

Research Article

Image Restoration Based on Adaptive Dual-Domain Filtering

Ruiqiang He ^{1,2}, Xiangchu Feng ¹, Chenping Zhao ¹, Huazhu Chen,¹
Xiaolong Zhu,¹ and Chen Xu ³

¹School of Mathematics and Statistics, Xidian University, Xi'an 710126, China

²Department of Mathematics, Xinzhou Teachers University, Xinzhou 034000, China

³Institute of Intelligent Computing Science, Shenzhen University, Shenzhen 518060, China

Correspondence should be addressed to Xiangchu Feng; xcfeng@mail.xidian.edu.cn

Received 17 May 2018; Revised 22 August 2018; Accepted 14 September 2018; Published 10 October 2018

Academic Editor: Nazrul Islam

Copyright © 2018 Ruiqiang He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Image restoration is a long-standing problem in low-level computer vision. In this paper, we offer a simple but effective estimation paradigm for various image restoration problems. Specifically, we first propose a model-based Gaussian denoising method Adaptive Dual-Domain Filtering (ADDF) by learning the optimal confidence factors which are adjusted adaptively with Gaussian noise standard deviation. In addition, by generalizing this learning approach to Laplace noise, the learning algorithm of the optimum confidence factors in Laplace denoising is presented. Finally, the proposed ADDF is tactfully plugged into the method frameworks of off-the-shelf image deblurring and single image super-resolution (SISR). The approach, coining the name Plug-ADDF, achieves promising performance. Extensive experiments validate that the proposed ADDF for Gaussian and Laplace noise removals indeed results in visual and quantitative improvements over some existing state-of-the-art methods. Moreover, our Plug-ADDF for image deblurring and SISR also demonstrates superior performance objectively and subjectively.

1. Introduction

Image restoration (IR), which aims to recover the latent clean image x from its degraded observation y , is a classical yet extensively studied problem for its highly practical value in various low-level computer vision applications [1–4]. One widely used degenerate model in image restoration is $y = Ax + v$, where A is a degradation operator and v is additive Gaussian or Laplace noise with standard deviation σ . By designating different degradation operator, one can obtain different IR tasks correspondingly. Three IR tasks will be image denoising when A is an identity operator, image deblurring when A is a blurring operator, and image super-resolution when A is a composite operator of blurring and downsampling. In this paper, we focus on these three IR tasks.

Although the actual noise is not independent identically distributed (i.i.d.), the removal of Gaussian and Laplace noise is still a meaningful work. In addition, there are also denoising problems in IR tasks such as image deblurring and single image super-resolution (SISR). It is well known that BM3D [5] is a highly engineered image denoising

method. The denoising quality of BM3D, which combines the advantages of patch-based techniques with transform domain filtering, is still considered state of the art today. The block matching based method produces visible artifacts in some homogeneous regions, manifesting as some low frequency noises [6–8]. Another method called dual-domain image denoising (DDID) [9] produces results comparable to BM3D, although it is exceptionally simple. DDID is an iterated and guided method that employs a combination of the bilateral filter (BF) [10–13] in the spatial domain and the short-time Fourier transform (STFT) [14] with wavelet shrinkage in the frequency domain. These methods are not good at protecting irregular image texture details and avoiding artifacts.

Recently, a sparse approximation approach [18] is presented for high-density impulse noise removal by utilizing an inverse-distance weighing prediction model. The method firstly decomposes the input image into overlapped patches based on the detection of noise and noise-free pixels; then the potential noise-free pixels are predicted by inverse-distance weighing model; finally, an error-constrained minimization

Step 1. Noise estimation in the spatial domain:

$$\bar{v}_p = \frac{a_n \sum_{q \in N_p} d_q k_q}{\sum_{q \in N_p} k_q},$$

where $d_q = y_q - y_p$, y_p is the center pixel value of the filter window N_p , $q \in N_p$ and k_q is the bilateral kernel.

Step 2. Noise estimation in the frequency domain:

$$\hat{v}_p = \frac{A_n \sum_{f \in F_p} D_f K_f}{(2r+1)^2} \triangleq DDF_p(y),$$

where K_f is the frequency kernel and r is the radius of N_p . D_f is the Fourier coefficients

$$D_f = \sum_{q \in N_p} (d_q - \bar{v}_p) k_q e^{-i(2\pi/(2r+1))f \cdot (q-p)}.$$

Step 3. Image denoising: initialize guide image x_N with the noisy input y , and perform the guided iteration

$$x_{n-1} = y - \text{Im}DDF(x_n + iy),$$

where n counts down from N to 1.

ALGORITHM 1: The Basic Steps of Dual-Domain Filtering (DDF).

problem is solved by the orthogonal matching pursuit. At the same time, a morphological mean (MM) filter [19] is also used for the removal of high-density impulse noise by two significant modules: noise-free pixel counter (NPC) module and morphological pixel dilation (MPD) module. Specifically, the NPC module is employed to detect the number and position of the noise-free pixels, while in the subsequent MPD module, the morphological dilatation operation on the noise-free pixels is iteratively implemented to replace the neighbor noise pixels. These two approaches exhibit excellent performance in removing the noise of images contaminated with high-density impulse noise in both qualitative and quantitative terms. In contrast, the proposed adaptive dual-domain filtering (ADDF) has obvious advantages in removing Gaussian noise and Laplace noise, especially when the noise variance is large.

AC-PT [20] implements detail-preserving image denoising based on adaptive clustering of nonlocal patches and progressive PCA thresholding approach. The basic process of AC-PT is the following. Initially, the global noise level is estimated based on the Marchenko-Pastur (MP) law [21]. Subsequently, an adaptive clustering method is utilized to collect the self-similar blocks of the image by using the global nonlocal self-similarity of the image. Lastly, a progressive PCA thresholding filtering is applied to denoise the clustered patches by MP-SVD step and LMMSE step. In a word, the AC-PT method is a great improvement of BM3D, and a superior denoising performance is obtained by adaptive clustering. Nevertheless, the proposed ADDF does not need to perform a search for similar blocks in an image, thereby effectively reducing the amount of calculation.

In [22], a denoising algorithm on low-dose CT images employs a multiframe statistical tool, blind source separation (BSS) combined with a standard denoiser block matching 3-D filter (BM3D) [5]. On the one hand, the essence of BSS is to iteratively solve a numerical optimization problem using a maximum-entropy principle [23]. On the other hand, the BSS plus BM3D hybrid model with different frame numbers is applied successfully to an image sequence so as to achieve excellent CT image denoising. The method improves the quality of CT image reconstruction while increasing the

complexity and computational time of the algorithm. This is mainly reflected in the optimization process of BSS.

A wavelet transform approach [24] is proposed to denoise 1-D experimental signals by selecting the number of decomposition levels and calculating new noise thresholds. This method utilizes multiresolution analysis to achieve time-frequency decomposition of the signal, thereby effectively separating the noise from the signal. To be specific, an objective method for selecting the number of wavelet decomposition levels is presented by classifying the sparsity of the detail components. This method can also formulate an appropriate threshold without noise estimation. Lots of experiments on electron spin resonance (ESR) spectroscopy [25] show that this method can effectively remove noise and accurately recover signal peaks. Generally speaking, existing sophisticated and highly effective denoising methods are touching the ceiling in the aspect of denoising performance [26], which indicates that any improvement on these basis will have to pay a heavy price.

Dual-domain filtering (DDF) (see Algorithm 1) [27], which is based on the work DDID, is a noise estimator in the spatial and frequency domains. One can see from Algorithm 1 that one of the most critical procedures in DDF is the choice of the confidence factors a_n and A_n . This is due to the fact that these two factors directly affect the accuracy of the noise estimate in the two domains and ultimately determine the final denoising effect. However, the confidence levels in DDF are too low to provide satisfactory results especially when the noise is severe. Therefore, we expect to learn the optimal confidence factors in the two domains to improve the accuracy of noise estimation. Meanwhile, we have noticed that although learning based approaches [28–34] have recently become very popular, a mass of parameters need to be learned and the training time is often very long. Furthermore, these learning methods need a lot of samples to train the constructed network, which makes the learning process very complicated. With the above considerations, we expect to improve the performance of the model-based method by simple learning with as few samples as possible. All these motivate us to learn the optimal confidence factors in the spatial and frequency domains, so as to further

improve the accuracy of noise estimation and enhance the performance of image restoration.

At present, many scholars have done a lot of works on noise estimation [35–38]. However, in this paper, instead of estimating the noise parameters, we estimate the optimal confidence factors a_n and A_n that are closely related to noise in the spatial and frequency domains. Surprisingly, by learning on a few samples, the proposed adaptive dual-domain filtering (ADDF) has good generalization ability. The learning of the optimal confidence factors, which does not increase the computational complexity of the algorithm, makes the denoising effect of ADDF significantly better than DDF.

Inspired by the fact that Plug-and-Play ADMM [39] allows one to plug any off-the-shelf image denoising algorithm as a subproblem in the ADMM algorithm and another line of our research is to plug the proposed denoiser ADDF into the method frameworks of the other two IR tasks: image deblurring and SISR. Experimental results demonstrate that the simple learning on a small number of samples can improve the effectiveness of model-based algorithm. Moreover, the proposed method, called Plug-ADDF, can effectively improve image deblurring and SISR performance. The major contributions of this paper are summarized as follows:

(1) Two novel confidence factors for Gaussian noise estimation are presented. By learning the optimal confidence power and iteration number that vary with Gaussian noise standard deviation, the accuracy of noise estimation in ADDF is better than DDF.

(2) By generalizing the learning approach to Laplace noise removal, an algorithm for learning the optimum confidence factors is developed to accurately estimate noise, resulting in a simple yet effective Laplace denoising method. We show that ADDF in Laplace denoising performance is obviously superior to DDF.

The rest of the paper is organized as follows. Section 2 presents the learning of the optimal confidence factors in Gaussian denoising. Section 3 describes the proposed algorithm and experiments of Gaussian denoising. In Section 4, we generalize the proposed learning approach to the removal of Laplace noise. Section 5 presents image deblurring based on ADDF, while single image super-resolution based on ADDF is given in Section 6. Finally, Section 7 concludes this work.

