Unifying local and nonlocal processing with partial difference operators on weighted graphs

Abderrahim Elmoataz, Olivier Lezoray, Sébastien Bougleux, Vinh Thong Ta

UNIFYING LOCAL AND NONLOCAL PROCESSING WITH PARTIAL DIFFERENCE OPERATORS ON WEIGHTED GRAPHS

Abderrahim Elmoataz[1], *Olivier Lézoray*[1], *Sébastien Bougleux*[2] *and Vinh-Thong Ta*[1]

[1]Université de Caen Basse-Normandie, GREYC, CNRS UMR 6072
6 Boulevard du Maréchal Juin, 14050 Caen Cedex FRANCE
abder.elmoataz@greyc.ensicaen.fr, {olivier.lezoray,vinhthong.ta}@unicaen.fr
[2]Université de Paris-Dauphine, CEREMADE, CNRS UMR 7534,
Place du Maréchal DeLattre De Tassigny, 75775 Paris Cedex 16 FRANCE
bougleux@ceremade.dauphine.fr

ABSTRACT

In this paper, local and nonlocal image processing are unified, within the same framework, by defining discrete derivatives on weighted graphs. These discrete derivatives allow to transcribe continuous partial differential equations and energy functionals to partial difference equations and discrete functionals over weighted graphs. With this methodology, we consider two gradient-based problems: regularization and mathematical morphology. The gradient-based regularization framework allows to connect isotropic and anisotropic p-Laplacians diffusions, as well as neighborhood filtering. Within the same discrete framework, we present morphological operations that allow to recover and to extend well-known PDEs-based and algebraic operations to nonlocal configurations. Finally, experimental results show the ability and the flexibility of the proposed methodology in the context of image and unorganized data set processing.

1. INTRODUCTION

In image processing and computer vision, techniques based on energy minimization and partial differential equations (PDEs) have shown their efficiency in solving many important problems, such as smoothing, denoising, interpolation and segmentation [1, 2, 3, 4].

We focus on two categories of problems based on gradient norms: regularization and mathematical morphology (MM). Solutions of such problems can be obtained by considering the input discrete data (e.g. images, meshes, data sets) as continuous functions defined on a continuous domain, and by designing continuous PDEs whose solutions are discretized in order to fit with the natural discrete domain. Such PDEs-based methods have the advantages of better mathematical modeling, connections with physics and better geometrical approximations. Differential operators involved in these PDEs are classically based on local derivatives, that reflect local interactions on the data. Recently, nonlocal derivatives have been proposed in the context of image processing to design gradient-based regularization functionals and PDEs associated with their minimization [5, 6, 7]. These nonlocal PDEs are linked to an important category of neighborhood filters which have shown their efficiency to better preserve fine and repetitive image structures than local ones [8, 9, 10, 11].

An alternative methodology to continuous PDEs-based regularization, is to formalize the problem directly in discrete settings. This is the case for neighborhood filters which are mainly based on discrete weighted Laplacians. See [12, 13] for a description of these operators in the general context of graph theory. In particular, it is shown that Laplacian filtering is equivalent to Markov matrix filtering, and by consequence it is also related to spectral graph filtering. These properties has been used in the context of image denoising by [14, 15]. Another interesting work is the digitization of the total variation (TV) and the Rudin-Osher-Fatemi (ROF) model of images [16] onto unweighted graphs [17, 18]. This discrete formulation has received much less attention than its continuous analog. An extension of this model, that uses a normalized p-Dirichlet energy on weighted graphs, is proposed in [19] in the context of semi-supervised learning.

We have presented a similar extension in the context of image and mesh processing [20, 21, 22]. There exist several advantages of these latter graph-based approaches. In particular, they lead to a family of discrete and semi-discrete diffusion processes based on p-Laplacians. These processes, parametrized by the graph structure (topology and geometry) and by the degree p of smoothness, allow to perform several filtering tasks such as smoothing/denoising and simplification. Moreover, local and nonlocal image regularizations are formalized within the same framework, that corresponds to the transcription of the nonlocal continuous regularizations proposed in [5, 6]. The unification of local and nonlocal gradient-based regularization is realized by defining explicitly discrete derivatives on graphs. These discrete derivatives can be used to transcribe other continuous PDEs and energy functionals to partial difference equations (PdEs) and discrete functionals over weighted graphs.

