

Real-Time Non-Local Means Image Denoising Algorithm Based on Local Binary Descriptor

Hancheng Yu and Aiting Li

Key Laboratory of Radar Imaging and Microwave Photonics (Nanjing Univ. Aeronaut. Astronaut.), Ministry of Education, Nanjing University of Aeronautics and Astronautics,

Nanjing, 210016, China

[e-mail: yuhc@nuaa.edu.cn]

*Corresponding author: Hancheng Yu

*Received August 31, 2015; revised November 9, 2015; accepted December 9, 2015;
published February 29, 2016*

Abstract

In this paper, a speed-up technique for the non-local means (NLM) image denoising method based on local binary descriptor (LBD) is proposed. In the NLM, most of the computation time is spent on searching for non-local similar patches in the search window. The local binary descriptor which represents the structure of patch as binary strings is employed to speed up the search process in the NLM. The descriptor allows for a fast and accurate preselection of non-local similar patches by bitwise operations. Using this approach, a tradeoff between time-saving and noise removal can be obtained. Simulations exhibit that despite being principally constructed for speed, the proposed algorithm outperforms in terms of denoising quality as well. Furthermore, a parallel implementation on GPU brings NLM-LBD to real-time image denoising.

Keywords: Local binary descriptor, non-local means, image denoising, real time image processing

1. Introduction

Digital images and videos play a more and more important role in the daily applications nowadays. In practical video systems, the sensor output carries both signal and noise components which make high-quality image acquisition difficult, especially in a low light environment. As more image sensors per unit area are packed on a chip, camera devices tend to be more sensitive to noise [1] [2]. Therefore, fast image denoising is very important for subsequent real-time application.

Generally, image denoising algorithms can be categorized as spatial domain, transform domain, and dictionary learning based upon the image representation. Prevailing transform domain algorithms are Gaussian Scale Mixture Model based method [3], Stein's Unbiased Risk Estimate (SURE) [4] and Block Matching and 3-D filtering (BM3D) [5]. K-clustering with singular value decomposition (K-SVD) [6] and learned simultaneous sparse coding (LSSC) [7] are two outstanding dictionary learning methods. Though the denoising quality is satisfactory, the transform-based methods require high algorithm complexity and the sparse model is computationally expensive thus both of them are not feasible for real-time application. Spatial techniques tend to be more practical for real-time image denoising.

Spatial filters are usually categorized as local and nonlocal filters. Local denoising algorithm such as bilateral filter [8] [9] and trained filter [10] are effective in terms of time complexity. Nevertheless, in the instance of high noise level, these algorithms can hardly perform well because the correlations between neighboring pixels have been corrupted by severe noise. On the contrary, the nonlocal filters take advantage of the self-similarity of natural images in a nonlocal manner. Nonlocal means (NLM) [11] is more robust in denoising compared with local filters as a small patch around each pixel is considered instead of considering only the center pixel. Though NLM brought a dramatic improvement in denoising, it is computationally demanding as each noisy pixel is replaced by a weighted average of all the pixels in a large search window or whole image.

Since NLM was proposed, many variants on it have been developed. Some of them focus on the acceleration of NLM [12]–[16]. A preselection of similar patches [12] has been introduced to obtain a speedup after the proposal of NLM. Recent NLM speed-up algorithms include modified multi-resolution pyramid architecture [13] and probabilistic early termination (PET) [14]. The former accelerates the computation of window similarity and the latter terminates the computation of the distortion term in a specific situation. Otherwise, a dictionary is built [15] to search similar patches very quickly and reduce the computational cost. Fast Fourier transform is used to accelerate the weight calculation [16]. Most of these fast algorithms achieve higher efficiency by abandoning part of denoising quality at the same time.

Others are for enhancing the denoising performance [17]–[22]. Singular value decomposition (SVD) [17] is introduced to eliminate dissimilar pixels. Some optimal techniques have been introduced in [18]. Shape-adaptive patches (NLM-SAP) [19] can take advantage of the local geometry of the image in the NLM. SURE-based linear expansion of thresholds (SURE-LET) [20] is a way to optimize the parameters of the NLM. The parameter optimization is pushed forward by modifying the weight of the center patch [21] and the probability density function of the distribution [22]. However, the superior performances of these NLM algorithms are achieved at the cost of higher computational complexity.

