Noise Removal based on Surface Approximation of Color Line

Koichiro Manabe, Takuro Yamaguchi, Masaaki Ikehara
EEE Dept., Keio Univ., Yokohama, Kanagawa 223-8522, Japan
Email: {manabe,yamaguchi,ikehara}@tkhm.elec.keio.ac.jp

***Abstract*—In a local region of a color image, the color distribution often takes the form of a linear line in the RGB space. We call this property "Color Line". We propose a denoising method based on this property. When the noise is added on an image, the color distribution spreads from Color Line. The denoising is achieved by reducing the spread. In conventional methods, Color Line is assumed to be only a single line, but actual distribution takes various shapes such as a single line, two lines, and a flat surface and so on. In our method, we estimate the distribution in more detail using surface approximation and denoise each patch by reducing the spread depending on the color distribution types. In this way, we can achieve better denoising result than a conventional method.**

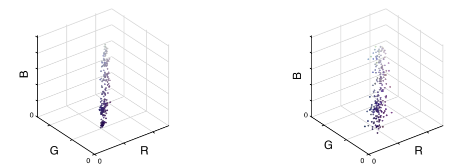
I. Introduction

Recently, we often take many digital pictures to post on SNS. However, photographs taken in dark places are often affected by Gaussian noise. We would like to obtain high quality images by denoising the deteriorated images.

There are many methods to denoise, such as filtering methods and patch based methods, etc. In patch based methods, we divide the image into small local regions to process, for example, Nonlocal-Means [1], BM3D [2], Color-BM3D (CBM3D) [3], and so on. In CBM3D, we use the self similarity and denoise by applying three dimensional transformation to the group of similar patches. CBM3D is known as a state-of-the-art color image denoising method. Some denoising methods to improve CBM3D have been researched. One of these methods is to use the Color Line property after denoising by CBM3D [4]–[7].

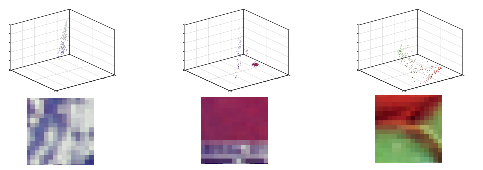
We can observe that the color distribution of a noiseless color image often takes the form of a linear line in the RGB space as shown in Fig. 1(a), and this property is called "Color Line" [8]. The rise of color line model has been utilized in dehazing [9], [10], image matting [11] and deblurring [12]. When gaussian noise is added to a color image, the color distribution spreads as shown in Fig. 1(b). However, Fig. 2 shows that an actual color distribution takes various shapes such as a single line, two lines, and a flat surface and so on. In a local edge region and a texture with high frequency component, the color distribution may shape two lines or a flat surface. However, conventional methods based on the Color Line property consider only a single line property [4]–[7]. So, our method judges the types of the Color Line property and brunches processing according to the types. With these processes, we improve CBM3D by considering many types of color distributions and can achieve better denoising performance than the conventional method.

International Workshop on Advanced Image Technology 2018 (IWAIT2018), Jan. 7–10, 2018, Chaing Mai, Thailand.
978-1-5386-2615-3/18/$31.00 ©2018 IEEE.



(a) In a noisefree region (b) In a noisy region

Fig. 1. Color Line property



(a) A single line (b) Two lines (c) A flat surface

Fig. 2. Some examples of Color Line

II. The Details of the Proposed Algotirhm

Conventional methods can denoise the image based on a single line property which corresponds to the principal vector of the Color Line property obtained by SVD [7], as shown in Fig. 3(a). However, they cannot denoise the patches with two lines or a flat surface property because the principal vector by SVD does not represent the color distribution in such the patches, as shown in Fig. 3(b).

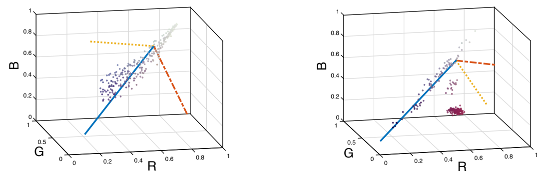
Our method proposes two major improvements from the conventional method. First is that our method estimates the Color Line property by two orthogonal surfaces called "projection surfaces", as shown in Fig. 4. Second is that our method judges the type of the Color Line property. We denote the types of the Color Line property in number as s_k for simplicity, as shown in Table I.

