A Late Adaptive Graph-based Edge-aware Filtering with Iterative Weight Updating Process

Hamidreza Sadreazami †, *Member, IEEE,* Amir Asif †, *Senior Member, IEEE,*
and Arash Mohammadi ∓, *Member, IEEE*
†Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada
∓Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
E-mail: {h_sadrea, amir.asif, arash.moh}@ece.concordia.ca

***Abstract*—In this work, we propose a parallelized graph-based framework for lowpass and highpass edge-aware filtering for detail manipulation (smoothening/boosting). Our proposed filter abstracts images through simplifying their visual content while preserving edges and emphasizing most of the perceptually important information. The proposed filtering framework is realized by using the graph similarity and Laplacian matrices to obtain smoothened image at each layer. The resulted smoothened images are iteratively treated as inputs to the next layer of the filtering framework and the weights are updated accordingly. The efficacy of the proposed image abstraction method is confirmed by conducting simulations. It is shown that the proposed method provides abstracted images having higher quality than those resulted from the other existing works.**

***Index Terms*—Edge-aware filtering, graph signal processing, vertex domain filtering, graph Laplacian matrix, graph similarity matrix.**

I. Introduction

In recent years, the traditional signal processing methodologies have been revisited in order to make them applicable to problems involving large-scale data. Graph signal processing [1]-[2] is a framework for model relation representation of data samples. It is realized by building a similarity graph for data samples at hand in such a way that it can identify the existence of any similarities between them. In image processing, for instance, an image is represented by constructing a graph using the image pixels/super-pixels which can be studied by the graph signal processing tools [1].

Leveraging this framework, we intend to consider the edge-aware image filtering, i.e., edge preserving smoothening filter, from a new perspective. There exist several edge-aware filters with different construction forms and filtering properties. Such filters have had a great importance in recent advances in various computer vision tasks. In addition, they have found to be inevitable and fundamental building blocks for image saliency detection, retargeting and image stylization applications [3], [4].

Edge-aware filtering is a problem of smoothening the low-contrast image features to facilitate further processing. Some of the existing edge-aware filtering methods are as follows; In [5], in an image editing application, an l_0 gradient minimization technique has been proposed in which a global function is optimized to regulate the gradients. In [6], in order to smoothen the image details, a scale-aware rolling guidance filter (RGF) has been proposed. This method is indeed an iterative joint bilateral filter which can control the detail smoothening process by a scale measure. In [7], an edge-aware filter has been proposed using a graph-based method through building a minimum spanning tree between image patches. In this method, the high-contrast details have been smoothened by an internal weight function, while major image structure are preserved. In [4], for image stylization purpose, a multi-layer image abstraction method has been proposed based on graph filtering with no weight updating. In this method, weights have been computed once for the original image and smoothening has been performed by iteratively adding a residual to the smoothened image at different layers. In [8], an image enhancement method has been proposed for smoothening, sharpening and tone manipulation, by approximating the graph Laplacian matrix. Although there exist many edge-preserving smoothening methods in the literature, we investigate the possibility of a parallelized and flexible edge-aware filtering for simultaneous data manipulation and boosting purposes.

It is known that most of the nonlinear multi-scale decomposition filters rely on successive computation of the filter weights [9]. In this sense, the proposed filtering method is similar to [6], since both methods employ an iterative weight updating process. In addition, a graph-based filtering framework has been proposed in [4]. However, the filter weights have been precomputed only once and different versions of the filter produced by direct product of the weights. Thus, the process at each layer does not depend on the result (smoothened image) of the previous layer. In this paper, we propose an adaptive multi-layer edge-aware filtering framework using the graph-based filtering of images. Different from [4], our proposed filtering framework adaptively updates the graph similarity and Laplacian matrices at each layer and iteratively produce images with different resolutions. The smoothened images at each layer are treated as inputs for the next layer. Simulation results are carried out to evaluate the effectiveness of the proposed graph-based filtering method with weight updating

This work was partially supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada through the Create Grant 466280-2015.