The aim of this paper is twofold. Firstly, the gradient-based regularization framework presented in [20, 21, 22] is extended by taking into account a more general regularization functional than the p-Dirichlet energy. In particular, this allows to connect isotropic and anisotropic versions of graph-based p-Laplacians. Secondly, based on the same discrete derivatives, we formulate mathematical morphology operators (dilation and erosion) which can be used to perform several morphological processes on weighted graphs, such as opening, closing, reconstruction and leveling. These operators and processes are analog to the ones encountered in continuous PDEs-based MM [23, 24] and in algebraic MM [25, 26, 27]. In this latter approach, only algebraic MM operations are considered on particular graphs (binary, minimum spanning tree). Also, both continuous PDEs-based MM and algebraic MM are considered in local settings. Our general graph-based approach has the advantage to handle local and nonlocal configurations within the definition of MM operators.

The rest of this paper is organized as follows: Section 2 recalls basic definitions related to graphs and introduces first and second order operators used in the rest of the paper. Section 3 presents the proposed regularization framework and associated filters. This framework is illustrated in Section 4 on two interpolation problems, namely semi-supervised image colorization and segmentation. Then Section 5 presents the proposed graph-based MM operators. Finally, Section 6 shows the application of the proposed methods to process image partitions and unorganized data sets.

2. DIFFERENCE OPERATORS ON GRAPHS

In this section, we recall some basic definitions on graphs, and we define first and second order operators which can be considered as discrete versions of continuous differential operators. Analog definitions and properties have also been used in the context of functional analysis on graphs [28, 29] and semi-supervised learning [19, 13].

2.1. Preliminary definitions

Let $G = (V, E, w)$ be a *weighted graph* with a finite set V of vertices and a finite set $E \subset V \times V$ of weighted edges. The weight w_{uv} of an edge $(u, v) \in E$ is generally defined from a function $w : V \times V \to \mathbb{R}^+$ such that

$$w(u,v) = \begin{cases} w_{uv} & \text{if } (u,v) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

It encodes the similarity between two vertices of the graph. In this paper, graphs are assumed to be connected, undirected, with no self-loops or multiple edges. Under these conditions, the weight function w is symmetric ($w(u, v) = w(v, u), \forall(u, v) \in V \times V$), and $w(u, u) = 0$ for all $u \in V$.

Let $A \subset V$ be a connected subset of V, i.e. for all $u \in A$ there exists a vertex $v \in A$ such that $(u, v) \in E$. Let $A^c = V \setminus A$ be the complement of A in the graph G. Then, the boundary of A in G is composed of the *outer boundary* and the *inner boundary* of A, that are defined respectively by the two following subsets of V:

$$\begin{cases} \partial^+ A \overset{\text{def.}}{=} \{u \in A^c \,:\, \exists v \in A, (u,v) \in E\}, \\ \partial^- A \overset{\text{def.}}{=} \{u \in A \,:\, \exists v \in A^c, (u,v) \in E\}. \end{cases} \quad (1)$$

One can note that the boundary of V cannot be directly defined by (1). In this special case, it must be given.

The set V of vertices can be regarded as a discrete space. Let $f, h : V \to \mathbb{R}$ be two discrete functions (vectors) that assign a real value to each vertex of the graph. The L^2 inner product of these functions is given by:

$$\langle f, h\rangle_V \overset{\text{def.}}{=} \sum_{u \in V} f(u)h(u). \quad (2)$$

The space of such function is noted $\mathcal{H}(V)$.

Similarly, we define the space $\mathcal{H}(E)$ of functions defined on the set E of edges. Let $F, H : E \to \mathbb{R}$ be two functions that assign a real value to each edge $(u, v) \in E$. The inner product of these functions is defined by:

$$\langle F, H\rangle_E \overset{\text{def.}}{=} \tfrac{1}{2} \sum_{u \in V} \sum_{v \sim u} F(u,v)H(u,v), \quad (3)$$

where $v \sim u$ denotes a vertex v connected to u by an edge of E.

