GROUP SPARSITY RESIDUAL WITH NON-LOCAL SAMPLES FOR IMAGE DENOISING

Zhiyuan Zha1,†, Xinggan Zhang1, Qiong Wang1,∗, Yechao Bai1, Lan Tang1,2 and Xin Yuan5,†

1School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China.
2National Mobile Commun. Research Lab., Southeast University, Nanjing 210023, China.
3Nokia Bell Labs, 600 Mountain Avenue, Murray Hill, NJ, 07974, USA.

ABSTRACT

Inspired by group-based sparse coding, recently proposed group sparsity residual (GSR) scheme has demonstrated superior performance in image processing. However, one challenge in GSR is to estimate the residual by using a proper reference of the group-based sparse coding (GSC), which is desired to be as close to the truth as possible. Previous researches utilized the estimations from other algorithms (i.e., GMM or BM3D), which are either not accurate or too slow. In this paper, we propose to use the Non-Local Samples (NLS) as reference in the GSR regime for image denoising, thus termed GSR-NLS. More specifically, we first obtain a good estimation of the group sparse coefficients by the image nonlocal self-similarity, and then solve the GSR model by an effective iterative shrinkage algorithm. Experimental results demonstrate that the proposed GSR-NLS not only outperforms many state-of-the-art methods, but also delivers the competitive advantage of speed.

Index Terms— Image denoising, group-based sparse coding, group sparsity residual, nonlocal self-similarity, iterative shrinkage algorithm.

1. INTRODUCTION

Image denoising plays an important role in various image processing tasks. In general, image denoising aims to restore the clean image \( y \) from its corrupted observation \( z = y + v \), where \( v \) is usually considered to be an additive white Gaussian noise. Image denoising problem is mathematically ill-posed and priors are thus usually employed to achieve good results. Over the past few years, numerous image prior models have been developed, including total variation based [1, 2], wavelet/curvelet based [3–6], sparse coding based [7, 8], nonlocal self-similarity based [9–11], and deep learning based [12, 13] ones.

One significant advance in image processing is to model the prior on patches, and a representative research is sparse coding [7, 8], which assumes that each patch of an image can be precisely modeled by a sparse linear combination of some fixed and trainable basis elements, which are therefore called atoms and these atoms compose a dictionary. The seminal work of K-SVD dictionary learning method [8] has not only shown promising denoising performance, but also been extended to other image processing and computer vision tasks [14, 15]. Meanwhile, since image patches with similar structures can be spatially far from each other and thus can be collected across the whole image, the so-called nonlocal self-similarity (NSS) prior is among the most remarkable priors for image restoration [10, 16–25]. The seminal work of nonlocal means (NLM) [9] exploited the NSS prior to carry out a form of the weighted filtering for image denoising. Compared with the local regularization methods (e.g., total variation based method [1]), the nonlocal regularization based methods can retain the image edges and the sharpness effectively. Inspired by the success of the NSS prior, group-based sparse coding (GSC) has attracted considerable interests in image denoising [18, 19, 22–25]. However, due to the influence of noise, the conventional GSC model is accurate enough to restore the original image.

Most recently, the group sparsity residual [26, 27] has been proposed for image denoising, which adopts a reference in each iteration to approximate the true sparse coefficients of each group. The reference was estimated by Gaussian mixture model (GMM) [23, 28] or other algorithms, which is either not accurate or too slow. In this paper, we propose a new method for image denoising via group sparsity residual scheme with non-local samples (GSR-NLS). We first obtain a good estimation of the group sparse coefficients of the original image by the image nonlocal self-similarity, and then the group sparse coefficients of the noisy image are inferred to approximate this estimation. Moreover, we develop an effective iterative shrinkage algorithm to solve the proposed GSR-NLS model. Experimental results show that the proposed GSR-NLS not only outperforms many state-of-the-art methods in terms of the objective and the perceptual metrics, but also delivers a competitive speed.