2. The Proposed Optimal Confidence Factors for Gaussian Denoising

In DDF [27], the confidence factors for Gaussian noise estimation in the spatial and frequency domains are expressed as

$$a_n = \cos\left(\frac{n-1}{N} \frac{\pi}{2}\right), \quad (1)$$

$$A_n = \cos\left(\frac{n-1}{N} \frac{\pi}{2}\right), \quad (2)$$

where $N = 8$ is the default iteration number in DDF and n counts down from N to 1 with iterations. In Step 2 of Algorithm 1, the purpose of $d_q - \bar{v}_p$ is to remove the

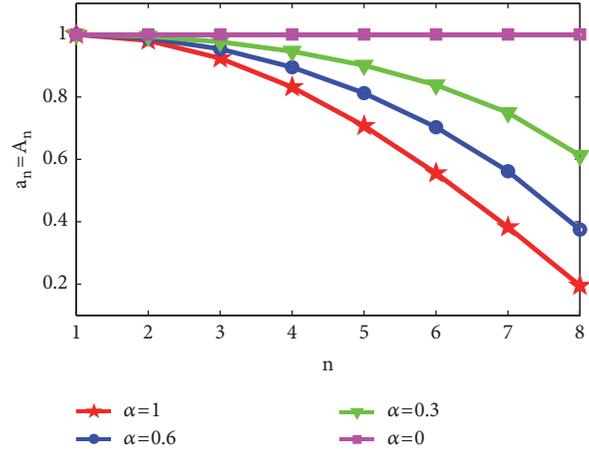


FIGURE 1: The graph of the confidence factors $a_n = A_n = \cos^\alpha(((n-1)/N)(\pi/2))$, ($N = 8$) corresponding to the four different values of $\alpha \in [0, 1]$.

noise \bar{v}_p in d_q , and the boundary of the remaining part is polished by the bilateral kernel k_q , so that the ringing effect can be efficaciously reduced when implementing DFT. However, when the noise is severe, the noise in d_q is also severe and a small confidence factor a_n makes the subtracted noise \bar{v}_p insufficient. Therefore, by adding a power $\alpha \in [0, 1]$, the confidence factor a_n of the noise estimate can be increased. Thus, \bar{v}_p can get a certain increase. In addition, according to Figure 7 in the paper, we can find from the statistical perspective that, in terms of noise estimation, ADDF utilizing the increased confidence factors is more accurate than DDF with the fixed confidence factors. These indicate that we should increase the confidence factors of the noise estimate, especially when the noise is severe. With the above considerations, in this paper we propose two novel confidence factors for Gaussian noise estimation in the two domains respectively as follows:

$$a_n = \cos^\alpha\left(\frac{n-1}{N} \frac{\pi}{2}\right), \quad (3)$$

$$A_n = \cos^\alpha\left(\frac{n-1}{N} \frac{\pi}{2}\right), \quad (4)$$

where we term α “confidence power.” We here take $N=8$ as an example to briefly analyze the relationship between the proposed confidence factors and the original ones. As shown in Figure 1, as α increases from 0 to 1, the proposed $a_n = A_n$ are decreasing. One can see that when $\alpha = 1$, the proposed $a_n = A_n$ are the original confidence factors; when $\alpha = 0$, $a_n = A_n$ are increased to the maximum value 1. Therefore, the proposed confidence factors (3) and (4) are an increase based on the original ones (1) and (2).

In our proposed confidence factors (3) and (4), the confidence power α and iteration number N vary adaptively with the noise standard deviation σ , i.e., $\alpha = \alpha(\sigma)$ and $N = N(\sigma)$. Obviously, (1) and (2) are special cases of (3) and (4) when $\alpha = 1$ and $N = 8$, respectively. Therefore, the proposed confidence factors are more general forms that are generalized by the ones in DDF. These adaptive confidence factors can more

TABLE 1: The PSNR table: each value represents the PSNR mean with the two images by DDF.

$\alpha \setminus \sigma$	20	30	40	50	60	70	80	90	100
1	34.1052	32.5666	31.4717	30.5835	29.8216	29.1446	28.5225	27.9326	27.3740
0.95	34.1049	32.5726	31.4873	30.6082	29.8543	29.1844	28.5684	27.9832	27.4286
0.9	34.1013	32.5749	31.4985	30.6277	29.8809	29.2173	28.6066	28.0256	27.4740
0.85	34.0947	32.5733	31.5053	30.6421	29.9016	29.2436	28.6375	28.0597	27.5106
0.8	34.0849	32.5682	31.5081	30.6517	29.9166	29.2635	28.6612	28.0860	27.5386
0.75	34.0720	32.5595	31.5069	30.6566	29.9262	29.2774	28.6781	28.1046	27.5583
0.7	34.0559	32.5475	31.5018	30.6570	29.9307	29.2855	28.6885	28.1160	27.5701
0.65	34.0366	32.5324	31.4932	30.6532	29.9302	29.2882	28.6928	28.1205	27.5744
0.6	34.0141	32.5143	31.4812	30.6454	29.9252	29.2860	28.6914	28.1187	27.5718

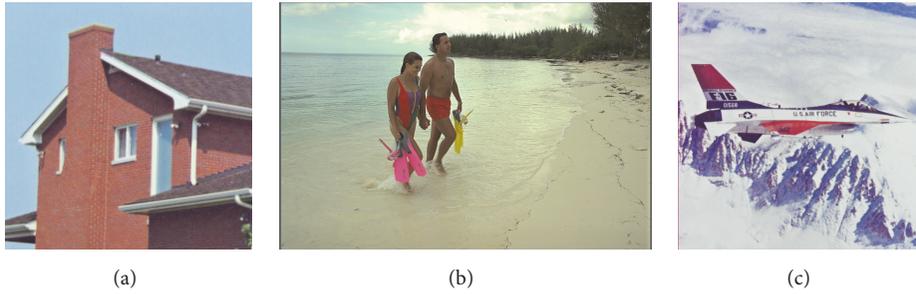


FIGURE 2: Three training images for learning the optimal confidence factors. Among them, images (a and b) are utilized to learn the optimal confidence power, while images (a and c) that are in Section 2.2 are employed to learn the optimum iteration number.

robustly estimate noise, which improves the performance of image recovery. In the following, we will use a small number of training samples to learn the optimal confidence power and the optimum iteration number, which constitute the optimal confidence factors for noise estimation.

2.1. Learning the Optimal Confidence Power. Firstly, in the two-dimensional plane shown in Table 1, different values of noise standard deviation σ and confidence power α are, respectively, listed in the horizontal and vertical axes. We specify that the location corresponding to the pair of numbers (σ, α) is marked as $L_{\sigma, \alpha}$. We utilize the confidence factors (3) and (4) and the different (σ, α) in Table 1 to implement DDF on the training images (a) and (b) in Figure 2. Then the mean values of PSNR corresponding to the two images are placed at position $L_{\sigma, \alpha}$. As a result, we obtain a matrix of size 9×9 , which is denoted as A and shown in Table 1. It should be mentioned that here the iteration number is $N = 8$ in [27].

Next, theoretically when a specific noise standard deviation σ in DDF is given, there is an optimal confidence power α that maximizes PSNR. From a lot of experiments, we have found that the optimal α changes with different σ . At the same time, we hope that the relationship curve about the optimal α and σ is monotonous and smooth. On the other hand, the sum of the PSNR is the largest when DDF with the corresponding values σ and α is implemented. In order to achieve these goals, we design Algorithm 2.

From Figure 3(a) obtained by Algorithm 2, it can be found that the whole curve shows basically monotonically decreasing trend. In addition, when the noise is small, the

reliability of v estimation changes greatly; when the noise is large, the reliability difference between the adjacent steps is small.

Finally, in order to smooth the curve in Figure 3(a), and look further for the relationship between α and σ , we perform a curve fitting. From the general trend of the curve in Figure 3(a), we empirically conjecture that the regression polynomial of the original data basically conform to the following negative exponential function.

$$\alpha = ae^{-b\sigma} + c. \quad (5)$$

In the above formula (5), the parameter a determines the decay rate of the function. In other words, with the increase of σ , the larger the parameter a , the faster the α decays. Here b is the attenuation coefficient, which is inversely proportional to the half-decay period, and the deviation c improves the agreement between the fitting curve and the experimental data.

By curve fitting with Matlab, the polynomial coefficients are acquired. Thus, the fitted polynomial is

$$\alpha = 0.9219e^{-0.04326\sigma} + 0.6242. \quad (6)$$

The fitted curve is shown in Figure 3(b). It can be found that with the increase of the noise standard deviation σ , the optimal confidence power α is adaptively monotonically decreasing. At the same time, it can be seen from Figure 1 that when α decreases (where $N = 8$ and for fixed n), the proposed confidence factors $a_n = A_n = \cos^\alpha(((n-1)/N)(\pi/2))$ increase. Thus the greater the noise is, the greater

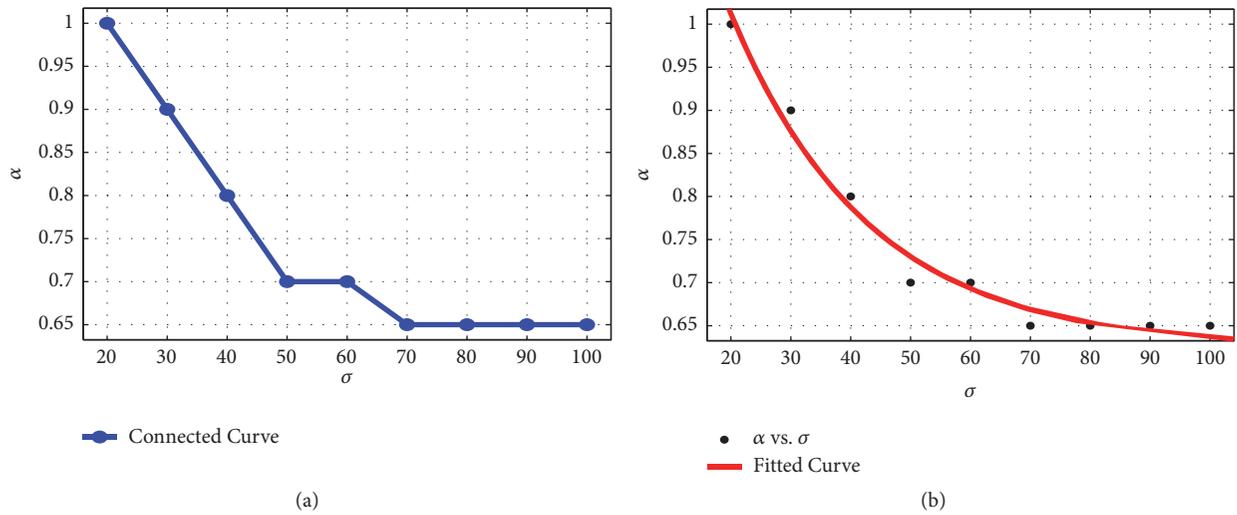
Step 1. The element at i -th row j -th column in the matrix A is denoted as $a_{i,j}$. And the largest element in column 1 of A is recorded as $b_{k_1,1}$;

Step 2. In column 2 of A , the element $a_{i,2}$ ($i = 1, \dots, 9$) is summed with each element above $a_{i,1}$ (including $a_{i,1}$) in column 1, respectively. The maximum sum is denoted as $b_{i,2}$, which then replaces $a_{i,2}$ in A . Finally, the largest one of the elements $b_{1,2}, b_{2,2}, \dots, b_{9,2}$ is marked as $b_{k_2,2}$;

Step 3. In column 3 of A , the element $a_{i,3}$ ($i = 1, \dots, 9$) is summed with each element above $b_{i,2}$ (including $b_{i,2}$) in column 2 respectively. And the maximum sum is denoted by $b_{i,3}$, which then replaces $a_{i,3}$ in A . Similarly, the largest one of the elements $b_{1,3}, b_{2,3}, \dots, b_{9,3}$ is labeled as $b_{k_3,3}$;

Step 4. Similar to the Step 4, the elements in columns from 4 to 9 in A are sequentially updated. The resulting matrix is denoted as B , as shown in Table 2. Then the largest element in each column of B is respectively recorded as $b_{k_4,4}, b_{k_5,5}, \dots, b_{k_9,9}$;

Step 5. Connect the elements $b_{k_1,1}, b_{k_2,2}, \dots, b_{k_9,9}$ (bold elements), and we get the curve shown in Figure 2(a).

