TRILATERAL FILTER ON GRAPH SPECTRAL DOMAIN

Masaki Onuki and Yuichi Tanaka

Graduate School of BASE, Tokyo University of Agriculture and Technology
Koganei, Tokyo, 184-8588 Japan
Email: masaki.o.0806@msp.lab.tuat.ac.jp, ytnk@cc.tuat.ac.jp

ABSTRACT

This paper presents the trilateral filter (TF) in the perspective of graph signal processing. The TF is a single-pass nonlocal filter for edge-preserving smoothing. To smooth an image, it does not require many iterations compared to conventional smoothing methods, e.g., the bilateral filter. Additionally, one parameter is only required for filtering. Since the TF coefficients depend on original image data, it is not possible to provide a frequency domain representation using regular signal processing. To overcome this problem, we firstly show the TF as a vertex domain transform on a graph and then define it on graph spectral domain. In the experimental results, the proposed method presents better denoising performances than conventional methods.

Index Terms— Trilateral filter, bilateral filter, graph signal processing, spectral graph theory, denoising

1. INTRODUCTION

So far, many smoothing filters have been proposed [1]-[9]. The gaussian filter [1, 2] has been widely used and it removes details well. The bilateral filter (BF) was proposed to preserve the image edges [4, 5]. In image processing, the BF has been widely used in different applications such as denoising and edge-preserving multiscale decomposition [6, 7]. It usually requires many iterative operations to smooth images, which lead to high computational cost. To overcome this problem, the trilateral filter (TF) was proposed [8, 9]. It does not need many iterations in comparison with the BF, and users are only required to set one parameter.

The weight of the TF is determined from original image pixels. Hence, it is data-dependent image filtering. Normally, the data-dependent method is not possible to provide a frequency domain representation with regular signal processing.

The BF, which is also a data-dependent filtering, can be represented in graph spectral domain by using the approach from graph signal processing [12]. In this method, a vertex of a graph is set as a pixel, and the BF coefficients are used as the link weights of the graph. From the spectral graph theory, the BF on graph was defined as spectral graph filters, where the spectral response is calculated by using eigenvectors and eigenvalues of the graph Laplacian matrix [12]-[17]. Similar to the Fourier transform of regular signals, the eigenvalues and the eigenvectors capture the oscillatory behavior of the graph signal. By using the spectral domain representation, the BF was suggested as a low pass filter that can be characterized by a spectral response corresponding to a linear spectral decay. This representation enables us to use a flexible low pass filter defined on spectral domain and it presents significant performance improvements compared to the traditional BF.

In this paper, we propose the TF represented in graph spectral domain. The TF is generally composed of the gradient smoothing and the detail smoothing. These smoothing steps can be represented on graph vertex domain. Then, similar to the aforementioned BF, the TF is implemented on graph spectral domain. Specifically, the gradient and the detail smoothing filters are represented in spectral domain, and we redefine the TF using these spectral filters. Our method can also be regarded as *double* low pass filters in spectral domain and can be extended to multi-lateral filter. In experiment, the proposed method indicates better denoising performances than the conventional BF and TF.

This paper is organized as follows. Section 2 briefly reviews the TF. Section 3 describes definitions of the graph spectral filters. The main contribution of this paper is presented in Section 4, which describes how the TF is constructed on vertex domain and spectral domain. We verified our method through image denoising, which is discussed in Section 5. Finally, Section 6 concludes the paper.

Notations: Upper case bold-face letters and calligraphic capital letters indicate matrices and sets, respectively. A subscript of a matrix represents its size if there is no explicit expression. Superscript $\cdot^T$ is the transpose of the matrix. Gaussian functions with variance σ_c and σ_s are denoted as $c(\cdot)$ and $s(\cdot)$, respectively.

2. TRILATERAL FILTER

The TF presents a strong noise reduction performance in high-gradient regions [8, 9]. It requires only one user-set parameter, and does not require many iterations to obtain a smoothed image.

