Nonlocal Discrete Regularization on Weighted Graphs: a framework for Image and Manifold Processing

Abderrahim Elmoataz, Olivier Lezoray, Sébastien Bougleux

Nonlocal Discrete Regularization on Weighted Graphs: a framework for Image and Manifold Processing

Abderrahim Elmoataz, Olivier Lezoray, Sébastien Bougleux

Abstract

We introduce a nonlocal discrete regularization framework on weighted graphs of the arbitrary topologies for image and manifold processing. The approach considers the problem as a variational one, which consists in minimizing a weighted sum of two energy terms: a regularization one that uses a discrete weighted p-Laplace operator, and an approximation one. This formulation leads to a family of simple and fast nonlinear processing methods, parameterized by the degree p of smoothness and by the graph weight function. This is a discrete analogue of recent well-known continuous nonlocal regularization. These discrete processing methods provide a graph-based version of recently proposed semi-local or nonlocal processing methods used in image and mesh processing, such as the bilateral filter, the TV digital filter or the nonlocal means filter. It works with equal ease on regular 2D-3D images, manifolds or any data. We illustrate the abilities of the approach by applying it to various types of images, meshes, manifolds and data represented as graphs.

Index Terms

Nonlocal discrete regularization, Weighted graph, p-Laplacian, Image and manifold processing.

The authors are with the University of Caen and the ENSICAEN in the GREYC Lab., Image Team, 6 Boulevard Maréchal Juin, F-14050 CAEN cedex FRANCE.

This research work was partially supported by the ANR foundation under grant ANR-06-MDCA-008-01/FOGRIMMI.

Email: abder.elmoataz@greyc.ensicaen.fr, olivier.lezoray@unicaen.fr, sebastien.bougleux@greyc.ensicaen.fr

Phone: +33(0)231472720, Fax: +00(0)231472698

I. INTRODUCTION

In many computer science applications, it is necessary to process images, meshes, manifolds and more generally data. This refers to the following research fields: image processing, computer graphics and data mining.

In the context of image processing, smoothing and denoising are key filtering processes. Among the existing methods, the variational ones, based on regularization, provide a general framework to design such efficient filter processes. Solutions of variational models can be obtained by minimizing appropriate energy functions. The minimization is usually performed by designing continuous Partial Differential Equations (PDE), whose solutions are discretized in order to fit with the image domain. A complete overview of these methods can be found in [1], [2], [3], [4] and references therein.

In the context of mesh processing, smoothing and denoising are also key processes dedicated to noise removal causing minimal damage to geometric features. Most of mesh smoothing methods are based on the discrete Laplace-Beltrami regularization or on the discrete curvature regularization [5], [6]. Other smoothing methods, based on feature preserving were mostly inspired by anisotropic diffusion in image processing [7], [8], [9]. Geometric flows have been extensively used in mesh processing [10]. In particular, surface flows based on functional minimization (i.e. evolving a surface so as to progressively decrease an energy functional) is a common methodology in geometric processing with applications spanning surface diffusion [11], denoising of scanned meshes [12], shape optimization and surface design [13], minimal surfaces [14], etc.

In the context of manifold and data processing, dimensionality reduction [15], [16], [17] (extracting low dimensional structure from high dimensional data), clustering [18], [19] (automatic identification of groups of similar objects) and classification [20], [21], [22] (assigning labels to instances) are key processes. In particular, methods based on graph Laplacian have became increasingly popular in machine learning to perform any of the above-mentioned key processes. However, manifolds and discrete data, as images and meshes, can also contain inner noise. Therefore, regularization occurs as a natural candidate for manifold processing [23], [24] since e.g. noise filtering can be useful to avoid overfitting in a learning process. Manifold regularization is also considered as a family of learning algorithms [20] based on regularization with both

labeled and unlabeled data. Transductive graph learning algorithms and standard methods like Support Vector Machines [22] can then be considered considered as special cases of manifold regularization. All these methods are based on the assumption that the data lies on a submanifold. However, sampled data lies almost never exactly on the submanifold due to the noise scattered around it. Since most of the previous methods are sensitive to noise, it is essential to denoise the manifold data to project it onto a submanifold [25].

As we have just mentioned it, regularization is a principle the interest of which concerns a wide range of computer science domains. The application of partial and variational methods to images, meshes, manifolds and any discrete data processing has shown its effectiveness allowing high quality regularization procedures. However, there are some limitations to the functionals used in regularization such as total variational ones [2]. Indeed, the latter are based on derivatives which consider local features of the data. Since the advent of the nonlocal means filter [1], the use of nonlocal interactions, to capture the complex structures of the data, has received a lot of interest and has shown to be very effective. A variational understanding of the nonlocal means filter was first developed in [26] as a non convex functional and later in [27] as a convex quadratic functional. Moreover, the use of manifolds to describe nonlocal interactions of geometric structures in signals and images has recently been explored [23]. In this latter work, diffusion over these manifolds is shown to be equivalent to nonlocal processing methods. However, most of these variational nonlocal formulations are expressed in a continuous setting and unfortunately the discretization of the underlying differential operators is difficult for high dimensional data.

Inspired by continuous regularization of images and recent works on nonlocal functional for continous regularization, we propose a general framework extending our previous works [28], [29], [30], [31] based on a nonlocal discrete regularization on weighted graphs of the arbitrary topologies. This framework can also be considered as a discrete analogue of the nonlocal continuous regularization for the case of images [32], [27]. Furthermore, the proposed framework works on any discrete data represented by weighted graphs which enables a nonlocal regularization with equal ease on images and manifolds. Our approach starts directly with a discrete variational problem and works with data living on any general discrete domain. To take into account the nonlocal interaction in images and manifolds, we explicitly introduce discrete nonlocal derivatives and functionals on graphs of the arbitrary topologies, to transcribe

the continuous regularization. This framework unifies the regularization of images, meshes, manifolds and data. The latter regularization enables local, semi-local or nonlocal regularization by using appropriated graphs topologies and edge weights. Let $G_w = (V, E)$ be a weighted graph consisting in a set of vertices V, a set of edges $E \subset V \times V$, and a similarity weight function w defined on edges. Let $\mathcal{H}(V)$ be a Hilbert space defined on the vertices of G_w. We formalize the discrete data regularization of a function $f^0 \in \mathcal{H}(V)$ by the following minimization problem:

$$\min_{f \in \mathcal{H}(V)} \left\{ E_w^p(f, f^0, \lambda) = \frac{1}{p} \sum_{v \in V} \|\nabla_w f\|^p + \frac{\lambda}{2} \|f - f^0\|^2_{\mathcal{H}(V)} \right\}, \tag{1}$$

where $p \in [1, +\infty)$ is the smoothness degree, λ is the fidelity parameter, and $\nabla_w f$ represents the weighted gradient of the function f over the graph. The solution of problem (1) leads to a family of nonlinear processing methods, parameterized by the weight function, the degree of smoothness, and the fidelity parameter. There exists two main advantages of using this framework, which can be considered as a discrete analogue of continuous regularization on weighted graphs.