In this paper, a new local binary descriptor based NLM denoising algorithm is proposed. Although the NLM algorithm has been well studied, the computing-cost is still hard to fill with the demand of real-time image denoising grows. The NLM filter can preserve image details better than point-wise filters in that it relies on the non-local self-redundancy of the images and measures the similarities between the central patch and other patches in the search window. To fully utilize the advantage of the NLM filter in preserving details and overcome the drawbacks in huge computation amount is the motivation of this paper. Inspired by local binary patterns (LBP) [23], a new local binary descriptor is proposed to encode the local information of images. The individual bits of the binary descriptor are obtained by assessment of the local similarity between the central pixel and its neighborhood to store structural information. Due to low matching and storage costs, these binary descriptors allow for a fast and accurate preselection of non-local similar patches which can speed up and improve the NLM algorithm. As shown in the experiments, the NLM-LBD outperforms the standard NLM in terms of denoising quality and is quite suitable for real-time denoising applications.

2. Local Binary Descriptor

2.1 The Non Local Means Filter

The non-local means filter replaces the considered pixel with the weighted mean of pixels in a large search window or whole image:

$$\hat{u}_p = \frac{\sum_{q \in B(p,r)} w_{p,q} u_q}{\sum_{q \in B(p,r)} w_{p,q}}, \quad (1)$$

where \hat{u}_p is the restored value at pixel p and $B(p, r)$ indicates the search window centered at p with $(2r+1) \times (2r+1)$ pixels. The weight $w_{p,q}$ measures the similarities between the central patch and candidate patches in its search window. This filter preserves image features due to its use of non-local self-similarity.

The patch similarity is calculated by the l^2 -distance between the central patch $B(p, f)$ and the candidate patch $B(q, f)$ centered respectively at p and q :

$$d_{p,q}^2(B(p, f), B(q, f)) = \frac{1}{(2f+1)^2} \sum_{j \in B(0, f)} (u_{p+j} - u_{q+j})^2, \quad (2)$$

where u_{p+j} denotes the value in the $(2f+1) \times (2f+1)$ central patch, and u_{q+j} denotes the corresponding value of candidate patch in the search window. In order to search the self-similarity of an image, each distance between patches in the search window is calculated which is computation-intensive and time-consuming.

2.2 Local Binary Descriptor

The NLM algorithm computes the similarity between two patches by scanning a vast portion of the image in search of all the pixels that really resemble the central pixel which causes large amount of calculation. There are many dissimilar neighbors in the search window,

how to group the similar patches efficiently is the key to speed up NLM. One compelling way is to represent the local patches as binary strings to take advantage of a bitwise operation and accelerate the matching process. The traditional LBP is not suitable for NLM. A new local binary descriptor is proposed to speed up the search process in the NLM. This local binary descriptor uses local patch comparisons and local minimum encoding to represent the structure of patch as binary strings which allows for a fast and accurate preselection of non-local similar patches by bitwise operations.

The bit string is set up out of a set of selected local patch comparisons to describe the structure of the central patch. The local patch comparison is defined by the distance between the central patch to the selected neighbor patches:

$$d_{p,q(m)}^2(B(p,f),B(q(m),f)) = \frac{1}{(2f+1)^2} \sum_{j \in B(0,f)} (s_{p+j} - s_{q(m)+j})^2. \quad (3)$$

Before moving forward, the noisy image is smoothed with Gaussian filter to reduce its sensitivity to noise, s_{p+j} and $s_{q(m)+j}$ denote the value in central patch and the corresponding value in its selected neighbor patches of the smoothed image respectively. A set of neighbor patches are chosen specifically and generate corresponding binary codes and $q(m)$ is the m th chosen neighbor patch. As shown in Fig. 1, the highlighted squares are the 16 central pixels of the selected neighbor patches of the central patch centered at p . Number 0 to 15 indicate their order in the bit string.