Let $\mathbf{I}$, $\mathbf{p}$, and $\mathbf{w}$ denote the input signal, the signal position vector, and the offset vector, respectively. First, the TF smoothes image gradient with the BF:

$$\mathbf{G}(\mathbf{p}) = \frac{1}{k}\int_{-\infty}^{\infty} \Delta\mathbf{I}(\mathbf{p}+\mathbf{w})c(\mathbf{w})s(\|\Delta\mathbf{I}(\mathbf{p}+\mathbf{w}) - \Delta\mathbf{I}(\mathbf{p})\|)d\mathbf{w}, \tag{1}$$

where $\Delta\mathbf{I}$ is the gradient of the input signal and k is the normalization term calculated as

$$k = \int_{-\infty}^{\infty} c(\mathbf{w})s(\|\Delta\mathbf{I}(\mathbf{p}+\mathbf{w}) - \Delta\mathbf{I}(\mathbf{p})\|)d\mathbf{w}.$$

For the subsequent second BF, the TF uses the smoothed gradient $\mathbf{G}(\mathbf{p})$ for estimating an approximating plane as follows:

$$\mathbf{T}(\mathbf{p},\mathbf{w}) = \mathbf{I}(\mathbf{p}) + \mathbf{w}\mathbf{G}(\mathbf{p}). \tag{2}$$

This work was supported in part by JSPS KAKENHI Grant Number 24760288 and MEXT Tenure-Track Promotion Program.

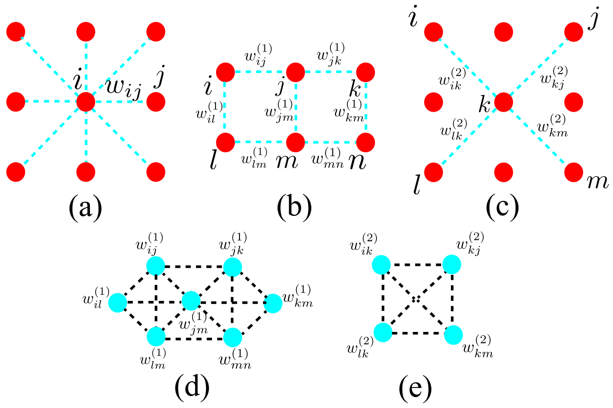

Fig. 1. Pixel intensity and gradient graphs. Red and blue vertices indecates pixels and gradients, respectively. (a) Eight-neighborhoods graph. (b) The vertically and horizontally connected graph. (c) The diagonally connected graph. (d) The gradient graph of (b). (e) The gradient graph of (c).

Let $\mathbf{I}_\triangle = \mathbf{I}(\mathbf{p}+\mathbf{w}) - \mathbf{T}(\mathbf{p},\mathbf{w})$. The TF output is calculated as follows:

$$\hat{\mathbf{I}}(\mathbf{p}) = \mathbf{I}(\mathbf{p}) + \frac{1}{k_\triangle}\int_{-\infty}^{\infty} \mathbf{I}_\triangle c(\mathbf{w}) s(\mathbf{I}_\triangle) f(\mathbf{p},\mathbf{w}) d\mathbf{w}, \tag{3}$$

where $k_\triangle$ is the normalization term calculated as

$$k_\triangle = \int_{-\infty}^{\infty} c(\mathbf{w}) s(\mathbf{I}_\triangle) d\mathbf{w}.$$

Additionally, $f(\mathbf{p},\mathbf{w})$ is a neighborhood function represented as

$$f(\mathbf{p},\mathbf{w}) = \begin{cases} 1 & \text{if } \|\mathbf{G}(\mathbf{p}+\mathbf{w}) - \mathbf{G}(\mathbf{p})\|_1 < R \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where R specifies the adaptive region and is discussed below.

The TF only requires a specification of the parameter σ_c for $c(\cdot)$, while σ_s of $s(\cdot)$ is self-adjusted. Let $\mathbf{g}$ be a gradient magnitude of $\mathbf{I}$. Then, σ_s is determined as follows:

$$\sigma_s = \beta \left\| \max(\mathbf{g}(\mathbf{p})) - \min(\mathbf{g}(\mathbf{p})) \right\|_1, \tag{5}$$

where $\beta = 0.15$ is recommended in [8] and $R = \sigma_s$.

