

Deeply Aggregated Alternating Minimization for Image Restoration

Youngjung Kim¹, Hyungjoo Jung¹, Dongbo Min², and Kwanghoon Sohn¹

¹Yonsei University

²Chungnam National University

Abstract

Regularization-based image restoration has remained an active research topic in computer vision and image processing. It often leverages a guidance signal captured in different fields as an additional cue. In this work, we present a general framework for image restoration, called deeply aggregated alternating minimization (DeepAM). We propose to train deep neural network to advance two of the steps in the conventional AM algorithm: proximal mapping and β -continuation. Both steps are learned from a large dataset in an end-to-end manner. The proposed framework enables the convolutional neural networks (CNNs) to operate as a prior or regularizer in the AM algorithm. We show that our learned regularizer via deep aggregation outperforms the recent data-driven approaches as well as the nonlocal-based methods. The flexibility and effectiveness of our framework are demonstrated in several image restoration tasks, including single image denoising, RGB-NIR restoration, and depth super-resolution.

1. Introduction

Image restoration is a process of reconstructing a clean image from a degraded observation. The observed data is assumed to be related to the ideal image through a forward imaging model that accounts for noise, blurring, and sampling. However, a simple modeling only with the observed data is insufficient for an effective restoration, and thus a priori constraint about the solution is commonly used. To this end, the image restoration is usually formulated as an energy minimization problem with an explicit regularization function (or regularizer). Recent work on joint restoration leverages a guidance signal, captured from different devices, as an additional cue to regularize the restoration process. These approaches have been successfully applied to various applications including joint upsampling [11], cross-field noise reduction [32], dehazing [31], and intrinsic im-

age decomposition [8].

The regularization-based image restoration involves the minimization of non-convex and non-smooth energy functionals for yielding high-quality restored results. Solving such functionals typically requires a huge amount of iterations, and thus an efficient optimization is preferable, especially in some applications the runtime is crucial. One of the most popular optimization methods is the alternating minimization (AM) algorithm [34] that introduces auxiliary variables. The energy functional is decomposed into a series of subproblems that is relatively simple to optimize, and the minimum with respect to each of the variables is then computed. For the image restoration, the AM algorithm has been widely adopted with various regularization functions, e.g., total variation [34], L_0 norm [36], and L_p norm (hyper-Laplacian) [16]. It is worth noting that these functions are all handcrafted models. The hyper-Laplacian of image gradients [16] reflects the statistical property of natural images relatively well, but the restoration quality of gradient-based regularization methods using the handcrafted model is far from that of the state-of-the-art approaches [9, 30]. In general, it is non-trivial to design an optimal regularization function for a specific image restoration problem.

Over the past few years, several attempts have been made to overcome the limitation of handcrafted regularizer by learning the image restoration model from a large-scale training data [9, 30, 39]. In this work, we propose a novel method for image restoration that effectively uses a data-driven approach in the energy minimization framework, called *deeply aggregated alternating minimization* (DeepAM). Contrary to existing data-driven approaches that just produce the restoration results from the convolutional neural networks (CNNs), we design the CNNs to implicitly learn the regularizer of the AM algorithm. Since the CNNs are fully integrated into the AM procedure, the whole networks can be learned simultaneously in an end-to-end manner. We show that our simple model learned from the deep aggregation achieves better results than the recent data-driven approaches [9, 17, 30] as well as the state-of-the-

art nonlocal-based methods [10, 12].

Our main contributions can be summarized as follows:

- We design the CNNs to learn the regularizer of the AM algorithm, and train the whole networks in an end-to-end manner.
- We introduce the aggregated (or multivariate) mapping in the AM algorithm, which leads to a better restoration model than the conventional point-wise proximal mapping.
- We extend the proposed method to joint restoration tasks. It has broad applicability to a variety of restoration problems, including image denoising, RGB/NIR restoration, and depth super-resolution.

2. Related Work

Regularization-based image restoration Here, we provide a brief review of the regularization-based image restoration. The total variation (TV) [34] has been widely used in several restoration problems thanks to its convexity and edge-preserving capability. Other regularization functions such as total generalized variation (TGV) [4] and L_p norm [16] have also been employed to penalize an image that does not exhibit desired properties. Beyond these handcrafted models, several approaches have been attempted to learn the regularization model from training data [9, 30]. Schmidt *et al.* [30] proposed a cascade of shrinkage fields (CSF) using learned Gaussian RBF kernels. In [9], a nonlinear diffusion-reaction process was modeled by using parameterized linear filters and regularization functions. Joint restoration methods using a guidance image captured under different configurations have also been studied [3, 11, 17, 31]. In [3], an RGB image captured in dim light was restored using flash and non-flash pairs of the same scene. In [11, 15], RGB images was used to assist the regularization process of a low-resolution depth map. Shen *et al.* [31] proposed to use dark-flashed NIR images for the restoration of noisy RGB image. Li *et al.* used the CNNs to selectively transfer salient structures that are consistent in both guidance and target images [17].

