Multi-Scale Patch-Based Image Restoration

Vardan Papyan and Michael Elad, *Fellow, IEEE*

***Abstract*— Many image restoration algorithms in recent years are based on patch processing. The core idea is to decompose the target image into fully overlapping patches, restore each of them separately, and then merge the results by a plain averaging. This concept has been demonstrated to be highly effective, leading often times to the state-of-the-art results in denoising, inpainting, deblurring, segmentation, and other applications. While the above is indeed effective, this approach has one major flaw: the prior is imposed on intermediate (patch) results, rather than on the final outcome, and this is typically manifested by visual artifacts. The expected patch log likelihood (EPLL) method by Zoran and Weiss was conceived for addressing this very problem. Their algorithm imposes the prior on the patches of the *final image*, which in turn leads to an iterative restoration of diminishing effect. In this paper, we propose to further extend and improve the EPLL by considering a multi-scale prior. Our algorithm imposes the very same prior on different scale patches extracted from the target image. While all the treated patches are of the same size, their footprint in the destination image varies due to subsampling. Our scheme comes to alleviate another shortcoming existing in patch-based restoration algorithms—the fact that a local (patch-based) prior is serving as a model for a global stochastic phenomenon. We motivate the use of the multi-scale EPLL by restricting ourselves to the simple Gaussian case, comparing the aforementioned algorithms and showing a clear advantage to the proposed method. We then demonstrate our algorithm in the context of image denoising, deblurring, and super-resolution, showing an improvement in performance both visually and quantitatively.**

***Index Terms*— Image restoration, expected patch log likelihood (EPLL), Gaussian mixture model, multi-scale, denoising, deblurring, super-resolution.**

I. Introduction

Assume a clean image X is degraded by a linear operator A and an additive white Gaussian noise N of standard deviation σ. Given the measurement $Y = AX + N$, we would like to restore the underlaying image X. To this end, some prior knowledge about the unknown image to be recovered is needed. Due to the curse of dimensionality, proposing a global model for the whole image is often found to be too hard, and especially so if we are dealing with learned models. Thus, many image restoration algorithms in recent years have chosen to address the matter of modeling by adopting patch (or local) priors, e.g., [1]–[9].

Since patches are low dimensional and therefore easier to model, patch priors are readily available. Popular choices include the sparity inspired model, GMM (Gaussian Mixture Model), ICA, an analysis model such as FoE, etc. [2], [10]–[13]. In the context of denoising, for example, once such a prior is set, the image can be denoised by decomposing it into overlapping patches, denoising every patch separately and finally aggregating the results by simple averaging. This straightforward yet effective paradigm has been shown to give excellent results in various inverse problems.

Despite the effectiveness of the above scheme, it has some major known flaws. For example, although every denoised patch is treated well under the chosen patch prior, the averaging of the patches ruins this behavior, as the resulting patches extracted from the aggregated image are no longer likely with respect to the local prior. Put differently, the above problem implies that solving the denoising problem by some local operations that do not share information between them (as indeed patch-based methods often do), is likely to be sub-optimal.

This rationale has led Zoran and Weiss to propose the EPLL (Expected Patch Log Likelihood) algorithm [11]. Their scheme employs a global prior which seeks an image such that every selected patch from it is likely given the local prior. Due to this delicate change from a local to a global prior, the complete denoised image is the unknown in the restoration task, rather than small and independent patches. To practically solve the MAP (maximum a posteriori) problem under this prior, the authors of [11] have used the method of Half Quadratic Splitting [14]. Their algorithm boils down to iteratively applying patch-based denoising with a decreasing noise level after each iteration. As the iterations proceed, the overlapping patches are pushed closer and closer to the local model. The EPLL was originally used with the GMM prior, and more recently extended to a sparsity-based patch model [15], leading to a comparable performance.

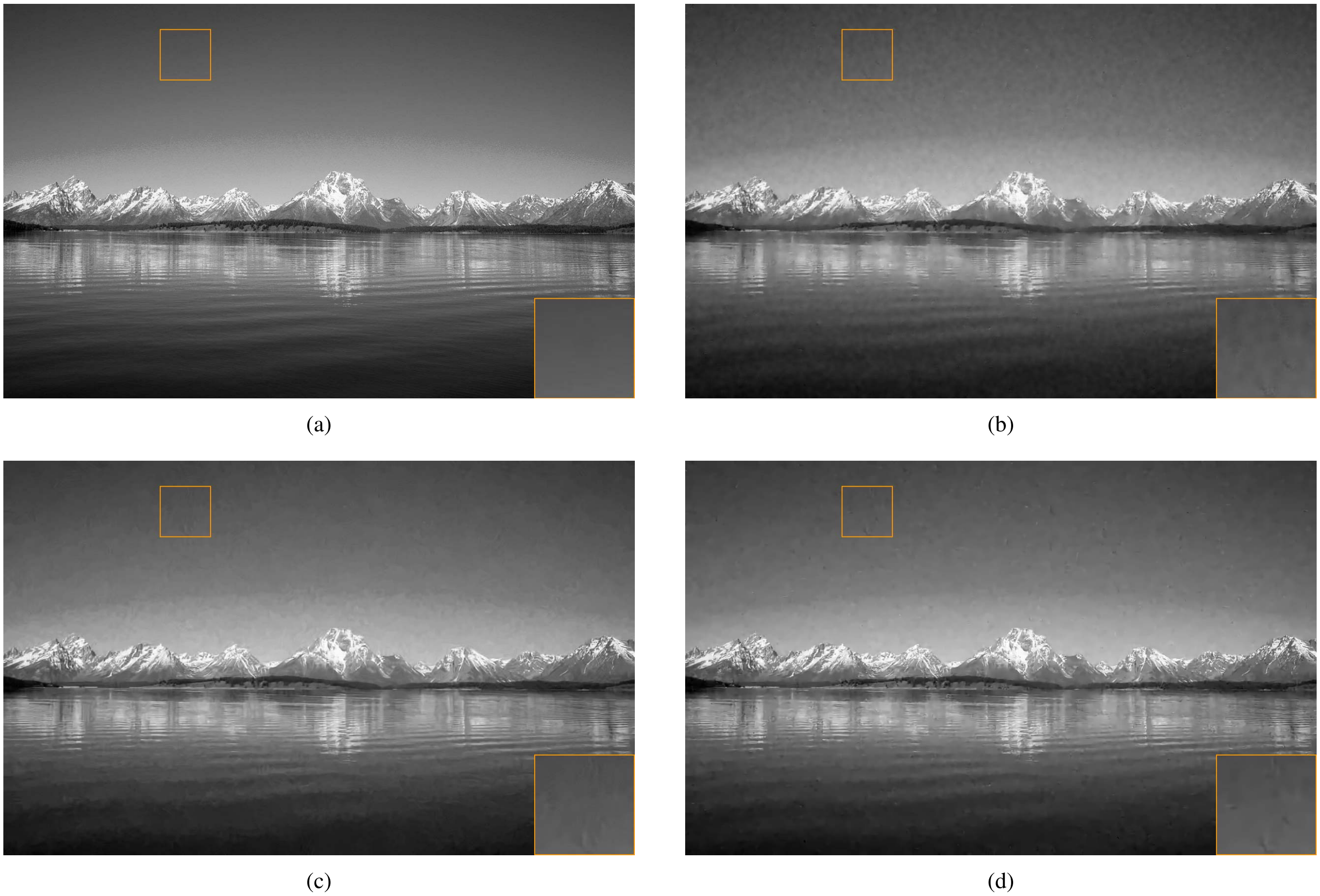
We should stress that, despite the success of the EPLL in improving denoising (and other applications) performance, this approach is only one way of globalizing an image prior, which is defined for local patches. Interestingly, the formation of a global prior from local interactions is also practiced by the line of work of Fields of Experts and papers that employ the analysis sparsity model [10], [12], [13]. Another attempt to bring a global flavor to patch-modeling is found in methods that exploit self-similarity of patches [1], [5], [16]–[21]. These methods tie together the treatment of different patches if they are found to have similar content. This way, far apart patches collaborate in their restoration process, thus leading to non-local processing. Interestingly, even these sophisticated methods fail in fully "globalizing" the processing,

Manuscript received July 19, 2015; revised October 8, 2015; accepted November 4, 2015. Date of publication November 11, 2015; date of current version December 3, 2015. This work was supported by the European Research Council through the European Unions Seventh Framework Program under Grant 320649. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yue M. Lu.

The authors are with the Computer Science Department, Technion–Israel Institute of Technology, Haifa 32000, Israel (e-mail: vardanp91@gmail.com; elad@cs.technion.ac.il).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2499698


(a)

(b)

(c)

(d)

Fig. 1. Denoising of a landscape image. All denoised images exhibit local artifacts (see for example the skies). BM3D, a self-similarity based method, fails as well despite making use of non-local processing. (a) Original Image. (b) K-SVD - PSNR: 32.52. (c) BM3D - PSNR: 33.21. (d) EPLL - PSNR: 33.31.

resulting often times with local artifacts, as can be seen in Figure 1.

Another and perhaps a more profound flaw in patch-based modeling is the enforced locality of the overall model. By working with small sized patches we are oblivious to a larger scale interactions present in the image. Broadly speaking, one can not hope to model a sophisticated stochastic phenomenon such as a whole image by simply looking at its local interactions. This flaw is true regardless of the method employed on the patches - be it plain averaging, a sophisticated fusion of information between patches, or a global expression that ties together local forces. In this work we come to alleviate some of this problem, by considering multi-scale processing.

The above discussion suggests that, on one hand, modeling whole images is prohibitively hard, and on the other hand, considering only local patch modeling might be too restrictive. In this work we propose a multi-scale image treatment, which also leads to a global image prior composed of local pieces. The key idea is that, by using a local model on various scales, we manage to keep the simplicity of a low dimensional model while enjoying the non-locality of bigger regions in the image.