2. GROUP-BASED SPARSE CODING FOR IMAGE DENOISING

In this section, we will briefly introduce the conventional group-based sparse coding (GSC) model for image denoising. Specifically, taking a clean image \( y \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}} \), as an example, it is divided into \( n \) overlapped patches of size \( \sqrt{b} \times \sqrt{b} \), and each patch is denoted by a vector \( y_i \in \mathbb{R}^b \), \( i = 1, 2, \ldots, n \). Then for each patch \( y_i \), its \( m \) similar patches are selected from a searching window with \( W \times W \) pixels to form a set \( S_i \). Following this, all patches in \( S_i \) are stacked into a matrix \( Y_i \in \mathbb{R}^{b \times m} \), i.e., \( Y_i = \{y_{i,1}, y_{i,2}, \ldots, y_{i,m}\} \). The matrix \( Y_i \) consisting of patches with similar structures is thereby called a group, where \( \{y_{i,j}\}_{j=1}^m \) denotes the \( j \)-th patch in the \( i \)-th group. Following this, similar to patch-based sparse coding [7, 8], given a dictionary \( D_i \), each group \( Y_i \) can be sparsely represented and solved by the following minimization problem,

\[
B_i = \arg \min_{B_i} \{||Y_i - D_i B_i||_F^2 + \lambda ||B_i||_1\}, \tag{1}
\]

where \( \lambda \) is the regularization parameter; \( || \cdot ||_F \) denotes the Frobenius norm. Here, \( \ell_1 \)-norm is imposed on each column of \( B_i \), which also holds true for the following derivation with \( \ell_1 \)-norm on matrix.
In image denoising, each patch $z_i$ is extracted from the noisy image $z$ and we search for its $m$ similar patches to generate a noisy image patch group $Z_i \in \mathbb{R}^{b \times m}$, i.e., $Z_i = \{z_{i,1}, z_{i,2}, \ldots, z_{i,m}\}$. Then, image denoising is translated into how to restore the $Y_i$ from $Z_i$ using the GSC model,

$$A_i = \arg \min_{A_i} \left( ||Z_i - DA_i||_F^2 + \lambda ||A_i||_1 \right).$$

Once all group sparse codes $\{A_i\}_{i=1}^n$ are obtained, the underlying clean image $\hat{y}$ can be reconstructed as $\hat{y} = DA_i$ where $A$ denotes the set of $\{A_i\}_{i=1}^n$. However, due to the noise, the conventional GSC mode of Eq. (2) cannot recover the underlying image $y$ accurately.

### 3. Denoising via Group Sparsity Residual with Non-Local Samples

Recalling Eq. (1) and Eq. (2), the noisy group sparse coefficient $A_i$ obtained by solving Eq. (2) is excepted to be as close to the true group sparse coefficient $B_i$ of the original image $y$ in Eq. (1) as possible. Accordingly, the quality of image denoising largely depends on the group sparsity residual, which is defined as the difference between the noisy group sparse coefficient $A_i$ and the true group sparse coefficient $B_i$,

$$R_i = A_i - B_i.$$  \hspace{1cm} (3)

In order to obtain a good performance in image denoising, we hope that the group sparsity residual $R_i$ of each group is as small as possible. To this end, to reduce the group sparsity residual $R = \{R_i\}_{i=1}^n$ and boost the accuracy of $A$, we propose the group sparsity residual (GSR) based model for image denoising. This is

$$A_i = \arg \min_{A_i} \left( ||Z_i - DA_i||_F^2 + \lambda ||A_i - B_i||_p \right).$$  \hspace{1cm} (4)

Similarly, $\ell_p$-norm is imposed on each column of $R_i$, which also holds true for the following derivation with $\ell_p$-norm on matrix. Since the original image $y$ is not available, it is impossible to get the true sparse coefficient $B_i$. We will describe how to estimate $B_i$ and $p$ below.