Use of energy minimization models in deep network

The CNNs lack imposing the regularity constraint on adjacent similar pixels, often resulting in poor boundary localization and spurious regions. To deal with these issues, the integration of energy minimization models into CNNs has received great attention [24–26, 38]. Ranftl *et al.* [24] defined the unary and pairwise terms of Markov Random Fields (MRFs) using the outputs of CNNs, and trained network parameters using the bilevel optimization. Similarly, the mean field approximation for fully connected conditional random fields (CRFs) was modeled as recurrent neural networks (RNNs) [38]. A nonlocal Huber regularization

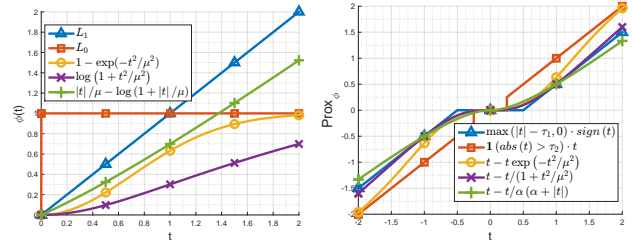


Figure 1: Illustrations of the regularization function Φ (left) and the corresponding proximal mapping (right). The main purpose of this mapping is to remove Du^k with a small magnitude, since they are assumed to be caused by noise. Instead of such handcrafted regularizers, we implicitly parameterize the regularization function using the deep aggregation, leading to a better restoration algorithm.

was combined with CNNs for a high quality depth restoration [25]. Riegler *et al.* [26] integrated anisotropic TGV into the top of deep networks. They also formulated the bilevel optimization problem and trained the network in an end-to-end manner by unrolling the TGV minimization. Note that the bilevel optimization problem is solvable only when the energy minimization model is convex and is twice differentiable [24]. The aforementioned methods try to integrate handcrafted regularization models into top of the CNNs. In contrast, we design the CNNs to parameterize the regularization process in the AM algorithm.

3. Background and Motivation

The regularization-based image reconstruction is a powerful framework for solving a variety of inverse problems in computational imaging. The method typically involves formulating a data term for the degraded observation and a regularization term for the image to be reconstructed. An output image is then computed by minimizing an objective function that balances these two terms. Given an observed image f and a balancing parameter λ , we solve the corresponding optimization problem¹:

$$\arg \min_u \frac{\lambda}{2} \|u - f\|^2 + \Phi(Du). \quad (1)$$

Du denotes the $[D_x u, D_y u]$, where D_x (or D_y) is a discrete implementation of x -derivative (or y -derivative) of the image. Φ is a regularization function that enforces the output image u to meet desired statistical properties. The unconstrained optimization problem of (1) can be solved using numerous standard algorithms. In this paper, we focus on the additive form of alternating minimization (AM)

¹For the super-resolution, we treat f as the bilinearly upsampled image from the low-resolution input.

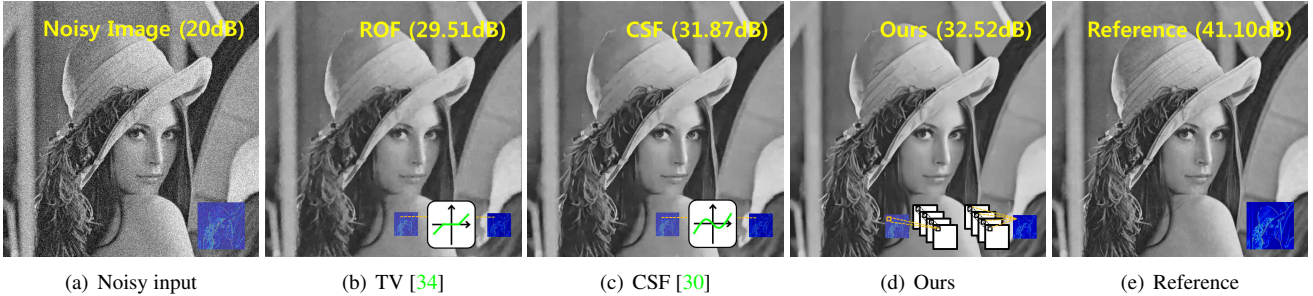


Figure 2: Examples of single image denoising: (a) input image, (b) TV [34], (c) CSF [30], and (d) ours. (e) is obtained after one step of the AM iteration using Du^* with $\lambda = 5$, where u^* is a noise-free image. Our deeply aggregated AM outperforms existing point-wise mapping operators.

method [34], which is the ad-hoc for a variety of problems in the form of (1).

3.1. Alternating Minimization

The idea of AM method is to decouple the data and regularization terms by introducing a new variable v and to reformulate (1) as the following constrained optimization problem:

$$\min_{u,v} \frac{\lambda}{2} \|u - f\|^2 + \Phi(v), \text{ subject to } v = Du. \quad (2)$$

We solve (2) by using the penalty technique [34], yielding the augmented objective function.

$$\min_{u,v} \frac{\lambda}{2} \|u - f\|^2 + \Phi(v) + \frac{\beta}{2} \|Du - v\|^2, \quad (3)$$

where β is the penalty parameter. The AM algorithm consists of repeatedly performing the following steps until convergence.

$$\begin{aligned} v^{k+1} &= \arg \min_v \Phi(v) + \frac{\beta^k}{2} \|Du^k - v\|^2, \\ u^{k+1} &= \arg \min_u \frac{\lambda}{2} \|u - f\|^2 + \frac{\beta^k}{2} \|Du - v^{k+1}\|^2, \\ \beta^{k+1} &= \alpha \beta^k, \end{aligned} \quad (4)$$

where $\alpha > 1$ is a continuation parameter. When β is large enough, the variable v approaches Du , and thus (3) converges to the original formulation (1).

3.2. Motivation

Minimizing the first step in (4) varies depending on the choices of the regularization function Φ and β . This step can be regarded as the proximal mapping [22] of Du^k associated with Φ . When Φ is the sum of L_1 or L_0 norm, it amounts to soft or hard thresholding operators (see Fig. 1

and [22] for various examples of this relation). Such mapping operators may not unveil the full potential of the optimization method of (4), since Φ and β are chosen manually. Furthermore, the mapping operator is performed for each pixel individually, disregarding spatial correlation with neighboring pixels.

Building upon this observation, we propose the new approach in which the regularization function Φ and the penalty parameter β are learned from a large-scale training dataset. Different from the point-wise proximal mapping based on the handcrafted regularizer, the proposed method learns and aggregates the mapping of Du^k through CNNs.