First, the regularization is expressed directly in a discrete setting. Involved processing methods are computed by simple and efficient iterative algorithms, without solving any PDE. Second, the topology of graphs can be arbitrary. Since the proposed approach is general, any discrete data set can be transformed into a weighted graph, by using a similarity measure between data features. Thus, we can consider any function defined on these data as a function defined on the graph vertices.

Our formulation leads to a family of processing methods. For filtering, the family of filters we propose includes graph-based versions of well-known filters used in image and mesh processing. If $w = 1$, we show that our filters correspond exactly to digitized PDE filters, introduced in the context of image restoration on grid graphs [33] (L_2 digital filter for $p = 2$ and TV digital filter for $p = 1$). If $w \neq 1$, our family of filters behaves like weighted TV or weighted L_2 discrete regularizations. In particular, if $p = 2$ and $\lambda = 0$ with an appropriate graph topology, the choice of the weight function w allows to find the bilateral [34] and the nonlocal means filters [1]. In this particular case, we also show that the discrete regularization is linked to spectral graph theory [35], and to Markov matrix filtering [36]. We can quote other existing methods, developed in the context of image filtering, that can be considered as discrete regularizations on unweighted graphs [37] [38]. They yield to Markov random fields where only binary variables are involved in the minimization.

The solution of problem (1) is obtained by defining nonlocal functionals on weighted graphs. The idea of using functionals on graphs, in a regularization process, has also been proposed in other context, such as semi-supervised data learning [39] and image segmentation [40].

In this paper, we firstly define functionals on weighted graphs in Section II. Section III presents the discrete regularization problem (1) and the associated filter family. Section IV analyzes the obtained filters and gives relations to existing methods. In Section V, we show some regularization examples, in the context of images, meshes, manifolds and data processing. Last Section concludes.

II. Functionals on Weighted Graphs

In this section, we recall some basic definitions on graphs, and we define nonlocal functionals which can be considered as discrete versions of continuous differential operators. Analogue definitions and properties have also been used in the context of semi-supervised learning [39], and differential calculus on graphs [41], [42].

A. Preliminary Definitions

A *graph* $G_w = (V, E)$ consists in a finite set V of N vertices and a finite set $E \subseteq V \times V$ of edges. We assume G_w to be undirected, with no self-loops and no multiple edges. Let (u, v) be the edge that connects the vertices u and v. An undirected graph is *weighted* if it is associated with a weight function $w : E \rightarrow \mathbb{R}_+$ satisfying $w(u, v) = w(v, u)$, for all $(u, v) \in E$, and $w(u, v) = 0$ if $(u, v) \notin E$. The weight function represents a similarity measure between two vertices of the graph. We use the notation $u \sim v$ for two adjacent vertices. We say that G_w is connected whenever, for any pair of nodes (u, v) there is a finite sequence $u_0 = u, u_1, \cdots, u_n = v$ such that u_{i-1} is a neighbor of u_i for every $i = 1$,...,$i = n$.

Let $\mathcal{H}(V)$ denote the Hilbert space of real-valued functions on vertices. A function $f : V \rightarrow \mathbb{R}^m$ in $\mathcal{H}(V)$ assigns a vector $f(v)$ to each vertex $v \in V$. Clearly, f can be represented by a column vector of $\mathbb{R}^N$, $f = [f_1, \ldots, f_N]^T$. By analogy with functional space we define $\int_V f = \sum_V f(u)$.

The function space $\mathcal{H}(V)$ is endowed with the usual inner product $\langle f, h\rangle_{\mathcal{H}(V)} := \sum_{v \in V} f(v)h(v)$, where $f, h \in \mathcal{H}(V)$. Similarly, one can define $\mathcal{H}(E)$, the space of real-valued functions on edges,

endowed with the inner product $\langle F, H\rangle_{\mathcal{H}(E)} := \sum_{v\in V}\sum_{u\sim v} F(u,v)H(u,v)$, where F and H are two functions in $\mathcal{H}(E)$ defined on $E \to \mathbb{R}^+$.

B. Weighted Gradient and Divergence Operators

Let $G_w = (V, E)$ denote a weighted graph. The *difference operator* $d : \mathcal{H}(V) \to \mathcal{H}(E)$ of a function $f \in \mathcal{H}(V)$ on an edge $(u, v) \in E$, is defined by:

$$(df)(u,v) := \sqrt{w(u,v)}(f(v) - f(u)),\ \forall (u,v) \in E. \tag{2}$$

The *directional derivative* (or *edge derivative*) of a function $f \in \mathcal{H}(V)$ at a vertex v along an edge $e = (u, v)$, is defined as $\partial_v f_u := (df)(u, v)$. This definition is consistent with the continuous definition of the derivative of a function, e.g., if $f(v) = f(u)$ then $\partial_v f(u) = 0$. Moreover, one has $\partial_v f(u) = -\partial_u f(v)$ and $\partial_v f(v) = 0$.

The *weighted gradient operator* ∇_w of a function $f \in \mathcal{H}(V)$ at a vertex v is the vector operator defined by $\nabla_w f(v) = (\partial_v f(u) : u \sim v)^T$. The *local variation* of f at v, is defined to be:

$$\|\nabla_w f(v)\| := \sqrt{\sum_{u\sim v}(\partial_v f(u))^2} = \sqrt{\sum_{u\sim v} w(u,v)(f(v) - f(u))^2}. \tag{3}$$

It can be viewed as a measure of the regularity of a function around a vertex. The amplitude of the graph gradient is defined by $\|\nabla_w f\| := \langle \nabla_w f, \nabla_w f\rangle^{1/2}_{\mathcal{H}(E)}$.

The *adjoint operator* of the difference operator, denoted by $d^* : \mathcal{H}(E) \to \mathcal{H}(V)$, is defined by $\langle df, h\rangle_{\mathcal{H}(E)} := \langle f, d^*h\rangle_{\mathcal{H}(V)}$, with $f \in \mathcal{H}(V)$ and $h \in \mathcal{H}(E)$. Using the definitions of the inner products in $\mathcal{H}(V)$ and $\mathcal{H}(E)$ (see Section II-A), and definition (2), we obtain the expression of d^* at a vertex v:

$$(d^*h)(v) = \sum_{u\sim v} \sqrt{w(u,v)}(h(u,v) - h(v,u)). \tag{4}$$

The *divergence operator*, defined by $-d^*$, measures the net outflow of a function in $\mathcal{H}(E)$ at each vertex of V.