Fig. 5 shows the flowchart of our method. We explain the processes in the k-th patch according to the flowchart.

A. Computing projection surface 1

We first compute the normal vector of a projection surface. The normal unit vector of the projection surface is denoted as $[l_k, m_k, n_k]$. Then the projection surface is as follows.

$$l_k r + m_k g + n_k b = 1 \quad (1)$$



(a) A single line property (b) Two lines property

Fig. 3. Principal vectors of Color Line property by SVD

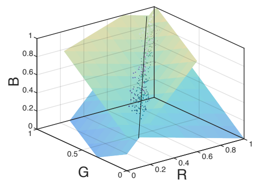


Fig. 4. Color Line property and two projection surfaces

This surface is called "projection surface 1".

B. Computing classification surface

As shown in Fig. 6(a), the classification surface is used to group the pixels of the patch, and to determine the type of the Color Line property. Classification surface has following three requirements:

1) Orthogonal to the projection surface 1 in Eq. (1)
2) Parallel to the approximation line
3) Go through the weighted average point $\boldsymbol{p}_k^w = (r_k^w, g_k^w, b_k^w)$ of the color distribution

In requirements 2, the line is computed with the least-squares method.

According to conditions 1 and 2, we obtain the normal unit vector of the classification surface $[p_k, q_k, r_k]$.

In requirements 3, the color distribution often has a lump of points as shown in Fig. 6(a). If a classification surface simply goes through the arithmetic average point , it will be affected by this lump. Hence, the points should be weighted avoiding the effect of this lump. Considering a sphere of radius r_{weight} around i-th point in the k-th patch, let N_i^w be the number of the points in this sphere. The weighting coefficient w_i of the i-th point is defined as

$$w_i = \frac{1}{N_i^w \times \sum \frac{1}{N_i^w}} \tag{2}$$

The weighted average point is calculated with Eq. (2) as follows

$$\boldsymbol{p}_k^w = \frac{1}{N} \sum_{i=1}^{N} w_i \boldsymbol{p}_i \tag{3}$$

where N is the number of all the points and $\boldsymbol{p}_i$ is the position vector of i-th point in the k-th patch.

From the above, the classification surface is represented as follows.

$$p_k(r - r_k^w) + q_k(g - g_k^w) + r_k(b - b_k^w) = 0 \tag{4}$$

TABLE I
THE TYPES OF THE COLOR LINE PROPERTY IN NUMBER

s_k	the type
1	A single line property
2	Two lines property
3	A flat surface property

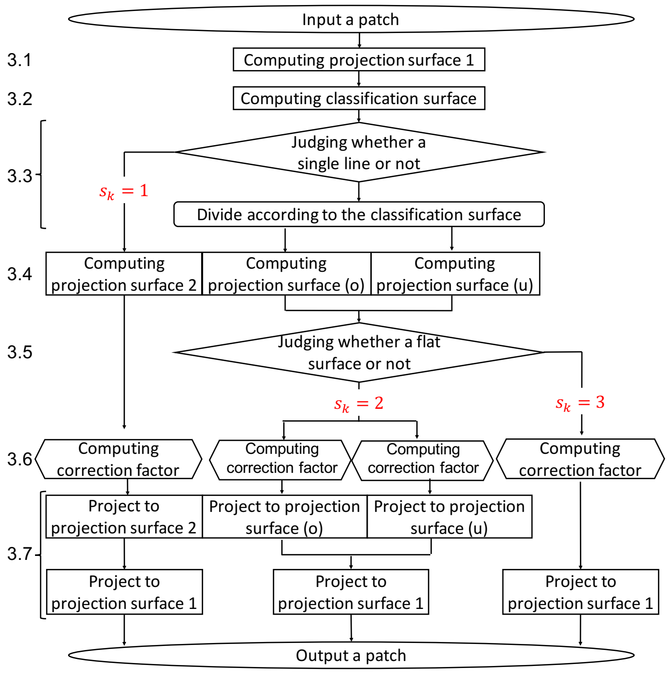


Fig. 5. Flowchart of our method

C. Judging the type and dividing Color Line

We judge whether the Color Line property is a single line or not with the projection surface 1 and the classification surface. If the Color Line has a single line property, sums of distances from each surface to all the points $d_{1,k}$ and $d_{c,k}$ are approximately equal, as shown in Fig. 7(a). On the other hand, if it has two lines or a flat surface property, they are not equal, as shown in Fig. 7(b). We set the threshold t, and the type of the Color Line property s_k can be determined from the following requirement.