4. Proposed Method

In this section, we first introduce the DeepAM for a single image restoration, and then extend it to joint restoration tasks. In the following, the subscripts i and j denote the location of a pixel (in a vector form).

4.1. Deeply Aggregated AM

We begin with some intuition about why our learned and aggregated mapping is crucial to the AM algorithm. The first step in (4) maps Du^k with a small magnitude into zero since it is assumed that they are caused by noise, not an original signal. Traditionally, this mapping step has been applied in a point-wise manner, not to mention whether it is learned or not. With $\Phi(v) = \sum_i \phi(v_i)$, Schmidt *et al.* [30] modeled the point-wise mapping function as Gaussian RBF kernels, and learned their mixture coefficients². Contrarily, we do not presume any property of Φ . We instead train the multivariate mapping process ($Du^k \rightarrow v^{k+1}$) associated with Φ and β by making use of the CNNs. Figure 2 shows the denoising examples of TV [34], CSF [30], and ours. Our method outperforms other methods using the point-wise mapping based on handcrafted model (Fig. 2(b))

²When $\Phi(v) = \sum_i \phi(v_i)$, the first step in (4) is separable with respect to each v_i . Thus, it can be modeled by point-wise operation.

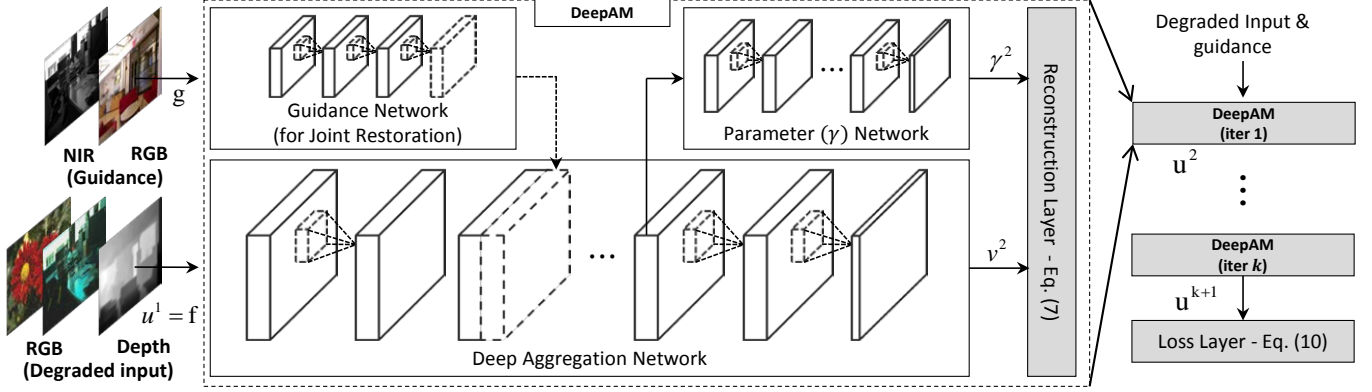


Figure 3: One iteration of our model consists of four major components: deep aggregation network, guidance network, γ -parameter network, and reconstruction layer. The spatially varying γ is estimated by exploiting features from intermediate layers of the deep aggregation network. All of these sub-networks are cascaded by iterating (5) and (6), and the final output is then entered into the loss layer.

or learned model (Fig. 2(c)) (see the insets).

We reformulate the original AM iterations in (4) with the following steps³.

$$(v^{k+1}, \gamma^{k+1}) \Leftarrow \mathcal{D}_{CNN}(u^k, w_u^k), \quad (5)$$

$$u^{k+1} = \arg \min_u \|\Gamma^{k+1}(u - f)\|^2 + \|Du - v^{k+1}\|^2, \quad (6)$$

where $\mathcal{D}_{CNN}(\cdot, w_u^k)$ denotes a convolutional network parameterized by w_u^k and $\Gamma^{k+1} = \text{diag}(\gamma^{k+1})$. Note that β is completely absorbed into the CNNs, and fused with the balancing parameter γ (which will also be learned). v^{k+1} is estimated by deeply aggregating u^k through CNNs. This formulation allows us to turn the optimization procedure in (1) into a cascaded neural network architecture, which can be learned by the standard back-propagation algorithm [20].

The solution of (6) satisfies the following linear system:

$$Lu^{k+1} = \Gamma^{k+1}f + D^T v^{k+1}, \quad (7)$$

where the Laplacian matrix $L = (\Gamma^{k+1} + D^T D)$. It can be seen that (7) plays a role of naturally imposing the spatial and appearance consistency on the intermediate output image u^{k+1} using a kernel matrix $A_{ij} = L_{ij}^{-1}$ [38]. The linear system of (7) becomes the part of deep neural network (see Fig. 3). When γ is a constant, the block Toeplitz matrix L is diagonalizable with the fast Fourier transform (FFT). However, in our framework, the direct application of FFT is not feasible since γ is spatially varying for the adaptive regularization. Fortunately, the matrix L is still sparse and positive semi-definite as the simple gradient operator D is used. We adopt the preconditioned conjugate gradient (PCG) method

³The gradient operator D is absorbed into the CNNs.

