Image Denoising Using Spectral Component Decomposition

[1]ShameemBanu L, [2]Dr Savita Sonoli
[1]M.Tech Student, RYMEC Ballari ,[2] Professor RYMEC Ballari.

Abstract— The propose a method for local spectral component decomposition based on the line feature of local distribution. The aim is to reduce noise on multi-channel images by exploiting the linear correlation in the spectral domain of a local region. Thefirst step is to calculate a linear feature over the spectral components of an *M*-channel image, which we call the spectral line, and then, using the line, we decompose the image into three components: a single *M*-channel image and two gray-scale images. By virtue of the decomposition, the noise is concentrated on the two images, and thus our algorithm needs to denoise only the two gray- scale images, regardless of the number of the channels. As a result, image deterioration due to the imbalance of the spectral component correlation can be avoided. The experiment shows that our method improves image quality with less deterioration while preserving vivid contrast. The method is especially effective for hyperspectral images. The experimental results demonstrate that our proposed method can compete with the other state-of- the-art denoising methods.

***Index Terms*— Spectral line, local spectral component decomposition, denoising, hyperspectral image.**

I. INTRODUCTION

MULTISPECTRAL/hyperspectral images are often noisy in many situations because sensors have narrower spectral sensitivity functions and thus capture less light than normal RGB imaging devices. Whereas various applications, such as classification, target detection, spectral unmixing, and change detection need detailed and accurate spectral information [2], the noise due to, forexample,thermalelectronics and dark current, unavoidablycontaminatestheimage acquisition process [3]–[5], which disrupts detailed spectral information and furthermore degrades its performance in the llisted applications. Thus, denoising the images is a crucial phase in the preprocessing steps of these applications.

It is effective for image denoising methods to exploit inter-channel correlation as well as spatial correlation. Unlike channel-by-channel methods that tend to produce an imbalance of colors, nowadays many smoothing and denoising methods take inter-channel correlation into account to avoid color deterioration as shown by state-of- the-art denoising methods [6]–[9]. For image restoration, Blomgren and Chan [6] propose the total variation (TV) for color and other vector-valued images. Chan *et al.* [7] improve the TV method for nonlinear color models with regard to: the chromaticity-brightness (CB) and hue-saturation-value (HSV). As a nonlocal filtering approach, Color BM3D (CBM3D) [8] is one of the most powerful denoising methods for RGB images. To reduce color artifacts found in CBM3D, local color nuclear norm (LCNN) has been introduced [9], which exploits the correlation among the channels using the low rank property of each localregion.

Similarly, it is expected that, for the hyperspectral image, exploiting the correlation in not only the spatial domain but also the spectral domain improves denoisingperfor- mance because the hyperspectral images have high corre- lation between adjacent channels since they are retrieved from channels with a high spectral resolution. Channel-by- channel hyperspectral image denoising has a consequence in a low SNR because it ignores the spectral correlation [10]. By exploiting the spectral correlation, Atkinson *et al.* [11] restore hyperspectral images based on the discrete Fourier transform (DFT) and 2-dimensional discrete wavelet trans- form (2D-DWT) for decorrelation in the spectral and spatial domain, respectively. The same motivation is also found in another work [3], which utilizes local/global redundancy and correlation(RAC)inthespatialandspectraldomain.However, this method is less competitive than other state-of-the-art methods when it comes to strong noise. Dabov et al. propose a video denoising method known as VBM3D [12], which is an extension of a single channel denoising method [13], where noiseisreducedbyusingpatchesfoundinneighboringframes. This method performs well for multi-channel images, which will be demonstrated in thisstudy.

As an efficient feature to represent the inter-channel correlation of local regions, a color line is introduced in the field of color image processing [14]. The color line is a linear cluster in the RGB color space that approximates the shape of color distribution in a local region. The feature is used to model the correlation among neighboring pixels as wellas

among the channels in many images processing frameworks. This work [14] precisely distinguishes one color from another by its color line. From this idea, they implement a color-line model for some applications, i.e. segmentation, compression, color editing and saturated color correction. For demosaicing, they also use natural image properties: least color variation and minimal corner value [15]. Fattal [16] exploits the color-line pixel regularity of a single image to introduce a new dehazing method. He derives a local formation model that explains color lines in hazy scenes and uses it for estimating scene transmission. Color-line-based noise reduction has also been introduced [17]. They elaborate the color-line model with conventional filters, such as the bilateral filter and then on local means filter, to improve their performance. Ono et al. exploit the color correlation by minimizing a convex function with the LCNN [9]. This method does not have denoising capability in itself, and its main purpose is to remove color artifacts. This method outperforms other RGB denoising methods, but its superiority in hyperspectral denoising is limited.