Our algorithm imposes a local low dimensional prior on patches *of different scales*, extracted from the target image. These, so called, scale-patches are all equal sized patches extracted from filtered-and-decimated versions of the target image. If scale invariance is assumed, the model of the scale patches remains unchanged and a single model can be used for all scales. Although all the treated scale patches are of equal size, their footprint in the destination image varies due to sub-sampling. For example, if we consider decimation by factor of two, scale patches extracted from the coarser image are in fact twice as big in the original image grid. Therefore, by constraining those patches to follow the local model, we constrain the lower frequencies of twice as big patches in the original image. This idea could be readily generalized to many filters or non decimation filters, with the exception that if this is the case, the model should be fitted to each filter differently, as one can no longer rely on scale-invariance.

When applicable, the scale invariance property can boost the performance of the methods we are developing in this paper, as we force a unified model to all resolution layers, thus getting a better estimate of the model. This assumption is central in visual data and has drawn a considerable attention along the years [22], [23].

The multi-scale treatment we propose here bares some similarity to denoising through the use of multi-scale dictionary learning. In [24] and [25] an image was denoised by decomposing it into different wavelet bands, denoising every band independently via patch-based K-SVD, and applying inverse wavelet transform to obtain the final reconstructed image. An unavoidable difficulty with this approach is the fact that the high frequency bands of the image are to be denoised, a task which is known to be problematic due to the low signal to noise ratio in such frequencies. In our work we avoid this problem altogether by considering low frequency bands which are known to have a high signal to noise ratio, and even more so, as we decrease the resolution, the SNR actually improves. Moreover, the approach we propose fuses all bands simultaneously by forming one unified penalty function that all patches from all scales must obey.

To motivate the extension of the multi-scale EPLL, we consider a "toy problem" where the signal is one dimensional and sampled from a Gaussian multivariate distribution. In this case, all aforementioned methods are reduced to closed-form solutions. Once an expression is obtained for each method, we compare the mean-squared-error (MSE) of the different algorithms and demonstrate how the multi-scale EPLL narrows the local-to-global gap created by working with patches. We further show that the proposed multi-scale treatment has an interesting interpretation in terms of approximation of the global Covariance matrix.

Returning to the realm of image processing and aiming at boosting leading methods, we demonstrate the proposed multi-scale EPLL for the task of solving three different inverse problems: denoising, deblurring and super-resolution. Comparisons to EPLL show clear advantage to the proposed paradigm across all tasks, and especially so when the problem is severely ill-posed. In this context we present two key ingredients that help achieve the above results: (i) As mentioned earlier, we assume that the patch-model remains intact after the scale-down and decimation. This scale-invariance property is true only for a carefully chosen filter, which we derive. (ii) The use of the multi-scale EPLL requires several parameters to be set up. One possible way to tune those is using the generalization of Stein's Unbiased Estimator (SURE) for non-linear denoisers [26]. Another approach is to optimize those parameters on a set of images. We demonstrate both methods in our work.

This paper is organized as follows: Section 2 briefly reviews the EPLL framework, Section 3 analyzes the various global and local algorithms in the Gaussian case, Section 4 introduces the proposed multi-scale EPLL for image processing, along with a discussion on automatic parameter setup and scale-invariance. In Section 5 we present experimental results and Section 6 concludes this paper.

II. Expected Patch Log Likelihood

In this section we briefly review the EPLL, as we will be relying on this method and extending it hereafter. Given a corrupted image $Y = AX + N$, we would like to restore X by solving the maximum a posteriori (MAP) problem

$$\begin{aligned}\max_X P(X|Y) &= \max_X P(Y|X)P(X) \\ &= \min_X -\log P(Y|X) - \log P(X). \qquad (1)\end{aligned}$$

To this end, a global prior $P(X)$ for the ideal image is needed. One such prior is the expected patch log likelihood (EPLL), which forms the prior by accumulating local (patch) ingredients,

$$EPLL(X) = \log P(X) = \sum_i \log P(\boldsymbol{R_i} X), \qquad (2)$$

where we have defined $\boldsymbol{R_i}$ to be an operator which extracts the i'th patch from the image. By placing the corruption model and the EPLL into (1) we obtain

$$\min_X \frac{\lambda}{2}\|\boldsymbol{A}X - Y\|_2^2 - \sum_i \log P(\boldsymbol{R_i} X), \qquad (3)$$

where $\lambda = \frac{p}{\sigma^2}$ and p is the patch-size. This problem can be solved using Half Quadratic Splitting [14] by introducing a set of auxiliary variables z_i that ideally should be equal to $\boldsymbol{R_i} X$. This eases the optimization and changes it into

$$\min_{X,\{z_i\}_i} \frac{\lambda}{2}\|\boldsymbol{A}X - Y\|_2^2 + \sum_i \left(\frac{\beta}{2}\|\boldsymbol{R_i} X - z_i\|_2^2 - \log P(z_i)\right). \qquad (4)$$

When $\beta \to \infty$ we obtain $\boldsymbol{R_i} X = z_i$ and as a result we converge to the solution of (3). This is the approach practiced by Zoran and Weiss in [11]. Interestingly, one could also adopt an ADMM path [27] for the numerical solution of (3), which bares some similarity to the above half-splitting method, but is free from the need to grow β to infinity. We shall not dwell on this option here, as our tests suggest that it is of comparable quality.

In practice, the problem is solved by iterating through a finite set of increasing β and for each fixed β reducing the objective using block coordinate descent. Assuming X is fixed, we minimize Equation (4) with respect to z_i. As a result, the z_i are updated by solving a local MAP problem

$$\min_{z_i} \frac{\beta}{2}\|\boldsymbol{R_i} X - z_i\|_2^2 - \log P(z_i). \qquad (5)$$

Assuming z_i are fixed, we minimize Equation (4) with respect to X. As a result, X is updated by minimizing a pure quadratic problem, resulting with the closed-form expression[1]

$$X = \left(\lambda \boldsymbol{A}^T \boldsymbol{A} + \beta \sum_i \boldsymbol{R_i}^T \boldsymbol{R_i}\right)^{-1} \left(\lambda \boldsymbol{A}^T Y + \beta \sum_i \boldsymbol{R_i}^T z_i\right). \qquad (6)$$

The special case of applying the above for one iteration only and initializing with $X = Y$ (i.e., fixing $X = Y$, finding all z_i and then updating X once) is a simpler algorithm that essentially boils down to denoising of the degraded patches in Y, followed by a patch-averaging combined with an inversion of the degradation operator $\boldsymbol{A}$. In the context of a sparsity-inspired model, the K-SVD denoising algorithm is such an approach [2], and the follow-up work reported in [15]

[1] Hereinafter, plain patch averaging is used; however, an alternative approach could be suggested. Assume the EPLL prior had a different weight for the log-likelihood of each patch (for example, using a robust measure of proximity instead of L_2 for each patch). As a result, in the aggregation process we would obtain a weighted patch averaging.

suggests a way to proceed the iterations in order to get to a better denoising performance. The key problem in EPLL (with any prior) is the choice of β and how to grow it during the iterations in order to guarantee an improved outcome. The work in [11] chose to simply set β to a fixed set of values, while [15] estimated it from the temporal image X.

For completeness of our discussion here, and due to the later use of the expression shown here in the next section, we briefly mention that the local prior used in conjunction with the EPLL in [11] is a GMM model defined as

$$P(z) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(z|0, \Sigma_k). \tag{7}$$

The Covariance and the weight for the k-th Gaussian are denoted by Σ_k and π_k, respectively, and the means of all Gaussians are set to zero. Since no closed-form solution exists for the MAP problem in (5) under this prior, an approximation is needed to update z_i given X. First, the most likely Gaussian is chosen,

$$\begin{aligned} &\max_k P(k|\boldsymbol{R_i} X) \\ &\quad = \max_k P(\boldsymbol{R_i} X|k)P(k) \\ &\quad = \min_k - \log P(\boldsymbol{R_i} X|k) - \log P(k) \\ &\quad = \min_k \frac{1}{2} \log\left(\left|\Sigma_k + \frac{1}{\beta} I\right|\right) \\ &\qquad + \frac{1}{2}(\boldsymbol{R_i} X)^T \left(\Sigma_k + \frac{1}{\beta} I\right)^{-1} (\boldsymbol{R_i} X) - \log(\pi_k). \end{aligned} \tag{8}$$

Once the best component denoted by $\hat{k}$ is selected, the restored patch is obtained by a classic Wiener filter

$$z_i = \Sigma_{\hat{k}} \left(\Sigma_{\hat{k}} + \frac{1}{\beta} I\right)^{-1} \boldsymbol{R_i} X. \tag{9}$$

III. Case Study - A Gaussian Signal

Before venturing into multi-scale priors for general content images, let us review currently used methods by considering a “toy problem”, so as to gain an insight to the various restoration approaches that can be taken. The order in which these methods are presented shows their evolution. Assume a signal X sampled from a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ is contaminated by white additive Gaussian noise sampled from $\mathcal{N}(0, \sigma^2 I)$, and our task is to denoise it. Below we present several possible approaches to clean up the noise.

A. MMSE - Minimum Mean Squared Error

The MMSE estimator minimizes the mean squared error and is therefore the best estimator one could hope for, if PSNR is our measure of quality. In the Gaussian case, this has a simple closed-form solution in the form of the Wiener filter, which we have seen above in Equation (9):

$$\hat{X} = \Sigma \left(\Sigma + \sigma^2 I\right)^{-1} (Y - \mu) + \mu. \tag{10}$$

On the down side, this estimator must have access to the distribution model of the whole signal, implying knowledge of the global mean and Covariance, an assumption which in our terminology is considered as unreasonable. All the next methods will aim to achieve the denoising performance of this method while adopting patch-based and thus simpler modeling. Another shortcoming of this method refers to the computational part, there is a need to invert a very big matrix. As we shall see next, the patch-based methods enjoy the ability of avoiding such “global inversion”.

B. Non Overlapping or Averaging Overlapping Patches

Assume that we have at our disposal the distribution of the small patches extracted from the complete signal. In our case, this distribution is obtained as marginals of the global one, and therefore the patches are Gaussian distributed as well with distribution $\mathcal{N}(\boldsymbol{R_i}\mu, \boldsymbol{R_i}\Sigma\boldsymbol{R_i}^T)$. A special case would be when the distributions of all the patches are identical. This case is of interest since in image processing this is a widely used assumption (with obviously much more complex and rich local priors). In the Gaussian case, such a situation is encountered when the global Covariance is circulant, and the mean vector is constant.

