IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 24, 2018, accepted November 21, 2018, date of publication November 27, 2018,
date of current version December 27, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2883459

Extremely Sparse Stripe Noise Removal From Nonremote-Sensing Images by Straight Line Detection and Neighborhood Grayscale Weighted Replacement

YUFU QU, XUAN ZHANG, QIANYI WANG, AND CHENGGUI LI
School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China

Corresponding author: Yufu Qu (qyf@buaa.edu.cn)

This work was supported by the Natural National Science Foundation of China under Grant NSFC 51675033.

ABSTRACT Traditional methods of stripe noise removal based on space domain or transformation domain generally cannot handle the case where the noise is extremely sparse. To solve this problem, we propose a novel approach to accurately detect and remove the stripe noise by analyzing the directional and structural information of the stripe noise. First, we build a preselected stripe noise lines set by using local progressive probabilistic Hough transform. Subsequently, the real stripe noise lines are screened out from this set according to the feature of grayscale discontinuities. Finally, our approach uses the strategy of neighborhood grayscale weighted replacement and a local Gaussian filter to perform image destriping. Extensive experiments demonstrate that our approach proposed in this paper outperforms other recent promising methods in terms of quantitative assessments, qualitative assessments, and computing time.

INDEX TERMS Grayscale weighted replacement, straight line detection, stripe noise removal.

I. INTRODUCTION

Owing to the interference of the data transmission link and difference between sensors, there is a special noise with a definite direction in image, which is generally called stripe noise. Most of existing stripe noise removal (destriping) methods are based on spatial domain or transform domain or combination of them, such as, histogram matching [1], [2], moment matching [3], optimization-based methods [4]–[8], Fourier transform [10], wavelet transform [11], and deep-learning-based method [9]. These methods, however, generally have limitations. For example, the histogram matching method needs to extract the standard noise template and then applies different degrees of filtering to different regions of the image according to the similarity to the extracted template. However, when the image background is complex and the stripe noise is extremely sparse and not obvious, extracting the template is challenging. Moment matching assumes that the mean and variance of the data recorded by any sensor will not differ significantly. However, this assumption is generally difficult to satisfy for relatively complex ground objects, especially for nonremote-sensing images. Optimization-based spatial methods, which are generally employed to construct a regularization model according to the feature of stripe noise in spatial domain, have achieved excellent results for the remote sensing image. However, for the nonremote-sensing image where the stripe noise is extremely sparse, these methods cannot obtain an ideal solution of the optimization model. Other methods based on transform domain generally require the stripe noise to be clearly identifiable in the frequency spectrum. But when the image contains useful information with a frequency comparable to the noise frequency, these methods cause serious distortions. In addition, if the stripe noise is extremely sparse, it will be barely observable in the frequency spectrum. Therefore, in this case, transform-based methods will completely fail.

To solve the above problems, we propose a novel algorithm that can effectively remove the extremely sparse stripe noise in the nonremote-sensing image. Fig. 1(a) shows a remote sensing image with periodic horizontal stripe noise. Note that in this image, the stripe noise is row-by-row, which means a whole row of pixels has noise. While in a nonremote-sensing image, the stripe noise does not have such a feature, i.e., it is extremely sparse, as shown in Fig. 2. Through the analysis of a large number of noise images, we have summed up

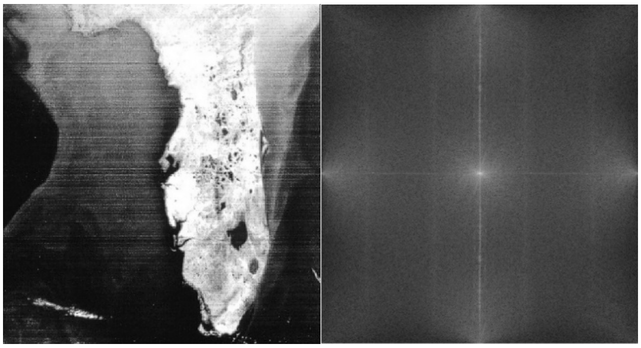

FIGURE 1. A remote sensing image with periodic horizontal stripe noise and its Fourier spectrum. (a) Raw image (provided by NOAA). (b) Fourier spectrum.

two basic features of extremely sparse stripe noise: straight line distribution and grayscale discontinuities—both in horizontal direction (x-axis) and vertical direction (y-axis)—which we call the "horizontal grayscale jump" and "vertical grayscale jump" respectively. According to these features, our approach first performs straight line detection by progressive probabilistic Hough transform (PPHT) and obtains a preselected noise line set. Subsequently, we can screen out the real stripe noise lines from this set, which should meet every grayscale discontinuity threshold. Finally, we use neighborhood grayscale weighted replacement and a local Gaussian filter to realize destriping. The advantage of our approach is that it does not require the stripe noise to exhibit spatial periodicity or obvious spectrum components. In addition, our approach can accurately remove extremely sparse stripe noise from complex image backgrounds with almost no loss of image quality.

The rest of this paper is organized as follows: Section II discusses related work; Section III describes our destriping approach to precisely locate and remove the extremely sparse stripe noise from a raw image; Section IV demonstrates results of our approach compared with other recent promising methods, followed by a summary of our work in Section V.

II. RELATED WORK

There are four classical well-established methods developed for removing stripe noise from an image. The first method is histogram matching [1], [2], which matches the histogram of a local image area to a standard noise area. This method is easy to implement and provides fast processing speed. It is obvious that this method requires the construction of the standard noise region histogram as the reference template, which is difficult in many situations.

The second approach is moment matching [3], which assumes that the means and standard deviations of the data