2.2. Construction of weighted graphs

Functions of the space $\mathcal{H}(V)$, defined in the previous section, represent the data to be processed. These functions can be originally defined on geometric spaces, such as images and unorganized set of points. Indeed, any discrete image $I : \Omega \subset \mathbb{Z}^2 \to \mathbb{R}$ can be regarded as a function $f^0 : V \subset \mathbb{Z}^2 \to \mathbb{R}$, where V is the set of pixels. More generally, this is also the case of any unorganized set of point $V \subset \mathbb{R}^n$, which can be seen as a function $f^0 : V \subset \mathbb{R}^n \to \mathbb{R}^n$.

There exist several popular methods that transform the set V, with a given pairwise distance measure $\mu : V \times V \to \mathbb{R}^+$, into a neighborhood graph (or similarity graph). Constructing such a graph consists in modeling neighborhood relationships between data.

Among the existing graphs, the simplest of them is the δ-neighborhood graph, noted G_δ, where two data $u, v \in V$ are connected by an edge of E if $\mu(u, v) \leq \delta$, with $\delta > 0$ a threshold parameter. We can also quote the minimum spanning tree, the k-nearest neighbors graph, the Delaunay triangulation, or the relative neighborhood graph, as other possible graph topologies (see [30] for a survey on neighborhood graphs used in pattern recognition).

In this paper, we consider the δ-neighborhood graph, and a modified version of the k-nearest neighbors graph since this latter graph is not necessarily directed. In order to make this graph undirected, let *nnk*(u) be the set of k-nearest neighbors of the vertex u. Then, a vertex v is connected to u if $u \in$ *nnk*(v) or $v \in$ *nnk*(u). The obtained graph is noted k-*NNG*. We also consider the complete graph that we note G_∞.

When the function f^0 is a discrete image $f^0 : V \subset \mathbb{Z}^2 \to \mathbb{R}$, the choice of the function μ to construct the graph can be defined as the Chebyshev distance:

$$\mu(u=(x_i,y_i), v=(x_j,y_j)) = \max\{|x_i - x_j|, |y_i - y_j|\}.$$

By using this distance, the shape of the neighborhood involved in the δ-neighborhood graph corresponds to the standard square window of size $(2\delta+1)\times(2\delta+1)$. In particular, G_1 is the 8-adjacency grid graph. The 4-adjacency grid graph is noted G_0.

When the function $f^0 : V \subset \mathbb{R}^n \to \mathbb{R}^n$ represents a discrete set of data, the function μ is simply chosen as the Euclidean distance.

In order to process a given function $f^0 \in \mathcal{H}(V)$ the construction of graphs can also take into account this function within the distance measure μ. Once the graph has been constructed, its weights are computed according to a measure of similarity $g : V \times V \to \mathbb{R}^+$, which satisfies:

$$w(u,v) = \begin{cases} g(u,v) & \text{if } (u,v) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

This measure can simply be defined as the inverse of the distance measure: $g = \mu^{-1}$. Distances between vertices are estimated by comparing their features. To this aim, every data $u \in V$ is assigned with a feature vector denoted by $F(f^0,u) \in \mathbb{R}^q$. Several choices can be considered for the expression of F, depending on the nature of the features to be preserved. In the simplest case, one can consider $F(f^0,u) = f^0(u)$. The weight function w, associated to a given graph, can naturally incorporate local or nonlocal features according to the topology of the graph. For instance, one can consider the following weight functions:

$$g_1(u,v) = \exp\left(-\frac{\rho(F(f^0,v),F(f^0,u))^2}{\sigma^2}\right),$$
$$g_2(u,v) = (\rho(F(f^0,u),F(f^0,v)) + \epsilon)^{-1}, \ \epsilon > 0, \ \epsilon \to 0,$$

where $\sigma > 0$ controls the feature similarity and $\rho : V \times V \to \mathbb{R}^+$ is a distance measure to be defined next.

