Motion Deblurring With Graph Laplacian Regularization

Amin Kheradmand and Peyman Milanfar

Department of Electrical Engineering
University of California, Santa Cruz

ABSTRACT

In this paper, we develop a regularization framework for image deblurring based on a new definition of the normalized graph Laplacian. We apply a fast scaling algorithm to the kernel similarity matrix to derive the symmetric, doubly stochastic filtering matrix from which the normalized Laplacian matrix is built. We use this new definition of the Laplacian to construct a cost function consisting of data fidelity and regularization terms to solve the ill-posed motion deblurring problem. The final estimate is obtained by minimizing the resulting cost function in an iterative manner. Furthermore, the spectral properties of the Laplacian matrix equip us with the required tools for spectral analysis of the proposed method. We verify the effectiveness of our iterative algorithm via synthetic and real examples.

Keywords: Motion deblurring, Graph Laplacian regularization, Kernel similarity matrix.

1. INTRODUCTION

Many images taken with existing cameras exhibit some level of distortion in the form of blur and/or noise. Specifically, with existing hand held cameras, motion blur in the captured images is inevitable. The underlying image capturing process can be described as

$$\mathbf{y} = A\mathbf{z} + \mathbf{e}, \tag{1}$$

in which, $\mathbf{y}$ is the captured blurry and noisy image in vector stacked form, $\mathbf{z}$ is the ground truth image vector, and $\mathbf{e}$ is additive white noise. All the vectors are of dimension n which is the total number of pixels in the image. A is an $n \times n$ blurring matrix whose structure depends on boundary condition assumptions.[1] In this paper, we focus on a simpler case where the motion blur kernel or equivalently the blurring matrix A is known or otherwise estimated using one of the existing kernel estimation methods.[2,3]

Deblurring is an ill-posed problem and most of the existing approaches are using some type of regularization to avoid noise amplification and ringing artifacts in the final solution. Some existing non-blind deblurring papers use TV regularization terms.[4,5] Nonlocal TV-based regularization terms are used in.[6–8] Hyper-Laplacian prior based on the statistics of natural images is used in.[9,10] In,[11] a hardware attachment is used to deblur images. Also, a progressive inter-scale intra-scale approach has been exploited for non-blind image deconvolution in.[12] Shan et al. have proposed a cost function for motion deblurring with different derivative terms in the data fidelity term.[3] Cho et al. propose a blur model for handling outliers in deblurring problem.[13] In,[14,15] multiple images are used to improve the performance of deblurring.

In this paper, we use a graph theoretic approach to define a new cost function for motion deblurring with a new definition of the normalized graph Laplacian. A fast, symmetry preserving matrix balancing algorithm is applied to the kernel similarity matrix from which filtering and Laplacian matrices are constructed (Figure 1). We use this definition of the normalized graph Laplacian to construct an objective function with new data fidelity and regularization terms. We will show that our approach has performance advantages over the traditional graph Laplacian for motion deblurring. As depicted in Figure 2, our proposed method consists of outer and inner iterations. In each outer iteration, the filtering matrix and the corresponding normalized graph Laplacian are computed from the estimate of the previous step and are used to derive an updated objective function. Due to the specific form of the proposed cost function and spectral properties of the normalized Laplacian, we are

Further author information: (Send correspondence to Amin Kheradmand)
Amin Kheradmand: E-mail: aminkh@soe.ucsc.edu
Peyman Milanfar: E-mail: peyman.milanfar@gmail.com

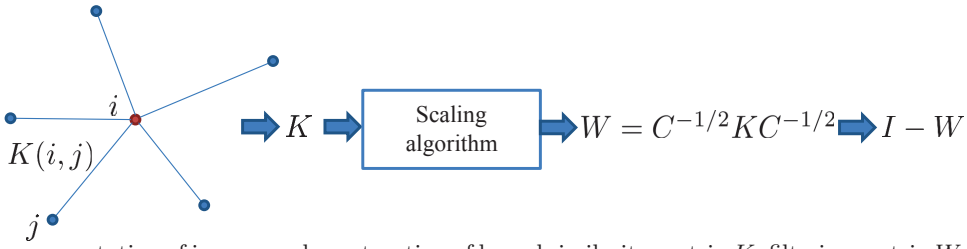


Figure 1. Graph representation of images and construction of kernel similarity matrix K, filtering matrix W and normalized Laplacian $I - W$.

