Learning Convolutional Proximal Filters

Ulugbek S. Kamilov, Hassan Mansour, and Dehong Liu
Mitsubishi Electric Research Laboratories (MERL)
201 Broadway, Cambridge, MA, 02139, USA
Email: kamilov@merl.com, mansour@merl.com, and liudh@merl.com

***Abstract*—In the past decade, sparsity-driven methods have led to substantial improvements in the capabilities of numerous imaging systems. While traditionally such methods relied on analytical models of sparsity, such as total variation (TV) or wavelet regularization, recent methods are increasingly based on data-driven models such as dictionary-learning or convolutional neural networks (CNN). In this work, we propose a new trainable model based on the proximal operator for TV. By interpreting the popular fast iterative shrinkage/thresholding algorithm (FISTA) as a CNN, we train the filters of the algorithm to minimize the error over a training data-set. Experiments on image denoising show that by training the filters, one can substantially boost the performance of the algorithm and make it competitive with other state-of-the-art methods.**

I. Introduction

We consider an imaging inverse problem $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}$, where the goal is to recover the unknown image $\mathbf{x} \in \mathbb{R}^N$ from the noisy measurements $\mathbf{y} \in \mathbb{R}^M$. The matrix $\mathbf{H} \in \mathbb{R}^{M\times N}$ is known and models the response of the acquisition device, while the vector $\mathbf{e} \in \mathbb{R}^M$ represents the unknown noise in the measurements.

Practical imaging inverse problems are often ill-posed [1]. A standard approach for solving such problems is the regularized least-squares estimator

$$\widehat{\mathbf{x}} = \mathop{\arg\min}_{\mathbf{x}\in\mathbb{R}^N} \left\{ \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_{\ell_2}^2 + \mathcal{R}(\mathbf{x}) \right\}, \quad (1)$$

where $\mathcal{R}$ is a regularizer promoting solutions with desirable properties. One of the most popular regularizers for images is the total variation (TV) [2], defined as $\mathcal{R}(\mathbf{x}) \triangleq \tau\|\mathbf{D}\mathbf{x}\|_{\ell_1}$, where $\tau > 0$ is parameter that controls the strength of the regularization, and $\mathbf{D} : \mathbb{R}^N \to \mathbb{R}^{N\times K}$ is the discrete gradient operator. The gradient can be represented with K separate filters, $\mathbf{D} \triangleq (\mathbf{D}_1, \dots, \mathbf{D}_K)$, computing finite-differences along each dimension of the image.

Two common methods for solving the TV regularized problem (1) are fast iterative shrinkage/thresholding algorithm (FISTA) [3] and alternating direction method of multipliers (ADMM) [4]. These algorithms are among the methods of choice for solving large-scale imaging problems due to their ability to handle the non-smoothness of TV and their low-computational complexity. Both FISTA and ADMM typically combine the operations with the measurement matrix with applications of the proximal operator

$$\mathrm{prox}_{\tau\mathcal{R}}(\mathbf{y}) \triangleq \mathop{\arg\min}_{\mathbf{x}\in\mathbb{R}^N} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_{\ell_2}^2 + \tau\mathcal{R}(\mathbf{x}) \right\}. \quad (2)$$

Beck and Teboulle [3] have proposed an efficient dual domain FISTA for computing TV proximal

$$\mathbf{s}^t = \mathbf{g}^{t-1} + ((q_{t-1} - 1)/q_t)\mathbf{g}^{t-2} \quad (3a)$$
$$\mathbf{z}^t = \mathbf{s}^t - \gamma\tau\mathbf{D}(\tau\mathbf{D}^{\mathrm{T}}\mathbf{s} - \mathbf{y}) \quad (3b)$$
$$\mathbf{g}^t = \mathcal{P}_\infty(\mathbf{z}^t), \quad (3c)$$

with $q_0 = 1$ and $\mathbf{g}^0 = \mathbf{g}^{-1} = \mathbf{g}_{\text{init}} \in \mathbb{R}^{N\times K}$. Here, $\mathcal{P}_\infty$ denotes a component-wise projection operator onto a unit ℓ_∞-norm ball, $\gamma = 1/L$ with $L = \tau^2\lambda_{\max}(\mathbf{D}^{\mathrm{T}}\mathbf{D})$ is a step-size, and $\{q_t\}_{t\in\mathbb{N}}$ are relaxation parameters. For a fixed $q_t = 1$, the guaranteed global convergence speed of the algorithm is $O(1/t)$; however, the choice $q_t = \frac{1}{2}(1 + \sqrt{1 + 4q_{t-1}})$ leads to a faster $O(1/t^2)$ convergence [3]. The final denoised image after T iterations of (3) is obtained as $\mathbf{x}^T = \mathbf{y} - \tau\mathbf{D}^{\mathrm{T}}\mathbf{g}^T$.