The aim of this paper is to generalize the color line to the *M*-dimensional spectral line feature and introduce a method for local spectral component decomposition based on the spectral line. In our framework, we first calculate a linear feature over the spectral components of an *M*-channel image, which we call the *spectral line*, and then using the line we decompose the image into three components: a single *M*-channel image and two grayscale images. By virtue of the decomposition, the noise is concentrated on the two images, and thus our algorithm needs to denoise only the two grayscale images, regardless of the number of channels. Accordingly, our method yields better denoising performance than the conventional methods.

The rest of this paper is organized as follows. In Section II, we first demonstrate that the line property also holds for hyper-spectral images. Then, we describe our proposed algorithm based on the spectral line property. In Section III, we present the experimental results and compare them with those of other methods for both RGB color images and multi-channel images. Finally, we conclude our research in Section IV.

II. Algorithm

Our method is partially based on our previous conference paper [1], in which the color-line property is used for denoising. The concept of the color line is illustrated in Fig. 1, where it can be seen that a linearity is formed from each patch of a noise-free RGB image as the color line. The proposed method comes from the idea that noisy RGB images tend to contain outliers located away from the color line. The color-line property is very useful to decorrelate the channels and has been applied to image denoising to reduce discoloration artifacts in RGB images [1]. The aim of this paper is to generalize the denoising method based on the line property [1] to the *M*-dimensional spectral line feature and show its effectiveness for multi-channel images.

As for hyperspectral images, the correlation among the channels is also expected high due to the narrow spectral resolution. In line with the principle of color line for

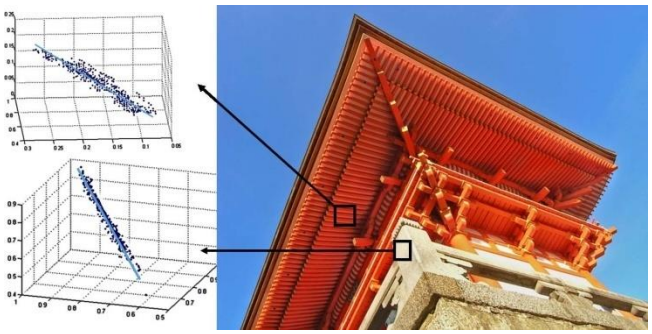


Fig. 1. Color line for RGB image.

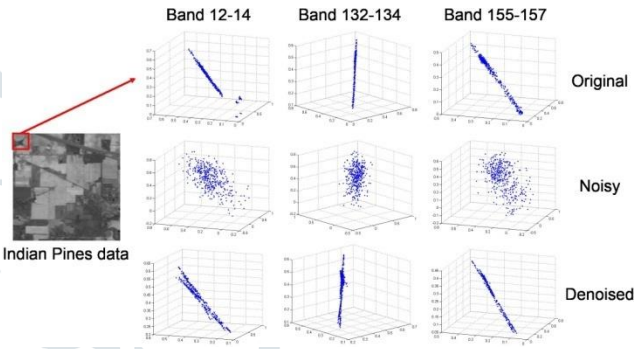


Fig. 2. Spectral line in hyperspectral images and the effect of denoising.

RGB images, the inter-channel correlation of hyperspectral images is observed by plotting the local intensity distribution. Figure 2 visualizes the linearity of three adjacent channels (the channels of 12-14, 132-134, and 155-157 in Indian Pines data). One can see from the figure that the line property also holds in the example. To find some more convincing evidence, we observe the principal component of the hyperspectral image via the principal component analysis (PCA). For various number of channels, we confirmed that the ratio between the maximum eigenvalues and the sum of all eigenvalues is high, which implies the channels are linearly correlated. For example, Pavia center data has a ratio of 0.86 for all channels, and a ratio of 0.92, 0.93, and 0.86 for 50, 35, and 10 adjacent channels,respectively.