 ALGORITHM 2: Learning of the relationship curve about α and σ .

 FIGURE 3: (a) The connection graph of bold data in Table 2. (b) The fitted curve about α and σ .

the confidence factors a_n and A_n for noise estimation are, which seems to be inconsistent with our intuition. In fact, here the number of iterations is assumed to be constant, which implies that the confidence factors a_n and A_n should also control the intensity of denoising.

2.2. Learning the Optimum Iteration Number. On the basis of the optimal confidence power, this part will focus on learning the optimal iteration number. We empirically speculate that the iterative number N should also be adjusted adaptively with noise standard deviation σ . Utilizing the idea the preliminary experimental results on the image (a) in Figure 2 are shown in Figure 4(a). It is exhilarating that the experimental results are consistent with our conjecture, which encourages us to further explore the optimal relationship between N and σ . Thus, we take advantage of the learned optimal α in (6) and increase the iteration number to implement DDF on the training images (a) and (c) in Figure 2. The results are shown in Table 3.

It can be found from Table 3 that when σ is small (such as $\sigma = 25$), PSNR decreases with increasing N . When the noise gets severe, this trend has changed. For the phenomenon, here

we simply do some analysis of the reason. As mentioned in Step 2 of Algorithm 1, the purpose of $d_q - \bar{v}_p$ is to remove the noise \bar{v}_p in d_q , and the boundary of the remaining part is polished by the bilateral kernel k_q , so that the ringing effect can be efficaciously reduced when implementing DFT. When σ is small, the noise in d_q is also small, so the confidence factor a_n in the subtracted \bar{v}_p cannot be too large. However, as N increases, the confidence factor $a_n = \cos^\alpha(((n-1)/N)(\pi/2))$ also increases (for the same α and n), which makes the noise estimation \bar{v}_p not accurate enough, resulting in a downward trend in PSNR. As shown in Table 3, this situation will change as σ gets large.

According to Table 3, in order to fit a strictly monotonous curve, we sacrifice a little PSNR; i.e., when $\sigma = 85$, N takes 11. Therefore, the fitted curve of σ and N is shown in Figure 4(b). Simultaneously the relationship between σ and N can be expressed as

$$N = 3.822\sigma^{0.2328}. \quad (7)$$

Considering that the number of iteration is an integer, the relationship (7) can be adjusted as follows:

$$N = \text{round}(3.822\sigma^{0.2328}). \quad (8)$$

TABLE 2: The table of PSNR summation, and at the point $(\sigma, \alpha) = (100, 0.65)$ the sum of PSNR is the largest.

$\alpha \setminus \sigma$	20	30	40	50	60	70	80	90	100
1	34.1052	66.6718	98.1435	128.7270	158.5486	187.6932	216.2157	244.1483	271.5223
0.95	34.1049	66.6778	98.1651	128.7733	158.6276	187.8120	216.3804	244.3636	271.7922
0.9	34.1013	66.6799	98.1786	128.8063	158.6872	187.9045	216.5111	244.5367	272.0107
0.85	34.0947	66.6785	98.1854	128.8275	158.7291	187.9727	216.6102	244.6699	272.1805
0.8	34.0849	66.6734	98.1882	128.8399	158.7565	188.0200	216.6812	244.7672	272.3058
0.75	34.0720	66.6647	98.1870	128.8448	158.7710	188.0484	216.7265	244.8311	272.3894
0.7	34.0559	66.6527	98.1819	128.8452	158.7759	188.0614	216.7499	244.8659	272.4360
0.65	34.0366	66.6376	98.1733	128.8414	158.7754	188.0641	216.7569	244.8774	272.4518
0.6	34.0141	66.6195	98.1613	128.8336	158.7704	188.0619	216.7555	244.8756	272.4492

TABLE 3: PSNR: using the learned optimal α and different N .

N	Test image (a)				Test image (c)			
	$N = 8$	$N = 9$	$N = 10$	$N = 11$	$N = 8$	$N = 9$	$N = 10$	$N = 11$
$\sigma = 25$	32.89	32.84	32.77	32.67	33.08	33.02	32.94	32.83
$\sigma = 40$	31.32	31.33	31.23	31.13	31.12	31.12	31.08	31.01
$\sigma = 55$	30.09	30.11	30.12	29.99	29.63	29.67	29.68	29.65
$\sigma = 70$	28.97	29.02	29.02	28.96	28.46	28.51	28.53	28.52
$\sigma = 85$	27.88	27.98	28.01	27.98	27.50	27.56	27.60	27.59
$\sigma = 100$	26.83	26.96	27.03	27.04	26.66	26.74	26.79	26.79

Based on the above mentioned function (8), we can obtain the optimal number of iteration when the noise level of noisy image is given. It can be seen that the more severe the noise, the more the iterations in DDF. Intuitively, this is primarily due to the fact that in DDF the result for each iteration is the guided image of the next iteration. As the number of iteration increases, the result of DDF is getting closer to the clean image.

In our ADDE, the method of determining the optimal confidence factors for noise estimation is essentially the regression of data. Our ADDE operates on the block corresponding to each pixel, and the results obtained on the other images have not changed much, so we believe that the optimal confidence factors obtained by regression are stable. This also indicates the learning algorithm of the optimal confidence factors proposed in this paper has good robustness.

Since we propose a model-based approach, it is possible to have a strong generalization ability by learning a small number of parameters with a small amount of training data. This is different from CNN, where a lot of data are needed to learn a lot of parameters. This view will be validated in Gaussian denoising experiments in the next section.

3. Proposed Algorithm and Experiments for Gaussian Denoising

3.1. Proposed Algorithm. The existing image denoising methods [5, 28, 31, 40–43] all assume that noise standard deviation σ is known; i.e., the estimation of σ is another issue [44, 45]. Therefore, from the previous learning, when σ in Gaussian denoising is given, we can get the optimal confidence power α and the optimal iteration number N . Then the optimal confidence factors a_n and A_n for v estimation can be calculated.

The proposed learning of the optimal confidence factors in Gaussian noise removal is summarized in Algorithm 3.

Since the optimal confidence factors are obtained, we can robustly estimate noise in the spatial and frequency domains. The improvement of the noise estimation accuracy in DDF promotes the denoising effect. For clarity, here we take a few notes:

- (1) When the iteration number N is fixed ($N = 8$), we learn the optimal confidence power $\alpha = \alpha(\sigma)$.
- (2) Based on the learned optimal $\alpha = \alpha(\sigma)$, we learn the optimal iteration number $N = N(\sigma)$.
- (3) The optimal α is robust to N ; i.e., α is not very sensitive to N , and in fact α is mainly determined by σ . In addition, the optimal α contributes principally to the improvement of denoising performance. Thus, α and N above are updated only once.

In addition, in the training phase we use the three clean images shown in Figure 2 to learn the optimal confidence factors for noise estimations in the spatial and frequency domains. In these three images, images (a) and (b) are employed to learn the optimal confidence power α , while images (a) and (c) are utilized to learn the optimum iteration number N . The optimal α and N here constitute the optimal confidence factors $a_n = A_n = \cos^\alpha(((n-1)/N)(\pi/2))$.

In summary, the noises are estimated in the spatial and frequency domains by utilizing the learned optimal confidence factors which vary adaptively with noise standard deviation. This image denoising method by adaptive noise estimation in the two domains is called adaptive dual-domain filtering (ADDE).

3.2. Experiments of Gaussian Denoising. Without loss of generality, in Gaussian denoising experiments, we use 9 color test

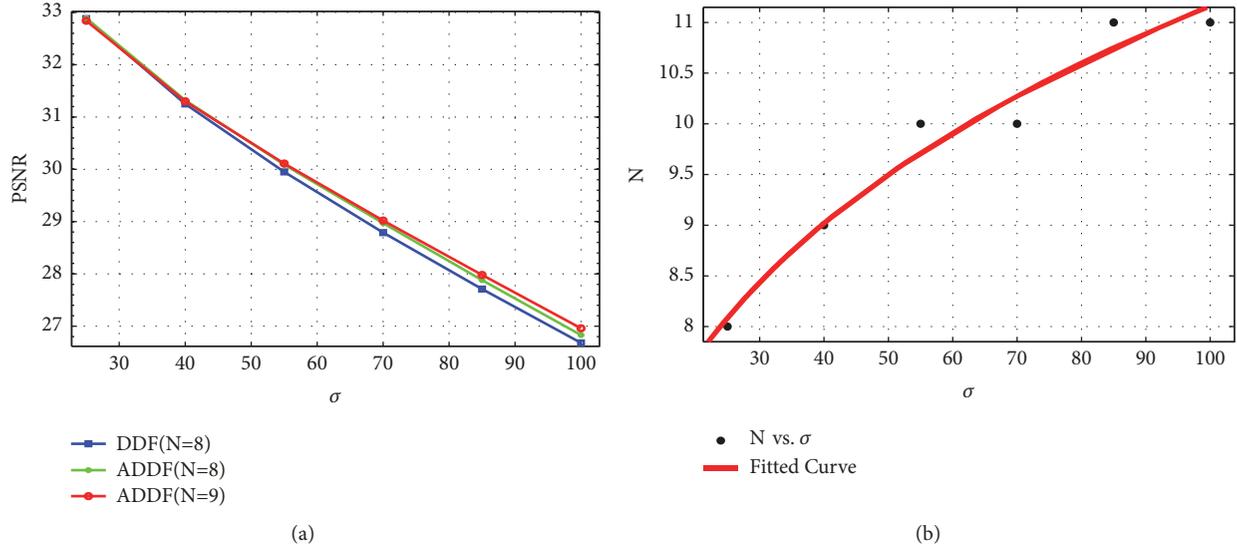


FIGURE 4: (a) PSNR comparison on the image (a) in Figure 1 by DDF and ADDE. (b) The fitted curve about N and σ based on the bold data in Table 3.

```

Input noise standard deviation  $\sigma$ 
1: Calculate the optimal confidence power  $\alpha$ 
 $\alpha = 0.9219e^{-0.04326\sigma} + 0.6242$ 
2: Calculate the optimal iteration number  $N$ 
 $N = \text{round}(3.822\sigma^{0.2328})$ 
3: Initialize  $n = N$ 
4: for  $N$  down to 1 do
5: Calculate the optimal confidence factors

$$a_n = \cos^\alpha\left(\frac{n-1}{N} \frac{\pi}{2}\right)$$


$$A_n = \cos^\alpha\left(\frac{n-1}{N} \frac{\pi}{2}\right)$$

6: end for
Output The optimal confidence factors  $a_n$  and  $A_n$ 
    
```

ALGORITHM 3: Learning of the optimal confidence factors for noise estimation in Gaussian denoising.

images from two different test datasets. One is the test dataset LIVE1 [15] containing 29 natural images, and the other one Set14 [16] includes 12 images. Note that all these images are widely utilized for image quality assessment.