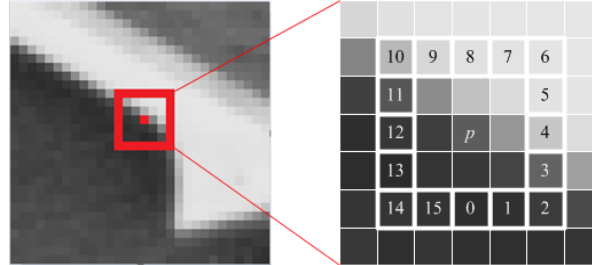


Fig. 1. Example of a central pixel with an outline from its 5×5 neighborhood

For simplicity, if local distance values are all smaller than a threshold T_1 , which means the central patch lacks local image feature, its bit string will be set zeros immediately:

$$\tau_p := 0, \quad \text{if } \max(d_{p,q(m)}^2) < T_1, \quad (4)$$

where $\max(d_{p,q(m)}^2)$ denotes the maximum of $d_{p,q(m)}^2$. Else if $\max(d_{p,q(m)}^2)$ is larger than T_1 , which means the central patch contains local image feature, its bit string will be as follows:

$$\tau_p(m) := \begin{cases} 1, & \text{if } d_{p,q(m)}^2 < d_{p,q(m-1)}^2 \text{ and } d_{p,q(m)}^2 \leq d_{p,q(m+1)}^2 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $\tau_p(m)$ denotes the m th bit in the bit string. The local minimum distances which are less than the distance of its adjacent patches are set value one and others zero, and $d_{p,q(0)}$ to $d_{p,q(15)}$ are considered as a circle. Thus, the local binary descriptor is set as a 16-bit binary code which describes the structure of the central patch. The value in the position where the

local similar patches exist can be distinguished from others. For the example in Fig. 1, the bit string of the example central patch is [00001000 0000 0100], that means its most similar patches at the position of 2 and 11 as shown in Fig. 1.

2.3 Descriptor Matching

These binary strings can speed up the search for high similar patches in the NLM by comparing the candidate string τ_q with its central string τ_p . But for better denoising results, more similar patches should be preselected against the risk of information loss. So a descriptor template is proposed to select more similar patches which can be realized with bitwise operation.

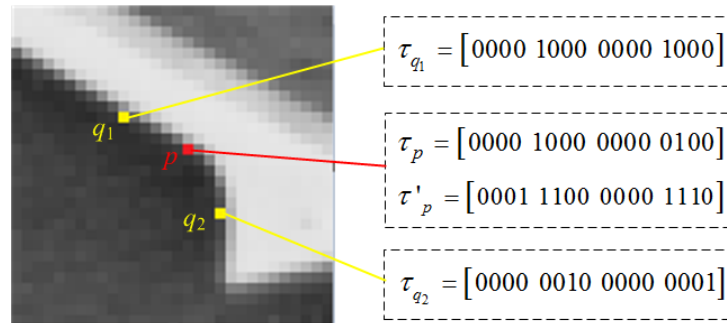


Fig. 2. Example of binary codes matching.

As shown in Fig. 2, for the central patch which is coded as $\tau_p = [0000 1000 0000 0100]$, two of its candidate patches at q_1 and q_2 are coded as $\tau_{q_1} = [0000 1000 0000 1000]$ and $\tau_{q_2} = [0000 0010 0000 0001]$ respectively. In order to retain more feature information, descriptor template τ'_p can be easily extended by an expansion of its binary descriptor τ_p as

$$\tau'_p(m) := 1 \text{ if } \tau_p(m-1)=1 \text{ or } \tau_p(m+1)=1 \text{ or } \tau_p(m)=1. \quad (6)$$

The template of the central patch is then compared with its candidate binary string according to bitwise-AND operation. The results equal to the candidate string illustrates a structural similarity with the central patch:

$$\begin{cases} (\tau'_p \& \tau_q) = \tau_q & \text{similar} \\ (\tau'_p \& \tau_q) \neq \tau_q & \text{dissimilar} \end{cases} \quad (7)$$

Given the example above, the template of the central patch is $\tau'_p = [0001 1100 0000 1110]$ as shown in Fig. 2. Between the two candidate patches q_1 and q_2 , only the patch at q_1 which is coded as $\tau_{q_1} = [0000 1000 0000 1000]$ matches the central descriptor template τ'_p and will be preselected as a non-local similar patch. From the above, the binary descriptor and its match template allow for a fast and accurate preselection of non-local similar patches only by bitwise operation.