II. Main Results

Our goal is to obtain a trainable variant of (3) by replacing the finite-difference filters of TV with K adaptable, iteration-dependent filters. The corresponding algorithm, illustrated in Fig. 1, can be interpreted as a convolutional neural network (CNN) of a particular structure with $T \times K$ filters $\mathbf{D}_t \triangleq (\mathbf{D}_{t1}, \dots, \mathbf{D}_{tK})$ that are learned from a set of L training examples $\{\mathbf{x}_\ell, \mathbf{y}_\ell\}_{\ell\in[1,\dots,L]}$. The filters can be optimized by minimizing the error

$$\widehat{\boldsymbol{\theta}} = \mathop{\arg\min}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left\{ \frac{1}{L}\sum_{\ell=1}^{L} \mathcal{E}_\ell(\boldsymbol{\theta}) \right\} \text{ with } \mathcal{E}(\boldsymbol{\theta}) \triangleq \|\mathbf{x} - \widehat{\mathbf{x}}(\mathbf{y};\boldsymbol{\theta})\|_{\ell_2}^2 \quad (4)$$

over the training set, where $\boldsymbol{\theta} = \{\mathbf{D}_t\}_{t\in[1,\dots,T]} \in \boldsymbol{\Theta}$ denotes the set of desirable filters. For the problem of image denoising, end-to-end optimization can be performed with the error backpropagation algorithm [5] that produces

$$[\nabla\mathcal{E}_\ell(\boldsymbol{\theta})]_{tk} = \begin{cases} \mathbf{q}_{tk} + \tau(\mathbf{g}_k^T \bullet (\mathbf{x} - \widehat{\mathbf{x}})) & \text{for } t = T \\ \mathbf{q}_{tk} & \text{for } 1 \leq t \leq T-1, \end{cases}$$

using the following iteration for $t = T, T-1, \dots, 1$,

$$\mathbf{v}^{t-1} = \mathrm{diag}\left(\mathcal{P}'_\infty(\mathbf{z}^t)\right)\mathbf{r}^t \quad (5a)$$
$$\mathbf{b}^{t-1} = \mathbf{v}^{t-1} - \gamma\tau^2\mathbf{D}_t\mathbf{D}_t^{\mathrm{T}}\mathbf{v}^{t-1} \quad (5b)$$
$$\mathbf{r}^{t-1} = \mu_t\mathbf{b}^{t-1} + (1 - \mu_{t+1})\mathbf{b}^t \quad (5c)$$
$$\mathbf{q}_{tk} = \gamma\tau[(\mathbf{v}_k^{t-1} \bullet (\mathbf{y} - \tau\mathbf{D}_t^{\mathrm{T}}\mathbf{s}^t)) - \tau(\mathbf{s}_k^t \bullet (\mathbf{D}_t^{\mathrm{T}}\mathbf{v}^{t-1}))] \quad (5d)$$

where $\bullet$ denotes filtering, $\mu_t = 1 - (1 - q_{t-1})/q_t$, $\mathbf{b}^T = 0$, and $\mathbf{r}^T = \tau\mathbf{D}_T(\mathbf{x} - \widehat{\mathbf{x}})$. The parameters are update iteratively with the standard stochastic gradient method as $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha\nabla\mathcal{E}_\ell(\boldsymbol{\theta})$.

We applied our method to image denoising by training $T = 10$ iterations of the algorithm with $K = 9$ iteration dependent kernels of size 6×6 pixels. For taining, we used 400 images from Berkeley dataset [6] cropped to 192×192 pixels. We evaluated the algorithm on 68 separate test images from the dataset and compared the results with three popular denoising algorithms (see Table I and Fig. 2–3). Our basic MATLAB implementation takes 0.69 and 3.27 seconds on images of 256×256 and 512×512 pixels, respectively, on an Apple iMac with a 4 GHz Intel Core i7 processor. We observe that our simple extension of TV significantly boosts the performance of the algorithm and makes it competitive with state-of-the-art denoising algorithms. The algorithm can be easily incorporated into FISTA and ADMM for solving more general inverse problems. Future work will address such extensions and further improve the performance by code optimization and considering more kernels. More generally, our work contributes to the recent efforts to boost the performance of imaging algorithms by incorporating latest ideas from deep learning [7]–[13].

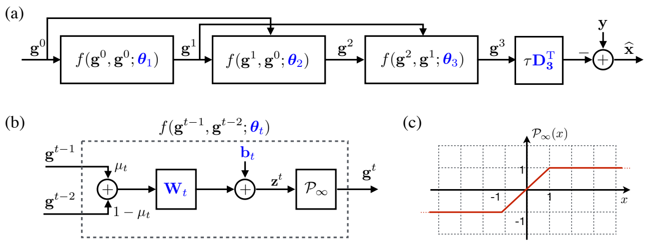

Fig. 1. A schematic representation of the trainable variant of (3) with adaptable parameters, $\mathbf{W}_t \triangleq \mathbf{I} - \gamma\tau^2\mathbf{D}_t\mathbf{D}_t^{\mathrm{T}}$ and $\mathbf{b}_t \triangleq \gamma\tau\mathbf{D}_t\mathbf{y}$, marked in blue. (a) The algorithm for $T = 3$ iterations with $\boldsymbol{\theta}_t \triangleq \mathbf{D}_t$. (b) The schematic view of a single iteration where $\mu_t = 1 - (1 - q_{t-1})/q_t$. (c) The plot of the scalar nonlinearity $\mathcal{P}_\infty$.