C. A Family of Weighted p-Laplace Operators

The *weighted p-Laplace operator*, $\Delta^p_w : \mathcal{H}(V) \to \mathcal{H}(V)$ with $1 \le p < +\infty$, is defined by $\Delta^p_w f := \frac{1}{p} d^*(\|\nabla_w f\|^{p-2} df)$. Substituting (2) and (4) into the definition of $\Delta^p_w f$, we obtain the

expression of Δ_w^p at a vertex v:

$$\Delta_w^p f(v) = \frac{1}{p} \sum_{u \sim v} \gamma(u,v) \left(f(v) - f(u) \right), \tag{5}$$

where γ is the function defined by $\gamma(u,v) := w_{uv} \left(\|\nabla_w f(v)\|^{p-2} + \|\nabla_w f(u)\|^{p-2} \right)$.

The operator Δ_w^p is nonlinear, with the exception of $p = 2$. Furthermore, Δ_w^p is positive semi-definite:

$$\begin{aligned} \langle f, p\Delta_w^p f \rangle_{\mathcal{H}(V)} &= \langle f, d^*(\|\nabla_w f\|^{p-2} df) \rangle_{\mathcal{H}(V)} = \langle df, \|\nabla_w f\|^{p-2} df \rangle_{\mathcal{H}(E)} \\ &= \sum_{v \in V} \|\nabla_w f(v)\|^{p-2} \sum_{u \sim v} ((df)(u,v))^2 = \sum_{v \in V} \|\nabla_w f(v)\|^p \geq 0. \end{aligned} \tag{6}$$

The definition of Δ_w^p can be considered as the discrete analogue of the p-Laplacian in the continuous case.

When $p = 2$, Δ_w^2 represents the *weighted Laplace operator* on G_w, and (5) reduces to:

$$\Delta f(v) := \Delta_w^2 f(v) = \sum_{u \sim v} w(u,v)(f(v) - f(u)). \tag{7}$$

When $p = 1$, Δ_w^1 represents the *weighted curvature operator* on G_w, and expression (5) reduces to:

$$\kappa f(v) := \Delta_w^1 f(v) = \frac{1}{2} \sum_{u \sim v} w(u,v) \left(\frac{1}{\|\nabla_w f(v)\|} + \frac{1}{\|\nabla_w f(u)\|} \right) (f(v) - f(u)). \tag{8}$$

In practice, to avoid zero denominator in (8), the local variation (3) is replaced by its regularized version: $\|\nabla_w f(v)\|_\epsilon := \sqrt{\|\nabla_w f(v)\|^2 + \epsilon^2}$, with $\epsilon \to 0$ a fixed small constant.

III. p-Laplace Regularization on Weighted Graphs

In this Section, we present the discrete regularization problem (1) and associated filters. Let $G_w = (V, E)$ be a weighted graph. The regularization of a given function $f^0 \in \mathcal{H}(V)$, using the weighted p-Laplace operator, consists in seeking for a function $f^* \in \mathcal{H}(V)$ which is not only smooth enough on G_w, but also close enough to f^0. This optimization problem can be formalized by the minimization of a weighted sum of two energy terms:

$$f^* = \min_{f \in \mathcal{H}(V)} \left\{ E_w^p(f, f^0, \lambda) := \frac{1}{p} \sum_{v \in V} \|\nabla_w f(v)\|^p + \frac{\lambda}{2} \|f - f^0\|_{\mathcal{H}(V)}^2 \right\}. \tag{9}$$

The first term in (9) is the smoothness term or regularizer, meanwhile the second is the fitting term. The parameter $\lambda \geq 0$ is a fidelity parameter, called the Lagrange multiplier, which specifies

the trade-off between the two competing terms. Both terms of the energy E_w^p are strictly convex functions of f. In particular, by standard arguments in convex analysis, the problem (9) has a unique solution, for $p = 1$ and $p = 2$, which satisfies:

$$\left.\frac{\partial E_w^p(f, f^0, \lambda)}{\partial f}\right|_v = \frac{1}{p}\frac{\partial}{\partial f}\|\nabla_w f(v)\|^p + \lambda(f(v) - f^0(v)) = 0, \ \ \forall v \in V. \tag{10}$$

From equation (6), the system of equations (10) is rewritten as:

$$\Delta_w^p f(v) + \lambda(f(v) - f^0(v)) = 0, \forall v \in V. \tag{11}$$

The solution of problem (9) is also the solution of the system of equations (11). This is a nonlinear system, with the exception of $p = 2$ (see Section II-C). Substituting the expression of the p-Laplace operator into (11), we obtain:

$$\left(\lambda + \frac{1}{p}\sum_{u\sim v}\gamma(u, v)\right) f(v) - \frac{1}{p}\sum_{u\sim v}\gamma(u, v)f(u) = \lambda f^0(v), \ \ \forall v \in V. \tag{12}$$

Among the existing methods to solve the system of equations (12), we use the Gauss-Jacobi iterative algorithm. Let t be an iteration step, and let $f^{(t)}$ be the solution of equation (12) at the step t. The corresponding linearized Gauss-Jacobi algorithm is given by:

$$\begin{cases} f^{(0)} = f^0 \\ \gamma^{(t)}(u, v) = w(u, v)\left(\|\nabla_w f^{(t)}(v)\|^{p-2} + \|\nabla_w f^{(t)}(u)\|^{p-2}\right), \ \ \forall (u, v) \in E \\ f^{(t+1)}(v) = \left(p\lambda + \sum_{u\sim v}\gamma^{(t)}(u, v)\right)^{-1}\left(p\lambda f^0(v) + \sum_{u\sim v}\gamma^{(t)}(u, v)f^{(t)}(u)\right), \ \ \forall v \in V \end{cases} \tag{13}$$

where $\gamma^{(t)}$ is the function γ at the step t. The weights $w(u, v)$ are computed from f^0, or can be given a priori. We define the function φ at an iteration t of algorithm (13) by:

$$\varphi^{(t)}(v, u) = \frac{\gamma^{(t)}(u, v)}{p\lambda + \sum_{u\sim v}\gamma^{(t)}(u, v)} \text{ if } u \neq v, \text{ and } \ \varphi^{(t)}(v, v) = \frac{p\lambda}{p\lambda + \sum_{u\sim v}\gamma^{(t)}(u, v)}$$

Then, an iteration of the regularization algorithm (13) is rewritten as:

$$f^{(t+1)}(v) = \varphi^{(t)}(v, v)f^0(v) + \sum_{u\sim v}\varphi^{(t)}(v, u)f^{(t)}(u), \ \ \forall v \in V. \tag{14}$$

At each iteration, the new value $f^{(t+1)}$, at a vertex v, depends on two quantities, the original value $f^0(v)$, and a weighted average of the existing values in a neighborhood of v. This shows that the proposed filter, obtained by iterating (14), is a low-pass filter which can be adapted to many graph structures and weight functions.

