbf{R}_i \mathbf{y} - \mathbf{q}_i^k \\ &= \mathbf{R}_i \mathbf{y} - \hat{\mathbf{p}}_i^k + \mathbf{R}_i \hat{\mathbf{x}}^k \\ &= \mathbf{R}_i \mathbf{y} - \hat{\mathbf{Z}}_i \mathbf{p}_i^k + \mathbf{R}_i \hat{\mathbf{x}}^k. \end{aligned} \quad (\text{C.4})$$

Following (C.2), the denoised version of  $\mathbf{p}_i^{k+1}$  is given by

$$\begin{aligned} \hat{\mathbf{p}}_i^{k+1} &= \hat{\mathbf{Z}}_i \mathbf{p}_i^{k+1} \\ &= \hat{\mathbf{Z}}_i \mathbf{R}_i \mathbf{y} - \hat{\mathbf{Z}}_i \hat{\mathbf{Z}}_i \mathbf{p}_i^k + \hat{\mathbf{Z}}_i \mathbf{R}_i \hat{\mathbf{x}}^k \\ &= \hat{\mathbf{Z}}_i \mathbf{R}_i \mathbf{y} - \hat{\mathbf{Z}}_i \mathbf{p}_i^k + \hat{\mathbf{Z}}_i \mathbf{R}_i \hat{\mathbf{x}}^k \\ &= \hat{\mathbf{Z}}_i \mathbf{R}_i (\mathbf{y} + \hat{\mathbf{x}}^k) - \hat{\mathbf{p}}_i^k, \end{aligned} \quad (\text{C.5})$$

where we use the idempotent property of  $\hat{\mathbf{Z}}_i$ . Similarly to (2.8) and based on (C.1), the global



**Figure 8.** Illustration of  $\phi(\tau, \rho, \lambda)$ , the eigenvalues of the SOS error’s transition matrix (in absolute value), as a function of  $\tau$ . The dashed (blue), solid (black), and dotted (magenta) lines correspond to  $\phi$  with the arguments  $\lambda_{min}$ ,  $\lambda_i$ , and  $\lambda_{max}$ , respectively. The horizontal black dash-dotted line denotes the condition for convergence, determining the values of  $\tau^{min}$  and  $\tau^{max}$  (red circles). The highlighted line illustrates the function  $\max_i \phi(\tau, \rho, \lambda_i)$ , where  $\tau^*$  obtains its minimal value (green circle).

denoised image is formulated by

$$\begin{aligned}
 \hat{\mathbf{x}}^{k+1} &= \frac{1}{n} \sum_{i=1}^N \mathbf{R}_i^T \hat{\mathbf{p}}_i^{k+1} \\
 &= \frac{1}{n} \sum_{i=1}^N \mathbf{R}_i^T \left( \hat{\mathbf{Z}}_i \mathbf{R}_i \left( \mathbf{y} + \hat{\mathbf{x}}^k \right) - \hat{\mathbf{p}}_i^k \right) \\
 &= \left( \frac{1}{n} \sum_{i=1}^N \mathbf{R}_i^T \hat{\mathbf{Z}}_i \mathbf{R}_i \right) \left( \mathbf{y} + \hat{\mathbf{x}}^k \right) - \frac{1}{n} \sum_{i=1}^N \mathbf{R}_i^T \hat{\mathbf{p}}_i^k \\
 &= \mathbf{W} \left( \mathbf{y} + \hat{\mathbf{x}}^k \right) - \hat{\mathbf{x}}^k.
 \end{aligned}
 \tag{C.6}$$

Thus, the SOS boosting (see (1.4)) and the “sharing the disagreement” algorithms are equivalent for a fixed  $\mathbf{W}$ .