3. GRAPH SPECTRAL FILTERS

Here, we consider an undirected graph $G = \{\mathcal{V}, E\}$ where vertices $\mathcal{V} = \{1, 2, \ldots, n\}$ are the pixels of the input image and the edges $E = \{(i, j, w_{ij})\}$ capture the similarity between two pixels, as shown in Fig. 1(a). Adjacency and diagonal degree matrices are defined as $\mathbf{W} = [w_{ij}]_{n \times n}$ and $\mathbf{D}_{jj} = \sum_i w_{ij}$, respectively.

The spectrum of a graph is defined by eigenvalues and eigenvectors of its Laplacian matrix [13]-[17]. The Laplacian matrix of the graph is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and its normalized form is $\boldsymbol{\mathcal{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. Since the normalized Laplacian matrix $\boldsymbol{\mathcal{L}}$ is a positive semi-definite matrix, the matrix $\boldsymbol{\mathcal{L}}$ is decomposed into an orthogonal set of eigenvectors $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ and eigenvalues $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ represented as follows:

$$\boldsymbol{\mathcal{L}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T. \tag{6}$$

In the graph setting, the graph Laplacian eigenvalues and eigenvectors provide a similar notion of frequency. The eigenvalues can be treated as graph frequencies, and are always situated in the internal $[0, 2]$ for $\boldsymbol{\mathcal{L}}$. The *graph Fourier transform* (GFT) of $\mathbf{x}$ is defined as

$$\widetilde{\mathbf{x}} = \mathbf{U}^T\mathbf{x}. \tag{7}$$

Clearly, the inverse GFT is given by $\mathbf{x} = \mathbf{U}\widetilde{\mathbf{x}}$.

Similar to regular signal processing, a graph spectral filtering can be defined as follows:

$$\widetilde{x}_{out}(\lambda_l) = h(\lambda_l)\widetilde{x}_{in}(\lambda_l). \tag{8}$$

By using the GFT and the inverse GFT, a graph spectral filtering with the matrix notation is represented as

$$\begin{aligned} \mathbf{x}_{out} &= \mathbf{U}h(\boldsymbol{\Lambda})\mathbf{U}^T\mathbf{x}_{in} \\ &= h(\boldsymbol{\mathcal{L}})\mathbf{x}_{in}, \end{aligned} \tag{9}$$

where $h(\boldsymbol{\Lambda})$ is the spectral response of a graph filter represented as

$$h(\boldsymbol{\Lambda}) = \begin{bmatrix} h(\lambda_1) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & h(\lambda_n) \end{bmatrix}. \tag{10}$$

4. TRILATERAL FILTER ON GRAPH

As mentioned in Section 2, the TF firstly smooths gradient, and then the input image is smoothed with this smoothed gradient. In this section, the TF is viewed as a vertex domain transform on a graph and it is further represented in graph spectral domain.

4.1. Vertex-Domain TF

To smooth gradient of the input image, the edge weight w_{ij} is initialized by gradient as follows:

$$w_{ij} = \begin{cases} x(j) - x(i) & \text{if edge of the graph exists} \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where $x(\cdot)$ is the input pixel value. Let $w_{ij}^{(1)}$ and $w_{ij}^{(2)}$ be the edge weights for the vertically and horizontally connected graph (Fig. 1(b)) and the diagonally connected graph (Fig. 1(c)), respectively. Then, edge weights calculated by (11) are set to vertices and the graph of the gradient is constructed such as Fig. 1(d) and (e). We consider these two gradient graphs in this paper.