3. The Proposed Algorithm

The binary descriptor increases the speed of similar patches preselection because the bitwise

operation can be executed extremely fast.

The proposed NLM based on local binary descriptor (NLM-LBD) is summarized as follows:

Algorithm: NLM-LBD

Step1: Generate local binary descriptor:

Smooth the input noisy image with a Gaussian filter.

for all point p do

for $m := 0$ to 15 do

 compute local patch distance $d_{p,q(m)}^2$ using (3)

end

for $m := 0$ to 15 do

 compute $\tau_p(m)$ using (4) and (5)

end

for $m := 0$ to 15 do

 compute $\tau'_p(m)$ using (6).

end

end

Step2: NLM denoising based on local binary descriptor:

for $i := 0$ to L step $:= N^{\text{step}}$ do

for $j := 0$ to C step $:= N^{\text{step}}$ do

$p := j \cdot L + i$

if $\tau_p > 0$ then

for all point $q \in B(p, r)$

if $(\tau'_p \ \& \ \tau_q) = \tau_q$ and $|s_p - s_q| < k \cdot \sigma_n$ then

 compute non-local patch distance $d_{p,q}^2$ using (2)

 compute $w_{p,q}$ using (9)

end

end

else

for all point $q \in B(p, r)$

if $\tau_q = 0$ and $|s_p - s_q| < k \cdot \sigma_n$ then

$w_{p,q} = 1$

end

end

end

 compute \hat{u}_p using (1)

end

end

In Step2, if the code of the central pixel is greater than zero, the patch is considered contain local image feature. And its structure and gray value similar patches will be preselected in the search window. The structural similarity patches are preselected by using bitwise operation in (6), and the smoothed image in Step1 is as the measure to pre-select similar gray value patches:

$$|s_p - s_q| < k \cdot \sigma_n \quad (8)$$

Then the non-local patch distance between the central patch and its preselected similar patches is calculated using (2). The weights of the preselected similar patches are as follows:

$$\begin{cases} w_{p,q} = 1, & d_{p,q}^2 < T_1 \\ w_{p,q} = \frac{T_2 - d_{p,q}^2}{T_2 - T_1}, & T_1 \leq d_{p,q}^2 < T_2 \\ w_{p,q} = 0, & d_{p,q}^2 \geq T_2 \end{cases} \quad (9)$$

where thresholds T_1 and T_2 are given by

$$\begin{cases} T_1 = \lambda \sigma_n^2 \\ T_2 = 2T_1 \end{cases} \quad (10)$$

For those pixels whose codes are equal to zero which means they are feature-free, their similar patches are preselected by judging whether the binary string are also zero and the gray values are close in the search window. For computational simplicity, the weights of featureless similar patches are set one directly.

Table 1. Parameters for the proposed NLM-LBD algorithm

| σ_n | f | r | λ | k | σ_s |
|------------|-----|-----|-----------|-----|------------|
| ≤ 30 | 2 | 6 | 2.1 | 0.6 | 1.1 |
| > 30 | 3 | 7 | 1.6 | 0.4 | 1.4 |

Table 1 lists the parameters in the proposed NLM-LBD algorithm. σ_s indicates the standard deviation of the Gaussian filter which is used to smooth the input noisy in Step1. It can be seen that the sizes of the patch, research window and σ_s depend on the standard deviation of noise σ_n . If $\sigma_n \leq 30$, $f = 2$ and $r = 6$ denote that if the standard deviation of noise is less than 30, the proposed NLM algorithm uses 5×5 patch and 13×13 research window. If standard deviation of noise is more than 30, large size patch, research window and σ_s are more conducive to the suppression of noise. The values of λ and k also depend on standard deviation of noise σ_n . The parameters listed in **Table 1** provide the best experimental results.