TABLE I
AVERAGE PSNR ON 68 IMAGES FROM THE BERKELEY DATASET.

Noise level	Proposed	TV [3]	K-SVD [14]	BM3D [15]
$\sigma = 15$	30.77 dB	29.91 dB	30.89 dB	31.08 dB
$\sigma = 30$	27.53 dB	26.69 dB	27.44 dB	27.76 dB
$\sigma = 45$	25.80 dB	24.99 dB	25.61 dB	25.98 dB

REFERENCES

[1] A. Ribés and F. Schmitt, "Linear inverse problems in imaging," *IEEE Signal Process. Mag.*, vol. 25, no. 4, pp. 84–99, July 2008.

[2] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1–4, pp. 259–268, November 1992.

[3] A. Beck and M. Teboulle, "Fast gradient-based algorithm for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, November 2009.

[4] M. V. Afonso, J. M.Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, September 2010.

[5] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, 1995.

[6] 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. IEEE Int. Conf. Comp. Vis. (ICCV)*, Vancouver, Canada, July 7-14, 2001, pp. 416–423.

[7] A. Barbu, "Training an active random field for real-time image denoising," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2451–2462, November 2009.

[8] K. Gregor and Y. LeCun, "Learning fast approximation of sparse coding," in *Proc. 27th Int. Conf. Machine Learning (ICML)*, Haifa, Israel, June 21-24, 2010, pp. 399–406.

[9] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, June 23-28, 2014, pp. 2774–2781.

[10] Y. Chen, W. Yu, and T. Pock, "On learning optimized reaction diffuction processes for effective image restoration," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 8-10, 2015, pp. 5261–5269.

[11] U. S. Kamilov and H. Mansour, "Learning optimal nonlinearities for iterative thresholding algorithms," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 747–751, May 2016.

[12] M. Borgerding and P. Schniter, "Onsanger-corrected deep networks for sparse linear inverse problems," 2016, arXiv:1612.01183 [cs.IT].

[13] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," 2016, arXiv:1611.03679 [cs.CV].

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

[15] 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. 16, pp. 2080–2095, August 2007.

Fig. 2. Comparison of four denoising algorithms on *Firefighter* (image no. 285079) from the Berkeley dataset at noise level $\sigma = 30$. The values in the bottom-left corner correspond to PSNR in dB.

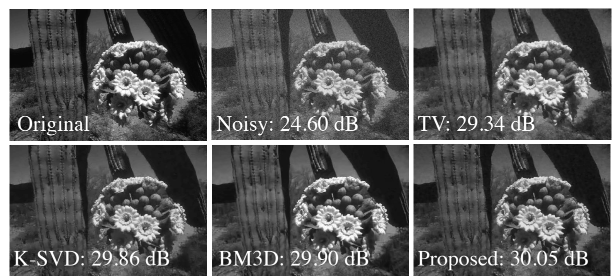

Fig. 3. Comparison of four denoising algorithms on *Desert* (image no. 19021) from the Berkeley dataset at noise level $\sigma = 15$. The values in the bottom-left corner correspond to PSNR in dB.

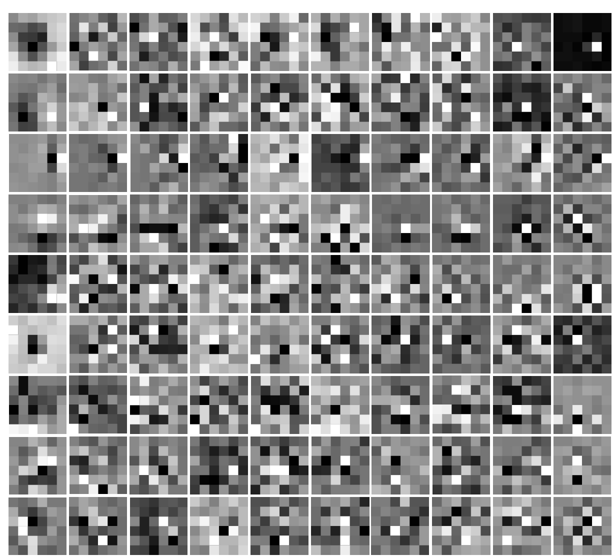

Fig. 4. Visual illustration of all the filters learned for $\sigma = 30$. Leftmost column shows all the filters of the first iteration, while rightmost column of the last iteration. Note the close resemblance of some filters to various differential operators.