IV. Analysis and Examples

The case of an arbitrary p is not considered in this paper. In the sequel, we discuss the cases of $p = 2$ and $p = 1$, and we give some existing results related to our filter. When $p = 2$, it follows from equation (11) that the solution of problem (9) is based on the weighted Laplace operator defined by equation (7). Indeed, Equation (11) reduces to

$$\Delta_w^2 f^* + \lambda(f^* - f^0) = 0 \tag{15}$$

It can be thought as a discrete analogue of the Euler-Lagrange equation on weighted graphs. In this case, the iterative filter (13) is linear on the graph structure, and the coefficients given by the function γ do not have to be updated at each iteration because they only depend on the function w.

When $p = 1$, it follows from equation (11) that the solution of problem (9) is based on the weighted curvature operator defined by equation (8). Indeed, Equation (11) reduces to

$$\Delta_w^1 f^* + \lambda(f^* - f^0) = 0 \tag{16}$$

In this case, the iterative filter (13) is nonlinear, and the coefficients given by the function γ are adaptively updated at each iteration in addition of updating the function f.

One can note that Equations (16) and (15) correspond to the discrete gradient flow associated to the functional (9). For $p = 2$ and $p = 1$, our discrete model corresponds to the nonlocal continuous regularization where the regularization term is either [27]

$$J(u) = \frac{1}{4}\int_{\Omega\times\Omega} (f(x) - f(y))^2 w(x,y)dxdy \tag{17}$$

for $p = 2$ or

$$J(u) = \frac{1}{2}\int_{\Omega\times\Omega} \|f(x) - f(y)\|\, w(x,y)dxdy \tag{18}$$

for $p = 1$. Our framework is the discrete analogue of the nonlocal continuous regularization with the following general regularization term

$$J(u) = \frac{1}{2p}\int_{\Omega\times\Omega} \|f(x) - f(y)\|^p\, w(x,y)dxdy \tag{19}$$

The latter is also a nonlocal analogue of the energy functional associated to the continuous p-Laplacian:

$$F_p(f) = \frac{1}{p}\int_{\Omega} |\nabla(f(y))|^p\, dy \tag{20}$$

A. Regularization of Functions on Discrete Data

The family of filters presented in Section III can be used to regularize any function defined on discrete data by constructing a weighted graph, and by considering the function to be regularized as a function defined on graph vertices.

Let $V = \{v_1, \ldots, v_N\}$ be a finite data set such that v_i is a vector of $\mathbb{R}^m$. There exist several popular methods that transform the set V with a given pairwise similarity measure w into a graph $G_w = (V, E)$. Constructing similarity graphs consists in modelizing local neighborhood relationships between data points. Among the existing methods, we can quote *the ϵ-neighborhood graph* where two points $u, v \in V$ are connected by an edge if $\|v - u\| < \epsilon$, $\epsilon > 0$. Another important graph is *the k-nearest neighbors graph* where two points $u, v \in V$ are connected by an edge if u is among the k nearest neihgbors of v. We can also quote the complete graph, the minimum spanning tree, and the relative neighbor graph as other possible graph topologies. For images, classical graph representations are the grid graph and the region adjacency graph.

Let $f \in \mathcal{H}(V)$ be a function defined on each point of the data set V. Similarities between data points are estimated by comparing their features. Features generally depend on the function f and the set V. Every point $v \in V$ is assigned with a feature vector denoted by $F_f(v) \in \mathbb{R}^q$. Several choices can be considered for the feature vector $F_f(v)$. In the simplest case, one can consider $F_f(v) = f(v)$. $F_f(v)$ can also be a feature vector computed from $f(v)$ only or from $f(v)$ and $f(u)$ with $u \sim v$. Therefore, the weight function w associated to a graph can incorporate local, semi-local or nonlocal features according to the topology of the graph under consideration. The general formulation of a weight function is the following : $w(u, v) = g(F_f(u), F_f(v)), \forall (u, v) \in E$. For instance, one can consider the following general weight functions :

1) $g_1(F_f(u), F_f(v)) = \frac{1}{\varepsilon + \left\| F_f(u) - F_f(v) \right\|}$
2) $g_2(F_f(u), F_f(v)) = exp\left(- \frac{\left\| F_f(u) - F_f(v) \right\|^2}{\sigma^2} \right)$
3) $g_3(F_f(u), F_f(v)) = exp\left(- \frac{\left\| F_f(u) - F_f(v) \right\|^2}{\sigma^2} \right) exp\left(- \frac{\|u - v\|^2}{2\sigma_d^2} \right)$

where σ_d and σ are two parameters depending on the variations of $\|u - v\|$ and $\|F_f(v) - F_f(u)\|$ over the graph.

The graph structure, associated with one of the above weight functions, describe a general regularizer family. By changing the graph topology and the edge weights, we naturally obtain an expression of local, semi-local or nonlocal processing methods. For instance, in the context

of image processing, if one considers a grid graph with a weight function the feature vector of which is $F_f(v) = f(v)$, a typical local processing is obtained. If the graph topology is changed by increasing its connectivity by adding edges between nodes spatially close but not spatially connected, one obtains a semi-local processing. Finally, if one considers a feature vector as the set of values around a given vertex, one obtains a non-local processing. This nonlocal processing can be fully nonlocal by considering the complete graph of an image. From these considerations, we show that with an adapted graph topology and a given appropriate weight function, our general regularizer family is linked to several filters defined in the context of image and mesh processing. When $p = 2$, filter (13), associated with the weight function g_3, is equivalent to the bilateral filter, introduced in the context of image denoising [34] [43]. It is a nonlinear filter that combines geometric and range filtering. Bilateral filtering is also used to denoise meshes [44]. It is obtained by using the scalar feature $F_f(v) = f(v)$ for all $v \in V$. Using the same parameters, filter (13) can also be considered as a discrete nonlocal means filter, introduced in the context of images [1]. Indeed, it is obtained by using the weight function g_2 with the feature vector $F_f(v) = [f(u) : u \in B_{v,s}]^T$ and $B_{v,s}$ a bounding box of size s centered at v. The norm used in function g_2 is then $\|F_f(u) - F_f(v)\|^2_{2,s}$ which is an Euclidian distance weighted by a Gaussian Kernel.

When $\lambda \neq 0$ and w is constant, filter (13) corresponds exactly to the digitized PDE filters proposed in the context of image restoration [33]. If $p = 1$, it is the TV regularization. If $p = 2$, it is the L_2 regularization. In general, if the weight function is not constant, filter (13) corresponds to the weighted L_2 and the weighted TV discrete regularizations on arbitrary graphs.