In Step2 of NLM-LBD, for a noisy image with size $L \times C$, let step $N^{\text{step}} = 1$ means every pixel is scanned. In order to realize a faster version, the number of processed pixel is reduced: rather than sliding by one pixel to every next, use a step of N^{step} in both horizontal and vertical directions. In the faster version, let $N^{\text{step}} = f$.

4. EXPERIMENTAL RESULTS

In this section, the proposed NLM-LBD and its fast version NLM-FLBD are evaluated and compared with many other existing techniques. Experiments are conducted on four standard grayscale test images and two real color noisy images with distinctly different features. The standard grayscale test images are corrupted by simulated additive Gaussian noise at different power levels. And the runtimes of NLM-LBD and NLM-FLBD are compared with the NLM. Furthermore, the NLM-FLBD is evaluated in restoring real color noisy images.

4.1 Comparing Performance for Noise Removal

To validate the effect of the proposed algorithm for image denoising objectively and subjectively, firstly, the denoising quality of the proposed NLM-LBD is compared with the standard patchwise NLM [11] and its two popular variants (NLM-SAP [19] and NLM-SURELET [20]) visually. Besides, experiments are conducted on four test images corrupted by Gaussian noise at different noise levels. Here, results are summarized in terms of the peak signal-to-noise ratio (PSNR).

Fig. 3 (a) is part of the *house* image corrupted by Gaussian noise ($\sigma_n = 20$). **Fig. 3** (b)-(e) compares the visual quality of various denoising algorithms. It is clearly that **Fig. 3** (c) is over-smoothed slightly and Fig. 3(d) preserves more details but contains more defects either. **Fig. 3** (e) yields a high visual quality both in respects of noise suppression and image detail preservation. **Table 2** reports the PSNR of the experiments at different noise levels for the reference images. Clearly, the proposed algorithm provides an overall improvement over the standard NLM algorithms in PSNR and close to the current outstanding NLM schemes.

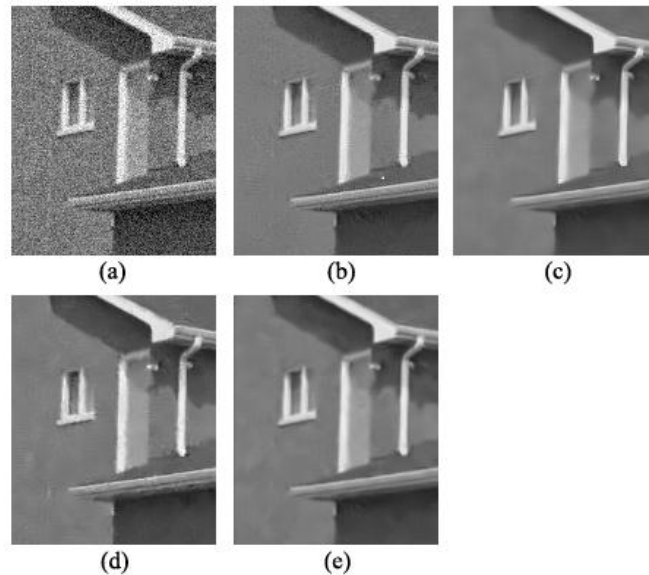


Fig. 3. Results of different NLM algorithms. (a) Part of the *house* image corrupted by Gaussian noise ($\sigma_n = 20$). (b) Standard NLM result. (c) NLM-SAP result. (d) NLM-SURELET result. (e) Proposed NLM-LBD result.

Table 2. Comparison of different NLM denoising algorithm in PSNR (dB)