In this regard, we exploit the property for denoising. We first extend the color line to more general multi-channels and call it the *spectral line*. We design a denoising method for multi-channel images based on the spectral line property. In the case of the multi-channel images, we consider the intensity distribution of *M* channels in every local region, which corresponds to the color distribution in RGB color images.

The spectral line is found by applying PCA to the local window centered at a pixel. In our case, a noisy input is given, which may result in inaccurate line estimation. One possible solution to address this problem is to apply pre-denoising before PCA, but the quality of the resultant image heavily depends on the pre-denoising method. For example, weak denoising does not improve the accuracy of the line estimation,

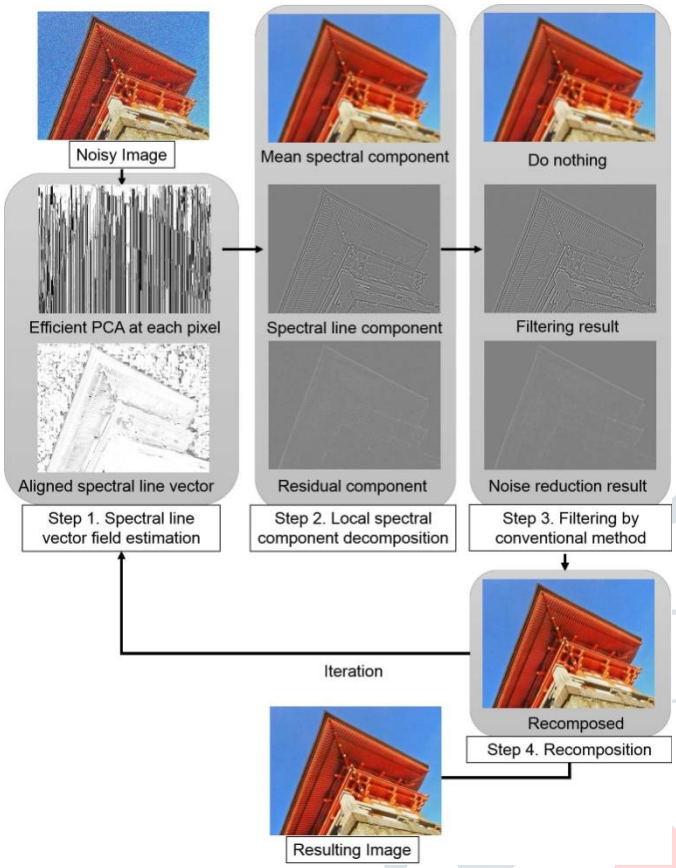


Fig. 3. Flowchart of the proposed algorithm with four main steps conducted iteratively.

but hard denoising may change the balance of the spectral information which also results in inaccurate estimation. In our method, we address this problem by an iterative fashion. First, we find an initial estimate for the spectral line from the noisy input, and then apply our decomposition algorithm. We apply a conventional denoising method to the two components yielded by the decomposition described in Sec.II-B and C. Then, we estimate the spectral line again using the denoised image, and repeat this procedure. As for the denoising method, after testing some conventional methods, we adopt BM3D, which gives the best performance with our method. The effect of the denoising is adjusted to be relatively weak at each iteration since the method is applied iteratively.

The whole procedure consists of four steps as illustrated in Fig.3.

1) We calculate the local spectral distribution in a window centered at each pixel and find the principal component for each pixel by PCA. We define the spectral line vector as the principal component and a spectral vector field as an image that has the spectral line vector at each pixel (described in Sec. II-A.1). Then, we align the direction of each vector by changing the sign so that the neighboring vector directions become smooth, which improves the resultant image as described in Sec.II-A.2.
2) Using the spectral line vector, we decompose each pixel $\mathbf{I}_i (\in \mathrm{R}^M)$ of an M-channel image into the three components: a mean spectral component $\boldsymbol{\mu}_i$ $(\in \mathrm{R}^M)$, spectral line component $D_i (\in \mathrm{R})$, and residual component $N_i (\in \mathrm{R})$ (Sec. II-B). The aim of the decomposition is to transfer noise only to the two components (spectral line and residual components). We assume that the noise is independent and zero-mean, and thus the mean spectral component tends to have little noise. Regardless of the number of channels, we only need to denoise the two components, which is especially effective for multi-channel images with $M > 3$.
3) We apply smoothing to the spectral line and residual components (Sec.II-C).
4) We reconstruct an image from the above components, and then go back to step 1 (Sec.II-D).

