Appl. Comput. Harmon. Anal. 36 (2014) 326–334

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Letter to the Editor

Perturbation of the eigenvectors of the graph Laplacian: Application to image denoising

François G. Meyer [a,*], Xilin Shen [b]

[a] *Department of Electrical Engineering, University of Colorado at Boulder, Boulder, CO, United States*
[b] *Department of Diagnostic Radiology, Yale University, CT, United States*

ARTICLE INFO

Article history:
Received 2 February 2012
Received in revised form 13 June 2013
Accepted 14 June 2013
Available online 20 June 2013
Communicated by Naoki Saito

Keywords:
Graph Laplacian
Eigenvector perturbation
Random kernel matrix
Patch
Image denoising

ABSTRACT

Patch-based denoising algorithms currently provide the optimal techniques to restore an image. These algorithms denoise patches locally in “patch-space”. In contrast, we propose in this paper a simple method that uses the eigenvectors of the Laplacian of the patch-graph to denoise the image. Experiments demonstrate that our denoising algorithm outperforms the denoising gold-standards. We provide an analysis of the algorithm based on recent results on the perturbation of kernel matrices (El Karoui, 2010) [1,2], and theoretical analyses of patch denoising algorithms (Levin et al., 2012) [3], (Taylor and Meyer, 2012) [4].

1. Introduction

1.1. Problem statement and motivation

Recent work in computer vision and image processing indicates that the elusive quest for the “universal” transform has been replaced by a fresh new perspective. Indeed, researchers have recently proposed to represent images as “collage” of small patches. Patch-based denoising algorithms (e.g., [4–12], and references therein), which denoise patches locally in “patch-space”, provide currently the optimal techniques to restore an image. Early on, the computer graphics community has been implementing discrete versions of this idea in order to smooth and denoise surface meshes (e.g. [13], and references therein). More recently, similar ideas were proposed in [14,9,11,7]. Singer et al. [15] provided a stochastic interpretation of nonlocal means as a diffusion process.

Levin et al. [3] suggest that these local methods have reached their theoretical limit. While at large noise level the denoising can be improved by using larger patches, in practice only smooth patches benefit from being larger. Indeed, textured patches, which are always at a large mutual distance of one another, would require very large images from which to find neighbors [3]. Our recent work [4] arrives at the same conclusion, but tries to break these limits by organizing patches according to commute-time instead of Euclidean distance. In [4] we demonstrate the following unexpected fact: the low-frequency eigenvectors of the graph Laplacian are almost constant over the subset of textured patches, and vary slowly over the subset of smooth patches. As a result, the low-frequency eigenvectors are uncorrelated with the noise, and can be used to denoise patches. Inspired by this observation, and earlier work along the same vein [13,8,10], we propose in this paper

* Corresponding author.
E-mail address: fmeyer@colorado.edu (F.G. Meyer).
URL: http://ecee.colorado.edu/~fmeyer (F.G. Meyer).

1063-5203/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.acha.2013.06.004

a simple method that uses the eigenvectors of the Laplacian of the patch-graph to denoise the image. Experiments demonstrate that our denoising algorithm outperforms the denoising gold-standards. We provide an analysis of the algorithm based on recent results on the perturbation of kernel matrices [1,2], and theoretical analyses of patch denoising algorithms [3,4].

2. The patch-graph

We consider an image $u(\boldsymbol{x})$, of size $N \times N$. We extend the image by even symmetry when the pixel location $\boldsymbol{x} = (i, j)$ becomes close to the border of the image. We first define the notion of a *patch*.

Definition 1. Let $\boldsymbol{x}_n = (i, j)$ be a pixel with linear index $n = i \times N + j$. We extract a $\rho \times \rho$ block, centered about $\boldsymbol{x}_n$, where, without loss of generality, we take ρ to be an odd integer, and $\rho/2$ is the result of the Euclidean division of ρ by 2. By concatenating the columns, we identify the $\rho \times \rho$ matrix with a vector in $\mathbb{R}^{\rho^2}$, and we define the *patch* $\boldsymbol{u}(\boldsymbol{x}_n)$ by

$$\boldsymbol{u}(\boldsymbol{x}_n) = \begin{bmatrix} u_1(\boldsymbol{x}_n) \\ \vdots \\ u_{\rho^2}(\boldsymbol{x}_n) \end{bmatrix} = \begin{bmatrix} u(i-\rho/2, j-\rho/2) \\ \vdots \\ u(i+\rho/2, j+\rho/2) \end{bmatrix}. \tag{1}$$

As we collect all the patches, we form the *patch-set* in $\mathbb{R}^{\rho^2}$, $\mathcal{P} = \{\boldsymbol{u}(\boldsymbol{x}_n),\ n = 1, 2, \ldots, N^2\}$.

In order to study the discrete structure formed by the patch-set, we connect patches to their nearest neighbors, and construct a graph that we call the *patch-graph*.

Definition 2. The *patch-graph*, Γ, is a weighted graph defined as follows.