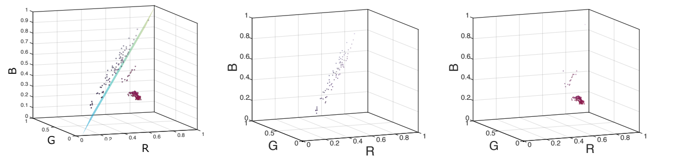
$$s_k = \begin{cases} 1 & \text{if } |d_{1,k} - d_{c,k}| \leq t \\ 2 \text{ or } 3 & \text{otherwise} \end{cases} \tag{5}$$

If Color Line property is not a single line, we classify the distribution into the over and under regions according to the classification surface, as shown in Fig. 6. In addition, we denote the number of the points in each region as N_k^{over} and N_k^{under}, respectively.

D. Computing projection surface 2

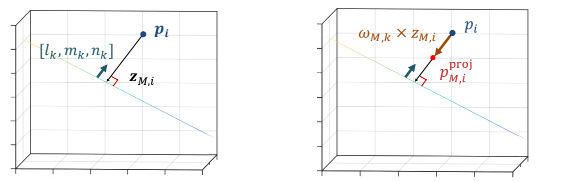
We calculate the second projection surface as the orthogonal surface to the projection surface 1. This surface is called "projection surface 2". The normal unit vector of this surface is denoted as $[l'_k, m'_k, n'_k]$. Note that it is different from the classification surface. The projection surface 2 can be obtained in the same way as sec. II-B except that it passes through the arithmetic average point $\boldsymbol{p}_k^0 = (r_k^0, g_k^0, b_k^0)$. It is represented as follow.

$$l'_k(r - r_k^0) + m'_k(g - g_k^0) + n'_k(b - b_k^0) = 0 \tag{6}$$



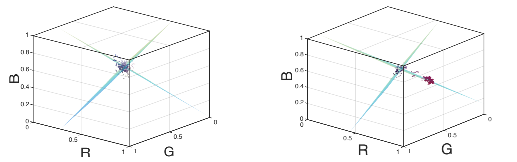
(a) Classification surface (b) The over region (c) The under region

Fig. 6. Classification surface and dividing two lines distribution



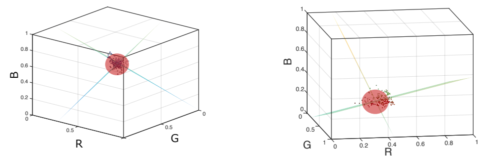
(a) The projection vector (b) The processing of projection

Fig. 8. How to project points onto the approximate surface



(a) A single line distribution (b) Two lines distribution

Fig. 7. To judge the number of lines



(a) The example of small spread (b) The example of large spread

Fig. 9. The expanse of Color Line property

If the type of the Color Line property is not a single line, the projection surface 2 should be computed for over and under regions, respectively.

E. Judging whether a flat surface distribution or not

Whether the Color Line property is a single line or not can be judged in sec. II-C. Next, we should judge whether the Color Line property is two lines or a flat surface. In the case of the two lines property, each of classified distributions has a single line property. On the other hand, either of them has the flat surface property. Hence, we can judge whether the Color Line property is two lines or a flat surface by repeating the processing in sec. II-C for both over and under regions.

We denote the sums of the distances from all the points to the projection surface M in the over and under regions as $d_{M,k}^{\mathrm{over}}$ and $d_{M,k}^{\mathrm{under}}$, where $M \in \{1, 2\}$, and the thresholds as t_k^{over} and t_k^{under}. Then, the type of Color Line property can be determined as follows.

$$s_k = \begin{cases} 2 & \text{if} \begin{cases} |d_{1,k}^{\mathrm{over}} - d_{2,k}^{\mathrm{over}}| \leq t_k^{\mathrm{over}} \text{ or} \\ |d_{1,k}^{\mathrm{under}} - d_{2,k}^{\mathrm{under}}| \leq t_k^{\mathrm{under}} \end{cases} \\ 3 & \text{otherwise} \end{cases} \tag{7}$$

Now, we can judge whether the Color Line property is a single line, two lines or a flat surface, from Eq. (5) and Eq. (7).