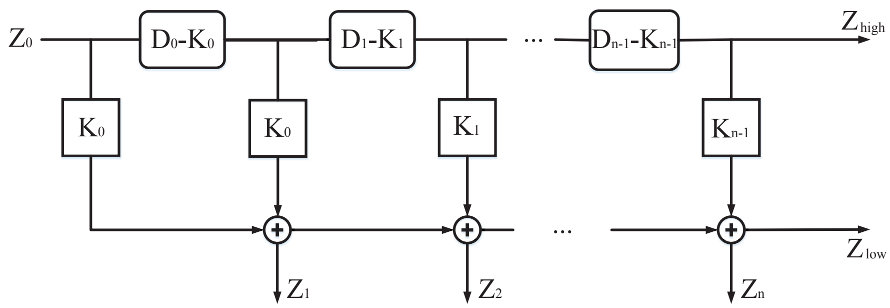


Fig. 1. Adaptive graph-based filtering for edge-aware filtering with graph similarity and Laplacian matrices updated at each layer.

strategy and to compare it with that of the other existing works including its no weight updating version, in terms of the quality of the smoothened images as well as the computational complexity of the methods.

II. Graph Construction

Graph signal processing is known to provide a framework for large-scale data analysis [1]. This framework regards signals in various domains indexed by graphs [2]. For instance in image processing, an image can be seen by its pixel intensity values as signals defined on a graph, where vertices V and edges E in the graph correspond to pixels and their similarities in the image, respectively. In other words, graph G is constructed by image pixels and edges having the weights $\boldsymbol{k}_{pq} \in \boldsymbol{K}$, identifying the relation between the pixels p and q in the graph. The corresponding graph similarity matrix is given by $\boldsymbol{K} = [\boldsymbol{k}_{pq}]$, where $\boldsymbol{k}_{pq}$ is defined as

$$\boldsymbol{k}_{pq} = \boldsymbol{k}_c * \boldsymbol{k}_g, \tag{1}$$

where

$$\boldsymbol{k}_c = \exp\left(-\left[\frac{(l_p - l_q)^2 + (a_p - a_q)^2 + (b_p - b_q)^2}{\sigma_c^2}\right]\right), \tag{2}$$

is the color distance, and

$$\boldsymbol{k}_g = \exp\left(-\left[\frac{(x_p - x_q)^2 + (y_p - y_q)^2}{\sigma_g^2}\right]\right), \tag{3}$$

is the geometrical distance, σ_c and σ_g determine the exponential decay rate, x and y represent the image pixel position, l denotes the luminance intensity level, and finally a and b represent the color components [10]. Having the graph similarity matrix $\boldsymbol{K}$, the corresponding normalized graph Laplacian matrix $\boldsymbol{L}$ is defined as [11]

$$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{K}\boldsymbol{D}^{-1/2}, \tag{4}$$

where $\boldsymbol{I}$ is the identity matrix and

$$\boldsymbol{D} = \operatorname{diag}\left\{\sum_q \boldsymbol{k}_{1q}, ..., \sum_q \boldsymbol{k}_{\acute{n}q}\right\}. \tag{5}$$

The graph Laplacian matrix is a real symmetric matrix, thus diagonalizable, and can be decomposed into a set of eigenvalues $\lambda = \{\boldsymbol{\lambda}_e\}_{e=1,...,\acute{n}}$ and a set of orthogonal eigenvectors $U = \{\boldsymbol{u}_e\}_{e=1,...,\acute{n}}$, as given by

$$\boldsymbol{L} = \sum_e \boldsymbol{\lambda}_e \boldsymbol{u}_e \boldsymbol{u}_e^T. \tag{6}$$

In the next Section, the proposed adaptive graph-based edge-aware filtering with weight updating at each layer is presented.