The simplest idea would be to divide the whole signal into a set of patches, and denoise every patch independently according to its distribution using a local Wiener filter

$$\boldsymbol{R_i}\Sigma\boldsymbol{R_i}^T \left(\boldsymbol{R_i}\Sigma\boldsymbol{R_i}^T + \sigma^2 I\right)^{-1} (\boldsymbol{R_i} Y - \boldsymbol{R_i}\mu) + \boldsymbol{R_i}\mu. \tag{11}$$

This set of patches can be either non-overlapping or overlapping. Once computed, the final signal can be constructed from the denoised patches by simply aggregating the results

$$\begin{aligned} \hat{X} = \mu + \frac{1}{p} \sum_i \boldsymbol{R_i}^T \left(\boldsymbol{R_i}\Sigma\boldsymbol{R_i}^T \left(\boldsymbol{R_i}\Sigma\boldsymbol{R_i}^T + \sigma^2 I\right)^{-1}\right) \boldsymbol{R_i} \\ \times (Y - \mu), \end{aligned} \tag{12}$$

where we have defined p as the number of contributions to each index, which is one in the case of non overlapping patches and it is the patch-size in the case of overlapping patches. The non-overlapping case creates artifacts on the boarders of the denoised patches and therefore is inferior to the overlapping option. Note that the above formula suggests a plain averaging of denoised patches, in the spirit of the discussion in the previous section.

This approach seems to be very promising and indeed was used by several algorithms [1]–[3]. However, a major flaw in this approach is the fact that while each denoised patch is treated well under the given local distribution, the averaging destroys this behavior, resulting with patches that are less likely under the very same local distribution.

C. Expected Log Likelihood (EPLL)

In the previous approach all patches are denoised independently and only once. As we have seen, the EPLL method defines the complete denoised signal as the unknown, while operating locally on patches. The key is to enforce the local (patch-based) model on the final outcome, rather than the intermediate patches that compose it. In general, the objective of the EPLL is

$$\min_X \frac{\lambda}{2}\|X-Y\|_2^2 - \sum_i \log P(\boldsymbol{R_i}X). \tag{13}$$

In the Gaussian case we have a specific probability function for all the patches, and the objective becomes

$$\min_X \frac{\lambda}{2}\|X-Y\|_2^2 + \sum_i \frac{1}{2}(\boldsymbol{R_i}X-\boldsymbol{R_i}\mu)^T \times(\boldsymbol{R_i}\Sigma\boldsymbol{R_i}^T)^{-1}(\boldsymbol{R_i}X-\boldsymbol{R_i}\mu). \tag{14}$$

This function is quadratic and therefore a closed-form solution exists

$$\hat{X} = \left(\lambda I + \sum_i \boldsymbol{R_i}^T(\boldsymbol{R_i}\Sigma\boldsymbol{R_i}^T)^{-1}\boldsymbol{R_i}\right)^{-1}\lambda(Y-\mu)+\mu. \tag{15}$$

We should note that we present (15) only for the purpose of analyzing the EPLL algorithm. In practice, constructing the matrix in (15) and inverting it would be as demanding as the global (MMSE) approach. As the whole purpose of the above discussion is patch-based processing, the proper way to solve (14) is by an iterative process which is composed of two steps, as described in Section 2, relying on half-splitting or ADMM: First, we denoise each patch separately, just as done in Equation (11) and then we aggregate these results to get the full signal using the relation (12).

D. Multi-Scale EPLL

The global MMSE method has the upper hand over any other local-processing method, as it has access to the whole distribution information. In contrast, the local methods, as wise as they may be, get only a partial picture of this distribution. This is evident, for example, when considering the content of the Covariance matrix, far away from the main diagonal. These entries are not seen by the patch-based methods, and thus one cannot expect a global MMSE result from these approximation algorithms. In order to bridge the gap between the sub-optimal estimators and the MMSE one, it is essential to use non-local statistics.

The EPLL prior seeks a denoised signal such that every patch extracted from it is likely given the patch-model. We propose to generalize this by demanding the same property on scaled-down portions of the denoised signal. The multi-scale EPLL prior[2] is defined as[3]

$$MSEPLL(X) = w_1\sum_i \log P_1(\boldsymbol{R_i}X) + w_2\sum_i \log P_2\left(\boldsymbol{\hat{R}_i}\boldsymbol{S}X\right), \tag{16}$$

where the operator $\boldsymbol{S}=\boldsymbol{DH}$ applies a low-pass filter $\boldsymbol{H}$ followed by down-sampling $\boldsymbol{D}$, the operator $\boldsymbol{\hat{R}_i}$ extracts the i'th patch from the decimated signal $\boldsymbol{S}X$, and the weights w_1 and w_2 represent the importance of the different scales. Note that in general the patches obtained after blur and decimation may have a different local model, which explains why we denote the two priors as P_1 and P_2. The MAP objective for our denoising task changes accordingly to

$$\min_X \frac{\lambda}{2}\|X-Y\|_2^2 - w_1\sum_i \log P_1(\boldsymbol{R_i}X) - w_2\sum_i \log P_2\left(\boldsymbol{\hat{R}_i}\boldsymbol{S}X\right). \tag{17}$$

In the Gaussian case this objective becomes

$$\begin{aligned}\min_X\ & \frac{\lambda}{2}\|X-Y\|_2^2 \\ &+ w_1\sum_i \frac{1}{2}(\boldsymbol{R_i}X-\boldsymbol{R_i}\mu)^T(\boldsymbol{R_i}\Sigma\boldsymbol{R_i}^T)^{-1}(\boldsymbol{R_i}X-\boldsymbol{R_i}\mu) \\ &+ w_2\sum_i \frac{1}{2}\left(\boldsymbol{\hat{R}_i}\boldsymbol{S}X-\boldsymbol{\hat{R}_i}\boldsymbol{S}\mu\right)^T\left(\boldsymbol{\hat{R}_i}\boldsymbol{S}\Sigma\boldsymbol{S}^T\boldsymbol{\hat{R}_i}^T\right)^{-1} \\ &\times\left(\boldsymbol{\hat{R}_i}\boldsymbol{S}X-\boldsymbol{\hat{R}_i}\boldsymbol{S}\mu\right).\end{aligned} \tag{18}$$

Once again, the function is quadratic and therefore a closed-form solution exists

$$\begin{aligned}\hat{X} = &\left(\lambda I + w_1\sum_i \boldsymbol{R_i}^T(\boldsymbol{R_i}\Sigma\boldsymbol{R_i}^T)^{-1}\boldsymbol{R_i}\right. \\ &\left.+w_2\sum_i \boldsymbol{S}^T\boldsymbol{\hat{R}_i}^T\left(\boldsymbol{\hat{R}_i}\boldsymbol{S}\Sigma\boldsymbol{S}^T\boldsymbol{\hat{R}_i}^T\right)^{-1}\boldsymbol{\hat{R}_i}\boldsymbol{S}\right)^{-1} \\ &\times\lambda(Y-\mu)+\mu.\end{aligned} \tag{19}$$

Again, this expression represents the closed-form solution for our problem, but in practice, it can be obtained by an iterative solver that does not invert the global-sized matrix.

The expression $\sum_i \boldsymbol{R_i}^T(\boldsymbol{R_i}\Sigma\boldsymbol{R_i}^T)^{-1}\boldsymbol{R_i}$ found in Equation (15) is an approximation of the global inverse Covariance. In fact, if it were equal to it, we would obtain the MMSE result presented in Equation (10). In Equation (19) we obtain a more sophisticated sum which attempts to better approximate the global inverse Covariance.

For example, if we are dealing with twice longer patches extracted from the signal, those will be filtered and decimated by factor of two, thus having the same core patch-size. Our modified model imposes the local patch model on all these patches, regardless of their original scale. This way, the processing remains local but the footprint on the signal is of wider and wider extent, as we move up with the scale. Intuitively, while the EPLL is allowed to look at the global distribution through a pin-hole which is the size of a patch, the multi-scale EPLL looks through the same sized pin-hole but from different distances, thus seeing a wider, but less detailed, picture. This idea can be easily generalized to the application of several filters and down-sampling factors.

In Figure 2 we illustrate the difference between the original EPLL and the multi-scale one. We generate a banded Covariance matrix and present the different local pieces both algorithms use in their operations. The original EPLL utilizes Covariances of local patches that are $\boldsymbol{R_i}\Sigma\boldsymbol{R_i}^T$. These are

[2] For simplicity, we refer here to the case of having two scales only, but the same concept can be applied to a complete pyramid of scales.

[3] The multi-scale EPLL prior degenerates to the original EPLL when a single scale is used with a weight equal to one.

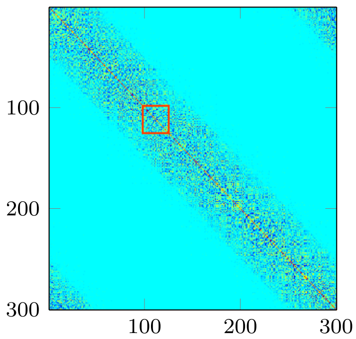


Fig. 2. Left - the full Covariance Σ and a local Covariance $\boldsymbol{R_i} \Sigma \boldsymbol{R}_i^T$ extracted from it. Right - the whole decimated Covariance $\boldsymbol{S} \Sigma \boldsymbol{S}^T$ and a local Covariance $\hat{\boldsymbol{R}}_i \boldsymbol{S} \Sigma \boldsymbol{S}^T \hat{\boldsymbol{R}}_i^T$ extracted from it. Both local Covariances are of equal size. The first scale is capable of only seeing part of the band while the second scale manages to see the whole band at a lower resolution due to sub-sampling.

obtained by extracting the rows and columns corresponding to the patch indices from the Covariance Σ. The multi-scale EPLL, on the other hand, utilizes also Covariances of patches extracted from decimated signals. Those are equal to $\hat{\boldsymbol{R}}_i \boldsymbol{S} \Sigma \boldsymbol{S}^T \hat{\boldsymbol{R}}_i^T$ and are obtained by extracting from the decimated Covariance $\boldsymbol{S} \Sigma \boldsymbol{S}^T$ the proper rows and columns.