1. The vertices of Γ are the N^2 patches $\boldsymbol{u}(\boldsymbol{x}_n)$, $n = 1, \ldots, N^2$.
2. Each vertex $\boldsymbol{u}(\boldsymbol{x}_n)$ is connected to its ν nearest neighbors using the metric

$$d(n,m) = \left\| \boldsymbol{u}(\boldsymbol{x}_n) - \boldsymbol{u}(\boldsymbol{x}_m) \right\| + \beta \|\boldsymbol{x}_n - \boldsymbol{x}_m\|. \tag{2}$$

3. The weight $w_{n,m}$ along the edge $\{\boldsymbol{u}(\boldsymbol{x}_n), \boldsymbol{u}(\boldsymbol{x}_m)\}$ is given by

$$w_{n,m} = \begin{cases} e^{-d^2(n,m)/\delta^2} & \text{if } \boldsymbol{x}_n \text{ is connected to } \boldsymbol{x}_m, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The parameter $\beta \geqslant 0$ controls the influence of the penalty term that measures the distance between the patches in the image domain. The distance $d(n,m)$ is small if the image intensity is similar at $\boldsymbol{x}_n$ and $\boldsymbol{x}_m$, $\boldsymbol{u}(\boldsymbol{x}_n) \approx \boldsymbol{u}(\boldsymbol{x}_m)$, and $\boldsymbol{x}_n$ and $\boldsymbol{x}_m$ are not too far from one another $\|\boldsymbol{x}_n - \boldsymbol{x}_m\| \approx 0$. The parameter δ controls the scaling of the similarity $d(n,m)$ between $\boldsymbol{u}(\boldsymbol{x}_n)$ and $\boldsymbol{u}(\boldsymbol{x}_m)$ when defining the edge weight $w_{n,m}$. The particular form of the weight (3) ensures that $w_{n,m}$ drops rapidly to zero as $d(n,m)$ becomes larger than δ. A very large δ (i.e., $w_{n,m} \approx 1$ if n, m are connected) emphasizes the topology of the graph and promotes a very fast diffusion of the random walk through the patch-set. The other alternative: $\delta \approx 0$ accentuates the difference between the patches, but is very sensitive to noise. We now define the *weight matrix*, which fully characterizes the patch-graph.

Definition 3. The *weight matrix* $\boldsymbol{W}$ is the $N^2 \times N^2$ symmetric matrix with entries $\boldsymbol{W}_{n,m} = w_{n,m}$. The *degree matrix* is the $N^2 \times N^2$ diagonal matrix $\boldsymbol{D}$ with entries $\boldsymbol{D}_{n,n} = \sum_{m=1}^{N^2} w_{n,m}$. Finally, the *normalized Laplacian matrix* $\boldsymbol{L}$ is the $N^2 \times N^2$ symmetric matrix defined by

$$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}^{-\frac{1}{2}}. \tag{4}$$

The matrix $\boldsymbol{L}$ is symmetric and positive definite, and has N^2 eigenvectors $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{N^2}$ with corresponding eigenvalues $\lambda_1 = 0 < \lambda_2 \leqslant \cdots \leqslant \lambda_{N^2}$. Each eigenvector $\boldsymbol{\phi}_k$ is a vector with N^2 components, one for each vertex of the graph. Hence, we write

$$\boldsymbol{\phi}_k = \begin{bmatrix} \boldsymbol{\phi}_k(\boldsymbol{x}_1) & \ldots & \boldsymbol{\phi}_k(\boldsymbol{x}_{N^2}) \end{bmatrix}^T,$$

Algorithm: Two-pass denoising

```
Input: noisy image $\tilde{u}$;
• number of nearest neighbors $\nu$; patch sizes $\rho_1, \rho_2$; scale parameters $\delta_1, \delta_2$;
• number of eigenvectors $K_1, K_2$; $\gamma \in [0, 1]$

Stage 1
1. build the graph from the image $\tilde{u}$ and compute $\widetilde{\boldsymbol{W}}$ and $\widetilde{\boldsymbol{D}}$;
2. compute the first $K_1$ eigenvectors, $\tilde{\boldsymbol{\phi}}_k$, $k = 1, \dots, K_1$, of $\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{W}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}}$;
3. construct the nonlinear estimator $\hat{\boldsymbol{u}}^{(1)}(\boldsymbol{x})$ of the patch-set using (6);
4. reconstruct the image $\hat{u}^{(1)}$ by averaging the patches $\hat{\boldsymbol{u}}^{(1)}(\boldsymbol{x})$ using (8);
5. compute the denoised image $\hat{u}^{(2)} = (1 - \gamma)\hat{u}^{(1)} + \gamma \tilde{u}$

Stage 2
1. build a second graph from the image $\hat{u}^{(2)}$; compute $\widehat{\boldsymbol{W}}^{(2)}$ and $\widehat{\boldsymbol{D}}^{(2)}$;
2. compute the first $K_2$ eigenvectors, $\hat{\boldsymbol{\phi}}_k, k = 1, \dots, K_2$, of $[\widehat{\boldsymbol{D}}^{(2)}]^{-\frac{1}{2}} \widehat{\boldsymbol{W}}^{(2)} [\widehat{\boldsymbol{D}}^{(2)}]^{-\frac{1}{2}}$;
3. construct the nonlinear estimator $\hat{\boldsymbol{u}}^{(3)}(\boldsymbol{x})$ of the patch-set using (6);
4. reconstruct the image $\hat{u}^{(3)}$ by averaging the patches $\hat{\boldsymbol{u}}^{(3)}(\boldsymbol{x})$ using (8)

Output: the denoised image $\hat{u}^{(3)}$.
```

Fig. 1. The two-stage denoising algorithm.

to emphasize the fact that we consider $\boldsymbol{\phi}_k$ to be a function sampled on the vertices of Γ. The eigenvectors $\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_{N^2}$ form an orthonormal basis for functions defined on the graph, where the inner-product on the graph is defined by

$$\langle f, g \rangle = \sum_{j=1}^{N^2} f(j) g(j).$$

3. Denoising

3.1. Iterative denoising

We describe in this section a denoising algorithm that constructs an estimate of the patch-set from the knowledge of the noisy image $\tilde{u}$. The denoising of the patches is not performed locally on the patch-set, but rather relies on the global eigenvectors $\boldsymbol{\phi}_k$.