A. Spectral Line VectorField

1) Spectral Line Vector Estimation by PCA: The spectral line vector is formulated as the eigenvector that corresponds to the maximum eigenvalue by using PCA. The detail of the procedure is listed as follows:

1) Calculate the mean spectral component of each pixel i for each channel,

$$\boldsymbol{\mu}_i = \frac{1}{w} \sum_{j \in N(i)} \mathbf{I}_j (i = 1, 2, \ldots, k), \quad (1)$$

where k is the number of pixels in an image, w is the number of pixels in a specified filter window $N(i)$ and $\mathbf{I}_j = [I_j^1, I_j^2, \ldots, I_j^M]^\mathrm{T}$ is the intensity of neighboring pixels $j \in N(i)$. M is the number of channels (e.g. $M = 3$ for an RGB image). This calculation produces the mean $\boldsymbol{\mu}_i = [\mu_i^1, \mu_i^2, \ldots, \mu_i^M]^\mathrm{T}$, which we call the mean spectral component.
2) Calculate the covariance matrix $\mathbf{C}_i \in \mathrm{R}^{M \times M}$ of neighboring pixels around each pixel i,

$$\mathbf{C}_i = \frac{1}{w} \sum_{j \in N(i)} \mathbf{I}_j \mathbf{I}_j^\mathrm{T} - \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\mathrm{T}. \quad (2)$$

3) To obtain the spectral line vector, find the maximum eigenvalue d_i of the covariance matrix $\mathbf{C}_i$, and subsequently derive its corresponding eigenvector as the spectral line vector $\mathbf{v}_i$.

2) Alignment of Spectral Line: The resulting eigenvector $\mathbf{v}_i$ may have sign si with an ambiguity $= +1$ or -1 or. The direction of the sign should vary smoothly in our framework, otherwise the resultant image will have jaggy artifacts, which we will demonstrate with an experiment in Sec .III-A.

To make the direction of the sign smoothly vary, we adopt the Jacobi relaxation method to determine the sign. For the vector direction alignment, the sign s_i should be set to fit the dominant direction of neighboring vectors by using the inner product as the criterion. To extend the pixel-wise flip to a larger region, a multi-resolution approach is used.

In the initialization, to determine the sign of each vector $\mathbf{v}_i$ that minimizes the following energy function among neighboring pixel pairs $\{i, j\}$: $\sum_{\{i,j\}} \| s_i \mathbf{v}_i - s_j \mathbf{v}_j \|^2$, we adopt the

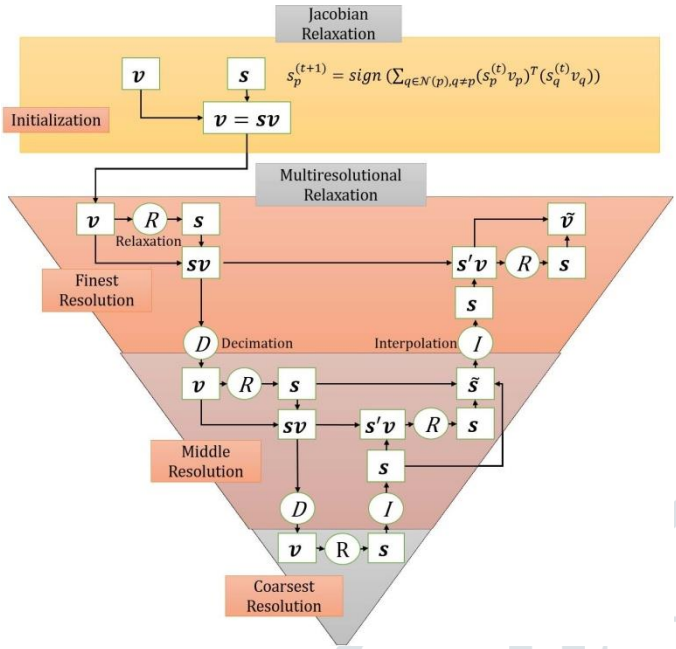


Fig. 4. Flowchart of the spectral line vector alignment.