**Appendix D. Seeking the fastest convergence.** We aim to provide conditions for the SOS algorithm to converge in terms of the parameters  $\rho$  and  $\tau$  and in addition get closed-form expression for  $\tau^*$ , the solution of (3.23). The eigenvalues of the SOS error’s transition matrix (in absolute value) are formulated by

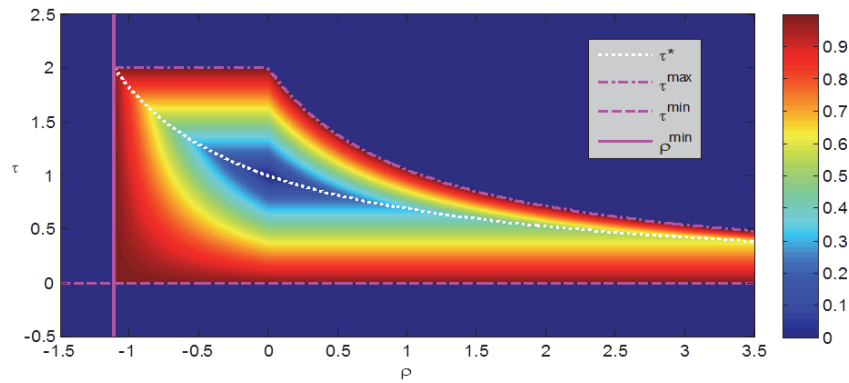
$$\phi(\tau, \rho, \lambda_i) = |\tau(\rho\lambda_i - \rho - 1) + 1|,
 \tag{D.1}$$

where  $\{\lambda_i\}_{i=1}^N$  are the eigenvalues of  $\mathbf{W}$ . In the following analysis we shall assume that  $\lambda_i \leq 1$ . Figure 8 plots  $\phi(\tau, \rho, \lambda_{min})$ ,  $\phi(\tau, \rho, \lambda_i)$ , and  $\phi(\tau, \rho, \lambda_{max})$  as a function of  $\tau$  for  $\rho > 0$ . As can be seen,  $\phi$  has a negative slope for  $0 \leq \tau \leq \frac{1}{\rho+1-\rho\lambda_i}$  and a positive slope for  $\tau > \frac{1}{\rho+1-\rho\lambda_i}$ . All of these are true under the assumption that  $\rho\lambda_i - \rho - 1 < 0$ , which always holds for  $\lambda_i = 1$  and for

$$\rho > \rho^{min} = \min_i \frac{-1}{1 - \lambda_i} \quad \forall \lambda_i \neq 1.
 \tag{D.2}$$

Next, we shall find the valid range of  $\tau \in (\tau^{min}, \tau^{max})$ , satisfying  $\phi < 1$ . Following Figure 8, the minimal  $\tau$  that leads to an intersection with  $\phi = 1$  is

$$\tau^{min} = 0.
 \tag{D.3}$$



**Figure 9.** Demonstration of  $\max_i \phi(\tau, \rho, \lambda)$  for  $\lambda \in [0.1, 1]$ , where cold and warm colors indicate small and large values, respectively. The magenta dash-dotted, dashed, and solid lines plot the boundaries for convergence, corresponding to  $\tau^{max}$ ,  $\tau^{min}$ , and  $\rho^{min}$ , respectively. The white dotted line plots the analytic expression of  $\tau^*$  as a function of  $\rho$ .

Then, by increasing  $\tau$  we get  $\phi < 1$ , until reaching  $\tau^{max}$ —the first  $\tau > \tau^{min}$  that obtains  $\phi = 1$  again. As demonstrated in Figure 8, for  $\rho > 0$  we get  $\tau^{max} = \frac{2}{\rho+1-\rho\lambda_{min}}$ , while for  $\rho < 0$  we get  $\tau^{max} = \frac{2}{\rho+1-\rho\lambda_{max}}$ ; thus

$$(D.4) \quad \tau^{max} = \min \left\{ \frac{2}{\rho+1-\rho\lambda_{min}}, \frac{2}{\rho+1-\rho\lambda_{max}} \right\}.$$

The obtained conditions on  $\rho$  and  $\tau$  are illustrated in Figure 9, which plots the function  $\max_i \phi(\tau, \rho, \lambda_i)$  for  $\lambda \in [0.1, 1]$ , where  $\phi < 1$ . As can be seen,  $\rho^{min}$ ,  $\tau^{min}$ , and  $\tau^{max}$  perfectly bound the valid range for convergence.

