# Hyperspectral Gaussian Filtering: Edge-Preserving Smoothing for Hyperspectral Image and Its Separable Acceleration

Shu Fujita, Norishige Fukushima
Nagoya Institute of Technology, Japan

**For hyperspectral imaging, we proposed an edge-preserving filter, named hyperspectral Gaussian filtering, and its separable implementation for accelerating the proposed filter. Experimental results show that the acceleration has superior performance to the brute-force implementation and the other state-of-the-art methods in denosing. Also, we successfully achieves 70-times speed-up with the acceleration.**

## 1 Introduction

Multi/Hyper-spectral imaging has various applications [1] and extends functionality and sensibility in robot vision and remote sensing fields. Denoising of hyperspectral images is an essential issue for these applications.

Bilateral filtering [2] is one of the most successful tools in 2D image processing for denoising and edge-preserved smoothing as pre-filtering of segmentation. For multispectral imagery, a variant of bilateral filtering—dual bilateral filtering— is proposed [3]. The dual bilateral filter uses two range kernels for filtering RGB and IR images. Vector bilateral filtering [4] is proposed for hyperspectral image denoising. The vector bilateral filter extends the bilateral filter to control range kernel distribution by band-per-band. To improve filtering of hyperspectral images, we can utilize redundancy of the wavelength dimension for improving the denoising performance. Bilateral filtering is, however, cost consuming, and its higher-dimensional version is more cost-consuming than bilateral filtering.

Accelerated bilateral filtering [5] is $O(1)$-order. The costs of the accelerated filter, however, is exponentially increased by increasing channels. Hyperspectral images have many channels; thus, we need more tolerant acceleration methods to the curse of dimensionality.

In this paper, we propose an effective filtering for hyperspectral images, named *hyperspectral Gaussian filtering (HGF)* and its acceleration technique of separable implementation.

## 2 Hyperspectral Gaussian Filtering

### 2.1 Definition

For hyperspectral Gaussian filtering, we create 3D kernels for a hyperspectral cube. Let $I^s_{\boldsymbol{p}}$ be a pixel value at position $\boldsymbol{p}{=}(x_{\boldsymbol{p}},y_{\boldsymbol{p}})$ with wavelength $s$, and a filtered pixel value is:

$$\bar{I}^s_{\boldsymbol{p}} = \frac{1}{K^s_{\boldsymbol{p}}} \sum_{\lambda\in\Lambda} \sum_{\boldsymbol{q}\in\Omega} w^{s,\lambda}_{\boldsymbol{p},\boldsymbol{q}} I^\lambda_{\boldsymbol{q}}, \tag{1}$$

where $K^s_{\boldsymbol{p}}$ is a normalization factor, $\Lambda$ is a support subset of wavelength-dimension and $\lambda$ is the element, and $\Omega$ is a support subset of special-dimension and $\boldsymbol{q} = (x_{\boldsymbol{q}},y_{\boldsymbol{q}})$ is the element. $I^\lambda_{\boldsymbol{q}}$ is a support pixel in the hyperspectral cube. The weight $w^{s,\lambda}_{\boldsymbol{p},\boldsymbol{q}}$ is defined by multipliers of three weight, which are weights of domain distance similarity ($wd$), intensity similarity ($wi$) and spectral similarity ($ws$):

$$w^{s,\lambda}_{\boldsymbol{p},\boldsymbol{q}} = wd^{s,\lambda}_{\boldsymbol{p},\boldsymbol{q}} wi^{s,\lambda}_{\boldsymbol{p},\boldsymbol{q}} ws^{s}_{\boldsymbol{p},\boldsymbol{q}}. \tag{2}$$

The domain distance similarity weight is defined as:

$$wd^{s,\lambda}_{\boldsymbol{p},\boldsymbol{q}} = \exp(\frac{\|(x_{\boldsymbol{p}}, y_{\boldsymbol{p}}, \alpha s)^{\mathrm{T}} - (x_{\boldsymbol{q}}, y_{\boldsymbol{q}}, \alpha\lambda)^{\mathrm{T}}\|^2_2}{-2\sigma^2_d}), \tag{3}$$

where $\alpha$ is balancing parameter between the spacial distance of image and the wavelength distance. The intensity similarity weight is defined as:

$$wi^{s,\lambda}_{\boldsymbol{p},\boldsymbol{q}} = \exp(\frac{\|I^s_{\boldsymbol{p}} - I^\lambda_{\boldsymbol{q}}\|^2_2}{-2\sigma^2_i}). \tag{4}$$