When $f^0 \in \mathcal{H}(V)$ is an image, an important feature vector is provided by image patches, i.e. $F(f^0,u)$ is the values of f^0 in a square window of size $(2k+1)\times(2k+1)$, centered at the vertex u, which we note $F_k(f^0,u)$. This feature vector has been proposed in the context of texture synthesis [31], and then used in the context of image processing (see [11, 7, 32, 6, 5] and references therein). The distance function ρ associated with this feature vector is given by:

$$\rho(F_k(f^0,u),F_k(f^0,v)) = \sum_{x=-k}^{k}\sum_{y=-k}^{k} K((x,y)) \|f^0(u+(x,y)) - f^0(v+(x,y))\|_2^2,$$

where K is a Gaussian kernel of a given standard deviation. This latter can be replaced by the Chebyshev distance between the position of pixels.

2.3. Difference operator and its adjoint

All the basic operators considered in this paper are defined from the difference operator or the discrete derivative. There exist several definitions of these operators on graphs [28, 29, 19, 13]. As in [20], we present here a definition of the difference operator that allows to retrieve the expression of combinatorial p-Laplace operators.

The *difference operator* $d : \mathcal{H}(V) \to \mathcal{H}(E)$ of a function $f \in \mathcal{H}(V)$ is the vector of all weighted discrete derivatives:

$$df \overset{\text{def.}}{=} ((df)(u,v))_{(u,v)\in E},$$

where

$$(df)(u,v) \overset{\text{def.}}{=} \sqrt{w_{uv}}\,(f(v) - f(u)), \ \ \forall (u,v) \in E, \quad (4)$$

and

$$\partial_v f(u) \overset{\text{def.}}{=} (df)(u,v) \quad (5)$$

is the *discrete (partial) derivative* of f, with respect to the edge (u,v), at a vertex u. One can observe that this derivative share the same properties as the continuous derivative of a function defined in the Euclidean space, i.e. $\partial_u f(v) = -\partial_v f(u)$, and $\partial_u f(u) = 0$, and if $f(u) = f(v)$ then $\partial_v f(u) = 0$.

We define also the vector:

$$|df| \overset{\text{def.}}{=} (|(df)(u,v)|)_{(u,v)\in E}, \quad (6)$$

where $|(df)(u,v)| = \sqrt{w_{uv}}|f(v) - f(u)|$. This one is used in the definition of the anisotropic p-Laplacian (see Section 2.5).

The *adjoint operator* $d^* : \mathcal{H}(E) \to \mathcal{H}(V)$, of the difference operator d is defined by:

$$\langle df, H\rangle_E = \langle f, d^*H\rangle_V,$$

for all $f \in \mathcal{H}(V)$ and $H \in \mathcal{H}(E)$. By using the expressions (2) and (3), it is easy to deduce the following expression of d^* at each vertex of the graph (see [13, 22]):

$$(d^*F)(u) = \tfrac{1}{2}\sum_{v\sim u} \sqrt{w_{uv}}\,(F(v,u) - F(u,v)). \quad (7)$$

We introduce also two other difference operators that constitute the basis of the morphological operators defined in Section 5. They are based on the difference operator d, and on the classical maximum (respectively minimum) operator as:

$$\begin{aligned} (d^+f)(u,v) &\overset{\text{def.}}{=} \max(0, (df)(u,v)), \text{ and} \\ (d^-f)(u,v) &\overset{\text{def.}}{=} \min(0, (df)(u,v)). \end{aligned} \quad (8)$$

As before, the corresponding partial derivatives are respectively given by $\partial_v^+ f(u)$ and $\partial_v^- f(u)$.

2.4. Gradients and their norms

As in the Euclidean space, one can define the gradient of a function $f \in \mathcal{H}(V)$ at each vertex $u \in V$ as the vector of all partial derivatives, with respect to the set of edges $(u, v) \in E$:

$$(\nabla f)(u) \overset{\text{def.}}{=} (\partial_v f(u))_{(u,v)\in E}. \tag{9}$$

In the sequel, we use the L^p-norm of this vector:

$$\|(\nabla f)(u)\|_p = \left(\sum_{v\sim u} w_{uv}^{\frac{p}{2}} |f(v) - f(u)|^p\right)^{\frac{1}{p}}, \tag{10}$$

as well as its infinite norm:

$$\|(\nabla f)(u)\|_\infty = \max_{v\sim u} \left(\sqrt{w_{uv}}|f(v) - f(u)|\right).$$

