# Redefining Self-Similarity in Natural Images for Denoising Using Graph Signal Gradient

Jiahao Pang*, Gene Cheung†, Wei Hu* and Oscar C. Au*
*The Hong Kong University of Science and Technology, Hong Kong, China
E-mail: {jpang, huwei, eeau}@ust.hk Tel: +852 2358-7053
†National Institute of Informatics, Tokyo, Japan
E-mail: cheung@nii.ac.jp Tel: +81-3-4212-2567

***Abstract*—Image denoising is the most basic inverse imaging problem. As an under-determined problem, appropriate definition of image priors to regularize the problem is crucial. Among recent proposed priors for image denoising are: i) graph Laplacian regularizer where a given pixel patch is assumed to be smooth in the graph-signal domain; and ii) self-similarity prior where image patches are assumed to recur throughout a natural image in non-local spatial regions. In our first contribution, we demonstrate that the graph Laplacian regularizer converges to a continuous time functional counterpart, and careful selection of its features can lead to a discriminant signal prior. In our second contribution, we redefine patch self-similarity in terms of patch gradients and argue that the new definition results in a more accurate estimate of the graph Laplacian matrix, and thus better image denoising performance. Experiments show that our designed algorithm based on graph Laplacian regularizer and gradient-based self-similarity can outperform non-local means (NLM) denoising by up to 1.4 dB in PSNR.**

## I. Introduction

Image denoising [1] is the most basic inverse imaging problem with a simple image model:

$$\mathbf{y} = \mathbf{x} + \mathbf{e} \tag{1}$$

where $\mathbf{y}$ is the observed noise-corrupted image, $\mathbf{x}$ is the desired noiseless image, and $\mathbf{e}$ is the additive noise. It is inherently an under-determined problem, and as such, appropriate definition of one or more image priors to regularize the problem into a well posed one is very important; *i.e.*,

$$\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{x}\|_2^2 \; + \; \lambda \; \text{prior}(\mathbf{x}), \tag{2}$$

where $\|\mathbf{y}-\mathbf{x}\|_2^2$ is a *fidelity term* requiring the desired signal $\mathbf{x}$ to be close to observation $\mathbf{y}$, and $\text{prior}(\mathbf{x})$ expresses *a priori* knowledge about the nature or characteristics of sought-after signal $\mathbf{x}$. Parameter $\lambda$ trades off the importance of the fidelity term with prior knowledge of $\mathbf{x}$.

Among proposed image priors in the literature such as total variation (TV) [2] and auto-regressive prior [3], one recent popular prior is the *graph Laplacian regularizer* [4], [5], [6]. Leveraging on recent advance in graph signal processing (GSP) [7], a graph Laplacian regularizer states that a natural image patch $\mathbf{x}$ in vector form should induce a small value $S_{\mathcal{G}}(\mathbf{x}) = \mathbf{x}^{\text{T}}\mathbf{L}\mathbf{x}$, where $\mathbf{L}$ is the *graph Laplacian matrix* for a well defined combinatorial graph whose vertices correspond to pixels in patch $\mathbf{x}$ (to be defined rigorously later). Using such regularization term, the denoising problem (2) becomes:

$$\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{x}\|_2^2 \; + \; \lambda \; \mathbf{x}^{\text{T}}\mathbf{L}\mathbf{x}. \tag{3}$$

Though good experimental results were reported in [4], [5] for image denoising, there was neither theoretical justification nor intuitive interpretation why the graph Laplacian regularizer is a reasonable image prior that would lead to good denoising performance. In this paper, our first endeavor is to provide this crucial missing link: we show that, under mild conditions, the graph Laplacian regularizer converges to a continuous space functional counterpart, and by suitably selecting features that define $\mathbf{L}$ and subsequently the converged functional, one can indeed obtain a powerful *discriminant* signal prior for image restoration.

In an orthogonal line of attack, it has been shown that natural images exhibit a *self-similiar* characteristic: an observed pixel patch is likely repeated throughout the image in far-away (non-local) spatial regions. Image denoising schemes such as the famed *non-local means* (NLM) [8] that exploit this characteristic have demonstrated significant performance gain. Self-similarity is typically defined in the *pixel domain*; patches $\mathbf{y}_i$ and $\mathbf{y}_j$ are similar if the $l_2$-norm of their *pixel intensity difference* $\|\mathbf{y}_i-\mathbf{y}_j\|_2^2$ is small. In contrast, our second endeavor is to defy convention and redefine self-similarity in the *gradient domain*—two patches are similar if their *gradient difference*[1] $\|\nabla\mathbf{y}_i - \alpha_j\nabla\mathbf{y}_j\|_2^2$ is small for some constant $\alpha_j$. We argue that this definition leads to a more accurate estimate of graph Laplacian $\mathbf{L}$, and hence a better graph Laplacian regularizer for denoising in (3). Our more esoteric definition of self-similarity is justified by both our theoretically derived conditions for a discriminant graph Laplacian regularizer and a well accepted intrinsic image model [9]. We design an image denoising algorithm based on the graph Laplacian regularizer and gradient-based self-similarity, and show experimentally that our proposal outperformed NLM by up to 1.4 dB in PSNR.

The outline of the paper is as follows. We first discuss our interpretation of graph Laplacian regularizer in the continuous space and argue for its usefulness in Section II. We present our image model in Section III, based on which we

[1]We use here the intuitive notation $\nabla\mathbf{y}$ to denote the gradient of an image patch $\mathbf{y}$, even though $\nabla$ is typically used for continuous functions.

978-616-361-823-8 © 2014 APSIPA 

formulate our optimization in Section IV. Our gradient-based image denoising algorithm is discussed in Section V. Finally, experimentation and conclusion are presented in Section VI and VII, respectively.

## II. Continuous Domain Interpretation of the Graph Laplacian Regularizer

To solve an under-determined inverse imaging problem, many recent works (*e.g.*, [5], [6], [10]) employed a graph Laplacian regularizer $S_{\mathcal{G}}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{L}\mathbf{x}$ during optimization as shown in (3), where $\mathbf{x}$ is a candidate signal and $\mathbf{L}$ is the graph Laplacian matrix. The weight of an edge $w_{ij}$ connecting two vertices in the graph—corresponding to two pixels $i$ and $j$ in $\mathbf{x}$—is computed using a Gaussian kernel:

$$w_{ij} = \exp\left(-\frac{[\mathrm{dist}(i,j)]^2}{2\epsilon^2}\right) \tag{4}$$

for a weight parameter $\epsilon$. $\mathrm{dist}(i,j)$ computes the "distance" (for a well defined metric) between pixels $i$ and $j$. However, no existing work has yet to provide theoretical justifications for the regularization term.