We implement dual-domain filtering with the learned optimal confidence factors on 9 color pictures for quantitative comparison with the similar state-of-the-art methods BM3D [5], DDID [9], and DDF [27] that are the standards for model-based denoising methods. It should be noted that, in [27], DDF has compared with the related denoising methods, so our ADDE only compares with these three methods. We consider five noise levels, i.e., $\sigma = 25, 40, 55, 70, 85$, and 100. The codes of the three competing methods are downloaded from the authors' websites and the default parameter settings are used in our experiments.

The experimental results are shown in Table 4. As can be seen from Table 4, our ADDE is robust to noise variance. The proposed ADDE has higher PSNR than BM3D, DDID, and

DDF. Moreover, as the noise standard deviation σ increases, the PSNR gain of ADDE is widening.

To further verify the visual performance of the proposed method ADDE, on the one hand we apply it to four images with various noise standard deviations $\sigma = 55, 70, 85, 100$. Figure 5 compares our new denoiser against other similar state-of-the-art denoising methods such as BM3D, DDID, and DDF. ADDE produces the cleanest and smoothest results. This robust noise estimator makes the restored image more consistent with our visual effects.

On the other hand, in Figure 6, examples of Gaussian denoising by the four methods are presented with $\sigma = 100$ on image Kodak23 that comes from the test dataset LIVE1 [15]. In order to reflect the difference, we take the local details and zoom in to show. One can see that ADDE can well enhance the image detail features such as edges and textures, which are often oversmoothed by other denoising methods. It should be noted that as the training images, the three natural images shown in Figure 2 have almost no textures. However, with the test image Kodak 23 which has more textures, ADDE has increased PSNR by 0.29dB over DDF.

Finally, Figure 7 shows a comparison of probabilistic density curves among the Ground Truth and the noises estimated by DDF, ADDE. From the partial enlarged illustration in Figure 7, we can find that, in the vicinity of the noise mean, where the probability density of the noise is the largest, the probability density curve of the noise estimated by ADDE is closer to the one of the true Gaussian noise. This implies that ADDE is more accurate than DDF in terms of noise estimation. Of course, this is mainly due to the fact that ADDE employs the confidence factors changing with the noise standard deviation for noise estimations in spatial and frequency domains, while DDF utilizes the fixed confidence factors.

These experiments above demonstrate that the advent of the optimal confidence power and iteration number has effectively contributed to significant improvements over the

TABLE 4: PSNR comparison on 9 color images from the datasets LIVE1 [15] and Set14 [16] for Gaussian denoising. One can see that the proposed ADDF really outperforms the state-of-the-art methods BM3D [5], DDID [9], and DDF [27].

σ	Methods	Lena	Peppers	Girl	Kodak2	Kodak7	Kodak10	Kodak14	Kodak19	Kodak23	Avg.
$\sigma = 25$	BM3D	32.27	31.20	31.33	32.44	33.66	33.82	30.03	31.78	34.79	32.37
	DDID	32.30	31.25	31.45	32.29	33.52	33.70	30.03	31.57	34.83	32.33
	DDF	32.45	31.41	31.58	32.49	33.74	33.89	30.18	31.72	34.97	32.49
	ADDF	32.45	31.40	31.58	32.47	33.77	33.90	30.18	31.71	34.99	32.49
$\sigma = 40$	BM3D	30.11	29.27	29.16	30.30	30.60	30.92	27.48	29.36	32.11	29.92
	DDID	30.68	29.88	29.58	30.46	31.09	31.46	27.82	29.66	32.75	30.38
	DDF	30.87	30.05	29.70	30.66	31.34	31.66	27.97	29.83	32.93	30.56
	ADDF	30.90	30.05	29.72	30.67	31.42	31.74	27.98	29.84	33.01	30.59
$\sigma = 55$	BM3D	29.54	28.58	28.25	29.51	29.52	30.05	26.35	28.71	31.32	29.09
	DDID	29.44	28.78	28.22	29.25	29.37	29.88	26.40	28.45	31.25	29.00
	DDF	29.66	28.97	28.38	29.48	29.64	30.11	26.56	28.68	31.48	29.22
	ADDF	29.75	29.01	28.44	29.54	29.78	30.25	26.60	28.72	31.60	29.30
$\sigma = 70$	BM3D	28.63	27.66	27.29	28.62	28.35	28.97	25.40	27.86	30.20	28.11
	DDID	28.41	27.82	27.17	28.33	28.08	28.67	25.35	27.52	30.06	27.93
	DDF	28.66	28.02	27.35	28.59	28.36	28.92	25.51	27.80	30.33	28.17
	ADDF	28.80	28.13	27.45	28.70	28.54	29.10	25.57	27.90	30.54	28.30
$\sigma = 85$	BM3D	27.88	26.87	26.54	27.87	27.43	28.12	24.65	27.17	29.26	27.31
	DDID	27.52	26.95	26.30	27.57	27.03	27.70	24.50	26.73	29.07	27.04
	DDF	27.79	27.16	26.52	27.85	27.31	27.97	24.66	27.07	29.37	27.30
	ADDF	27.97	27.31	26.64	28.01	27.54	28.20	24.79	27.21	29.60	27.47
$\sigma = 100$	BM3D	27.11	26.06	25.85	27.06	26.44	27.25	23.89	26.38	28.32	26.48
	DDID	26.75	26.16	25.58	26.91	26.16	26.89	23.81	26.04	28.21	26.28
	DDF	27.04	26.38	25.84	27.22	26.43	27.18	23.97	26.41	28.53	26.56
	ADDF	27.26	26.58	25.97	27.42	26.69	27.44	24.10	26.61	28.82	26.77

previous state-of-the-art works. Therefore, we believe that the learned optimal confidence factors in noise estimation are the critical contributors of our proposed method.

4. Generalization to Laplace Denoising

From the previous section, one can see that, to a large extent, the proposed confidence factors can change the distribution of the estimated noise, which can be illustrated in Figure 7. At the same time, we note that Laplace noise has many important applications in the industrial field. Therefore, we are motivated to generalize the proposed ADDF for Gaussian Denoising to the removal of Laplace noise.

Here the degraded image y is given by $y = x + v$, where v is additive Laplace noise. In this section, we concentrate on the assumption that v obeys Laplace distribution [46–49]; i.e., $v \sim L(\lambda, c)$, and its probability density function (PDF) is defined as

$$p_v(x; c, \lambda) = \frac{1}{2c} e^{-|x-\lambda|/c}, \quad (9)$$

where λ represents the location parameter and is equal to the mean, while $c > 0$ is related to the variance as $\sigma^2 = 2c^2$.

4.1. Learning the Optimal Confidence Factors. Here, it is assumed in this work that the mean of the additive Laplace noise is zero, and the standard deviation σ is known. Hence,

the noise v obeys such a Laplace distribution $v \sim L(0, \sigma/\sqrt{2})$. Similar to the learning procedure of the optimal α and N in the confidence factors (3) and (4) for Gaussian denoising, the relationship between the optimal confidence power α and noise standard deviation σ in Laplace denoising can be obtained below, and the fitted curve is shown in Figure 8(a),

$$\alpha = 1.078 \sin(0.01654\sigma + 0.9821) + 0.4856 \sin(0.03076\sigma + 3.159). \quad (10)$$

The reason we take the sine functions is that when the noise standard deviation is large, the fitted curve can well approach the original data. Similarly, the expression between the optimal iteration number N and noise standard deviation σ is the following, and Figure 8(b) shows the fitted curve:

$$N = \text{round}(1.733e^{0.01287\sigma} + 4.747). \quad (11)$$

Based on the above learning results, as the standard deviation of Laplace noise is given, we can get the optimal confidence power and iteration number, so as to obtain the optimal confidence factors in Laplace noise estimation. We summarize them as Algorithm 4.

4.2. Experiments of Laplace Denoising. In order to further test the capacity of the proposed ADDF for Laplace denoising, we also consider five noise levels, i.e., $\sigma = 25, 40, 55, 70, 85$,

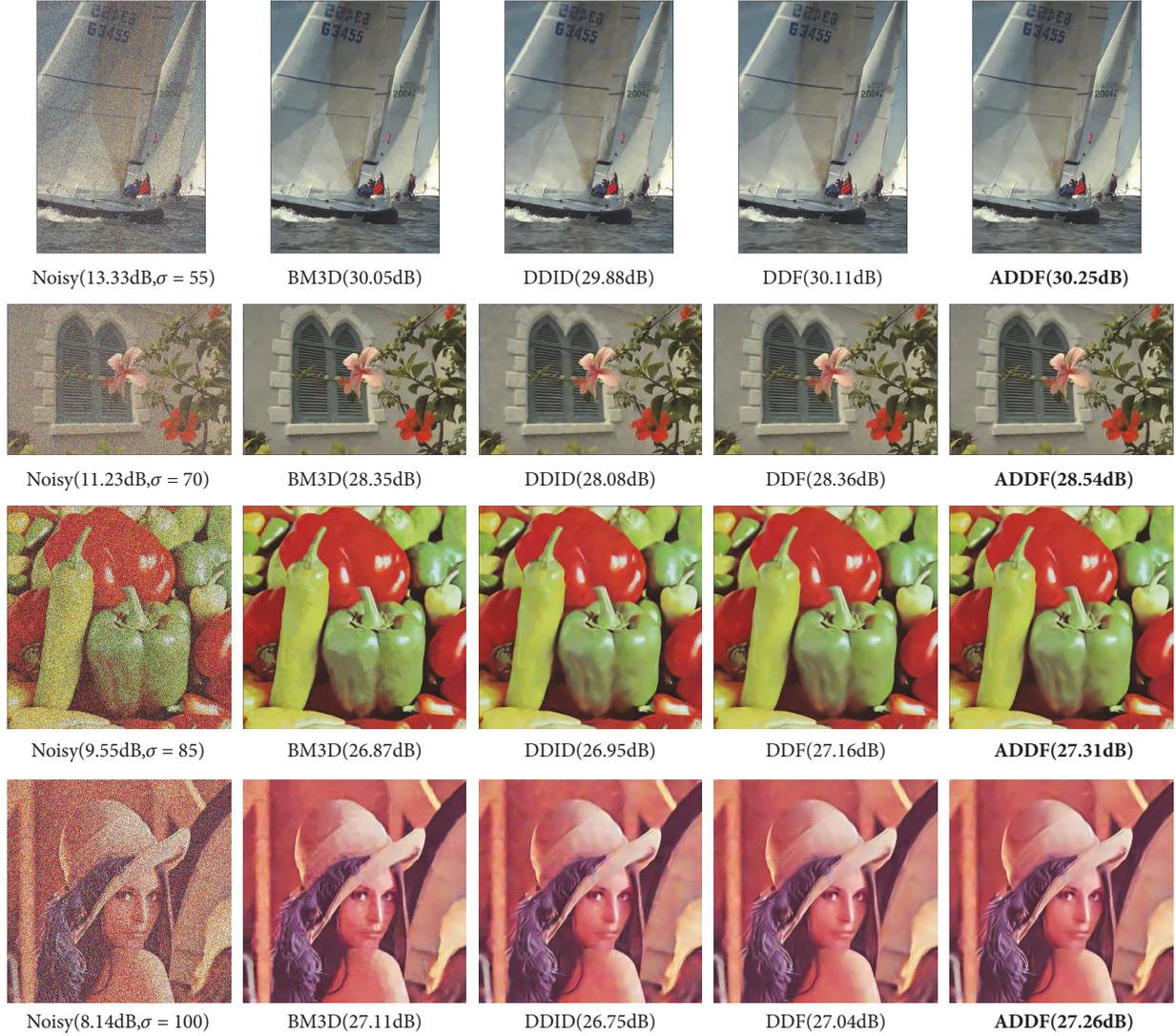


FIGURE 5: PSNR (dB) comparison for four color images Kodak10, Kodak7, Peppers, and Lena from the datasets LIVE1 [15] and Set14 [16] with four different methods. The proposed ADDF produces the most aesthetically pleasing results.

