Supplementary Materials to “A Blind Denoising Algorithm for Real Noisy Images”

YISU ZHOU[1], YUHAO DOU[1], YI YAO[2], BO XIN[1], and ZHANGQING ZHU[1],
[1]School of Management and Engineering, Nanjing University, Nanjing 210093, China
[2]Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, United Kingdom

This work was supported by the National Key Research and Development Program (NO.2016YFD0702100), the National Natural Science Foundation of China (Nos.61432008 and 71732003).

In this supplementary file, we provide:
Part 1. The proof of the Theorem 1 in the main paper;
Part 2. The detailed performance of the noise level estimator.
Part 3. The reason we choose the region size (256*256) in our paper.

Part 1. The proof of Theorem 1：

1. Denote by $\mathrm{U}_{\mathrm{k-1}}\Sigma_{\mathrm{k-1}}\mathrm{V}_{\mathrm{k-1}}^{\mathrm{T}}$ is the SVD of the matrix $\boldsymbol{D}-\boldsymbol{S}_k+\mu_{k-1}^{-1}\boldsymbol{Y}_{k-1}$ in the k-th iteration, where $\mathit{\Sigma}_k=diag\left(\sigma_k^1,\sigma_k^2,\dots\dots,\sigma_k^n\right)$ is the diagonal singular value matrix. According to the corollary in Weighted Nuclear Norm Minimization and Its Applications to Low Level Vision [Gu, Shuhang and Xie, Qi and Meng, Deyu and Zuo, Wangmeng and Feng, Xiangchu and Zhang, Lei: Weighted Nuclear Norm Minimization and Its Applications to Low Level Vision. International Journal of Computer Vision. 121(2), 183-208(2017)], we have

$$\boldsymbol{L}_{k+1}=U_k\mathit{\Lambda}_kV_k^T,$$

where $\mathit{\Lambda}_k=S_{\frac{w}{\mu_k}}\mathit{\Sigma}_k$. According to the Lagrange multiplier updating method in Algorithm 2, we have

$$\begin{aligned}\|\boldsymbol{Y}_{k+1}\|_F&=\|\boldsymbol{Y}_k+\mu_k(\boldsymbol{D}-\boldsymbol{L}_{k+1}-\boldsymbol{S}_{k+1})\|_F\\&=\mu_k\left\|\mu_k^{-1}\boldsymbol{Y}_k+\boldsymbol{D}-\boldsymbol{L}_{k+1}-\boldsymbol{S}_{k+1}\right\|_F\\&=\mu_k\left\|U_k\mathit{\Sigma}_kV_k^T-U_k\mathit{\Lambda}_kV_k^T\right\|_F\\&=\mu_k\|\mathit{\Sigma}_k-\mathit{\Lambda}_k\|_F\\&=\mu_k\left\|\mathit{\Sigma}_k-S_{\frac{w}{\mu_k}}(\mathit{\Sigma}_k)\right\|_F\\&\le\mu_k\sqrt{\mathit{\Sigma}_i\left(\frac{w_i}{\mu_k}\right)^2}\\&=\sqrt{\mathit{\Sigma}_iw_i^2}\end{aligned}$$

Hence, sequence $\{\boldsymbol{Y}_k\}$ is bounded.

2. Secondly, we prove that the sequence of $\mathcal{L}(\boldsymbol{L}_{k+1},\boldsymbol{S}_{k+1},\boldsymbol{Y}_{k+1},\mu_k)$ is also upper bounded. The globally optimal solutions of $\boldsymbol{L}$ and $\boldsymbol{S}$ subproblems ensures that:

$$\mathcal{L}(\boldsymbol{L}_{k+1}, \boldsymbol{S}_{k+1}, \boldsymbol{Y}_{k+1}, \mu_{k+1}) \leq \mathcal{L}(\boldsymbol{L}_k, \boldsymbol{S}_k, \boldsymbol{Y}_k, \mu_k)$$

Then, based on the updating method of $\boldsymbol{Y}_k$:

$$\boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \mu_k(\boldsymbol{D} - \boldsymbol{L}_{\boldsymbol{k+1}} - \boldsymbol{S}_{k+1})$$