| σ_n | 10 | 20 | 40 | 10 | 20 | 40 |
|-------------|-------------------------------|-------|-------|----------------------------------|-------|-------|
| | <i>House(256²)</i> | | | <i>Peppers (256²)</i> | | |
| NLM | 34.95 | 32.00 | 28.32 | 33.11 | 29.79 | 26.23 |
| NLM-SAP | 35.49 | 32.48 | 27.89 | 32.97 | 29.60 | 26.04 |
| NLM-SURELET | 35.68 | 32.76 | 29.44 | 34.10 | 30.96 | 27.74 |
| NLM-LBD | 35.57 | 32.86 | 29.43 | 34.23 | 31.00 | 27.17 |
| NLM-FLBD | 35.29 | 32.64 | 29.17 | 33.88 | 30.69 | 26.88 |
| | <i>Boat(512²)</i> | | | <i>Lena (512²)</i> | | |
| NLM | 33.11 | 29.78 | 26.23 | 34.51 | 31.58 | 28.17 |
| NLM-SAP | 32.97 | 29.60 | 26.04 | 35.04 | 31.91 | 28.21 |
| NLM-SURELET | 33.31 | 30.22 | 27.12 | 35.25 | 32.37 | 29.34 |
| NLM-LBD | 33.17 | 30.06 | 27.00 | 35.27 | 32.35 | 29.30 |
| NLM-FLBD | 32.90 | 29.92 | 26.80 | 34.96 | 32.19 | 29.06 |

4.2 Comparing Time Consuming

Table 3. Time cost of different NLM denoising algorithm (seconds)

| σ_n | 10 | 20 | 40 | 10 | 20 | 40 |
|------------|---------------------------------|------|------|-------------------------------|------|-------|
| | <i>Peppers(256²)</i> | | | <i>Lena (512²)</i> | | |
| NLM | 1.50 | 2.70 | 9.82 | 5.52 | 9.10 | 34.53 |
| NLM-LBD | 0.20 | 0.27 | 0.49 | 0.80 | 1.05 | 2.08 |
| NLM-FLBD | 0.10 | 0.13 | 0.14 | 0.28 | 0.37 | 0.39 |

In this section, the time consuming of NLM-LBD and its fast version NLM-FLBD are compared with standard NLM. **Table 3** shows the total time required for detecting similar patches, preselect in the research window, restore each pixel and remove Gaussian noise. The NLM-SAP and NLM-SURELET are not listed in the **Table 3** because the two algorithms not competitive in this aspect. The proposed NLM-LBD and NLM-FLBD are compared with standard NLM in the computation time on personal computer with 64-bit, 3.30 GHz. The computation time has been averaged over 20 runs. It can be seen from **Table 3** that the computation time of proposed NLM-FLBD is about 15 to 90 times faster than standard NLM, which represents a substantial reduction in computational cost. Therefore a balance between speed and performance for practical application is achieved.

With the rapid development of GPU (graphics processing unit) recently, it has been widely used to further accelerate the image processing. NLM-LBD method can be easily implemented on CUDA. In Step1, the noisy image is loaded as a texture, smooth the noisy image and generate local binary descriptor. In Step2, the smooth result and binary descriptor are loaded as texture, the filter is implemented in a CUDA kernel. To achieve the real-time performance, the NLM-LBD algorithm is implemented in parallel using GPU (NVIDIA GeForce GTX660). The denoising time for a GigE industrial camera with 1.3 megapixel CMOS sensor is about 15ms which means over 65 frames of megapixel image can be processed in one second. In contrast, denoising time of NLM is about 170ms. It shows that a parallel implementation on GPU could bring NLM-LBD to real-time image denoising.

4.3 Real noisy images restoration

The assumption of the white Gaussian noise is not always accurate in real noisy images. The overall the Non-Gaussian noise characteristics in a real image depend on many factors and the real noisy images restoration is rather difficult to realize.

In this section, the proposed NLM-FLBD is evaluated and compared with multiresolution bilateral filter (MBF) [24] and edge-based bilateral filter (EBF) [25] in restoring real color noisy images. The noisy *Color blocks* in Fig. 4 (a) was captured with digital camera at ISO 1600, and noisy *Moire scanned* in Fig. 5 (a) was downloaded from [26]. In Fig. 4 and Fig. 5, in order to remove noise, the MBF loses some image features and color. By contrast, the EBF and NLM-FLBD preserves more features, and the NLM-FLBD suppresses more noise.