and 100. We implement ADDF on 9 test images that are from the three different test datasets LIVE1 [15], Set14 [16], and Set5 [17]. Table 5 lists the comparison of our proposed ADDF and DDF in Laplace noise removal. As can be seen from all the examples, in terms of PSNR, our learning method leads to significant improvements over DDF. Moreover, compared with DDF, with the increase of σ , PSNR gain in ADDF generally continues to expand. Figure 9 shows the visual effect of ADDF on the image “Butterfly” that is in the dataset Set5. One can see that when the Laplace denoising experiment is applied to the test image “Butterfly” that is textured, ADDF in terms of PSNR achieves an increase of 0.23 dB over DDF. All of these indicate that our ADDF is robust to the images of the different datasets.

In order to further validate our thesis, we carry out an experiment based on a synthesized image with three people playing basketball. This image is collected from the internet and the noise level is $\sigma = 25$. The results are shown in

Input noise standard deviation σ

1: Calculate the optimal confidence power α

$$\alpha = 1.078 \sin(0.01654\sigma + 0.9821) + 0.4856 \sin(0.03076\sigma + 3.159)$$

2: Calculate the optimal iteration number N
 $N = \text{round}(1.733e^{0.01287\sigma} + 4.747)$

3: **Initialize** $n = N$

4: **for** N down to 1 **do**

5: Calculate the optimal confidence factors

$$a_n = \cos^\alpha\left(\frac{n-1}{N} \frac{\pi}{2}\right)$$

$$A_n = \cos^\alpha\left(\frac{n-1}{N} \frac{\pi}{2}\right)$$

6: **end for**

Output The optimal confidence factors a_n and A_n

ALGORITHM 4: Learning of the optimal confidence factors in Laplace denoising.



FIGURE 6: Examples of Gaussian denoising on Kodak23 from the test dataset LIVE1 [15]. One can see that the proposed ADDF can recover more texture details than the other methods. **Best viewed on screen and zoom in.**

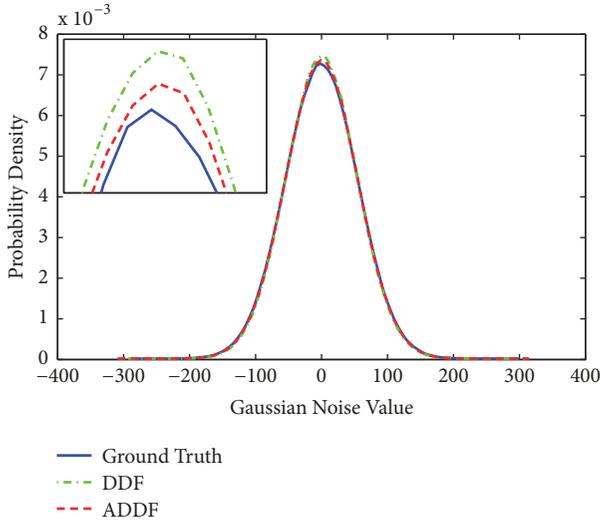


FIGURE 7: Comparison of probabilistic density curves among the Ground Truth and the noises estimated by DDF, ADDF on image “Kodak10” from the test dataset LIVE1 with $\sigma = 55$.

Figure 10, where one can see that our approach ADDF can obtain better visual and quantitative advantages. From the above experimental analysis, it is not difficult to see that compared with DDF, the noise estimated by the proposed ADDF is closer to the true Laplace noise.

Based on the three images shown in Figure 2, we have learned the optimal confidence factors for removing Gaussian and Laplace noise and obtained the ideal effects. We replace

the three images in Figure 2 with the other images and learn the optimal confidence power and iteration number by the similar training methods. It can be found from the training results that the optimum confidence factors used for estimating noises in spatial and frequency domains are basically the same as the original ones, which indicates that the learning algorithm proposed in this paper has good robustness.

5. Image Deblurring Based on ADDF

5.1. Proposed Algorithm Plug-ADDF for Image Deblurring. Due to the ill-posed property of image restoration, the degraded image we observed is y via $y = Ax + v$, where A is a blurring operator and v is Gaussian or Laplace noise. To recover x from y , we need to solve the following optimization problem:

$$\hat{x} = \arg \min_x \frac{1}{2} \|y - Ax\|_2^2 + \lambda \varphi(x), \quad (12)$$

where one minimizes the energy function that is composed of the data fidelity term $(1/2)\|y - Ax\|_2^2$, the regularization term $\varphi(x)$, and the parameter λ . The data fidelity term guarantees that the solution conforms to the degradation process, and the regularization term performs desired attribute of the output.

Because of involving the inverse operator of A , solving the problem (12) is sometimes difficult. With the proximal operator [50], the proximal forward-backward splitting

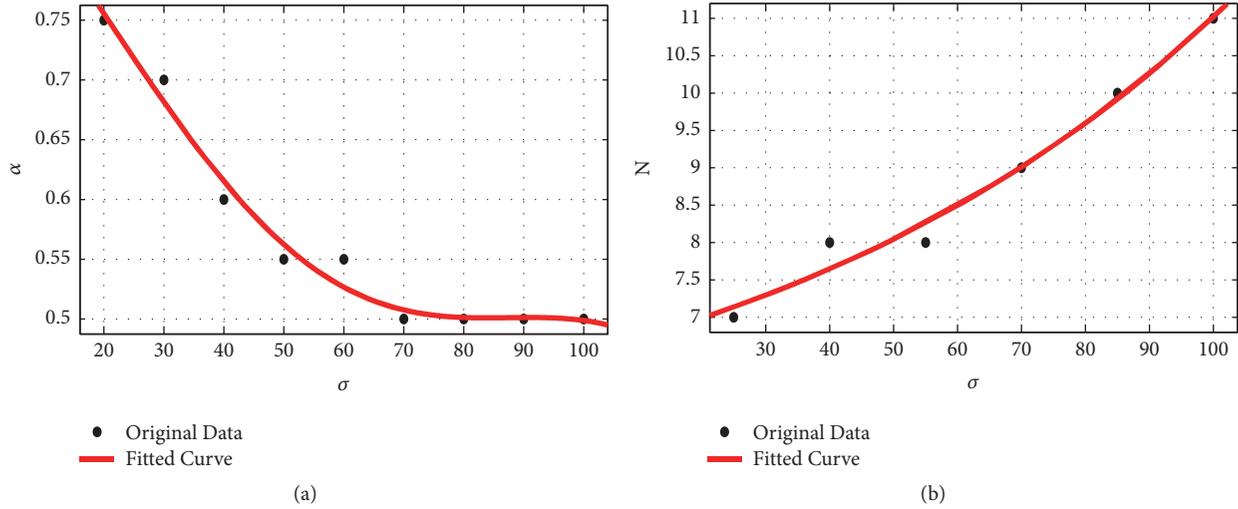


FIGURE 8: (a) The relationship graph about α and σ . (b) The relationship graph about N and σ .

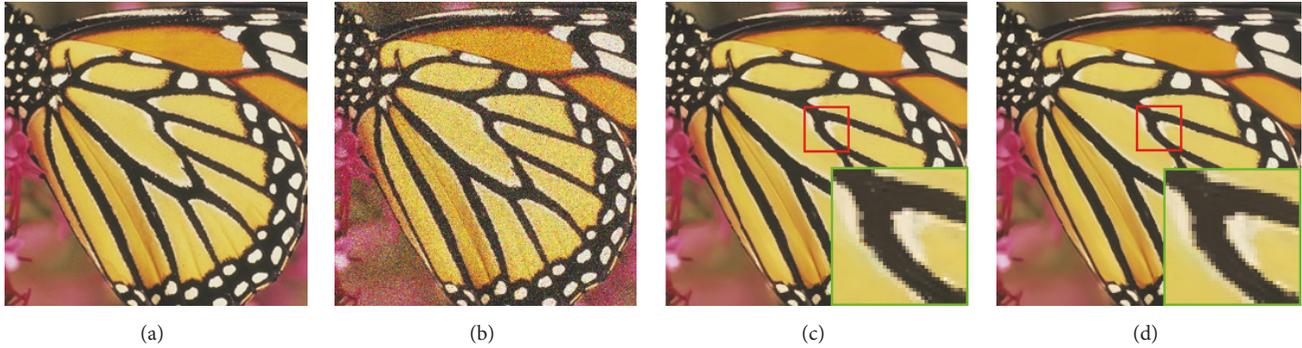


FIGURE 9: An example of Laplace noise removal for the “Butterfly” image from the test dataset Set5 with $\sigma = 25$. (a) Original; (b) Noisy; (c) DDF (30.09dB); (d) ADDF (30.32dB).