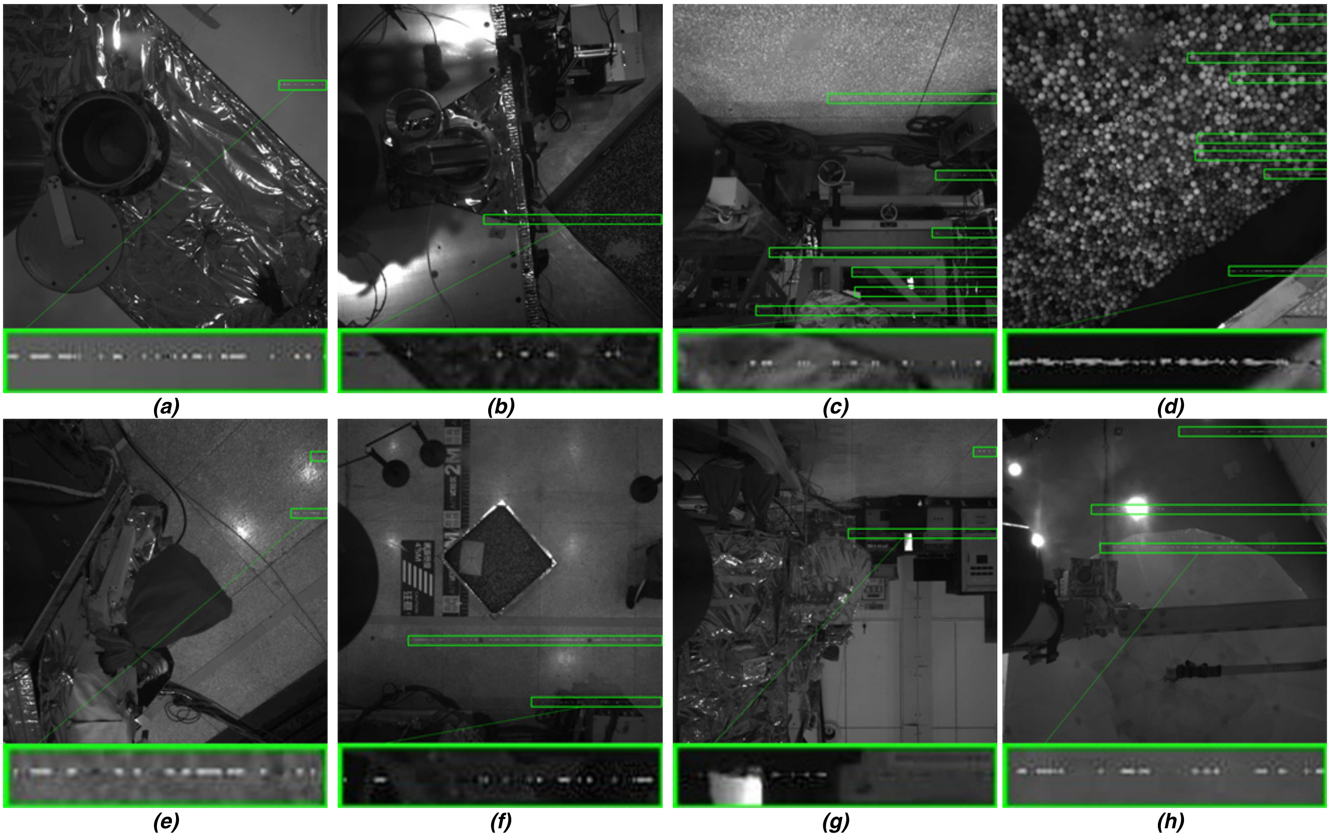

FIGURE 2. (a) – (h) Typical nonremote-sensing images with extremely sparse stripe noise.

recorded by any sensor will not differ significantly. This method is used mainly for processing images acquired by remote sensing photography and solving the problem of nonuniform response of different pixels in remote sensing images. However, since the source of the stripe noise in the nonremote-sensing image is not nonuniform response of different pixels, the fundamental hypothesis of moment matching is no longer valid. Therefore, this method cannot be used for nonremote-sensing images with complex backgrounds and extremely sparse stripe noise.

The third destriping method is the traditional Fourier transform [10]. This method removes the stripe noise component by constructing a suitable filter according to the distribution of the stripe noise in the frequency spectrum, which means that the stripe noise should be clearly identifiable in the frequency spectrum. At the same time, the overlap between the noise components and useful signal components of the image should be as small as possible in order to reduce image distortion. Therefore, for the extremely sparse stripe noise that does not exhibit spatial periodicity or significant frequency domain characteristics, this method will not perform well.

The fourth method is wavelet transform [11]. Wavelet transform uses the scaling and direction attributes to detect and eliminate stripe noise in the wavelet domain. The key to this method is the choice of the wavelet functions and the accurate acquisition of the frequency components of the stripe noise. Therefore, this method has limitations that similar to the Fourier transform. In other words, for images where the stripe noise cannot be clearly identified, the wavelet transform method exhibits insufficient performance. In Zhao *et al.* [12] proposed an improved wavelet transform method to eliminate the stripe noise, whose key point is the thresholds in various scales of each vertical direction are determined by themselves after wavelet decomposition.

Based on the above four classical methods, some new destriping methods have been proposed recently. Carfantan *et al.* [13] introduced a statistical self-calibration destriping method to perform linear response correction, which had been demonstrated that it performed better than simpler techniques based on column averages. The combination of wavelet transform and local interpolation [14] can protect more geometric and detailed information when removing the stripe noise; however, it still requires the stripe noise to have obvious frequency components. Another method [15] aims at processing continuous image sequences using an adaptive grayscale adjustment, and obtains the spectrum of the stripe noise accurately by accumulating the same frequency spectrum of the image sequences. Obviously, this method depends on the high correlation between sequential adjacent images. Pande-Chheetri *et al.* [16] developed a destriping method based on wavelet analysis and adaptive Fourier zero-frequency amplitude normalization, and this method showed promising effect for both stripe noise and random noise in a hyperspectral image. A new two-dimensional Fourier domain slope filter [17] is applied to remove the stripe noise at certain angles from the remote sensing image by setting a certain filtering width. However, if the stripe noise is not obvious in the image frequency spectrum, this method fails to provide good results.

Recently, the optimization-based methods have shown great superiorities for remote sensing image destriping problems. These approaches generally create an optimization model to obtain the optimal estimation of the underlying image. Rudin *et al.* [5] proposed the classical total variation (TV) model, which used the inherent regularity of a natural image (the underlying image) and removed the stripe noise via solving a partial differential equation. Although it can preserve the edges while destriping, it is also well known for producing staircase-like artifacts. Liu *et al.* [18] presented to use a usual l_2 data-fidelity term and an overlapping group sparsity TV regularizer to avoid the staircase effect. Bouali *et al.* [6] proposed a unidirectional total variation (UTV) model for the MODIS image (a kind of remote sensing image) stripe noise removal. Chang *et al.* [7] proposed to treat the multispectral images as a spectral-spatial volume and posed an anisotropic spectral-spatial total variation regularization to remove a more comprehensive stripe noise. This method tentatively categorizes the stripe noise in a more comprehensive manner, which can utilize both the spectral consistent information in spectral domain and the directional information of stripe noise in spatial domain. By taking the intrinsic properties of the stripe noise and image characteristics into consideration, Chen [4] integrated the unidirectional TV regularization, group sparsity regularization, and TV regularization together in an image decomposition framework and removed the stripe noise through its statistical analysis. This method, however, requires the stripe noise to exhibit obvious statistical features in the remote sensing image. Dou *et al.* [8] created a $\ell 0$ sparse model to estimate the stripe noise in the remote sensing image, and then they obtained the final destriped image from the difference of the known stripe noise image and the estimated stripe noise. Chen *et al.* [19] presented to use group sparsity to depict the column sparse structure of stripe component, and then constructed a novel convex optimization model, including a unidirectional TV term, a group sparsity term and a gradient domain fidelity term. This approach showed promising destriping performance for the remote sensing image. Chang *et al.* [20] proposed a low-rank-based single-image decomposition model (LRSID) to separate the original image from the stripe component, which convert creatively the image destriping task as an image decomposition even though it is effective mainly for the remote sensing image where there is some obvious stripe noise. In summary, although these optimization-based methods perform well for the remote sensing image, for the extremely sparse stripe noise in the nonremote-sensing image, the effect of destriping is not satisfactory.

For hyperspectral images (HSIs) stripe noise removal, Chang *et al.* [9] further proposed to use the deep convolution neural network (CNN) to achieve this target. The biggest advantage of their CNN-based HSIs denoising method (HSI-DeNet) is that the filters used in denoising can be learned without damaging the spectral-spatial structures. In addition, their HSI-DeNet is flexible for both single image and multiple images with slightly modifying the channels of the filters. However, it is still a great challenge for this method to remove the extremely sparse stripe noise in nonremote-sensing image such as Fig. 2.

Therefore, the above-mentioned methods can handle some remote sensing images with stripe noise, which has obvious spectrum components. However, for the nonremote-sensing image with relatively complex backgrounds, where the stripe noise is extremely sparse and its spectrum feature is insufficient, these methods are challenging to apply.