We now turn to discuss  $\tau^*$ , the solution of (3.23),

$$\tau^* = \min_{\tau} \max_{1 \leq i \leq N} \phi(\tau, \rho, \lambda_i) \quad \text{s.t.} \quad \forall i \quad \phi(\tau, \rho, \lambda_i) < 1.$$

Following Figure 8, under the obtained conditions on  $\rho$  and  $\tau$ , we seek the minimum value of the highlighted graph (the green circle). As can be seen,  $\tau^*$  is obtained by the following equality:

$$(D.5) \quad \phi(\tau^*, \rho, \lambda_{min}) = \phi(\tau^*, \rho, \lambda_{max}).$$

Based on condition (D.2), we can infer that

$$(D.6) \quad \phi(\tau^*, \rho, \lambda_{min}) = -\tau^*(\rho\lambda_{min} - \rho - 1) - 1$$

and

$$(D.7) \quad \phi(\tau^*, \rho, \lambda_{max}) = \tau^*(\rho\lambda_{max} - \rho - 1) + 1.$$

Substituting (D.6) and (D.7) into (D.5) leads to

$$\tau^* = \frac{2}{2(\rho+1) - \rho(\lambda_{min} + \lambda_{max})}.$$

In addition, by substituting  $\tau^*$  into (D.7) we get

$$\gamma^* = \frac{\rho(\lambda_{\max} - \lambda_{\min})}{2(\rho + 1) - \rho(\lambda_{\min} + \lambda_{\max})}.$$

An illustration that  $\tau^*$  obtains the minimal eigenvalue of the error's transition matrix is shown in Figure 9, as a curve running through all possible values of  $\rho$ . Note that the fastest convergence is obtained for the couple  $\tau = 1$  and  $\rho = 0$  ( $\gamma^* = 0$ ), i.e., applying the original denoising algorithm only once, without any SOS step. However, as we aim to improve the denoising performance by strengthening the underlying signal (setting  $\rho \neq 0$ ), this choice of parameters is meaningless.

## REFERENCES

- [1] M. AHARON, M. ELAD, AND A. BRUCKSTEIN, *The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation*, IEEE Trans. Signal Process., 54 (2006), pp. 4311–4322.
- [2] S. BOUGLEUX, A. ELMOATAZ, AND M. MELKEMI, *Local and nonlocal discrete regularization on weighted graphs for image and mesh processing*, Internat. J. Comput. Vis., 84 (2009), pp. 220–236.
- [3] S. BOYD, N. PARIKH, E. CHU, B. PELEATO, AND J. ECKSTEIN, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Machine Learning, 3 (2011), pp. 1–122.
- [4] L. M. BREGMAN, *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, USSR Comput. Math. Math. Phys., 7 (1967), pp. 200–217.
- [5] A. M. BRUCKSTEIN, D. L. DONOHO, AND M. ELAD, *From sparse solutions of systems of equations to sparse modeling of signals and images*, SIAM Rev., 51 (2009), pp. 34–81.
- [6] A. BUADES, B. COLL, AND J. M. MOREL, *A review of image denoising algorithms, with a new one*, Multiscale Model. Simul., 4 (2005), pp. 490–530.
- [7] A. BUADES, B. COLL, AND J. M. MOREL, *Non-local means denoising*, Image Processing On Line, 1 (2011). [http://www.ipol.im/pub/art/2011/bcm\\_nlm/](http://www.ipol.im/pub/art/2011/bcm_nlm/).
- [8] P. BÜHLMANN AND B. YU, *Boosting with the  $l_2$  loss: Regression and classification*, J. Amer. Statist. Assoc., 98 (2003), pp. 324–339.
- [9] H. C. BURGER, C. J. SCHULER, AND S. HARMELING, *Image denoising: Can plain neural networks compete with BM3D?*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Washington, DC, 2012, pp. 2392–2399.
- [10] M. R. CHAREST, M. ELAD, AND P. MILANFAR, *A general iterative regularization framework for image denoising*, in Proceedings of the IEEE Conference on Information Sciences and Systems, Princeton, NJ, 2006, pp. 452–457.
- [11] P. CHATTERJEE AND P. MILANFAR, *Patch-based near-optimal image denoising*, IEEE Trans. Image Process., 21 (2012), pp. 1635–1649.
- [12] S. S. CHEN, D. L. DONOHO, AND M. A. SAUNDERS, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput., 20 (1998), pp. 33–61.
- [13] K. DABOV, A. FOI, V. KATKOVNIK, AND K. EGIAZARIAN, *Image denoising by sparse 3-D transform-domain collaborative filtering*, IEEE Trans. Image Process., 16 (2007), pp. 2080–2095.
- [14] M. ELAD, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, New York, 2010.
- [15] M. ELAD AND M. AHARON, *Image denoising via sparse and redundant representations over learned dictionaries*, IEEE Trans. Image Process., 15 (2006), pp. 3736–3745.
- [16] A. ELMOATAZ, O. LEZORAY, AND S. BOUGLEUX, *Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing*, IEEE Trans. Image Process., 17 (2008), pp. 1047–1060.

