BILATERAL FILTER: GRAPH SPECTRAL INTERPRETATION AND EXTENSIONS

Akshay Gadde, Sunil K Narang and Antonio Ortega

Ming Hsieh Department of Electrical Engineering
University of Southern California
agadde@usc.edu, kumarsun@usc.edu, ortega@sipi.usc.edu

ABSTRACT

In this paper we study the bilateral filter proposed by Tomasi and Manduchi and show that it can be viewed as a spectral domain transform defined on a weighted graph. The nodes of this graph represent the pixels in the image and a graph signal defined on the nodes represents the intensity values. Edge weights in the graph correspond to the bilateral filter coefficients and hence are data adaptive. The graph spectrum is defined in terms of the eigenvalues and eigenvectors of the graph Laplacian matrix. We use this spectral interpretation to generalize the bilateral filter and propose new spectral designs of "bilateral-like" filters. We show that these spectral filters can be implemented with k-iterative bilateral filtering operations and do not require expensive diagonalization of the Laplacian matrix.

Index Terms— Bilateral filter, graph based signal processing, polynomial approximation

1. INTRODUCTION

The bilateral filter (BF) proposed by Tomasi and Manduchi [1] has emerged as a powerful tool for image processing. Bilateral filtering smooths images while preserving edges, by taking the weighted average of the nearby pixels. The weights depend on both the spatial distance and photometric distance which provides local adaptivity to the given data. The bilateral filter and its variants are widely used in different applications such as denoising, edge preserving multiscale decomposition, detail enhancement or reduction and segmentation etc. [2–6]. Bilateral filtering was initially developed as an intuitive tool without theoretical justification. Since then, connections between the BF and other well known filtering frameworks such as anisotropic diffusion, weighted least squares, Bayesian methods, kernel regression and non-local means have been explored [7–11].

BF coefficients depend on original pixel values, so that BF is data dependent, leading to a non-linear and non-shift invariant filter. Thus, it is not possible to provide a frequency domain interpretation of BF using traditional image-domain Fourier techniques. Note that a spectral interpretation would be desirable, as it may lead to improved designs by taking into account known spectral characteristics of the input signal. To overcome this difficulty, we view BF as a vertex domain transform on a graph with pixels as vertices, intensity values of each node as the graph signal and filter coefficients as link weights that capture the similarity between nodes. While this graph-based perspective of the BF has been adopted by several authors [6, 9, 12–14], to the best of our knowledge *graph spectral domain filter design extensions of the BF have not been studied to date.*

This work was supported in part by NSF under grant CCF-1018977.

We can define spectral filters on these graphs, where the spectral response is calculated in terms of eigenvectors and eigenvalues of the graph Laplacian matrix [15–18]. This spectral interpretation captures the oscillatory behavior of the graph signal [19] and thus allows us to extend of the concept of frequency to irregular domains. This has led to the design of frequency selective filtering operations on graphs, similar to those in traditional signal processing. These graph spectral filters also have a vertex-domain implementation, i.e., they do not necessarily require to first transform the input signal into the spectral domain.

In this paper, we interpret the BF as a 1-hop localized transform on the aforementioned graph. Under this framework, we show that the BF is a lowpass transform that can be characterized by a spectral response corresponding to a linear spectral decay. We also calculate the spectral response of iterated BF. We extend this novel insight to build more general bilateral-like filters using the machinery to design graph based transforms with a desired spectral response. Our design allows the flexibility to choose the spectral response of the filter depending on the application. We also give a theoretical justification for the design using a framework of regularization on graphs. We show that these spectral filters do not require computationally expensive diagonalization of the graph Laplacian matrix and then provide an efficient algorithm for implementing these filters using the BF as a building block. We examine the performance of the proposed filters in two example applications.