### 3.1. Determine $p$

Let us come back to Eq. (4), it is clear that one important problem of the proposed GSR based image denoising is the determination of $p$. Here, we conduct some experiments to investigate the statistical property of $R = \{R_i\}_{i=1}^n$. We compute $A = \{A_i\}_{i=1}^n$ and $B = \{B_i\}_{i=1}^n$ by solving Eq. (2) and Eq. (1), respectively. The principle component analysis (PCA) based dictionary [16] is used in these experiments. One typical image $House$ is used as example, where Gaussian white noise is added with standard deviation $\sigma_n = 30$. We plot the empirical distribution of $R$ as well as the fitting Gaussian, Laplacian and hyper-Laplacian distributions in Fig. 1(a). To better observe the fitting of the tails, we also plot these distributions in the log domain in Fig. 1(b). It can be seen that the empirical distribution of $R$ can be well characterized by the Laplacian distribution. Therefore, we set $p = 1$ and thus the $\ell_1$-norm is adopted to regularize the proposed GSR model,

$$A_i = \arg \min_{A_i} \left( ||Z_i - DA_i||_F^2 + \lambda ||A_i - B_i||_1 \right).$$  \hspace{1cm} (5)

### 3.2. Estimate of the Unknown Group Sparse Coefficient $B$

Since the original image $y$ is not available in real applications, the true group sparse coefficient $B_i$ is unknown. Thus, we need to estimate $B_i$ in Eq. (5). In general, there are a variety of methods to estimate $B_i$, which depends on the prior knowledge of the original image $y$. Different from [26, 27] in this paper, based on the fact that natural images often contain repetitive structures [29], we search nonlocal similar patches (i.e., non-local samples) to the given patch and use the method similar to nonlocal means [9] to estimate $B_i$. Specifically, a good estimation of $b_{i,1}$ can be computed by the weighted average of each element $a_{i,j}$ in $A_i$ associated with each group including $m$ nonlocal similar patches, where $b_{i,1}$ and $a_{i,j}$ represent the first and the $j$-th element of $B_i$ and $A_i$, respectively. Then we have

$$b_{i,1} = \sum_{j=1}^m w_{i,j} a_{i,j},$$  \hspace{1cm} (6)

where $w_{i,j}$ is the weight, which is inversely proportional to the distance between patches $z_i$ and $z_{i,j}$; $w_{i,j} = \exp(-||z_i - z_{i,j}||_2^2/h/L)$, where $h$ is a predefined constant and $L$ is a normalization factor [9]. After this, we simply copy $b_{i,1}$ by $m$ times to estimate $B_i$, i.e.,

$$B_i = \{b_{i,1}, b_{i,2}, \ldots, b_{i,m}\},$$  \hspace{1cm} (7)

where $\{b_{i,j}\}_{j=1}^m$ denotes the $j$-th element in the $i$-th group sparse coefficient $B_i$ and they are the same. The flowchart of the proposed GSR-NLS model for image denoising is illustrated in Fig. 2. Another important issue of the proposed GSR-NLS based image denoising is the selection of the dictionary. To adapt to the local image structures, instead of learning an over-complete dictionary for each group $Z_i$ as in [24], we utilized the PCA based dictionary [16].

### 3.3. Iterative Shrinkage Algorithm to Solve the Proposed GSR-NLS Model

Due to orthogonality (obtained by PCA) of each dictionary $D_i$, Eq. (5) can be rewritten as

$$A_i = \arg \min_{A_i} \{ ||G_i - A_i||_F^2 + \lambda ||A_i - B_i||_1 \} = \arg \min_{A_i} \{ ||g_i - \alpha_i||_F^2 + \lambda ||\alpha_i - \beta_i||_1 \},$$  \hspace{1cm} (8)

where $Z_i = D_i G_i$; $\alpha_i$, $\beta_i$ and $g_i$ denote the vectorization of the matrix $A_i$, $B_i$ and $G_i$, respectively.