- [17] K. ENGAN, S. O. AASE, AND H. J. HAKON, *Method of optimal directions for frame design*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 5, Phoenix, AZ, 1999, pp. 2443–2446.
- [18] A. GADDE, S. K. NARANG, AND A. ORTEGA, *Bilateral Filter: Graph Spectral Interpretation and Extensions*, arXiv preprint, <http://arxiv.org/abs/1303.2685>, 2013.
- [19] G. GILBOA AND S. OSHER, *Nonlocal linear image regularization and supervised segmentation*, Multiscale Model. Simul., 6 (2007), pp. 595–630.
- [20] G. GILBOA AND S. OSHER, *Nonlocal operators with applications to image processing*, Multiscale Model. Simul., 7 (2008), pp. 1005–1028.
- [21] S. M. HAQUE, G. PAI, AND V. M. GOVINDU, *Symmetric smoothing filters from global consistency constraints*, IEEE Trans. Image Process., 24 (2014), pp. 1536–1548.
- [22] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 2012.
- [23] A. KHERADMAND AND P. MILANFAR, *A general framework for kernel similarity-based image denoising*, in Proceedings of the Global Conference on Signal and Information Processing, Austin, TX, 2013, pp. 415–418.
- [24] A. KHERADMAND AND P. MILANFAR, *A general framework for regularized, similarity-based image restoration*, IEEE Trans. Image Process., 23 (2014), pp. 5136–5151.
- [25] P. A. KNIGHT AND D. RUIZ, *A fast algorithm for matrix balancing*, IMA J. Numer. Anal., 33 (2013), pp. 1029–1047.
- [26] M. LEBRUN, A. BUADES, AND J.-M. MOREL, *Implementation of the “Non-Local Bayes” (NL-Bayes) Image Denoising Algorithm*, Image Processing On Line, 3 (2013), pp. 1–42. <http://www.ipol.im/pub/art/2013/16/article.pdf>.
- [27] M. LEBRUN, M. COLOM, A. BUADES, AND J.-M. MOREL, *Secrets of image denoising cuisine*, Acta Numer., 21 (2012), pp. 475–576.
- [28] A. LEVIN AND B. NADLER, *Natural image denoising: Optimality and inherent bounds*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2011, pp. 2833–2840.
- [29] X. LIU, D. ZHAI, D. ZHAO, G. ZHAI, AND W. GAO, *Progressive image denoising through hybrid graph Laplacian regularization: A unified framework*, IEEE Trans. Image Process., 23 (2014), pp. 1491–1503.
- [30] F. G. MEYER AND X. SHEN, *Perturbation of the eigenvectors of the graph Laplacian: Application to image denoising*, Appl. Comput. Harmon. Anal., 36 (2014), pp. 326–334.
- [31] P. MILANFAR, *A tour of modern image filtering: New insights and methods, both practical and theoretical*, IEEE Signal Process. Mag., 30 (2013), pp. 106–128.
- [32] S. OSHER, M. BURGER, D. GOLDFARB, J. XU, AND W. YIN, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489.
- [33] Y. C. PATI, R. REZAIIFAR, AND P. S. KRISHNAPRASAD, *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*, in Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 1993, pp. 40–44.
- [34] M. PROTTER, M. ELAD, H. TAKEDA, AND P. MILANFAR, *Generalizing the nonlocal-means to super-resolution reconstruction*, IEEE Trans. Image Process., 18 (2009), pp. 36–51.
- [35] I. RAM, M. ELAD, AND I. COHEN, *Image processing using smooth ordering of its patches*, IEEE Trans. Image Process., 22 (2013), pp. 2764–2774.
- [36] Y. ROMANO AND M. ELAD, *Improving K-SVD denoising by post-processing its method-noise*, in Proceedings of the IEEE International Conference on Image Processing, Melbourne, Australia, 2013, pp. 435–439.
- [37] Y. ROMANO AND M. ELAD, *Patch-disagreement as a way to improve K-SVD denoising*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 2015.
- [38] Y. ROMANO, M. PROTTER, AND M. ELAD, *Single image interpolation via adaptive nonlocal sparsity-based modeling*, IEEE Trans. Image Process., 23 (2014), pp. 3085–3098.
- [39] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
- [40] E. SENETA, *Non-negative Matrices and Markov Chains*, 2nd ed., Springer-Verlag, New York, 1981.