In contrast, we demonstrate that by interpreting a graph as a discrete object composed of random samples on a manifold [7], [11], $S_{\mathcal{G}}$ converges to a continuous functional. Further, appropriate design of a set of *feature functions* that determine $\mathbf{L}$ leads to a discriminant regularizer $S_{\mathcal{G}}$ for image restoration.

### A. Graph Construction from Continuous functions

Denote a bounded region on the 2D plane by $\Omega \subset \mathbb{R}^2$; we call $\Omega$ the *domain* in the sequel. Let $f_n(x,y) : \Omega \to \mathbb{R}$, $1 \leq n \leq N$, be continuous *feature functions* defined on domain $\Omega$ and can be freely chosen by users. Let $\Gamma = \{\mathbf{s}_i = [x_i \;\; y_i]^{\mathrm{T}} \mid \mathbf{s}_i \in \Omega, 1 \leq i \leq M\}$ be a set of $M$ *random samples* uniformly distributed on $\Omega$. By sampling the feature functions $\{f_n\}_{n=1}^N$ at positions in $\Gamma$, $N$ vectors of length $M$ are obtained,

$$\mathbf{f}_n^D = [f_n(x_1,y_1) \;\; f_n(x_2,y_2) \;\; \ldots \;\; f_n(x_M,y_M)]^{\mathrm{T}}, \tag{5}$$

where $1 \leq n \leq N$, and superscript $D$ means "discrete". Fig. 1 illustrates the sampling process of an example function $f_n$—a simple ramp in domain $\Omega$. The red crosses are sampling positions in $\Gamma$. The blue dots are samples of $f_n$ and collectively form vector $\mathbf{f}_n^D$.

In practice, $\Omega$ takes the shape of an image (or image patch) which is typically a rectangle (*e.g.*, Fig. 1). Feature functions $\{f_n\}_{n=1}^N$ can be chosen as continuous images. Since images are typically sampled uniformly on $\Omega$ in a 2D grid, samples in the set $\Gamma$ can be construed as pixel coordinates. $\mathbf{f}_n^D$ are the discretized versions of feature functions $f_n$, $1 \leq n \leq N$.

For each sample location $\mathbf{s}_i \in \Gamma$, we construct a length $N+2$ vector $\mathbf{v}_i$ with spatial coordinates $x_i$, $y_i$ and previously defined $\mathbf{f}_n^D$,

$$\mathbf{v}_i = [x_i \;\; y_i \;\; \beta\mathbf{f}_1^D(i) \;\; \beta\mathbf{f}_2^D(i) \;\; \ldots \;\; \beta\mathbf{f}_N^D(i)]^{\mathrm{T}}, \tag{6}$$

where $\mathbf{f}_n^D(i)$ is the $i$-th entry of $\mathbf{f}_n^D$, and $\beta$ is a tunable scaling factor since the spatial coordinates $\mathbf{s}_i$ and samples $\mathbf{f}_n^D(i)$ are inherently different quantities. With vectors $\{\mathbf{v}_i\}_{i=1}^M$, we can build a weighted neighborhood graph $\mathcal{G}$, where each location (or pixel) $\mathbf{s}_i \in \Gamma$ is represented by a vertex $V_i$. The weight $w_{ij}$ between two different vertices $V_i$ and $V_j$ is computed as

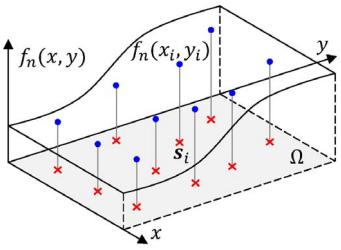



Fig. 1. Sampling the function $f_n$ uniformly in domain $\Omega$. Red crosses are sampling positions and blue dots are the samples.

$$w_{ij} = (\rho_i\rho_j)^{-\gamma}\psi(d_{ij}), \tag{7}$$

where the distance $d_{ij}^2 = \|\mathbf{v}_i - \mathbf{v}_j\|_2^2$ is also given by

$$d_{ij}^2 = \|\mathbf{s}_i - \mathbf{s}_j\|_2^2 + \beta^2 \sum\nolimits_{n=1}^{N} \left(\mathbf{f}_n^D(i) - \mathbf{f}_n^D(j)\right)^2, \tag{8}$$

the weighting kernel $\psi(\cdot)$ is a truncated Gaussian

$$\psi(d) = \begin{cases} \exp\left(-\dfrac{d^2}{2\epsilon^2}\right) & \text{if } |d| \leq r, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

and the degree of $V_i$ before normalization is

$$\rho_i = \sum\nolimits_{j=1}^{M} \psi(d_{ij}). \tag{10}$$

Under these settings, $\mathcal{G}$ is an $r$-neighborhood graph; *i.e.*, there is no edge connecting two vertices with distance greater than $r$. Here $r = \epsilon C_r$, and $C_r$ is a constant. The parameter $\epsilon$ controls the sensitivity of the graph weights to the distances, and $\gamma$ controls the normalization of the weights. Denote the adjacency matrix of $\mathcal{G}$ by $\mathbf{A}$, where entry $(i,j)$ of $\mathbf{A}$ is $w_{ij}$. The degree matrix of $\mathcal{G}$ is a diagonal matrix $\mathbf{D}$ with entry $(i,i)$ computed as $\sum_{j=1}^M w_{ij}$. The unnormalized graph Laplacian $\mathbf{L}$ is simply $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

Note that graphs employed in many recent works (*e.g.*, [4], [5], [10]) are special cases of our more generally defined graph $\mathcal{G}$. For example, it can be shown that the widely-adopted 4-connected 2D grid graph can be obtained by choosing a sufficiently small scaling factor $\beta$ and a proper $r$.

### B. Convergence of Graph Laplacian Regularization

Let $h(x,y) : \Omega \to \mathbb{R}$ be yet another continuous function defined on domain $\Omega$, which we call the *candidate function*. Sampling $h$ at positions in $\Gamma$ leads to its discretized version, $\mathbf{h}^D = [h(x_1,y_1) \;\; h(x_2,y_2) \;\; \ldots \;\; h(x_M,y_M)]^{\mathrm{T}}$. The graph Laplacian $\mathbf{L}$ obtained in Section II-A induces a quadratic functional $S_{\mathcal{G}}$ on $\mathbb{R}^M$:

$$S_{\mathcal{G}}(\mathbf{h}^D) = (\mathbf{h}^D)^{\mathrm{T}}\mathbf{L}\mathbf{h}^D. \tag{11}$$

$S_{\mathcal{G}}$ is the graph Laplacian regularizer we are considering.

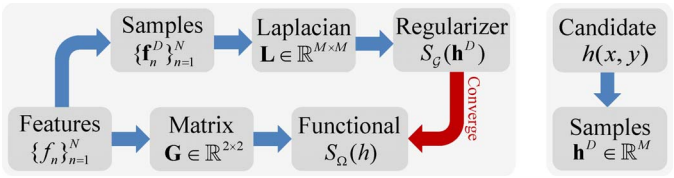



Fig. 2. Relationships among essential defined quantities, where a blue arrow pointing from $A$ to $B$ means $B$ is derived from $A$.