To solve Eq. (8) effectively, an iterative shrinkage algorithm [30] is adopted. To be concrete, for fixed $g_i$, $\beta_i$, and $\lambda$, we have

$$\alpha_i = \text{soft}(g_i - \beta_i, \lambda) + \beta_i,$$  \hspace{1cm} (9)

where soft $(\cdot)$ is the soft-thresholding operator [30]. Following this, the latent clean patch group $Y_i$ can be calculated by $Y_i = D_i A_i$. After obtaining the estimate of all groups $\{Y_i\}$, we get the full image $\hat{y}$ by putting the groups back to their original image.
locations and averaging the overlapped pixels. Moreover, we could execute the above denoising procedures several iterations for better results. In the k-th iteration, the iterative regularization strategy is utilized to update the estimation of the noise variance, and thus updating $z^k$. The standard deviation of noise in k-th iteration is adjusted as $\sigma_n^k = \gamma \sqrt{\sigma_n^2 - ||z - y^{k-1}||_2^2}$, where $\gamma$ is a constant.

The parameter $\lambda$ that balances the fidelity term and the regularization term should be adaptively determined in each iteration. Inspired by [3], the regularization parameter $\lambda$ of each noisy group $Z_i$ is set to $\lambda = c \times \sqrt{2\pi \sigma_n^2}/(\delta_i + \varepsilon)$, where $\delta_i$ denotes the estimated variance of $R_i$ and $c, \varepsilon$ are small constants.

Throughout the numerical experiments, we choose the following stopping iteration for the proposed denoising algorithm, i.e., $||y^k - y^{k-1}||_2^2/||y^{k-1}||_2^2 < \tau$, where $\tau$ is a small constant. The complete description of the proposed GSR-NLS for image denoising is exhibited in Algorithm 1.

**Algorithm 1 The Proposed GSR-NLS for Image Denoising.**

Require: Noisy image $z$.
1: Initialize $\hat{y}^0 = z$, $z^0 = z$, $\sigma_n$, $b$, $c$, $m$, $h$, $W$, $\gamma$, $\eta$, $\tau$ and $\varepsilon$.
2: for $k = 1$ to Max-Iter do
3: Iterative Regularization $z^k = y^{k-1} + \gamma (z - y^{k-1})$.
4: for Each patch $z_i$ in $z^k$ do
5: Find nonlocal similar patches to form a group $Z_i$.
6: Constructing dictionary $D_i$ by $Z_i$ using PCA.
7: Update $A_i$ by computing $A_i = D_i^{-1} Z_i$.
8: Estimate $B_i$ by Eq. (6) and Eq. (7).
9: Update $\lambda$ by computing $\lambda = c \times 2\sqrt{2\sigma_n^2}/(\delta_i + \varepsilon)$.
10: Estimate $A_i$ by Eq. (9).
11: Get the estimation: $\hat{Y}_i = D_i \hat{A}_i$.
12: end for
13: Aggregate $Y_i$ to form the denoised image $\hat{y}^k$.
14: end for
15: Output: The final denoised image $\hat{y}$.

### 3.4. Different from Existing Methods

Now we discuss the difference between the proposed GSR-NLS method, and NCSR [31] and our previous approaches [26, 27].

The main difference between NCSR [31] and the proposed GSR-NLS is that NCSR is essentially a patch-based sparse coding method, which usually ignored the relationship among similar patches [18, 22, 24, 32]. In addition, NCSR extracted image patches from noisy image $z^k$ and used $K$-means algorithm to generate $K$ clusters. Following this, it learned $K$ PCA sub-dictionaries from each cluster. However, since each cluster includes thousands of patches, the dictionary learned by PCA from each cluster may not accurately centralize the image features. The proposed GSR-NLS learned the PCA dictionary from each group and the patches in each group are similar, and therefore, our PCA dictionary is more appropriate. An additional advantage of GSR-NLS is that it only requires 1/3 computational time of NCSR but achieves $\sim 0.5$dB improvement on average over NCSR (see Section 4 for details).