III. PROPOSED METHOD

In this section we describe our approach to remove the extremely sparse stripe noise from nonremote-sensing images. Fig. 4 is our destriping algorithm pipeline. The algorithm input is the raw noise image **n** and its output is the noise-removed image **u**. In addition, the image **l** shows the position of the stripe noise in the raw image.

In Section III-A, we first show several typical nonremote-sensing images with extremely sparse stripe noise, and give a general description of the features of stripe noise. Consequently, we introduce the first step of our approach in Section III-B, which is the acquisition of the preselected stripe noise lines set. Section III-C explains how we screen out the real stripe noise line from the acquired stripe noise lines set. Finally, we present the last step of our method in Section III-D, where we use neighborhood grayscale weighted replacement and the local Gaussian filter to realize the destriping.

A. EXAMPLES OF EXTREMELY SPARSE STRIPE NOISE

Fig. 1(a) shows a common remote sensing image with stripe noise. The stripe noise in this image exhibits spatial periodicity with a distinct noise component in the frequency domain (the vertical center line of the spectrum has a higher brightness than the horizontal line) as shown in Fig. 1(b).

However, for most nonremote-sensing images that will be processed in this paper, the stripe noise is not spatially periodic. Fig. 2 shows several typical images with extremely sparse stripe noise. Note that we use the green box to mark the position of the stripe noise for all eight images, and at the bottom of each image is the zoom result of one stripe noise line. In these images, the stripe noise has different lengths, different positions, and there is a low degree of discrimination between the noise and image background, especially in Fig. 2(d). Moreover, there is a lot of randomly distributed stripe noise in the raw images, as shown in Fig. 2(c) and Fig. 2(d).

We denote x-axis is along stripe noise direction and y-axis is across stripe noise direction. After the statistical analysis of a large number of noise images, the general features of such stripe noise can be summarized as follows:

1) the stripe noise occupies $StripN_h$ lines of pixels in the vertical direction (y-axis), i.e., the noise area has a certain height of $StripN_h$;
2) the stripe noise consists of a series of horizontal discrete points;
3) the stripe noise exhibits horizontal grayscale jumps along the horizontal direction (x-axis) and the vertical grayscale jumps along the vertical direction (y-axis), which means that there are grayscale discontinuities both in this two directions.

B. OBTAINING THE SET OF PRESELECTED STRIPE NOISE LINES

The first step of our approach is to obtain the set of preselected stripe noise lines. For the raw image displayed in Fig. 2, we first perform the necessary image preprocessing, including converting the raw image **n** into a grayscale image **g**, calculating the horizontal direction gradient by Sobel edge detection operator, and applying the adaptive threshold segmentation to the gradient image by the OTSU [21] algorithm in order to obtain the binary image **b**, of which the pixels' grayscale values are 0 or 255. Note that the stripe noise is generally distributed horizontally in the image. Hence, intuitively we should calculate the gradient in vertical direction (y-axis) by using the Sobel operator. However, as mentioned above, the stripe noise consists of a series of discrete horizontal points, which means that there are grayscale discontinuities in both horizontal and vertical directions. In order to reduce the interference of the other horizontal lines in the grayscale image as much as possible when screening out the real stripe noise lines in Section III-C, we calculate the gradient along the x-axis by Sobel operator. Fig. 3. shows the horizontal gradient map and the binary image for facilitating readers to observe the intermediate results.

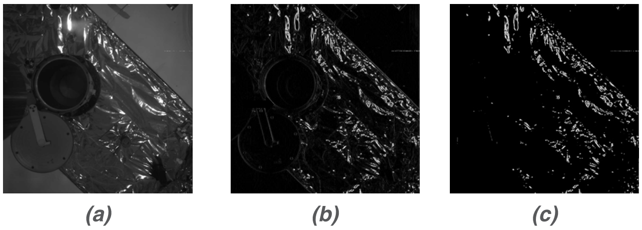

FIGURE 3. Intermediate results of the preprocessing of our algorithm. (a) Raw image. (b) Horizontal gradient map. (c) Binary image.

Subsequently, local progressive probabilistic Hough transform (LPPHT) is applied to the binary image **b** to obtain the set of preselected stripe noise lines. Progressive probabilistic Hough transform (PPHT) [22] is an improvement of the Hough transform (HT), which can detect a line that satisfies certain requirements (such as line length, point spacing, et al.) and return the image coordinates of the endpoints of this line. In this way, a preselected stripe noise line can be represented

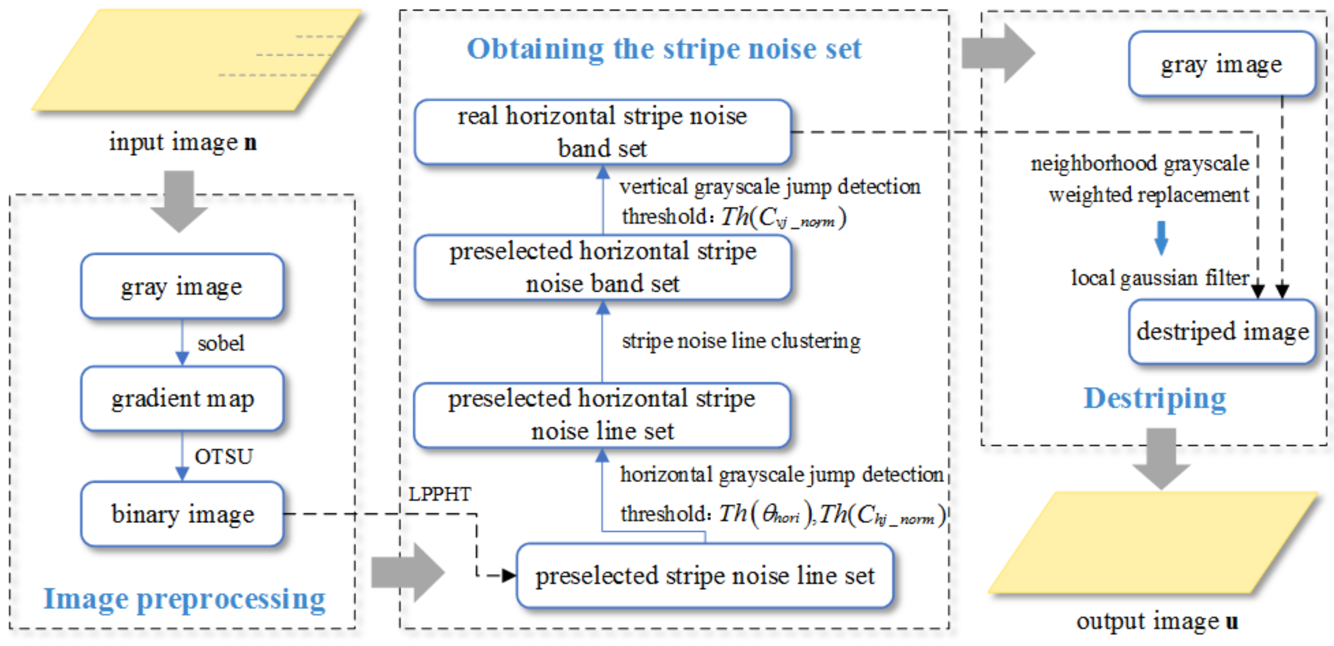
input image **n**
gray image
sobel
gradient map
OTSU
binary image
Image preprocessing
LPPHT
Obtaining the stripe noise set
real horizontal stripe noise band set
vertical grayscale jump detection
threshold: $Th(C_{vj\_norm})$
preselected horizontal stripe noise band set
stripe noise line clustering
preselected horizontal stripe noise line set
horizontal grayscale jump detection
threshold: $Th(\theta_{hori})$, $Th(C_{hj\_norm})$
preselected stripe noise line set
gray image
neighborhood grayscale weighted replacement
local gaussian filter
destriped image
Destriping
output image **u**

FIGURE 4. Pipeline of the proposed destriping algorithm.

as a four-dimension vector:

$$\mathbf{L} = (x_l, y_l, x_r, y_r), \tag{1}$$

where (x_l, y_l) are the image coordinates of the left endpoint of the stripe noise line, and (x_r, y_r) are the image coordinates of its right endpoint.