Jacobi relaxation method [18]:

$$s_p^{(t+1)} = \text{sign}\left(\sum_{q \in N(p),\, q \neq p} (s_p^{(t)} \mathbf{v}_p)^\top (s_q^{(t)} \mathbf{v}_q)\right), \quad (3)$$

which means that the sign s_p of a pixel p is aligned with the dominant sign of 3 3 neighboring vectors $q \in N(p)$, $q \neq p$, considering the inner product. In practice, we calculate it with the use of the box filter [19] for acceleration. After finding the sign, we update the spectral line vector as

$$\tilde{\mathbf{v}}_i = s_i \mathbf{v}_i$$

The result of the initial flip is then processed by multi-grid's V-cycle [18] as a multi-resolution approach. The multi-resolution pyramid for vector and sign images in Fig. 4 is generated using Gaussian pyramid decomposition [20]. Additionally, in the decimation process, to give priority to pixels around edges that have large eigenvalues, we multiply the eigenvalue d_i as a weight for the pixel $d_i s_i \mathbf{v}_i$ and then apply the decimation filter and re-normalize the half-sized vector field. As for the multi-resolution eigenvalue images that consists of d_i, they are generated by using the same approach as the one used for the Gaussian pyramid.

Figure 5 visualizes the effect of the method, in which the pixel values represent the cosine of the angle between $\mathbf{v}_i$ and a fixed vector $\mathbf{a}$, that is $\mathbf{v}_i, \mathbf{a} / \|\mathbf{v}_i\| \|\mathbf{a}\|$ with $\langle \cdot, \cdot \rangle$ the inner product, $\|\cdot\|$ the pixel values are normalized to show it appropriately for visualization. One can see from the figure that the sign flip works effectively for the vector field.

B. Spectral Component Decomposition

Using the mean spectral component $\boldsymbol{\mu}_i$ and the spectral line vector $\tilde{\mathbf{v}}_i$, we decompose the original pixel $\mathbf{I}_i$ to the three terms. We begin with calculating the difference vector of the center pixel and the mean spectrum of the local window, $\mathbf{OI}_i = \mathbf{I}_i - \boldsymbol{\mu}_i$. Using the spectral line vector $\mathbf{v}_i$, the spectral line component D_i is calculated as the inner product of the normalized spectral line vector $\tilde{\mathbf{v}}_i$ and $\mathbf{OI}_i$,

$$D_i = \tilde{\mathbf{v}}_i^\top \mathbf{OI}_i, \quad (4)$$

which is the component of $\mathbf{OI}_i$ w.r.t. the direction of $\tilde{\mathbf{v}}_i$, since $\tilde{\mathbf{v}}_i$ is normalized to $\|\tilde{\mathbf{v}}_i\| = 1$. Then, we calculate the difference between the two vectors:

$$\mathbf{r}_i = \mathbf{OI}_i - D_i \tilde{\mathbf{v}}_i. \quad (5)$$

Finally, a residual component is derived from ℓ_2 norm of the vector $\mathbf{r}_i$,

$$N_i = \|\mathbf{r}_i\|. \quad (6)$$

Note that the dimensions of the mean spectral component $\boldsymbol{\mu}_i$, the spectral line vector $\mathbf{v}_i$ and residual vector $\mathbf{r}_i$ are the same as the number of channels, that is M, and the spectral line component D_i and the residual component N_i are scalars. The main purpose of this decomposition is to concentrate noise, which is originally scattered in all the channels of an input image, into the two components, D_i and N_i. The noise in the components is reduced in the next step.

C. Filtering

The spectral line component D_i obtained in the previous step contains noise. Consequently, denoising the spectral line components is required in the spatial domain. We refine the spectral line component by denoising it with BM3D [13]. Since we iteratively apply BM3D, the denoising effect is adjusted not to be too strong. This procedure results in the filtered spectral line component $\overline{D_i}$.