As before, one can define two other gradients (and their associated norms) based on the partial derivatives $\partial^+ f$ and $\partial^- f$, which we note respectively ∇^+ and ∇^-. For instance, we have the following norms for ∇^+:

$$\|(\nabla^+ f)(u)\|_p = \left(\sum_{v\sim u} w_{uv}^{\frac{p}{2}} (\max(0, f(v) - f(u)))^p\right)^{\frac{1}{p}},$$
$$\|(\nabla^+ f)(u)\|_\infty = \max_{v\sim u} \left(\sqrt{w_{uv}} \max(0, f(v) - f(u))\right).$$

2.5. Second order operators

One of the most important second order operator on graphs is the Laplacian, which has several well-established expressions (see [12, 13] for a complete study). All the expressions can be derived from the following definition.

The *Laplacian* $\Delta : \mathcal{H}(V) \to \mathcal{H}(V)$ is the linear operator defined by:

$$\Delta f \overset{\text{def.}}{=} d^* df.$$

By using expressions (7) and (4), we retrieve the *combinatorial Laplacian* (or *unnormalized Laplacian*), which is expressed at each vertex $u \in V$ by:

$$\begin{aligned}(\Delta f)(u) &= \sum_{v\sim u} w_{uv}(f(u) - f(v)),\\ &= f(u)\sum_{v\sim u} w_{uv} - \sum_{v\sim u} w_{uv} f(v).\end{aligned} \tag{11}$$

An extension of the Laplacian is the *(isotropic) p-Laplacian* $\Delta_p^i : \mathcal{H}(V) \to \mathcal{H}(V)$ defined for $p \in (0, +\infty)$ by:

$$\Delta_p^i f \overset{\text{def.}}{=} d^*(\|\nabla f\|_2^{p-2} df).$$

Again by using (7) and (4), we get the *combinatorial* version of the *p-Laplacian*, which is expressed at each vertex $u \in V$ by (see [22] for more details):

$$(\Delta_p^i f)(u) = \sum_{v\sim u} \gamma_{uv}^{p,f}(f(u) - f(v)),$$
$$\text{with } \gamma_{uv}^{p,f} = \tfrac{1}{2} w_{uv}\left(\|(\nabla f)(u)\|_2^{p-2} + \|(\nabla f)(v)\|_2^{p-2}\right). \tag{12}$$

It is a nonlinear operator for $p{\neq}2$. An interesting case is provided by $p{=}1$, which corresponds to the *combinatorial (mean) curvature* of the function f over the graph (see for instance [18] for a similar definition on unweighted graphs in the context of image restoration).

We now define another p-Laplacian, which is based on the vector $|df|$. The *(anisotropic) p-Laplacian* $\Delta_p^a : \mathcal{H}(V) \to \mathcal{H}(V)$ is defined by:

$$\Delta_p^a f = d^*(|df|^{p-2} df).$$

Then, from (7) and (4), we obtain the *combinatorial* version of the *anisotropic p-Laplacian*, which is expressed at each vertex $u \in V$ by (see [33] for more details):

$$(\Delta_p^a f)(u) = \sum_{v\sim u} w_{uv}^{\frac{p}{2}} |f(u) - f(v)|^{p-2}(f(u) - f(v)). \tag{13}$$

As Δ_p^i, this operator is nonlinear if $p{\neq}2$. If $p{=}2$, both p-Laplacians corresponds to the Laplacian. It can be shown that they can be linked within a same operator defined by $d^*(|df|^{p-2}\|\nabla f\|_2^{p-2} df)$. In the same spirit, one can also define higher order operators.

3. REGULARIZATION MODELS AND DIFFUSION PROCESSES

In this section, we propose a variational model to regularize functions defined on the vertices of graphs, and the discrete diffusion processes associated with it.

3.1. Problem formulation and equations on graphs

Let $f^0 \in \mathcal{H}(V)$ be a given function defined on the vertices of a weighted graph $G = (V, E, w)$. In a given context, this function represents an observation of a clean function $h \in \mathcal{H}(V)$ corrupted by an additive noise $n \in \mathcal{H}(V)$ such that $f^0 = h + n$.

