AUTOMATIC PARAMETER TUNING FOR IMAGE DENOISING WITH LEARNED SPARSIFYING TRANSFORMS

Luke Pfister and Yoram Bresler

Department of Electrical and Computer Engineering and the Coordinated Science Laboratory,
University of Illinois, Urbana-Champaign, IL 61801, USA

ABSTRACT

Data-driven and learning-based sparse signal models outperform analytical models (e.g, wavelets), for image denoising, but require careful parameter tuning to reach peak performance. In this work, we provide a solution to the problem of parameter tuning for image denoising with transform sparsity regularization. We show that by viewing a learned sparsifying transform as a filter bank we can utilize the SURELET denoising algorithm to automatically tune parameters for an image denoising task.

Numerical experiments show that combining SURELET with a learned sparsifying transform provides the best of both worlds. Our approach requires no parameter tuning for image denoising, yet outperforms SURELET with analytic transforms and matches the performance of transform learning denoising with hand-tuned parameters.

Index Terms— Sparsifying transform learning, Sparse representations, linear expansion of thresholds (LET), image denoising, Stein unbiased risk estimator

1. INTRODUCTION

The assumption that a signal admits a sparse representation is now ubiquitous throughout signal and image processing. These representations typically obey the synthesis sparsity model, wherein a signal is constructed as the sum of a few atomic signals, or the analysis sparsity model, wherein a signal becomes sparse after being acted on by a linear operator. The *transform sparsity model* is a close cousin of the analysis sparsity model. A signal $x \in \mathbb{R}^n$ satisfies the transform sparsity model if there is a matrix, $W \in \mathbb{R}^{m \times n}$, such that $Wx = z + \eta$, where z is sparse and $\|\eta\|_2$ is small. The matrix W is called an *sparsifying transform* and z is called an *transform sparse code*. For non-square W, the transform model differs from the strict analysis model by allowing for nonzero η and for the sparse component z to lie outside of the range space of W.

The problem of finding z given W and x is called *transform sparse coding* and is written

$$\arg\min_z \frac{1}{2}\|Wx - z\|_2^2 + \nu\phi(z) \tag{1}$$

where $\phi : \mathbb{R}^m \to \mathbb{R}$ is a sparsity-promoting functional such as the ℓ_0 quasi-norm or the indicator function for s-sparse vectors. The solution of this problem is given by $\text{prox}_\phi(Wx, \nu)$[1]. If ϕ is a coordinate-wise separable function ($\phi(z) = \sum_{i=1}^m \hat{\phi}(z_i)$), then (1) reduces to a set of scalar minimization problems. These minimization problems can be solved in closed-form expression for many popular choices of ϕ, *e.g.* if $\phi(z) = \|z\|_0$, then the solution is given by hard thresholding: $z_i = [Wx]_i$ if $([Wx]_i)^2 \geq \nu$ and 0 otherwise.

Sparse representations have traditionally been designed to provide desirable properties on a mathematical classes of signals, but it is difficult to specialize these representations to real-world or high-dimensional signals. This observation inspired the development of algorithms to *learn* as sparse model directly from data; an approach that sacrifices optimality properties on mathematical signal classes for empirical performance on a limited class of real-world signals. We refer to these as *data-driven sparse representations*. Algorithms have been proposed to learn representations for the synthesis [1, 2, 3], analysis [4, 5, 6, 7, 8, 9], and transform [10, 11, 12] models.

Data-driven sparse representations have been used to achieve state-of-the-art performance in a variety of inverse problems, including image denoising [8, 13, 9, 11, 10] , magnetic resonance imaging [14, 15], and computed tomography[16, 17, 18, 19, 20, 21]. This is achieved by either learning a sparse representation offline over a set of training data, which we call the "universal" approach, or while solving the inverse problem, which we call the "adaptive" approach.

One of the key challenges in regularization with data-driven sparse models is the need to tune many parameters; a slow and cumbersome process. In this work, we present a fast automated parameter tuning method solution for a restricted class of problems: denoising an image, corrupted by Gaussian noise, using transform sparsity regularization with a pre-learned sparsifying transform. Our method combines the recently proposed filter bank formulation of transform sparsity [15] with the SURELET algorithm, and [22], requires no parameter tuning for denoising, yet outperforms SURELET with DCT and Haar transforms and performs on par with existing universal and adaptive transform learning denoising algorithms with carefully chosen parameters.