The continuous counterpart of regularizer $S_{\mathcal{G}}$ is given by a functional $S_\Omega$ on domain $\Omega$,

$$S_\Omega(h) = \iint_\Omega (\nabla h)^{\mathrm{T}} \mathbf{G}^{-1} (\nabla h) \left(\sqrt{\det \mathbf{G}}\right)^{2\gamma-1} dxdy, \quad (12)$$

where $\nabla h = [\partial_x h \ \partial_y h]^{\mathrm{T}}$ is the gradient of candidate function $h$. $\mathbf{G}$ is a 2×2 matrix given by

$$\mathbf{G} = \beta^2 \begin{bmatrix} \sum_{n=1}^N (\partial_x f_n)^2 & \sum_{n=1}^N \partial_x f_n \cdot \partial_y f_n \\ \sum_{n=1}^N \partial_x f_n \cdot \partial_y f_n & \sum_{n=1}^N (\partial_y f_n)^2 \end{bmatrix} + \mathbf{I}_2, \quad (13)$$

where $\mathbf{I}_2$ is a 2×2 identity matrix. $\mathbf{G}$ is a function of location $(x, y)$. For a certain $(x, y)$, $\mathbf{G}$ is the sum of $\mathbf{I}_2$ and the scaled *structure tensor* of gradients $\{\nabla f_n(x, y)\}_{n=1}^N$ [12].

We see that the set of feature functions $\{f_n\}_{n=1}^N$ determine the graph Laplacian regularizer $S_{\mathcal{G}}$ and the continuous functional $S_\Omega$. Relationships among several key quantities are illustrated in Fig. 2. We can now declare the following theorem:

**Theorem 1** (Convergence of $S_{\mathcal{G}}$)**.** *Under mild conditions as stated in* [11] *for $\epsilon$, functions $\{f_n\}_{n=1}^N$ and $h$,*

$$\lim_{\substack{M\to\infty \\ \epsilon\to 0}} \frac{M^{2\gamma-1}}{\epsilon^{4(1-\gamma)}(M-1)} S_{\mathcal{G}}(\mathbf{h}^D) \sim S_\Omega(h), \quad (14)$$

*where "$\sim$" means there exists a constant depending on $\Omega$, $C_r$ and $\gamma$, such that equality holds.*

In other words, as the number of samples (or resolution) $M$ increases, while the neighborhood size $r = \epsilon C_r$ shrinks, graph Laplacian regularizer $S_{\mathcal{G}}$ approaches the continuous functional $S_\Omega$. The Appendix proves Theorem 1 by viewing a graph as a *proxy* of Riemannian manifold[2].

### C. Justification of Graph Laplacian Regularizer

The convergence of the graph Laplacian regularizer $S_{\mathcal{G}}$ to the continuous functional $S_\Omega$ allows us to justify the usage of $S_{\mathcal{G}}$ as a regularization term in inverse imaging problems via the analysis of $S_\Omega$.

In (12), the quadratic term $(\nabla h)^{\mathrm{T}} \mathbf{G}^{-1} (\nabla h)$ measures the length of gradient $\nabla h$ in a 2D metric space established by the matrix $\mathbf{G}$ [13]. From (13), $\mathbf{G}$ is computed using the set of gradients $\{\nabla f_n(x, y)\}_{n=1}^N$; $\mathbf{G}$'s eigenvectors and eigenvalues capture the statistics of $\{\nabla f_n(x, y)\}_{n=1}^N$. Fig. 3 shows two example diagrams of such metric spaces at some location $(x, y) \in \Omega$ drawn in gradient coordinates. In each diagram, the red cross marked by $A$ is the gradient of the target signal, which can be different for different location $(x, y)$. The blue dots are gradients $\{\nabla f_n\}_{n=1}^N$. The eigenvector corresponding to the largest eigenvalue of $\mathbf{G}$ has direction $l$. Intuitively, the line $l$ passes through the origin and "aligns" with the cluster of gradients $\{\nabla f_n\}_{n=1}^N$. The ellipses are isolevel lines (norm balls) of the metric space, and the flatness of their shapes reflects how concentrated the gradients $\{\nabla f_n\}_{n=1}^N$ are.

[2]We prove the convergence by applying the result of [11], where uniform convergence of graph Laplacian regularizer to a functional on Riemannian manifold is proved. In our work, to facilitate presentation, we weaken such result to point-wise convergence.

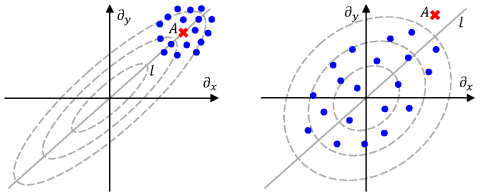



Fig. 3. At a position $(x, y) \in \Omega$, gradients $\{\nabla f_n(x, y)\}_{n=1}^N$ establish a metric space for regularization. (a) A proper metric space has norm balls stretch towards the target gradient $A$. (b) The features are wrongly chosen, resulting in non-discriminant regularization.

Consider using the continuous functional $S_\Omega$ as the regularizer. Ideally, we seek for a *discriminant* regularizer; *i.e.,* $\forall (x, y) \in \Omega$, a small Euclidean distance from the corresponding ground-truth gradient yields a large value in the metric space defined by $\mathbf{G}$. To achieve this goal, one should design features $\{f_n\}_{n=1}^N$ appropriately, so that: i) $l$ goes through the ground-truth gradient $A$; and ii) the ellipses should be stretched flat along $l$. Fig. 3(a) shows a reasonable scheme of choosing good features $\{f_n\}_{n=1}^N$, where the blue dots are clustering closely around $A$; *i.e.*, the gradients $\{\nabla f_n\}_{n=1}^N$ are *similar* to the ground truth $A$. As a counterexample, in Fig. 3(b) the blue dots spread out in the space and $A$ does not lie on $l$, resulting in a regularizer that is not discriminant. Therefore to achieve the same effect as Fig. 3(a) and obtain a discriminant regularizer $S_\Omega$, one should let the gradients of the features $\{f_n\}_{n=1}^N$ be close to the ground-truth gradient for all positions $(x, y) \in \Omega$.

When operating in discrete domain, the set $\{\mathbf{f}_n^D\}_{n=1}^N$ should have gradients similar to the ground-truth gradients. Specifically, denote the intensity difference between pixel $i$ and pixel $j$ of the ground truth discrete image as $g^{ij}$, which is the graph signal gradient by viewing the ground truth image as a graph signal. Then for all pixels $i$ and $j$, $\{\mathbf{f}_n^D(i) - \mathbf{f}_n^D(j)\}_{n=1}^N$ should be close to $g^{ij}$. This means that to obtain a discriminant graph Laplacian regularizer $S_{\mathcal{G}}$, one should seek for patches similar to the ground truth patch in terms of *gradients*.

