Discrete Regularization on Weighted Graphs for Image and Mesh Filtering

Sébastien Bougleux, Abderrahim Elmoataz, Mahmoud Melkemi

Discrete Regularization on Weighted Graphs for Image and Mesh Filtering

Sébastien Bougleux[1], Abderrahim Elmoataz[2], and Mahmoud Melkemi[3]

[1] GREYC CNRS UMR 6072 - Équipe Image,
ENSICAEN, 6 BD du Maréchal Juin, 14050 Caen Cedex France
sebastien.bougleux@greyc.ensicaen.fr
[2] Université de Caen Basse-Normandie, GREYC CNRS UMR 6072
6 BD du Maréchal Juin, 14050 Caen Cedex France
abder.elmoataz@greyc.ensicaen.fr
[3] Université de Haute-Alsace, LMIA - MAGE
4 rue des Frères Lumière, 68093 Mulhouse Cedex France
mahmoud.melkemi@uha.fr

October 2006 (preprint submitted to SSVM 2007)

Abstract. We propose a discrete regularization framework on weighted graphs of arbitrary topology, which unifies image and mesh filtering. 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 the discrete p-Laplace operator, and an approximation one. This formulation leads to a family of simple nonlinear filters, parameterized by the degree p of smoothness and by the graph weight function. Some of these filters provide a graph-based version of well-known filters used in image and mesh processing, such as the bilateral filter, the TV digital filter or the nonlocal mean filter.

1 Introduction

In many computer vision applications, it is necessary to filter and to simplify images or meshes. In the context of image processing, smoothing and denoising constitute important steps of filtering processes. Among the existing methods, the variational ones, based on regularization, provide a general framework to design efficient filters. Solutions of variational models can be obtained by minimizing appropriate energy functions. The minimization is usually performed by designing continuous partial differential equations (PDEs), 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.

Another important problem of computer vision is mesh smoothing or denoising. This process is 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].

Variational Beltrami flows have also been used to denoise and regularize data defined on manifolds [7]. Other mesh smoothing methods, based on feature preserving, were mostly inspired by anisotropic diffusion in image processing [8][9][10].

Inspired by continuous regularization of images, we propose a general framework based on a discrete regularization on weighted graphs of arbitrary topology. This framework unifies the regularization of images and meshes. Let $G_w = (V, E)$ be a weighted graph which consists of 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 regularization of a function $f^0 \in \mathcal{H}(V)$ by the following minimization problem:

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

where $p \in [1, +\infty)$ is the smoothness degree, λ is the fidelity parameter, and $\bigtriangledown f$ represents the weighted gradient of the function f over the graph. The solution of problem (1) leads to a family of nonlinear filters, parameterized by the weight function, the degree of smoothness and the fidelity parameter. There exist two main advantages of using this framework, which can be considered as a discrete analogue of continuous regularization on weighted graphs. Firstly, the regularization is expressed directly in discrete settings. Secondly, filters are computed by simple and efficient iterative algorithms, without solving any PDEs. 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. Thus, we can consider any function defined on these data as a function defined on the vertices of the graph.

The family of filters we propose includes graph-based versions of well-known filters used in image and mesh processing. If $w = 1$, they correspond exactly to the digitized PDE filters, introduced in the context of image restoration on grid graphs [11] (L_2 digital filter for $p = 2$ and TV digital filter for $p = 1$). If $w \neq 1$, our the filters behave like weighted TV regularization or weighted L_2 regularization. In particular, if $p = 2$ and $\lambda = 0$, the choice of the weight function w allows to find the bilateral filter [12] and the nonlocal mean filters [1]. In this particular case, we also show that the discrete regularization is linked to spectral graph theory [13] and to Markov matrix filtering [14]. We can quote other existing methods, developed in the context of image filtering, that can be considered as discrete regularizations on unweighted graphs [15] [16]. These regularizations yield to Markov random fields where only binary variables are involved in the minimization.