2. FILTER BANK SPARSIFYING TRANSFORMS

Data-driven sparse representations often do not directly model the image, but rather smaller, possibly overlapping, sub-images called "patches". The resulting model is said to be *patch-based*. We represent the j-th patch of an image $x \in \mathbb{R}^n$ as $R_j x \in \mathbb{R}^K$, where $K \ll n$. The matrix $R_j \in \mathbb{R}^{K \times n}$ is called a patch extraction operator. Its adjoint, R_j^T, places a patch into an n-dimensional vector at the original location of the j-th patch. We have flexibility in choosing the degree of overlap between neighboring patches and behavior at image boundaries. As the patch size is chosen to be much smaller than the image, few parameters are needed to describe a patch-based sparsifying transform. This leads to computationally efficient algorithms and reduces the risk of overfitting.

Patch-based models are commonly used for sparsifying trans-

This work was supported in part by the National Science Foundation (NSF) under Grants CCF 1018660 and CCF-1320953.

[1] $\text{prox}_f(x, \nu) \triangleq \arg\min_z 0.5\|x - z\|_2^2 + \nu f(z)$

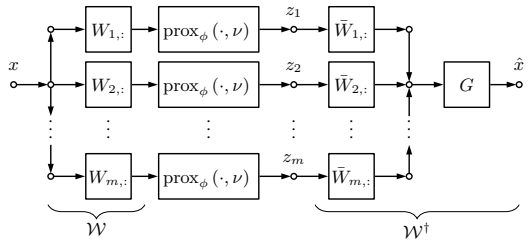

Fig. 1: Filter bank structure of transform sparse coding and reconstruction. Here, $\bar{W}_{i,:}$ is the i-th time-reversed filter.

forms, and typical practice is to treat the transform as a local operator that acts on each patch independently. However, we can group the patch extraction operators and the transform W into a single matrix $\mathcal{W} \in \mathbb{R}^{mn \times n}$ that acts on the entire image x. We call $\mathcal{W}$ an image-based, rather than patch-based, sparsifying transform.

The matrix structure of $\mathcal{W}$ depends on the patch extraction procedure. If maximally overlapping patches are extracted using periodic boundary conditions, then $\mathcal{W} = [\mathcal{W}_1^T, \mathcal{W}_2^T, \dots \mathcal{W}_m^T]^T$ is a stack of block-circulant matrices with circulant blocks, with each row of W generating a block circulant matrix. We can thus view $\mathcal{W}$ as an m-channel undecimated filter bank whose filters are given by the rows of W. With this view, transform sparse coding has a particularly nice interpretation: it is viewed as of as passing x through the filter bank $\mathcal{W}$, then applying $\text{prox}_\phi(\cdot, \nu)$ to each channel. If $\mathcal{W}$ is left invertible, its pseudoinverse $\mathcal{W}^\dagger = (\mathcal{W}^T\mathcal{W})^{-1}\mathcal{W}^T$ is a synthesis filter bank. This can be implemented by filtering using time-reversed versions of the filters in $\mathcal{W}$, followed by the common filter $G = (\mathcal{W}^T\mathcal{W})^{-1}$. This notion of an image-level transform and the structure of transform sparse coding will be key in the development of our parameter tuning method.

3. TRANSFORM DENOISING SCHEMES

We begin by reviewing existing transform sparsity denoising algorithms with a fixed, patch-based transform $W \in \mathbb{R}^{m \times k}$. We assume W is full column rank (as is required in transform learning algorithms). Our goal is to recover an image $x^* \in \mathbb{R}^n$ from a noisy observation y. Several methods have been proposed to solve this task, differing primarily in they manner in which local estimates are related to the final denoised image.

The simplest approach is apply transform sparse coding to each patch of y and construct the final estimate by averaging the overlapping denoised patches (Algorithm 1). Note that if the patches are non-overlapping, W is orthonormal, and ϕ is the ℓ_0 norm, then Algorithm 1 is identical to classical orthonormal transform denoising.