The original image u has been corrupted by additive white Gaussian noise with variance σ^2, and we measure at every pixel $\boldsymbol{x}$ the noisy image $\tilde{u}(\boldsymbol{x})$, given by

$$\tilde{u}(\boldsymbol{x}) = u(\boldsymbol{x}) + \eta(\boldsymbol{x}).$$

Let $\tilde{\boldsymbol{u}}(\boldsymbol{x}_n)$ be the patch centered at $\boldsymbol{x}_n$ associated with the noisy image $\tilde{u}$, and let $\widetilde{\mathcal{P}}$ be the patch-set formed by the collection of noisy patches. We propose to construct an estimate $\widehat{\mathcal{P}}$ of the clean patch-set $\mathcal{P}$ from the noisy patch-set $\widetilde{\mathcal{P}}$. Lastly, we will combine several denoised patches $\hat{\boldsymbol{u}}(\boldsymbol{x}_m)$ to reconstruct an estimate $\hat{u}(\boldsymbol{x}_n)$ of the clean image at the pixel $\boldsymbol{x}_n$.

Let $\widetilde{\Gamma}$ be the graph associated with the noisy patch-set $\widetilde{\mathcal{P}}$. We define $\widetilde{\boldsymbol{W}}$ and $\widetilde{\boldsymbol{L}}$ to be the weight and Laplacian matrices computed from the noisy image, respectively. Finally, let $\tilde{\boldsymbol{\phi}}_k$ be the eigenvectors of $\widetilde{\boldsymbol{L}}$. As explained in the next section, the low-frequency eigenvectors $\tilde{\boldsymbol{\phi}}_k$ remain stable, but the higher frequency eigenvectors rapidly degrade. As a result, $\tilde{\boldsymbol{\phi}}_k$ can only be used to reconstruct a coarse estimate, $\hat{u}^{(1)}$, of a lowpass version of the original image. The details of the reconstruction are provided in the next section.

Because too few eigenvectors are used to compute $\hat{u}^{(1)}$, we add back a scaled version of the residual error (the difference between the noisy image $\tilde{u}$ and the denoised image $\hat{u}^{(1)}$) to obtain the intermediate image $\hat{u}^{(2)}$ defined by

$$\hat{u}^{(2)} = \hat{u}^{(1)} + \gamma\big(\tilde{u} - \hat{u}^{(1)}\big) = (1 - \gamma)\hat{u}^{(1)} + \gamma \tilde{u}, \tag{5}$$

where $\gamma \in (0, 1)$. At this point the image $\hat{u}^{(2)}$ has been sufficiently denoised, and we construct a new graph. We finally compute the eigenvectors $\hat{\boldsymbol{\phi}}_k$ of $\widehat{\boldsymbol{L}}^{(2)}$ associated to the image $\hat{u}^{(2)}$. These eigenvectors are used to denoise the original image (see Fig. 1).

Szlam et al. [8] proposed a related approach to bootstrap a semi-supervised classifier: a new graph is constructed after one iteration of the classifier to improve the classification performance.

3.2. The details: one pass of denoising

Let $\tilde{u}_1(\boldsymbol{x}_n), \dots, \tilde{u}_{\rho^2}(\boldsymbol{x}_n)$ be the ρ^2 coordinates of the patch $\tilde{\boldsymbol{u}}(\boldsymbol{x}_n)$. Let $\tilde{\boldsymbol{\phi}}_k$ be the set of eigenvectors computed during the first or second iteration. We denoise each coordinate function $\tilde{u}_j$ independently. This is achieved by expanding the function $\tilde{u}_j$ into the basis formed by the $\tilde{\boldsymbol{\phi}}_k$, and performing a nonlinear thresholding. The result is the denoised function $\hat{u}_j$, given by

$$\hat{u}_j = \sum_{k=1}^{N^2} \kappa\big(\langle \tilde{u}_j, \tilde{\boldsymbol{\phi}}_k\rangle\big)\tilde{\boldsymbol{\phi}}_k, \tag{6}$$

where the function κ performs a nonlinear thresholding of the coefficients $\langle \tilde{u}_j, \tilde{\boldsymbol{\phi}}_k\rangle$. In practice, we use a hard thresholding, i.e., only using the first K_1, K_2 eigenvectors as indicated in Fig. 1. After denoising all the coordinate functions $\hat{u}_1, \dots, \hat{u}_{\rho^2}$, we have access to an estimate $\{\hat{\boldsymbol{u}}(\boldsymbol{x}_n),\ n = 1, \dots, N^2\}$ of the clean patch-set.

Because the patch-set has a maximum overlap (i.e., we have a patch for each pixel), there are ρ^2 patches that overlap (in the image domain) with the pixel $\boldsymbol{x}_n$. Let $N(\boldsymbol{x}_n)$ be the ball for the l_∞ norm of radius $\rho/2$ pixels centered at $\boldsymbol{x}_n$,

$$N(\boldsymbol{x}_n) = \big\{\boldsymbol{x}_l\colon\ \|\boldsymbol{x}_n - \boldsymbol{x}_l\|_\infty \leqslant \rho/2\big\}. \tag{7}$$