## III. Gradient-Based Self-Similarity for Natural Images

In the previous section, we show that similar patches in terms of gradients can lead to a discriminant graph Laplacian

regularizer. However, do patches with similar gradients exist in natural images? From the perspective of intrinsic image decomposition, we argue that they do exist, where the gradients of such patches are *similar up to a scaling factor*. We first discuss the intrinsic image model. We then present our gradient-based image formation model and introduce a noise model for our intended application. Finally, we discuss the advantages of gradient-based self-similarity over traditional notions defined in terms of squared intensity difference.

### A. Justification of Gradient-Based Self-Similarity

We argue that patches with similar gradients do exist in natural images via the *intrinsic image model* in [9], where a natural image can be written as the product of two intrinsic images: a *reflectance image* and an *illumination image*. Specifically, a natural image $I(x, y)$ can be decomposed as:

$$I(x, y) = R(x, y) \cdot L(x, y), \tag{15}$$

where $(x, y)$ is a given pixel location, and $I(x, y)$, $R(x, y)$ and $L(x, y)$ denote the natural image, the reflectance image and the illumination image, respectively. Reflectance $R$ describes how each pixel responds to light; it contains details of the scene. Illumination $L$ describes the interaction between the geometry in the scene and lighting; typically it changes much slower than $R$.

Given this intrinsic image model, we assume:

(i) in the reflectance image $R$, there exist many similar patches in terms of squared intensity difference;
(ii) most local regions in illumination $L$ are constants.

Given these assumptions and (15), similar patches in $R$ are multiplied by different scalars from illumination $L$. This means that *pixel patches in a natural image $I$ with similar gradients scaled by different factors can be readily observed.*

### B. Image Formation Based on Signal Gradient

More concretely, for a $\sqrt{M} \times \sqrt{M}$ image patch $\mathbf{p}_0$ in $I$, one can identify a set of $K-1$ patches $\{\mathbf{p}_k\}_{k=1}^{K-1}$ similar to $\mathbf{p}_0$ in terms of gradients with scaling factors $\{\alpha_k\}_{k=1}^{K-1}$. Together with $\mathbf{p}_0$, the $K$ patches $\{\mathbf{p}_k\}_{k=0}^{K-1}$ are collectively called a *cluster* in the sequel, where each patch in the cluster comes from a different spatial location in image $I$.

If we now interpret $\mathbf{p}_k$ as a signal on a graph, then $g_k^{ij} = \mathbf{p}_k(i) - \mathbf{p}_k(j)$ is the graph signal gradient, where $\mathbf{p}_k(i)$ denotes the value of $\mathbf{p}_k$ at pixel location $i$. For simplicity, hereafter the superscripts $ij$ are neglected. With the notion of gradient-based self-similarity, we model the graph signal gradient $g_k$ as

$$g_k = \alpha_k g + e_{f,k},\ 0 \leq k \leq K-1, \tag{16}$$

where $\alpha_0 = 1$ and $e_{f,0} = 0$, $g$ is the unknown ground-truth graph signal gradient on $\mathbf{p}_0$. $e_{f,k}$ is a noise term introduced in the image formation process, which is an instance of a random variable $e_f$.

### C. Noise Corruption on Graph Signal Gradient

Having defined our notion of gradient-based self-similarity, we now present our noise model of graph signal gradient. We assume that the input noisy image is corrupted by independent and identically distributed (i.i.d.) additive white Gaussian noise (AWGN) in the pixel domain. Then equivalently in the gradient domain, $\{g_k\}_{k=0}^{K-1}$ from observations $\{\mathbf{p}_k\}_{k=0}^{K-1}$ are also corrupted by AWGN[3]. The graph signal gradient becomes

$$\hat{g}_k = g_k + e_{g,k} = \alpha_k g + e_{f,k} + e_{g,k},\ 0 \leq k \leq K-1, \tag{17}$$

where $e_{g,k}$ represents the i.i.d. AWGN, which is an instance of a random variable $e_g$. Note that the image formation noise $e_f$ and the AWGN $e_g$ are independent, zero-mean, and have variances $\sigma_f^2$ and $\sigma_g^2$, respectively.

### D. Advantages of Gradient-Based Self-Similarity

Compared to the squared intensity difference based metric typically used in the literature [4], [8], self-similarity in terms of gradients broadens the notion of similarity and leads to more number of similar patches, since patches that are similar in terms of squared intensity difference are also similar in terms of gradients, but not vice versa.

Moreover, self-similarity in terms of gradients turns out to be more sensitive to edges and textures, which helps to restore and preserve edges and textures. We demonstrate this prominent feature through our experiments in Section VI.

Finally, the similarity notion introduced in this section matches up well with the discussion on discriminant graph Laplacian regularizer presented in Section II-C, because it considers self-similarity directly in the gradient domain.

## IV. Problem Formulation

We now adopt a patch-based recovery framework to denoise a corrupted image, as done in previous works [4], [6]. Given the aforementioned noise model on graph signal gradient, we first estimate graph weights accurately, which translate to entries in the graph Laplacian matrix. We then formulate the patch-based denoising problem as a quadratically constrained quadratic programming (QCQP) problem with the graph Laplacian regularizer as objective. Our developed image denoising method is named *graph-based denoising using gradient-based self-similarity* (GDGS).

### A. Graph Weights from Noise-Corrupted Gradients

For a given noisy patch $\mathbf{p}_0$, we seek to build an appropriate graph $\mathcal{G}$ with the cluster $\{\mathbf{p}_k\}_{k=0}^{K-1}$. According to the analysis in Section II-C, to obtain a discriminant regularizer $S_{\mathcal{G}}$, one should let the graph signal gradients of the feature functions $\{\mathbf{f}_n^D(i) - \mathbf{f}_n^D(j)\}_{n=1}^{N}$ in (8) be similar to the ground-truth gradient $g$. However, from (17), the gradients $\{\hat{g}_k\}_{k=1}^{K-1}$ on $\{\mathbf{p}_k\}_{k=1}^{K-1}$ are similar to $g$, *up to a scaling factor*.

Before addressing the scaling issue, for intuition we first consider the simpler special case where the scalings in (17) are

[3]For simplicity, we assume the i.i.d. AWGN condition in the pixel domain carries to the gradient domain.

all ones. Consequently, the noisy gradient observations become instances of a random variable $\hat{g}$, where

$$\hat{g} = g + e_f + e_g. \tag{18}$$

By interpreting the gradients $\{\mathbf{f}_n^D(i) - \mathbf{f}_n^D(j)\}_{n=1}^N$ as observations of $\hat{g}$, and letting $N$ to be sufficiently large, (8) becomes

$$d_{ij}^2 = \|\mathbf{s}_i - \mathbf{s}_j\|_2^2 + W \cdot \mathrm{E}[\hat{g}^2], \tag{19}$$

where the weight $W = N\beta^2$ is a constant. Recall that $N$ denotes the number of feature functions and $\beta$ is the scaling factor.