The solution of problem (1) is obtained by defining local differential geometry operators on weighted graphs. The idea of using differential geometry on graphs, in a regularization process, has also been proposed in other context, such as semi-supervised data learning [17] and image segmentation [18].

In this paper, we firstly define differential geometry operators on weighted graphs in Section 2. Section 3 presents the regularization problem (1) and the

associated family of filters. Section 4 analyzes the obtained filters and gives relations to existing methods. In Section 5, we show some regularization examples, in the context of image denoising, image segmentation and mesh denoising.

2 Differential Geometry on Weighted Graphs

In this section, we recall some basic definitions on graphs, and we define local differential geometry operators 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 [17], and differential calculus on graphs [19][20].

2.1 Preliminary Definitions

A *graph* $G_w = (V, E)$ consists of 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 uv 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 $w_{uv} = w_{vu}$, for all $uv \in E$, and $w_{uv} = 0$ if $uv \notin E$. The weight function represents a similarity measure between two vertices of the graph.

Let $\mathcal{H}(V)$ denotes the Hilbert space of real-valued functions on vertices. A function $f : V \rightarrow \mathbb{R}$ 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 $\mathbb{R}^N$, $f = [f_1, \ldots, f_N]^T$. 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 $\langle f, h\rangle_{\mathcal{H}(E)} := \sum_{v \in V} \sum_{u \sim v} f_{uv} h_{uv}$, where f and h are two functions in $\mathcal{H}(E)$.

2.2 Weighted Gradient and Divergence Operators

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

$$(df)_{uv} := \sqrt{w_{uv}}(f_v - f_u), \ \forall uv \in E. \tag{2}$$

The *directional derivative* (or *edge derivative*) of a function $f \in \mathcal{H}(V)$ at a vertex v along an edge $e = uv$, is defined as $\partial_v f_u := (df)_{uv}$. 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, note that $\partial_v f_u = -\partial_u f_v$, and $\partial_v f_v = 0$.

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

$$\|\bigtriangledown_v f\| := \sqrt{\sum_{u \sim v} (\partial_v f_u)^2} = \sqrt{\sum_{u \sim v} w_{uv} (f_v - f_u)^2}. \tag{3}$$

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 $\| \bigtriangledown f\| := \langle \bigtriangledown f, \bigtriangledown f \rangle^{1/2}_{\mathcal{H}(E)}$.

The *adjoint operator* of the difference operator, denoted by $d^* : \mathcal{H}(E) \rightarrow \mathcal{H}(V)$, 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 2.1), and definition (2), we obtain the expression of d^* at a vertex v:

$$(d^*h)_v = \sum_{u\sim v} \sqrt{w_{uv}}(h_{uv} - h_{vu}). \tag{4}$$

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

2.3 A Family of Weighted p-Laplace Operators

The *weighted p-Laplace operator*, $\Delta_p : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ with $1 \leq p < +\infty$, is defined by $\Delta_p f := d^*(\| \bigtriangledown f\|^{p-2} df)$. Substituting (2) and (4) into the definition of $\Delta_p f$, we obtain the expression of Δ_p at a vertex v:

$$(\Delta_p f)_v = \sum_{u\sim v} \gamma_{uv} \left(f_v - f_u\right), \tag{5}$$

where γ is the function defined by $\gamma_{uv} := w_{uv} \left(\| \bigtriangledown_v f\|^{p-2} + \| \bigtriangledown_u f\|^{p-2}\right)$.

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

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

The definition of Δ_p can be considered as the discrete analogue of the p-Laplacian in the continuous case. When $p = 2$, Δ_2 represents the *weighted Laplace operator* on G_w, and (5) reduces to:

$$(\Delta f)_v := (\Delta_2 f)_v = 2 \sum_{u\sim v} w_{uv}(f_v - f_u). \tag{7}$$

When $p = 1$, Δ_1 represents the *weighted curvature operator* on G_w, and expression (5) reduces to:

$$(\kappa f)_v := (\Delta_1 f)_v = \sum_{u\sim v} w_{uv} \left(\frac{1}{\| \bigtriangledown_v f\|} + \frac{1}{\| \bigtriangledown_u f\|}\right) (f_v - f_u). \tag{8}$$