Algorithm 1 Denoising by Transform Sparse Coding

1: $z_j \leftarrow \text{prox}_\phi(WR_jy, \nu)$
2: $x \leftarrow \frac{1}{K}\sum_j R_j^T W^\dagger z_j$

A second strategy is to model each noisy patch, R_jy, as being close to a "clean" patch $\hat{x}_j$ that is sparsified by W . Recovery of the j-th clean patch can be posed as [14]

$$\min_{x,z_j} \frac{1}{2}\|W\hat{x}_j - z_j\|_2^2 + \frac{\tau}{2}\|R_jy - \hat{x}_j\|_2^2 + \nu\phi(z_j), \qquad (2)$$

where τ is a penalty parameter. An alternating minimization algorithm has been proposed to solve (2), and is given as Algorithm 2. We call this algorithm Iterative Patch Denoising. Again, the final denoised image is given by averaging each denoised patch. These approaches are natural when non-overlapping patches are used, but overlapping patches are typically used in practice as they reduce artifacts near patch boundaries. In this setting we expect Algorithms 1 and 2 to be sub-optimal as they neglect any correlation between neighboring image patches.

Algorithm 2 Iterative Patch Denoising (IPD)

1: $\hat{x}_j^0 \leftarrow R_jy, \quad k \leftarrow 0$
2: **repeat**
3: $\quad z_j^k \leftarrow \text{prox}_\phi\left(W\hat{x}_j^k, \nu\right)$
4: $\quad \hat{x}_j^k \leftarrow (W^TW + \tau I)^{-1}(W^Tz_j^k + \tau R_jy)$
5: $\quad k \leftarrow k + 1$
6: **until** Halting condition
7: $x = \frac{1}{K}\sum_j R_j^T\hat{x}_j^k$

A third denoising scheme involves explicitly modeling the relationship between denoised image patches and the final image within the objective function. This recovery problem is given by

$$\min_{x,z_j} \frac{1}{2}\sum_{j=1}\|WR_jx - z_j\|_2^2 + \frac{\tau}{2}\|y - x\|_2^2 + \nu\phi(z_j), \qquad (3)$$

and can be solved using Algorithm 3, which we call Iterative Global Denoising (IGD). Note that a single IGD iteration with $\tau = 0$ is equivalent to performing transform sparse coding denoising using the image-based sparsifying transform $\mathcal{W}$.

Algorithm 3 Iterative Global Denoising (IGD)

1: $x^0 \leftarrow y, \quad k \leftarrow 0$
2: **repeat**
3: $\quad z_j^k \leftarrow \text{prox}_\phi\left(WR_jx^k, \nu\right)$
4: $\quad x^k \leftarrow (\sum_j R_j^TW^TWR_j + \tau I)^{-1}(\sum_j R_j^TW^Tz_j^k + \tau y)$
5: $\quad k \leftarrow k + 1$
6: **until** Halting condition

Many features of Algorithms 1-3 are attractive. They are well-suited for both universal and adaptive transform learning regularization, and there is no explicit reliance on the noise distribution. Further, Algorithms 2 and 3 can be adapted to solve a general inverse problem. However, these algorithms suffer from the need to tune many parameters, including

- the sparsity penalty parameter ν (often varied per-patch),
- the penalty parameter τ,
- the halting criterion,

and, if the transform is to be learned jointly during denoising,

- any parameters to regularize the learning problem,
- sparsity level scheduling (typically enforcing stronger sparsity as the transform is refined),
- the number of transform update steps per each image update,

among others. Careful selection of these parameters is key to successful performance of a transform sparsity denoising algorithm.

4. SURE-BASED TRANSFORM DENOISING

Our goal is to develop an efficient transform denoising algorithm that does not require careful parameter tuning. We will write our denoised signal as $x = F_\theta(y)$, where y represents the noisy signal and θ is the set of parameters for our denoising algorithm. Our metric of interest is the mean squared error (MSE) between the true signal x^*

and our estimate $F_\theta(y)$, defined as $\text{MSE} = n^{-1}\|x^* - F_\theta(y)\|_2^2$. We could hope to minimize MSE by optimizing over θ; of course, this procedure requires knowledge of x^*, which we lack. What is needed is a way to estimate the MSE from y and $F_\theta(y)$.