B. Relationships with Spectral Graph Filtering

We consider the regularization problem (9) for $p = 2$ and $\lambda = 0$, and we show that it can be expressed in terms of spectral graph theory [35]. From expression (14), the filter reduces to:

$$f_v^{(t+1)} = \sum_{u \sim v} \varphi_{vu} f_u^{(t)}, \ \forall v \in V, \tag{21}$$

where $\varphi_{vu} = w_{uv} / \sum_{u \sim v} w_{uv}, \ \forall uv \in E$. As we have $\varphi_{vu} \geq 0$ and $\sum_{u \sim v} \varphi_{vu} = 1$, φ_{vu} can be interpreted as the probability of a random walker to jump from v to u in a single step [36]. Let P be the $N \times N$ Markov matrix defined by: $P(v, u) = \varphi_{vu}$ if the edge $uv \in E$, and $P(v, u) = 0$ otherwise. Let F be the matrix form of the function f. With these notations, expression (21) is

rewritten as:

$$F^{(t+1)} = PF^{(t)} = P^t F^{(0)}. \tag{22}$$

An element $P^t(v,u)$, $vu \in E$, describes the probability of transition in t steps. The matrix P^t encodes local similarities between vertices of the graph and it diffuses or integrates this local information for t steps to larger and larger neighborhoods of each vertex.

The spectral decomposition of the matrix P is given by $P\phi_i = \lambda_i \phi_i$, with $1 \geq \lambda_1 \geq \ldots \geq \lambda_i \geq \ldots \geq \lambda_N \geq 0$ the eigenvalues of P, and ϕ_i its eigenvectors. The eigenvectors associated with the k first eigenvalues contain the principal information. Thus, an equivalent way to look at the power of P in filter (22) is to decompose each value of F on the first eigenvectors of P. Moreover, the eigenvectors of the matrix P can be seen as an extension of the Fourier transform basis functions with eigenvalues representing frequencies [45]. This defines a basis of any function f in $\mathcal{H}(V)$, and the function f can be decomposed on the k first eigenvectors of P as: $f = \sum_{i=1}^{i=k} \langle f, \phi_i \rangle_{\mathcal{H}(V)} \phi_i$. This can be interpreted as a filtering in the spectral domain.

V. Applications

The family of filters proposed in Section III can be used to regularize any function defined on the vertices of a graph, or on any discrete data set. Through examples, we show its efficiency in the case of image denoising, image simplification, polygonal curve denoising, surface mesh denoising, manifold and data simplification.

For any of these different applications, one wants to regularize a function $f^0 : V \subset \mathbb{R}^n \to \mathbb{R}^m$. When the regularization operates on vector-valued vertices, one regularization process per vector component is considered. Therefore, for vector-valued vertices, the regularization can be applied on each channel leading to an iteration of filter (13) rewritten as:

$$\left[\begin{array}{c} f_1 \\ \vdots \\ f_m \end{array}\right]_v^{(t+1)} = \left(p\lambda + \sum_{u\sim v} \gamma_{uv}^{(t)}\right)^{-1} \left(p\lambda \left[\begin{array}{c} f_1 \\ \vdots \\ f_m \end{array}\right]_v^{0} + \sum_{u\sim v} \gamma_{uv}^{(t)} \left[\begin{array}{c} f_1 \\ \vdots \\ f_m \end{array}\right]_u^{(t)} \right), \tag{23}$$

where $\gamma_{uv}^{(t)}$ depends on the norm of the p-Laplace operator defined by:

$$\|\nabla_w f(v)\| = \sqrt{\sum_{i=1}^{m} \|\nabla_w f_i(v)\|^2}. \tag{24}$$

This norm takes into account the inner correlation aspect of vector data (in the case of $p \neq 2$).

Image Smoothing and Denoising : Let f^0 be a color image of N pixels, $f^0 = [f_1^0, \ldots, f_N^0]^T$ with $f^0 : V \subset \mathbb{Z}^2 \rightarrow \mathbb{R}^3$ which defines a mapping from the vertices to a vector of color channels. Therefore, this comes to consider three independent regularization processes. Figure 1 presents

(a) Noisy image ($\sigma = 15$). (b) $p = 2$, local processing. (c) $p = 1$, local processing. (d) $p = 1$, nonlocal processing (3×3 patch). (e) $p = 1$, nonlocal processing (5×5 patch).

Fig. 1. Image denoising illustration for different values of p for local and nonlocal processing.

several results of image denoising on an original image corrupted by Gaussian noise ($\sigma = 15$, Figure 1(a)). Figures 1(b) and 1(c) present the result of a local processing with a 8-grid graph with edges weighted by g_2 and $F_f(v) = f(v)$. Figure 1(c) presents the result of a nonlocal processing on a 48-grid graph (a search window of size 7×7) with edges weighted by g_2 and $F_f(v)$ defined on a window of size 3×3. Figure 1(d) presents the result of a nonlocal processing on a 288-grid graph (a search window of size 17×17) with edges weighted by g_2 and $F_f(v)$ defined on a window of size 5×5. The parameters λ and σ are estimated (see in [30] for further details). Figure 2 presents several results of image smoothing on a real image with $p = 1$ with edges weighted by g_2. Figure 2(b) shows a local processing on a 8-grid graph with $F_f(v) = f(v)$. Figure 2(c) shows a nonlocal processing on a 24-grid graph (search window of size 5×5) with $F_f(v)$ defined on a window of size 3×3. Figure 2(d) shows the same nonlocal processing with a Chromaticity-Brightness representation.

Image Simplification : On the contrary to classical image simplification which considers grid graphs, one can simplify an image by first considering a fine partition of this image (or over-

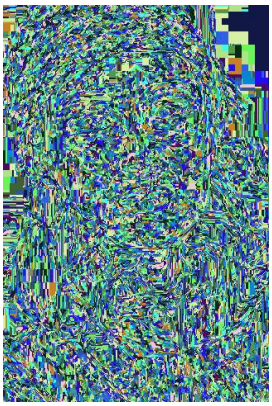
(a) Original noisy image. (b) Local processing. (c) nonlocal processing. (d) nonlocal processing with a Chrominaticity-Brightness representation.

Fig. 2. Image smoothing illustration on a real image ($p = 1$, other parameters are estimated).

segmentation), where the pixel values of each region of the partition are replaced by the mean or the median pixel value of this region.

The partition can be associated with a region adjacency graph (RAG), where vertices represent regions and where edges link adjacent regions. Let $G_w = (V, E)$ be a RAG. Let $f^0 : V \subset \mathbb{Z}^2 \rightarrow \mathbb{R}^m$ be a mapping from the vertices of G_w to the mean or median value of their regions. Then, the simplification is achieved by regularizing the function f^0 on G_w. Figure 3 presents such an image simplification on an original image (Figure 3(a)) to which a fine partition is associated (Figure 3(b)). The concept of homogeneous zones extraction was used to build the fine partition [46]. Figure 3(c) presents the color median image of Figure 3(b). A simplification is performed with $p = 2$ and $\lambda = 0$ or $\lambda = 0.5$. Edges of the RAG are weighted by g_1. The graph is simplified along the iterations by merging regions the difference of which is lower than 2. The use of the RAG instead of the classical grid-graph enables a faster processing [28]. Moreover, merging similar regions during the simplification enables to decimate the RAG under study and this accelerates the simplification processing. Indeed, the original image has 154401 pixels whereas the RAG has 41212 vertices; the PSNR between the color median image (Figure 3(c)) and the

(a) Original Image.