The PPHT algorithm follows the principle of probability priority, which means a point only belongs to the most likely straight line. However, for a noisy image with a very sparse stripe noise, if PPHT is carried out directly, the points that are a part of the stripe noise lines are likely to be assigned to background lines. In other words, it will not detect the real stripe noise lines. Hence, we propose the novel local progressive probabilistic Hough transform (LPPHT), which uses a sliding window to traverse the binary image $\mathbf{b}$ along the y-axis, and then carries out PPHT on a series of sub-images $\mathbf{b}^i_{sub}(i = 1, 2, \ldots)$, which are defined as follows:

$$\mathbf{b} = \sum_i \mathbf{b}^i_{sub}. \tag{2}$$

Assuming that the width of $\mathbf{b}$ is b_w, we set the width (SW_w) of the sliding window as b_w and the height (SW_h) of the sliding window as $StripN_h$, i.e., $SW_w = b_w$, $SW_h = StripN_h$. In this way, we can obtain the preselected set:

$$Q = \{\mathbf{L}\}. \tag{3}$$

C. SCREENING OUT REAL STRIPE NOISE LINES

In this section, we screen out the real stripe noise lines from Q according to the noise features mentioned in Section III.

1) HORIZONTAL STRAIGHT LINE DETECTION

The first step is to screen out the horizontal lines from Q. However, preceding this step, we need to expand the dimension of $\mathbf{L}$. Defining θ_{hori} as the levelness of a line:

$$\theta_{hori} = \arctan\left(\left|(y_r - y_l)/(x_r - x_l)\right|\right), \tag{4}$$

where x_l, x_r, y_l, y_r has the same meaning as in (1). $\mathbf{L}$ can be then expressed as:

$$\mathbf{L} = (x_l, y_l, x_r, y_r, \theta_{hori}). \tag{5}$$

Deleting the lines whose θ_{hori} is greater than the threshold $Th\,(\theta_{hori})$, and updating Q:

$$Q = \{\mathbf{L}\,|\theta_{hori} \leq Th\,(\theta_{hori})\}. \tag{6}$$

2) HORIZONTAL GRAYSCALE JUMP DETECTION

As described in Section III-A, there is a mutation of the grayscale values between the stripe noise pixel and the neighborhood background pixel in the horizontal direction. According to this feature, we can perform horizontal grayscale jump detection on the remaining lines in Q to delete pseudo-stripe-noise lines. The normalized horizontal grayscale jump counting factor C_{hj_norm} is defined as:

$$C_{hj\_norm} = \frac{\sum\limits_{x_l \leq u \leq x_r, v = \lfloor (y_l + y_r)/2 \rfloor} |\mathbf{b}\,(u+1, v) - \mathbf{b}\,(u, v)|}{255 \cdot |x_r - x_l|}, \tag{7}$$

where $\lfloor x \rfloor$ represents the largest integer that is not greater than x (the same below). While using (7), boundary detection is necessary to avoid the image coordinate indices from overstepping the boundary. It can be seen from (7) that C_{hj_norm}

represents the number of grayscale jumps on a unit pixel. Expanding $\mathbf{L}$:

$$\mathbf{L} = \left(x_l, y_l, x_r, y_r, \theta_{hori}, C_{hj\_norm}\right). \tag{8}$$

Through statistical analysis, we obtain that the C_{hj_norm} value of the real stripe noise is larger, which implies "horizontal grayscale jump" in Section III-A. Therefore, if we delete each line whose C_{hj_norm} value is less than the threshold $Th(C_{hj\_norm})$, every element in the updated Q will satisfy this feature:

$$Q = \left\{\mathbf{L} \left| \theta_{hori} \le Th\left(\theta_{hori}\right), C_{hj\_\mathrm{norm}} \ge Th\left(C_{hj\_\mathrm{norm}}\right)\right.\right\}. \tag{9}$$

3) STRIPE NOISE LINE CLUSTERING

In Section III-A, it is mentioned that the features of vertical grayscale jumps of the stripe noise are aimed at the stripe noise band whose height is $StripN_h$ instead of the single stripe line whose height is only one pixel, as shown in Fig. 5.

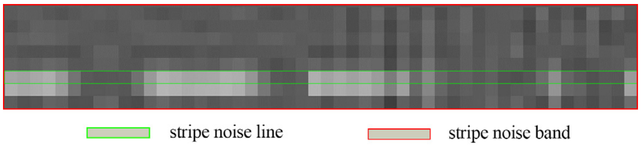

FIGURE 5. Stripe noise band and stripe noise line.

Therefore, to use this feature for further screening of Q, we need to cluster the noise lines in Q into the noise bands. First, the lines in Q are sorted by $\lfloor (y_l + y_r)/2 \rfloor$ from the smallest to largest, defining the equivalent row coordinate of the line as follows:

$$\bar{y} = \left\lfloor \frac{y_l + y_r}{2} \right\rfloor. \tag{10}$$

When the sorting is complete, clustering is executed in the vertical direction, traversing every line. The band $\mathbf{S}$ can be described as follows:

$$\mathbf{S} = \begin{bmatrix} \mathbf{L}_0 \\ \vdots \\ \mathbf{L}_p \end{bmatrix}, \tag{11}$$

where p is the total number of lines contained in a band. In addition, the lines are ordered from the smallest to the largest according to the value of the row coordinates:

$$L^i_{\bar{y}_i} > L^j_{\bar{y}_j}, \quad 0 \le j < i \le p, \tag{12}$$

where $L^i_{\bar{y}_i}$ represents the equivalent row coordinate of the i^{th} line in $\mathbf{S}$, and $L^j_{\bar{y}_i}$ represents the equivalent row coordinate of the j^{th} line in $\mathbf{S}$. Finally, updating the elements of Q, we can obtain the set of stripe noise bands:

$$Q = \{\mathbf{S}\}. \tag{13}$$

4) VERTICAL GRAYSCALE JUMP DETECTION

Similarly to the horizontal direction, there is a mutation of grayscale value between the stripe noise pixels and the neighborhood background pixels in the vertical direction. Assuming that $\mathbf{L}_m$ is the m^{th} line with the maximum brightness in $\mathbf{S}$, where $0 \le m \le p$. Thus, for each $\mathbf{S}$, the normalized vertical grayscale jump counting factor C_{hj_norm} is defined as:

$$C_{vj\_norm} = \frac{\sum\limits_{L^m_{x_l} \le u \le L^m_{x_r}, v = L^m_{\bar{y}}} \left|\mathbf{b}(u,v) - \mathbf{b}(u, v-m)\right|}{2 \cdot 255 \cdot \left|L^m_{x_r} - L^m_{x_l}\right|} + \frac{\sum\limits_{L^m_{x_l} \le u \le L^m_{x_r}, \quad v = L^m_{\bar{y}}} \left|\mathbf{b}(u,v) - \mathbf{b}(u, v+m)\right|}{2 \cdot 255 \cdot \left|L^m_{x_r} - L^m_{x_l}\right|}, \tag{14}$$

where $L^m_{x_l}$ represents the x_l coordinate of $\mathbf{L}_m$, $L^m_{x_r}$ represents the x_r coordinate of $\mathbf{L}_m$, $L^m_{\bar{y}}$ represents the equivalent y coordinate of $\mathbf{L}_m$, and C_{vj_norm} represents the number of grayscale jump on the unit pixel of $\mathbf{S}$ in the vertical direction.