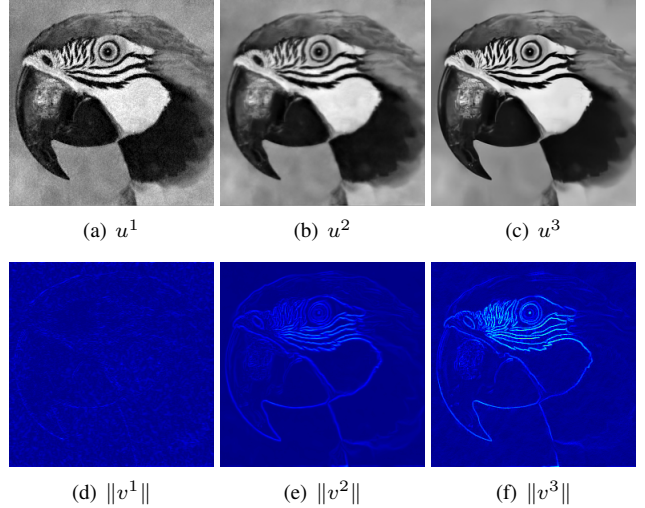


Figure 4: The denoising results obtained by our DeepAM (trained with $K = 3$ iterations in Fig 3). See the text for details.

to solve the linear system of (7). The incomplete Cholesky factorization [1] is used for computing the preconditioner.

Very recently, Chan *et al.* [7] replaced the proximal mapping in (4) with an off-the-shelf image denoising algorithm \mathcal{D}_σ , e.g., nonlocal means [5], as follows:

$$v^{k+1} \Leftarrow \mathcal{D}_\sigma(Du^{k+1}). \quad (8)$$

Although this is conceptually similar to our aggregation approach⁴, the operator \mathcal{D}_σ in [7] still relies on the handcrafted model. Figure 3 shows the proposed learning model for image restoration tasks. The DeepAM, consisting of deep ag-

⁴Aggregation using neighboring pixels are commonly used in state-of-the-arts denoising methods.

gregation network, γ -parameter network, guidance network (which will be detailed in next section), and reconstruction layer, is iterated K times, followed by the loss layer.

Figure 4 shows the denoising result of our method. Here, it is trained with three passes of DeepAM. The input image is corrupted by Gaussian noise with standard deviation $\sigma = 25$. We can see that as iteration proceeds, the high-quality restoration results are produced. The trained networks in the first and second iterations remove the noise, but intermediate results are over smoothed (Figs. 4(a) and (b)). The high-frequency information is then recovered in the last network (Fig. 4(c)). To analyze this behavior, let us date back to the existing soft-thresholding operator, $v_i^{k+1} = \max\{|Du^k|_i - 1/\beta^k, 0\} \text{sign}(Du)_i$ in [34]. The conventional AM method sets β as a small constant and increases it during iterations. When β is small, the range of v is shrunk, penalizing large gradient magnitudes. The high-frequency details of an image are recovered as β increases. Interestingly, the DeepAM shows very similar behavior (Figs. 4(d)-(f)), but outperforms the existing methods thanks to the aggregated mapping through the CNNs, as will be validated in experiments.

4.2. Extension to Joint Restoration

In this section, we extend the proposed method to joint restoration tasks. The basic idea of joint restoration is to provide structural guidance, assuming structural correlation between different kinds of feature maps, e.g., depth/RGB and NIR/RGB. Such a constraint has been imposed on the conventional mapping operator by considering structures of both input and guidance images [15]. Similarly, one can modify the deeply aggregated mapping of (5) as follows:

$$(v^{k+1}, \gamma^{k+1}) \leftarrow \mathcal{D}_{CNN}((u^k \otimes g), w_u^k), \quad (9)$$

where g is a guidance image and \otimes denotes a concatenation operator. However, we find such early concatenation to be less effective since the guidance image mixes heterogeneous data. This coincides with the observation in the literature of multispectral pedestrian detection [18]. Instead, we adopt the halfway concatenation similar to [17, 18]. Another sub-network $\mathcal{D}_{CNN}(g, w_g^k)$ is introduced to extract the effective representation of the guidance image, and it is then combined with intermediate features of $\mathcal{D}_{CNN}(u^k, w_u^k)$ (see Fig. 3).

4.3. Learning Deeply Aggregated AM

In this section, we will explain the network architecture and training method using standard back-propagation algorithm. Our code will be publicly available later.

Network architecture One iteration of the proposed DeepAM consists of four major parts: deep aggregation network, γ -parameter network, guidance network (for joint

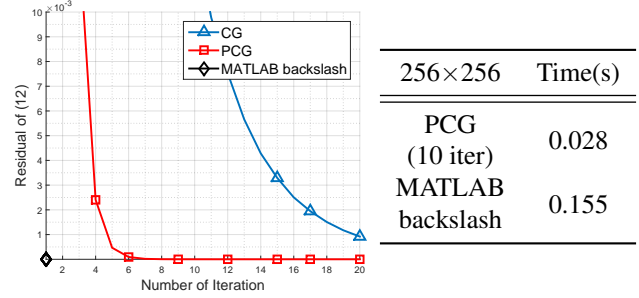


Figure 5: Figure in left shows the convergence of the PCG solver. A small number of PCG iterations are enough for the back-propagation. The results of the MATLAB backslash is plotted in the origin. The table in right compares the runtime of PCG with 10 iterations and direct MATLAB solver.

restoration), and reconstruction layer, as shown in Fig. 3. The deep aggregation network consists of 10 convolutional layers with 3×3 filters (a receptive field is of 21×21). Each hidden layer of the network has 64 feature maps. Since v contains both positive and negative values, the rectified linear unit (ReLU) is not used for the last layer. The input distributions of all convolutional layers are normalized to the standard Gaussian distribution [21]. The output channel of the deep aggregation network is 2 for the horizontal and vertical gradients. We also extract the spatially varying γ by exploiting features from the eighth convolutional layer of the deep aggregation network. The ReLU is used for ensuring the positive values of γ .

For joint image restoration, the guidance network consists of 3 convolutional layers, where the filters operate on 3×3 spatial region. It takes the guidance image g as an input, and extracts a feature map which is then concatenated with the third convolutional layer of the deep aggregation network. There are no parameters to be learned in the reconstruction layer.