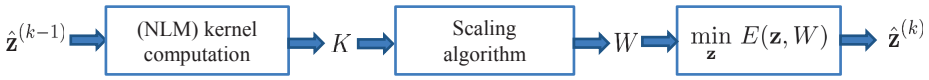


Figure 2. Block diagram of the proposed iterative deblurring method. $\hat{\mathbf{z}}^{(k)}$ is the estimate at the kth outer iteration of the algorithm.

able to minimize the updated cost function at each step using Conjugate Gradient (CG) iterations. Synthetic and real experimental results show that our proposed motion deblurring algorithm has comparable performance with some of the best existing non-blind motion deblurring algorithms and in some cases outperforms them. The paper is organized as follows. In Section 2, we introduce the procedure for deriving the filtering and corresponding normalized graph Laplacian matrices and discuss their spectral properties. In Section 3, we derive the objective function for motion deblurring and analyze the performance of the resulting algorithm based on spectral properties of the normalized graph Laplacian. Section 4 presents experimental results for both synthetic and real motion deblurring examples to verify the effectiveness of the proposed method.

2. PROPOSED NORMALIZED GRAPH LAPLACIAN

As illustrated in Figure 1, any image can be represented as a weighted graph $G = (V, E, K)$ where V is the set of vertices (image pixels). E is the set of edges in the graph with weights computed using some similarity measure between pixels i and j in the image. The similarity measure is usually defined by a kernel function $K(i, j)$. In this paper, we use the NLM kernel definition.[16] As such, the kernel similarity coefficients are computed as

$$K(i,j) = \exp\{-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{h^2}\}, \tag{2}$$

in which $\mathbf{z}_i$ and $\mathbf{z}_j$ are patches around pixels i and j in the image $\mathbf{z}$ and h is a smoothing (scaling) parameter. These coefficients constitute the corresponding elements in the kernel similarity matrix K. As such, the matrix K contains the similarity information among different pixels of the image and is used to compute the filtering and Laplacian matrices. We consider a small neighborhood around each pixel (e.g., a search neighborhood of size 11×11) to construct the sparse matrix K. Due to the specific definition of the kernel similarity function in (2), the similarity matrix K is also symmetric and non-negative. Applying the Sinkhorn matrix scaling algorithm to the matrix K produces the diagonal matrix $C^{-1/2}$ such that the resulting filtering matrix $W = C^{-1/2}KC^{-1/2}$ is symmetric, non-negative and doubly stochastic.[17,18] The filtering matrix W has the constant eigenvector corresponding to the largest eigenvalue 1.[19] This is a desired property for filtering purposes to preserve the average value of the input signal.[19,20] Also, W can be diagonalized as $W = VSV^T$ where V is an orthonormal matrix containing the eigenvectors of W in its columns and S is a diagonal matrix whose diagonal elements are the eigenvalues of W. We define the new normalized Laplacian matrix as

$$I - W = I - C^{-1/2}KC^{-1/2}. \tag{3}$$

Note that the traditional symmetric normalized graph Laplacian is defined as $I - D^{-1/2}KD^{-1/2}$ where $D = diag\{K\mathbf{1}\}$, with $\mathbf{1}$ being an n-dimensional vector of all ones.[21] As will be shown, our definition has performance advantages over the traditional definition. Therefore, we use our normalized graph Laplacian throughout this paper as an appropriate definition for image processing applications.

Table 1. SSIM and PSNR performance of our deblurring algorithm in comparison with those of motion deblurring methods in References[3, 10] for synthetic camera motion blur. In each cell, the first number denotes SSIM value, and the second number represents PSNR value.

Image	ours	Ref.[10]	Ref.[3]
Building	0.9525 28.45	0.9507 28.48	0.9572 28.45
Motocross Bikes	0.9589 26.89	0.9470 26.45	0.9496 26.51
Girl	0.9354 32.15	0.9351 31.99	0.9379 32.08
Street	0.9584 29.75	0.9507 29.32	0.9558 29.35
Average performance	**0.9513** **29.31**	0.9458 29.06	0.9501 29.09

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

Figure 3. Blur kernel and set of color images used for evaluation of our method: (a) Motion blur kernel, (b) Building image (480×640), (c) Motocross bikes image (494×494) , (d) Girl image (496×700), and (e) Street image (480×640).