To that end, we restrict our attention to the following setting: our sparsifying transform $W \in \mathbb{R}^{m\times K}$ is fixed, and its induced image-based sparsifying transform $\mathcal{W}$ is left invertible[2]. We wish to recover an image $x^* \in \mathbb{R}^n$ from noisy observations $y = x^* + e$, where $e \sim \mathcal{N}(0, \sigma^2 I_n)$ with σ^2 known.

In this limited context, we gain a powerful tool: Stein's Unbiased Risk Estimator (SURE) provides an unbiased estimate of the true MSE, provided that our estimator $F_\theta(y)$ is differentiable with respect to y [23]. Each of Algorithms $1-3$ satisfy this requirement whenever $\text{prox}_\phi(\cdot, \nu)$ is differentiable. Unfortunately, this precludes selecting ϕ as the ℓ_1 or ℓ_0 norms, as the soft and hard thresholding functions are non-differentiable. This can be avoided by using a smoothed version of these prox functions. We take a different approach: we replace the prox function entirely by a differentiable, pointwise, but otherwise arbitrary thresholding function ψ with no regard to the variational formulation (1). [3]

SURE provides an estimate of the MSE for any differentiable estimator $F_\theta(y)$, and many algorithms have been proposed for automatic parameter tuning using the SURE criterion. Often, the choice of minimizing θ cannot be written in closed form and an iterative solution is necessary- as is the case if we use SURE to tune τ in Algorithms 2 or 3 [25, 26]. This is unattractive if we wish to tune thresholding functions on a channel-by-channel basis.

An alternative is to use the SURE Linear Expansion of Thresholds (SURELET) strategy: we expand our denoising function as $F_\theta(y) = \sum_{i,j} c_{i,j} F_{i,j}(y)$, where $F_{i,j} : \mathbb{R}^n \to \mathbb{R}^n$ is an "elementary denoising" function and $c_{i,j}$ are mixing coefficients [22]. For this parameterization of $F_\theta(y)$, minimizing SURE can be done in closed-form and the resulting denoising algorithm is both non-iterative and computationally inexpensive.

The elementary denoising functions are of the form $F_{i,j}(y) = \mathcal{R}_i \psi_j(\mathcal{D}_i y)$ for some decomposition operator $\mathcal{D}_i$, reconstruction operator $\mathcal{R}_i$, and pointwise thresholding function ψ_j. We could use a patch-based method and set $F_{i,j}(y) = K^{-1} R_i^T W^{-1} \psi_j(W R_i y)$. Unfortunately, this results in Pn mixing coefficients if P thresholding functions are used. As we have only n independent samples of y, the SURE will have high variance and be of little use as an estimator.

Instead, we adopt the filter bank perspective. Let $\mathcal{W}_i$ be the linear operator implementing the i-th channel of the analysis filter and $\mathcal{S}_i = (\mathcal{W}^T\mathcal{W})^{-1}\mathcal{W}_i$ be the corresponding channel of the synthesis filter bank. Our elementary denoising functions are of the form $F_{i,j}(y) = \mathcal{S}_i \psi_j(\mathcal{W}_i y)$ for $1 \le i \le m$ and $1 \le j \le P$. This results in $Pm \ll n$ coefficients and thus reasonable variance of the SURE. With the form of F_θ fixed, the SURE can be expressed as:

Theorem 1 (Adapted from Corollary 1, [22])**.** *Let* $F_\theta(y) = \sum_{i,j} c_{i,j} \mathcal{S}_i \psi_j(\mathcal{W}_i y)$ *satisfy* $\mathbb{E}\,|\partial [F_\theta(y)]_k / \partial y_k| < \infty$ *for* $k = 1, \dots N$. *Then*

$$\varepsilon = \frac{1}{n}\|F_\theta(y) - y\|_2^2 + \frac{2\sigma^2}{n}\sum_{i,j} c_{i,j}\alpha_i^T \psi_j'(\mathcal{W}_i y) + \sigma^2, \qquad (4)$$

where $\alpha_i = \text{diag}\{\mathcal{S}_i \mathcal{W}_i\}$*, is an unbiased estimate of the MSE* $= n^{-1}\|F_\theta(y) - x^*\|_2^2$.