The residual component N_i contains a lot of weak noise. To filter them out, we adopt a two-phase denoising procedure. First, we apply the Geman McClure robust function [21] to reduce the noise with small intensities by the formula

$$w(N_i) = \frac{N_i^2}{k + N_i^2}, \quad (7)$$

Where k is a small constant. Afterward, it is added as the weight as

$$\hat{N}_i = w(N_i)\, N_i. \quad (8)$$

Then, the filtering step is followed by BM3D, which results in the residual component $\overline{N_i}$.

As for the mean spectral component $\boldsymbol{\mu}_i$, unlike the other components, there is no need to apply filtering because it has been already generated by averaging (1).

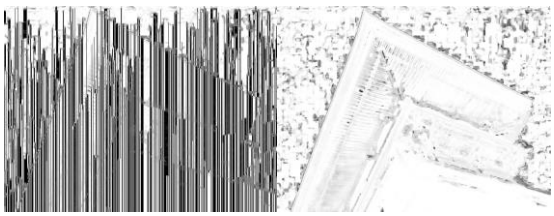


Fig. 5. The effect of sign flip: sign maps (left) before and (right) after signflip.

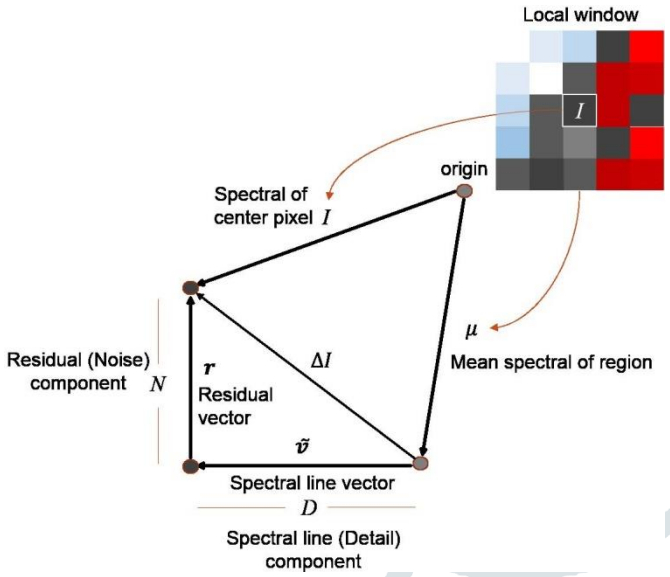


Fig. 6. Spectral decomposition: In local spectral component decomposition, eachpixelisdecomposedintoaspectralline vector,aspectrallinecomponent, and a residualcomponent.

D. Recomposition

The final step is recomposition of the resulting image from its constituent components. The expected image can reconstructed as follows (see Fig.6):

$$I_i = \boldsymbol{\mu}_i + D_i\tilde{\mathbf{v}}_i + \frac{\overline{N_i}}{N_i}\mathbf{r}_i. \quad (9)$$

III. Experiment

In the experiment, all the methods are implemented in MATLAB, except for calculating the mean spectral component, where the MEX file is used for running the box filter inC.

A. Effect of SignFlip

First, to discuss the importance of sign flip before filtering, we perform an experiment that emphasizes how it works on an image. The Jacobi relaxation used for the vector sign alignment significantly improves the performance by flipping the sign to the same direction as the local mean spectrum. Figure 7 shows the effect of the vector sign flip in our method. The top and bottom row show the results when the sign flip is not performed and performed, respectively. After PCA, the generated spectral line vectors are not smooth (top left). If we continue to the next steps without this procedure, the spectral line components and the filtered result are affected and fail to preserve the details. To avoid this problem, we use the Jacobi relaxation method for the spectral line vectors. As a result, the method can perform better as shown in the bottom images.The comparison of the final resultant images is depicted in Fig. 8. The difference can easily be noticed in some regions of the image.

B. RGB ColorImage

1) Parameter Setting: The parameters of the conventional methods are adjusted so as to give the best evaluation values.