- [41] D. I. SHUMAN, S. K. NARANG, P. FROSSARD, A. ORTEGA, AND P. VANDERGHEYNST, *The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains*, IEEE Signal Process. Mag., 30 (2013), pp. 83–98.
- [42] L. N. SMITH AND M. ELAD, *Improving dictionary learning: Multiple dictionary updates and coefficient reuse*, Signal Process. Lett., 20 (2013), pp. 79–82.
- [43] J. SULAM AND M. ELAD, *Expected patch log likelihood with a sparse prior*, in Proceedings of the 10th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (Hong Kong, 2015), Lecture Notes in Comput. Sci. 8932, Springer-Verlag, Berlin, 2015, pp. 99–111.
- [44] A. D. SZLAM, M. MAGGIONI, AND R. R. COIFMAN, *Regularization on graphs with function-adapted diffusion processes*, J. Machine Learning Res., 9 (2008), pp. 1711–1739.
- [45] H. TALEBI AND P. MILANFAR, *Global image denoising*, IEEE Trans. Image Process., 23 (2014), pp. 755–768.
- [46] H. TALEBI, X. ZHU, AND P. MILANFAR, *How to SAIF-ly boost denoising performance*, IEEE Trans. Image Process., 22 (2013), pp. 1470–1485.
- [47] C. TOMASI AND R. MANDUCHI, *Bilateral filtering for gray and color images*, in Proceedings of the IEEE International Conference on Computer Vision, Bombay, India, 1998, pp. 839–846.
- [48] J. A. TROPP AND S. J. WRIGHT, *Computational methods for sparse solution of linear inverse problems*, Proc. IEEE, 98 (2010), pp. 948–958.
- [49] J. W. TUKEY, *Exploratory Data Analysis*, Pearson, Reading, MA, 1977.
- [50] U. VON LUXBURG, *A tutorial on spectral clustering*, Statist. Comput., 17 (2007), pp. 395–416.
- [51] G. YU, G. SAPIRO, AND S. MALLAT, *Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity*, IEEE Trans. Image Process., 21 (2012), pp. 2481–2499.
- [52] D. ZORAN AND Y. WEISS, *Scale invariance and noise in natural images*, in Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 2009, pp. 2209–2216.
- [53] D. ZORAN AND Y. WEISS, *From learning models of natural image patches to whole image restoration*, in Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 479–486.