2. BILATERAL FILTER AS A GRAPH BASED TRANSFORM

We first explain BF as originally proposed in [1]. Consider an input image $\mathbf{x}_{in}$ to the BF. The value at each position in the output image $\mathbf{x}_{out}$ is given by the weighted average of the pixels in $\mathbf{x}_{in}$:

$$\mathbf{x}_{out}[j] = \sum_i \frac{w_{ij}}{\sum_i w_{ij}} \mathbf{x}_{in}[i] \tag{1}$$

The weights w_{ij} depend on both the euclidean and photometric distance between the pixels $\mathbf{x}_{in}[i]$ and $\mathbf{x}_{in}[j]$[1]. Let p_i denote the position of the pixel i. The weights are then given by

$$w_{ij} = \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_d^2}\right) . \exp\left(-\frac{(\mathbf{x}_{in}[i] - \mathbf{x}_{in}[j])^2}{2\sigma_r^2}\right), \tag{2}$$

where σ_d and σ_r are the filter parameters. Spatial Gaussian weighting decreases the influence of distant pixels and intensity Gaussian weighting decreases the influence of pixels with different intensities. The intuition is that only similar nearby pixels should get averaged

[1]For color images, photometric distance can be computed in the CIE-Lab color space as suggested in [1]

so that blurring of edges is avoided. Note that (1) shows that the averaging is done over all pixels. However in practice, one assumes non-zero weights only for the pixels which have $\|p_i - p_j\| \leq 2\sigma_s$ [2].

Now, consider an undirected graph $G = (\mathcal{V}, E)$ where the nodes $\mathcal{V} = \{1, 2, \ldots, n\}$ are the pixels of the input image and the edges $E = \{(i, j, w_{ij})\}$ capture the similarity between two pixels as given by the BF weights (Figure 1). Image $\mathbf{x}_{in}$ can be considered as a signal defined on this graph $\mathbf{x}_{in} : \mathcal{V} \to \mathbb{R}$ where the signal value at each node equals the corresponding pixel intensity. Adjacency matrix $\mathbf{W}$ of this graph is given by $\mathbf{W} = [w_{ij}]_{n \times n}$. Let $\mathbf{D}$ be the diagonal degree matrix where each diagonal element $\mathbf{D}_{jj} = \sum_i w_{ij}$. With this notation the filtering operation in (1) can be written as [9]

$$\mathbf{x}_{out} = \mathbf{D}^{-1}\mathbf{W}\mathbf{x}_{in} \tag{3}$$

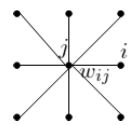

Fig. 1: The BF graph

It can be seen from (1) that the output at each node in the graph depends only on the nodes in its 1-hop neighborhood. Thus, the BF is a 1-hop localized graph based transform. Also, note that the BF includes the current pixel in the weighted average. Thus, the graph corresponding to the BF has a *self loop*, i.e., an edge connecting each node to itself with weight 1. One can consider BF operation on a graph *without self loops*, in which the center pixel intensity is not included in the weighted average. Other filtering techniques such as Gaussian smoothing and non-local means can also be described using similar graph-based models [9, 12].

2.1. Spectrum of a Graph

The spectrum of a graph is defined in terms of the eigenvalues and eigenvectors of its Laplacian matrix. The combinatorial Laplacian matrix for the graph G is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. We use the normalized form of the Laplacian matrix given as $\boldsymbol{\mathcal{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. $\boldsymbol{\mathcal{L}}$ is a non-negative definite matrix [20]. As a result $\boldsymbol{\mathcal{L}}$ has an orthogonal set of eigenvectors $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_2\}$ with corresponding eigenvalues $\sigma(G) = \{\lambda_1, \ldots, \lambda_n\}$ and can be diagonalized as

$$\boldsymbol{\mathcal{L}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^t \tag{4}$$

where $\boldsymbol{\Lambda} = diag\{\lambda_1, \ldots, \lambda_n\}$.

Similar to classical Fourier transform, the eigenvectors and eigenvalues of the Laplacian matrix $\boldsymbol{\mathcal{L}}$ provide a spectral interpretation of the graph signals. The eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ can be treated as graph frequencies, and are always situated in the interval $[0, 2]$ on the real line. The eigenvectors of the Laplacian matrix demonstrate increasing oscillatory behavior as the magnitude of the graph frequency increases [19]. The *Graph Fourier Transform* (GFT) of a signal $\mathbf{x}$ is defined as its projection onto the eigenvectors of the graph, i.e., $\tilde{x}(\lambda_i) = \langle \mathbf{x},\ \mathbf{u}_i \rangle$, or in matrix form $\tilde{\mathbf{x}} = \mathbf{U}^t\mathbf{x}$. The inverse GFT is given by $\mathbf{x} = \mathbf{U}\tilde{\mathbf{x}}$.