F. Computing correction factor

After judging the type of the Color Line, we calculate the correction factor $\omega_{M,k}$. It represents the ratio to reduce the distances from the points to the projection surfaces in order to denoise the image by reducing the spread of the color distribution, as shown in Fig. 8(b). The correction factor has following two requirements.

1) The appropriateness of projection surface of Color Line

2) The spread of the Color Line

For requirements 1, the ideal projection surface halves the number of points, however the computed projection surface often does not. Therefore we need to calculate the appropriateness of the each projection surface M. Let $\omega_{M,k}^{\max}$ be the appropriateness, $N_{M,k}^{\mathrm{O}}$ and $N_{M,k}^{\mathrm{U}}$ be the number of the points over and under the projection surface M, respectively. The appropriateness $\omega_{M,k}^{\max}$ is calculated as follows.

$$\omega_{M,k}^{\max} = \begin{cases} \frac{N_{M,k}^{\mathrm{U}}}{N_{M,k}^{\mathrm{O}}} & \text{if } N_{M,k}^{\mathrm{O}} > N_{M,k}^{\mathrm{U}} \\ \frac{N_{M,k}^{\mathrm{O}}}{N_{M,k}^{\mathrm{U}}} & \text{otherwise} \end{cases} \tag{8}$$

If the Color Line property is two lines, $\omega_{M,k}^{\max}$ should be calculated for the over and under regions, respectively.

For requirements 2, when the spread of the color distribution is large as shown in Fig. 9(b), which means the noise is considered as large, the correction factor $\omega_{M,k}$ should be set large. To determine the spread of Color Lines, we first consider a cylinder with radius of r, which is changed as a variable, centered at the intersection of two projection surfaces. Let $N_k^{\mathrm{proj}}(r)$ be the number of points in the cylinder, and $r_k^{\max}$ be the radius when $N_k^{\mathrm{proj}}(r)$ is equal to N. Fig. 10 shows the graph of $N_k^{\mathrm{proj}}(r)$. Then, the spread of color distribution e_k is as follows.

$$e_k = \frac{\sqrt{(\int_0^{r_k^{\max}} N dr)^2 - (\int_0^{r_k^{\max}} N_k^{\mathrm{proj}}(r) dr)^2}}{\int_0^{r_k^{\max}} N dr} \tag{9}$$

We consider a flat board of thickness $2r$ centered at the projection surface 1 for a flat surface property.

We use $\omega_{M,k}^{\max}$ and e_k to compute the correction factor $\omega_{M,k}$.

$$\omega_{M,k} = \omega_{M,k}^{\max} \, e_k \tag{10}$$

G. Projection processing

Fig. 8 shows how the proposed method denoises by reducing the spread of noised color distribution. This process is performed for the each projection surface M. The projection vector $\boldsymbol{z}_{M,i}$ is the normal vector from i-th point to the each projection surface.

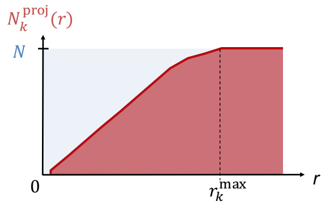


Fig. 10. The graph of N_k^{proj}

TABLE II
PARAMETER SETTING

parameter	r_{weight}	t	t_k^{over}	t_k^{under}
	$\sigma/1000$	$0.007 \times N$	$0.007 \times N_k^{\mathrm{over}}$	$0.007 \times N_k^{\mathrm{under}}$

The projection processing is represented with the following equation

$$\boldsymbol{p}_{M,i}^{\mathrm{proj}} = \boldsymbol{p}_i + \omega_{M,k}\, \boldsymbol{z}_{M,i} \begin{cases} M \in \{1,2\} & \text{if } s_k = 1 \text{ or } 2 \\ M \in 1 & \text{if } s_k = 3 \end{cases} \tag{11}$$

where $\boldsymbol{p}_i$ means the position vector of a point and $\boldsymbol{p}_{M,i}^{\mathrm{proj}}$ means the position vector of the denoised point. Thanks to this processing, the spread of the color distribution by noise is reduced, which means we get the denoised patch. This processing is applied to all the patches to obtain the denoised image.