3. PROPOSED DEBLURRING ALGORITHM

For the deblurring problem, we introduce the following objective function:

$$E(\mathbf{z}) = (\mathbf{y} - A\mathbf{z})^T\{I + \beta(I - W)\}(\mathbf{y} - A\mathbf{z}) + \eta\ \mathbf{z}^T(I - W)\mathbf{z}, \tag{4}$$

in which the first term is the data fidelity term and the second term is the regularization term. $\beta > 0$ and $\eta > 0$ are parameters that need to be tuned based on the amount of noise and blur in the input image. Note that the first term can be rewritten in terms of the matrix $I + \beta(I - W)$ as

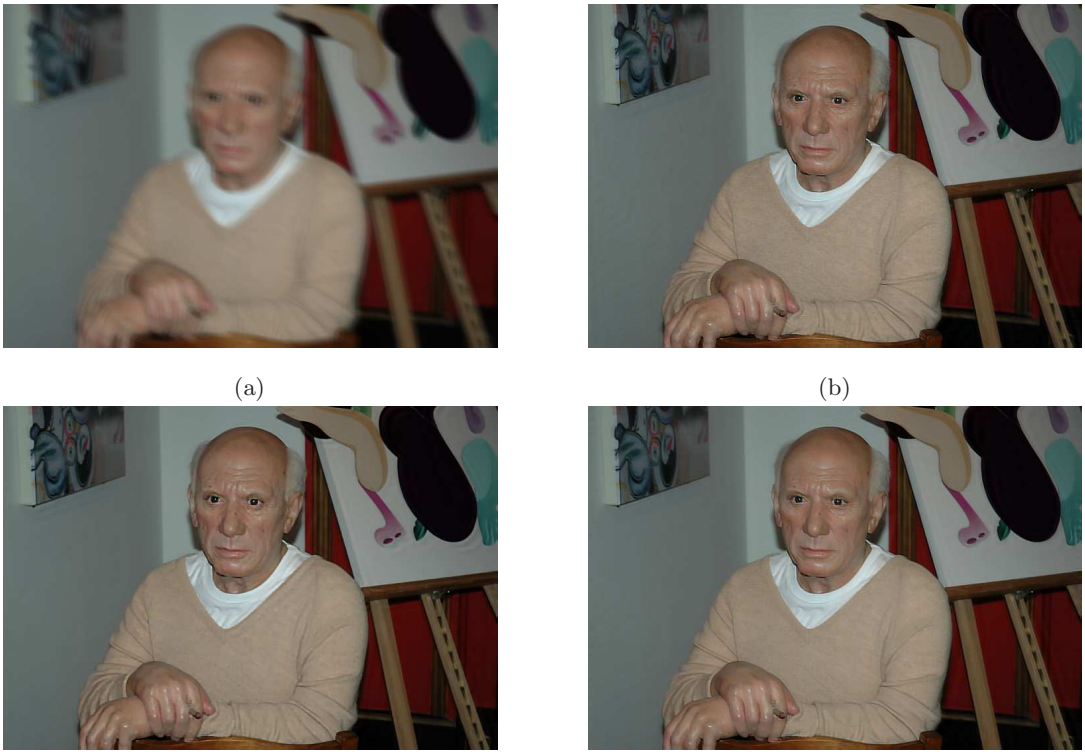
$$\|\{I + \beta(I - W)\}^{1/2}(\mathbf{y} - A\mathbf{z})\|^2. \tag{5}$$

As a result, the first term in (4) prefers a solution $\mathbf{z}$ such that its blurred and then filtered version is as close as possible to the filtered version of the input $\mathbf{y}$. The frequency behavior of this filter is selected by the parameter β. For example, when the amount of noise is high and the image is moderately blurred, then small values of β is chosen, while for low noise and strong blur cases larger values of β are selected. Note that for $\beta > 0$, $I+\beta(I-W)$ behaves like a sharpening filter and so does its square root*. Based on the analysis in,[3] the data term involving different derivatives of the residual is better able to model the underlying motion blurring phenomenon. On the other hand, note that the second term can be written in terms of the normalizing coefficients in the diagonal matrix C as

$$\mathbf{z}^T(I - W)\mathbf{z} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j,j\sim i} K(i,j)(\frac{z(i)}{\sqrt{C(i,i)}} - \frac{z(j)}{\sqrt{C(j,j)}})^2, \tag{6}$$

where $j \sim i$ stands for pixel j connected to pixel i in the corresponding graph or equivalently $(i,j) \in E$. Also, $C(i,i)$ denotes the ith diagonal element of the diagonal matrix C. As is evident from equation (6), the

*The reason is that $I - W$ is a high pass filter, and for $\beta > 0$, $I + \beta(I - W)$ can be considered as a data-adaptive un-sharp mask filter.