For each gradient graph, the edge weights are calculated similar to the BF. They are represented as follows:

$$\widehat{w}_{ij}^{(t)} = \begin{cases} \exp\left(-\frac{\|\widehat{p}_j - \widehat{p}_i\|^2}{2\sigma_c^2}\right) \exp\left(-\frac{(w^{(t)}(j) - w^{(t)}(i))^2}{2\sigma_s^2}\right) \\ \qquad\qquad \text{if edge exists} \\ 0 \qquad\qquad \text{otherwise,} \end{cases} \tag{12}$$

where $\widehat{p}_i$ is the coordinate of the ith gradient and $t = 1, 2$. Let $\widehat{\mathbf{W}}^{(t)}$ and $\widehat{\mathbf{D}}^{(t)}$ be the adjacency matrix and the diagonal degree matrix of the gradient graph given by $\widehat{\mathbf{W}}^{(t)} = [\widehat{w}_{ij}^{(t)}]$ and $\widehat{\mathbf{D}}_{jj}^{(t)} = \sum_i \widehat{w}_{ij}^{(t)}$. The gradient $\mathbf{w}^{(t)}$ is bilaterally smoothed as follows:

$$\widetilde{\mathbf{w}}^{(t)} = (\widehat{\mathbf{D}}^{(t)})^{-1}\widehat{\mathbf{W}}^{(t)}\mathbf{w}^{(t)}, \tag{13}$$

where $\widetilde{\mathbf{w}}^{(t)} = [\widetilde{w}_{ij}^{(t)}]$ is the smoothed gradient for each direction. Note that (1) is equivalent to (13) for $t = 1$. However, the diagonal

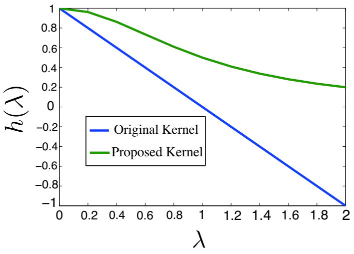

Fig. 2. Spectral responses. The blue line is the original kernel represented as (17) and (18). The green line is the proposed for $h_{reg}(\lambda) = \lambda$.

direction, which is the case of $t = 2$, is not considered in the original TF and is newly introduced in this paper.

The edge weight of the original graph $\mathbf{w}$ is updated with the smoothed gradients $\widetilde{\mathbf{w}}^{(t)}$ in each direction. Let the smoothed edge weight be denoted as $\mathbf{w}' = [w'_{ij}]$. Next, to smooth the image, the edge weights of the original image is updated as

$$\overline{w}_{ij} = \begin{cases} \exp\left(-\frac{\|\bar{p}_j - \bar{p}_i\|^2}{2\sigma_c^2}\right) \exp\left(-\frac{(w'(j) - w'(i))^2}{2\sigma_s^2}\right) & \text{if } w'_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{14}$$

where $\bar{p}_i$ is the coordinate of the ith pixel. Finally, the output of the vertex-domain TF is represented as

$$\bar{\mathbf{x}} = \overline{\mathbf{D}}^{-1}\overline{\mathbf{W}}\mathbf{x}, \tag{15}$$

where $\overline{\mathbf{W}} = [\,\overline{w}_{ij}]$ and $\overline{\mathbf{D}}_{jj} = \sum_j \overline{w}_{ij}$. The output of the TF in (3) corresponds to (15).

4.2. Spectral Representation of TF

Here, the vertex-domain filtering in (15) is rewritten as

$$\begin{aligned} \bar{\mathbf{x}} &= \overline{\mathbf{D}}^{-1}\overline{\mathbf{W}}\mathbf{x} \\ &= \overline{\mathbf{D}}^{-1/2}\overline{\mathbf{D}}^{-1/2}\overline{\mathbf{W}}\overline{\mathbf{D}}^{-1/2}\overline{\mathbf{D}}^{1/2}\mathbf{x} \\ &= \overline{\mathbf{D}}^{-1/2}(\mathbf{I} - \overline{\mathcal{L}})\overline{\mathbf{D}}^{1/2}\mathbf{x}, \end{aligned} \tag{16}$$

similar to that of [12]. With the normalized input $\mathbf{x}_{nor} = \overline{\mathbf{D}}^{1/2}\mathbf{x}$ and the normalized output $\bar{\mathbf{x}}_{nor} = \overline{\mathbf{D}}^{1/2}\bar{\mathbf{x}}$, (16) becomes

$$\begin{aligned} \bar{\mathbf{x}}_{nor} &= (\mathbf{I} - \overline{\mathcal{L}})\mathbf{x}_{nor} \\ &= \overline{\mathbf{U}}(\mathbf{I} - \overline{\mathbf{\Lambda}})\overline{\mathbf{U}}^T\mathbf{x}_{nor}. \end{aligned} \tag{17}$$