(PFBS) approach was applied in [51]. The PFBS algorithm [50] applied to the problem (12) can be described as

$$x_{k+1} = \text{Prox}_{\lambda\varphi}(x_k - \delta A^T(Ax_k - y)), \quad (13)$$

where δ is the step size and the proximal operator is defined as

$$\text{Prox}_{\lambda\varphi}(\tau) = \underset{u}{\text{argmin}} \left(\frac{1}{2} \|u - \tau\|_2^2 + \lambda\varphi(u) \right). \quad (14)$$

The main advantage of this proximal operator is that the problem (14) is convex. Therefore, there must be a unique minimum solution for any τ . By utilizing an auxiliary variable \tilde{x}_{k+1} , the solution to minimize the problem (12) can be obtained by the two-step iterative scheme,

$$\begin{aligned} \tilde{x}_{k+1} &= x_k - \delta A^T(Ax_k - y) \\ x_{k+1} &= \underset{x}{\text{argmin}} \frac{1}{2} \|x - \tilde{x}_{k+1}\|_2^2 + \lambda\varphi(x). \end{aligned} \quad (15)$$

Motivated by the idea of the Plug-and-Play ADMM [39], one can find that computing the proximal operator is actually

a standard denoising problem which can be easily solved with any off-the-shelf denoising algorithm, denoted by the following \mathcal{D} :

$$x_{k+1} = \mathcal{D}(\tilde{x}_{k+1}). \quad (16)$$

Here we choose our proposed denoising approach ADDE, which indeed results in visual and quantitative improvements over similar methods. Then the solution of the problem (12) can be eventually obtained via the following alternately iterative algorithm:

$$\begin{aligned} \tilde{x}_{k+1} &= x_k - \delta A^T(Ax_k - y) \\ x_{k+1} &= \text{ADDF}(\tilde{x}_{k+1}), \end{aligned} \quad (17)$$

In this work we vividly name the proposed method for image deblurring as Plug-ADDE.

5.2. Experiments of Image Deblurring. As a common setting, we apply a blur kernel and add additive Gaussian or Laplace noise with standard deviation σ to synthesize the blurred images. Furthermore, we assume that the convolution is performed under circular boundary conditions.

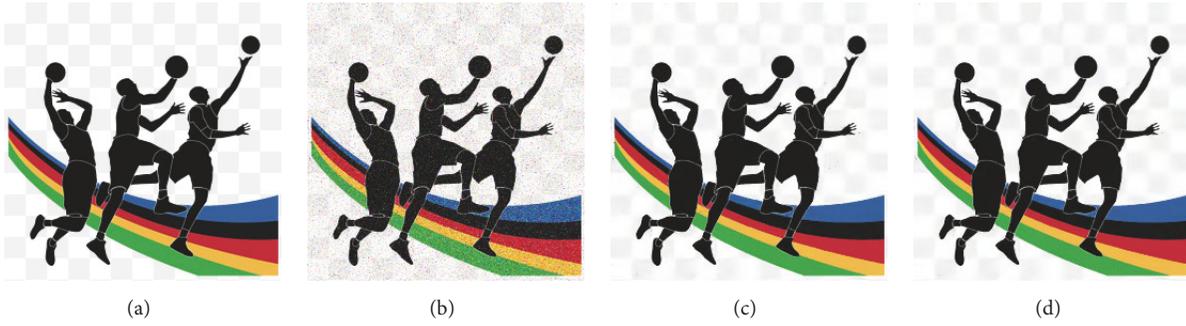


FIGURE 10: An experiment of Laplace noise removal on a synthesized image from the internet with $\sigma = 25$. (a) Original; (b) Noisy; (c) DDF (32.57dB); (d) ADDF (33.03dB).

TABLE 5: PSNR comparison on 9 color images from the three datasets LIVE1 [15], Set14 [16], and Set5 [17] in Laplace noise removal. One can see that ADDF is clearly better than DDF.

σ	Methods	Butterfly	Lena	Peppers	Zebra	Couple	Girl	Kodak3	Kodak10	Kodak23	Avg.
$\sigma = 25$	DDF	30.09	32.32	31.30	29.93	32.13	31.45	34.52	33.75	34.83	32.26
	ADDF	30.32	32.38	31.35	30.00	32.16	31.45	34.60	33.81	34.96	32.34
$\sigma = 40$	DDF	27.91	30.66	29.92	27.74	30.06	29.59	32.26	31.49	32.76	30.27
	ADDF	28.05	30.78	30.00	27.83	30.11	29.63	32.32	31.56	32.96	30.36
$\sigma = 55$	DDF	26.51	29.38	28.77	26.33	28.53	28.24	30.71	29.90	31.30	28.85
	ADDF	26.70	29.53	28.94	26.44	28.62	28.29	30.78	29.98	31.57	28.98
$\sigma = 70$	DDF	25.38	28.32	27.77	25.26	27.37	27.17	29.54	28.67	30.10	27.73
	ADDF	25.56	28.58	28.00	25.38	27.55	27.30	29.74	28.85	30.44	27.93
$\sigma = 85$	DDF	24.38	27.41	26.82	24.35	26.41	26.29	28.57	27.66	29.07	26.77
	ADDF	24.56	27.69	27.09	24.51	26.69	26.50	28.90	27.92	29.47	27.04
$\sigma = 100$	DDF	23.49	26.60	25.95	23.57	25.61	25.53	27.77	26.83	28.19	25.95
	ADDF	23.62	26.88	26.22	23.75	25.97	25.80	28.11	27.11	28.56	26.22

TABLE 6: PSNR (dB) results by four deblurring methods on the ten test images that are shown in Figure 11. The test images are from the two datasets LIVE1 [15] and Sets5 [17]. It should be noted here that Gaussian noise is added to the blurry images, and the Plug-ADDF employs the optimal confidence factors of Gaussian denoising from the Algorithm 3.

Methods	Leaves	Butterfly	Plants	Bird	Starfish	Baby	Flower	Hat	Face	Parrots
Plug-BM3D	24.74	25.22	28.54	29.23	26.61	28.30	27.04	27.80	27.66	27.62
Plug-DDID	25.15	25.72	28.95	29.49	26.85	28.50	27.37	28.13	27.65	28.05
Plug-DDF	25.20	25.76	29.07	29.63	26.92	28.59	27.52	28.18	27.71	28.07
Plug-ADDF	25.31	25.87	29.21	29.81	27.00	28.70	27.60	28.30	27.78	28.18

5.2.1. Blurry Images Contaminated by Gaussian Noise. To make an evaluation, we employ a commonly-used 5×5 Gaussian blur kernel with standard deviation 1.0. Then the Gaussian noise with $\sigma = 18$ is added to the blurry images. For the compared methods, we still choose the three model based methods BM3D, DDID, and DDF and nest these methods separately into the framework of image deblurring. Here we name the three deblurring methods Plug-BM3D, Plug-DDID, and Plug-DDF.

The subsequent issue we consider is the parameter settings. We note that there are two parameters step size δ and iterative number N . In the gradient descent, we empirically set the step size $\delta = 0.2$. In our experiment, we need the iteration number $N = 10$ to achieve satisfactory results.

It is worth noting that, in the iterations of the proposed Plug-ADDF, ADDF makes full use of the optimal confidence factors for Gaussian denoising in Algorithm 3. We believe that these adaptive optimal confidence factors corresponding to different noises are really suitable for denoising in image deblurring.

To verify the performance of the proposed Plug-ADDF for image deblurring, we apply it to ten test images, which are from the two datasets LIVE1 and Set5 and shown in Figure 11. The PSNR results for deblurring utilizing the four different methods are shown in Table 6. As one can see, the proposed Plug-ADDF method achieves promising PSNR results. Figure 12 illustrates deblurred image “Bird” by the four methods. We can see that Plug-BM3D, Plug-DDID,



FIGURE 11: Ten test images from the two datasets LIVE1 [15] and Set5 [17]. From left to right and top to bottom, they are, respectively, named Leaves, Parrots, Butterfly, Baby, Plants, Bird, Starfish, Flower, Hat, and Bike.

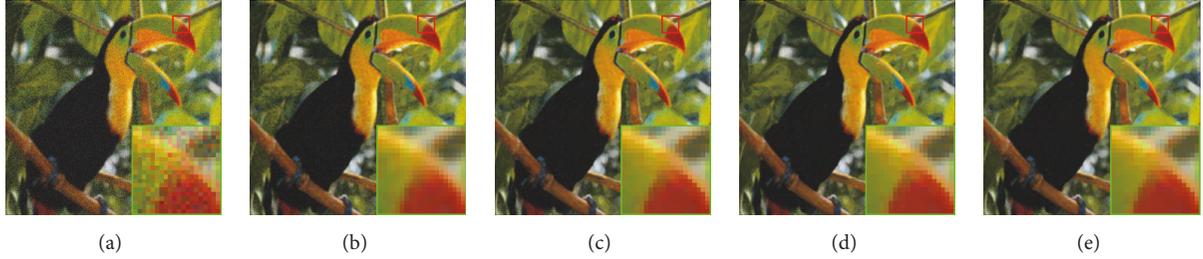


FIGURE 12: Deblurring performance comparison on the test image “Bird” from the dataset Set5 [17]. (a) Blurry image contaminated by Gaussian noise; (b) Plug-BM3D (29.23dB); (c) Plug-DDID (29.49dB); (d) Plug-DDF (29.63dB); (e) **Plug-ADDF (29.81dB)**.

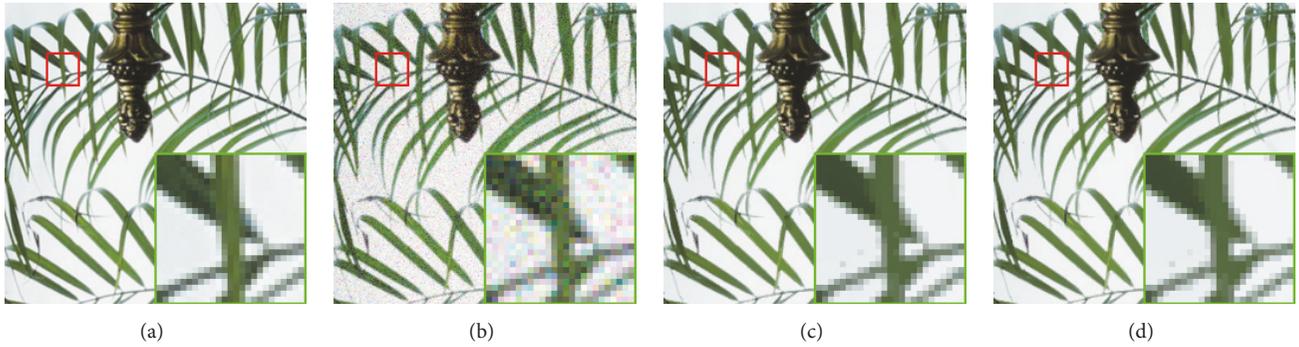


FIGURE 13: Deblurring performance comparison on the test image “Leaves.” (a) Original image; (b) blurry image contaminated by Laplace noise; (c) Plug-DDF (29.77dB); (d) **Plug-ADDF (30.17dB)**.

and Plug-DDF tend to smooth the edges and produce color artifacts. In contrast, the proposed Plug-ADDF can restore the clarity of the images.

5.2.2. Blurry Images Contaminated by Laplace Noise. This subsection focuses on the recovery problem of the blurred images contaminated by Laplace noise. First the basic parameter settings in the experiments are introduced. We utilize 5×5 Gaussian blur kernel with standard deviation 0.5. Then the Laplace noise with $\sigma = 18$ is added to the blurry images. In the gradient descent, we choose a small step size $\delta = 0.2$. We implement the alternating iterative process in (17) for ten times. It should be emphasized here that in the iterations we implement Plug-ADDF with the optimal confidence factors of Laplace denoising from Algorithm 4 so as to obtain excellent experimental results of image deblurring.