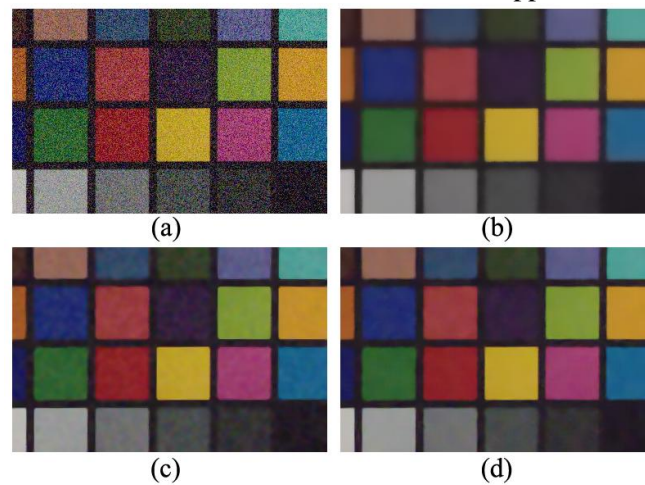


Fig. 4. Results on noisy Color blocks image. (a) Noisy image (385×250); (b) MBF result; (c) EBF result; (d). NLM-FLBD result.

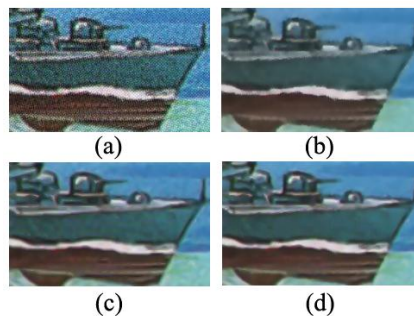


Fig. 5. Results on noisy Moire scanned image. (a) Noisy image (385×250); (b) MBF result; (c) EBF result; (d). NLM-FLBD result.

5. Conclusion

In this paper, an efficient local binary descriptor for image feature extraction is applied to NLM in pursuit of a higher speed of image processing. Meanwhile, the performance of the proposed NLM-LBD is also comparable to the current state-of-art NLM algorithms. Furthermore, the implement on GPU proved the proposed algorithm is feasible for real-time image processing.

References

- [1] J. H. Hong, S. B. Cho, and U. K. Cho, "A novel evolutionary approach to image enhancement filter design: Method and applications," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1446–1457, Dec. 2009. [Article \(CrossRef Link\)](#)
- [2] L. Shao, R. Yan, X. Li, and Y. Liu, "From heuristic optimization to dictionary learning: a review and comprehensive comparison of image denoising algorithms," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol.44, no. 7, pp. 1001–1013, Jul, 2014. [Article \(CrossRef Link\)](#)
- [3] G. Varghese and W. Zhou, "Video denoising based on a spatiotemporal Gaussian scale mixture model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 7, pp. 1032–1040, Jul. 2010. [Article \(CrossRef Link\)](#)
- [4] F. Luisier, T. Blu, and M. Unser, "SURE-LET for orthonormal wavelet-domain video denoising," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 913–919, Jun. 2010. [Article \(CrossRef Link\)](#)
- [5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, pp. 2080–2095, Aug. 2007. [Article \(CrossRef Link\)](#)
- [6] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006. [Article \(CrossRef Link\)](#)
- [7] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. of IEEE Int. Conf. Comput. Vision*, pp. 2272–2279, 2009. [Article \(CrossRef Link\)](#)
- [8] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. of 6th Int. Conf. Computer Vision*, pp. 839–846, 1998. [Article \(CrossRef Link\)](#)
- [9] W. Kao and Y. Chen, "Multistage bilateral noise filtering and edge detection for color image enhancement," *IEEE Trans. Consumer Electron.*, vol. 51, no. 4, pp. 1346-1351, Nov. 2005. [Article \(CrossRef Link\)](#)
- [10] L. Shao, H. Zhang, and G. de Haan, "An overview and performance evaluation of classification-based least squares trained filters," *IEEE Trans. Image Process.*, vol. 17, pp. 1772–1782, Oct. 2008. [Article \(CrossRef Link\)](#)
- [11] A. Buades, B. Coll, and J. M. Morel, "A nonlocal algorithm for image denoising," in *Proc. of IEEE Int. Conf. Comput. Vision Pattern Recognit.*, vol. 2, pp. 60–65, 2005. [Article \(CrossRef Link\)](#)
- [12] M. Mahmoudi, and G. Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhoods," *IEEE Signal Process. Lett.*, vol. 12, no. 12, pp. 839 –842, Dec. 2005. [Article \(CrossRef Link\)](#)
- [13] V. Karnati, M. Uliyar, and S. Dey, "Fast non-local algorithm for image denoising," in *Proc. of 16th IEEE Int. Conf. Image Process.*, pp. 3873-3876, 2009. [Article \(CrossRef Link\)](#)
- [14] R. Vignesh, B. T. Oh, and C.-C. J. Kuo, "Fast non-local means (NLM) computation with probabilistic early termination," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 277-280, Mar. 2010. [Article \(CrossRef Link\)](#)
- [15] H. Bhujle, and S. Chaudhuri, "Novel speed-up strategies for non-local means denoising with patch and edge patch based dictionaries," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 356–365, Jan. 2014. [Article \(CrossRef Link\)](#)
- [16] J. Wang, Y. Guo, Y. Ying, Y. Liu, and Q. Peng, "Fast non-local algorithm for image denoising," in *Proc. of IEEE Int. Conf. Image Process.*, pp. 1429–1432, 2006. [Article \(CrossRef Link\)](#)
- [17] T. Thaipanich, B. T. Oh, P.-H Wu, D. Xu, and C. -C. J. Kuo, "Improved image denoising with adaptive nonlocal means (ANL-means) algorithm," *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2623–2630, Nov. 2010. [Article \(CrossRef Link\)](#)
- [18] Brox, T., Kleinschmidt, O., and Cremers, D., "Efficient nonlocal means for denoising of textural patterns," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1083-1092, Jul. 2008. [Article \(CrossRef Link\)](#)