Since the spectral response is $h(\overline{\mathbf{\Lambda}}) = \mathbf{I} - \overline{\mathbf{\Lambda}}$, (17) is considered as the spectral filtering. This spectral response corresponds to the linear decay shown in the blue line of Fig. 2. It tries to preserve the low frequency components and attenuate the high frequency component. Hence, $h(\overline{\mathbf{\Lambda}})$ acts as a low pass filter.

Similarly, (13) expressed as the gradient smoothing can be represented similar to (17) as

$$\widetilde{\mathbf{w}}_{nor}^{(t)} = \widehat{\mathbf{U}}(\mathbf{I} - \widehat{\mathbf{\Lambda}})\widehat{\mathbf{U}}^T\mathbf{w}_{nor}^{(t)}, \tag{18}$$

where $\mathbf{w}_{nor}^{(t)} = \widehat{\mathbf{D}}^{1/2}\mathbf{w}^{(t)}$ and $\widetilde{\mathbf{w}}_{nor}^{(t)} = \widehat{\mathbf{D}}^{1/2}\widetilde{\mathbf{w}}^{(t)}$. Therefore, the gradient smoothing is also considered as the low pass spectral filtering.

Let $\mathbf{w}_1$ be the first-order derivative of the input image. The gradient smoothing in (18) can be rewritten as follows:

$$\mathbf{h}_1(\mathbf{x}) = \widehat{\mathbf{U}}h(\widehat{\mathbf{\Lambda}})\widehat{\mathbf{U}}^T\mathbf{w}_1. \tag{19}$$

Similarly, the detail smoothing in (17) can be rewritten with the spectral response $h(\overline{\mathbf{\Lambda}})$ as

$$\mathbf{h}_0(\mathbf{h}_1, \mathbf{x}) = \bar{\mathbf{x}} = \overline{\mathbf{U}}h(\overline{\mathbf{\Lambda}})\overline{\mathbf{U}}^T\mathbf{x}, \tag{20}$$

where $\overline{\mathbf{U}}$ and $\overline{\mathbf{\Lambda}}$ are derived from $\mathbf{h}_1$. From (20), the TF can be regarded as double (recursive) low pass filters.

4.3. Multi-lateral Filter

The TF firstly smooths the first-order derivative of the input image, and then smooths the image with the smoothed derivative. Here, we consider the multi-lateral filter as a generalized BF and TF.

Let $\mathbf{w}_m$ be the mth-order derivative of the image. The derivative is smoothed as

$$\mathbf{h}_m(\mathbf{x}_{in}) = \widehat{\mathbf{D}}_m^{-1}\widehat{\mathbf{W}}_m\mathbf{w}_m, \tag{21}$$

where $\widehat{\mathbf{W}}_m$ and $\widehat{\mathbf{D}}_m$ be the adjacency matrix and the diagonal degree matrix given by $\widehat{\mathbf{W}}_m = [(\widehat{w}_m)_{ij}]$ and $(\widehat{\mathbf{D}}_m)_{jj} = \sum_i (\widehat{w}_m)_{ij}$, respectively. The mth-order derivative of the image $\mathbf{w}_m$ is derived from $\mathbf{x}_{in}$, and then $\widehat{\mathbf{W}}_m$ is derived from $\mathbf{w}_m$. Similarly, $(m-1)$th-order smoothed derivative is represented as

$$\mathbf{h}_{m-1}(\mathbf{h}_m, \mathbf{x}_{in}) = \widehat{\mathbf{D}}_{m-1}^{-1}\widehat{\mathbf{W}}_{m-1}\mathbf{w}_{m-1}, \tag{22}$$