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

3 *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:

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

The first energy 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 energy functions in E_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:

$$\left. \frac{\partial E_p(f, f^0, \lambda)}{\partial f} \right|_v = \frac{\partial}{\partial f} \| \bigtriangledown_v f \|^p + 2\lambda (f_v - f^0_v) = 0, \ \ \forall v \in V. \tag{10}$$

Using equation (6) to compute the derivative of the first term in E_p, the system of equations (10) is rewritten as:

$$(\Delta_p f)_v + 2\lambda (f_v - f^0_v) = 0, \forall v \in V. \tag{11}$$

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 2.3). Substituting the expression of the p-Laplace operator into (11), we obtain:

$$\left(2\lambda + \sum_{u \sim v} \gamma_{uv} \right) f_v - \sum_{u \sim v} \gamma_{uv} f_u = 2\lambda f^0_v, \ \ \forall v \in V. \tag{12}$$

Among the exiting 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:

$$\begin{cases} f^{(0)} = f^0 \\ \gamma^{(t)}_{uv} = w_{uv} \left(\| \bigtriangledown_v f^{(t)} \|^{p-2} + \| \bigtriangledown_u f^{(t)} \|^{p-2} \right), \ \ \forall uv \in E \\ f^{(t+1)}_v = \left(2\lambda + \sum_{u \sim v} \gamma^{(t)}_{uv} \right)^{-1} \left(2\lambda f^0_v + \sum_{u \sim v} \gamma^{(t)}_{uv} f^{(t)}_u \right), \ \ \forall v \in V \end{cases} \tag{13}$$

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

$$\varphi^{(t)}_{vu} = \frac{\gamma^{(t)}_{uv}}{2\lambda + \sum_{u \sim v} \gamma^{(t)}_{uv}} \text{ if } u \neq v, \text{ and } \ \varphi^{(t)}_{vv} = \frac{2\lambda}{2\lambda + \sum_{u \sim v} \gamma^{(t)}_{uv}}$$

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

$$f_v^{(t+1)} = \varphi_{vv}^{(t)} f_v^0 + \sum_{u \sim v} \varphi_{vu}^{(t)} f_u^{(t)}, \ \forall v \in V. \tag{14}$$

At each iteration, the new value $f^{(t+1)}$, at a vertex v, depends on two quantities, the original value f_v^0, 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.

4 Filter 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). Equation (11) reduces to $\Delta f^* + 2\lambda(f^* - f^0) = 0$. 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 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). Equation (11) reduces to $\kappa f^* + 2\lambda(f^* - f^0) = 0$. 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.

4.1 Regularization of Functions on Discrete Data

The family of filters presented in Section 3 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. This definition leads to a directed graph because the neighborhood relationship is not symmetric. In order to make this graph symmetric, let $nn_k(u)$ be the set of k-nearest neighbors of the point u. Then, a point v is connected to u if $u \in nn_k(v)$ or $v \in nn_k(u)$.

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 $F_f(v) \in \mathbb{R}^q$. We propose to use one of the two general weight functions:

$$w_{uv} = \exp\left(-\frac{\|F_f(v) - F_f(u)\|^2}{h^2}\right), \ \forall uv \in E, \tag{15}$$

and,

$$w_{uv} = \exp\left(-\frac{\|u - v\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|F_f(v) - F_f(u)\|^2}{h^2}\right), \ \forall uv \in E. \tag{16}$$

where σ and h 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 family of filters. This 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 (16), is equivalent to the bilateral filter, introduced in the context of image denoising [12] [21]. It is a nonlinear filter that combines geometric and range filtering. Bilateral filtering is also used to denoise meshes [22]. 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 mean filter, introduced in the context of images [1]. Indeed, it is obtained by using the weight function (15) 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.

When $\lambda \neq 0$ and w is constant, filter (13) corresponds exactly to the digitized PDE filters proposed in the context of image restoration [11]. 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 regularization and the weighted TV regularization on arbitrary graphs.