In Equation (16) several down-sampling grids can be used after filtering the signal. For example, when down-sampling by a factor of two, one can choose all even or all odd indices. To avoid artifacts created by restricting ourselves to a single grid, we use of all of them. Thus, the prior changes into

$$
\begin{aligned}
MSEPLL(X) &= w_1 \sum_i \log P_1(\boldsymbol{R_i} X) \\
&\quad + w_2 \sum_i \sum_j \log P_2\left(\hat{\boldsymbol{R}}_i \boldsymbol{D}_j \boldsymbol{H} X\right), \qquad (20)
\end{aligned}
$$

where $\boldsymbol{D_j}$ corresponds to the j-th down-sampling grid. For simplicity of formulas, we omit the sum over all patterns in later expressions, but in practice it is essential to use them all.

E. Performace Evaluation

All the methods presented above are of the form

$$
\hat{X} = \boldsymbol{W}(Y - \mu) + \mu, \qquad (21)
$$

where $\boldsymbol{W}$ is some denoising operator. Given the operator $\boldsymbol{W}$, the MSE has a simple expression:

$$
tr\left((\boldsymbol{W} - I)\Sigma(\boldsymbol{W} - I)^T + \sigma^2 \boldsymbol{W}\boldsymbol{W}^T\right). \qquad (22)
$$

We now turn to present a synthetic example that is used to compare all methods presented in this section. A random circulant and banded Covariance matrix and a random constant mean vector were generated to simulate a global model. The length of the signal was 1000, the width of the band of the Covariance was 75 (i.e., there are 75 non-zero main diagonals) and the patch size was set to 25. Since the marginal Covariance of every patch is equal, it is possible to learn the patch-model from a single signal, which explains the appeal of local methods. In Figure 3 we compare the root mean squared error (RMSE) of the different methods. The MSE for each method was calculated using Equation (22). In Figure 4 we exhibit the various denoising operators $\boldsymbol{W}$ obtained from the different methods. For the proposed multi-scale EPLL method, plain sub-sampling was used, as we need not use the scale-invariance assumption.

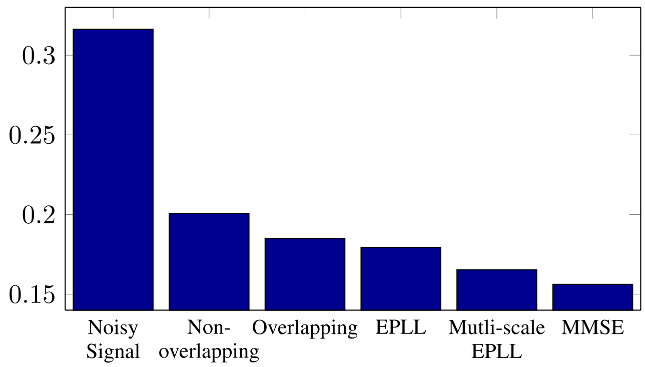
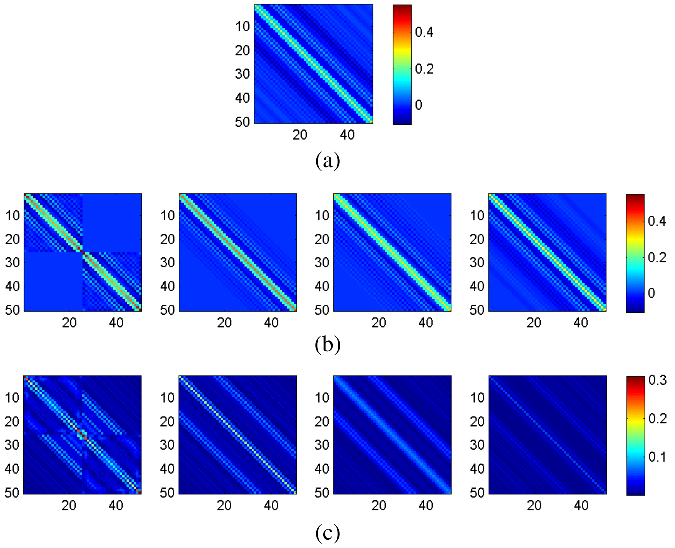


Fig. 3. RMSE for the different methods.



(a)

(b)

(c)

Fig. 4. The filters from left to right: non overlapping patches, overlapping patches, EPLL, multi-scale EPLL. As can be seen, as the methods evolve and improve, the denoising operator becomes wider and more similar to the ultimate MMSE. (a) Zoom-in on the MMSE filter of formula (10). (b) Zoom-in on the filters of the formulas (12),(15) and (19). (c) Zoom-in on the absolute error with respect to the MMSE estimator.

IV. Multi-Scale EPLL

A. The General Scheme

Motivated by the success of the multi-scale EPLL in the Gaussian case, we proceed by moving to inverse problems handling natural images. The objective of the multi-scale EPLL for a general degradation operator $\boldsymbol{A}$ is

$$
\begin{aligned}
\min_X \frac{\lambda}{2} \|\boldsymbol{A}X - Y\|_2^2 &- w_1 \sum_i \log P(\boldsymbol{R_i} X) \\
&- w_2 \sum_i \log P\left(\hat{\boldsymbol{R}}_i \boldsymbol{S} X\right). \qquad (23)
\end{aligned}
$$

To optimize this function, we adopt a similar approach to that of [11], namely, we use Half Quadratic Splitting. We add auxiliary variables z_i and $\hat{z}_i$ which correspond to patches extracted from the original scale and from the added scale. As a result, we obtain the following objective

$$\begin{aligned}\min_X \frac{\lambda}{2}\|\boldsymbol{A}X - Y\|_2^2 & \\ + w_1 \sum_i \min_{z_i} \left\{ \frac{\beta}{2}\|\boldsymbol{R_i}X - z_i\|_2^2 - \log P(z_i) \right\} & \\ + w_2 \sum_i \min_{\hat{z}_i} \left\{ \frac{\hat{\beta}}{2}\|\hat{\boldsymbol{R}}_i \boldsymbol{S}X - \hat{z}_i\|_2^2 - \log P(\hat{z}_i) \right\}. & \quad (24)\end{aligned}$$

The parameter β is proportional to the disagreement between the local model and the reconstructed image X, while the parameter $\hat{\beta}$ does the same for the decimated version of the reconstructed image $\boldsymbol{S}X$. As $\beta \to \infty$ and $\hat{\beta} \to \infty$ we converge to the solution of (23). Thus, we obtain a reconstruction in which every patch extracted from it is likely given the local model, and similarly, every patch extracted from a decimated version of it is likely as well.

Although we have added an extra parameter $\hat{\beta}$, the following analysis suggests that it can be set as being equal to β up to a constant, and therefore only one parameter needs to be tuned. Denote the reconstructed image after several iterations of the proposed algorithm as $\hat{X}$ and recall that we denote X as the clean image. We assume that the expectation of $\left(\hat{X} - X\right)\left(\hat{X} - X\right)^T$ is equal to $\frac{1}{\beta}I$, where I is the identity matrix. This assumption, which has been used in the original EPLL work, is reasonable since the parameter $\frac{1}{\beta}$ is an estimation of the current noise level in the solution. Similarly, $\frac{1}{\hat{\beta}}$ is an approximation for the noise level of the added scale, thus

$$\frac{1}{\hat{\beta}} = \frac{1}{p}\|\hat{\boldsymbol{R}}_i \boldsymbol{S}\hat{X} - \hat{\boldsymbol{R}}_i \boldsymbol{S}X\|_2^2. \quad (25)$$

Where we have normalized the expression by dividing by p, the number of pixels in the patch, so as to get a pixel-wise measure. This expression can be further simplified to

$$\begin{aligned}\frac{1}{\hat{\beta}} &= \frac{1}{p}\left(\hat{\boldsymbol{R}}_i \boldsymbol{S}\hat{X} - \hat{\boldsymbol{R}}_i \boldsymbol{S}X\right)^T \left(\hat{\boldsymbol{R}}_i \boldsymbol{S}\hat{X} - \hat{\boldsymbol{R}}_i \boldsymbol{S}X\right) \\ &= \frac{1}{p}tr\left(\left(\hat{\boldsymbol{R}}_i \boldsymbol{S}\hat{X} - \hat{\boldsymbol{R}}_i \boldsymbol{S}X\right)^T \left(\hat{\boldsymbol{R}}_i \boldsymbol{S}\hat{X} - \hat{\boldsymbol{R}}_i \boldsymbol{S}X\right)\right) \\ &= \frac{1}{p}tr\left(\left(\hat{X} - X\right)^T \left(\hat{\boldsymbol{R}}_i \boldsymbol{S}\right)^T \left(\hat{\boldsymbol{R}}_i \boldsymbol{S}\right) \left(\hat{X} - X\right)\right) \\ &= \frac{1}{p}tr\left(\left(\hat{X} - X\right)\left(\hat{X} - X\right)^T \left(\hat{\boldsymbol{R}}_i \boldsymbol{S}\right)^T \left(\hat{\boldsymbol{R}}_i \boldsymbol{S}\right)\right). \quad (26)\end{aligned}$$

Using the assumption $\left(\hat{X} - X\right)\left(\hat{X} - X\right)^T = \frac{1}{\beta}I$, presented above, we obtain

$$\begin{aligned}\frac{1}{\hat{\beta}} &= \frac{1}{\beta}\frac{1}{p}tr\left(\left(\hat{\boldsymbol{R}}_i \boldsymbol{S}\right)^T \left(\hat{\boldsymbol{R}}_i \boldsymbol{S}\right)\right) \\ &= \frac{1}{\beta}\frac{1}{p}tr\left(\hat{\boldsymbol{R}}_i \boldsymbol{S}\boldsymbol{S}^T \hat{\boldsymbol{R}}_i{}^T\right) \\ &= \frac{1}{\beta}\frac{1}{p}tr\left(\hat{\boldsymbol{R}}_i \boldsymbol{D}\boldsymbol{H}\boldsymbol{H}^T \boldsymbol{D}^T \hat{\boldsymbol{R}}_i{}^T\right). \quad (27)\end{aligned}$$