3. SPECTRAL INTERPRETATION OF THE BILATERAL FILTER

Similar to conventional signal processing, *graph spectral filtering* is defined as

$$\tilde{x}_{out}(\lambda_i) = h(\lambda_i)\tilde{x}_{in}(\lambda_i) \tag{5}$$

$h(\lambda_i)$ is the spectral response of the filter according to which spectral components of an input signal are modulated. Using the definition of GFT and the diagonalized form of $\boldsymbol{\mathcal{L}}$, we can write graph spectral filtering in matrix notation as [17]

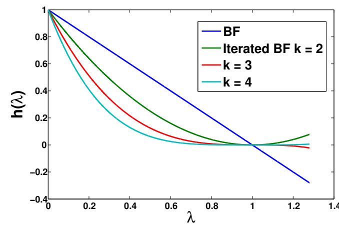

Fig. 2: Spectral responses of the BF and iterated BF.

$$\mathbf{x}_{out} = \underbrace{\mathbf{U}}_{\substack{\text{Inverse}\\ \text{GFT}}}\underbrace{h(\boldsymbol{\Lambda})}_{\substack{\text{Spectral}\\ \text{response}}}\underbrace{\mathbf{U}^t\mathbf{x}_{in}}_{\text{GFT}} = h(\boldsymbol{\mathcal{L}})\mathbf{x}_{in}. \tag{6}$$

To exploit this framework of graph spectral filtering we rewrite the BF in (3) as:

$$\begin{aligned} \mathbf{x}_{out} &= \mathbf{D}^{-1/2}\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}\mathbf{D}^{1/2}\mathbf{x}_{in} \\ \Rightarrow \mathbf{D}^{1/2}\mathbf{x}_{out} &= (\mathbf{I} - \boldsymbol{\mathcal{L}})\mathbf{D}^{1/2}\mathbf{x}_{in} \end{aligned} \tag{7}$$

From this equation, we can see that the BF is a graph transform, similar to the one in (6), operating on the normalized input signal $\hat{\mathbf{x}}_{in} = \mathbf{D}^{1/2}\mathbf{x}_{in}$ producing the normalized output $\hat{\mathbf{x}}_{out} = \mathbf{D}^{1/2}\mathbf{x}_{out}$. This normalization allows us to define the BF in terms of the non-negative definite matrix $\boldsymbol{\mathcal{L}}$ and thus have a spectral interpretation. It also ensures that a constant signal, when normalized, is an eigenvector of $\boldsymbol{\mathcal{L}}$ associated with zero eigenvalue [21]. Following (6) we have,

$$\hat{\mathbf{x}}_{out} = \mathbf{U}(\mathbf{I} - \boldsymbol{\Lambda})\mathbf{U}^t\hat{\mathbf{x}}_{in} \tag{8}$$

This shows that the BF is a frequency selective graph transform with a spectral response $h_{BF}(\lambda_i) = 1 - \lambda_i$ which corresponds to linear decay (See Figure 2). The BF tries to preserve the low frequency components and attenuate the high frequency components.

The BF is used iteratively in many applications. There are two ways to iterate the BF: (1) by changing the weights at each iteration using the result of previous iteration, or (2) by using fixed weights at each iteration as calculated from the initial image. In the first method the BF graph changes in every iteration and thus we only have separate spectral interpretations for each stage. In the second method the graph remains fixed at every iteration and it is possible to provide a spectral interpretation of the cascaded operation. The second method is also faster to compute since the BF weights are only computed once in the beginning. Here we consider the second method. Iterating preserves strong edges while removing weaker details. This type of effect is desirable for applications such as stylization [2]. The BF iterated k-times can be written in matrix notation as

$$\mathbf{x}_{out} = \left(\mathbf{D}^{-1}\mathbf{W}\right)^k \mathbf{x}_{in} = (\mathbf{I} - \boldsymbol{\mathcal{L}}_r)^k \mathbf{x}_{in} \tag{9}$$

where $\boldsymbol{\mathcal{L}}_r = \mathbf{D}^{-1}\mathbf{L}$ is called the random walk Laplacian matrix. It can be shown that any graph transform $h(\boldsymbol{\mathcal{L}}_r)$ can be written in terms of $\boldsymbol{\mathcal{L}}$ as $h(\boldsymbol{\mathcal{L}}_r) = \mathbf{D}^{-1/2}h(\boldsymbol{\mathcal{L}})\mathbf{D}^{1/2}$ [21, Proposition 2]. Using this fact, we can rewrite (9) as

$$\hat{\mathbf{x}}_{out} = \mathbf{U}(\mathbf{I} - \boldsymbol{\Lambda})^k\mathbf{U}^t\hat{\mathbf{x}}_{in} \tag{10}$$