Training The DeepAM is learned via standard back-propagation algorithm [20]. We do not require the complicated bilevel formulation [24, 26]. Given M training image pairs $\{f^{(p)}, g^{(p)}, t^{(p)}\}_{p=1}^M$, we learn the network parameters by minimizing the L_1 loss function.

$$\mathcal{L} = \frac{1}{M} \sum_p \|u^{(p)} - t^{(p)}\|_1, \quad (10)$$

where $t^{(p)}$ and $u^{(p)}$ denote the ground truth image and the output of the last reconstruction layer in (7), respectively. It is known that L_1 loss in deep networks reduces splotchy artifacts and outperforms L_2 loss for pixel-level prediction tasks [37]. We use the stochastic gradient descent (SGD) to minimize the loss function of (10). The derivative for the

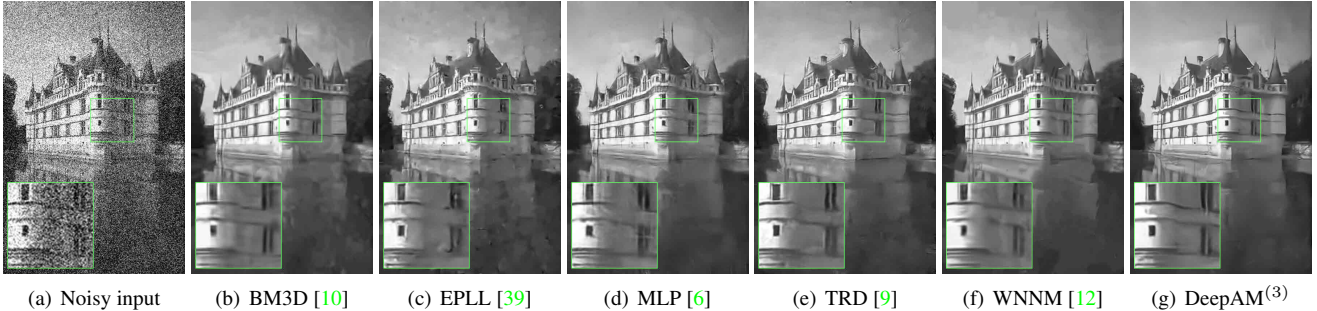


Figure 6: Denoising examples with $\sigma = 50$. (from left to right) noisy input, BM3D [10], EPLL [39], MLP [6], TRD [9], WNNM [12], and DeepAM⁽³⁾. The input image is from the BSD68 [27].

Table 1: The PSNR results on 12 images ($\sigma = 25$). The CSF [30] and TRD [9] run 5 stages with 7×7 kernels.

	C. Man	House	Pepp.	Starf.	Fly	Airpl.	Parrot	Lena	Barb.	Boat	Man	Couple
BM3D [10]	29.47	32.99	30.29	28.57	29.32	28.49	28.97	32.03	30.73	29.88	29.59	29.70
CSF [30]	29.51	32.41	30.32	28.87	29.69	28.80	28.91	31.87	28.99	29.75	29.68	29.50
EPLL [39]	29.21	32.14	30.12	28.48	29.35	28.66	28.96	31.58	28.53	29.64	29.57	29.46
MLP [6]	29.36	32.53	30.20	28.88	29.73	28.84	29.11	32.07	29.17	29.86	29.79	29.68
TRD [9]	29.71	32.62	30.57	29.05	29.97	28.95	29.22	32.02	29.39	29.91	29.83	29.71
WNNM [12]	29.63	33.39	30.55	29.09	29.98	28.81	29.13	32.24	31.28	29.98	29.74	29.80
DeepAM ⁽³⁾	29.97	33.35	30.89	29.43	30.27	29.03	29.41	32.52	29.52	30.23	30.07	30.15

back-propagation is obtained as follows:

$$\frac{\partial \mathcal{L}^{(p)}}{\partial u^{(p)}} = \text{sign}(u^{(p)} - t^{(p)}). \quad (11)$$

To learn the parameters in the network, we need the derivatives of the loss $\mathcal{L}^{(p)}$ with respect to $v^{(p)}$ and $\gamma^{(p)}$. By the chain rule of differentiation, $\frac{\partial \mathcal{L}^{(p)}}{\partial v^{(p)}}$ can be derived from (7):

$$L \frac{\partial \mathcal{L}^{(p)}}{\partial v^{(p)}} = \left[D_x \frac{\partial \mathcal{L}^{(p)}}{\partial u^{(p)}}, D_y \frac{\partial \mathcal{L}^{(p)}}{\partial u^{(p)}} \right]. \quad (12)$$

$\frac{\partial \mathcal{L}^{(p)}}{\partial v^{(p)}}$ is obtained by solving the linear system of (12). Similarly for $\frac{\partial \mathcal{L}^{(p)}}{\partial \gamma^{(p)}}$, we have:

$$\frac{\partial \mathcal{L}^{(p)}}{\partial \gamma^{(p)}} = \left(L^{-1} \frac{\partial \mathcal{L}^{(p)}}{\partial u^{(p)}} \right) \circ (f^{(p)} - u^{(p)}), \quad (13)$$

where “ \circ ” is an element-wise multiplication. Since the loss $\mathcal{L}^{(p)}$ is a scalar value, $\frac{\partial \mathcal{L}^{(p)}}{\partial \gamma^{(p)}}$ and $\frac{\partial \mathcal{L}^{(p)}}{\partial v^{(p)}}$ are $N \times 1$ and $N \times 2$ vectors, respectively, where N is total number of pixels. More details about the derivations of (12) and (13) are available in the supplementary material. The system matrix L is shared in (12) and (13), thus its incomplete factorization is performed only once.