4.2 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 [13]. From expression (14), the filter reduces to:

$$f_v^{(t+1)} = \sum_{u \sim v} \varphi_{vu} f_u^{(t)}, \ \forall v \in V, \tag{17}$$

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 [14]. 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 (17) is rewritten as:

$$F^{(t+1)} = PF^{(t)} = P^t F^{(0)}. \tag{18}$$

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

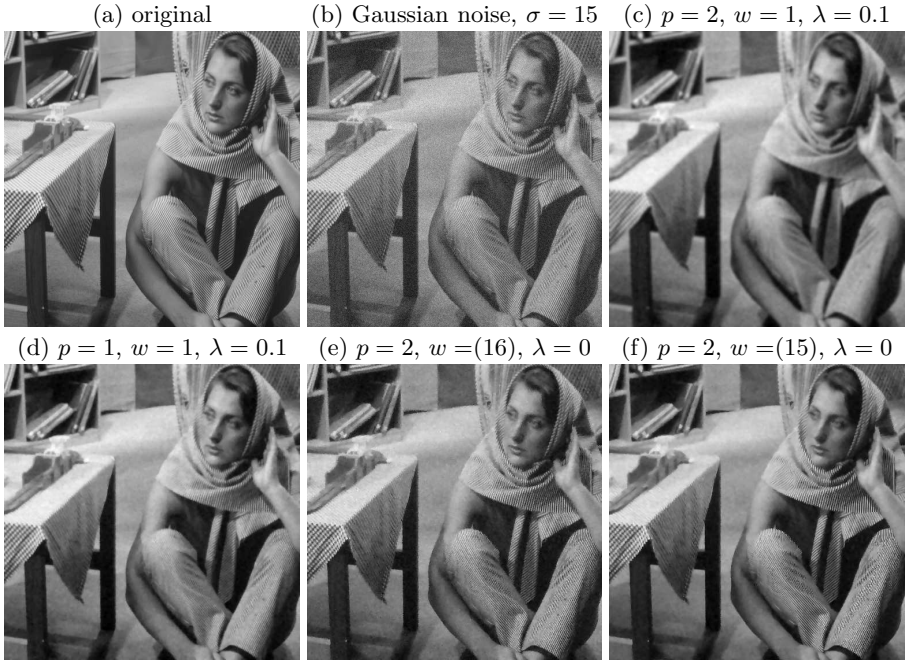

Fig. 1. Application to image denoising. Regularizations performed with $t = 5$. (c) Unweighted L_2 regularization (13) on an 8-adjacency graph. (d) Unweighted TV regularization (13) on an 8-adjacency graph ($p = 1$). (d) Regularization (13) with $p = 2$ and weight function (16) and scalar feature $F_f = f$ on an 8-adjacency graph. (e) Regularization (13) with weight function (15) and feature vector $F_f(v) = [f_u : u \in B_{v,7}]^T$ on an ϵ-neighborhood graph with $\epsilon = 4$.

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 (18) 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 [23]. It 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 process in the spectral domain.

5 Applications

The family of filters proposed in Section 3 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 and surface mesh denoising. We also compare several results obtained for different weight functions and regularization parameters.