The boundary detection is necessary in actual calculation, as in (7). For the real stripe noise, the value of C_{vj_norm} is generally greater than the threshold $Th(C_{vj\_norm})$, which means that after vertical grayscale jump detection, Q will be updated as:

$$Q = \left\{\mathbf{S} \left| C_{vj\_norm} \ge Th\left(C_{vj\_norm}\right)\right.\right\}. \tag{15}$$

Now, all the elements in Q represent the real stripe noise in the image, detected by our approach.

D. REMOVING THE STRIPE NOISE

In fact, Q stores the complete position information of the stripe noise in image, including the endpoint coordinates of the noise lines, which means that our approach has realized strict localization of the stripe noise. Based on this information, the neighborhood grayscale weighted replacement and the local Gaussian filtering can be used for destriping.

1) NEIGHBORHOOD GRAYSCALE WEIGHTED REPLACEMENT

This step consists of traversing all bands in Q and performing the neighborhood grayscale weighted replacement one by one. As is described in (11), there are p lines in every band, and $L^0_{\bar{y}}, L^1_{\bar{y}}, \ldots, L^p_{\bar{y}}$ $(L^0_{\bar{y}} \le L^1_{\bar{y}} \le \ldots \le L^p_{\bar{y}})$ represents the equivalent row coordinates of these p lines. We take the two adjacent lines at the upper and lower boundaries of the noise area as reference lines:

$$y_{up} = L^0_{\bar{y}} - 1, \quad y_{down} = L^p_{\bar{y}} + 1. \tag{16}$$

and perform pixel grayscale weighted replacement for the stripe noise line by line according to (17):

$$\mathbf{g}(u,v) = \omega^i_{up} \cdot \mathbf{g}\left(y_{up}, v\right) + \omega^i_{down} \cdot \mathbf{g}\left(y_{down}, v\right), \tag{17}$$

where $\mathbf{g}$ represents the raw grayscale image and (u, v) represents the pixel coordinates of the i^{th} line, where $L^i_{x_l} \le u \le L^i_{x_r}$, $v = L^i_{\bar{y}}$, $i = 0, 1, \ldots, p$. ω^i_{up} is the weight of the reference line y_{up} relative to the i^{th} line L^i and ω^i_{down} is the weight of the reference line y_{down} relative to the i^{th} line L^i. The weight calculation is as follows:

$$\omega^i_{up} = 1 - \left|\frac{L^i_{\bar{y}} - L^0_{\bar{y}}}{L^p_{\bar{y}} - L^0_{\bar{y}}}\right|$$
$$\omega^i_{down} = \left|\frac{L^i_{\bar{y}} - L^0_{\bar{y}}}{L^p_{\bar{y}} - L^0_{\bar{y}}}\right|. \quad (18)$$

During the traversal of the p stripe noise lines, ω^i_{up} gradually changes from 1 to 0, while ω^i_{down} behaves in the opposite manner, which means that for the stripe noise line closer to the upper boundary, the reference line above will occupy a higher proportion during the grayscale replacement.

2) LOCAL GAUSSIAN FILTER

In order to make the stripe noise area smoother after grayscale weighted replacement, the local Gaussian filtering is required for each band. We consider the area of the noise band as the region of interest (ROI), then we set its width equal to the width of the longest stripe noise line in this band and set its height equal to the height of the noise area:

$$R_w = \max\left(\left|L^i_{x_r} - L^i_{x_l}\right|\right), \quad i = 0, 1, \ldots, p$$
$$R_h = L^p_{\bar{y}} - L^0_{\bar{y}}, \quad (19)$$

where R_w and R_h represent the width and height of the ROI, respectively. Then each ROI is processed with a local Gaussian filter. Fig. 6 shows the effect of stripe noise localization and removal by our approach.

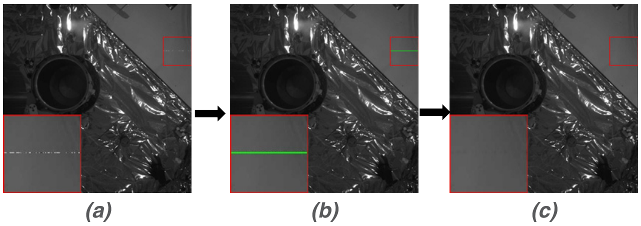

FIGURE 6. Results of stripe noise detection, localization and removal. (a) Raw image with extremely sparse horizontal stripe noise. (b) Noise localization image. (c) Destriped image.

IV. EXPERIMENTAL EVALUATION

In this section, we compare our method proposed in the paper with five recent promising destriping approaches, including the directional $\ell0$ sparse model (DSM) [8], the wavelet Fourier adaptive filter (WFAF) [16], the statistical linear destriping (SLD) [13], the unidirectional total variation (UTV) model [6] and the Low-Rank Single-Image Decomposition (LRSID) [20].

In order to evaluate the destriping effect of these approaches in a more comprehensive way, we will compare both qualitative and quantitative assessments. On the qualitative aspect, we use subjective human eye evaluation (the visual results). On the quantitative aspect, we employ two acknowledged objective image quality metrics [23], i.e., peak-signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), to evaluate the results of these methods.

Note that in our experiments, the stripe noise in the raw image is extremely sparse, which means the proportion of the noise in the raw image is very small. Therefore, in the ideal case, i.e., if the stripe noise is removed while all the useful image information is retained, the destriped image will be very similar to the raw image. In other words, we can calculate the evaluation index (PSNR, SSIM) between these two images. And this is why we don't use the no-reference image quality evaluation algorithm to evaluate the destriped image quality although there are some promising no-reference image quality assessments, such as blind pseudo-reference image (BPRI) [24], multiple pseudo reference images (MPRIs) [25], and unified content-type adaptive (UCA) blind image quality assessment [26].

A. EVALUATION INDEX

1) PSNR

PSNR is an objective evaluation metric. Assuming that both the raw image $\mathbf{n}$ and the noise-removed image $\mathbf{u}$ are grayscale (8 bits) images with a size of $M \times N$, we can calculate the PSNR between them as follows:

$$PSNR\left(\mathbf{n}, \mathbf{u}\right) = 10\log_{10}\left(\frac{255^2}{MSE\left(\mathbf{n}, \mathbf{u}\right)}\right), \quad (20)$$

where $MSE\left(\mathbf{n}, \mathbf{u}\right)$ is the mean squared error of $\mathbf{n}$ and $\mathbf{u}$:

$$MSE\left(\mathbf{n}, \mathbf{u}\right) = \frac{1}{MN}\sum_{u=1}^{M}\sum_{v=1}^{N}\left(\mathbf{n}\left(u, v\right) - \mathbf{u}\left(u, v\right)\right)^2. \quad (21)$$

A larger PSNR value corresponds to a higher image quality.