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

Figure 4. Real motion deblurring example: (a) input blurred noisy image, (b) Output of hyper-Laplacian algorithm,[10] (c) output of,[3] and (d) output of our algorithm ($\eta = 0.031, \beta = 0.6$).

regularization term is essentially a normalized data-adaptive difference term that penalizes high frequencies in the final solution to avoid noise amplification and ringing artifacts.

The kernel similarity matrix is initially computed from a denoised version of the input image. For this purpose, we use the denoising algorithm in.[22] The kernel similarity coefficients are then updated at each outer iteration from the estimate of the previous step and the resulting objective function is minimized in an iterative manner. The specific form of the cost function and the spectral properties of the normalized Laplacian make it possible to use CG iterations for minimizing (4). Furthermore, for fast computation of the similarity coefficients, we take advantage of the idea of integral images.[23]

4. EXPERIMENTS

In this section, we verify the effectiveness of the proposed deblurring method for motion deblurring problem. For this purpose, we consider both synthetic and real examples. For synthetic examples, we use the nonlinear motion blur kernel in Figure 3. Test images in Figure 3 are then circularly convolved with the blur kernel and additive white Gaussian noise with standard deviation 1 is added to generate noisy blurry images[†]. The performance of our method is then compared with those of algorithms in.[3,10] For real motion deblurring experiments, as our method is a non-blind algorithm, we use existing blur kernel estimation methods[2,3] to derive the motion blur kernel. Our algorithm is then applied independently to the blurred images and the results are compared with those of References.[3,10] The same motion blur kernel estimate is used for all algorithms.

[†]Test images are from Kodak Lossless True Color Image Suite (http://r0k.us/graphics/kodak/) and the web page for.[24]

(a) (b)

(c) (d)

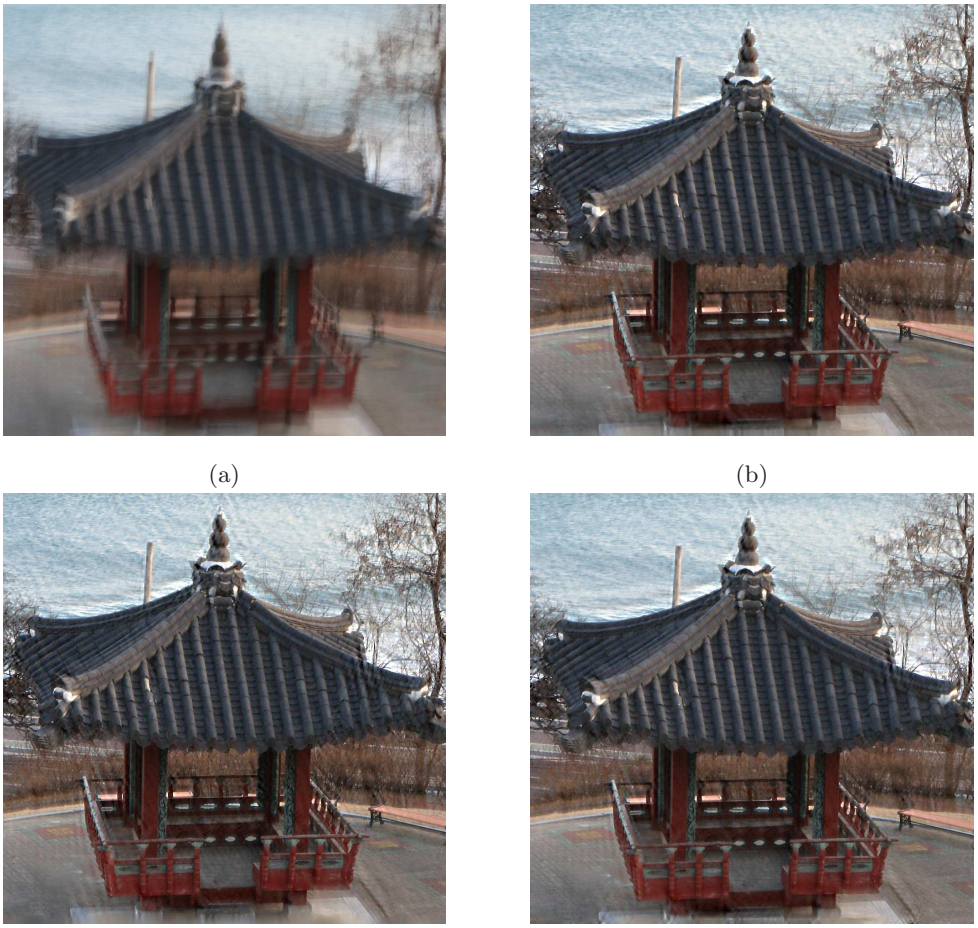
Figure 5. Real motion deblurring example: (a) input blurred noisy image, (b) Output of hyper-Laplacian algorithm,[10] (c) output of,[3] and (d) output of our algorithm ($\eta = 0.25, \beta = 2.5$).