Next, we briefly describe the experimental effects of image deblurring. The PSNR results of Plug-DDF and the proposed Plug-ADDF for image deblurring on the ten test images are shown in Table 7. Figure 13 illustrates deblurred images “Leaves” by the two methods. As one can see, our Plug-ADDF achieves promising deblurring results.

6. Single Image Super-Resolution Based on ADDF

6.1. Proposed Algorithm Plug-ADDF for Single Image Super-Resolution (SISR). In this section, we improve the performance of SISR by subtly embedding the proposed ADDF into the SISR method framework. Let $y = Ax + v$ be the observed degraded image, where A is a composite operator of blurring and downsampling and v is Gaussian or Laplace noise. Our goal is to recover the original image x from y . The SISR problem can be formulated as follows:

$$\hat{x} = \arg \min_x \frac{1}{2} \|y - Ax\|_2^2 + \gamma \psi(x). \quad (18)$$

Motivated by the SISR approach proposed in [52] that iteratively updates the back-projection [53] step and the image denoising step, we first utilize the following back-projection:

$$\tilde{x}_{k+1} = x_k - \alpha (y - x_k \downarrow_{sf}) \uparrow_{bicubic}^{sf}, \quad (19)$$

where \downarrow_{sf} represents the degradation operator whose down-scaling factor is sf , $\uparrow_{bicubic}^{sf}$ denotes bicubic interpolation

TABLE 7: PSNR (dB) result comparison of image deblurring on the ten test images that are shown in Figure 10. The blurry images are contaminated by Laplace noise.

Methods	Leaves	Butterfly	Plants	Bird	Starfish	Baby	Flower	Hat	Face	Parrots
Plug-DDF	29.77	29.29	30.88	29.73	29.06	30.13	30.11	31.31	28.80	31.43
Plug-ADDF	30.17	29.56	31.07	29.93	29.33	30.29	30.34	31.51	28.83	31.73

TABLE 8: PSNR (dB) results by four SISR methods on the ten test images which are from the two datasets LIVE1 [15] and Sets5 [17] and shown in Figure 10. These images are contaminated by Gaussian noise. One can see that, in the SISR performance, the proposed Plug-ADDF is significantly better than the other three competing methods.

Methods	Leaves	Butterfly	Plants	Bird	Starfish	Baby	Flower	Hat	Face	Parrots
Plug-BM3D	20.91	22.13	26.60	27.19	24.46	26.34	24.73	25.93	26.75	24.98
Plug-DDID	20.93	22.14	26.62	27.30	24.46	26.52	24.74	25.94	26.98	25.16
Plug-DDF	20.94	22.15	26.66	27.32	24.48	26.55	24.76	25.98	27.01	25.19
Plug-ADDF	21.29	22.46	27.36	27.71	24.82	27.20	25.26	26.83	27.68	25.67

operator whose upscaling factor is sf , and here α is the step size.

Next, with regard to the denoising step in SISR, we also employ the Proximal operator [50] and the auxiliary variable \tilde{x}_{k+1} to solve this problem. Hence, the problem (18) can be solved by the following iterative algorithm:

$$\begin{aligned}\tilde{x}_{k+1} &= x_k - \alpha (y - x_k \downarrow_{sf}) \uparrow_{bicubic}^{sf} \\ x_{k+1} &= \text{Prox}_{\gamma\psi}(\tilde{x}_{k+1}) \\ &= \underset{x}{\text{argmin}} \frac{1}{2} \|x - \tilde{x}_{k+1}\|_2^2 + \gamma\psi(x).\end{aligned}\quad (20)$$

Finally, based on the fact that Plug-and-Play ADMM [39] supports any existing denoiser, we replace the solution of the proximal operator with the proposed denoising approach ADDF, which has an excellent effect on noise removal. Therefore, the proposed algorithm Plug-ADDF for SISR is the following two-step alternate iteration:

$$\begin{aligned}\tilde{x}_{k+1} &= x_k - \alpha (y - x_k \downarrow_{sf}) \uparrow_{bicubic}^{sf} \\ x_{k+1} &= \text{ADDF}(\tilde{x}_{k+1}).\end{aligned}\quad (21)$$

6.2. Experiments for Single Image Super-Resolution. In the experimental part of SISR, the images we deal with include two cases: LR blurry images contaminated by Gaussian noise and LR blurry ones contaminated by Laplace noise.

6.2.1. LR Blurry Images Contaminated by Gaussian Noise. In our SISR, the experimental low-resolution (LR) images are generated by firstly blurring a high-resolution (HR) image with a 3×3 Gaussian kernel with standard deviation 0.2 and subsequently downsampling on the blurred image by a scaling factor 2. The additive Gaussian noise with standard deviation $\sigma = 18$ is also added to the LR blurry images, which makes the image restoration more challenging. In this paper, we consider using the Matlab function “imresize” to perform typical image degradation downsampling.

In our experiment, we repeatedly implement (21) eight times for SISR. We set the step size α to 0.2. Here we use

BM3D, DDID, and DDF as the competitive denoisers, which we also vividly term Plug-BM3D, Plug-DDID, and Plug-DDF. It is worth noting that, in the denoising step of our Plug-ADDF, the optimal confidence factors for Gaussian denoising in Algorithm 3 are employed.

Table 8 shows the PSNR results of four different methods for SISR on the ten test images from LIVE1 and Set5. We can see that the proposed Plug-ADDF is significantly superior to the other three competing methods in terms of PSNR. We show in Figure 14 an example to illustrate the SISR results by the four methods. One can see that the image restored by our approach Plug-ADDF is clearer and more image details are recovered. Such a promising result is attributed to the fact that the proposed Plug-ADDF skillfully takes advantage of adaptive dual-domain filtering during the iterations of SISR.

6.2.2. LR Blurry Images Contaminated by Laplace Noise. In this subsection, we mainly deal with the recovery problem of LR blurry images contaminated by Laplace noise. Firstly we introduce the parameter settings in the experiments. The experimental LR images are modeled by a 7×7 Gaussian blurring kernel with standard deviation 0.3 and downsampling on the blurred image with a scaling factor 2. Subsequently the additive Laplace noise with standard deviation $\sigma = 12$ is added to the LR blurry images. In the gradient descent, we set a small step size $\delta = 0.1$, while in the denoising step we carry out ADDF with the optimal confidence factors of Laplace denoising in Algorithm 4. Utilizing these settings, we iteratively implement (21) eight times for SISR.

Secondly, we outline the experimental effects of SISR. Table 9 shows PSNR results of Plug-DDF and the proposed Plug-ADDF on the ten images that are from the datasets LIVE1 [15] and Set5 [17]. Figure 15 illustrates the SISR performance comparison for the image “Starfish.” We can find that, for SISR with Laplace noise, the proposed Plug-ADDF outperforms Plug-DDF both objectively and subjectively.

7. Conclusions

Different from the popular deep learning methods which usually require a large number of samples for a long time

TABLE 9: PSNR (dB) result comparison of Plug-DDF and Plug-ADDF on the ten test images. The LR blurry images are contaminated by Laplace noise.

Methods	Leaves	Butterfly	Plants	Bird	Starfish	Baby	Flower	Hat	Face	Parrots
Plug-DDF	23.06	23.88	27.76	28.52	25.90	27.78	26.07	26.93	27.65	26.55
Plug-ADDF	23.17	23.98	27.80	28.52	25.99	27.84	26.16	27.03	27.68	26.62

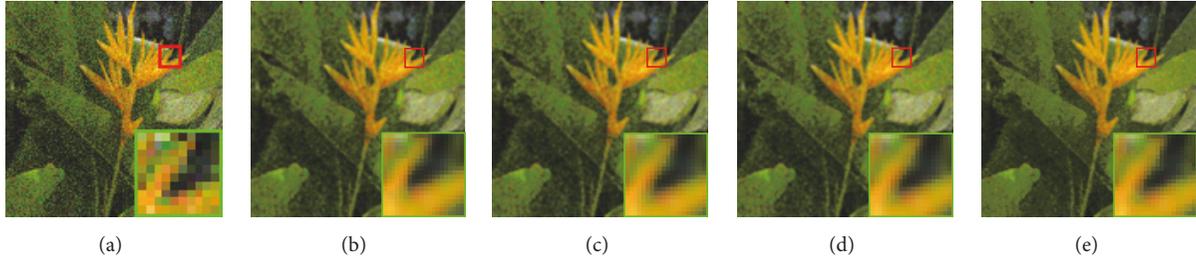


FIGURE 14: SISR performance comparison on the test image “Plants.” (a) Zoomed LR blurry image contaminated by Gaussian noise; (b) Plug-BM3D (26.60dB); (c) Plug-DDID (26.62dB); (d) Plug-DDF (26.66dB); (e) **Plug-ADDF (27.36dB)**.

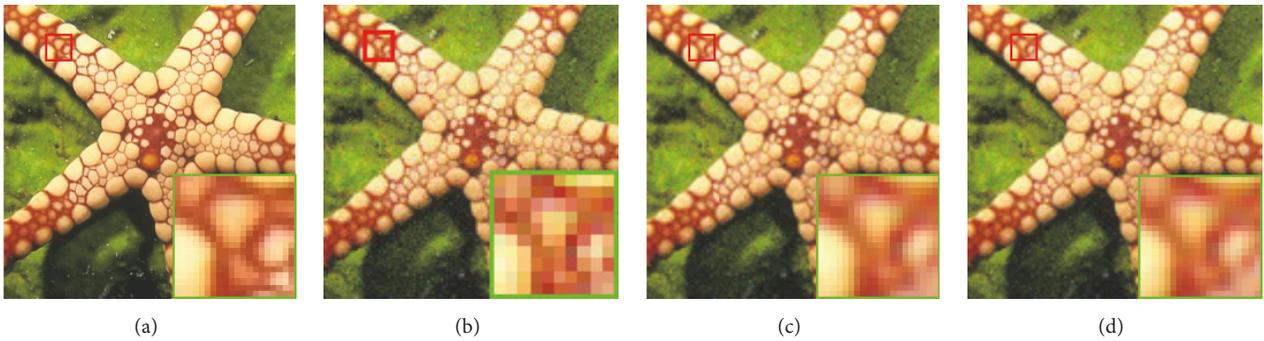


FIGURE 15: SISR performance comparison on the test image “Starfish.” (a) Original image; (b) zoomed LR blurry image contaminated by Laplace noise; (c) Plug-DDF (25.90dB); (d) **Plug-ADDF (25.99dB)**.

to learn vast parameters, in this paper the proposed ADDF, which is developed by simple learning on a small amount of samples, can effectively improve the performance of model-based methods. The model-based approach is still based on the traditional methods such as differential equation diffusion, TV regularization, and transform domain filtering. These methods estimate different original images from the noisy images mainly by means of image prior information such as local smoothness, nonlocal self-similarity, and sparsity. The core idea of these model-based approaches is that image priors rely on knowledge rather than on data learning. By learning the optimal confidence factors in the spatial and frequency domains, we present a robust estimation paradigm ADDF which is suitable for removing additive Gaussian and Laplace noise. One key procedure in the proposed ADDF scheme is to adaptively estimate noise according to different noise levels. DDID and DDF are simpler than BM3D to implement and yet rival BM3D in quality. It is gratifying our ADDF outperforms the similar state-of-the-art approaches BM3D, DDID, and DDF under almost no increase in the amount of computation. Moreover, we subtly embed ADDF into the classical image deblurring and SISR method framework. The experimental results confirm that

our proposed Plug-ADDF method produces a satisfactory image recovery effect.