Figure 5 shows the convergence of the PCG method for solving the linear system of (12). We find that a few PCG iterations are enough for the backpropagation. The average

residual, $\|L \frac{\partial \mathcal{L}^{(p)}}{\partial v^{(p)}} - D_x \frac{\partial \mathcal{L}^{(p)}}{\partial u^{(p)}}\|$ on 20 images is 1.3×10^{-6} , after 10 iterations. The table in Fig. 5 compares the runtime of PCG iterations and MATLAB backslash (on 256×256 image). The PCG with 10 iterations is about 5 times faster than the direct linear system solver.

5. Experiments

We jointly train our DeepAM for 20 epochs. From here on, we call DeepAM^(K) the method trained through a cascade of K DeepAM iterations. The MatConvNet library [2] (with 12GB NVIDIA Titan GPU) is used for network construction and training. The networks are initialized randomly using Gaussian distributions. The momentum and weight decay parameters are set to 0.9 and 0.0005, respectively. We do not perform any pre-training (or fine-tuning). The proposed method is applied to single image denoising, depth super-resolution, and RGB/NIR restoration. The results for the comparison with other methods are obtained from source codes provided by the authors. Additional results and analyses are available in the supplementary material.

5.1. Single Image Denoising

We learned the DeepAM⁽³⁾ from a set of 10^5 , 32×32 patches sampled from the BSD300 [19] dataset. Here K was set to 3 as the performance of the DeepAM^(K) con-

Table 2: Average PSNR/SSIM on 68 images from [27] for image denoising with $\sigma = 15, 25,$ and 50 .

σ	PSNR / SSIM						
	BM3D [10]	MLP [6]	CSF [30]	TRD [9]	DeepAM ⁽¹⁾	DeepAM ⁽²⁾	DeepAM ⁽³⁾
15	31.12 / 0.872	-	31.24 / 0.873	31.42 / 0.882	31.40 / 0.882	31.65 / 0.885	31.68 / 0.886
25	28.61 / 0.801	28.84 / 0.812	28.73 / 0.803	28.91 / 0.815	28.95 / 0.816	29.18 / 0.824	29.21 / 0.825
50	25.65 / 0.686	26.00 / 0.708	-	25.96 / 0.701	25.94 / 0.701	26.20 / 0.714	26.24 / 0.716

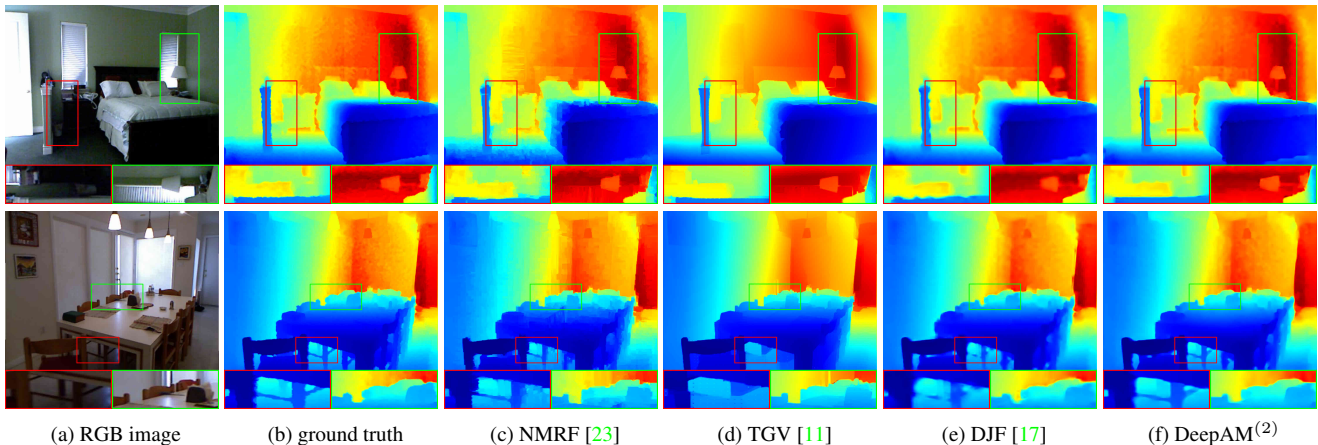


Figure 7: Depth super-resolution examples ($\times 8$): (a) RGB image, (b) ground truth, (c) NMRF [23], (d) TGV [11], (e) DJF [17], and (f) DeepAM⁽²⁾.

verges after 3 iterations (refer to Table 2). The noise levels were set to $\sigma = 15, 25,$ and 50 . We compared against a variety of recent state-of-the-art techniques, including BM3D [10], WNNM [12], CSF [30], TRD [9], EPLL [39], and MLP [6]. The first two methods are based on the nonlocal regularization and the others are learning-based approaches.

Table 1 shows the peak signal-to-noise ratio (PSNR) on the 12 test images [10]. The best results for each image are highlighted in bold. The DeepAM⁽³⁾ yields the highest PSNR results on most images. We could find that our deep aggregation used in the mapping step outperforms the point-wise mapping of the CSF [30] by 0.3~0.5dB. Learning-based methods tend to have better performance than hand-crafted models. We, however, observed that the methods (BM3D [10] and WNNM [12]) based on the nonlocal regularization usually work better on images that are dominated by repetitive textures, e.g., ‘House’ and ‘Barbara’. The non-local self-similarity is a powerful prior on regular and repetitive texture, but it may lead to inferior results on irregular regions.

Figure 6 shows denoising results using one image from the BSD68 dataset [27]. The DeepAM⁽³⁾ visually outperforms state-of-the-art methods. Table 2 summarizes an objective evaluation by measuring average PSNR and structural similarity indexes (SSIM) [35] on 68 images from the BSD68 dataset [27]. As expected, our method achieves a

Table 3: Average BMP ($\delta = 3$) on 449 images from the NYU v2 dataset [33] and on 10 images from the Middlebury dataset [29]. Depth values are normalized within the range [0,255].