4.1 SYNTHETIC EXAMPLES

For synthetic examples, the set of parameters are selected for the best average performance for each algorithm and are fixed for all the test images. For these experiments, we use periodic boundary conditions. We also use patch size of 5×5, search neighbourhood size of 11×11, and the number of outer iterations equal to 3. The values of the parameters β, η, and h are selected to be 0.01, 0.01, and 6.5, respectively. In order to evaluate the performance of the algorithms, we use two commonly used measures for assessing the performance of image enhancement methods, namely PSNR and SSIM.[25] Table 1 summarizes the numerical results. As can be seen, our algorithm shows better average performance compared to these algorithms.

4.2 REAL EXAMPLES

In this section, we evaluate the performance of the proposed method when applied to real motion blurred examples. In Figures 4 and 5, the motion blur kernels are estimated using the blur kernel estimation method in.[3] Figures 4 and 5 depict the deblurring outputs of our algorithm in comparison with the results of the non-blind motion deblurring algorithms in.[3,10] As can be seen, our algorithm shows comparable results. Specifically, it is better able to deal with the ringing artifacts due to the noise and inaccuracy in blur kernel estimation. In Figure 6, the blur kernel is estimated using the method in[2] and the same blur kernel estimate is used for all algorithms. Also, note that in all experiments, we are using reflective boundary conditions.[26] These examples show the effectiveness of our algorithm for dealing with more complex real motion blur scenarios.



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

Figure 6. Real motion deblurring example: (a) input blurred noisy image, (b) Output of hyper-Laplacian algorithm,[10] (c) output of,[3] and (d) output of our algorithm ($\eta = 0.032, \beta = 0.6$).

4.3 COMPARISON WITH TRADITIONAL NORMALIZED GRAPH LAPLACIAN

In this section, we set up an experiment to investigate the performance of the proposed graph Laplacian with respect to that of the traditional normalized graph Laplacian. If we denote $W_D = D^{-1/2}KD^{-1/2}$, then the traditional normalized graph Laplacian would be $I - W_D$. We define the following cost function by replacing W with W_D in (4) as

$$E_D(\mathbf{z}) = (\mathbf{y} - A\mathbf{z})^T\{I + \beta(I - W_D)\}(\mathbf{y} - A\mathbf{z}) + \eta\ \mathbf{z}^T(I - W_D)\mathbf{z}. \tag{7}$$

The motion blurred Street image in Figure 7(b) is used for the experiments. The deblurring result from minimizing (7) related to traditional normalized graph Laplacian along with that of our proposed method are shown in Figure 7. As can be seen, our proposed normalized Laplacian produces deblurring output with higher quality within the same framework.

5. CONCLUSION

We have introduced a new framework based on a new definition of normalized graph Laplacian for non-blind motion deblurring. We have verified the performance of the proposed algorithm for both synthetic and real motion deblurring examples. Also, we have compared the properties and performance of the proposed normalized Laplacian with that of the traditional normalized graph Laplacian for motion deblurring. For future works, this framework can be extended to the blind motion deblurring problem. It also can be generalized to more complicated cases with non-uniform motion blur in the input image.

(a) (b)

(c) (d)

Figure 7. Motion deblurring examples with blurred noisy Street image by motion blur kernel in Figure (3) and additive white Gaussian noise with standard deviation $\sigma = 1$: (a) clean image, (b) blurred noisy image, (c) output of the deblurring algorithm with the corresponding traditional normalized Laplacian (PSNR = 27.93dB), and (d) output of our proposed deblurring algorithm (PSNR = 29.75dB).

REFERENCES

[1] Hansen, P. C., Nagy, J. G., and O'Leary, D. P., [*Deblurring Images: Matrices, Spectra, and Filtering*], SIAM, 1 ed. (2006).