If we elect to use periodic boundary conditions, the vector α_i can easily be found as each of the circulant matrices $\mathcal{W}_i$ is diagonalized by the DFT. Let $Q \in \mathbb{C}^{n\times k}$ be a matrix implementing a zero-padded DFT, and let $w_i \in \mathbb{R}^k$ be the i-th row of W formed into a column vector. Then the eigenvalues of $\mathcal{W}_i^T\mathcal{W}_i$ are $|Qw_i|^2$, and we can calculate $\alpha_i = \| |Qw_i|^2 / (\sum_{j=1}^M |Qw_j|^2) \|_1$ where $|\cdot|^2$ and division are applied elementwise. For non-periodic boundary conditions, a randomized algorithm can be used [22].

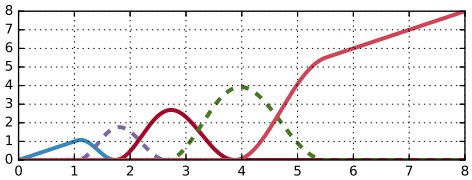

Fig. 2: SURE-BUMP thresholding functions

As F_θ is linear in the coefficients $c_{i,j}$, the SURE is quadratic in these coefficients, and can thus be minimized by solving the linear system of equations $Ac = u$. Here, the matrix $A \in \mathbb{R}^{Pm\times Pm}$ and vector $u \in \mathbb{R}^{Pm}$ have entries

$$A_{i+mj,i'+mj'} = \sum_{i,j} F_{i,j}(y) F_{i'j'}(y), \quad 1 \le i \le m \qquad (5)$$

$$u_{i+jm} = F_{i,j}(y)^T y - \sigma^2 \alpha_i^T \psi_j'(\mathcal{W}_i y) \quad 1 \le j \le P. \qquad (6)$$

This is a small system of equations: of size 320×320 for a 64×64 sparsifying transform and 5 thresholding functions.

Our remaining task is to choose the thresholding functions. We found that the SURE-BUMP basis of thresholding functions [27] yielded better performance with our learned transforms than the basis proposed in the original SURELET work. These thresholding functions are given by $\psi_j(z) = z \cdot f(\alpha \log\left(|z|\,\sigma^{-1} + 1\right) + \beta - j)$ where $f(x) = \cos^2\left(\frac{\pi x}{2}\right)$ if $|x| \le 1$ and 0 otherwise. We choose α and β to set the center of ψ_1 and ψ_4 to $\sqrt{3}\|w_i\|_2$ and $\sqrt{15}\|w_i\|_2$, where w_i is the i-th row of W. This ensures that our denoising result is invariant to a rescaling of the rows of W; a property that proves to be welcome in practice. We found that using 5 thresholding functions provides good performance without leading to large estimator variance. We modify ψ_1 to linear to the left of its peak and modify ψ_5 to be linear on the right of its peak for added flexibility in handling small and large coefficients, respectively. Our basis of thresholding functions is shown in Figure 2.

The complete SURELET algorithm is listed as Algorithm 4. The dominant computation being calculation of $\mathcal{S}_i\psi_j(\mathcal{W}_i y)$ for $i = 1, \dots m$ and $j = 1 \dots P$. Assuming periodic boundary conditions and using Fourier-based convolution, Algorithm 4 requires a total of $2m(P+1)+1$ FFTs. Using a filter bank structure to implement a single iteration of Algorithm 3 requires $3m+2$ FFTs, and denoising typically requires 5 to 20 iterations. Thus using $P = 5$ thresholding functions, Algorithm 4 slightly cheaper than 5 iterations of Algorithm 3. However, Algorithm 3 must typically be run many times to tune parameters. The true computational advantage of the SURELET-based algorithm 4 is that it must be run only once.