Method	BMP ($\delta = 3$): NYU v2 [33] / Middlebury [29]		
	$\times 4$	$\times 8$	$\times 16$
NMRF [23]	1.41 / 4.56	4.21 / 7.59	16.25 / 13.22
TGV [11]	1.58 / 5.72	5.42 / 8.82	17.89 / 13.47
SD filter [13]	1.27 / 2.41	3.56 / 5.97	15.43 / 12.18
DJF [17]	0.68 / 3.75	1.92 / 6.37	5.82 / 12.63
DeepAM ⁽²⁾	0.57 / 3.14	1.58 / 5.78	4.63 / 10.45

significant improvement over the nonlocal-based method as well as the recent data-driven approaches. Due to the space limit, some methods were omitted in the table, and full performance comparison is available in the supplementary materials.

5.2. Depth Super-resolution

Modern depth sensors, e.g. MS Kinect, provide dense depth measurement in dynamic scene, but typically have a low resolution. A common approach to tackle this problem is to exploit a high-resolution (HR) RGB image as guidance. We applied our DeepAM⁽²⁾ to this task, and evaluated it on the NYU v2 dataset [33] and Middlebury dataset [29].

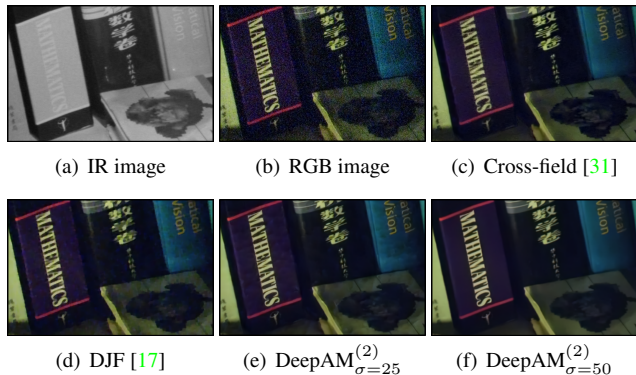


Figure 8: RGB/NIR restoration for real-world examples: (a) RGB image, (b) NIR image, (c) Cross-field [31], (d) DJF [17], (e) DeepAM⁽²⁾ trained with $\sigma = 25$, and (f) DeepAM⁽²⁾ trained with $\sigma = 50$. The result of (c) is from the project webpage of [31].

The NYU v2 dataset [33] consists of 1449 RGB-D image pairs of indoor scenes, among which 1000 image pairs were used for training and 449 image pairs for testing. Depth values are normalized within the range [0,255]. To train the network, we randomly collected 10^5 RGB-D patch pairs of size 32×32 from training set. A low-resolution (LR) depth image was synthesized by nearest-neighbor downsampling ($\times 4$, $\times 8$, and $\times 16$). The network takes the LR depth image, which is bilinearly interpolated into the desired HR grid, and the HR RGB image as inputs.

Figure 7 shows the super-resolution results of NMRF [23], TGV [11], deep joint image filtering (DJF) [17], and DeepAM⁽²⁾. The TGV model [11] uses an anisotropic diffusion tensor that solely depends on the RGB image. The major drawback of this approach is that the RGB-depth coherence assumption is violated in textured surfaces. Thus, the restored depth image may contain gradients similar to the color image, which causes texture copying artifacts (Fig. 7(d)). Although the NMRF [23] combines several weighting schemes, computed from RGB image, segmentation, and initially interpolated depth, the texture copying artifacts are still observed (Fig. 7(c)). The NMRF [23] preserves depth discontinuities well, but shows poor results in smooth surfaces. The DJF [17] avoids the texture copying artifacts thanks to faithful CNN responses extracted from both color image and depth map (Fig. 7(e)). However, this method lacks the regularization constraint that encourages spatial and appearance consistency on the output, and thus it over-smooths the results and does not protect thin structures. Our DeepAM⁽²⁾ preserves sharp depth discontinuities without notable artifacts as shown in Fig. 7(f). The quantitative evaluations on the NYU v2 dataset [33] and Middlebury dataset [29] are summarized in Table 3. The accuracy is measured by the bad matching percentage (BMP)

Table 4: The PSNR results with 5 RGB/NIR pairs from [14]. The noisy RGB images are generated by adding the synthetic Gaussian noise.

	(a) #1	(b) #2	(c) #3	(d) #4	(e) #5
	PSNR				
$\sigma = 50$	BM3D [10]	SD filter [13]	Cross-field [31]	DeepAM ⁽²⁾	
Sequence 1	31.86	30.97	31.45	32.84	
Sequence 2	27.62	26.13	27.59	28.10	
Sequence 3	28.08	28.06	28.47	30.43	
Sequence 4	26.85	25.65	26.91	28.13	
Sequence 5	26.52	26.11	26.98	26.94	
Average	28.19	27.38	28.28	29.28	

[29] with tolerance $\delta = 3$.

5.3. RGB/NIR Restoration

The RGB/NIR restoration aims to enhance a noisy RGB image taken under low illumination using a spatially aligned NIR image. The challenge when applying our model to the RGB/NIR restoration is the lack of the ground truth data for training. For constructing a large training data, we used the indoor IVRL dataset consisting of 400 RGB/NIR pairs [28] that were recorded under daylight illumination⁵. Specifically, we generated noisy RGB images by adding the synthetic Gaussian noise with $\sigma = 25$ and 50, and used 300 image pairs for training.