The spectral responses corresponding $k = 2, 3, 4$ are shown in Figure 2. The figure suggests that iterative application of the BF suppresses more of the high frequency component which is consistent with the observation. Equations (8) and (10) give a different perspective to look at the bilateral filter. They hint at filter designs with better spectral responses that can be tailored to particular applications.

4. APPLICATION SPECIFIC SPECTRAL DESIGNS

The BF and iterated BF have fixed spectral responses. But these responses may not be suitable for all applications. Below we discuss two applications (1) image denoising and (2) segmentation to illustrate possible alternative spectral filter designs.

Denoising. We consider the problem of image denoising with additive zero-mean white noise.

$$\mathbf{y}[i] = \mathbf{x}[i] + \mathbf{e}[i] \tag{11}$$

where $\mathbf{x}$ is the original image that we want to estimate, $\mathbf{y}$ is the observed noisy image and $\mathbf{e}$ is zero-mean white noise with variance σ^2. As explained above, we can represent the image as a graph with links weights between pixels computed as in (2). Bilateral filtering on the image amounts to lowpass filtering with a linear decay on the corresponding BF graph. Since energy in the high frequency spectrum of BF graph mostly corresponds to noise, attenuating high graph frequency components of the noisy image leads to improved SNR. However, the BF has a fixed spectral decay profile, because of which attenuation of different frequencies can not be controlled. Using the proposed spectral interpretation, we can design better frequency selective alternatives. For this, we use a regularization framework where the problem of denoising is equivalent to minimizing a penalty functional [7]. This penalty functional can be formulated as in (12), which is composed of two terms: the first term is a fit measure and the second term is a data dependent smoothness constraint as captured by a regularization operator $\mathbf{H}$ on the BF graph [22, 23].

$$C(\hat{\mathbf{x}}) = \frac{1}{2}\|\hat{\mathbf{y}} - \hat{\mathbf{x}}\|^2 + \frac{\rho}{2}\|\mathbf{H}\hat{\mathbf{x}}\|^2. \tag{12}$$

Note that we normalize $\mathbf{x}$ and $\mathbf{y}$ as in (7). In order to formulate this regularization framework in the spectral domain, we choose $\mathbf{H} = h_p(\boldsymbol{\mathcal{L}})$, which has a spectral response $h_p(\lambda)$. Thus, smoothness constraint in (12) can be rewritten as:

$$\|h_p(\boldsymbol{\mathcal{L}})\hat{\mathbf{x}}\|^2 = \sum_{i=1}^{n}[h_p(\lambda_i)]^2[\tilde{\hat{x}}(\lambda_i)]^2. \tag{13}$$

The response $h_p(\lambda)$ is chosen to be a non-negative, non-decreasing function in λ, so that it strongly penalizes the high frequency components. Setting $\partial C(\hat{\mathbf{x}})/\partial\hat{\mathbf{x}} = 0$ in (12), and using (13) we get the optimal $\hat{\mathbf{x}}$ as

$$\begin{aligned}\hat{\mathbf{x}}_{opt} &= (\mathbf{I} + \rho h_p^2(\boldsymbol{\mathcal{L}}))^{-1}\hat{\mathbf{y}} \\ &= \mathbf{U}(\mathbf{I} + \rho h_p^2(\boldsymbol{\Lambda}))^{-1}\mathbf{U}^t\hat{\mathbf{y}}\end{aligned} \tag{14}$$

Thus, for a chosen regularization functional $h_p(\lambda)$, the spectral response of the denoising filter is given by

$$h_{opt}(\lambda) = \frac{1}{1 + \rho h_p^2(\lambda)}, \tag{15}$$

which can be translated in vertex domain as :

$$h_{opt}(\boldsymbol{\mathcal{L}}) = (\mathbf{I} + \rho h_p^2(\boldsymbol{\mathcal{L}}))^{-1}. \tag{16}$$