The spectral similarity weight is defined by the Gaussian weighted total product of distance of each wavelength:

$$\begin{aligned} ws^s_{\boldsymbol{p},\boldsymbol{q}} &= \prod_{n=1}^{N} \mathrm{pow}(\exp(\frac{\|I^n_{\boldsymbol{p}} - I^n_{\boldsymbol{q}}\|^2_2}{-2\sigma^2_s}), \exp(\frac{\|n-s\|^2_2}{-2\sigma^2_{sg}})) \\ &= \exp(\frac{1}{-2\sigma^2_s}(w^{s,1}(I^1_{\boldsymbol{p}}{-}I^1_{\boldsymbol{q}})^2 + w^{s,2}(I^2_{\boldsymbol{p}}{-}I^2_{\boldsymbol{q}})^2 + \cdots + w^{s,N}(I^N_{\boldsymbol{p}}{-}I^N_{\boldsymbol{q}})^2), \\ w^{s,n} &= \exp(\frac{\|s-n\|^2_2}{-2\sigma^2_{sg}}), \end{aligned} \tag{5}$$

where $N$ is the maximum number of the spectral image.

Brute-force implementation of HGF has $O(lr^2)$-order, where $l$ is the number of elements in $\Lambda$ and $r$ is the kernel radius of $\Omega$. The computational is huge; thus, the efficient implementation is required.

### 2.2 Separable Acceleration

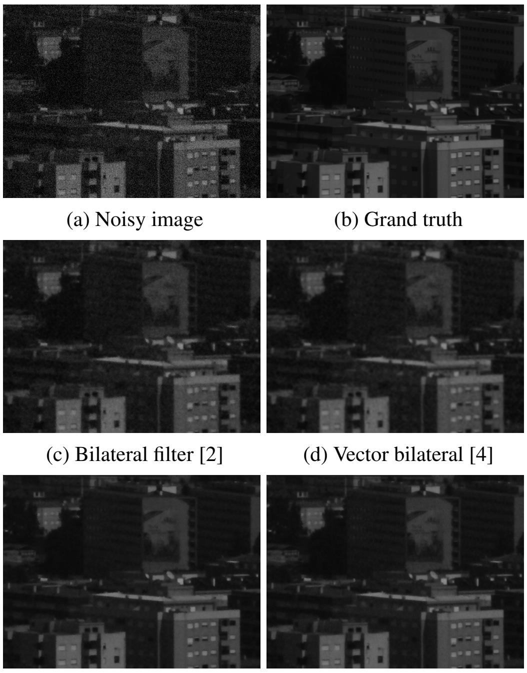
Accelerating HGF, we approximate the filter by forcefully separating the 3D kernel in the hyperspectral cube to 1D kernels. We call this filter *separable hyperspectral Gaussian filtering (SHGF)*. Algorithm 1 presents the overview of SHGF. The computational order of SHGF is reduced to $O(l + r)$.

With this algorithm, firstly, we filter the cube along the 1D horizontal dimension, and then perform the filtered cube along the 1D vertical dimension. Finally, we filter the double filtered cube along the 1D spectral dimension. Especially in the second and third filtering, weights should be computed from non-filtered spectral images [6], i.e., joint/cross filtering style; thus, these weights are independent of the sequence of separable filtering. Moreover, the weight computation technique can prevent the over-smoothing problem. Also, we control Gaussian distribution parameter of $\sigma$ to prevent the streaking noise problem in separable filtering. We narrow the distribution of Gaussian for intensity and spectral weights in the second and the third pass; thus, we use these parameters: $\beta\sigma_i, \beta\sigma_s, \gamma\sigma_i$ $(0 \leq \beta, \gamma \leq 1)$. Note that spectral parameters are shrunk in the spectral dimensional filtering, because each sample has the same spectrum in this dimension.