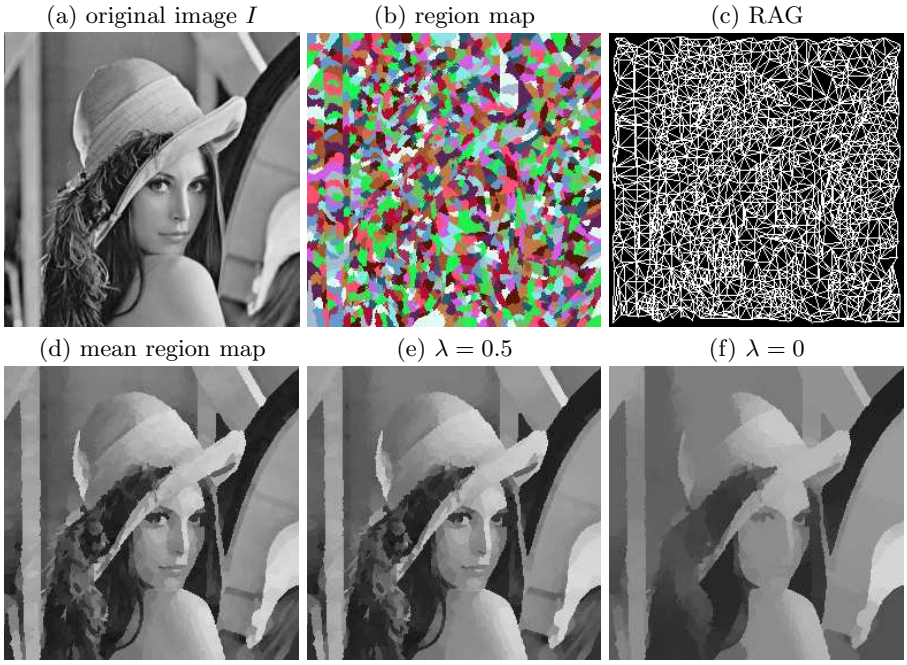

Fig. 2. Image simplification. (a) Original image I. (b) The region map R obtained by an energy partition of I. (c) The RAG G_w associated with R, weighted with weight function (15), $h = 8$ and $F_f(v) = f$. The function considered on the vertices corresponds to the mean value of a region. (d) The mean region map associated with R. (e,f) Regularization of G_w with filter (13).

Image Smoothing and Denoising : Let f^0 be a gray level image of N pixels, $f^0 = [f_1^0, \ldots, f_N^0]^T$ with $f^0 : V \subset \mathbb{Z}^2 \to \mathbb{R}$. Figure 1 illustrates the regularization of a noisy image f^0 (Fig. 1(b)) on a grid graph of 8-adjacency (Fig. 1(c), (d) and (e)), and on a ϵ-neighborhood graph (Fig. 1(f)). The weight function is chosen such that Fig. 1(c) corresponds to the unweighted L_2 regularization, Fig. 1(d) to the unweighted TV regularization, Fig. 1(e) to the bilateral filter, and Fig. 1(f) to the nonlocal mean filter. The use of a non-constant weight function implies an anisotropic diffusion which better preserves image features.

In the case of color images, the function $f^0 : V \subset \mathbb{Z}^2 \to \mathbb{R}^3$ is a mapping from the vertices of the graph to a vector of color channels. The regularization can be applied on each channel leading to an iteration of filter (13) rewritten as:

$$\begin{bmatrix} f_{c1} \\ f_{c2} \\ f_{c3} \end{bmatrix}_v^{(t+1)} = \left(p\lambda + \sum_{u \sim v} \gamma_{uv}^{(t)} \right)^{-1} \left(p\lambda \begin{bmatrix} f_{c1} \\ f_{c2} \\ f_{c3} \end{bmatrix}_v^{0} + \sum_{u \sim v} \gamma_{uv}^{(t)} \begin{bmatrix} f_{c1} \\ f_{c2} \\ f_{c3} \end{bmatrix}_u^{(t)} \right), \quad (19)$$

where $\gamma_{uv}^{(t)}$ depends on the norm of the p-Laplace operator defined by $\| \bigtriangledown_v f \| = \sqrt{\| \bigtriangledown_v f_{c1} \|^2 + \| \bigtriangledown_v f_{c2} \|^2 + \| \bigtriangledown_v f_{c3} \|^2}$. This norm takes into account the inner correlation aspect of color vector data (in the case of $p \neq 2$). The above iteration can be extended to any vector-valued function $\mathbb{R}^m \to \mathbb{R}^k$.