$\mathrm{E}[\hat{g}^2]$ can be written as:

$$\mathrm{E}[\hat{g}^2] = (\mathrm{E}[\hat{g}])^2 + \mathrm{Var}[\hat{g}] = g^2 + (\sigma_f^2 + \sigma_g^2). \tag{20}$$

In [4], Hu *et al.* first averaged all similar patches to compute the graph weights, which is equivalent to using only $(\mathrm{E}[\hat{g}])^2$, neglecting $\mathrm{Var}[\hat{g}]$ in (20). However, with our interpretation of $\{\mathbf{f}_n^D(i) - \mathbf{f}_n^D(j)\}_{n=1}^N$, such a simplification is clearly inaccurate in general for $\mathrm{Var}[\hat{g}] > 0$; equivalently, *Jensen's inequality* states that $\mathrm{E}[\hat{g}^2] \geq (\mathrm{E}[\hat{g}])^2$ always holds.

Unlike [4], we estimate $\mathrm{E}[\hat{g}^2]$ to obtain appropriate graph weights. In order to do so, we generalize back to the noise model with scalings (17) and perform the estimation via the following two-step approach. Accounting for the scaling factors $\alpha_k$, the mean gradient $g$ is first estimated by weighted averaging $\{\hat{g}_k\}_{k=0}^{K-1}$ as

$$\tilde{g} = \sum\nolimits_{k=0}^{K-1} \left(\frac{\omega_k}{\alpha_k}\right) \hat{g}_k, \tag{21}$$

where $\tilde{g}$ denotes an estimate of $g$, and $\{\omega_k\}_{k=0}^{K-1}$ are weights of the gradients. Then we compute $\{\hat{g}_k - \alpha_k \tilde{g}\}_{k=0}^{K-1}$ and evaluate their sample variance, leading to the estimate of $\sigma_f^2 + \sigma_g^2$ in (20), denoted as $\tilde{\sigma}^2$. Estimate of $\mathrm{E}[\hat{g}^2]$ is then given by $\tilde{g}^2 + \tilde{\sigma}^2$.

### B. QCQP Formulation of GDGS

Having obtained the graph $\mathcal{G}$ and the graph Laplacian $\mathbf{L}$ for the noisy patch $\mathbf{p}_0 \in \mathbb{R}^M$, the following QCQP is formulated:

$$\mathbf{q}^\star = \arg\min_{\mathbf{q}} \ \mathbf{q}^{\mathrm{T}}\mathbf{L}\mathbf{q} \quad \text{s.t.} \ \|\mathbf{p}_0 - \mathbf{q}\|_2^2 \leq C_e \sigma_e^2, \tag{22}$$

where $\sigma_e^2$ is the variance of the Gaussian noise and $C_e < 1$ is a constant. The constraint of (22) is the fidelity term which preserves the closeness between the noisy observation and the denoised patch. The formulation (22) allows us to control the difference $\|\mathbf{p}_0 - \mathbf{q}^\star\|_2^2$ directly via $C_e$.

## V. Algorithm Development

We now develop an iterative patch-based image denoising algorithm to solve the formulated QCQP problem (22) in Section IV. We describe our algorithm step-by-step in details next. A summary is shown in Algorithm 1.

**Algorithm 1** Image denoising with GDGS

1: **Input:** One noisy grayscale image $I$
2: $\hat{I}^{(0)} = I$;
3: **for** $i = 1$ to $iter$ **do**
4: Step A: Clustering of patches with similar gradients
5: Step B: Graph Laplacian from similar patches
6: Step C: Lagrangian optimization for patch denoising
7: Step D: Image update
8: **end for**
9: **Output:** The denoised grayscale image $\hat{I}^{(iter)}$

### A. Clustering of Patches with Similar Gradients

Given a $\sqrt{M} \times \sqrt{M}$ noisy image patch $\mathbf{p}_0$, we first search for similar patches on the image in terms of patch gradients; let the number of sufficiently similar patches be $K - 1$.

Specifically, since each pixel in $\mathbf{p}_0$ induces a 2D gradient vector containing partial derivatives along horizontal and vertical directions, by concatenating these $M$ vectors, a combined column vector of length $2M$ is obtained. This vector, denoted as $\mathbf{f}_0$, is the *gradient field* of $\mathbf{p}_0$. Given a candidate patch $\mathbf{p}_c$ with gradient field $\mathbf{f}_c$, to measure the gradient-based similarity between $\mathbf{p}_0$ and $\mathbf{p}_c$, we solve the following scalar minimization problem,

$$t = \min_{\alpha} \left\|\mathbf{f}_0 - \alpha^{-1}\mathbf{f}_c\right\|_2^2 \ \text{ s.t. } \alpha_l \leq \alpha \leq \alpha_u, \tag{23}$$

where the permissible range of the scaling is specified by $\alpha_l \leq 1$ and $\alpha_u \geq 1$. Objective value $t$ is the metric reflecting the gradient similarity between $\mathbf{p}_0$ and $\mathbf{p}_c$; $\alpha\mathbf{f}_0$ is similar to $\mathbf{f}_c$. By solving (23), both the similarity metric $t$ and the estimate of the scaling factor $\alpha$ are obtained simultaneously. Note that since the input image is noise-corrupted, we apply the robust 2D Savitzky-Golay filter [14] to estimate the gradients. Savitzky-Golay filter fits low-degree 2D polynomials to each image patch. Then for a particular pixel, its estimated gradient comes from the gradient of its corresponding 2D polynomial.

### B. Graph Laplacian from Similar Patches

Having found $K - 1$ similar patches in terms of patch gradients, we construct a weighted neighborhood graph using the methodology described in Section IV-A. The corresponding graph Laplacian $\mathbf{L}$ is then obtained.

### C. Lagrangian Optimization for Patch-Based Denoising

Up to a convex hull approximation [15], the QCQP in (22) is equivalent to the following unconstrained quadratic programming (QP) problem,

$$\mathbf{q}^\star = \arg\min_{\mathbf{q}} \ \mathbf{q}^{\mathrm{T}}\mathbf{L}\mathbf{q} + \lambda \|\mathbf{p}_0 - \mathbf{q}\|_2^2, \tag{24}$$

where $\lambda$ is a Lagrange multiplier chosen such that $\|\mathbf{p}_0 - \mathbf{q}^\star\|_2^2$ is as close to $C_e\sigma_e^2$ as possible, while $\|\mathbf{p}_0 - \mathbf{q}^\star\|_2^2 \leq C_e\sigma_e^2$. Because (24) has a closed-form solution that can be easily solved, we solve (22) by solving (24) multiple times while adjusting $\lambda$ appropriately, *e.g.*, via a bisection search [16].