We have

$$\begin{aligned}
\mathcal{L}(\boldsymbol{L}_k, \boldsymbol{S}_k, \boldsymbol{Y}_k, \mu_k) &= \mathcal{L}(\boldsymbol{L}_{k-1}, \boldsymbol{S}_{k-1}, \boldsymbol{Y}_{k-1}, \mu_{k-1}) + \frac{\mu_k - \mu_{k-1}}{2} \|\boldsymbol{D} - \boldsymbol{L}_k - \boldsymbol{S}_k\|_F^2 + \\
&\quad < \boldsymbol{Y}_k - \boldsymbol{Y}_{k-1}, \boldsymbol{D} - \boldsymbol{L}_k - \boldsymbol{S}_k > \\
&= \mathcal{L}(\boldsymbol{L}_{k-1}, \boldsymbol{S}_{k-1}, \boldsymbol{Y}_{k-1}, \mu_{k-1}) + \frac{\mu_k - \mu_{k-1}}{2} \|\boldsymbol{D} - \boldsymbol{L}_k - \boldsymbol{S}_k\|_F^2 + \\
&\quad < \boldsymbol{Y}_k - \boldsymbol{Y}_{k-1}, \mu_{k-1}^{-1}(\boldsymbol{Y}_k - \boldsymbol{Y}_{k-1}) > \\
&= \mathcal{L}(\boldsymbol{L}_{k-1}, \boldsymbol{S}_{k-1}, \boldsymbol{Y}_{k-1}, \mu_{k-1}) + \frac{\mu_k + \mu_{k-1}}{2\mu_{k-1}^2} \|\boldsymbol{Y}_k - \boldsymbol{Y}_{k-1}\|^2 \\
&\leq \mathcal{L}(\boldsymbol{L}_1, \boldsymbol{S}_1, \boldsymbol{Y}_0, \mu_0) + a \sum_{k=1}^{\infty} \frac{\mu_k + \mu_{k-1}}{2\mu_{k-1}^2}
\end{aligned}$$

Where a is the upper bound of $\|\boldsymbol{Y}_{k+1} - \boldsymbol{Y}_k\|_F^2$ for all $\{k = 1, \dots, \infty\}$. Recalling that the sequence $\{\mu_k\}$ is unbounded, thus $\sum_{k=1}^{\infty} \frac{\mu_k + \mu_{k-1}}{2\mu_{k-1}^2} < +\infty$. Therefore, the Lagrangian function $\mathcal{L}(\boldsymbol{L}_k, \boldsymbol{S}_k, \boldsymbol{Y}_k, \mu_k)$ is also upper bounded.

3. Thirdly, we prove that the sequences $\{\boldsymbol{L}_k\}$ and $\{\boldsymbol{S}_k\}$ are also upper bounded.

$$\|\boldsymbol{L}_k\|_{w,*} + \lambda \|\boldsymbol{W}\boldsymbol{S}_k\|_1 = \mathcal{L}(\boldsymbol{L}_{k-1}, \boldsymbol{S}_{k-1}, \boldsymbol{Y}_{k-1}, \mu_{k-1}) - \frac{1}{2\mu_{k-1}} (\|\boldsymbol{L}_k\|_F^2 - \|\boldsymbol{L}_{k-1}\|_F^2)$$

Thus, $\{\boldsymbol{L}_k\}$ and $\{\boldsymbol{W}\boldsymbol{S}_k\}$ generated by the proposed algorithm are all bounded, since the weighting matrix $\boldsymbol{W}$ has a precise value in each iteration, therefore, sequence $\{\boldsymbol{S}_k\}$ is also upper bounded. Given that, we obtain

$$\lim_{k\to\infty} \|\boldsymbol{D} - \boldsymbol{L}_{k+1} - \boldsymbol{S}_{k+1}\|_F = \lim_{k\to\infty} \frac{1}{\mu_k} \|\boldsymbol{Y}_{k+1} - \boldsymbol{Y}_k\|_F = 0$$

Hence, (a) in Theorem 1 is proved.

4. Then we prove that

$$\begin{aligned}
&\lim_{k\to\infty} \|\boldsymbol{S}_{k+1} - \boldsymbol{S}_k\|_F \\
&\quad = \lim_{k\to\infty} \left\| S_{\frac{1}{\mu_k}}\left(\boldsymbol{D} + \mu_k^{-1}\boldsymbol{Y}_k - \boldsymbol{L}_k\right) - \left(\boldsymbol{D} + \mu_k^{-1}\boldsymbol{Y}_k - \boldsymbol{L}_k\right) - 2\mu_k^{-1}\boldsymbol{Y}_k - \mu_k^{-1}\boldsymbol{Y}_{k-1} \right\|_F \\
&\quad \leq \lim_{k\to\infty} \frac{mn}{\mu_k} + \left\| 2\mu_k^{-1}\boldsymbol{Y}_k + \mu_k^{-1}\boldsymbol{Y}_{k-1} \right\|_F \\
&\quad = 0
\end{aligned}$$

In which is the soft-thresholding operation with parameter $1/\mu_k$ and m and *n* are the size of matrix $\boldsymbol{D}$. Thus, (b) is proved.