III. SIMULATION

We test the performance of the proposed method by simulation with the twelve test images with PSNR, then compared to CBM3D and conventional method LCNN [7]. We add Gaussian noise with the standard deviation $\sigma = 30$ on these images. In the conventional and the proposed method, we pretreated with CBM3D [3] to recover the Color Line property.

A. Parameter setting

The parameters of our method are denoted as follows. We divide the input images into $N = 8 \times 8$ patches, sliding 4 pixels at times. The parameters of our method are set, as shown in Table II. They are denoted in sec. II-B, II-C and II-E.

B. Result

Table III shows the results of PSNR compared to CBM3D and LCNN. The proposed method produces good results with many images. The images with relatively large smooth domains considerably improved, for example, Airplane and Milkdrop. Earth and Sailboat have many regions having Color Line property well, and proposed method can denoise these regions compared to conventional method. In addiition, two lines and a flat surface distribution are tend to appear in local edge region and high frequency area. So, in images with intense color difference, such as Jelly-beans and Pepper, proposed method has improved well, while the conventional method has not improved so much. However, the results of Aerial and Mandrill are lower than those of CBM3D. This is because we can't set the correction factor well due to the large spread of color distribution in the original image and Color Line don't appear well in the case of these images.

TABLE III
COMPARISON OF PSNR

image	CBM3D	LCNN	our
Aerial	26.0221	**26.0630**	25.9361
Airplane	29.2688	29.3720	**29.6942**
Ballon	32.1015	32.1825	**32.3579**
Earth	29.7775	29.8996	**30.2420**
House	31.8035	31.8709	**31.8869**
Jelly-beans	31.4608	31.5751	**31.7716**
Lenna	30.1447	30.2200	**30.3999**
Mandrill	25.3524	**25.4090**	25.2845
Milkdrop	31.9797	32.0153	**32.5647**
Pepper	29.3119	29.3470	**29.5124**
Sailboat	28.6915	28.8290	**29.1311**
Tree	27.5878	27.6220	**27.7009**

IV. CONCLUSION

We proposed the new denoising method based on Color Line using the approximation surface for removing Gaussian noise. In the conventional method, the Color Line property is assumed to be only a single line and estimate the principal vectors by SVD. Proposed method estimates the appropriate type of Color Line by computing approximation surfaces from the color distribution. We achieved the purpose of improving CBM3D thanks to estimating the type of the Color Line property appropriately.

Future work includes denoising of complex images, such as Aerial and Mandrill, where Color Line does not appear well.

REFERENCES

[1] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, June 2005, pp. 60–65 vol. 2.

[2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3d filtering," pp. 606 414–606 414–12, 2006. [Online]. Available: http://dx.doi.org/10.1117/12.643267

[3] ——, "Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 1, Sept 2007, pp. I – 313–I – 316.

[4] K. Shirai, M. Okuda, and M. Ikehara, "Color-line vector field and local color component decomposition for smoothing and denoising of color images," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov 2012, pp. 3050–3053.

[5] W. C. Tu, C. L. Tsai, and S. Y. Chien, "Collaborative noise reduction using color-line model," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 2465–2469.

[6] S. Ono and I. Yamada, "A convex regularizer for reducing color artifact in color image recovery," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 1775–1781.

[7] ——, "Color-line regularization for color artifact removal," *IEEE Transactions on Computational Imaging*, vol. 2, no. 3, pp. 204–217, Sept 2016.

[8] I. Omer and M. Werman, "Color lines: image specific color representation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, June 2004, pp. II–946–II–953 Vol.2.

[9] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 1956–1963, 2009.

[10] R. Fattal, "Dehazing using color-lines," *ACM Trans. Graph.*, vol. 34, no. 1, pp. 13:1–13:14, Dec. 2014. [Online]. Available: http://doi.acm.org/10.1145/2651362

[11] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228–242, Feb 2008.

[12] N. Joshi, C. L. Zitnick, R. Szeliski, and D. J. Kriegman, "Image deblurring and denoising using color priors," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 1550–1557.