Since $h_p(\boldsymbol{\mathcal{L}})$ is designed as a highpass filter, *the denoising filters* $h_{opt}(\boldsymbol{\mathcal{L}})$ *in* (16)*, is essentially a lowpass filter on the BF graph.*

Image Segmentation. Shi and Malik [24] formulated the problem of image segmentation as a graph partitioning problem on a graph similar to the BF graph. To obtain an m-way partition of the graph, we find the projection of the graph signal on the first few eigenvectors with the smallest eigenvalues. This projection gives a very coarse version of the signal which then can be used to perform image segmentation. Iterative application of the BF also gives a coarse version of the image, but it favors the eigenvector with the smallest eigenvalue [6], which corresponds to the DC component, and attenuates the contributions of other eigenvectors. The projection of the graph signal onto the first few eigenvectors can be interpreted as *applying a low pass filter in the graph spectral domain with a small cut off frequency and sharp transition band*. In Section 6, we compare the iterated BF with such a lowpass filter on the BF graph, and show that the latter preserves the edge structure of the image better than the former.

To summarize, different applications require filters having different spectral responses $h(\lambda)$. So, we would like to design more general bilateral-like filters of the form of (6) with desired spectral response. A direct implementation of these filters requires diagonalization of $\boldsymbol{\mathcal{L}}$ which is of the order $O(N^3)$. For large graphs such as the one considered here, this is computationally very expensive. However, we show that filters with a polynomial representation in the spectral domain have an easy and efficient implementation scheme in terms of cascaded BFs, as explained in the next section.

5. POLYNOMIAL APPROXIMATION AND FAST IMPLEMENTATION

The spectral response of the iterative bilateral filter (Figure 3(a)) given in (10) is a degree k polynomial. This is a special case of a general class of real polynomials of degree k given as

$$h(\boldsymbol{\Lambda}) = r_0\prod_{i=1}^{k}(\mathbf{I} - r_i\boldsymbol{\Lambda}), \tag{17}$$

where the roots r_i can be either real or complex conjugate pairs. This generalization allows $k+1$ degrees of freedom in choosing the spectral response of the filter. Further, the corresponding transform $\mathbf{H}$ in pixel domain is a matrix polynomial of $\boldsymbol{\mathcal{L}}_r$ with the same roots as in (17) i.e.,

$$\mathbf{H} = \mathbf{D}^{-1/2}\mathbf{U}h(\boldsymbol{\Lambda})\mathbf{U}^t\mathbf{D}^{1/2} = r_0\prod_{i=1}^{k}(\mathbf{I} - r_i\boldsymbol{\mathcal{L}}_r). \tag{18}$$

This leads to following result:

Theorem 5.1. *Any graph filter on an image having a polynomial spectral response of degree k can be implemented in the pixel domain as an iterative k step bilateral filter operation.*

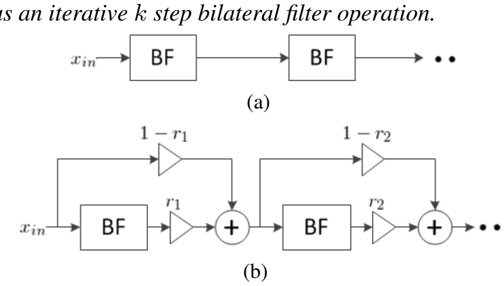

Fig. 3: (a) Iterated BF (b) Fast implementation of a bilateral-like filter with polynomial spectral response using BF as a building block