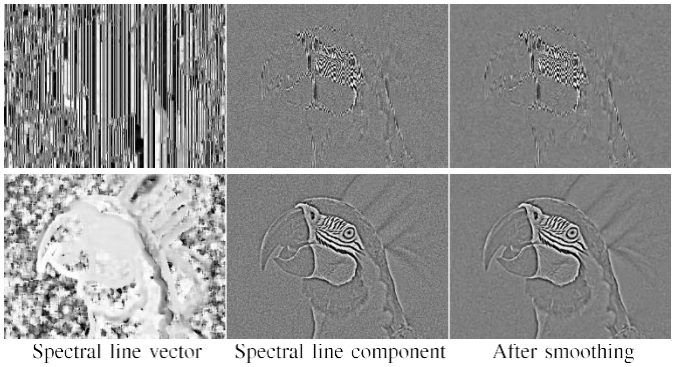


Fig. 7. The results obtained without sign flip (top) and with sign flip (bottom) for the spectral line vector (left), the spectral line component (middle), and the smoothingresult of spectral line component (right).

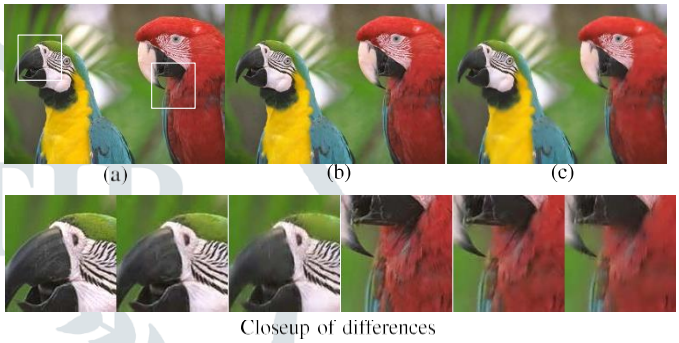
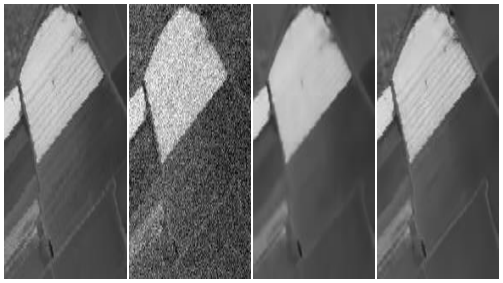


Fig. 8. Effect of sign flip in the resultant image. From the left, (a) original image, (b) our method, and (c) our method without sign flip.

On the other hand, for the conventional methods used in combination with our method, we set their parameters to give a similar degree of denoising. Then, we will show that our method preserves more vivid contrast than the others with the same degree ofdenoising.

To achieve the expected results, the number of main iterations is set to 4. To calculate the spectral orientation, we set half of the window size to 3. We found after some trial-and-error that these values yield sufficient results for most of the tested images.

2) Experimental Results: Before evaluating the whole of our method, we examine the contribution of our decomposition step. Only for Fig. 9, we use the anisotropic diffusion to denoise the spectral line component instead of BM3D to fairly evaluate the validity of the decomposition in the denoising process, and compare it with the stand-alone anisotropic diffusion. Thus, the two rows of Fig. 9 show the comparison with the anisotropic diffusion. Figure 9 (a) and (b) depict original images and noisy images, respectively, in which we add the Gaussian noise of a standard deviation σ 0.06
for normalized intensity range [0, 1]. Figure 9 (c) and (d) are the results of the anisotropic diffusion alone, where the method is applied in the RGB and YCbCr color spaces, respectively, and (e) is obtained by the anisotropic diffusion after our decomposition. One can see that our method is able to significantly reduce the discoloration artifact, compared to the anisotropic diffusion used alone. The artifacts can be noticed distinctly in the background part of theimages. Furthermore,



(b) (c) (d)

Fig. 9 Cropped Salinas data, 47th channel for hybrid LCNN comparison. (a) Original (PSNR). (b) Noisy. (c) LCNN (35.20 dB). (d) Ours (37.59 dB).