where the matrix $\widehat{\mathbf{W}}_{m-1}$ is derived from $\mathbf{h}_m$, i.e., the smoothed mth-order derivative of the image and the $(m-1)$th-order derivative of the image $\mathbf{w}_{m-1}$ is derived from $\mathbf{x}_{in}$. Therefore, the multi-lateral filter is represented as follows:

$$\mathbf{x}_{out} = \mathbf{h}_0(\mathbf{h}_1(\ldots\mathbf{h}_{m-1}(\mathbf{h}_m(\mathbf{x}_{in}), \mathbf{x}_{in})\ldots), \mathbf{x}_{in}), \mathbf{x}_{in}). \tag{23}$$

5. EXPERIMENT

5.1. Spectral Kernel Design for Denoising

Let the noise signal be $\mathbf{e}$. An observed ith pixel value of noisy image is represented as

$$\mathbf{y}(i) = \mathbf{x}(i) + \mathbf{e}(i). \tag{24}$$

With the regularization operator $h_{reg}(\overline{\mathcal{L}})$, the denoising problem can be represented as [18, 19]

$$\mathbf{x}_{reg} = \arg\min\{\|\mathbf{x} - \mathbf{y}\|_2^2 + \rho\|h_{reg}(\overline{\mathcal{L}})\mathbf{x}\|_2^2\}, \tag{25}$$

where $\rho > 0$ is a regularization parameter. $\|\mathbf{x} - \mathbf{y}\|_2^2$ in (25) is a distance measure between the noisy image and the original image. The term $\|h_{reg}(\overline{\mathcal{L}})\mathbf{x}\|_2^2$, on the other hand, indicates the regularization term. Because it may lead to oversmoothing without the regularization term, it is the ill-posed problem. When $h_{reg}(\overline{\mathcal{L}})$ is considered as an arbitrary high pass filter, the high frequency component is added to $\|\mathbf{x} - \mathbf{y}\|_2^2$. The objective function is represented as

$$\mathbf{K} = \|\mathbf{x} - \mathbf{y}\|_2^2 + \rho\|h_{reg}(\overline{\mathcal{L}})\mathbf{x}\|_2^2. \tag{26}$$

Setting the partial derivative of $\mathbf{K}$ to zero, we obtain

$$\frac{\partial\mathbf{K}}{\partial\mathbf{x}} = 2(\mathbf{x} - \mathbf{y}) + 2\rho h_{reg}^2(\overline{\mathcal{L}})\mathbf{x} = \mathbf{0}. \tag{27}$$