2) SSIM

SSIM is defined as an image quality metric, which is independent both on image brightness and contrast. For $\mathbf{n}$ and $\mathbf{u}$, SSIM is calculated as follows:

$$SSIM\left(\mathbf{n}, \mathbf{u}\right) = \frac{\left(2\mu_{\mathbf{n}}\mu_{\mathbf{u}} + c_1\right)\left(2\sigma_{\mathbf{nu}} + c_2\right)}{\left(\mu^2_{\mathbf{n}} + \mu^2_{\mathbf{u}} + c_1\right)\left(\sigma^2_{\mathbf{n}} + \sigma^2_{\mathbf{u}} + c_2\right)}, \quad (22)$$

where $\mu_{\mathbf{n}}$ and $\mu_{\mathbf{u}}$ are the mean values of $\mathbf{n}$ and $\mathbf{u}$, respectively. $\sigma_{\mathbf{n}}$ and $\sigma_{\mathbf{u}}$ are the standard deviations of $\mathbf{n}$ and $\mathbf{u}$, respectively. $\sigma_{\mathbf{nu}}$ is the covariance of $\mathbf{n}$ and $\mathbf{u}$, and the positive constants c_1, c_2 are used to avoid a null denominator. The value of SSIM ranges from 0 to 1. When it is equal to 1, $\mathbf{n}$ and $\mathbf{u}$ are equal.

B. EXPERIMENT RESULTS

We selected 35 nonremote-sensing images with extremely sparse stripe noise to compare the destriping effects of our approach and the other methods mentioned above. Note that all these images were captured by the Basler avA1000-100gm GigE camera with the KAI-1050 CCD sensor, and the resolution is 1024×1024. Throughout every experiment, we set $Th\left(\theta_{hori}\right) = 5$, $Th\left(C_{hj\_norm}\right) = 1/6$, and $Th\left(C_{vj\_norm}\right) =$

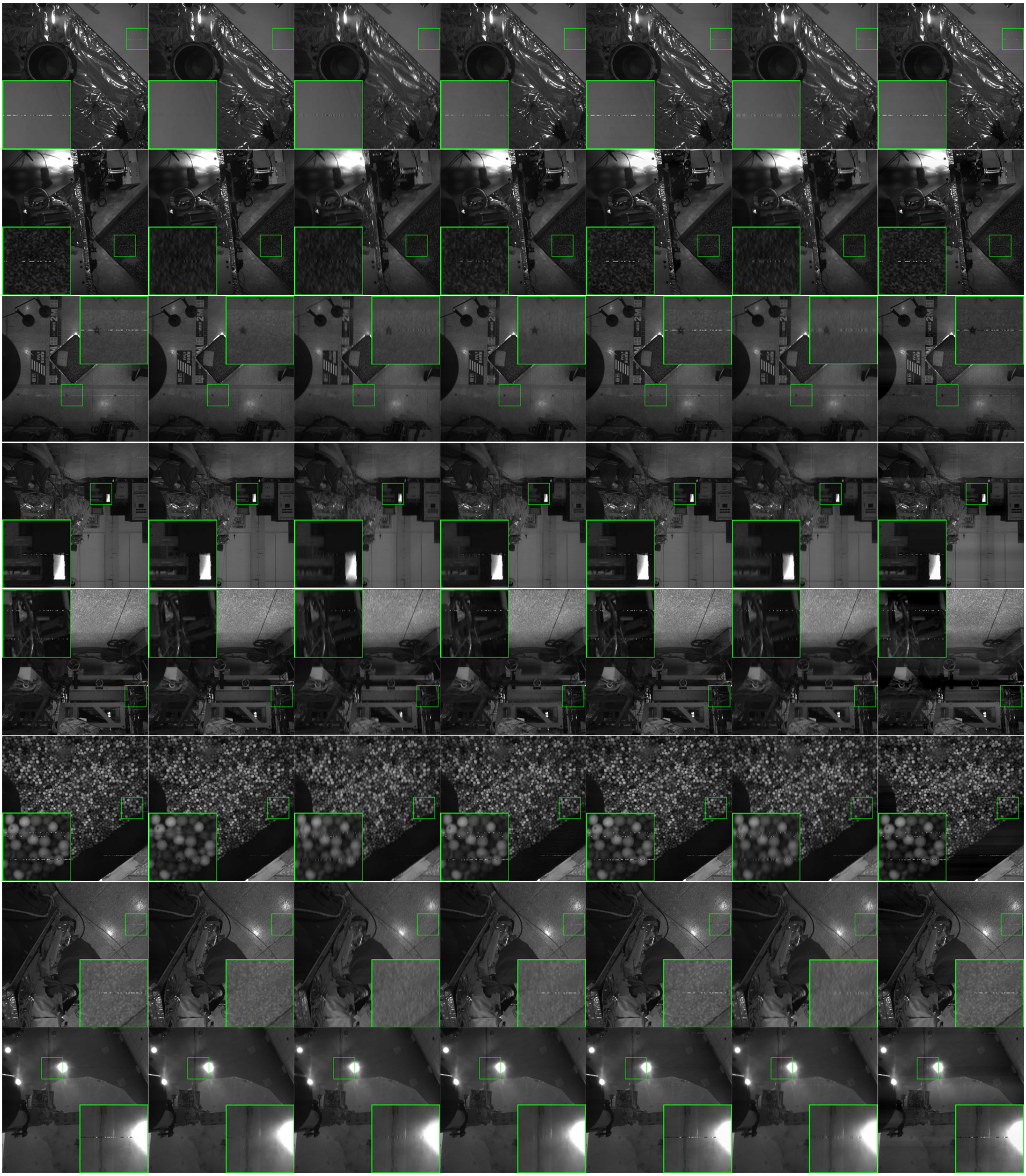

FIGURE 7. Qualitative results of destriping on test images. From left to right: raw images, the destriped images of Ours, DSM, LRSID, SLD, UTV and WFAF. Zoom in for detail or see the larger images in Fig. 8. Note that the size of these images is all 1024×1024.

1/11, which were found empirically to perform well according to our analysis on the features of these stripe noise images and massive practical tests.

1) QUALITATIVE EVALUATION

Fig. 7 shows the qualitative comparisons of each method, and these destriping images are zoomed in as shown in Fig. 8.