The ρ^2 patches $\boldsymbol{u}(\boldsymbol{x}_l)$ that overlap with $\boldsymbol{x}_n$ all have their center $\boldsymbol{x}_l$ in $N(\boldsymbol{x}_n)$. For each of the pixels $\boldsymbol{x}_l \in N(\boldsymbol{x}_n)$, there exists an index i_n such that the coordinate i_n of the patch $\hat{\boldsymbol{u}}(\boldsymbol{x}_l)$ corresponds to the pixel location $\boldsymbol{x}_n$. In other words, $[\hat{\boldsymbol{u}}(\boldsymbol{x}_l)]_{i_n}$ is an estimate of the image intensity at the location $\boldsymbol{x}_n$ obtained from patch $\hat{\boldsymbol{u}}(\boldsymbol{x}_l)$. We combine all these estimates using a moving least squares approximation given by Shepard's method [16], and we define the denoised image at pixel $\boldsymbol{x}_n$ by

$$\hat{u}(\boldsymbol{x}_n) = \sum_{\boldsymbol{x}_l \in N(\boldsymbol{x}_n)} \alpha(\boldsymbol{x}_n, \boldsymbol{x}_l)\ \big[\hat{\boldsymbol{u}}(\boldsymbol{x}_l)\big]_{i_n}, \tag{8}$$

where the exponential weight $\alpha(\boldsymbol{x}_n, \boldsymbol{x}_l)$ is given by

$$\alpha(\boldsymbol{x}_n, \boldsymbol{x}_l) = \frac{\exp\left(-\|\boldsymbol{x}_n - \boldsymbol{x}_l\|^2\right)}{\sum_{\boldsymbol{x}_m \in N(\boldsymbol{x}_n)} \exp\left(-\|\boldsymbol{x}_n - \boldsymbol{x}_m\|^2\right)}. \tag{9}$$

We note that we can interpret the jth coordinate function u_j as the original image u shifted by (p, q), where $j = \rho(p + \rho/2) + q + \rho/2 + 1$. The reconstruction formula (8) can then be interpreted as a translation-invariant denoising procedure.

The algorithm that implements the two passes of denoising is described in Fig. 1.

4. Analysis of the algorithm

4.1. Perturbation of the weight matrix

We first analyze the effect of the noise on the weight matrix $\boldsymbol{W}$. Each patch $\boldsymbol{u}(\boldsymbol{x}_n)$ of the clean image is corrupted by a multivariate Gaussian noise $\boldsymbol{\eta}(\boldsymbol{x}_n)$ with diagonal covariance matrix, $\sigma^2 I_{\rho^2}$,

$$\tilde{\boldsymbol{u}}(\boldsymbol{x}_n) = \boldsymbol{u}(\boldsymbol{x}_n) + \boldsymbol{\eta}(\boldsymbol{x}_n). \tag{10}$$

The squared distance between any two noisy patches $\tilde{\boldsymbol{u}}(\boldsymbol{x}_n)$ and $\tilde{\boldsymbol{u}}(\boldsymbol{x}_m)$ is given by

$$\big\|\tilde{\boldsymbol{u}}(\boldsymbol{x}_n) - \tilde{\boldsymbol{u}}(\boldsymbol{x}_m)\big\|^2 = \big\|\boldsymbol{u}(\boldsymbol{x}_n) - \boldsymbol{u}(\boldsymbol{x}_m)\big\|^2 + 2\big\langle \boldsymbol{u}(\boldsymbol{x}_n) - \boldsymbol{u}(\boldsymbol{x}_m), \boldsymbol{\eta}(\boldsymbol{x}_n) - \boldsymbol{\eta}(\boldsymbol{x}_m)\big\rangle + \big\|\boldsymbol{\eta}(\boldsymbol{x}_n) - \boldsymbol{\eta}(\boldsymbol{x}_m)\big\|^2.$$

We note that the vector $\boldsymbol{\eta}(\boldsymbol{x}_n) - \boldsymbol{\eta}(\boldsymbol{x}_m)$ is a Gaussian with covariance matrix $2\sigma^2 I_{\rho^2}$. If the patch dimension ρ^2 is large, then by concentration of the Gaussian measure, the norm squared, $\|\boldsymbol{\eta}(\boldsymbol{x}_n) - \boldsymbol{\eta}(\boldsymbol{x}_m)\|^2$, is concentrated around its mean $2\sigma^2\rho^2$, with high probability. Similarly the projection of the noise $\langle \boldsymbol{u}(\boldsymbol{x}_n) - \boldsymbol{u}(\boldsymbol{x}_m), \boldsymbol{\eta}(\boldsymbol{x}_n) - \boldsymbol{\eta}(\boldsymbol{x}_m)\rangle$ on the deterministic vector $\boldsymbol{u}(\boldsymbol{x}_n) - \boldsymbol{u}(\boldsymbol{x}_m)$ is concentrated around 0 with high probability. This argument was recently made formal in [2], where the author provides a theoretical analysis of the weight matrix when the data are perturbed by Gaussian noise of small magnitude. Using our notation, El Karoui [2] assumes that

$$\tilde{\boldsymbol{u}}(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x}) + \frac{1}{\rho}\boldsymbol{\eta}(\boldsymbol{x}),$$

where $\boldsymbol{\eta}(\boldsymbol{x})$ is multivariate Gaussian noise $\boldsymbol{\eta}(\boldsymbol{x}_n)$ with diagonal covariance matrix, $\sigma^2 I_{\rho^2}$. In other words the norm of the noise does not increase with the size of the patch. In this regime, El Karoui [2] shows that the perturbed weight matrix $\widetilde{\boldsymbol{W}}$ computed from $\tilde{\boldsymbol{u}}(\boldsymbol{x})$ is simply a rescaled version of the original weight, in the case of Gaussian kernels,

$$\tilde{w}_{n,m} = e^{-2\sigma^2\rho^2/\delta^2} \exp\left(-\big\|\boldsymbol{u}(\boldsymbol{x}_n) - \boldsymbol{u}(\boldsymbol{x}_m)\big\|/\delta^2\right) = e^{-2(\sigma\rho/\delta)^2} w_{n,m}$$

with high probability. Unfortunately, this work cannot be extended to our model where the norm of the noise grows with the patch size. In fact, our experiments demonstrate that at large noise level we observe a phenomenon of crossover whereby eigenvectors associated with eigenvalues that are smaller than the largest eigenvalues of the noise cannot be estimated.