With respect to the computational cost, we here mainly compare ADDF with DDF. Specifically, we select ten noisy 512×512 color images with $\sigma = 25$ for denoising and then take the average time for comparison. One can see that ADDF (121s) has almost the same computing time as DDF (120s) but can result in better image recovery performance than DDF. Therefore, compared with DDF, the proposed method does not increase the computational cost. We implement the experiments on one PC with two Intel CPUs (2.10 GHz and 2.10GHz) and 64.0 GB RAM in the Matlab R2017b operating environment.

We here make a comprehensive comparison of the proposed ADDF with a typical CNN-based method DnCNN [28]. DnCNN needs to utilize 400 images to learn more than 600,000 parameters. For such high computational complexity, it takes more than one day to train the required neural network. Moreover, such training generally has to be accomplished by GPU. This is because the implementation of back propagation (BP) in DnCNN requires a great price. Although DnCNN has good image recovery performance, this method relies heavily on data and cannot systematically

understand the learned data. In contrast, our proposed ADDF uses only three images to learn two parameters with moderate computation cost. Specifically, in ADDF it takes only 12 seconds on CPU by regression method to get the optimal confidence power and the optimum iteration number, which constitute the optimal confidence factors. It can be found that ADDF is much faster than DnCNN in terms of time-cost of training. Besides, it is surprising that our proposed ADDF has a strong generalization ability, mainly because ADDF is a model-based image restoration method. The model itself is determined by the physical properties of the image and noise.

In terms of image restoration performance, the proposed method is inferior to the large sample learning method such as DnCNN, which is the main shortcoming of the current model-based approach. Thus, one of our future research directions is to further enhance the effect of image restoration under the small sample condition, for example, for medical high-dose images. On the other hand, under the condition of large sample, the deep learning method can be used to solve the new model such as the improvement of the trainable nonlinear reaction diffusion [40], to further improve the image restoration effect.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors state that there are no conflicts of interest about the publication of this paper.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China (Grants 61472303, 61472257, and 61271294), the Fundamental Research Funds for the Central Universities (Grant NSIY21), and the HD Video R & D Platform for Intelligent Analysis and Processing in Guangdong Engineering Technology Research Centre of Colleges and Universities under Grant GCZX-A1409.

References

- [1] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.
- [2] C. Je, H. S. Jeon, C.-H. Son, and H.-M. Park, "Disparity-based space-variant image deblurring," *Signal Processing: Image Communication*, vol. 28, no. 7, pp. 792–808, 2013.
- [3] W. H. Richardson, "Bayesian-based iterative method of image restoration," *Journal of the Optical Society of America*, vol. 62, no. 1, pp. 55–59, 1972.
- [4] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," *IEEE Signal Processing Magazine*, vol. 14, no. 2, pp. 24–41, 1997.
- [5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [6] P. Chatterjee and P. Milanfar, "Is denoising dead?" *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 895–911, 2010.
- [7] A. Levin and B. Nadler, "Natural image denoising: optimality and inherent bounds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 2833–2840, Providence, RI, USA, June 2011.
- [8] A. Levin, B. Nadler, F. Durand, and W. T. Freeman, "Patch Complexity, Finite Pixel Correlations and Optimal Denoising," in *Computer Vision – ECCV 2012*, vol. 7576 of *Lecture Notes in Computer Science*, pp. 73–86, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [9] C. Knaus and M. Zwicker, "Dual-domain image denoising," in *Proceedings of the 2013 20th IEEE International Conference on Image Processing, ICIP 2013*, pp. 440–444, Australia, September 2013.
- [10] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1141–1151, 2002.
- [11] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, 2007.
- [12] K. N. Chaudhury, D. Sage, and M. Unser, "Fast O(1) bilateral filtering using trigonometric range kernels," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3376–3382, 2011.
- [13] C.-H. Lin, J.-S. Tsai, and C.-T. Chiu, "Switching bilateral filter with a texture/noise detector for universal noise removal," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2307–2320, 2010.
- [14] L. Durak and O. Arikan, "Short-time Fourier transform: two fundamental properties and an optimal implementation," *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1231–1242, 2003.
- [15] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3440–3451, 2006.
- [16] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces*, vol. 6920 of *Lecture Notes in Computer Science*, pp. 711–730, 2012.
- [17] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Low-complexity single-image super-resolution based on non-negative neighbor embedding," in *Proceedings of the 2012 23rd British Machine Vision Conference, BMVC 2012*, Guildford, UK, September 2012.
- [18] B. Chen, C. Chang, and S. Huang, "Denoising Using Inverse-Distance Weighting with Sparse Approximation," in *Proceedings of the 2016 IEEE International Symposium on Multimedia (ISM)*, pp. 439–444, San Jose, CA, USA, December 2016.
- [19] P.-H. Lin, B.-H. Chen, F.-C. Cheng, and S.-C. Huang, "A Morphological Mean Filter for Impulse Noise Removal," *Journal of Display Technology*, vol. 12, no. 4, pp. 344–350, 2016.
- [20] W. Zhao, Y. Lv, Q. Liu, and B. Qin, "Detail-Preserving Image Denoising via Adaptive Clustering and Progressive PCA Thresholding," *IEEE Access*, vol. 6, pp. 6303–6315, 2018.
- [21] V. A. Marcenko and L. A. Pastur, "Distribution of eigenvalues for some sets of random matrices," *Mathematics of the USSR-Sbornik*, vol. 1, no. 4, p. 457, 1967.

- [22] A. M. Hasan, A. Melli, K. A. Wahid, and P. Babyn, "Denoising Low-Dose CT Images Using Multiframe Blind Source Separation and Block Matching Filter," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 2, no. 4, pp. 279–287, 2018.
- [23] A. Hyvarinen, "New approximations of differential entropy for independent component analysis and projection pursuit," *Neural Information Processing Systems*, vol. 10, pp. 273–279, 1998.
- [24] M. Srivastava, C. L. Anderson, and J. H. Freed, "A New Wavelet Denoising Method for Selecting Decomposition Levels and Noise Thresholds," *IEEE Access*, vol. 4, pp. 3862–3877, 2016.
- [25] J. A. Weil and J. R. Bolton, *Electron Paramagnetic Resonance: Elementary Theory and Practical Applications*, Wiley, New York, NY, USA, 1994.
- [26] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.
- [27] C. A. Knaus and M. Zwicker, "Dual-domain filtering," *SIAM Journal on Imaging Sciences*, vol. 8, no. 3, pp. 1396–1420, 2015.
- [28] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [29] W. Zuo, D. Ren, D. Zhang, S. Gu, and L. Zhang, "Learning iteration-wise generalized shrinkage-thresholding operators for blind deconvolution," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1751–1764, 2016.
- [30] Y. Xia, H. Leung, and M. S. Kamel, "A discrete-time learning algorithm for image restoration using a novel L2-norm noise constrained estimation," *Neurocomputing*, vol. 198, pp. 155–170, 2016.
- [31] X. Jia, X. Feng, and W. Wang, "Adaptive regularizer learning for low rank approximation with application to image denoising," in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3096–3100, IEEE, Phoenix, AZ, USA, September 2016.
- [32] Q. Jiang, F. Shao, W. Lin, and G. Jiang, "On Predicting Visual Comfort of Stereoscopic Images: A Learning to Rank Based Approach," *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 302–306, 2016.
- [33] R. Liu, G. Zhong, J. Cao, Z. Lin, S. Shan, and Z. Luo, "Learning to Diffuse: A New Perspective to Design PDEs for Visual Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 2457–2471, 2016.
- [34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, Las Vegas, NV, USA, 2016.
- [35] J. Tian and L. Chen, "Image noise estimation using a variation-adaptive evolutionary approach," *IEEE Signal Processing Letters*, vol. 19, no. 7, pp. 395–398, 2012.
- [36] M. Rakhshanfar and M. A. Amer, "Estimation of GAUSSian, Poissonian-GAUSSian, and processed visual noise and its level function," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4172–4185, 2016.
- [37] M. Mäkitalo and A. Foi, "Noise parameter mismatch in variance stabilization, with an application to Poisson-Gaussian noise estimation," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5348–5359, 2014.
- [38] X. Zhang, W. Lin, R. Xiong, X. Liu, S. Ma, and W. Gao, "Low-rank decomposition-based restoration of compressed images via adaptive noise estimation," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4158–4171, 2016.
- [39] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: fixed-point convergence and applications," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 84–98, 2017.
- [40] Y. Chen and T. Pock, "Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2017.
- [41] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pp. 2392–2399, Rhode Island, USA, June 2012.
- [42] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 2862–2869, IEEE, Columbus, OH, USA, June 2014.
- [43] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *International Journal of Computer Vision*, vol. 121, no. 2, pp. 183–208, 2017.
- [44] S. Pyatykh, J. Hesser, and L. Zheng, "Image noise level estimation by principal component analysis," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 687–699, 2013.
- [45] M. M. Ghazi and H. Erdogan, "Image noise level estimation based on higher-order statistics," *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 2379–2397, 2017.
- [46] R. J. Marks, G. L. Wise, D. G. Haldeman, and J. L. Whited, "Detection In Laplace Noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 14, no. 6, pp. 866–872, 1978.
- [47] C. W. Helstrom, "Detectability of Signals in Laplace Noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 25, no. 2, pp. 190–196, 1989.
- [48] M. W. Thompson and H.-S. Chang, "Coherent Detection in Laplace Noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 2, pp. 452–461, 1994.
- [49] S. Jiang and N. C. Beaulieu, "Precise BER computation for binary data detection in bandlimited white laplace noise," *IEEE Transactions on Communications*, vol. 59, no. 6, pp. 1570–1579, 2011.
- [50] J.-J. Moreau, "Fonctions convexes duales et points proximaux dans un espace hilbertien," *Comptes Rendus Mathematique*, vol. 255, pp. 2897–2899, 1962.
- [51] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.
- [52] K. Egiazarian and V. Katkovnik, "Single image super-resolution via BM3D sparse coding," in *Proceedings of the 23rd European Signal Processing Conference, EUSIPCO 2015*, pp. 2849–2853, Nice, France, September 2015.
- [53] M. Irani and S. Peleg, "Motion analysis for image enhancement: resolution, occlusion, and transparency," *Journal of Visual Communication and Image Representation*, vol. 4, no. 4, pp. 324–335, 1993.