TABLE I
NATURAL IMAGE DENOISING WITH GDGS: PERFORMANCE COMPARISON IN PSNR (DB) WITH FOUR COMPETING METHODS

| Image | Method | Standard Deviation $\sigma_e$ | | | | |
|---|---|---|---|---|---|---|
| | | 10 | 15 | 20 | 25 | 30 |
| Lena | GDGS | **35.24** | **33.47** | **32.35** | **31.39** | **30.61** |
| | GDSS | 35.21 | 33.41 | 32.28 | 31.32 | 30.54 |
| | NLGBT | 34.98 | 33.22 | 31.90 | 30.97 | 30.19 |
| | NLM | 34.26 | 32.03 | 31.51 | 30.38 | 29.45 |
| | BF | 29.48 | 27.00 | 24.80 | 23.00 | 21.52 |
| Barbara | GDGS | **33.69** | **31.71** | **30.33** | **29.24** | **28.33** |
| | GDSS | 33.53 | 31.48 | 30.13 | 29.05 | 28.14 |
| | NLGBT | 33.20 | 31.22 | 29.62 | 28.57 | 27.67 |
| | NLM | 33.13 | 30.76 | 30.15 | 28.94 | 27.91 |
| | BF | 28.16 | 25.78 | 23.86 | 22.32 | 21.03 |
| Boats | GDGS | **33.40** | **31.59** | **30.30** | **29.34** | **28.55** |
| | GDSS | 33.38 | 31.55 | 30.27 | 29.28 | 28.45 |
| | NLGBT | 32.89 | 31.05 | 29.56 | 28.60 | 27.77 |
| | NLM | 32.88 | 30.69 | 29.74 | 28.62 | 27.68 |
| | BF | 27.91 | 26.42 | 24.89 | 23.47 | 22.19 |
| Peppers | GDGS | 34.78 | 33.30 | **32.38** | **31.54** | **30.83** |
| | GDSS | **34.79** | **33.31** | **32.38** | 31.53 | **30.83** |
| | NLGBT | 34.62 | 33.18 | 32.09 | 31.24 | 30.49 |
| | NLM | 33.97 | 31.96 | 31.48 | 30.42 | 29.50 |
| | BF | 28.96 | 26.70 | 24.67 | 22.95 | 21.49 |

We choose $C_e$ so that $C_e\sigma_e^2$ is smaller than the variance of $\mathbf{p}_0$; otherwise a constant patch $\mathbf{q}$ with values as the mean of $\mathbf{p}_0$ makes $\mathbf{q}^{\mathrm{T}}\mathbf{L}\mathbf{q}$ becomes zero and minimizes (22).

### D. Image Update

We denoise a given image iteratively to gradually enhance its quality. At the end of the $i$-th iteration, all the denoised patches are aggregated to form an intermediate denoised image $\hat{I}^{(i)}$. To do so, we estimate each pixel with the weighted average of the values from different overlapping patches. In general, if a patch $\mathbf{p}_0$ corresponds to a cluster $\{\mathbf{p}_k\}_{k=0}^{K-1}$ with better agreement with $\mathbf{p}_0$, then it should be recovered with higher quality. As a result, we empirically set the weight of a denoised $\mathbf{p}_0$ to be inversely proportional to $\sum_{k=0}^{K-1} t_k$.

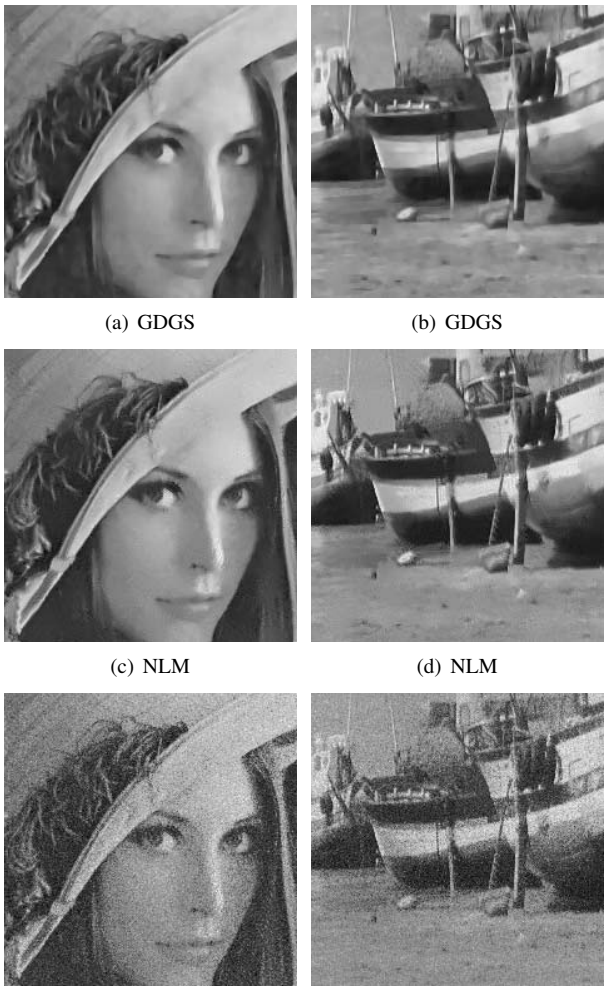
## VI. EXPERIMENTATION

We now evaluate the denoising performance of our proposed GDGS and compare it against several competing denoising algorithms.

### A. Experimental Setup

The test image set was composed of four 512×512 gray-scale images: *Lena*, *Barbara*, *Boats*, and *Peppers*. In the experiments, the images were corrupted by i.i.d. AWGN, with standard deviation $\sigma_e$ ranging from 10 to 30.

We first compared GDGS with two competing methods: bilateral filtering (BF) [8] and non-local means denoising (NLM) [17]. Then we swapped the patch similarity metric in our method to one based on squared intensity difference, which we call graph-based denoising using squared-difference-based self-similarity (GDSS). Performance of GDGS and GDSS were compared to validate the notion of self-similarity in terms of gradients.