Our previous work in [26] utilized the pre-filtering BM3D [10] to estimate $B$, and thus the denoising performance largely depend on the pre-filtering. In other words, if the pre-filtering cannot be used, the denoising method will fail. Our another work [27] estimated $B$ from example image set based on GMM [28]. However, under many practical situations, the example image set is simply unavailable. Moreover, the computation speed of this method is very slow. Therefore, the proposed GSR-NLS is more reasonable and generic while providing very similar results as in [26, 27].

### 4. EXPERIMENTAL RESULTS

In this section, we report the performance of the proposed GSR-NLS method for image denoising and compare it with several state-of-the-art denoising methods, including BM3D [10], EPLL [28], Plow [33], NCSR [31], PID [34], PGPD [23], aGMM [35], LINC [36] and AST-NLS [37]. The parameter setting of the proposed GSR-NLS is as follows. The searching window $W \times W$ is set to be $25 \times 25$ and $\varepsilon$ is set to 0.2. The size of patch $\sqrt{b} \times \sqrt{b}$ is set to be $6 \times 6, 7 \times 7, 8 \times 8$ and $9 \times 9$ for $\sigma_n \leq 20, 20 < \sigma_n \leq 50, 50 < \sigma_n \leq 75$ and $75 < \sigma_n \leq 100$, respectively. The parameters $(c, \eta, \gamma, m, h, \tau)$ are set to $(0.8, 0.2, 0.5, 60, 45, 0.0003), (0.7, 0.2, 0.6, 60, 45, 0.0008), (0.6, 0.1, 0.6, 60, 60, 0.0002), (0.7, 0.1, 0.5, 80, 80, 0.0002), (0.7, 0.1, 0.5, 80, 115, 0.0001), (0.7, 0.1, 0.5, 90, 160, 0.0005)$ and $(1, 0.1, 0.5, 100, 160, 0.0005)$ for $\sigma_n \leq 10, 10 < \sigma_n \leq 20, 20 < \sigma_n \leq 30, 30 < \sigma_n \leq 40, 40 < \sigma_n \leq 50, 50 < \sigma_n \leq 75$ and $75 < \sigma_n \leq 100$, respectively. The test images are displayed in Fig. 3. The source code of the proposed GSR-NLS for image denoising can be downloaded at: https://drive.google.com/open?id=0B0wKhHwcknCjZkh4NkprbVhBMFk.

Due to the page limit, we only present the denoising results at four noise levels, i.e., Gaussian white noise with standard deviations $(\sigma_n = 20, 40, 50, 75)$. As shown in Table 1, the proposed GSR-NLS outperforms the other competing methods in most cases in terms of PSNR. The average gains of the proposed GSR-NLS over BM3D, EPLL, Plow, NCSR, PID, PGPD, aGMM, LINC and.
This paper has proposed a new method for image denoising using group sparsity residual scheme with nonlocal samples. We obtained a good estimation of the group sparsity coefficients of the original image from the image nonlocal self-similarity and used it in the group sparsity residual model. An effective iterative shrinkage algorithm has been developed to solve the proposed GSR-NLS model. Experimental results have demonstrated that the proposed GSR-NLS not only outperforms many state-of-the-art methods, but also leads to a competitive speed.

5. CONCLUSION

AST-NLS are as much as 0.40dB, 0.94dB, 0.85dB, 0.45dB, 0.31dB, 0.25dB, 0.73dB, 0.33dB and 0.16dB, respectively. The visual comparisons of different denoising methods with two example images are shown in Fig. 4 and Fig. 5. It can be seen that BM3D, PID, LINC and AST-NLS are leading to over-smooth phenomena, while EPLL, PID, NCSR, PGPD and aGMM are likely to produce some undesirable ringing artifacts. By contrast, the proposed GSR-NLS is able to preserve the image local structures and suppress undesirable ringing artifacts more effectively than the other competing methods.