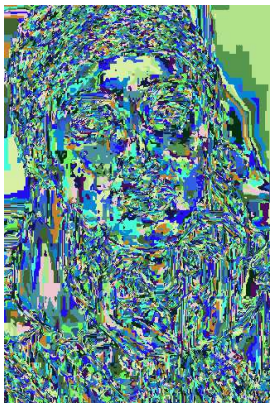


(b) Region map of (a).

(c) Color median image of (b).

(d) Region map after regularization ($p = 2$, $\lambda = 0$).

(e) Color median image of (d) ($p = 2$, $\lambda = 0$).



(f) Region map ($p = 2$, $\lambda = 0.5$).

(g) Color median image of (f) ($p = 2$, $\lambda = 0.5$).

Fig. 3. Image simplification illustration. From an original image (a), one computes a pre-segmentation (b) and the associated color median image (c). Regularization is applied on the RAG of the pre-segmentation and simplified region maps (Figures (d) and (f)) and color median (Figures (e) and (g)) images are obtained for different values of λ.

original image (Figure 3(a)) is 20.60db. For the case of $p = 2$ and $\lambda = 0$, only 22412 vertices survived (Figure 3(d)) and the PSNR between the color median image (Figure 3(e)) and the original one is 20.49db. With $p = 2$ and $\lambda = 0.5$, 34412 vertices survived (Figure 3(f)) and the PSNR between the color median image (Figure 3(g)) and the original one is 20.59db. Performing the regularization on the RAG is a faster alternative to usual image simplification a RAG has less nodes than a grid graph and its structure can be simplified along the iterations.

Mesh Smoothing and Denoising : By nature, polygonal curves and surface meshes have a graph structure. First, we consider the case of a polygonal curve represented by a graph: $f^0 : V \subset \mathbb{R}^2 \to \mathbb{R}^2$ and edges are weighted by function g_1. Figure 4 presents results of filter (23) on a polygonal curve.

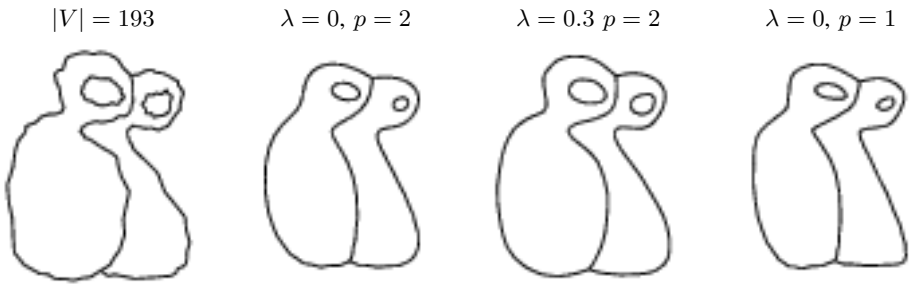
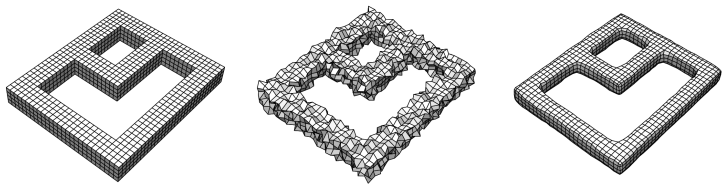


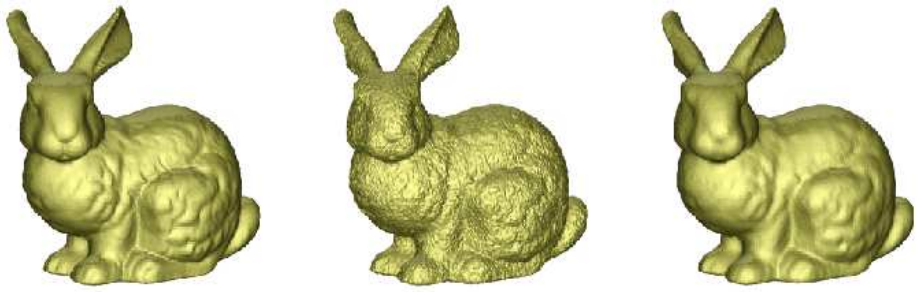
Fig. 4. Polygonal curve denoising. Edges of polygons are weighted with g_1. The regularization of vertices position is performed in 10 steps. When $\lambda = 0$, the regularization introduces shrinkage effects. They are reduced using a value of $\lambda \neq 0$.

Second, we consider the case of surface meshes. Let V be the set of mesh vertices, and let E be the set of mesh edges. If the input mesh is noisy, we can regularize vertex coordinates or any other function $f^0 : V \subset \mathbb{R}^3 \to \mathbb{R}^3$ defined on the graph $G_w = (V, E)$. The problem of mesh smoothing is to approximate (by a regularization process) the 3D coordinated functions mapping each vertex to each point. Results of filter (23) are given in Figure 5 and Figure 6 for quadrangular and triangular meshes. Edges are weighted by a function g_1. Of course any other weight functions could have been used.



(a) Original mesh. (b) Normal noise. (c) $p = 1$, $\lambda = 0.25$.

Fig. 5. Mesh denoising by regularizing the position of vertices ($t = 10$ and $w = g_1$).



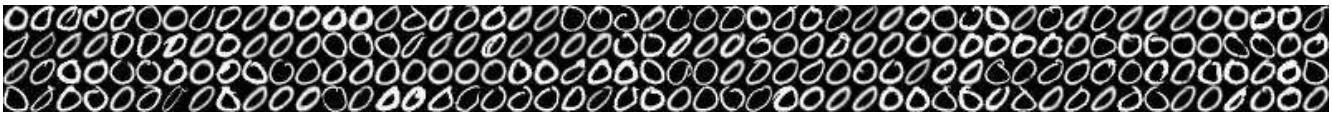
(a) Original mesh. (b) Normal noise. (c) $p = 2$, $\lambda = 0.5$.

Fig. 6. Mesh denoising. (a) Original Stanford Bunny with $|V| = 35949$. (b) Noisy Bunny with normal noise. (c) Regularization of vertex coordinates on mesh edges in 8 iterations. We use the weight function g_1 with the scalar feature $F_f(v) = f_v$.