III. Proposed Edge-aware Image Filtering

The proposed edge-aware filtering method is realized by applying a progressive filtering process consisting of smoothening and sharpening filters. In other words, the original image is smoothened using the proposed graph-based multi-layer filter, where graph similarity and Laplacian matrices are updated at each layer. As a result, the level of details can be controlled in such a way that in low-contrast regions, weak edges and possible noise are eliminated and the contrast is adaptively reduced, while in high-contrast regions, the significant edges are not overly smoothened. In [4], the filter weights were computed from the original image and used in the entire process without update. Thus, the process at each layer does not depend on the result (smoothened image) of the previous layer. The filtering process proposed in [4] can be formulated for an image Z as follows

$$\begin{aligned} Z_1 &= ZK + Z(D-K)K \\ Z_2 &= Z_1 + Z(D-K)^2K \\ &. \\ &. \\ &. \\ Z_i &= Z_{i-1} + Z(D-K)^iK. \end{aligned} \tag{7}$$

where $\{\boldsymbol{Z}_i\}_{i=1,...,n}$ are the smoothened images at different layers. By substituting for Z_1 into Z_2 and following the substituting iteratively for $Z_j, (1 \le j \le i-1)$, we have

$$Z_{\text{low}} = I - (I - D^{-1/2}KD^{-1/2})^{n+1}Z. \tag{8}$$

In contrast, the proposed adaptive edge-aware filtering is capable of iteratively smoothening images in different layers, by obtaining the graph similarity and Laplacian matrices at each layer from the resulted abstracted images. In other words, the proposed method is iteratively smoothened images in different layers. The block diagram of the proposed graph-based filtering is shown in Fig. 1. The proposed graph-based multi-layer filtering with consecutive weight update is given by

$$\begin{aligned}
Z_1 &= Z_0K_0 + Z_0(D_0 - K_0)K_0 \\
Z_2 &= Z_1 + Z_1(D_1 - K_1)K_1 \\
Z_3 &= Z_2 + Z_2(D_2 - K_2)K_2 \\
&. \\
&. \\
&. \\
Z_i &= Z_{i-1} + Z_{i-1}(D_{i-1} - K_{i-1})K_{i-1}.
\end{aligned} \tag{9}$$

By repeating the similar substitution procedure for Z_j, the Z_{low} is obtained as

$$\begin{aligned}
Z_{\text{low}} &= Z_1(I + \sum_{i=1}^{n-1} (D_i - K_i)\, K_i \\
&+ \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} (D_i - K_i)\, K_i\, (D_j - K_j)\, K_j \\
&+ \prod_{i=1}^{n-1} (D_i - K_i)\, K_i).
\end{aligned} \tag{10}$$

It should be noted that for both the cases, with or without weight updating, the filtering process can also be performed in the spectral domain. In graph spectral domain, U and λ represent the graph basis function and frequencies for the underlying graph signals, respectively [12]. Accordingly, the graph Fourier transform $\tilde{\boldsymbol{z}}$ of a signal $\boldsymbol{z}$ can be defined as $\tilde{\boldsymbol{z}} = \sum_e \boldsymbol{u}_e \boldsymbol{z}$ [1]. The signal is filtered in the graph Fourier domain as

$$\tilde{z}_f = h(\lambda)\tilde{z}, \tag{11}$$

where

$$h(\boldsymbol{\lambda}) = \sum_{0 \le e \le f} a_e \boldsymbol{\lambda}_e, \tag{12}$$

is the spectral response of the filter [13]. In the proposed filtering method, $h(\lambda)$ is represented by a polynomial expressed as a function of $\boldsymbol{\lambda}$ with degree f. Note that, the filter spectral response can be derived for different polynomial degrees [3].

IV. Simulation Results

We carry out experiments on standard color images to assess the performance of the proposed adaptive edge-aware filter with iterative weight updating process. To this end, the graph similarity matrices $\{\boldsymbol{K}_i\}_{i=1,\ldots,n}$ are consecutively constructed for the original and smoothened images, where image pixels are connected to their eight neighbors; horizontally, vertically and diagonally. The corresponding normalized Laplacian matrices are also obtained at each layer and used as residuals to enhance the smoothened images. The obtained image at each layer is treated as a new input to the graph construction procedure and the weights are updated accordingly. Fig. 2 shows one of the test images, its synthetically-noisy and smoothened versions obtained using the proposed adaptive graph-based filtering framework with consecutive weight updating as well as those obtained using the method in [4], for two iteration indices, $n = 3$ and 5. It is seen from this figure that graph filters smoothen the image at each layer by reducing fine details and preserving the strong edges. It is also seen that updating weights at each layer increases the level of smoothness and to some extent compromise the sharpness of the edges. In other words, the proposed edge-aware filter do well in reducing noise level and weak edges while it makes discontinuous values more continuous and causing blurriness at edges.