[2] Cho, S. and Lee, S., "Fast motion deblurring," in [*ACM Transactions on Graphics (SIGGRAPH ASIA)*], , 145 (2009).

[3] Shan, Q., Jia, J., and Agarwala, A., "High-quality motion deblurring from a single image," *ACM Transactions on Graphics (SIGGRAPH)* **27**, 73:1–73:10 (Aug. 2008).

[4] Wang, Y., Yang, J., Yin, W., and Zhang, Y., "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Img. Sci.* **1**, 248–272 (Aug. 2008).

[5] Oliveira, J. P., Bioucas-Dias, J. M., and Figueiredo, M. A., "Adaptive total variation image deblurring: A majorization–minimization approach," *Signal Processing* **89**(9), 1683–1693 (2009).

[6] Peyré, G., Bougleux, S., and Cohen, L., "Non-local regularization of inverse problems," in [*Proc. 10th European Conference on Computer Vision (ECCV08)*], , 57–68 (Oct. 2008).

[7] Zhang, X., Burger, M., Bresson, X., and Osher, S., "Bregmanized nonlocal regularization for deconvolution and sparse reconstruction," *SIAM Journal on Imaging Sciences* **3**(3), 253–276 (2010).

[8] Yun, S. and Woo, H., "Linearized proximal alternating minimization algorithm for motion deblurring by nonlocal regularization," *Pattern Recognition* **44**(6), 1312–1326 (2011).

[9] Levin, A., Fergus, R., Durand, F., and Freeman, W. T., "Image and depth from a conventional camera with a coded aperture," *ACM Transactions on Graphics (SIGGRAPH)* **26**(3), 70 (2007).

[10] Krishnan, D. and Fergus, R., "Fast image deconvolution using hyper-laplacian priors," *NIPS* , 1033–1041 (2009).

[11] Joshi, N., Kang, S. B., Zitnick, C. L., and Szeliski, R., "Image deblurring using inertial measurement sensors," *ACM Transactions on Graphics (TOG)* **29**(4), 30 (2010).

[12] Yuan, L., Sun, J., Quan, L., and Shum, H.-Y., "Progressive inter-scale and intra-scale non-blind image deconvolution," in [*ACM Transactions on Graphics (SIGGRAPH)*], , 74 (2008).

[13] Cho, S., Wang, J., and Lee, S., "Handling outliers in non-blind image deconvolution," in [*IEEE International Conference on Computer Vision (ICCV), 2011*], , 495–502, IEEE (2011).

[14] Zhu, X., Šroubek, F., and Milanfar, P., "Deconvolving PSFs for a better motion deblurring using multiple images," in [*Computer Vision–ECCV 2012*], , 636–647, Springer (2012).

[15] Zhang, H., Wipf, D., and Zhang, Y., "Multi-image blind deblurring using a coupled adaptive sparse prior," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], , 1051–1058, IEEE (2013).

[16] Buades, A., Coll, B., and Morel, J. M., "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation* **4**, 490–530 (2005).

[17] Sinkhorn, R. and Knopp, P., "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific J. Math* **21**(2), 343–348 (1967).

[18] Knight, P. A. and Ruiz, D., "A fast algorithm for matrix balancing," *IMA Journal of Numerical Analysis* **33**, 1029–1047 (2013).

[19] Milanfar, P., "A tour of modern image filtering," *IEEE, Signal Processing Magazine* **30**(1), 106–128 (2013).

[20] Milanfar, P., "Symmetrizing smoothing filters," *SIAM Journal on Imaging Sciences* **6**(1), 263–284 (2013).

[21] Chung, F. R., [*Spectral graph theory*], vol. 92, American Mathematical Soc. (1997).

[22] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing* **16**, 2080 –2095 (Aug. 2007).

[23] Darbon, J., Cunha, A., Chan, T. F., Osher, S., and Jensen, G. J., "Fast nonlocal filtering applied to electron cryomicroscopy," in [*Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*], , 1331–1334, IEEE (2008).

[24] Gupta, A., Joshi, N., Zitnick, L., Cohen, M., and Curless, B., "Single image deblurring using motion density functions," in [*ECCV '10: Proceedings of the 10th European Conference on Computer Vision*], (2010).

[25] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E., "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on* **13**, 600 –612 (April 2004).

[26] Nagy, J. G., Palmer, K., and Perrone, L., "Iterative methods for image deblurring: a MATLAB object-oriented approach," *Numerical Algorithms* **36**(1), 73–93 (2004).