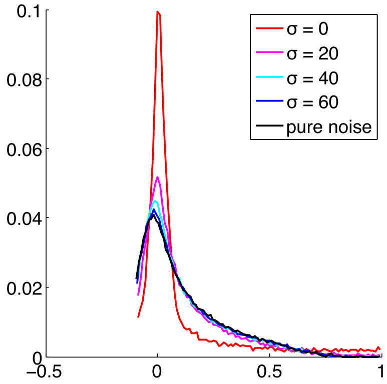

Fig. 2. Eigenvalue distribution for the image Lena for different noise levels. The black curve corresponds to the limiting case of pure noise (no image); the distribution approaches the Marčenko–Pastur distribution.

4.2. *Noise saturation: the crossover phenomenon*

We observe that as the noise level increases, the number of reliable eigenvectors $\tilde{\boldsymbol{\phi}}_k$ becomes smaller (see Fig. 3). Visually, the higher order $\tilde{\boldsymbol{\phi}}_k$ start tracking the noise in the image. This experimental observation can be explained using the following derivation.

First, we observe that when the signal to noise is very low, the spectrum of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ approaches the spectrum of a random kernel matrix (see Fig. 2). We know that the distribution of this spectrum is well approximated by the spectrum of a regular inner-product (Gram) matrix [1,17] (the approximation becomes better as the dimension of the patches increases). In addition, we also know that the spectrum of an inner-product matrix is given by the Marčenko–Pastur distribution (see the black curve in Fig. 2).

Finally, we know [18] that the angle between the true eigenvector $\boldsymbol{\phi}_k$ and the perturbed eigenvector $\tilde{\boldsymbol{\phi}}_k$ (with the corresponding eigenvalue $\tilde{\lambda}_k$) is bounded by

$$\sin\angle(\boldsymbol{\phi}_k, \tilde{\boldsymbol{\phi}}_k) \leqslant \frac{\|\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{W}}\widetilde{\boldsymbol{D}}^{-1/2} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}\|_2}{\min_{\mu\in\Lambda(\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2})}|\tilde{\lambda}_k - \mu|}, \tag{11}$$

where $\Lambda(\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2})$ is the spectrum of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$. As demonstrated in [19], if an eigenvalue $\tilde{\lambda}_k$ becomes smaller than the largest eigenvalue of the noise, then the corresponding perturbed eigenvector $\tilde{\boldsymbol{\phi}}_k$ is no longer aligned with the original eigenvector $\boldsymbol{\phi}_k$, and the upper bound (11) blows up.

Putting everything together, we conclude that the fact that the eigenvectors $\tilde{\boldsymbol{\phi}}_k$ become more unreliable more early, as the noise level increases, is simply a manifestation of the "crossover" phenomenon for a kernel version of PCA.

4.3. *The geometry of the patch-set and the shape of the eigenvectors*

As explained by Taylor and Meyer [4], patches extracted from smooth regions of the image align themselves along smooth low-dimensional structures, and patches that contain large gradients and texture are usually scattered across $\mathbb{R}^{\rho^2}$ [4]. Other researchers have made similar experimental observations [12,3]. In [4], we demonstrate the following surprising fact: the low-frequency eigenvectors of the graph Laplacian are almost constant over the subset of textured patches, and vary slowly over the subset of smooth patches. Consequently, we claim that for small values of k, we have

$$\langle\tilde{u}, \tilde{\boldsymbol{\phi}}_k\rangle \approx \langle u, \boldsymbol{\phi}_k\rangle. \tag{12}$$

Indeed,

$$\langle\tilde{u}, \boldsymbol{\phi}_k\rangle = \sum_{\boldsymbol{x}\in\text{smooth patches}} \widetilde{u(\boldsymbol{x})}\boldsymbol{\phi}_k(\boldsymbol{x}) + \sum_{\boldsymbol{x}\in\text{textured patches}} \widetilde{u(\boldsymbol{x})}\boldsymbol{\phi}_k(\boldsymbol{x}).$$

Now if $\tilde{\boldsymbol{\phi}}_k$ is constant over the textured patches, then we have

$$\begin{aligned}\sum_{\boldsymbol{x}\in\text{textured patches}} \widetilde{u(\boldsymbol{x})}\boldsymbol{\phi}_k(\boldsymbol{x}) &= \boldsymbol{\phi}_k(\text{textured patches})\Bigg(\sum_{\boldsymbol{x}\in\text{textured patches}} u(\boldsymbol{x}) + \eta(\boldsymbol{x})\Bigg)\\ &\approx \boldsymbol{\phi}_k(\text{textured patches}) \sum_{\boldsymbol{x}\in\text{textured patches}} u(\boldsymbol{x}),\end{aligned}$$