Image Simplification : One can simplify an image by first considering a fine partition of this image (or over-segmentation), where the pixel values of each region of the partition is replaced by the mean or the median pixel value of this region. The partition can be structured by a region adjacency graph (RAG), where each vertex represents a region and where edges are linking adjacent regions. Figure 2(b) and (c) illustrate a fine partition of an image and its associated RAG. Let $G_w = (V, E)$ be a RAG. Let $f^0 : V \subset \mathbb{Z}^2 \to \mathbb{R}^m$ 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 the subgraph of G_w composed of the edges of E for which $w_{uv} > \mu > 0$. Figure 2(e) shows the graph G'_w used to regularize the partition of Fig. 2(d). The filtered partition is depicted in Fig. 2(f).

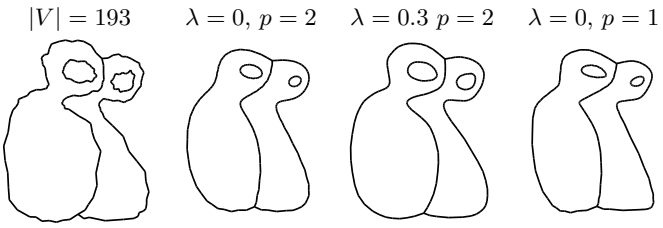

Fig. 3. Polygonal curve denoising. Edges of polygons are weighted with equation (15). The regularization of vertices position is performed in 10 steps. When $\lambda = 0$, the regularization introduces shrinkage effects. They are reduced using a value of $\lambda \neq 0$.

Mesh Smoothing and Denoising : By nature, polygonal curves and surface meshes have a graph structure. 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 $f^0 : V \subset \mathbb{R}^3 \to \mathbb{R}^m$ defined on the graph $G_w = (V, E)$. Results of filter (19) are given in Fig. 3 for polygonal curves, and in Fig. 5 and Fig. 4 for surface meshes.

6 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 mean filter and the bilateral filter, both widely used in image processing. Also, the family is linked to spectral graph filtering.

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. The main ongoing work is to use the proposed framework in the context of hierarchical mesh segmentation and point cloud clustering.

References

1. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. Multiscale Modeling and Simulation **4**(2) (2005) 490–530
2. Chan, T., Shen, J.: Image Processing and Analysis - variational, PDE, wavelets, and stochastic methods. SIAM (2005)
3. Tsai, Y.H.R., Osher, S.: Total variation and level set methods in image science. Acta Numerica **14** (2005) 509–573
4. Alvarez, L., Guichard, F., Lions, P.L., Morel, J.M.: Axioms and fundamental equations of image processing. Archive for Rational Mechanics and Analysis **123**(3) (1993) 199–257
5. Taubin, G.: Geometric signal processing on polygonal meshes. In: Eurographics, State of the Art Report. (2000)
6. Ohtake, Y., Belyaev, A., Bogaeski, I.: Mesh regularization and adaptive smoothing. Computer-Aided Design **33** (2001) 789–800
7. Sochen, N., Deriche, R., Lopez-Perez, L.: Variational Beltrami flows over manifolds. In: ICIP'03: Proc of the Inter. Conf. on Image Processing. Volume I., IEEE Computer Society (2003) 861–864

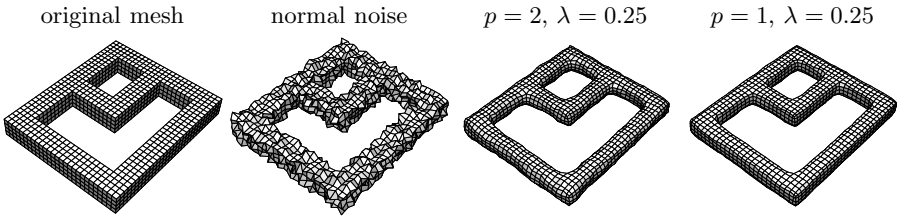

Fig. 4. Mesh denoising by regularizing the position of vertices ($t = 10$ and $w =$(15)).