The matrix $\boldsymbol{R_i}\boldsymbol{D}\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{D}^T\boldsymbol{R}_i^T$ is a p by p matrix with diagonal entries equal to the filter's squared norm. We conclude that $\hat{\beta}$ is equal to β divided by the squared-norm of the filter.[4]

For a given β, the objective can be optimized using block-coordinate descent. First, we assume the reconstructed image X is fixed and we update the auxiliary variables z_i and $\hat{z}_i$. Afterwards we assume the auxiliary patches are fixed and use those to update the reconstructed image. To compute z_i we solve a MAP problem on patches from the original scale

$$\min_{z_i} \frac{\beta}{2}\|\boldsymbol{R_i}X - z_i\|_2^2 - \log P(z_i), \quad (28)$$

and to update $\hat{z}_i$ we solve another MAP problem, this time on patches from the added scale,

$$\min_{\hat{z}_i} \frac{\hat{\beta}}{2}\|\hat{\boldsymbol{R}}_i \boldsymbol{S}X - \hat{z}_i\|_2^2 - \log P(\hat{z}_i). \quad (29)$$

The local prior used for both scales is the GMM model, as defined in (7). When solving the local MAP problem under this prior we use Equations (8) and (9). Given z_i and $\hat{z}_i$, X is updated by solving a simple quadratic problem

$$\begin{aligned}X = &\left(\lambda\boldsymbol{A}^T\boldsymbol{A} + w_1\beta\sum_i \boldsymbol{R}_i^T\boldsymbol{R_i} + w_2\hat{\beta}\sum_i \boldsymbol{S}^T\hat{\boldsymbol{R}}_i{}^T\hat{\boldsymbol{R}}_i\boldsymbol{S}\right)^{-1} \\ &\times\left(\lambda\boldsymbol{A}^T Y + w_1\beta\sum_i \boldsymbol{R}_i^T z_i + w_2\hat{\beta}\sum_i \boldsymbol{S}^T\hat{\boldsymbol{R}}_i{}^T\hat{z}_i\right). \quad (30)\end{aligned}$$

This can be easily computed using Conjugate Gradient without explicitly computing the matrix inverse. Clearly, this algorithm can be easily generalized to more than one scale.

The observant reader might wonder about the split of the minimization operations in this expression. The overall expression is minimized w.r.t. X, but when it comes to the optimization w.r.t. z_i and $\hat{z_i}$, these are pushed inside the overall objective function. This choice of splitting is deliberate and important, as it comes to allow for negative values in the parameters w_1 and/or w_2, as indeed happens in practice. One holistic minimization w.r.t to all the variables would have turned the local MAP stage into a maximization if the weights are negative, which is clearly working against the very idea of half-quadratic splitting. By this split, regardless of the sign of w_1 and w_2, the above optimization scheme remains correct, as long as the optimization over X, when the auxiliary patches are fixed, remains convex. In all our experiments we verify this is indeed the case.

B. Stein's Unbiased Risk Estimate (SURE)

As evident from Equation (30), and in the context of $\boldsymbol{A} = \boldsymbol{I}$ (denoising), at each iteration of the proposed algorithm several denoised images which originate from the different scales are combined by a weighted sum in order to create a new estimate. As a result, a set of weights that will dictate the contribution of

[4] Notice that we have omitted the index i from $\hat{\beta}$ throughout our derivation due to the fact that the final expression does not depend on i.

each scale must be set. It is possible to use the Stein Unbiased Risk Estimator (SURE) to find the optimal weights per image rather than per dataset. SURE offers a way to estimate the expected error without using the true original image [28]. It is given by

$$MSE = \|h(Y,\theta)\|_2^2 - 2h(Y,\theta)^T Y + 2\sigma^2 \nabla \cdot h(Y,\theta), \quad (31)$$

where $h(Y,\theta)$ represents the denoising of the noisy image Y and θ is a parameter-vector the algorithm depends on. Computing the divergence $\nabla \cdot h(Y,\theta)$ is challenging due to the non-linearity incorporated in the multi-scale EPLL by the choice of the Gaussians from the GMM. In [26], it was suggested to circumvent the calculation of the divergence by using a numerical approximation. By drawing a single zero-mean unit variance i.i.d. random vector B, the divergence can be estimated by

$$\nabla \cdot h(Y,\theta) = \frac{1}{\epsilon} B^T \left(h\left(Y + \epsilon B, \theta\right) - h\left(Y, \theta\right)\right), \quad (32)$$

where ϵ is a small constant. While the above can be averaged over several randomly generated vectors B, we find that one is enough. In the results section we shall demonstrate this approach.

Notice that the SURE estimator can set some of the weights of the added scales to be zero. This, in turn, means that these layers are not needed in the final reconstruction. In other words, we can use SURE to find the optimal number of layers to be used by the multi-scale EPLL.

C. Scale Invariance

A desired property when working with different scales of an image is *scale-invariance*. Under such an assumption, all local models of all scales are equal and we need not train a model per each scale separately. In the proposed multi-scale EPLL, a filter is applied before the down-sampling operation, and it can be learned/tuned as to obtain such a property. Similar to the original EPLL, we assume that the local model for the original scale is a GMM

$$P(z) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(z|0, \Sigma_k). \quad (33)$$

We would like to tune the filters such that the additional scales can be modeled by the very same prior.

Given the filter $\boldsymbol{H}$ and the down-sampling pattern $\boldsymbol{D}$, we extract from the image X the scaled-down patches $\hat{\boldsymbol{R}}_{\boldsymbol{i}}\boldsymbol{DH}X$. Our task is to estimate the similarity between the empirical distribution defined by these patches and the local distribution $P(z)$. We begin by calculating the probability for each of the patches to be in every one of the K Gaussians, thus giving a soft assignment of the i-th patch to belong to the k-th Gaussian. This membership probability, which is denoted as $T_{i,k}$, is given by

$$T_{i,k} = \frac{\pi_k \mathcal{N}\left(\hat{\boldsymbol{R}}_{\boldsymbol{i}}\boldsymbol{DH}X|0, \Sigma_k\right)}{\sum_j \pi_j \mathcal{N}\left(\hat{\boldsymbol{R}}_{\boldsymbol{i}}\boldsymbol{DH}X|0, \Sigma_j\right)}. \quad (34)$$

We then compute an empirical GMM distribution

$$\hat{P}(z) = \sum_{k=1}^{K} \hat{\pi}_k \cdot \mathcal{N}(z|0, \hat{\Sigma}_k), \quad (35)$$

where the empirical Covariances and weights for the different Gaussians are defined as

$$\hat{\Sigma}_k = \frac{\sum_i T_{i,k} \left(\hat{\boldsymbol{R}}_{\boldsymbol{i}}\boldsymbol{DH}X\right)\left(\hat{\boldsymbol{R}}_{\boldsymbol{i}}\boldsymbol{DH}X\right)^T}{\sum_i T_{i,k}}$$

$$\hat{\pi}_k = \frac{\sum_i T_{i,k}}{\sum_{i,k} T_{i,k}}. \quad (36)$$

Given the empirical distribution, we aim to calculate its similarity to the local distribution using the Kullback-Leibler (KL) divergence. Unfortunately, the KL divergence between GMMs is not analytically traceable and we therefore approximate it as suggested in [29]. Instead of calculating the exact KL, we calculate an upper bound of the form

$$D_{KL}\left(\hat{P}||P\right) \leq D_{KL}(\hat{\pi}||\pi) + \sum_k \hat{\pi}_k D_{KL}\left(\mathcal{N}(0, \hat{\Sigma}_k), \mathcal{N}(0, \Sigma_k)\right). \quad (37)$$

The KL divergence for a pair of Gaussians has a closed form solution, and thus the above bound is simplified into

$$D_{KL}\left(\hat{P}||P\right) \leq \sum_k \hat{\pi}_k \left(\log\left(\frac{\hat{\pi}_k}{\pi_k}\right) + \frac{1}{2}\left(tr\left(\Sigma_k^{-1}\hat{\Sigma}_k\right) - \log\left|\Sigma_k^{-1}\hat{\Sigma}_k\right|\right)\right). \quad (38)$$

We shall use the above upper-bound in our experiments to tune the filter $\boldsymbol{H}$ as to obtain scale-invariance. Rather than optimizing $\boldsymbol{H}$ in full, we shall sweep over a parametric form of it, seeking the parameter that minimizes the approximated KL-distance. More specifically, we will assume $\boldsymbol{H}$ to be a centered and isotropic Gaussian and search for its optimal width.

V. Experimental Results

A. Scale Invariance

Throughout our experiments we use the GMM model which was used in [11] for the prior of the original scale. As for the others scales, we tune their filters such that the model is unchanged. Assume we restrict our filter $\boldsymbol{H}$ to be a Gaussian and we are interested in finding its optimal standard deviation, the one which would lead, as close as possible, to scale-invariance. We can sweep through a set of standard deviation values and use Equation (38) as an objective to minimize. In Figure 5 we present such an experiment and identify the best standard deviation for down-sampling by a factor of two and four. We will use these very filters in the denoising experiments that follow.

Prior to finding the optimal widths using the minimization of the KL divergence bound, we have searched for these values by directly optimizing the PSNR performance. The parameters found by both methods strongly agree, suggesting that the approximated method of minimizing the bound is indeed effective.