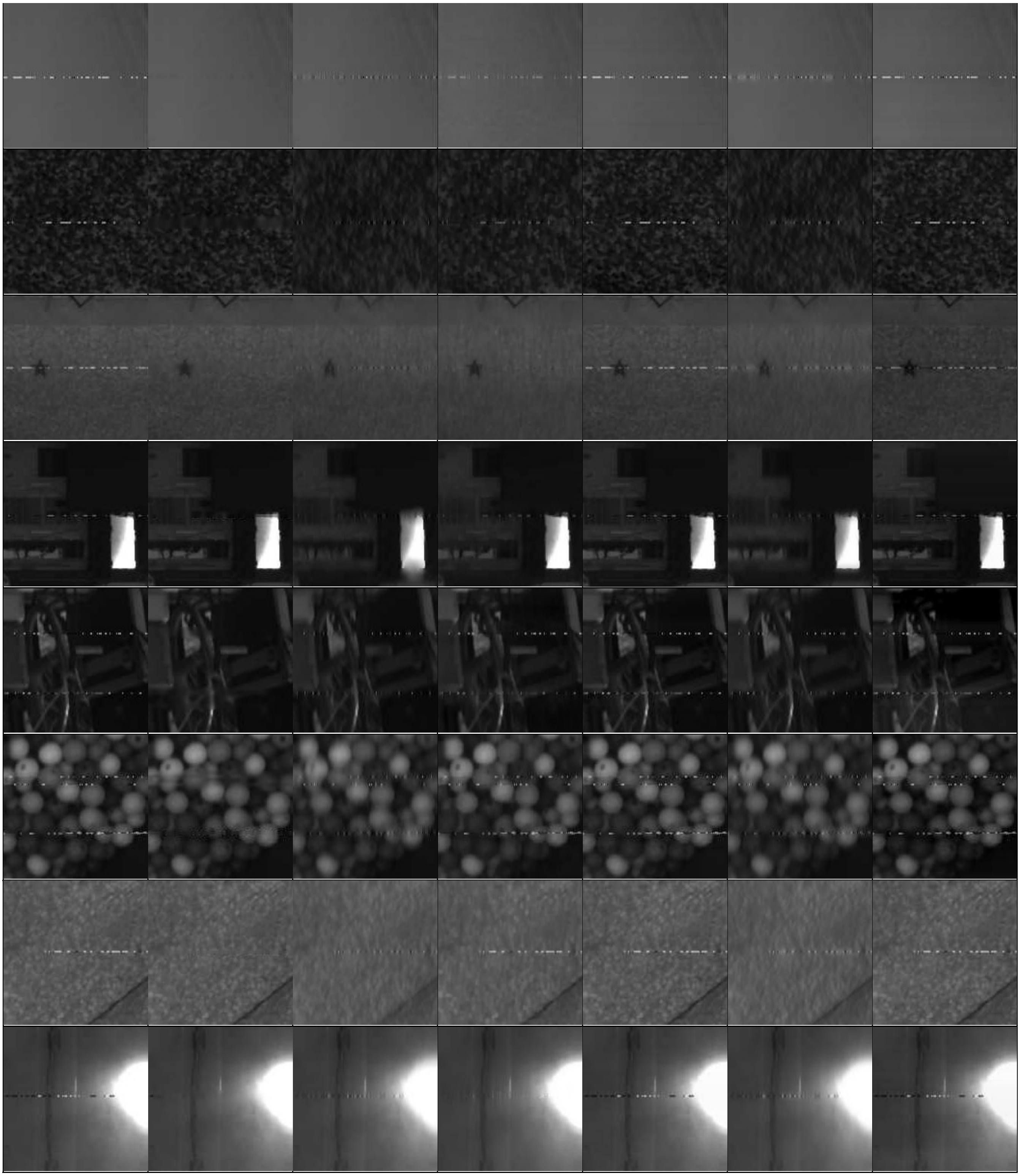

FIGURE 8. The zoom results of the destriped images in Fig. 7. From left to right: raw images, destriped images of Ours, DSM, LRSID, SLD, UTV and WFAF.

Owing to space constraints, we have only displayed eight images in this two figures. From left to right, these images represent the raw images, the destriping results of Ours, DSM, LRSID, SLD, UTV and WFAF respectively.

As we can see, the stripe noise is extremely sparse and its distribution varies both in position and length among different images, which causes the noise feature to be absent in the frequency domain. Therefore, the WFAF method (column 7) performs poorly on almost all samples and has obvious black lines along the stripe noise lines. Similarly, SLD (column 5) also fails to remove most of the stripe noise. Note that the optimization-based methods, such as DSM (column 3), LRSID (column 4) and UTV (column 6), usually show relatively good results in destriping, which can be attributed to the reason that these approaches take the intrinsic properties of stripe noise and the inherent regularity of underlying image into account. Although these approaches can remove the stripe noise to a certain extent, the destriped image become more blurred obviously and some regions belonging to the useful information of image change the intensity, which is shown clearly in the fourth row.

In the second column in Fig. 7 and Fig. 8, the proposed approach shows a much better performance than the other methods on all the test images. No matter how sparse the stripe noise is or how complex the backgrounds is, as can be seen in the third row images, our method can nearly remove the stripe noise completely and preserve image details well. These experiments exhibit excellent performance of our destriping method.

2) QUANTITATIVE EVALUATION

In order to compare the effects of these selected destriping methods more comprehensively and objectively, we use PSNR and SSIM to evaluate the quality of destriped images quantificationally.

TABLE 1. **Results of objective image quality metrics.**

	PSNR	SSIM
Ours	**41.0035±6.2272**	**0.9887±0.0091**
DSM	33.0746±3.9811	0.9647±0.0260
LRSID	38.5410±1.5275	0.9704±0.0097
SLD	**42.8881±3.1389**	**0.9958±0.0023**
UTV	36.8307±2.8512	0.9642±0.0166
WFAF	35.6446±4.9989	0.9800±0.0174

Table 1 reports the objective image quality metrics of these methods, which contains the mean value and the standard deviation calculated on 35 test images. Note that the two best results have been highlighted in bold. As shown in Table 1, all methods have high PSNR and SSIM. This is mainly because the stripe noise in the raw image is extremely sparse, which means that even if we remove it completely, there would still be significant similarities between the raw image and the destriped image. Note that the SLD has the highest value in terms of both PSNR and SSIM. As shown in Fig. 7 and Fig. 8, however, this approach does not exhibit good performance visually. On the contrary, it maybe the worst method considering its inability of destriping. Therefore, in the following analysis, we will no longer consider this approach. Although the standard deviation of the PSNR is not the smallest, the PSNR and SSIM of our method exhibit excellent performance. The LRSID obtains comparatively high values and shows good stability on this two metrics, while the UTV and DSM are a little worse because they cause more severely distortion of the original image. Therefore, from Table 1, our approach outperforms the other promising methods.

3) TIME COMPLEXITY

We run our and other methods on an Inter i7 3.4GHZ computer with 8GB RAM. Our unoptimized C++ implementation takes 148ms on average to process an image and the total number of test images is 35. We record the running time of all the approaches in Table 2. Note that the shortest running time have been highlighted in bold. As shown in this table, the computed cost of our method is minimum because we can fix the position of the stripe noise accurately and reduce unnecessary calculations. Optimization-based methods including DSM, LRSID and UTV need much more time to preform destriping than SLD and WFAF. Table1 combined with Table 2 can demonstrate that our method can achieve high-speed and high-precision simultaneously.

TABLE 2. **Processing times (sec.).**

Ours	DSM	LRSID	SLD	UTV	WFAF
0.148	9.043	26.151	0.914	13.383	0.398

The size of all 35 test images is 1024 × 1024.

V. CONCLUSION

In this paper, we proposed a novel approach for removing extremely sparse stripe noise from nonremote-sensing images.