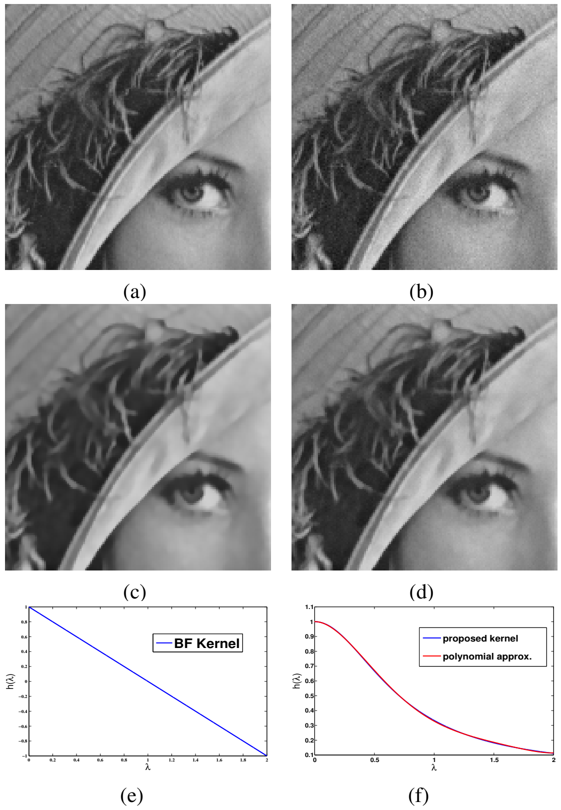

Fig. 4: (a) Original image (b) Noisy image, SNR = 20 dB (c) Output of the BF($\sigma_r = 0.035, \sigma_d = 2$), SNR = 20.65 dB (d) Output of the proposed filter, SNR = 22.64 dB (e) Spectral response of the BF (f) Spectral response of proposed filter

Proof. For $k = 1$, the filter $\mathbf{H} = \mathbf{I} - r_1 \boldsymbol{\mathcal{L}} = (1 - r_1)\mathbf{I} + r_1 \mathbf{D}^{-1}\mathbf{W}$. The output in this case is given in (19). Thus, filter $\mathbf{H}$ can be interpreted as a *generalized linear BF* with intercept $(1 - r_1)$, and slope r_1 in the graph spectrum.

$$\mathbf{y} = \mathbf{H}\mathbf{x} = (1 - r_1)\mathbf{x} + r_1 \mathbf{D}^{-1}\mathbf{W}\mathbf{x}, \tag{19}$$

For $k > 1$, the transform $\mathbf{H}$ is a cascaded form of k such bilateral filtering operations (See Figure 3(b)). □

Further, if $h(\lambda)$ is not a polynomial of λ, it can be approximated with a polynomial kernel which can then be implemented as a generalized iterative bilateral filtering operation. It has been shown in [15] that minimax polynomial approximation of any kernel $h(\lambda)$ not only minimizes the Chebychev norm (worst-case norm) of the error between kernel and its approximation, it also minimizes the upper-bound on the error $||H^{\text{exact}} - H^{\text{approx}}||$ between exact and approximated filters. In our experiments, we approximate any non-polynomial $h(\lambda)$ with the truncated Chebychev polynomials (which are a good approximation of minimax polynomials).

6. EXAMPLES

We examine the performance of the proposed spectral design of bilateral-like filters in two applications. First, we consider the image denoising problem, and use denoising filter obtained by the regularization framework in Section 4. We take the regularization functional $h_p(\lambda) = \lambda$ in (15), which results in a denoising filter $h_{opt}(\boldsymbol{\mathcal{L}})$ with response $h_{opt}(\lambda) = 1/(1 + \lambda^2)$. Since, $h_{opt}(\lambda)$ is not a polynomial, we approximate it with a 5 degree polynomial using Chebychev approximation, which can be implemented as 5 iterations of cascaded generalized linear BFs, using (18). Figure 4 shows the denoising results using this filter and the BF [2]. It can be seen that the BF preserves edges, but it blurs the texture in the denoising process while the proposed denoising filter does a better job of preserving texture. This is also reflected in the SNR values.

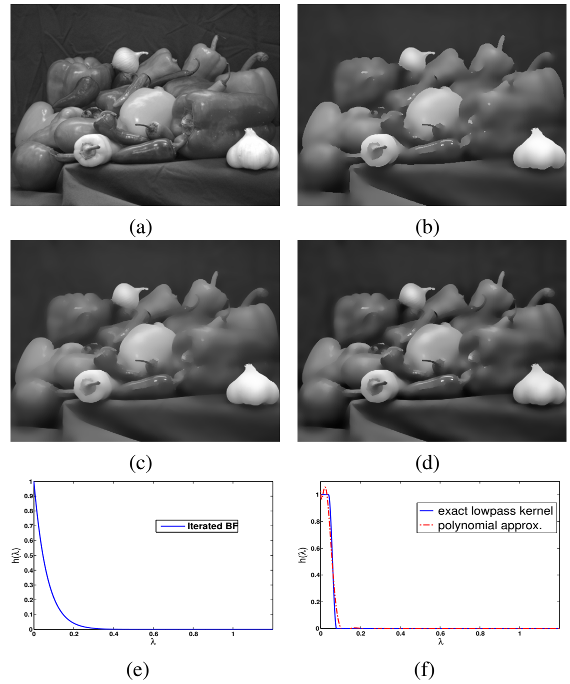

Fig. 5: (a) Original image (*peppers*) (b) Output of 20 iteration of the BF with changing weights (c) Output of 20 iteration of the BF with fixed weights($\sigma_r = 0.05, \sigma_d = 2$) (d) output of the proposed spectral filter (e) Iterated BF's spectral response (f) Proposed spectral response