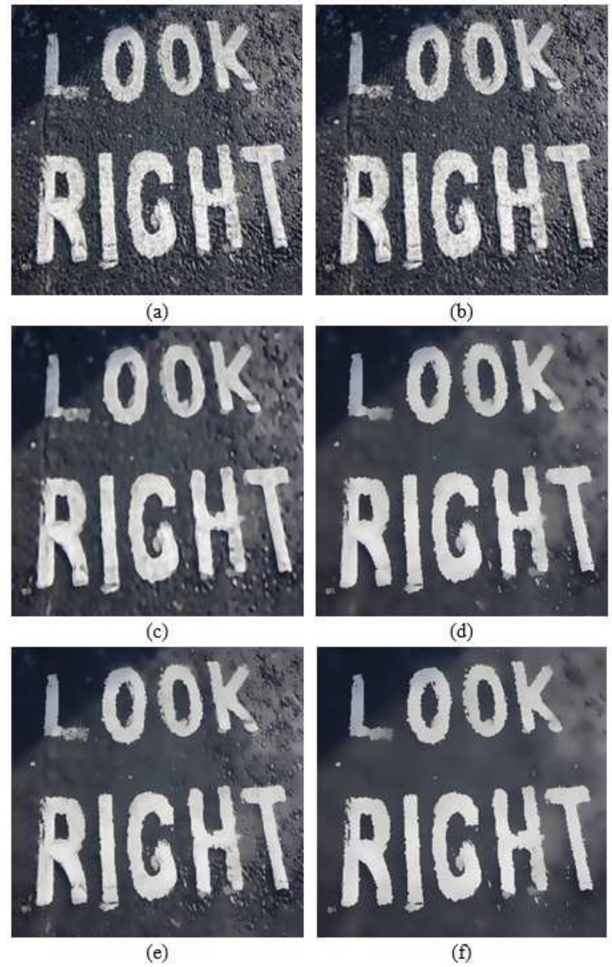
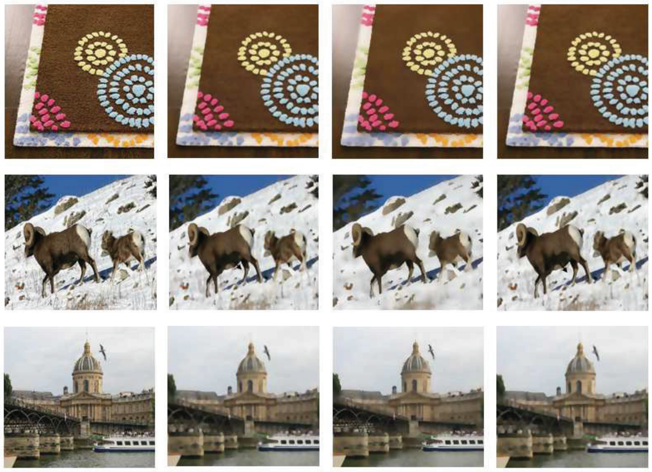


Fig. 2. (a) original image. (b) Noisy image. (c) Result of [4] with $n = 3$. (d) Result of [4] with $n = 5$. (e) Proposed method with $n = 3$. (f) Proposed method with $n = 5$.

The performance of the proposed edge-aware filter is com-



(a) (b) (c) (d)

Fig. 3. Smoothened images obtained using the proposed adaptive graph-based edge-aware filtering method as well as those obtained using [4] and [6]. (a) Original image. (b) Result of [6]. (c) Result of [4]. (d) Proposed Method.

TABLE I
NUMBER OF MULTIPLICATIONS AND ADDITIONS REQUIRED BY THE PROPOSED ADAPTIVE GRAPH-BASED FILTERING AND THOSE OF [4], FOR A SIMILARITY MATRIX K OF SIZE $m \times m$.

	multiplications	additions
Proposed	$21m^3$	$21m^3 - 6m^2$
[4]	$4m^3 + 2m^2$	$4m^3 - 2m$

pared to that of the [4] and [6]. Fig. 3 shows the original images and their corresponding filtered versions obtained using various methods. It can be seen from this figure that the proposed adaptive graph-based filtering method with consecutive weight updating can smoothen the images better than [4] and [6]. It is also seen from this figure that as a result of updating weights at each layer, the smoothened images tend to be less sharper and resolution contrast decreases.