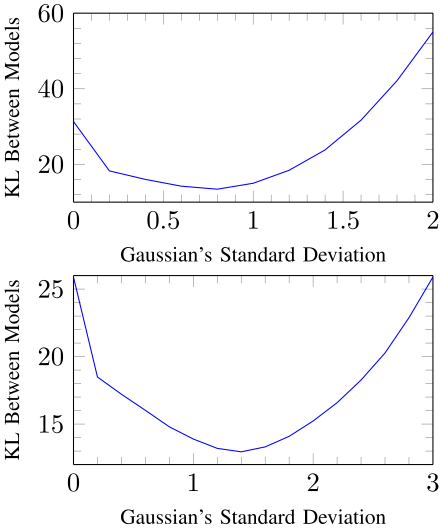
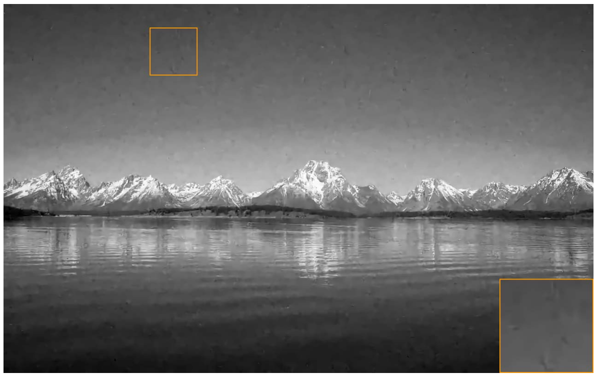
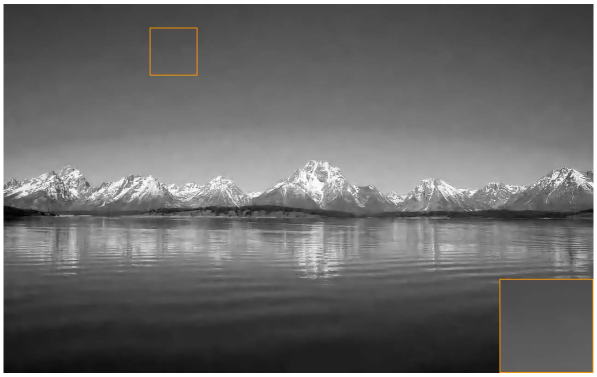


Fig. 5. The upper-bound on the KL divergence between the local model and the empirical distribution as appears in (38) for varying standard deviations of the blur operator. Left: down-sampling by factor of two. Right: down-sampling by factor of four.



(a)



(b)

Fig. 6. Denoising of a landscape image using the proposed method and the EPLL for comparison. (a) EPLL - PSNR: 33.31. (b) MSEPLL - PSNR: 33.61.

B. Image Denoising

We return now to the landscape image presented in Figure 1 in the introduction. One of the motivations for constructing a multi-scale prior was to remedy the artifacts obtained from local processing of the image. We present the denoised images obtained by the EPLL [11] and the proposed algorithm in Figure 6, showing that indeed artifacts are better treated. Furthermore, we compute the PSNR of both results in a sliding window and present the difference between them in Figure 7. One might expect the multi-scale EPLL to over-smooth the image content, thus leading to an improvement in smooth regions and degradation in highly textured areas. The result shown in Figure 7 shows that this is not the case. While indeed improving smooth areas, the multi-scale EPLL is also keeping the good quality of the regular EPLL and even slightly improves over it. For the multi-scale EPLL we added a single scale to the standard non-decimated one, which applied a Gaussian filter of standard deviation 1.5 followed by down-sampling by a factor of four.

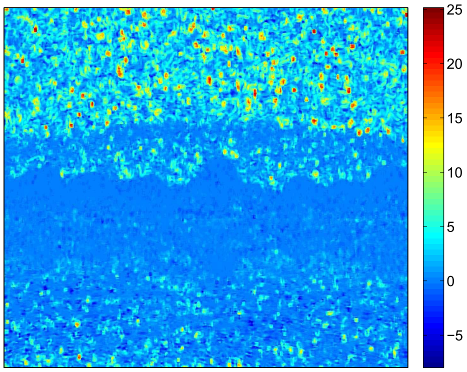


Fig. 7. The difference in PSNR between the EPLL and the multi-scale EPLL in a sliding window in the image. Notice how the difference in the PSNR in texture areas such as the mountains is small. Thus, texture areas are not smoothed.

TABLE I

Average PSNR for the Task of Denoising of 12 Images. The Standard Deviations of the Filters Were Chosen Based on the Previous Subsections Analysis.[5] The Weights Were Tuned by Seeking the Combination That Leads to the Best Average PSNR Over the Data-Set

σ	EPLL	MS-EPLL	Filter	DS	Weight
15	32.17	32.30	Identity	1	1
			Identity minus Gaussian with $\sigma = 0.6$	1	0.2
			Gaussian with $\sigma = 0.8$	2	0.2
25	29.74	29.89	Identity	1	1
			Gaussian with $\sigma = 0.8$	2	0.15
50	26.60	26.77	Identity	1	1
			Gaussian with $\sigma = 0.8$	2	0.05
			Gaussian with $\sigma = 1.5$	4	0.05
100	23.56	23.80	Identity	1	1
			Gaussian with $\sigma = 1.1$	2	0.04
			Gaussian with $\sigma = 1.7$	4	0.05

We continue the denoising experiments by testing our method on a set of 12 standard images[6] and comparing the results to those obtained by the original EPLL. A summary of the experiments and their results is presented in Table I. For every noise level we specify the filters used, the down-sampling (DS) factors and the weights for the

[5] The filters' width for the highest noise level are slightly higher than the ones for the other noise levels. This is because more aggressive filters have a smoothing effect on the final image, which is needed when we are dealing with high noise levels.

[6] The images are: Barbara, Boat, Cameraman, Couple, Fingerprint, Hill, House, Lena, Man, Montage, Pentagon and Peppers. These are commonly used in papers dealing with image denoising.

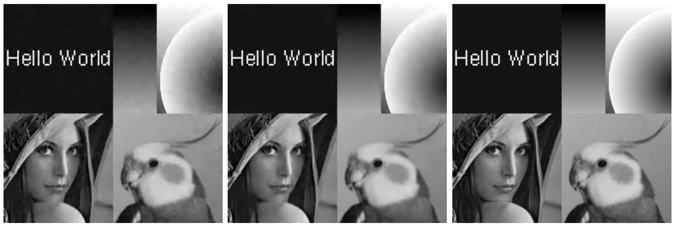


Fig. 8. Denoising results for the image Montage corrupted with a noise standard deviation $\sigma = 15$. From left to right: EPLL (PSNR = 34.18), MSEPLL (PSNR = 34.42) and original image.

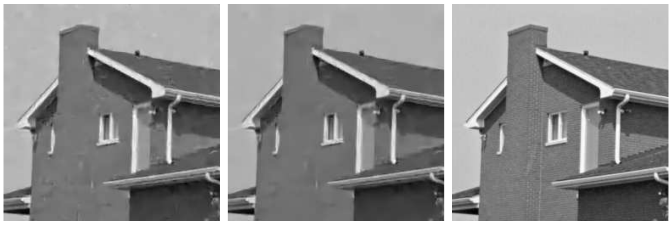


Fig. 9. Denoising results for the image House corrupted with a noise standard deviation $\sigma = 25$. From left to right: EPLL (PSNR = 32.04), MSEPLL (PSNR = 32.34) and original image.

different components. For most noise levels we use a structure that is similar to a Gaussian pyramid (i.e. down-scaling the image itself) and only for the lowest noise level we use a structure which reminds of a Laplacian pyramid (i.e. operating on the difference between the image and its down-scaled and up-scaled back). Since the Laplacian pyramid uses an added high frequency scale, we train a local high frequency model in a similar fashion to that of [11]. We extract patches from images which had their blurred version subtracted from them and afterwards learn a GMM high frequency model using the EM algorithm.[7]

In Figures 8, 9 and 10 we present examples of denoised images obtained by the EPLL and multi-scale EPLL. In addition, in Figure 10 we show the decimated versions of the denoised images to demonstrate the improvement in the denoising across all scales which is due to the additional terms of the multi-scale EPLL. We should note that in terms of the gain achieved in PSNR over the 12 test images, the average improvement might not be impressive (0.1-0.2dB). However, these test images are all relatively small and with emphasized texture parts, which are not typical of natural photos. Moreover, the visual improvement, as evident from Figures 8, 9 and 10 is substantial, something that does not reflect well in the PSNR results.

C. SURE Estimator

In this subsection we demonstrate the nonlinear SURE estimator presented in the previous section. We denoise an image, obtained from the Berkeley Segmentation Database [30], contaminated with noise of standard deviation $\sigma = 25$ using multi-scale EPLL and a single added Gaussian scale. We explore the different weights for the added scale and

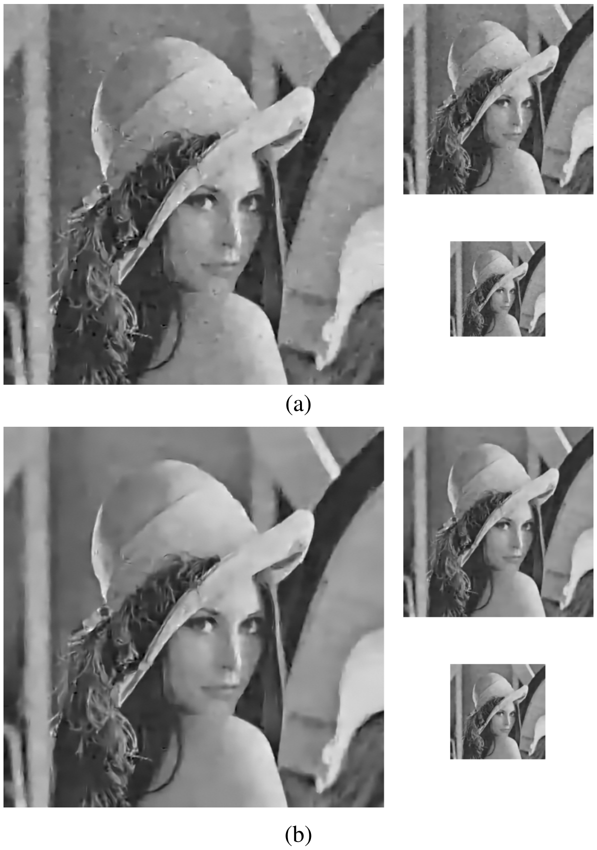


Fig. 10. Denoising results for Lena corrupted with a noise standard deviation $\sigma = 50$. Left: denoised image. Top right: the denoised image filtered and down-sampled by factor of two. Bottom right: the denoised image filtered and down-sampled by factor of four. (a) EPLL - PSNR: 28.61. (b) Multi-Scale EPLL - PSNR: 28.99.