In Table 4, we performed an objective evaluation using 5 test images in [14]. The DeepAM⁽²⁾ gives better quantitative results than other state-of-the-art methods [10, 13, 31]. Figure 8 compares the RGB/NIR restoration results of Cross-field [31], DJF [17], and our DeepAM⁽²⁾ on the real-world example. The input RGB/NIR pair was taken from the project website of [31]. This experiment shows the proposed method can be applied to real-world data, although it was trained from the synthetic dataset. It was reported in [14] that the restoration algorithm designed (or trained) to work under a daylight condition could also be used for both daylight and night conditions.

6. Conclusion

We have explored a general framework called the DeepAM, which can be used in various image restoration

⁵This dataset [28] was originally introduced for semantic segmentation.

applications. Contrary to existing data-driven approaches that just produce the restoration result from the CNNs, the DeepAM uses the CNNs to learn the regularizer of the AM algorithm. Our formulation fully integrates the CNNs with an energy minimization model, making it possible to learn whole networks in an end-to-end manner. Experiments demonstrate that the deep aggregation in the mapping step is the critical factor of the proposed learning model. As future work, we will further investigate an adversarial loss in pixel-level prediction tasks.

References

- [1] <http://faculty.cse.tamu.edu/davis/suitesparse.html/>. 4
- [2] <http://www.vlfeat.org/matconvnet/>. 6
- [3] A. Agrawal, R. Raskar, S. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Trans. Graph.*, 24(3), 2005. 2
- [4] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM J. Imag. Sci.*, 3(3), 2010. 2
- [5] A. Buades, B. Coll, and J. Morel. A non-local algorithm for image denoising. *CVPR*, 2005. 4
- [6] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: can plain neural networks compete with bm3d? *CVPR*, 2012. 6, 7
- [7] S. Chan, X. Wang, and O. Elgandy. Plug-and-play admm for image restoration: fixed point convergence and applications. *arXiv*, 2016. 4
- [8] Q. Chen and V. Koltun. A simple model for intrinsic image decomposition with depth cues. *ICCV*, 2013. 1
- [9] Y. Chen, W. Yu, , and T. Pock. On learning optimized reaction diffusion processes for effective image restoration. *CVPR*, 2015. 1, 2, 6, 7
- [10] K. Dabov, A. Foi, V. Katkovich, and K. Egiazarian. Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8), 2007. 2, 6, 7, 8
- [11] D. Ferstl, C. Reinbacher, R. Ranftl, M. Ruther, and H. Bischof. Image guided depth upsampling using anisotropic total generalized variation. *ICCV*, 2013. 1, 2, 7, 8
- [12] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. *CVPR*, 2014. 2, 6, 7
- [13] B. Ham, M. Cho, and J. Ponce. Robust image filtering using joint static and dynamic guidance. *CVPR*, 2015. 7, 8
- [14] H. Honda and L. V. G. R. Timofte. Make may day - high-fidelity color denoising with near-infrared. *CVPRW*, 2015. 8
- [15] Y. Kim, B. Ham, C. Oh, and K. Sohn. Structure selective depth superresolution for rgb-d cameras. *IEEE Trans. Image Process.*, 25(11), 2016. 2, 5
- [16] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. *NIPS*, 2009. 1, 2
- [17] Y. Li, J. Huang, N. Ahuja, and M. Yang. Deep joint image filtering. *ECCV*, 2016. 1, 2, 5, 7, 8
- [18] J. Liu, S. Zhang, S. Wang, and D. Metaxas. Multispectral deep neural networks for pedestrian detection. *BMVC*, 2016. 5
- [19] 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. *ICCV*, 2001. 6
- [20] M. Mozer. A focused back-propagation algorithm for temporal pattern recognition. *Complex Systems*, 3(4), 1989. 4, 5
- [21] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. *ICCV*, 2015. 5
- [22] N. Parikh and S. Boyd. Proximal algorithms. *Found. and Trends in optimization*, 2014. 3
- [23] J. Park, H. Kim, Y. W. Tai, M. S. Brown, and I. Kweon. High quality depth map upsampling for 3d-tof cameras. *ICCV*, 2011. 7, 8
- [24] R. Ranftl and T. Pock. A deep variational model for image segmentation. *GCPR*, 2014. 2, 5
- [25] G. Riegler, D. Ferstl, M. R  ther, and H. Bischof. A deep primal-dual network for guided depth super-resolution. *BMVC*, 2016. 2
- [26] G. Riegler, M. R  ther, and H. Bischof. Atgv-net: Accurate depth super-resolution. *ECCV*, 2016. 2, 5
- [27] S. Roth and M. J. Black. Fields of experts. *IJCV*, 82(2), 2009. 6, 7
- [28] N. Salamati, D. Larlus, G. Csurka, and S. Susstrunk. Incorporating near-infrared information into semantic image segmentation. *arXiv*, 2014. 8
- [29] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1). 7, 8
- [30] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. *CVPR*, 2014. 1, 2, 3, 6, 7
- [31] X. Shen, Q. Yan, L. Xu, L. Ma, and J. Jia. Multispectral joint image restoration via optimizing a scale map. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(1), 2015. 1, 2, 8
- [32] X. Shen, C. Zhou, L. Xu, and J. Jia. Mutual-structure for joint filtering. *ICCV*, 2015. 1
- [33] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgb-d images. *ECCV*, 2012. 7, 8
- [34] Y. Wang, J. Yang, W. Yin, , and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Imag. Sci.*, 1(3), 2008. 1, 2, 3, 5
- [35] Z. Wang, A. C. Bovik, H. Rahim, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4), 2004. 7
- [36] L. Xu, C. Lu, Y. Xu, , and J. Jia. Image smoothing via l_0 gradient minimization. *ACM Trans. Graph.*, 30(6), 2011. 1
- [37] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for neural networks for image processing. *arXiv*, 2015. 5
- [38] S. Zheng, S. Jayasumana, B. Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. *ICCV*, 2015. 2, 4
- [39] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. *ICCV*, 2011. 1, 6, 7