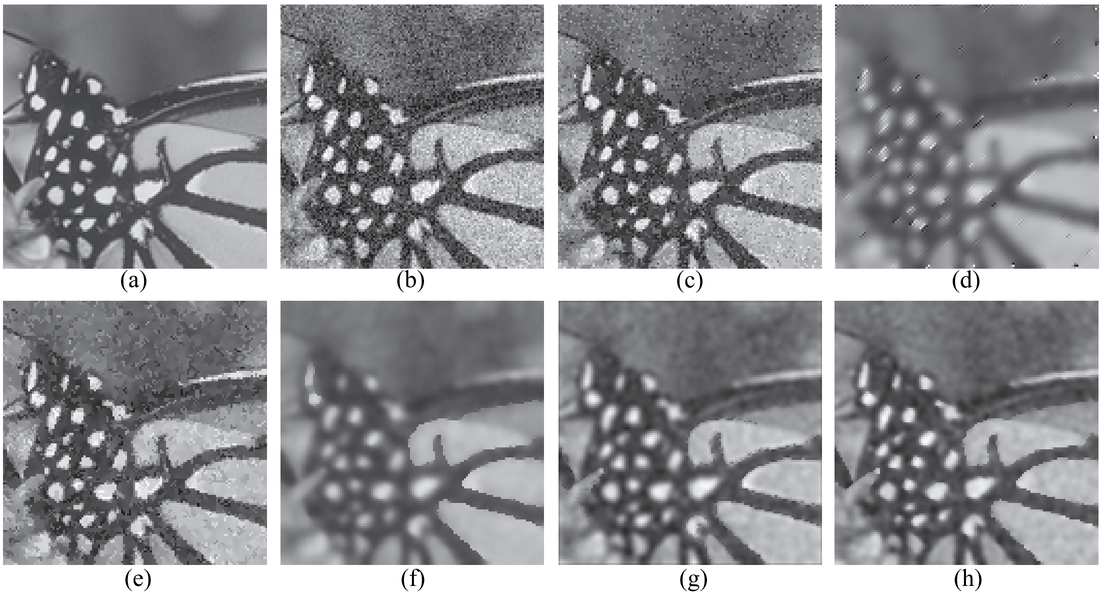

Fig. 3. Denoised results. (a) Original image. (b) Noisy image ($\sigma = 25$), PSNR=20.27 dB. (c) BF. (d) TF. (e) Vertex-domain BF. (f) Vertex-domain TF. (g) Spectral-domain BF. (h) Spectral-domain TF.

Table 1. The denoising results. These are indicated with PSNR (dB).

Images	BF	TF	Vertex-domain BF	Vertex-domain TF	Spectral-domain BF	Spectral-domain TF
Lena	21.60	22.23	23.01	25.33	26.18	**27.01**
Watch	21.87	23.32	23.55	25.11	25.85	**26.14**
Boat	21.94	21.89	23.46	24.48	25.50	**26.09**
Monarch	21.85	19.67	22.76	21.60	23.84	**25.15**

Then, $\mathbf{x}_{reg}$ can be expressed as

$$\begin{aligned}\mathbf{x}_{reg} &= (\mathbf{I} + \rho h_{reg}^2(\overline{\boldsymbol{\mathcal{L}}}))^{-1}\mathbf{y} \\ &= \overline{\mathbf{U}}(\mathbf{I} + \rho h_{reg}^2(\overline{\boldsymbol{\Lambda}}))^{-1}\overline{\mathbf{U}}^T\mathbf{y}. \end{aligned} \quad (28)$$

Since $h_{reg}(\overline{\boldsymbol{\mathcal{L}}})$ is designed as the high pass filter, $(\mathbf{I} + \rho h_{reg}^2(\overline{\boldsymbol{\mathcal{L}}}))^{-1}$ indicates the low pass filter. Our method uses this low pass filter instead of $(\mathbf{I} - \overline{\boldsymbol{\Lambda}})$ in (17), i.e., $h(\overline{\boldsymbol{\Lambda}}) = (\mathbf{I} + \bar{\rho} h_{reg}^2(\overline{\boldsymbol{\Lambda}}))^{-1}$. Similarly, (18) is replaced with $h(\widehat{\boldsymbol{\Lambda}}) = (\mathbf{I} + \widehat{\rho} h_{reg}^2(\widehat{\boldsymbol{\Lambda}}))^{-1}$. Since the kernel $h(\lambda)$ is generally not a polynomial of λ, this kernel is approximated with Chebychev polynomials [13].

5.2. Experimental Results

Four eight-bit grayscale images, *Lena*, *Watch*, *Boat*, and *Monarch* are used for the experiments. These images are 128×128 pixels and partial images of the original ones.

We compare the image denoising performances with the BF [4]-[7], the TF [8, 9], the vertex-domain BF [10, 11], the vertex-domain TF, the spectral-domain BF [12], and the spectral-domain TF. The vertex-domain BF and TF use the aforementioned linear decay kernel, and those for the spectral-domain BF and TF were changed to the proposed kernel explained in Section 5.1 where $h_{reg}(\lambda) = \lambda$. This spectral response is shown in the green line of Fig. 2. White gaussian noise with $\sigma = 25$ are added to the images. In this paper, the preprocessing step in edge detection is used for constructing the graphs of the vertex-domain and spectral-domain methods, so that the connection between vertices are disconnected if the edge of the graph connects two distinct image regions.