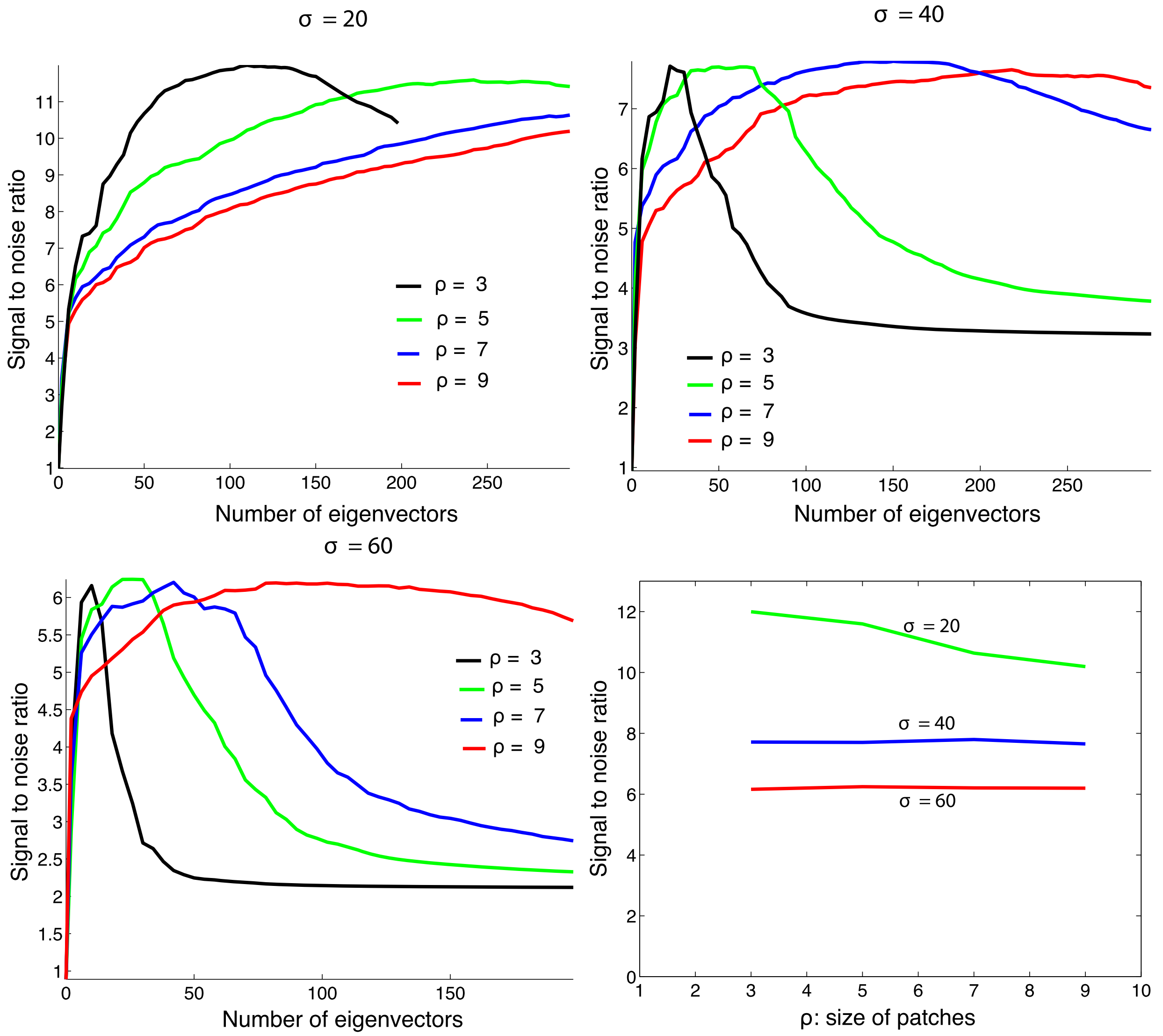

Fig. 3. Top to bottom, left to right: SNR as a function of the number of eigenvectors K after a one pass reconstruction, for increasing noise levels. Each curve corresponds to a different patch size. Bottom right: optimal reconstruction error as a function of the patch size ρ at different noise levels.

where the last approximate equality holds because the noise is white and the empirical mean of the noise over the textured patches is close to zero. For small values of k, the eigenvector $\boldsymbol{\phi}_k$ oscillates very slowly on the smooth patch, and therefore,

$$\sum_{\boldsymbol{x}\in\text{smooth patches}} \widetilde{u(\boldsymbol{x})}\boldsymbol{\phi}_k(\boldsymbol{x}) \approx \boldsymbol{\phi}_k(\text{smooth patches}) \sum_{\boldsymbol{x}\in\text{smooth patches}} \widetilde{u(\boldsymbol{x})} \approx \boldsymbol{\phi}_k(\text{smooth patches}) \sum_{\boldsymbol{x}\in\text{smooth patches}} u(\boldsymbol{x}).$$

Putting everything together we get

$$\langle \tilde{u}, \boldsymbol{\phi}_k\rangle \approx \boldsymbol{\phi}_k(\text{textured patches}) \sum_{\boldsymbol{x}\in\text{textured patches}} u(\boldsymbol{x}) + \boldsymbol{\phi}_k(\text{smooth patches}) \sum_{\boldsymbol{x}\in\text{smooth patches}} u(\boldsymbol{x}) \approx \langle u, \boldsymbol{\phi}_k\rangle.$$

We conclude that since the perturbed eigenvectors $\tilde{\boldsymbol{\phi}}_k$ are very similar to the original eigenvectors $\boldsymbol{\phi}_k$ for the small values of k, we have

$$\sum_{k=1}^{K} \langle \tilde{u}, \tilde{\boldsymbol{\phi}}_k\rangle \tilde{\boldsymbol{\phi}}_k \approx \sum_{k=1}^{K} \langle u, \boldsymbol{\phi}_k\rangle \boldsymbol{\phi}_k, \tag{13}$$

as long as K is small. In other words, the partial reconstruction of the low frequencies part of the image can be recovered by projecting the noisy image on the perturbed eigenvectors. This provides some justification for the first pass of the algorithm.

Table 1
Peak signal-to-noise ratio (larger is better).