(a) GDGS (b) GDGS (c) NLM (d) NLM (e) BF (f) BF

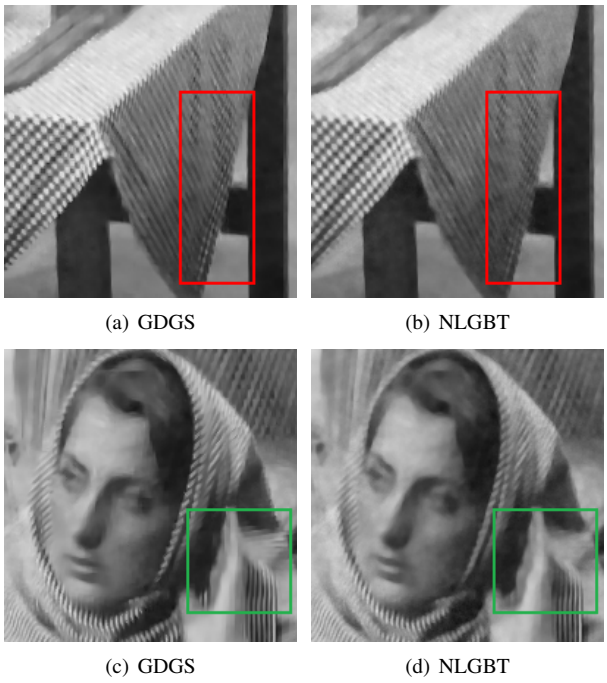
Fig. 4. Fragments of different denoised versions of the images *Lena* and *Boats*. The original images are corrupted by AWGN with $\sigma_e = 25$.

We also modified GDSS and estimated $\mathrm{E}[\hat{g}^2]$ using only $(\mathrm{E}[\hat{g}])^2$ in (20). For convenience, we call the obtained method non-local GBT (NLGBT), given that it bears the same rationale as the NLGBT method proposed in [4] (though the formulation in [4] exploits the sparsity prior in graph transform domain). Comparing NLGBT with GDGS and GDSS validates the superiority of our graph weight estimation.

In our implementation, the patch size $\sqrt{M}$ was set as 25. To compute $\tilde{g}$ in (21), we empirically assigned the weights of the 40% most similar patches to be three times heavier than the rest. We set the threshold $r$ in (9) such that each vertex had at least eight edges, the normalization factor in (7) was chosen as $\gamma = 0.4$. For a reasonable scaling of the gradients, $\alpha_l$ and $\alpha_u$ in (23) were set to be 0.5 and 2, respectively. We let $iter = 3$ for a reasonable tradeoff between denoising quality and complexity.

### B. Objective Evaluation

Objective performance of the competing denoising methods (measured in PSNR) are tabulated in Table I. We see that



(a) GDGS (b) NLGBT

(c) GDGS (d) NLGBT

Fig. 5. Fragments of different denoised version of image *Barbara* corrupted by AWGN ($\sigma_e = 25$). The fragments of the two rows come from different positions on the denoised images.

GDGS provided superior denoising performance, achieving up to 1.4 dB gain over NLM (*Lena*, $\sigma_e = 15$) and 9.3 dB gain over BF (*Peppers*, $\sigma_e = 30$). Note that for image *Barbara* that contains plenty of textures and edges, GDGS performed consistently better than GDSS by about 0.2 dB. GDGS also produced consistently better results than NLGBT, ranging from about 0.1 dB (*Peppers*, $\sigma_e = 15$) to 0.8 dB (*Boats*, $\sigma_e = 30$).

### C. Subjective Evaluation

We now present visual comparisons of different denoising methods. We neglect GDSS here since visually it gives results similar to GDGS.

Comparison of GDGS with NLM and BF is shown in Fig. 4, which shows segments from different denoised versions of the images *Lena* and *Boats* corrupted by AWGN ($\sigma_e = 25$). We see that results of GDGS exhibit natural and clear appearances, while NLM smears details and BF fails to clean up the noise.

To compare GDGS against NLGBT, Fig. 5 shows segments from the denoised version of image $Barbara$ corrupted by AWGN ($\sigma_e = 25$). Note that NLGBT smoothed out some fine details from the image; while GDGS provided well-preserved textures and edges. The faithful results of GDGS are credited to the gradient-based self-similarity and our modeling on graph weights.

## VII. Conclusion

Image denoising is an under-determined problem and requires good signal priors to regularize the problem appropriately. In this paper, we first provide theoretical justification of why and under what conditions a recently proposed graph Laplacian regularizer can be discriminant. We then redefine the notion of patch self-similarity in a natural image in terms of gradients. We design an image denoising algorithm using a graph Laplacian regularizer as objective, where the graph Laplacian matrix is computed from non-local patches with similar gradients. Experiments show that our proposal can outperform non-local means (NLM) by up to 1.4 dB in PSNR.

## Appendix
## Proof of Theorem 1

*Proof:* Let $\mathcal{M}$ be a 2D Riemannian manifold embedded in $(N+2)$-dimensional ambient space through the embedding $\mathbf{\Phi} : \mathcal{M} \to \mathbb{R}^{N+2}$. Specifically,

$$\mathbf{\Phi} : (\sigma_1, \sigma_2) \to (\sigma_1, \sigma_2, \beta f_1(\sigma_1, \sigma_2), \ldots, \beta f_N(\sigma_1, \sigma_2)), \tag{25}$$

where $(\sigma_1, \sigma_2)$ is the global coordinate of $\mathcal{M}$. Under embedding $\mathbf{\Phi}$, the induced metric of $\mathcal{M}$ in $\mathbb{R}^{N+2}$ can be pulled back [18],[19], which is given by the matrix $\mathbf{G}$ (13).

Then we link the sampling positions in $\Gamma$ to a probability density function (PDF) defined on manifold $\mathcal{M}$. Let the one-to-one mapping $\mathbf{\Psi} : \mathcal{M} \to \Omega$ be

$$\mathbf{\Psi} : (\sigma_1, \sigma_2) \to (x = \sigma_1, y = \sigma_2). \tag{26}$$

Then let the function $p(x, y) : \Omega \to \mathbb{R}$ be

$$p(x, y) = 1/(|\Omega| \sqrt{\det \mathbf{G}}), \tag{27}$$

where $|\Omega|$ denotes the area of $\Omega$. Through mapping $\mathbf{\Psi}$, a function $p^{\mathcal{M}}(\sigma_1, \sigma_2) : \mathcal{M} \to \mathbb{R}$, is obtained, where $p^{\mathcal{M}}(\sigma_1, \sigma_2) = p(\mathbf{\Psi}(\sigma_1, \sigma_2))$. Because of $\mathbf{\Psi}$, $p^{\mathcal{M}}$ and $p$ have same functional form though they are defined in different domains. Moreover, $p^{\mathcal{M}}$ is a PDF on $\mathcal{M}$ since

$$\int_{\mathcal{M}} p^{\mathcal{M}}(\sigma_1, \sigma_2) dV = \iint_{\Omega} p(x, y) \sqrt{\det \mathbf{G}}\, dxdy = 1, \tag{28}$$

where $dV = \sqrt{\det \mathbf{G}}\, dxdy$ is the natural volume element of $\mathcal{M}$.