5. Finally, recalling the updating strategy in Algorithm 2, we have the following inequalities:

$$\lim_{k\to\infty}\|\boldsymbol{L}_{k+1}-\boldsymbol{L}_k\|_F=\lim_{k\to\infty}\left\|(\boldsymbol{D}+\mu_k^{-1}\boldsymbol{Y}_k-\boldsymbol{S}_{k+1}-\mu_k^{-1}\boldsymbol{Y}_{k+1})-\boldsymbol{L}_k\right\|_F$$

$$=\lim_{k\to\infty}\left\|(\boldsymbol{D}+\mu_k^{-1}\boldsymbol{Y}_k-\boldsymbol{S}_{k+1}-\mu_k^{-1}\boldsymbol{Y}_{k+1})-\boldsymbol{L}_k+\left(\boldsymbol{S}_k+\mu_{k-1}^{-1}\boldsymbol{L}_{k-1}\right)-\left(\boldsymbol{S}_k+\mu_{k-1}^{-1}\boldsymbol{L}_{k-1}\right)\right\|_F$$

$$\leq\lim_{k\to\infty}\left\|\boldsymbol{D}+\mu_{k-1}^{-1}\boldsymbol{Y}_{k-1}-\boldsymbol{S}_k-\boldsymbol{L}_k\right\|_F+\left\|\boldsymbol{S}_k-\boldsymbol{S}_{k-1}+\mu_k^{-1}\boldsymbol{Y}_k-\mu_k^{-1}\boldsymbol{Y}_{k+1}-\mu_{k-1}^{-1}\boldsymbol{Y}_{k-1}\right\|_F$$

$$\leq\lim_{k\to\infty}\left\|\Sigma_{k-1}-S_{\frac{w}{\mu_{k-1}}}(\Sigma_{k-1})\right\|_F+\|\boldsymbol{S}_k-\boldsymbol{S}_{k+1}\|_F+\left\|\mu_k^{-1}\boldsymbol{Y}_k-\mu_k^{-1}\boldsymbol{Y}_{k+1}-\mu_k^{-1}\boldsymbol{Y}_{k-1}\right\|_F$$

$$=0$$

Therefore, (c) is proved.

Part 2. The detailed performance of the noise level estimator.

We add AWGN to an image whose size is 1200*800, and crop images from the whole image. We evaluate the noise level estimator in images corrupted with different Gaussian standard deviations and scales. The detailed error is shown in Table 1.

Table 1. The detailed performance of the noise level estimator.

sigma *size*	*32*32*	*64*64*	*96*96*	*128*128*	*160*160*	*192*192*
5	0.25788	0.106924	0.0673	0.055161	0.04072	0.034186
10	0.2522	0.123357	0.06926	0.057312	0.04237	0.040077
18	0.249583	0.126736	0.073444	0.054082	0.038606	0.041251
25	0.254308	0.128341	0.067956	0.064896	0.042948	0.035538
33	0.276979	0.104858	0.080773	0.051363	0.051539	0.039601
40	0.252385	0.110462	0.07756	0.059788	0.05469	0.039997
50	0.250868	0.132255	0.075972	0.056955	0.03885	0.034717
60	0.293063	0.106316	0.066268	0.065348	0.039012	0.036249
23	0.247366	0.121033	0.084651	0.061321	0.043117	0.038052
31	0.228122	0.133609	0.073455	0.06567	0.036836	0.041243
38	0.233971	0.144312	0.070069	0.065633	0.044736	0.03737
46	0.273635	0.118261	0.089663	0.063504	0.046659	0.038555
53	0.274544	0.137155	0.075883	0.060193	0.045742	0.036193
63	0.269922	0.114115	0.068734	0.053417	0.037959	0.033621