### Table 1. PSNR (dB) results of different denoising methods.

<table>
<thead>
<tr>
<th>Images</th>
<th>BM3D</th>
<th>EPLL</th>
<th>Plow</th>
<th>NCSR</th>
<th>PID</th>
<th>PGPD</th>
<th>aGMM</th>
<th>LINC</th>
<th>AST-NLS</th>
<th>GSR-NLS</th>
<th>BM3D</th>
<th>EPLL</th>
<th>Plow</th>
<th>NCSR</th>
<th>PID</th>
<th>PGPD</th>
<th>aGMM</th>
<th>LINC</th>
<th>AST-NLS</th>
<th>GSR-NLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>25.76</td>
<td>25.96</td>
<td>25.64</td>
<td>25.63</td>
<td>25.09</td>
<td>25.43</td>
<td>26.04</td>
<td>26.02</td>
<td>26.17</td>
<td>23.99</td>
<td>26.88</td>
<td>27.08</td>
<td>26.78</td>
<td>27.25</td>
<td>27.13</td>
<td>26.95</td>
<td>27.08</td>
<td>27.01</td>
<td>27.02</td>
<td>26.97</td>
</tr>
<tr>
<td>Barbara</td>
<td>31.24</td>
<td>30.60</td>
<td>29.48</td>
<td>31.10</td>
<td>30.98</td>
<td>31.12</td>
<td>30.51</td>
<td>31.78</td>
<td>34.14</td>
<td>31.47</td>
<td>27.26</td>
<td>27.59</td>
<td>27.59</td>
<td>27.25</td>
<td>27.68</td>
<td>27.43</td>
<td>26.94</td>
<td>27.77</td>
<td>27.41</td>
<td>27.85</td>
</tr>
<tr>
<td>Boat</td>
<td>31.42</td>
<td>30.87</td>
<td>30.99</td>
<td>31.26</td>
<td>31.27</td>
<td>31.38</td>
<td>31.52</td>
<td>31.50</td>
<td>31.25</td>
<td>27.76</td>
<td>27.62</td>
<td>27.55</td>
<td>27.52</td>
<td>27.73</td>
<td>27.90</td>
<td>27.00</td>
<td>27.96</td>
<td>27.80</td>
<td>27.97</td>
<td></td>
</tr>
<tr>
<td>Fence</td>
<td>25.93</td>
<td>25.84</td>
<td>24.13</td>
<td>30.05</td>
<td>30.05</td>
<td>29.39</td>
<td>29.46</td>
<td>30.08</td>
<td>30.28</td>
<td>31.30</td>
<td>26.94</td>
<td>27.51</td>
<td>26.62</td>
<td>26.46</td>
<td>26.96</td>
<td>26.94</td>
<td>26.91</td>
<td>27.00</td>
<td>27.11</td>
<td>26.16</td>
</tr>
<tr>
<td>Foreman</td>
<td>34.54</td>
<td>33.67</td>
<td>34.21</td>
<td>34.42</td>
<td>34.63</td>
<td>34.40</td>
<td>34.20</td>
<td>34.76</td>
<td>35.55</td>
<td>34.67</td>
<td>29.32</td>
<td>30.28</td>
<td>30.90</td>
<td>31.52</td>
<td>31.81</td>
<td>31.55</td>
<td>30.95</td>
<td>31.31</td>
<td>31.29</td>
<td>31.81</td>
</tr>
<tr>
<td>House</td>
<td>33.77</td>
<td>32.99</td>
<td>31.40</td>
<td>33.81</td>
<td>33.70</td>
<td>33.85</td>
<td>33.52</td>
<td>33.82</td>
<td>33.87</td>
<td>33.91</td>
<td>30.63</td>
<td>29.89</td>
<td>30.25</td>
<td>30.79</td>
<td>30.76</td>
<td>31.02</td>
<td>30.40</td>
<td>31.00</td>
<td>30.91</td>
<td>31.16</td>
</tr>
<tr>
<td>Lena</td>
<td>31.52</td>
<td>31.25</td>
<td>30.98</td>
<td>31.48</td>
<td>31.53</td>
<td>31.64</td>
<td>31.47</td>
<td>31.63</td>
<td>31.61</td>
<td>31.73</td>
<td>27.82</td>
<td>27.78</td>
<td>27.88</td>
<td>28.00</td>
<td>28.18</td>
<td>28.22</td>
<td>27.91</td>
<td>28.13</td>
<td>28.00</td>
<td>28.16</td>
</tr>
<tr>
<td>Monarch</td>
<td>30.35</td>
<td>30.49</td>
<td>29.50</td>
<td>30.52</td>
<td>30.59</td>
<td>30.68</td>
<td>30.31</td>
<td>30.46</td>
<td>30.84</td>
<td>30.98</td>
<td>26.72</td>
<td>26.89</td>
<td>26.43</td>
<td>26.81</td>
<td>27.27</td>
<td>27.02</td>
<td>26.87</td>
<td>27.14</td>
<td>27.20</td>
<td>27.33</td>
</tr>
<tr>
<td>Average</td>
<td>31.31</td>
<td>30.79</td>
<td>30.67</td>
<td>31.34</td>
<td>31.30</td>
<td>31.42</td>
<td>31.10</td>
<td>31.47</td>
<td>31.56</td>
<td>31.64</td>
<td>27.70</td>
<td>27.28</td>
<td>27.48</td>
<td>27.18</td>
<td>27.98</td>
<td>27.97</td>
<td>27.48</td>
<td>27.97</td>
<td>27.99</td>
<td>27.97</td>
</tr>
</tbody>
</table>