There is a key difference in perspective in denoising via transform sparse coding and SURELET denoising. Denoising by transform sparse coding aims to minimize the sparsification residual in the transform domain by solving (1), with the hope is that if W is well-conditioned this serves as a cheap and effective a proxy for minimizing the true quantity of interest, the image-domain MSE. In contrast, minimizing SURE (either directly or using SURELET) reduces the MSE directly in the image domain.

5. EXPERIMENTS

We call our combination of SURELET with a Learned Sparsifying Transform SURELET-LST . We compare this method against a suite

[2] A sufficient condition is that W is square and invertible [15].

[3] The link between shrinkage functions and their induced penalty functions has been investigated [24], but this is of little importance to us here.

Algorithm 4 SURELET Denoising

INPUT: $W \in \mathbb{R}^{M \times k}$; Noisy image $y \in \mathbb{R}^n$; Noise variance σ^2.

$Q \leftarrow$ DFT matrix
for $i = 1, \ldots, m, \quad j = 1, \ldots, P$ **do**
  $\alpha_i \leftarrow \| |Qw_i|^2 / (\sum_{j=1}^{M} |Qw_j|^2) \|_1$
  Calculate and store $F_{i,j}(y) = \mathcal{S}_i \psi_j (\mathcal{W}_i y)$
end for
Construct A and u according to (5), (6)
Solve $Ac = u$
$x \leftarrow \sum_{i=1}^{m} \sum_{j=1}^{P} c_{i,j} F_{i,j}(y)$

of competing methods. Software to reproduce these experiments will be made available[4]. To evaluate the benefit of the learning procedure, we compare against SURELET using the stationary wavelet transform (Haar wavelets with 5 levels) and DCT (8×8 filters). We call these methods SURELET-SWT and SURELET-DCT respectively. We also compare against Algorithm 3 using the 8×8 DCT (denoted IGD-DCT) and our pre-learned sparsifying transform (IGD-DCT) with a hard-thresholding nonlinearity. We test these two algorithms in an unrealistically favorable scenario by allowing an "oracle" to return the true MSE at each iteration, and we use this information to tune the parameters ν, λ, and the number of iterations for each noise level. These results indicate the upper bound of performance for hand-tuned parameters and hard-thresholding. Note that we could use SURE to estimate these parameters if utilized a differentiable thresholding function in place of hard thresholding.

We compare against an adaptive, joint learning/denoising method combined with IPD (Algorithm 2), which we denote IPD-AST [10]. Finally, we compare against BM3D, a popular denoising algorithm whose only parameter is the noise variance [28].

Our fixed sparsifying transform, used in SURELET-LST and IGD-LST , was pre-learned using the five 512×512 images shown in Figure 3a. Each image was normalized to unit ℓ_2 norm. We used 1000 iterations of the square transform learning algorithm with closed-form updates [10] with regularization parameters λ and ξ both set to 0.1. The 64×64 transform was learned using hard thresholding with threshold set to 5×10^{-4}. The algorithm was initialized with the DCT. The resulting transform filters are shown in Figure 3. All patch-methods utilized maximally overlapping 8×8 patches and periodic boundary conditions; DCT methods should be interpreted as cycle-spinning a block-DCT transform. Unlike the original SURELET work, we saw little denoising improvement using our learned transforms by using symmetric boundary conditions.

We evaluated performance at various noise levels using six popular test images and 10 noise realizations. Table 1 reports the mean reconstruction peak signal to noise ratio (PSNR) in dB, defined by $\mathrm{PSNR} = 20 \log_{10}(255/(512^2 \cdot \mathrm{MSE}))$.

These results demonstrate that the learned transforms denoise better than DCT and SWT. Our SURELET based methods perform on-par or better than the oracle versions, especially at the lower noise levels. That SURELET-LST occasionally outperforms IGD-LST suggests that the latter may be improved by replacing hard thresholding with a different nonlinearity. Importantly, we note that SURELET-LST typically outperforms IPD-AST even at low noise, when joint learning/denoising methods are typically expected to outperform universal methods [1].

[4] http://transformlearning.csl.illinois.edu/

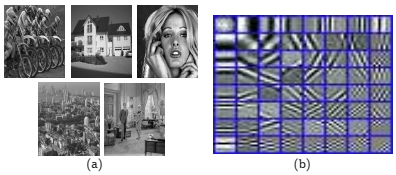

Fig. 3: (a): Training data for transform learning. (b): Learned 64×64 sparsifying transform, with rows reshaped into 8×8 filters.