73	0.23091	0.11849	0.077629	0.05803	0.03563	0.036124
ave	0.256382	0.121748	0.074575	0.059511	0.042628	0.037518
std	0.018377	0.012014	0.006764	0.004809	0.005393	0.00253

sigma size	*224*224*	*256*256*	*288*288*	*320*320*	*352*352*	*384*384*
5	0.02846	0.029158	0.02312	0.019417	0.02144	0.017127
10	0.0341	0.028567	0.02473	0.024259	0.0214	0.020263
18	0.030606	0.033831	0.023194	0.026472	0.017989	0.018914
25	0.025912	0.023357	0.021572	0.017719	0.016176	0.015607
33	0.031164	0.026581	0.027779	0.024444	0.019676	0.019097
40	0.035435	0.028568	0.03164	0.022272	0.023798	0.021181
50	0.032116	0.026589	0.026222	0.02082	0.02053	0.018464
60	0.034355	0.028526	0.023827	0.023161	0.022067	0.020828
23	0.028187	0.033266	0.023071	0.021863	0.020276	0.01908
31	0.029752	0.030299	0.025571	0.025881	0.020886	0.018808
38	0.027259	0.026132	0.021891	0.02062	0.020526	0.018438
46	0.03375	0.028121	0.026763	0.022834	0.023917	0.01925
53	0.030794	0.025598	0.021796	0.019994	0.018884	0.015868
63	0.030682	0.028645	0.021777	0.022343	0.018759	0.017723
73	0.028683	0.023607	0.022504	0.018606	0.018861	0.017459
ave	0.03075	0.028056	0.024364	0.022047	0.020346	0.01854
std	0.002802	0.002973	0.002815	0.002566	0.00208	0.001606

sigma size	*416*416*	*448*448*	*480*480*	*512*512*	*544*544*	*576*576*
5	0.0174	0.017617	0.01464	0.015476	0.01258	0.013982
10	0.01788	0.017833	0.01645	0.015137	0.01246	0.012238
18	0.015261	0.014018	0.013944	0.01204	0.013528	0.010424
25	0.01434	0.012245	0.012232	0.01112	0.009936	0.011457
33	0.014494	0.016787	0.012124	0.014062	0.011148	0.012441
40	0.019905	0.018337	0.015555	0.015798	0.012803	0.012697
50	0.017212	0.015251	0.015076	0.013147	0.011046	0.011854
60	0.018533	0.017094	0.013675	0.016353	0.012017	0.012912

23	0.019121	0.016703	0.015573	0.01544	0.014067	0.014249
31	0.015556	0.015122	0.013394	0.012991	0.011969	0.011791
38	0.016697	0.014636	0.014597	0.013127	0.011186	0.012667
46	0.018521	0.013837	0.014709	0.013567	0.013308	0.011989
53	0.013395	0.014529	0.012876	0.012885	0.012426	0.012593
63	0.015525	0.017299	0.014542	0.013401	0.012378	0.010325
73	0.017211	0.01257	0.014618	0.011355	0.011648	0.010386
ave	0.016737	0.015592	0.014267	0.013727	0.012167	0.012134
std	0.001914	0.001941	0.001234	0.001617	0.001072	0.001175

sigma size	*608*608*	*640*640*	*672*672*	*704*704*
5	0.01064	0.010789	0.01022	0.010194
10	0.01238	0.010963	0.01019	0.009645
18	0.012339	0.009471	0.010461	0.009404
25	0.009712	0.009881	0.008372	0.008532
33	0.0097	0.011459	0.009061	0.010869
40	0.011908	0.011552	0.009553	0.009136
50	0.010112	0.0106	0.009374	0.01037
60	0.011355	0.011259	0.00982	0.010118
23	0.012955	0.013131	0.011385	0.011994
31	0.011539	0.011112	0.010892	0.009999
38	0.009757	0.010904	0.009096	0.009986
46	0.01252	0.010851	0.011103	0.009138
53	0.011461	0.01012	0.009313	0.009607
63	0.010761	0.008984	0.009554	0.009587
73	0.009866	0.008954	0.010109	0.007258
ave	0.011134	0.010669	0.0099	0.009723
std	0.001139	0.00108	0.000834	0.00106

Part 3. The reason we choose the region size (256*256) in our paper.

In our experiment, the images are split into a few regions to estimate the noise levels. The split mode depends on the performance of the noise level estimator and the size of the image.

On one hand, we test the noise level estimator which we employ on a noisy image whose size is 1200*800 and is corrupted by AWGN. The average results of different standard deviations are shown in the Fig.1. All results can be referred to the above part, and there is little difference on the regions with the same scale and different standard deviations. We argue that the experimental results of noise level estimation on the synthetic noisy images are referable to the real noisy images.