(a) original mesh (b) normal noise (c) $p = 2$, $\lambda = 0.5$

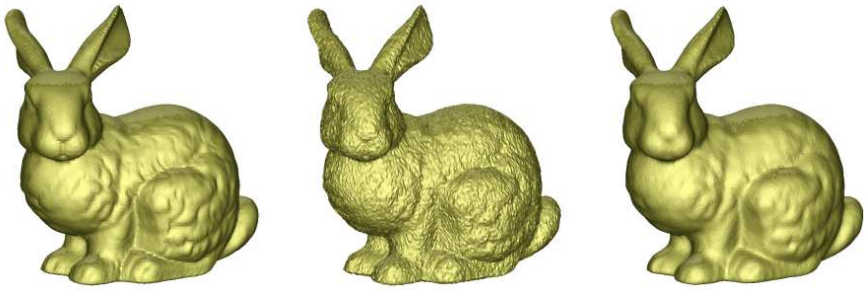

Fig. 5. Mesh denoising. (a) Original Stanford Bunny with $|V| = 35949$. (b) Noisy Bunny with normal noise. (c) Regularization of vertex coordinates on mesh edges in 8 iterations. We use the weight function (15) with the scalar feature $F_f(v) = f_v$.

8. Desbrun, M., Meyer, M., Schröder, P., Barr, A.: Anisotropic feature-preserving denoising of height fields and bivariate data. Graphics Interface (2000) 145–152
9. Bajaj, C.L., Xu, G.: Anisotropic diffusion of surfaces and functions on surfaces. ACM Trans. on Graph. **22**(1) (2003) 4–32
10. Hildebrandt, K., Polthier, K.: Anisotropic filtering of non-linear surface features. Eurographics 2004: Comput. Graph. Forum **23**(3) (2004) 391–400
11. Osher, S., Shen, J.: Digitized PDE method for data restoration. In Anastassiou, E.G.A., ed.: In Analytical-Computational methods in Applied Mathematics. Chapman&Hall/CRC (2000) 751–771
12. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV'98: Proc. of the 6th Int. Conf. on Computer Vision, IEEE Computer Society (1998) 839–846
13. Chung, F.: Spectral graph theory. CBMS Regional Conference Series in Mathematics **92** (1997) 1–212
14. Coifman, R., Lafon, S., Maggioni, M., Keller, Y., Szlam, A., Warner, F., Zucker, S.: Geometries of sensor outputs, inference, and information processing. In: Proc. of the SPIE: Intelligent Integrated Microsystems. Volume 6232. (2006)
15. Chambolle, A.: Total variation minimization and a class of binary MRF models. In: LNCS 3757, Proc. of the 5th Int. Work. EMMCVPR, Springer-Verlag (2005) 136–152
16. Darbon, J., Sigelle, M.: Exact optimization of discrete constrained total variation minimization problems. In R.Klette, Zunic, J., eds.: LNCS 3322: Proc. of the 10th Int. Workshop on Combinatorial Image Analysis. (2004) 548–557
17. Zhou, D., Schölkopf, B.: Regularization on discrete spaces. In: LNCS 3663, Proc. of the 27th DAGM Symp., Springer-Verlag (2005) 361–368
18. Bougleux, S., Elmoataz, A.: Image smoothing and segmentation by graph regularization. In: LNCS 3656, Proc. Int. Symp. on Visual Computing, Springer-Verlag (2005) 745–752
19. Bensoussan, A., Menaldi, J.L.: Difference equations on weighted graphs. Journal of Convex Analysis **12**(1) (2005) 13–44
20. Friedman, J., Tillich, J.P.: Wave equations for graphs and the edge-based laplacian. Pacific Journal of Mathematics **216**(2) (2004) 229–266

21. Barash, D.: A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. IEEE Trans. Pattern Anal. Mach. Intell. **24**(6) (2002) 844–847
22. Fleishman, S., Drori, I., Cohen-Or, D.: Bilateral mesh denoising. ACM Trans. on Graphics **22**(3) (2003) 950–953
23. Coifman, R., Lafon, S., Lee, A., Maggioni, M., Nadler, B., Warner, F., Zucker, S.: Geometric diffusions as a tool for harmonic analysis and structure definition of data. Proc. of the National Academy of Sciences **102**(21) (2005)