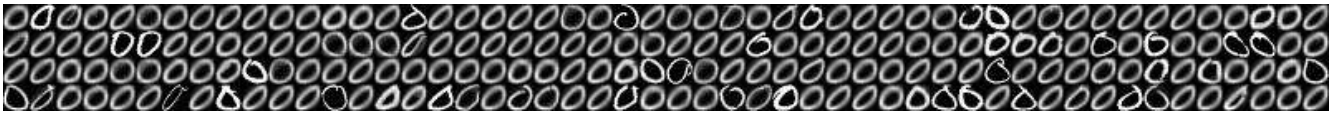
Manifold Smoothing : Graph-based methods have recently emerged as a powerful tool for analyzing high dimensional data that has been sampled from a low dimensional submanifold [20], [17]. These methods begin by constructing a graph in which the nodes represent input patterns and the edges represent neighborhood relations. The resulting graph (assumed connected) can be viewed as a discretized approximation of the submanifold sampled by the input patterns. Let $G_w = (V, E)$ be a graph defined over the manifold. Typical graphs are the complete graph and the k-nearest neighbor graph. Let $f^0 : V \to \mathbb{R}^{16\times 16}$ be a mapping from the vertices of G_w to the elements of the manifold. Typical manifolds being image libraries, we consider the USPS handwritten digit database for illustration. Each digit is a 16×16 image which is considered as a vector of 256 dimensions. We consider the complete graph weighted by g_2 and constructed over the manifold. To each vertex is associated a feature vector representing each digit. The filter (23) is considered (256 parallel regularization processes are performed) with $p = 2$ and different amounts of data attachment. Since the complete graph is considered, the regularization is naturally nonlocal. Figure 7 presents regularization results on 200 digits from the USPS database. Without the data term, the manifold reduces to its main digit which is an artificial one since the manifold is smoothed. By increasing the data term, the regularized manifold remains closer to the original one. Such manifold regularization can be useful for classification purposes on a noiseless submanifold [25] extracted from a noisy manifold.

Data Smoothing : Given any data, the latter can be associated with a graph by considering

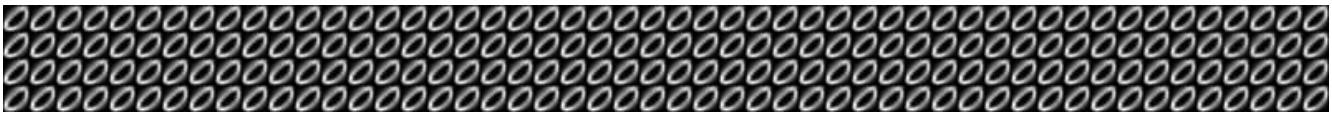
(a) Original USPS handwritten digit 0 database.



(b) Nonlocal regularization with $\lambda = 1$.



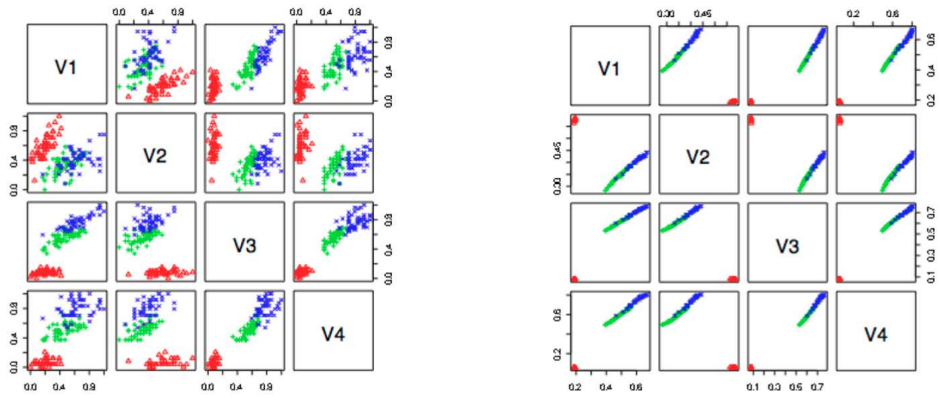
(c) Nonlocal regularization with $\lambda = 0.01$.



(d) Nonlocal regularization with $\lambda = 0$.

Fig. 7. Nonlocal manifold regularization ($p = 2$, $\lambda = 0$).

a similarity measure. Therefore, any data base can be regularized with our method as long as it is associated with a graph structure. To show the efficiency of such processing, we consider the well-known Iris database [47], where each element of the database is described by a 4-dimension feature vector. In this data base, three different classes of points are available. Figure 8(a) presents a pairwise feature projection of this database. Let $G_w = (V, E)$ be a graph defined over the data. Let $f^0 : V \to \mathbb{R}^4$ be a mapping from the vertices of G_w to the elements of the data base. We consider a complete graph weighted by g_2. The regularization is performed with $p = 2$ and $\lambda = 0$. Figure 8(b) presents the regularization result on the data of Figure 8(a). One can see the benefits of the regularization: input points which belong to the same class tend to be closer than in the original data base. Therefore, this is an efficient method to map input points into a regularized space where clusters are more easily separable: the submanifold where the data lies has been recovered.



(a) Original Iris data base.

(b) Regularized Iris data base.

Fig. 8. Nonlocal Data base regularization ($p = 2$, $\lambda = 0$).

VI. Conclusion

We propose a general discrete framework for regularizing real-valued or vector-valued functions on weighted graphs of arbitrary topology. The regularization, based on the p-Laplace operator, leads to a family of nonlinear iterative filters. This family includes the TV digital filter, the nonlocal means filter and the bilateral filter, both widely used in image processing. Also, the family is linked to spectral graph filtering and is the discrete analogue of the recent continuous nonlocal regularization.

The choice of the graph topology and the choice of the weight function allow to regularize any discrete data set or any function on a discrete data set. Indeed, the data can be structured by neighborhood graphs weighted by functions depending on data features. This can be applied in the context of image smoothing, denoising or simplification. We also show that mesh smoothing and denoising can be performed by the same filtering process. Similarly, manifolds can be processed by the same means to recover a noiseless submanifold form a noisy manifold.

References

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

[2] T. Chan and J. Shen, *Image Processing and Analysis - variational, PDE, wavelets, and stochastic methods*. SIAM, 2005.

[3] Y.-H. R. Tsai and S. Osher, "Total variation and level set methods in image science," *Acta Numerica*, vol. 14, pp. 509–573, mai 2005.

[4] L. Alvarez, F.Guichard, P.-L. Lions, and J.-M. Morel, "Axioms and fundamental equations of image processing," *Archive for Rational Mechanics and Analysis*, vol. 123, no. 3, pp. 199–257, 1993.

[5] G. Taubin, "Geometric signal processing on polygonal meshes," in *Eurographics, State of the Art Report*, 2000.

[6] Y. Ohtake, A. Belyaev, and I. Bogaeski, "Mesh regularization and adaptive smoothing," *Computer-Aided Design*, vol. 33, pp. 789–800, 2001.