### Table 2. Average running time (s) on the 10 test images.

<table>
<thead>
<tr>
<th>Methods</th>
<th>BM3D</th>
<th>EPLL</th>
<th>Plow</th>
<th>NCSR</th>
<th>PID</th>
<th>PGPD</th>
<th>aGMM</th>
<th>LINC</th>
<th>AST-NLS</th>
<th>GSR-NLS</th>
</tr>
</thead>
</table>

Efficiency is another key factor in evaluating a denoising algorithm. To evaluate the computational cost of the competing algorithms, we compare the running time on 10 test images with different noise levels. All experiments are conducted under the Matlab 2015b environment on a computer with Intel (R) Core(TM) i3-4150 with 3.56GHz CPU and 4GB memory. The average running time (in seconds) of different methods is shown in Table 2. It can be seen that the proposed GSR-NLS uses less computation time than the competing methods except for BM3D, EPLL and PGPD. However, BM3D is implemented with compiled C++-mex function and is performed in parallel. EPLL and PGPD are based on learning methods, which require a significant amount of time in learning stage. Therefore, the proposed GSR-NLS enjoys a competitive speed.

Fig. 4. Denoising results on image boats by different methods (noise level $\sigma_n = 75$). (a) Original image; (b) Noisy image; (c) BM3D [10] (PSNR = 24.82dB); (d) EPLL [28] (PSNR = 24.33dB); (e) Plow [33] (PSNR = 24.23dB); (f) NCSR [31] (PSNR = 24.44dB); (g) PID [34] (PSNR = 24.51dB); (h) PGPD [23] (PSNR = 24.83dB); (i) aGMM [35] (PSNR = 24.91dB); (j) LINC [36] (PSNR = 24.44dB); (k) AST-NLS [37] (PSNR = 24.76dB); (l) GSR-NLS (PSNR = 24.94dB).

Fig. 5. Denoising results on image House by different methods (noise level $\sigma_n = 75$). (a) Original image; (b) Noisy image; (c) BM3D [10] (PSNR = 27.51dB); (d) EPLL [28] (PSNR = 26.70dB); (e) Plow [33] (PSNR = 26.52dB); (f) NCSR [31] (PSNR = 27.16dB); (g) PID [34] (PSNR = 27.35dB); (h) PGPD [23] (PSNR = 27.81dB); (i) aGMM [35] (PSNR = 27.11dB); (j) LINC [36] (PSNR = 27.56dB); (k) AST-NLS [37] (PSNR = 28.06dB); (l) GSR-NLS (PSNR = 28.59dB).
6. REFERENCES