6. CONCLUSIONS

We have limited our scope to removing Gaussian noise from an image using a fixed sparsifying transform using the SURELET algorithm. Experiments show that combining SURELET with a learned sparsifying transform outperforms SURELET using the SWT and DCT. Our method involves no parameter tuning during the denoising stage, yet performs as well as existing transform learning denoising algorithms that use carefully tuned parameters.

While SURELET-LST does not uniformly outperform BM3D, it represents a first step in parameter-free denoising with learned sparsifying transforms. It has been shown that transform learning with sophisticated, structured transforms provides uniformly better denoising performance than the transforms considered in this paper, but introduce more parameters to tune [11, 29]. Future work will investigate the use of SURE and SURELET for these transforms.

We have only applied our method to pre-learned transforms. This is a sub-optimal approach: the transform is learned using a fixed thresholding function, but denoises using the SURELET thresholds. We will address the use of SURE to optimize parameters in a joint learning/denoising framework. Finally, we anticipate using the iterative SURELET [30] and Projected Generalized SURE[25, 31] to provide automatic parameter tuning for general inverse problems with noise from the exponential family of distributions.

7. REFERENCES

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein, "K-SVD : An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[2] Julien Mairal, Francis Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning*. 2009, pp. 689–696, ACM.

[3] Ivana Tosic and Pascal Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, Mar 2011.

[4] Mehrdad Yaghoobi, Sangnam Nam, Remi Gribonval, and Mike E. Davies, "Constrained overcomplete analysis operator learning for cosparse signal modelling," *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2341–2355, May 2013.

[5] Mehrdad Yaghoobi, Sangnam Nam, Rémi Gribonval, and Michael E Davies, "Noise aware analysis operator learning for approximately cosparse signals," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5409–5412.

[6] Mehrdad Yaghoobi, Sangnam Nam, Rémi Gribonval, Mike E Davies, et al., "Analysis operator learning for overcomplete cosparse representations," in *European Signal Processing Conference (EUSIPCO'11)*, 2011.

Table 1: Reconstruction PSNR averaged over 10 noise realizations.

σ	10	20	30	10	20	30
Input PSNR	28.13	22.11	18.59	28.13	22.11	18.59
Method	**baboon**			**peppers512**		
BM3D	30.47	26.45	24.40	34.75	32.51	31.01
SURELET-SWT	30.03	25.84	23.80	34.14	31.49	29.72
SURELET-DCT	30.44	26.31	24.22	34.55	31.70	29.91
IGD-DCT	30.40	26.31	24.26	34.57	32.07	30.50
IPD-AST	30.43	26.35	24.22	34.58	31.90	30.13
IGD-LST	30.49	26.40	24.31	34.67	32.25	30.68
SURELET-LST	30.52	26.43	24.31	34.68	31.89	30.07
Method	**barbara**			**man**		
BM3D	34.95	31.74	29.78	33.94	30.56	28.81
SURELET-SWT	32.69	28.48	26.17	33.16	29.75	28.02
SURELET-DCT	34.22	30.37	28.18	33.47	29.92	28.05
IGD-DCT	33.98	30.16	27.97	33.53	30.10	28.33
IPD-AST	34.34	30.52	28.27	33.64	30.00	28.09
IGD-LST	33.98	30.00	27.71	33.77	30.24	28.39
SURELET-LST	34.30	30.51	28.24	33.74	30.16	28.20
Method	**boat**			**sailboat**		
BM3D	33.90	30.84	29.04	32.51	29.52	27.84
SURELET-SWT	33.26	29.89	28.07	32.11	28.89	27.02
SURELET-DCT	33.64	30.24	28.28	32.56	29.26	27.41
IGD-DCT	33.68	30.45	28.61	32.41	29.36	27.66
IPD-AST	33.63	30.24	28.28	32.51	29.26	27.40
IGD-LST	33.76	30.51	28.65	32.48	29.42	27.67
SURELET-LST	33.79	30.38	28.39	32.68	29.36	27.48

[7] Gabriel Peyré, Jalal Fadili, et al., "Learning analysis sparsity priors," *Sampta'11*, 2011.