For any sub-domain $\mathcal{M}' \subseteq \mathcal{M}$, its counterpart on $\Omega$ is $\Omega' = \{\mathbf{\Psi}(\sigma_1, \sigma_2) | (\sigma_1, \sigma_2) \in \mathcal{M}'\} \subseteq \Omega$. Suppose the tuple $(\hat{\sigma_1}, \hat{\sigma_2})$ is a 2D random variable on $\mathcal{M}$ with density function $p^{\mathcal{M}}$, then $(\hat{x}, \hat{y}) = \mathbf{\Psi}(\hat{\sigma_1}, \hat{\sigma_2})$ is a 2D random variable on $\Omega$. And

$$\begin{aligned} Pr((\hat{x}, \hat{y}) \in \Omega') &= Pr((\hat{\sigma_1}, \hat{\sigma_2}) \in \mathcal{M}') \\ &= \int_{\mathcal{M}'} p^{\mathcal{M}}(\sigma_1, \sigma_2) dV \\ &= \iint_{\Omega'} p(x, y) \sqrt{\det \mathbf{G}}\, dxdy = \frac{|\Omega'|}{|\Omega|}, \end{aligned} \tag{29}$$

so $(\hat{x}, \hat{y})$ follows *uniform distribution* on $\Omega$. As a result, the set of uniform sampling positions in $\Gamma$ is obtained as follows: $M$ points on manifold $\mathcal{M}$ are drawn independently according to $p^{\mathcal{M}}$, then are mapped onto $\Omega$ through $\mathbf{\Psi}$.

With the above settings, from (6)(7)(25), graph $\mathcal{G}$ is built upon $M$ samples from manifold $\mathcal{M}$. Under the mapping $\mathbf{\Psi}$, the sampling positions are uniformly distributed on $\Omega$. As noticed

in [11],[20],[21],[22], the graph $\mathcal{G}$, which is a discrete object, is an approximation of the corresponding manifold $\mathcal{M}$.

For a continuous image $h$ on $\Omega$, its counterpart on $\mathcal{M}$ is $h^{\mathcal{M}}(\sigma_1, \sigma_2) = h(\mathbf{\Psi}(\sigma_1, \sigma_2))$, and its discrete sampling is $\mathbf{h}^D$. With mild conditions on $\epsilon$, manifold $\mathcal{M}$ and function $h$, from the convergence result of [11],

$$\lim_{\substack{M\to\infty \\ \epsilon\to 0}} \frac{M^{2\gamma-1}}{\epsilon^{4(1-\gamma)}(M-1)} S_{\mathcal{G}}(\mathbf{h}^D) \sim S_{\Delta}(h^{\mathcal{M}}), \qquad (30)$$

where "$\sim$" means there exist a constant only depends on $C_r$ such that equality holds. And the functional $S_{\Delta}$ is induced by the $2(1-\gamma)$-th weighted *Laplace-Beltrami operator* on manifold $\mathcal{M}$ with PDF $p^{\mathcal{M}}$. It can be written as

$$S_{\Delta}(h^{\mathcal{M}}) = \int_{\mathcal{M}} \left\langle \nabla h^{\mathcal{M}}, \nabla h^{\mathcal{M}} \right\rangle (p^{\mathcal{M}})^{2(1-\gamma)} dV. \qquad (31)$$

With (27), $S_{\Delta}(h^{\mathcal{M}})$ becomes

$$\begin{aligned} S_{\Delta}(h^{\mathcal{M}}) &= \iint_{\Omega} (\mathbf{G}^{-1}\nabla h)^{\mathrm{T}} \mathbf{G} (\mathbf{G}^{-1}\nabla h) p^{2(1-\gamma)} \sqrt{\det \mathbf{G}}\, dxdy \\ &= |\Omega|^{-2(1-\gamma)} \iint_{\Omega} (\nabla h)^{\mathrm{T}} \mathbf{G}^{-1} (\nabla h) \left(\sqrt{\det \mathbf{G}}\right)^{2\gamma-1} dxdy, \end{aligned} \qquad (32)$$

which equals $|\Omega|^{-2(1-\gamma)} S_{\Omega}(h)$ therefore (14) holds. ■

## REFERENCES


[1] P. Milanfar, "A tour of modern image filtering," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106–128, January 2013.

[2] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[3] X. Wu and X. Zhang, "Model-based non-linear estimation for adaptive image restoration," in *IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, April 2009, pp. 1185–1188.

[4] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *IEEE Int'l Workshop on Multimedia Signal Processing*, 2013.

[5] A. Kheradmand and P. Milanfar, "A general framework for kernel similarity-based image denoising," in *IEEE Global Conf. Sig. Info. Process.*, 2013, pp. 415–418.

[6] P. Wan, G. Cheung, D. Florencio, C. Zhang, and O. Au, "Image bit-depth enhancement via maximum-a-posterior estimation of graph ac component," in *IEEE Int'l Conf. Imag. Process. (to appear)*, 2014.

[7] D. Shuman, S. 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.

[8] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Int'l Conf. Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 60–65.

[9] H. G. Barrow and J. M. Tenenbaum, "Recovering intrinsic scene characteristics from images," *Computer Vision Systems*, pp. 3–26, 1978.

[10] X. Liu, D. Zhai, D. Zhao, G. Zhai, and W. Gao, "Progressive image denoising through hybrid graph laplacian regularization: a unified framework." *IEEE Trans. Imag. Process.*, vol. 23, no. 4, pp. 1491–1503, 2014.

[11] M. Hein, "Uniform convergence of adaptive graph-based regularization," in *Learning Theory*. Springer, 2006, pp. 50–64.

[12] H. Knutsson, C.-F. Westin, and M. Andersson, "Representing local structure using tensors ii," in *Image Analysis*. Springer, 2011, vol. 6688, pp. 545–556.

[13] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[14] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[15] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1445–1453, September 1988.

[16] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.

[17] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *IEEE Int'l Conf. Computer Vision*, 1998, pp. 839–846.

[18] N. Sochen, R. Kimmel, and R. Malladi, "A general framework for low level vision," *IEEE Trans. Imag. Process.*, vol. 7, no. 3, pp. 310–318, 1998.

[19] G. Rosman, X.-C. Tai, L. Dascal, and R. Kimmel, "Polyakov action minimization for efficient color image processing," *Trends and Topics in Computer Vision*, pp. 50–61, 2012.

[20] M. Hein, J.-Y. Audibert, and U. von Luxburg, "Graph laplacians and their convergence on random neighborhood graphs," *Journal of Machine Learning Research*, vol. 8, no. 6, 2007.

[21] M. Hein and M. Maier, "Manifold denoising," in *Advances in Neural Information Processing Systems*, 2006, pp. 561–568.

[22] D. Ting, L. Huang, and M. I. Jordan, "An analysis of the convergence of graph laplacians," in *Int'l Conf. Machine Learning*, 2010, pp. 1079–1086.