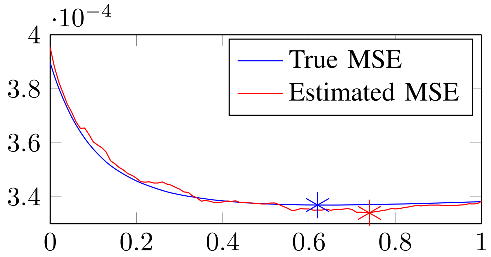


Fig. 11. True MSE and estimated MSE using SURE for different weight values. The estimated MSE was shifted by a constant such that the graphs overlap. The star marker highlights the location of the minimum in the graphs.

present the results in Figures 11 and 12. We compare the results to the original EPLL and the multi-scale EPLL using the weight 0.15 which was found in the previous subsection.

D. Deblurring

We continue our experiments with image deblurring. We test our method on a set of 10 standard images[8] and compare the results to those obtained by the original EPLL. The summary of the experiments is presented in table II. For every degradation method we specify the down-sampling

[7] The code used was downloaded from http://www.mathworks.com/matlabcentral/fileexchange/26184-em-algorithm-for-gaussian-mixture-model.

[8] The images are: Barbara, Boats, Butterfly, Cameraman, House, Leaves, Lena, Parrots, Peppers and Starfish. These are commonly used in papers dealing with image deblurring.

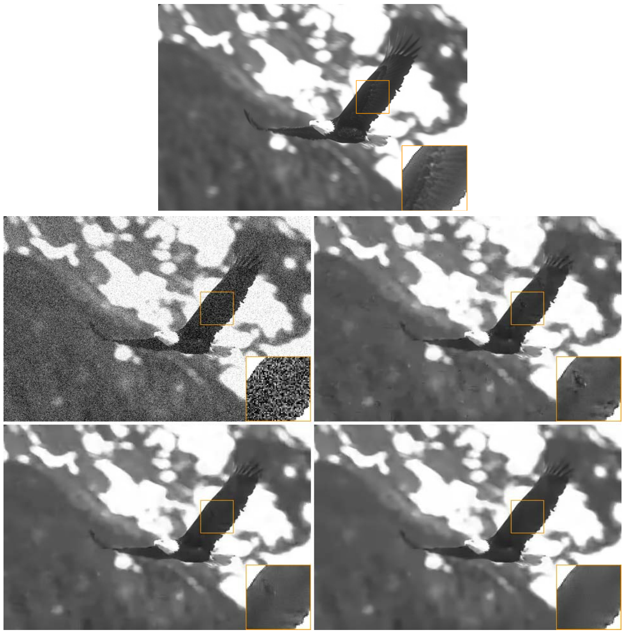


Fig. 12. Top: original image. Middle left: noisy image (PSNR=20.17). Middle right: EPLL (PSNR=34.09). Bottom left: multi-scale EPLL with the weight optimized over the dataset (PSNR=34.55). Bottom right: multi-scale EPLL with the weight optimized using SURE (PSNR=34.72). A linear contrast stretch was applied on all the magnified areas. The image obtained using the optimal weight is not presented since it was found to be almost the same as the one obtained with SURE.

TABLE II

AVERAGE PSNR FOR THE TASK OF DEBLURRING ON 10 IMAGES

σ	Blur	EPLL	MS-EPLL	DS	Weight
$\sqrt{2}$	Uniform 9×9	28.74	29.01	1	1
				2	-0.25
				4	0.02
2	Uniform 9×9	27.90	28.23	1	1
				2	-0.25
				4	0.02
$\sqrt{2}$	Gaussian with standard deviation $\sigma = 1.6$	29.89	30.16	1	1
				2	-0.25
				4	0.02
2	Gaussian with standard deviation $\sigma = 1.6$	29.34	29.61	1	1
				2	-0.25
				4	0.02

factors and the weights for the different components. We should note that in this part of our experiments we do not use a filter before down-sampling as we found its omission to lead to best results.[9] Notice that the weight w_2 is found to be negative for getting best performance. Discussion on this follows towards the end of this section. In Figures 13, 14 and 15 we present examples of deblurred images obtained by the EPLL and multi-scale EPLL.

E. Super-Resolution

We conclude our experiments by tackling the super-resolution problem. To degrade an image, we filter and down-sample it, and then add noise of standard deviation σ. We test our method on a set of 9 standard color images.[10] To treat the color, we convert the image to the YCbCr color space. Then, we use bicubic interpolation on the luminance channel and use the output as an initialization for the super-resolution algorithm. As for the chroma channels, we use bicubic interpolation on the low resolution image. The summary of the experiments for different up-scale factors and noise levels is presented in table III. We specify the down-sampling factors and the weights for the different components used by the multi-scale EPLL. Similarly to deblurring, in the mutli-scale EPLL we do not use a filter before down-sampling as we found this to lead to best results. Here as well, some of the weights are found to be negative. In Figures 16, 17 and 18 we present examples of resulting images obtained by the EPLL and multi-scale EPLL.

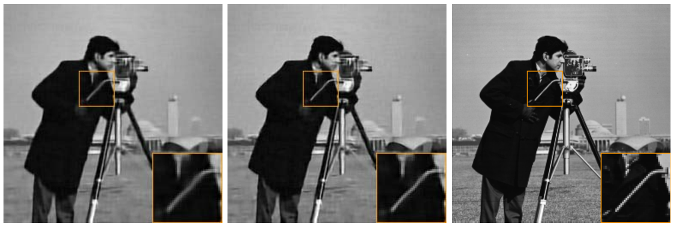


Fig. 13. Deblurring results for the image Cameraman blurred with uniform 9×9 filter and corrupted by noise of standard deviation $\sigma = 2$. From left to right: EPLL (PSNR = 26.90), MSEPLL (PSNR = 27.38) and original image.

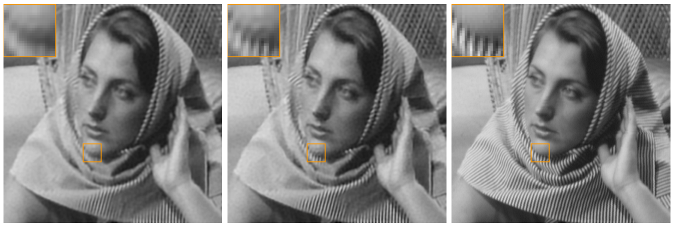


Fig. 14. Deblurring results for the image Barbara blurred with Gaussian filter with standard deviation $\sigma = 1.6$ and corrupted by noise of standard deviation $\sigma = \sqrt{2}$. From left to right: EPLL (PSNR=25.84), MSEPLL (PSNR=26.51) and original image.

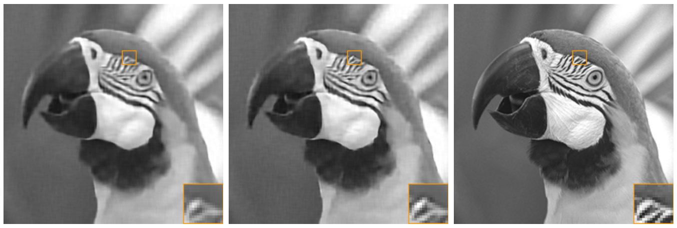


Fig. 15. Deblurring results for the image Parrot blurred with uniform 9×9 filter and corrupted by noise of standard deviation $\sigma = 2$. From left to right: EPLL (PSNR=28.42), MSEPLL (PSNR=28.89) and original image.

F. Negative Weights

Throughout the coarse of searching for the optimal weights for the different scales in the multi-scale EPLL prior, we have found in some cases that some weights should be set negative.

[9] At each iteration of the multi-scale EPLL, we reconstruct the decimated image by patch-averaging. This alone has the effect of a low-pass filter, and thus, even if no filtering is done explicitly, we still obtain a smoothing effect.

[10] The images are: Bike, Butterfly, Flower, Girl, Hat, Parrot, Parthenon, Plants and Raccoon. These are commonly used in papers dealing with image super-resolution.

TABLE III

AVERAGE PSNR FOR THE TASK OF SUPER RESOLUTION ON 9 IMAGES

Factor	σ	EPLL	MS-EPLL	DS	Weight
2	5	30.78	31.52	1	1
				2	-0.45
3	5	27.80	28.24	1	1
				2	-0.4
2	1	32.15	32.68	1	1
				2	-0.35
3	1	29.17	29.46	1	1
				2	-0.35

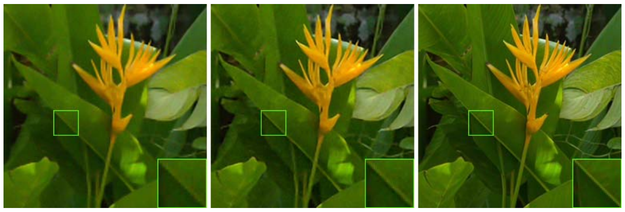


Fig. 16. Super-resolution performance comparison on the image Plant. Scaling factor is 2 and noise standard deviation is $\sigma = 5$. From left to right: EPLL (PSNR=33.90), MSEPLL (PSNR=35.28) and original image.

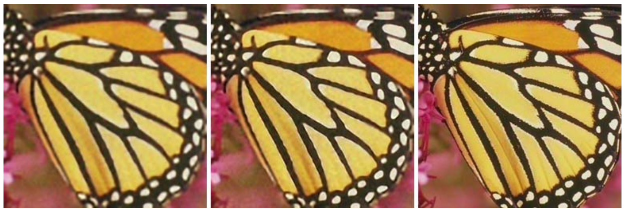


Fig. 17. Super-resolution performance comparison on the image Butterfly. Scaling factor is 3 and noise standard deviation is $\sigma = 5$. From left to right: EPLL (PSNR=25.81), MSEPLL (PSNR=26.88) and original image.