Next, we consider an iterative application of the BF. Iterated BF removes minor details from the image while preserving prominent edges. This can be used as an effective preprocessing step in edge-detection and segmentation etc. As stated before, iterated BF favors the DC component which is not useful for segmentation. We use a low pass spectral kernel with small cut-off and sharp transition band so that the second (and a few higher) spectral components get at least as much weight as the DC component. We use a 20 degree polynomial approximation of this kernel and compare it with 20 iterations of the BF. Figure 5 shows that weak edges are blurred more with iterated BF (as expected from the spectral response) compared to the proposed filter.

7. CONCLUSION

In this paper we interpreted the bilateral filter as a graph spectral filtering operation. With this novel perspective, we proposed a family of more general bilateral-like filters with desired spectral responses. We gave an easy implementation scheme for these filters. Their utility was motivated through few examples. An immediate interesting extension to this work would be to explore different spectral filters suitable for particular applications. Another topic of interest is the design of filter banks using these bilateral-like filters.

[2]The BF is not the best denoising filter available. We use it in our comparison to emphasize the qualitative differences in filtering results.

8. REFERENCES

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

[2] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "Bilateral filtering: Theory and applications," *Foundations and Trends in Computer Graphics and Vision*, vol. 4, no. 1, 2009.

[3] Ming Zhang and B. K. Gunturk, "Multiresolution bilateral filtering for image denoising," *Image Processing, IEEE Transactions on*, vol. 17, no. 12, pp. 2324 –2333, Dec 2008.

[4] R. Fattal, M. Agrawala, and S. Rusinkiewicz, "Multiscale shape and detail enhancement from multi-light image collections," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.

[5] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 257–266, July 2002.

[6] C. Ye, Y. Lin, M. Song, C. Chen, and D. Jacobs, "Spectral graph cut from a filtering point of view," *arXiv:1205.4450v2 [cs.CV]*, 2012.

[7] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1141–1151, Oct 2002.

[8] D. Barash, "Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 844–847, 2002.

[9] 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.

[10] A. Singer, Y. Shkolnisky, and B. Nadler, "Diffusion interpretation of nonlocal neighborhood filters for signal denoising," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 118–139, 2009.

[11] A. Buades, B. Coll, and J.M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[12] G. Peyré, "Image processing with nonlocal spectral bases," *SIAM Journal on Multiscale Modeling and Simulation*, vol. 7, no. 2, pp. 703–730, 2008.

[13] G. Noel, K. Djouani, B. Van Wyk, and Y. Hamam, "Bilateral mesh filtering," *Pattern Recognition Letters*, vol. 33, no. 9, pp. 1101–1107, 2012.

[14] C. Jian, K. Inoue, K. Hara, and K. Urahama, "Fixed-coefficient iterative bilateral filters for graph-based image processing," in *Advances in Image and Video Technology*, vol. 5414 of *Lecture Notes in Computer Science*, pp. 473–484. Springer, 2009.

[15] 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.

[16] S.K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2786–2799, June 2012.

[17] 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," *to appear in IEEE Signal Processing Magazine*, 2013.

[18] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *arXiv:1210.4752v2 [cs.SI]*, 2012.

[19] E.B. Davies, G.M.L. Gladwell, J. Leydold, and P.F. Stadler, "Discrete nodal domain theorems," *Linear Algebra and its Applications*, vol. 336, no. 1, pp. 51–60, 2001.

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

[21] S.K. Narang and A. Ortega, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *arXiv:1210.8129 [cs.IT]*, 2012.

[22] A. Smola and R. Kondor, "Kernels and regularization on graphs," in *Proceedings of the Annual Conference on Computational Learning Theory*. 2003, Lecture Notes in Computer Science, Springer.

[23] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *ICML Workshop on Statistical Relational Learning*, 2004, pp. 132–137.

[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.