First, we detected the position of the stripe noise lines accurately in the raw image by LPPHT. Second, we deleted the pseudo noise lines from the preselected set according to the features of horizontal grayscale jump. Then, the stripe noise lines belonging to the same noise band were clustered. Next, we screened out the real stripe noise bands based on the feature of vertical grayscale jump, which were similar to the horizontal grayscale jump. Finally, our approach used neighborhood grayscale weighted replacement and a local Gaussian filter to remove the stripe noise. The experimental results showed that our approach can be used to remove extremely sparse stripe noise from nonremote- sensing images and it outperforms some recent promising methods both in speed and precision. Most importantly, the proposed approach hardly causes any distortion of the original image.

REFERENCES

[1] P. Rakwatin, W. Takeuchi, and Y. Yasuoka, "Stripe noise reduction in MODIS data by combining histogram matching with facet filter," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1844–1856, Jun. 2007.

[2] B. Cao, Y. Du, Q. Liu, and Q. Liu, "The improved histogram matching algorithm based on sliding windows," presented at RSETE, Nanjing, China, 2011.

[3] P. Antonelli, M. di Bisceglie, R. Episcopo, and C. Galdi, "Destriping MODIS data using IFOV overlapping," in *Proc. IEEE IGARSS*, Sep. 2004, pp. 4568–4571.

[4] Y. Chen, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and J. Huang, "Stripe noise removal of remote sensing images by total variation regularization and group sparsity constraint," *Remote Sens.*, vol. 9, no. 6, p. 559, 2017.

[5] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," in *Proc. 11th Annu. Int. Conf. Center Nonlinear Stud. Exp. Math., Comput. Issues Nonlinear Sci., Comput. Issues Nonlinear Sci.* Elsevier: New York, NY, USA, 1992, pp. 259–268.

[6] M. Bouali and S. Ladjal, "Toward optimal destriping of MODIS data using a unidirectional variational model," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 8, pp. 2924–2935, Aug. 2011.

[7] Y. Chang, L. Yan, H. Fang, and C. Luo, "Anisotropic spectral-spatial total variation model for multispectral remote sensing image destriping," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1852–1866, Jun. 2015.

[8] H.-X. Dou, T.-Z. Huang, L.-J. Deng, X.-L. Zhao, and J. Huang, "Directional $\ell 0$ sparse modeling for image stripe noise removal," *Remote Sens.*, vol. 10, no. 3, p. 361, 2018.

[9] Y. Chang, L. Yan, H. Fang, S. Zhong, and W. Liao, "HSI-DeNet: Hyperspectral image restoration via convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2018.2859203.

[10] S.-W. W. Chen and J.-L. Pellequer, "DeStripe: frequency-based algorithm for removing stripe noises from AFM images," *BMC Structural Biol.*, vol. 11, no. 1, pp. 1–10, 2011.

[11] C. R. Jung and J. Scharcanski, "Adaptive image denoising and edge enhancement in scale-space using the wavelet transform," *Pattern Recognit. Lett.*, vol. 24, no. 7, pp. 965–971, 2003.

[12] B. H. Zhao *et al.*, "Methods and realization for removing inherent stripe noises in TDI-CCD images," (in Chinese), *Chin. J. Liq. Cryst. Displays*, vol. 25, no. 5, pp. 752–758, 2010.

[13] H. Carfantan and J. Idier, "Statistical linear destriping of satellite-based pushbroom-type images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 4, pp. 1860–1871, Apr. 2010.

[14] S. Huang, Z. Liu, Y. Wang, and R. Wang, "Wide-stripe noise removal method of hyperspectral image based on fusion of wavelet transform and local interpolation," *Opt. Rev.*, vol. 24, no. 2, pp. 177–187, 2017.

[15] X. Sui, Q. Chen, and G. Gu, "Adaptive grayscale adjustment-based stripe noise removal method of single image," *Infr. Phys. Technol.*, vol. 60, pp. 121–128, Sep. 2013.

[16] R. Pande-Chhetri and A. Abd-Elrahman, "De-striping hyperspectral imagery using wavelet transform and adaptive frequency domain filtering," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 5, pp. 620–636, 2011.

[17] D. Wilken, P. Feldens, T. Wunderlich, and C. Heinrich, "Application of 2D Fourier filtering for elimination of stripe noise in side-scan sonar mosaics," *Geo-Mar. Lett.*, vol. 32, no. 4, pp. 337–347, 2012.

[18] J. Liu, T.-Z. Huang, I. W. Selesnick, X.-G. Lv, and P.-Y. Chen, "Image restoration using total variation with overlapping group sparsity," *Inf. Sci.*, vol. 295, pp. 232–246, Feb. 2015.

[19] Y. Chen, T.-Z. Huang, L.-J. Deng, X.-L. Zhao, and M. Wang, "Group sparsity based regularization model for remote sensing image stripe noise removal," *Neurocomputing*, vol. 267, pp. 95–106, Dec. 2017.

[20] Y. Chang, L. Yan, T. Wu, and S. Zhong, "Remote sensing image stripe noise removal: From image decomposition perspective," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7018–7031, Dec. 2016.

[21] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 2007.

[22] J. Matas, C. Galambos, and J. Kittler, "Progressive probabilistic Hough transform," presented at the BMVC, Southampton, U.K., 1998.

[23] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. IEEE ICPR*, Aug. 2010, pp. 2366–2369.

[24] X. Min, K. Gu, G. Zhai, J. Liu, X. Yang, and C. W. Chen, "Blind quality assessment based on pseudo-reference image," *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 2049–2062, Aug. 2018.

[25] X. K. Min, G. T. Zhai, K. Gu, Y. Liu, and X. Yang, "Blind image quality estimation via distortion aggravation," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 508–517, Jun. 2018.

[26] X. Min, K. Ma, K. Gu, G. Zhai, Z. Wang, and W. Lin, "Unified blind quality assessment of compressed natural, graphic, and screen content images," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5462–5474, Nov. 2017.

YUFU QU was born in Jingbian, Shaanxi, China, in 1976. He received the M.S. and Ph.D. degrees in instrument science and technology from the Harbin Institute of Technology in 2001 and 2004, respectively, China. He joined Beihang University in 2004, where he is currently an Associate Professor with the School of Instrumentation and Optoelectronic Engineering. Since 2008, he has been a Visiting Professor with the Visual Computing Laboratory, Department of Computer Science, City University of New York, USA, as a Post-Doctoral Follow. His research interests include multimodal sensor design and integration, optical sensing, precise inspection, and machine vision.

XUAN ZHANG was born in Jinzhong, Shanxi, China, in 1993. He received the B.S. degree in information engineering from Beihang University, Beijing, China, in 2016, where he is currently pursuing the M.S. degree in instrument science and technology.

His current research interests include computer vision, image processing, and 3-D reconstruction.

QIANYI WANG was born in Dalian, Liaoning, China, in 1994. He received the B.S. degree in information engineering from Beihang University, Beijing, China, in 2017, where he is currently pursuing the M.S. degree in instrument science and technology.

His current research interests include computer vision, image processing, and 3-D reconstruction.

CHENGGUI LI was born in Jingxing, Hebei, China, in 1964. He joined Beihang University in 2004, where he is currently an Associate Professor with the School of Instrumentation and Optoelectronic Engineering. His research interests include embedded computer test and control technology, fiber Bragg grating sensors, and comprehensive performance testing and evaluation techniques for micro and nanoscale surfaces.