	Airplane	Barbara	Boats	Camera	Clown	Couple	Fingerprint
Laplacian	**25.50**	**28.15**	**26.36**	**26.86**	**25.41**	**25.30**	**23.09**
BM3D	24.83	27.38	26.04	26.20	25.11	25.28	21.55

	Gold hill	House	Lenna	Man	Mandrill	Peppers	Pentagon	Roof
Laplacian	**26.03**	**29.13**	**25.64**	**25.68**	**23.20**	**29.36**	**26.08**	**25.92**
BM3D	25.84	28.60	25.28	25.49	23.03	28.77	25.45	24.87

5. Optimization of the parameters of the algorithm

In this section we study in detail the two most important parameters of the algorithm: the patch sizes (ρ_1 and ρ_2) and the numbers of eigenvectors (K_1 and K_2) that are used during the two stages of the algorithm. Our findings are consistent with some recent studies [3,12].

5.1. The patch size

While the eigenvectors $\boldsymbol{\phi}_k$ provide a global basis on the patch-set, the patch size introduces a notion of local scale in the image. We notice visually that as the patch size increases, the eigenvectors become less crisp and more blurred. As explained in [15,4] when the patch size is larger, patches are at a larger distance of one another in the patch-set. Consequently, a larger patch size prevents patches that are initially different from accidentally becoming neighbors after the image is corrupted by noise, since their mutual distance is relatively less perturbed by the noise. Because the eigenvectors $\tilde{\boldsymbol{\phi}}_k$ are sensitive to the geometry of the patch-set, larger patches help minimize the perturbation of the eigenvectors. We note that, as patches become larger, the eigenvectors become more blurred, therefore requiring more eigenvectors to describe the texture. Finally, we need to be aware of the following practical limitation: if the patch size is large, and the image size is relatively small, then we have few patches to estimate the geometry of the patch-set; a problem aggravated by the fact that the patch-set lives in a large ambient space in that case.

We have compared the effect of the patch size on the quality of the denoised image $\hat{u}^{(2)}$ after the first pass (stage 1) of the denoising algorithm (see Fig. 1). We define the signal-to-noise ratio (SNR) by $\text{SNR} = \|u\|/\|u - \hat{u}\|$. Fig. 3 displays the signal-to-noise ratio as a function of the number of eigenvectors K used to reconstruct $\hat{u}^{(2)}$, for different patch sizes ρ, and at different noise levels ($\sigma = 20, 40$, and 60). For low noise levels ($\sigma = 20$) the SNR is larger for smaller patches. For moderate noise levels, the maximum SNR achieved with different patch size is approximately the same (see Fig. 3 bottom right). Nevertheless, the visual quality of the denoised images – at the same SNR – are quite different: with larger ρ, the denoised images are much smoother. The choice of the patch size ρ is in general based on the noise level σ. In practice, we choose $\rho = 3$ for $\sigma = 20$, $\rho = 5$ for $\sigma = 40$, and $\rho = 7$ for $\sigma = 60$.

5.2. The number of eigenvectors K

At both stages of the algorithm, the first K eigenvectors are used to compute a denoised image. In general, the eigenvectors degrade faster as the variance of the noise level increases, a manifestation of the crossover phenomenon.

6. Experiments

We implemented the two-stage denoising algorithm (see Fig. 1) and evaluated its performance against the BM3D algorithm [11]. The parameters of BM3D used in the experiments are those that are recommended by the authors in [11], which require the knowledge of the noise level. The evaluation was performed using 15 images (see Fig. 4) taken from the BM3D website http://www.cs.tut.fi/~foi/GCF-BM3D/. White Gaussian noise with $\sigma = 40$ was added to the images. We report in Table 1 the average peak signal-to-noise ratio ($20 \log_{10}(255/\|u - \hat{u}\|)$) computed over 40 experiments associated with independent realizations of the random noise. Fig. 4 displays two images where our algorithm significantly outperforms BM3D both in terms of visual quality and PSNR. The residual images (computed as the difference between the original clean image and the reconstructed image) associated with our approach contains less information (edges, texture, etc.) from the original image than with BM3D, and therefore the residual image can be interpreted as noise. Additional results for the other images can be found on the supplementary material. For all experiments, we used patch sizes combinations (ρ_1, ρ_2) that were either (7, 5) or (5, 3). Overall, our algorithm provides a moderate improvement in terms of PSNR (see Table 1). Images with regular periodic patterns, such as fingerprints and Barbara resulted in the greatest improvements over BM3D, both visually, and in terms of PSNR. Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.acha.2013.06.004.