TABLE II
MPSNR COMPARISON

Method	Cuprite	Indian	Pavia-C	Pavia-U	Salinas
NLM	31.09	33.99	26.25	27.28	37.54
PRI-NL-PCA	31.69	35.61	31.59	32.35	36.88
VBM3D	33.44	36.95	33.10	**33.69**	38.94
VBM4D	32.72	36.35	31.68	32.42	38.53
Ours	**36.03**	**37.84**	**33.17**	32.76	**41.35**
LCNN	34.77	36.62	33.84	33.90	39.24
Ours	**36.33**	**37.84**	**33.89**	**34.34**	**41.44**

(For the comparison with LCNN, the images are cropped so that the size is a multiple of the window size, and our method is simulated under the same condition. Note that according to our experiments, the PSNR of LCNN becomes a little lower without this treatment.)

a noticeable visual difference. Meanwhile, the Salinas and Pavia University images show more significant visual difference. As depicted by the red rectangle, the original image has stripes that can be recognized easier on our resultant images than on those of the other methods.

For the Indian Pines, the overall image has a distinct difference in appearance between ours and the compared images. Both of VBM3D and VBM4D produce images with too smooth areas that results in a salient difference compared to the original image. In addition, they fail to preserve the straight line features in the images. Meanwhile, our method still performs better than the others. It is clear that for perceptual appearance, our resulting image is the most similar to the original one. Despite the LCNN improvement, Fig. 16 to 20 show that this hybrid method is lower in visual quality than ours. It takes about 43 seconds to employ our algorithm for a 200 ×200 35× image on the PC with Intel(R) Core(TM) i5-4690 CPU @ 3.5 GHz, using unoptimized MATLAB codes. This is mainly due to the PCA calculation of 35 dimensions in our algorithm that costs muchtime.

IV. CONCLUSION

A new denoising method based on the spectral line has been proposed for the remote sensing field. Hyperspectral image denoising using a spectral line vector field uses the correlation among spectral information in the local region. The vectors are obtained by local spectral component decomposition followed by iterative filtering steps. Filtering the spectral line component and residual component gives significant effectsin Reducing the noise and smoothing results the image. Moreover, the use of local spectral components contributes to achieving better results compared with the result of the stand-alone con- ventional method. The experiment demonstrated that the pro- posed method successfully achieved competitive performance compared to other powerful denoising methods. However, the increase in noise power and the number of channels processed affects the complexity of achieving more accurate spectral line vector estimation. Future work may involve solving this computationalcomplexity.

REFERENCES

[1] K. Shirai, M. Okuda, and M. Ikehara, "Color-line vector field and local color component decomposition for smoothing and denoising of color images," in *Proc. IAPR 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp.3050–3053.

[2] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P.Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysisandfuturechallenges,"*IEEEGeosci.RemoteSens.Mag.*,vol.1, no. 2, pp. 6–36, Jun. 2013.

[3] Y.-Q. Zhao and J. Yang, "Hyperspectral image denoising via sparse representation and low-rank constraint," *IEEE Trans. Geosci. Remote Sens.*,vol.53,no.1,pp.296–308,Jan.2015.

[4] P. W. Yuen and M. Richardson, "An introduction to hyperspec- tral imaging and its application for security, surveillance and targetacquisition,"*Imag.Sci.J.*,vol.58,no.5,pp.241–253,2010.

[5] T. Skauli, "Sensor noise informed representation of hyperspectral data, with benefits for image storage and processing," *Opt. Exp.*, vol. 19, no. 14, pp. 13031–13046, Jul.2011.

[6] P. Blomgren and T. F. Chan, "Color TV: Total variation methods for restoration of vector-valued images," *IEEE Trans. Image Process*, vol. 7, no. 3, pp. 304–309,Mar. 1998.

[7] T.F.Chan, S. H. Kang, and J. Shen, "Total variation denois- ing and enhancement of color images based on the CB and HSV color models," *J. Vis. Commun. Image Represent.*, vol. 12, no. 4, pp. 422–435, Dec.2001.

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, vol. 1. Sep./Oct. 2007, pp.I-313–I-316.

[9] S. Ono and I. Yamada, "A convex regularizer for reducing color artifact in color image recovery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp.1775–1781.

[10] B. Demir and S. Ertürk, "Improved hyperspectral image classification with noise reduction pre-process," in *Proc. 16th Eur. Conf. Signal Process.*, Aug. 2008, pp.1–4.

[11] I. Atkinson, F. Kamalabadi, and D. L. Jones, "Wavelet-based hyperspec- tral image estimation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*,vol.2.Jul.2003,pp.743–745.

[12] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," in *Proc. Eur. Conf. Signal Process.*, Sep. 2007, pp.145–149.