A. Computational Complexity

We now analyze the computation requirements of the our proposed adaptive graph-based filter with consecutive weight update and compare it to its counterpart which requires no weight updating process. More specifically, the number of multiplications and additions required by both the methods are computed according to [14]. Table I gives the computational complexity of the two approaches. It is seen from this table that as a result of using the proposed filter with consecutive weight updating procedure, the computational complexity increases. In other words, the computation required by the filtering method in [4] without requiring any weight updates is lower by a factor of about 5 than that of the proposed edge-aware filtering with weight updating. Note that the number of iteration is set to $n = 4$ for both the cases.

V. CONCLUSION

In this paper, we have proposed an adaptive edge-aware filtering framework by leveraging the recent advances in graph signal processing. The proposed filtering method has been realized by iterative graph construction for the smoothened images and updating the graph similarity and Laplacian matrices at each layer. Experiments have been carried out using standard images. The results have shown that the proposed method can produce multi-layer filtered images while retaining much of their perceptually important information. It has been shown that the proposed adaptive filter with consecutive weight updating is capable of removing fine image details better than the other methods. It has also been shown that the proposed method compromises the sharpness of the edges.

REFERENCES

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Processing Magazine,* vol. 30, no. 3, pp. 83-98, 2013.

[2] A. Sandryhaila and J. M. F. Moura, “Big data processing with signal processing on graphs,” *IEEE Signal Processing Magazine,* vol. 31, no. 5, pp. 80-90, 2014.

[3] H. Sadreazami, A. Asif and A. Mohammadi, “Image stylization using iterative graph filtering,” in *Proc. IEEE Canadian Conf. on Electrical and Computer Engineering (CCECE),* pp. 1-4, 2017.

[4] H. Sadreazami, A. Asif and A. Mohamamdi, “Iterative graph-based filtering for image abstraction and stylization,” *IEEE Transactions on Circuits & Systems II: Express Briefs,* 2017.

[5] L. Xu, C. Lu, Y. Xu and J. Jia, “Image smoothening via l_0 gradient minimization,” *ACM Trans. on Graphics,* vol. 30, no. 6, 2011.

[6] Q. Zhang, L. Xu and J. Jia, “Rolling guidance filter,” in *Proc. European Conference on Computer Vision (ECCV),* 2014.

[7] F. Zhang, L. Dai, Sh. Xiang and X. Zhang, “Segment graph based image filtering: fast structure-preserving smoothing,” in *Proc. IEEE International Conference on Computer Vision (ICCV),* 361-369, 2015.

[8] H. Talebi and P. Milanfar, “Fast multilayer Laplacian enhancement,” *IEEE Transactions on Computational Imaging,* vol. 2, No. 4, pp. 496-509, 2016.

[9] S. Paris, P. Kornprobst, J. Tumblin and F. Durand, “Bilateral filtering: theory and applications,” *Fundamental and Trends in Computer Graphics and Vision,* vol. 4, no. 1, 1-73, 2008.

[10] S. Yagyu, A. Sakiyama and Y. Tanaka, “Pyramidal image representation with deformation: Reformulation of domain transform and filter designs,” in *Proc. IEEE International Conference on Image Processing (ICIP),* pp. 3608-3612, 2016.

[11] P. Milanfar, “A tour of modern image filtering,” *IEEE Signal Processing Magazine,* vol. 30, no. 1, pp. 106-128, 2013.

[12] H. Sadreazami, A. Asif and A. Mohammadi, “Data-adaptive color image denoising and enhancement using graph-based filtering,” in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS),* pp. 2751-2754, 2017.

[13] A. Gadde, S. K. Narang and A. Ortega, “Bilateral filter graph spectral interpretation and extensions,” in *Proc. IEEE International Conference on Image Processing (ICIP),* pp. 1222-1226, 2013.

[14] J. M. Mendel, “Computational requirements for a discrete Kalman filter,” *IEEE Transactions on Automatic Control,* vol. ac-16, no. 6, 1971.