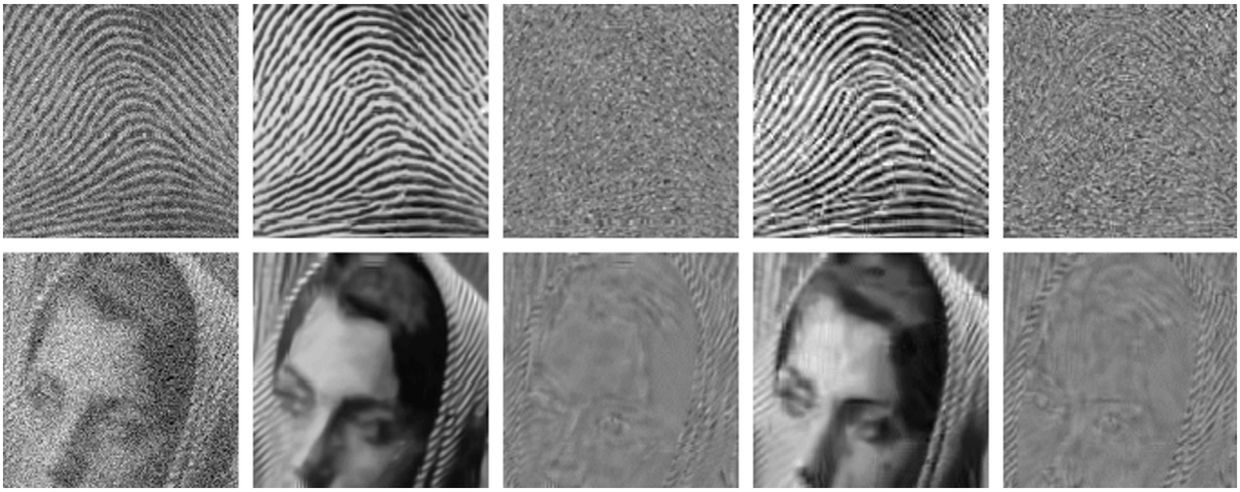

Fig. 4. From left to right for both rows: noisy image ($\sigma = 40$), denoised image and residual image with our approach; denoised image and residual image with BM3D. The residual errors are all displayed using the range $[-128, 128]$.

7. Discussion

We proposed a two-stage algorithm to estimate a denoised set of patches from a noisy image. The algorithm relies on the following two observations: (1) the low-index eigenvectors of the diffusion, or Laplacian, operators are very robust to random perturbations of the weights and random changes in the connections of the graph; and (2) patches extracted from smooth regions of the image are organized along smooth low-dimensional structures of the patch-set, and therefore can be reconstructed with few eigenvectors. Experiments demonstrate that our very simple denoising algorithm outperforms the denoising gold-standards.

7.1. *Fast computation of the eigenvectors*

A key tenet of this work is our ability to compute the eigenvectors of the sparse matrix $\boldsymbol{L}$, which has size $N^2 \times N^2$ but with only ν nonzero entries on each row. For the experiments we used the restarted Arnoldi method for sparse matrices implemented by the Matlab function `eigs` to solve the eigenvalue problem.

There are several options for further speeding up the computation of the eigenvectors. Saito [20] proposes to modify the eigenvalue problem, and compute the eigenvectors via the integral operator that commutes with the Laplacian. This approach leads to fast algorithms. The recent work of Kushnir et al. [21] indicates that multigrid methods yield an immediate improvement over Krylov subspace projection methods (e.g. Arnoldi's method).

Acknowledgments

We thank the reviewers and the editor for comments that helped improve the paper. F.G.M. was partially supported by National Science Foundation Grants DMS 0941476, ECS 0501578, and DOE award DE-SCOO04096. This work benefited from fruitful discussions with Raphy Coifman, Peter Jones, Arthur Szlam, and Amit Singer.

References

[1] N. El Karoui, The spectrum of kernel random matrices, Ann. Statist. 38 (2010) 1–50.
[2] N. El Karoui, On information plus noise kernel random matrices, Ann. Statist. 38 (2010) 3191–3216.
[3] A. Levin, B. Nadler, F. Durand, W. Freeman, Patch complexity, finite pixel correlations and optimal denoising, in: Computer Vision – ECCV 2012, vol. 7576, 2012, pp. 73–86.
[4] K. Taylor, F. Meyer, A random walk on image patches, SIAM J. Imaging Sci. 5 (2) (2012) 688–725.
[5] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Image Process. 54 (2006) 4311.
[6] A. Buades, B. Coll, J.M. Morel, A review of image denoising algorithms, with a new one, Multiscale Model. Simul. 4 (2005) 490–530.
[7] G. Gilboa, S. Osher, Nonlocal operators with applications to image processing, Multiscale Model. Simul. 7 (3) (2008) 1005–1028.
[8] A. Szlam, M. Maggioni, R. Coifman, Regularization on graphs with function-adapted diffusion processes, J. Mach. Learn. Res. 9 (2008) 1711–1739.
[9] S. Bougleux, A. Elmoataz, M. Melkemi, Local and nonlocal discrete regularization on weighted graphs for image and mesh processing, Int. J. Comput. Vis. 84 (2009) 220–236.
[10] G. Peyré, Image processing with non-local spectral bases, Multiscale Model. Simul. 7 (2008) 703–730.
[11] V. Katkovnik, A. Foi, K. Egiazarian, J. Astola, From local kernel to nonlocal multiple-model image denoising, Int. J. Comput. Vis. 86 (2010) 1–32.
[12] M. Zontak, M. Irani, Internal statistics of a single natural image, in: IEEE Conf. CVPR, 2011, pp. 977–984.
[13] G. Taubin, A signal processing approach to fair surface design, in: Comp. Graph., ACM, 1995, pp. 351–358.

[14] M. Hein, M. Maier, Manifold denoising, in: Adv. Neur. In., 2007, pp. 561–568.
[15] A. Singer, Y. Shkolnisky, B. Nadler, Diffusion interpretation of nonlocal neighborhood filters for signal denoising, SIAM J. Imaging Sci. 2 (2009) 118–139.
[16] G. Fasshauer, Meshfree Approximation Methods with MATLAB, Interdiscip. Math. Sci., World Scientific, 2007.
[17] X. Cheng, A. Singer, The spectrum of random inner-product kernel matrices, arXiv:1202.3155, 2012.
[18] G. Stewart, J. Sun, Matrix Perturbation Theory, Academic Press, 1990.
[19] B. Nadler, Finite sample approximation results for principal component analysis: A matrix perturbation approach, Ann. Statist. 36 (2008) 2791–2817.
[20] N. Saito, Data analysis and representation on a general domain using eigenfunctions of Laplacian, Appl. Comput. Harmon. Anal. 25 (2008) 68–97.
[21] D. Kushnir, M. Galun, A. Brandt, Efficient multilevel eigensolvers with applications to data analysis tasks, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1377–1391.