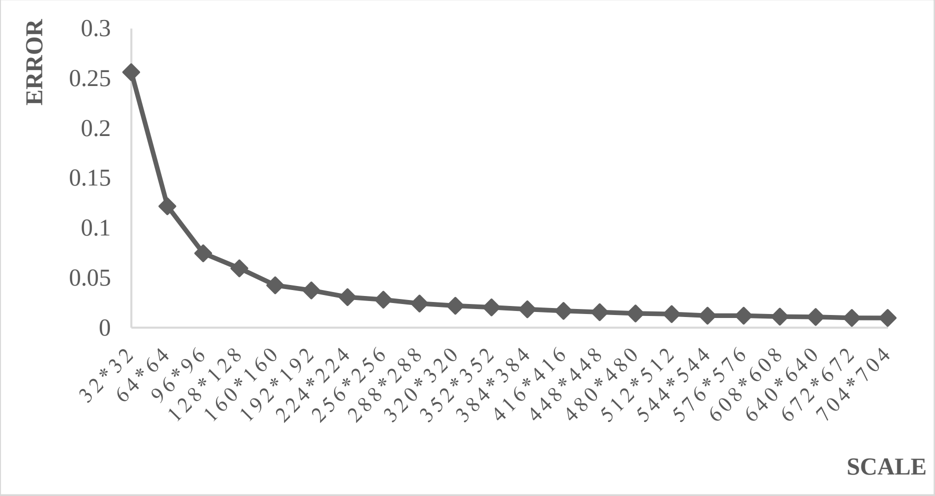

Fig.1: The performance of the noise level estimator.

We can see that if the part is too small, the estimator will result in a big error. When the scale of the part is larger than 192*192, the estimator can meet the good performance. However, if the part is too big, the part with more serious or slight noise will not be well processed. That is to say, if the noise level is the same among the whole image, the bigger estimated scale is better, otherwise, the local noise condition will be dismissed when the noise level is estimated on the oversize region. So we should find the small region but with good performance of noise level estimation on it.

On the other hand, considering that the sizes of the images in the open dataset 1 we employ are all 512*512*3, we test our method on the real noisy dataset1 with different split modes. To simplify the process, we split the images with the four scales as follows, and the results are shown in the Table 2, and the best results are in bold.

Table 2: The results of our method with different split modes.

Scale	PSNR	SSIM
512*512(*1)	37.1757	0.9508
256*256(*4)	**37.2107**	**0.9511**
128*128(*16)	36.9282	0.9477
64*64(*64)	32.9230	0.9015

When the images are split into four regions whose scales are all 256*256, our method gets the best performance. In fact, the images with different sizes can be split properly by our computer program, and the image with the size a*b*3 can be split into ⌊ a/256⌋*⌊ b/256⌋ parts. If the length of the side cannot be divided by 256, there will be some regions whose scales are bigger than 256*256. The examples are shown in Fig.2.

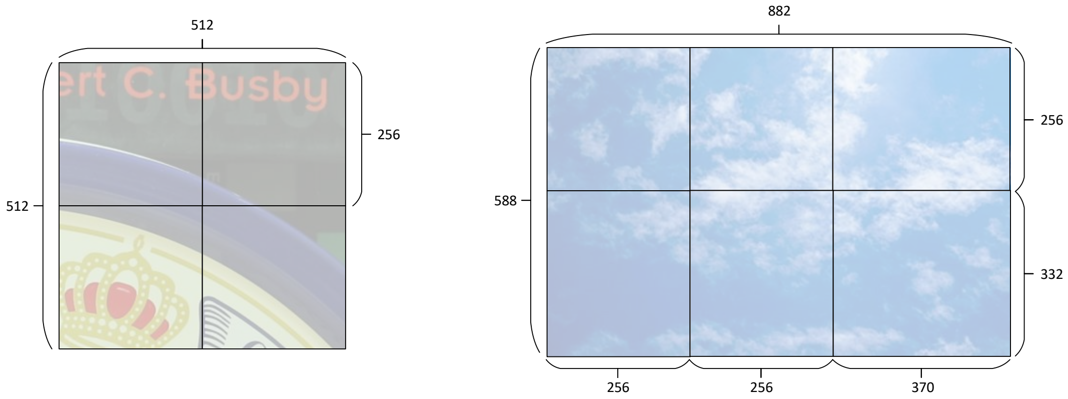

Fig. 2: Two examples of split modes. The left shows the split mode when the length of the side of the image can be divided by 256. The right shows the split mode when the length of the side of the image cannot be divided by 256.