**Algorithm 1** Separable Hyperspectral Gaussian Filtering

**Input:** $\boldsymbol{I}, \sigma_d, \sigma_i, \sigma_s, \sigma_{sg}, \alpha, \beta, \gamma$
**Output:** $\bar{I}^s$

1: Horizontal 1D filtering for $\boldsymbol{I}$ with $(\sigma_d, \sigma_i, \sigma_s, \sigma_{sg}, \alpha)$. Output is $\bar{\boldsymbol{I}}_{\boldsymbol{h}}$.
2: Vertical 1D joint filtering for $\bar{\boldsymbol{I}}_{\boldsymbol{h}}$ (kernel is computed form $\boldsymbol{I}$) with $(\sigma_d, \beta\sigma_i, \beta\sigma_s, \sigma_{sg}, \alpha)$. Output is $\bar{\boldsymbol{I}}_{\boldsymbol{v}}$.
3: Spectral 1D joint filtering for $\bar{\boldsymbol{I}}_{\boldsymbol{v}}$ ((kernel is computed form $\boldsymbol{I}$) with $(\sigma_d, \gamma\sigma_i, \alpha)$.



(a) Noisy image (b) Grand truth

(c) Bilateral filter [2] (d) Vector bilateral [4]

(e) HGF (f) SHGF

Fig. 1: Denoising results.

## 3 Experimental Results

In our experiment, we use the hyperspectral natural image dataset [7]. Note that we use a hyperspectral image whose spectral dimension is 21. The wavelength of the images is from 450 nm to 650 nm at 10 nm intervals. We employ bilateral filtering [2] and vector bilateral filtering [4] as our compared methods. Note that we apply bilateral filtering to each spectral image captured about a wavelength, and vector bilateral filtering is performed without a noise level estimation for each spectral image, i.e., the most simplified implementation. For our HGF and SHGF, we set the 3D kernel to $7 \times 7 \times 7$.

Denoising results are shown in Fig. 1. We add Gaussian noises, where the standard deviation $\sigma$ is 20, to the noisy image. The noises remain in the bilateral filter and the vector bilateral filter results. By contrast, ours of the HGF and SHGF can remove the noises while preserving the texture.

The denoising performance is also presented in PSNR accuracy results (Tab. 1). Our methods have the best results as compared to the other state-of-the-art methods. Moreover, our separable implementation superior to the brute-force. The fact indicates that spectral domain information has an important role in hyperspectral denosing. We will investigate the relation as our future work.

Table 1: PSNR accuracy results [dB].

| Method | $\sigma = 10$ | $\sigma = 20$ | $\sigma = 30$ |
|---|---|---|---|
| Input | 28.32 | 22.89 | 19.85 |
| Bilateral filter [2] | 37.21 | 33.45 | 31.46 |
| Vector bilateral [4] | 38.98 | 34.90 | 31.75 |
| HGF | 39.17 | 36.25 | 33.16 |
| SHGF | **40.19** | **36.43** | **33.42** |

Table 2: Computational time results [sec].

| Method | time [sec] |
|---|---|
| Bilateral filter | 2.01 |
| HGF | 468 |
| SHGF | 6.84 |

Computational time results are shown in Tab. 2. Note that the input image resolution is $1338 \times 1021$, and the CPU is Intel Core i7-3770K 3.50 GHz. The computational cost of HGF is too expensive, while our separable implementation can accelerate $\times 70$ from the brute-force implementation.

## 4 Conclusion

We proposed hyperspectral Gaussian filtering (HGF) and also presented the separable acceleration for HGF. Our proposed methods have the best denoising performance among the compared methods. Moreover, our separable implementation allows us to compute efficiently ($\times 70$ speed-up) while keeping enough denoising performance. As our future work, we will investigate more sophisticated method through a noise level estimation.


### Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 15K16023.


### References


[1] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 29–43, 2002.

[2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *International Conference on Computer Vision*, 1998, pp. 839–846.

[3] E. P. Bennett, J. L. Mason, and L. McMillan, "Multispectral bilateral video fusion," *IEEE Trans. on Image Processing*, vol. 16, no. 5, pp. 1185–1194, 2007.

[4] H. Peng, R. Rao, and S. A. Dianat, "Multispectral image denoising with optimized vector bilateral filter," *IEEE Trans. on Image Processing*, vol. 23, no. 1, pp. 264–273, 2014.

[5] Q. Yang, N. Ahuja, and K.-H. Tan, "Constant time median and bilateral filtering," *International Journal of Computer Vision*, vol. 112, no. 3, pp. 307–318, 2014.

[6] N. Fukushima, S. Fujita, and Y. Ishibashi, "Switching dual kernels for separable edge-preserving filtering," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 1588–1592.

[7] D. H. Foster, K. Amano, S. Nascimento, and M. J. Foster, "Frequency of metamerism in natural scenes," *Journal of the Optical Society of America A*, vol. 23, no. 10, pp. 2359–2372, 2006.