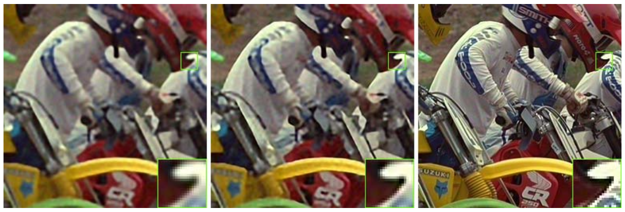


Fig. 18. Super-resolution performance comparison on the image Bike. Scaling factor is 3 and noise standard deviation is $\sigma = 5$. From left to right: EPLL (PSNR=23.21), MSEPLL (PSNR=23.80) and original image.

A similar phenomenon has been reported in [31], where it was suggested that denoising filters could benefit from having negative entries. In this subsection we discuss this intriguing phenomenon.

Intuitively, when solving the half-splitting optimization we iterate two steps. In the first, we update the auxiliary patches z_i and $\hat{z}_i$, and in the second, the image is updated by solving a quadratic problem

$$X = \left(\lambda \boldsymbol{A}^T \boldsymbol{A} + w_1 \beta \sum_i \boldsymbol{R}_i^T \boldsymbol{R}_i + w_2 \hat{\beta} \sum_i \boldsymbol{S}^T \hat{\boldsymbol{R}}_i^T \hat{\boldsymbol{R}}_i \boldsymbol{S}\right)^{-1} \times \left(\lambda \boldsymbol{A}^T Y + w_1 \beta \sum_i \boldsymbol{R}_i^T z_i + w_2 \hat{\beta} \sum_i \boldsymbol{S}^T \hat{\boldsymbol{R}}_i^T \hat{z}_i\right). \tag{39}$$

The images $\sum_i \boldsymbol{R}_i^T z_i$ and $\sum_i \boldsymbol{S}^T \hat{\boldsymbol{R}}_i^T \hat{z}_i$ are approximations of the original image and its blurred version, respectively. By setting the weight w_2 to be negative, we subtract from the original image its blurred version, thus obtaining a sharpening effect which aids us in the tasks of deblurring and super-resolution.

A related explanation originates from the realm of partial differential equations in image processing. The EPLL method is an iterative restoration algorithm, and as such, it can be viewed as a diffusion process. When dealing with the deblurring and super-resolution problems, plain diffusion is insufficient since it does not add new details to the already existing ones in the initial image. To alleviate the problem, it is possible to use a reaction-diffusion process, also known as forward-and-backward diffusion [32]–[35]. The forward process denoises smooth regions while the backward procedure enhances edges and features. Intuitively, the backward diffusion is an attempt to move back in time and reverse the diffusion process, as to reconstruct the lost features. The negative weight in our multi-scale EPLL prior can be viewed as a similar attempt to add a backward diffusion process to the already existing forward diffusion done by the original EPLL. By changing the sign of the added scale we attempt to invert the diffusion and as such enhance the details.

Another interpretation of the negative weights is the following: By assigning the second weight to be negative, our global prior becomes a function of the ratio between the probabilities of the different scales. This, in turn, means we do not care about how likely a single scale is, we only care about how likely the scales are compared to each other. This observation might lead to a new family of priors which aim to minimize the ratio, or perhaps even the distance, between the probabilities of the different scales. Such a global prior might seek a reconstructed image which obeys the scale invariance property.

VI. CONCLUSION

Image priors are of utmost importance in image processing restoration tasks. An example of such a prior is the GMM, as practiced by the EPLL algorithm. EPLL models a whole image by characterizing its overlapped patches, forcing every patch extracted from the image to be likely given a local GMM model. In this paper we propose a multi-scale EPLL which imposes the very same patch-based model on different scale patches extracted from the image. We motivate its use by looking at the simplified Gaussian case, showing that such an approach manages to narrow the gap to the global modeling while preserving the local treatment. We then compare the proposed method to the original EPLL on the tasks of image denoising, deblurring and super-resolution, and show a clear improvement across all tasks.

ACKNOWLEDGMENT

V. Papyan would like to thank Yaniv Romano and Jeremias Sulam for fruitful discussions and advices which helped in developing the presented work. The authors of this paper also thank D. Zoran and Y. Weiss for sharing their code. Finally, the authors thank the reviewers for their constructive and insightful comments.

REFERENCES

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

[2] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[3] 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.*, vol. 21, no. 5, pp. 2481–2499, May 2012.

[4] I. Ram, M. Elad, and I. Cohen, “Image processing using smooth ordering of its patches,” *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2764–2774, Jul. 2013.

[5] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2272–2279.

[6] P. Chatterjee and P. Milanfar, “Patch-based near-optimal image denoising,” *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1635–1649, Apr. 2012.

[7] J. Mairal, M. Elad, and G. Sapiro, “Sparse representation for color image restoration,” *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

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

[9] W. Dong, L. Zhang, G. Shi, and X. Li, “Nonlocally centralized sparse representation for image restoration,” *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[10] S. Roth and M. J. Black, “Fields of experts,” *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 205–229, Apr. 2009.

[11] D. Zoran and Y. Weiss, “From learning models of natural image patches to whole image restoration,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 479–486.

[12] S. Hawe, M. Kleinsteuber, and K. Diepold, “Analysis operator learning and its application to image reconstruction,” *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2138–2150, Jun. 2013.

[13] R. Rubinstein, T. Peleg, and M. Elad, “Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model,” *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 661–677, Feb. 2013.

[14] D. Geman and C. Yang, “Nonlinear image recovery with half-quadratic regularization,” *IEEE Trans. Signal Process.*, vol. 4, no. 7, pp. 932–946, Jul. 1995.

[15] J. Sulam and M. Elad, “Expected patch log likelihood with a sparse prior,” in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. New York, NY, USA: Springer-Verlag, 2015, pp. 99–111.

[16] G. Gilboa and S. Osher, “Nonlocal operators with applications to image processing,” *Multiscale Model. Simul.*, vol. 7, no. 3, pp. 1005–1028, 2008.

[17] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Jun. 2005, pp. 60–65.

[18] C.-Y. Yang, J.-B. Huang, and M.-H. Yang, “Exploiting self-similarities for single frame super-resolution,” in *Computer Vision—ACCV*. New York, NY, USA: Springer-Verlag, 2011, pp. 497–510.

[19] M. Zontak, I. Mosseri, and M. Irani, “Separating signal from noise using patch recurrence across scales,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1195–1202.

[20] T. Michaeli and M. Irani, “Nonparametric blind super-resolution,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 945–952.

[21] T. Michaeli and M. Irani, “Blind deblurring using internal patch recurrence,” in *Computer Vision—ECCV*. New York, NY, USA: Springer-Verlag, 2014, pp. 783–798.

[22] D. L. Ruderman and W. Bialek, “Statistics of natural images: Scaling in the woods,” *Phys. Rev. Lett.*, vol. 73, no. 6, pp. 814–817, Aug. 1994.

[23] X. Pitkow, “Exact feature probabilities in images with occlusion,” *J. Vis.*, vol. 10, no. 14, p. 42, 2010.

[24] J. Sulam, B. Ophir, and M. Elad, “Image denoising through multi-scale learnt dictionaries,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 808–812.

[25] B. Ophir, M. Lustig, and M. Elad, “Multi-scale dictionary learning using wavelets,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 1014–1024, Sep. 2011.

[26] S. Ramani, T. Blu, and M. Unser, “Monte-Carlo sure: A black-box optimization of regularization parameters for general denoising algorithms,” *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1540–1554, Sep. 2008.

[27] 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 Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[28] C. M. Stein, “Estimation of the mean of a multivariate normal distribution,” *Ann. Statist.*, vol. 9, no. 6, pp. 1135–1151, 1981.

[29] M. N. Do, “Fast approximation of Kullback–Leibler distance for dependence trees and hidden Markov models,” *IEEE Signal Process. Lett.*, vol. 10, no. 4, pp. 115–118, Apr. 2003.

[30] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. 8th Int. Conf. Comput. Vis.*, vol. 2. Jul. 2001, pp. 416–423.

[31] P. Milanfar, “A tour of modern image filtering: New insights and methods, both practical and theoretical,” *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2013.

[32] G. Plonka and J. Ma, “Nonlinear regularized reaction-diffusion filters for denoising of images with textures,” *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1283–1294, Aug. 2008.

[33] G.-H. Cottet and L. Germain, “Image processing through reaction combined with nonlinear diffusion,” *Math. Comput.*, vol. 61, no. 204, pp. 659–673, Oct. 1993.

[34] L. Kuhnert, K. I. Agladze, and V. I. Krinsky, “Image processing using light-sensitive chemical waves,” *Nature*, vol. 337, pp. 244–247, Jan. 1989.

[35] G. Gilboa, N. Sochen, and Y. Y. Zeevi, “Forward-and-backward diffusion processes for adaptive image enhancement and denoising,” *IEEE Trans. Image Process.*, vol. 11, no. 7, pp. 689–703, Jul. 2002.

Vardan Papyan received the B.Sc. degree from the Department of Computer Science, Technion–Israel Institute of Technology, in 2013, where he is currently pursuing the M.Sc. degree. His research interests are signal and image processing, in particular, inverse problems.

Michael Elad (F’12) received the B.Sc., M.Sc., and D.Sc. degrees from the Department of Electrical Engineering, Technion, Israel, in 1986, 1988, and 1997, respectively. Since 2003, he has been a Faculty Member with the Computer-Science Department, Technion, and has held a full-professorship position since 2010. He serves as the Head of the Prestigious Rothschild Technion Program for Excellence. He works in the field of signal and image processing, specializing in particular on inverse problems, sparse representations, and superresolution. He received the Technion’s Best Lecturer Award six times. He was a recipient of the 2007 Solomon Simon Mani Award for excellence in teaching, the 2008 and 2015 Henri Taub Prize for academic excellence, and the 2010 Hershel-Rich prize for innovation. He serves as the Editor-in-Chief of the SIAM JOURNAL ON IMAGING SCIENCES.