The denoising results with five iterations are shown in Table 1. Our method, i.e., the spectral-domain TF, shows better performances than the BF and the TF. By using the spectral filtering, the performance of the TF is significantly improved. Especially, in the result of *Monarch*, the spectral-domain TF outperformed the vertex-domain one by more than 3.5 dB.

The filtered *Monarch* images are shown in Fig. 3. The BF and TF in pixel-domain presents worse performances than the graph-based methods. The vertex-domain TF smoothes the noisy image while edge is preserved. However, the details are considerably blurred, which is observed in Fig. 3(f). It is clear that the spectral-domain BF and TF present good smoothing performances with edge preservation properties. Furthermore, the spectral-domain TF retains clear details compared to the spectral-domain BF.

6. CONCLUSION

In this paper, the trilateral filter was represented on vertex and spectral graph domains. The trilateral filter can be regarded as the double low pass filters. Furthermore, the multi-lateral filter was introduced. Our trilateral filter obtained better PSNRs than the conventional methods as well as the significant improvements of the visual qualities.

7. REFERENCES

[1] R. Szeliski *Computer Vision Algorithms and Applications*, USA: Springer, 2010.

[2] R. C. Gonzalez and R. E. Woods *Digital Image processing. 3 edn*, USA: Prentice Hall, 2007.

[3] K. He, J. Sun, and X. Tang "Guided image filtering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, issue. 6, pp. 1397-1409, Oct., 2012.

[4] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in Proc. *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 839-846, Bombay, India, Jan., 1998.

[5] S. Paris, P. Jirboribst, J. Tumblin, and F. Durand, "Bilateral filtering: Theory and applications," *Foundations and Trends in Computer Graphics and Vision*, vol. 4, no. 12, pp. 2324-2333, Dec., 2009.

[6] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic range images," *ACM Trans. on Graphics*, vol. 21, issue. 3, pp. 257-266, Jul., 2002.

[7] S. Fleishman, I. Drori and D. Cohen-Or, "Bilateral mesh denoising," *ACM Trans. on Graphics*, vol. 22, issue. 3, pp. 950-953, Jul., 2003.

[8] P. Choudhury and J. Tumblin, "The trilateral filter for high contrast images and meshes," *Eurographics Symp. on Rendering*, pp. 1-11, 2003.

[9] W. C. K. Wong and A. C. S. Chung, "Trilateral filtering for biomedical images," In Proc. *IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro (ISBI)*, pp. 820-823, Arlington, VA, USA, Apr., 2004.

[10] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106-128, Jan., 2013.

[11] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: A Framework for Image and Manifold Processing," *IEEE Trans. on Image Processing*, vol. 17, no. 7, pp. 1047-1060, Jul., 2008.

[12] A. Gadde, S. K. Narang, and A. Ortega, "Bilateral filter: Graph spectral interpretation and extensions," in Proc. *IEEE Int. Conf. on Image Processing (ICIP)*, pp. 1222-1226, Sep., 2013.

[13] D. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129-150, 2011.

[14] S.K. Narang, A. Ortega, "Perfect reconstruction two-channnel wavelet filter banks for graph structured data," *IEEE Trans. on Signal Processing*, vol. 60, no. 6, pp. 2786-2799, Jun., 2012.

[15] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "Signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular data domains," *IEEE Signal Processing Magazine*, vol. 30, issue. 3, pp. 83-98, May, 2013.

[16] N. Leonardi and D. V. D. Ville, "Tight wavelet frames on multislice graphs," *IEEE Trans. on Signal Processing*, vol. 61, no. 13, pp. 3357-3367, Jul., 2013.

[17] F. R. K. Chung, *Spectral graph theory*, vol. 92, CBMS Regional Conference Series in Mathematics, AMS, 1997.

[18] S. Mallat, *A Wavelet tour of signal processing The Sparse Way*, Burlington: Academic, 2009.

[19] Y. Yang, and N. P. Galatsanos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images", *IEEE Trans. circuits syst*, vol. 3, no. 6, pp. 421-432, Dec., 1993.