[7] M. Desbrun, M. Meyer, P. Schröder, and A. Barr, "Anisotropic feature-preserving denoising of height fields and bivariate data," *Graphics Interface*, 2000.

[8] C. Bajaj and G. Xu, "Anisotropic diffusion of surfaces and functions on surfaces," *ACM Trans. on Graph.*, vol. 22, no. 1, pp. 4–32, 2003.

[9] K. Hildebrandt and K. Polthier, "Anisotropic filtering of non-linear surface features," *Eurographics 2004: Comput. Graph. Forum*, vol. 23, no. 3, pp. 391–400, 2004.

[10] I. Eckstein, J.-P. Pons, Y. Tong, C. Kuo, and M. Desbrun, "Generalized surface flows for mesh processing," in *Eurographics Symposium on Geometry Processing*, 2007.

[11] G. Taubin, "A signal processing approach to fair surface design." in *SIGGRAPH*, 1995, pp. 351–358.

[12] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow." in *SIGGRAPH*, 1999, pp. 317–324.

[13] G. Xu, Q. Pan, and C. L. Bajaj, "Discrete surface modelling using partial differential equations." *Computer Aided Geometric Design*, vol. 23, no. 2, pp. 125–145, 2006.

[14] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," *Experimental Mathematics*, vol. 2, no. 1, 1993.

[15] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation." *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[16] S. Lafon, Y. Keller, and R. R. Coifman, "Data fusion and multicue data matching by diffusion maps." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1784–1797, 2006.

[17] S. Lafon and A. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning and data set parameterization." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1393–1403, 2006.

[18] U. von Luxburg, "A tutorial on spectral clustering," Max Planck Institute for Biological Cybernetics, Tech. Rep. 149, 2006, to appear in Statistics and Computing.

[19] J. Shi and J. Malik, "Normalized cuts and image segmentation." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[20] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *COLT*, 2004, pp. 624–638.

[21] D. Zhou and B. Scholkopf, "A regularization framework for learning from graph data," in *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*, 2004, pp. 132–137.

[22] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *COLT: Proc of the 16th Annual Conf. on Computational Learning Theory*, B. Schölkopf and M. Warmuth, Eds., 2003, pp. 144–158.

[23] G. Peyré, "Manifold models for signals and images," ceremade, preprint 17, 2007.

[24] A. Szlam, M. Maggioni, and R. Coifman, "A general framework for adaptive regularization based on diffusion processes on graphs," Yale, Tech. Rep., 2006.

[25] M. Hein and M. Maier, "Manifold denoising," in *Advances in Neural Information Processing Systems*, vol. 20, 2006, pp. 1–8.

[26] S. Kindermann, S. Osher, and P. Jones, "Deblurring and denoising of images by nonlocal functionales," UCLA, Tech. Rep., 2005.

[27] G. Gilboa and S. Osher, "Nonlocal linear image regularization and supervised segmentation," *SIAM - Multiscale Modeling and Simulation (MMS)*, vol. to appear, 2007.

[28] O. Lezoray, A. Elmoataz, and S. Bougleux, "Graph regularization for color image processing," *Computer Vision and Image Understanding (CVIU)*, vol. 107, no. 1-2, pp. 38–55, 2007.

[29] S. Bougleux, M. Melkemi, and A. Elmoataz, "Discrete regularization on weighted graphs for image and mesh filtering," in *International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*, ser. LNCS, Springer, Ed., vol. 4485, 2007, pp. 128–139.

[30] O. Lezoray, S. Bougleux, and A. Elmoataz, "Parameterless discrete regularization on graphs for color image filtering," in *ICIAR*, ser. Lecture Notes in Computer Science, Springer, Ed., 2007.

[31] V. Ta, O. Lezoray, and A. Elmoataz, "Graphs regularization for data sets and images: Filtering and semi-supervised classification," in *PASCAL Workshop: Learning from and with graphs, 6th IAPR -TC-15 Workshop on Graph based Representations in Pattern Recognition*, 2007.

[32] G. Gilboa, J. Darbon, S. Osher, and T. Chan, "Nonlocal convex functionals for image regularization," UCLA, CAM Report 06-57, October 2006.

[33] S. Osher and J. Shen, "Digitized PDE method for data restoration," in *In Analytical-Computational methods in Applied Mathematics*, E. G. A. Anastassiou, Ed. Chapman&Hall/CRC, 2000, pp. 751–771.

[34] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV'98: Proc. of the 6th Int. Conf. on Computer Vision*. IEEE Computer Society, 1998, pp. 839–846.

[35] F. R. Chung, "Spectral graph theory," *CBMS Regional Conference Series in Mathematics*, vol. 92, pp. 1–212, 1997.

[36] R. Coifman, S. Lafon, M. Maggioni, Y. Keller, A. Szlam, F. Warner, and S. Zucker, "Geometries of sensor outputs, inference, and information processing," in *Proc. of the SPIE: Intelligent Integrated Microsystems*, vol. 6232, 2006.

[37] A. Chambolle, "Total variation minimization and a class of binary MRF models," in *LNCS 3757, Proc. of the 5th Int. Work. EMMCVPR*. Springer-Verlag, 2005, pp. 136–152.

[38] J. Darbon and M. Sigelle, "Exact optimization of discrete constrained total variation minimization problems," in *LNCS 3322: Proc. of the 10th Int. Workshop on Combinatorial Image Analysis*, R.Klette and J. Zunic, Eds., 2004, pp. 548–557.

[39] D. Zhou and B. Schölkopf, "Regularization on discrete spaces," in *LNCS 3663, Proc. of the 27th DAGM Symp.* Springer-Verlag, august 2005, pp. 361–368.

[40] S. Bougleux and A. Elmoataz, "Image smoothing and segmentation by graph regularization." in *LNCS 3656, Proc. Int. Symp. on Visual Computing*. Springer-Verlag, 2005, pp. 745–752.

[41] A. Bensoussan and J.-L. Menaldi, "Difference equations on weighted graphs," *Journal of Convex Analysis*, vol. 12, no. 1, pp. 13–44, 2005.

[42] J. Friedman and J.-P. Tillich, "Wave equations for graphs and the edge-based laplacian," *Pacific Journal of Mathematics*, vol. 216, no. 2, pp. 229–266, 2004.

[43] D. Barash, "A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 844–847, 2002.

[44] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *ACM Trans. on Graphics*, vol. 22, no. 3, pp. 950–953, 2003.

[45] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data," *Proc. of the National Academy of Sciences*, vol. 102, no. 21, 2005.

[46] C. Meurie and O. Lezoray, "A new method of morphological hierarchical segmentation," in *Visual Information Engineering*, IEEE and IET, Eds., 2007.

[47] S. Hettich, C. Blake, and C. Merz, "UCI repository of machine learning databases," University of California, Irvine, Dept. of Information and Computer Sciences, Tech. Rep., 1998. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html