[8] Simon Hawe, Martin Kleinsteuber, and Klaus Diepold, "Analysis operator learning and its application to image reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2138–2150, June 2013.

[9] Yunjin Chen, Rene Ranftl, and Thomas Pock, "Insights into analysis operator learning: From patch-based sparse models to higher order MRFs," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1060?1072, Mar 2014.

[10] S. Ravishankar and Y. Bresler, "Sparsifying transform learning with efficient optimal updates and convergence guarantees," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2389–2404, May 2015.

[11] Bihan Wen, Saiprasad Ravishankar, and Yoram Bresler, "Structured overcomplete sparsifying transform learning with convergence guarantees and applications," *International Journal of Computer Vision*, Oct 2014.

[12] Saiprasad Ravishankar and Yoram Bresler, "Learning sparsifying transforms," *IEEE Trans. Signal Process.*, vol. 61, no. 5, pp. 1072–1086, 2013.

[13] Michael Elad and Michal Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–45, Dec. 2006.

[14] Saiprasad Ravishankar and Yoram Bresler, "Sparsifying transform learning for compressed sensing MRI," in *International Symposium on Biomedical Imaging*, 2013.

[15] Luke Pfister and Yoram Bresler, "Learning sparsifying filter banks," in *Proc. SPIE Wavelets & Sparsity XVI*. August 2015, vol. 9597, SPIE.

[16] Qiong Xu, HY Yu, and XQ Mou, "Low-dose x-ray CT reconstruction via dictionary learning," *IEEE Trans. Med. Imag.*, vol. 31, no. 9, pp. 1682–1697, Sept. 2012.

[17] Hstau Y. Liao and Guillermo Sapiro, "Sparse representations for limited data tomography," in *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. May 2008, pp. 1375–1378, IEEE.

[18] J. Shtok, M. Elad, and M. Zibulevsky, "Sparsity-based sinogram denoising for low-dose computed tomography," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. May 2011, number 1, pp. 569–572, IEEE.

[19] Luke Pfister and Yoram Bresler, "Model-based iterative tomographic reconstruction with adaptive sparsifying transforms," in *Proc. SPIE Computational Imaging XII*, Charles A. Bouman and Ken D Sauer, Eds. Mar 2014, pp. 90200H–90200H–11, SPIE.

[20] Luke. Pfister and Yoram. Bresler, "Tomographic reconstruction with adaptive sparsifying transforms," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 6914–6918.

[21] Luke Pfister and Yoram Bresler, "Adaptive sparsifying transforms for iterative tomographic reconstruction," in *International Conference on Image Formation in X-Ray Computed Tomography*, 2014.

[22] T. Blu and F. Luisier, "The SURE-LET approach to image denoising," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2778–2786, 2007.

[23] Charles M. Stein, "Estimation of the mean of a multivariate normal distribution," *Ann. Statist.*, vol. 9, no. 6, pp. 1135–1151, nov 1981.

[24] R. Chartrand, "Shrinkage mappings and their induced penalty functions," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 1026–1029.

[25] R. Giryes, M. Elad, and Y.C. Eldar, "The projected GSURE for automatic parameter tuning in iterative shrinkage methods," *Applied and Computational Harmonic Analysis*, vol. 30, no. 3, pp. 407–422, 2011.

[26] S. Ramani, T. Blu, and M. Unser, "Monte-carlo sure: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1540–1554, 2008.

[27] M. Raphan and E.P. Simoncelli, "Optimal denoising in redundant representations," *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1342–1352, 2008.

[28] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.

[29] Bihan Wen, Yanjun Li, and Yoram Bresler, "When sparsity meets low-rankness: transform learning with non-local low-rank constraint for image restoration," in *Submitted to Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017.

[30] Hanjie Pan and T. Blu, "An iterative linear expansion of thresholds for -based image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3715–3728, 2013.

[31] Y.C. Eldar, "Generalized sure for exponential families: Applications to regularization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 471–481, 2009.