- [19] Deledalle, C., Duval, V., and Salmon, J., “Non-local methods with shape adaptive patches (NLM-SAP),” *J. Math. Imag. Vis.*, vol. 43, no. 2, pp.103–120, 2012. [Article \(CrossRef Link\)](#)
- [20] D. Van De. Ville and M. Kocher, “Nonlocal means with dimensionality reduction and SURE-based parameter selection,” *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2683–2690, Sep. 2011. [Article \(CrossRef Link\)](#)
- [21] Y. Wu, B. Tracey, P. Natarajan, and J. P. Noonan, “James-stein type center pixel weights for non-local means image denoising,” *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 411–414, Apr. 2013. [Article \(CrossRef Link\)](#)
- [22] Y. Wu, B. Tracey, P. Natarajan, and J. Noonan, “Probabilistic non-local means,” *IEEE Signal Process. Lett.*, vol. 20, no. 8, pp. 763–766, Aug. 2013. [Article \(CrossRef Link\)](#)
- [23] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002. [Article \(CrossRef Link\)](#)
- [24] M. Zhang and B. K. Gunturk, “Multiresolution bilateral filtering for image denoising,” *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2324-2333, Dec. 2008. [Article \(CrossRef Link\)](#)
- [25] H. Yu, L. Zhao, H. Wang, “An Efficient Edge-based Bilateral Filter for Restoring Real Noisy Image,” *IEEE Trans. Consum. Electron.*, vol. 57, no. 2, pp. 682–687, May, 2011. [Article \(CrossRef Link\)](#)
- [26] Test Images [Online]. [Article \(CrossRef Link\)](#)



Hancheng Yu received the B.S. and M.S. degrees in Electronic and Information Engineering from Nanjing Normal University, China, in 2000 and 2003 respectively, and Ph.D. degree in the School of Information Science and Engineering, Southeast University, Nanjing, China, in 2011. He is currently an associate professor in the College of Electronic and Information Engineering, Nanjing university of Aeronautics and Astronautics, China. His research interests include Computer Vision and Video Processing.



Aiting Li received the B.S. degree in the College of Electronic and Information Engineering at Nanjing University of Aeronautics and Astronautics, China in 2013. She is now working towards the M.S. degree on Information Engineering in Nanjing University of Aeronautics and Astronautics, China. Her research interests include image/video processing and real-time digital signal processing.