To recover the uncorrupted function h, the processing task is to remove the noise n from f^0. A commonly used method is to seek for a function $f \in \mathcal{H}(V)$, which is regular enough on G, and also close enough to f^0. This can be formalized by the minimization of an energy functional which involves a regularization term (or penalty term) plus an approximation one (or fitting term). In this paper, we consider the following model:

$$h \approx \underset{f:V\to\mathbb{R}}{\operatorname{argmin}}\ J(f) + \tfrac{\lambda}{2}\|f - f^0\|_2^2, \tag{14}$$

$$\text{where}\quad J(f) \overset{\text{def.}}{=} \sum_{u\in V} \phi\left(\|(\nabla f)(u)\|_p\right) \tag{15}$$

is a gradient-based functional, and $\lambda \in \mathbb{R}$ is a regularization parameter, called Lagrange multiplier, that controls the trade-off between the penalty term and the fitting term. The function $\phi(\cdot)$ is a kernel that penalizes large variations of f in the neighborhood of each vertex. Several penalty kernels have been proposed in literature, in different situations. Among them, we can quote $\phi(s) = s^2$ (known in the context of Tikhonov regularization [34]), $\phi(s) = s$

(total variation [16, 17]), $\phi(s) = \sqrt{s^2+\epsilon^2} - \epsilon$ (regularized total variation [16, 17]), and $\phi(s) = r^2 \log(1+s^2/r^2)$ (nonlinear diffusion [35]).

To get the solution of (14), we consider the following system of equations (Euler-Lagrange equation):

$$\frac{\partial J(f)}{\partial f(u)} + \lambda(f(u) - f^0(u)) = 0, \ \forall u \in V, \tag{16}$$

where the first term denotes the variation of (15) with respect to f at a vertex u. It is easy to show that this variation is equal to:

$$\begin{aligned}\frac{\partial J(f)}{\partial f(u)} &\overset{(15)}{=} \frac{\partial \phi\left(\|(\nabla f)(u)\|_p\right)}{\partial f(u)} + \sum_{v\sim u} \frac{\partial \phi\left(\|(\nabla f)(v)\|_p\right)}{\partial f(u)} \\ &= \phi'\left(\|(\nabla f)(u)\|_p\right) \frac{\partial \|(\nabla f)(u)\|_p}{\partial f(u)} \\ &\quad + \sum_{v\sim u} \phi'\left(\|(\nabla f)(v)\|_p\right) \frac{\partial \|(\nabla f)(v)\|_p}{\partial f(u)} \\ &\overset{(10)}{=} \sum_{v\sim u} \alpha_{uv}^{\phi,p,f} |f(v) - f(u)|^{p-2}(f(u) - f(v)),\end{aligned} \tag{17}$$

where

$$\alpha_{uv}^{\phi,p,f} = w_{uv}^{\frac{p}{2}} \left(\frac{\phi'\left(\|(\nabla f)(u)\|_p\right)}{\|(\nabla f)(u)\|_p^{p-1}} + \frac{\phi'\left(\|(\nabla f)(v)\|_p\right)}{\|(\nabla f)(v)\|_p^{p-1}} \right).$$

One can observe that this expression has the form of an operator $\Delta_\phi : \mathcal{H}(V) \to \mathcal{H}(V)$, $\Delta_\phi \overset{\text{def.}}{=}$ (17) closely related to the second order operators introduced in Section 2.5. Indeed, the q-Laplacian operator Δ_q^i is equal to (17) if $p = 2$ and $\phi(s) = s^q$, and the q-Laplacian Δ_q^a is equal to (17) if $p = q$ and $\phi(s) = s^q$.

In most cases (values of q), the system (16) is nonlinear, and thus it is difficult to find a close solution. Approximated solutions are given in the following sections. Also, the regularization functional J must be convex to ensure that the solution of (16) is also the solution of (14), which depends on ϕ and p.

3.2. Diffusion processes

The first method, that is considered to get the solution of (16), is based on the gradient descent of (16):

$$\partial_t f(u,t) = -(\Delta_\phi f)(u,t) + \lambda(f^0(u) - f(u,t)), \ \forall u \in V, \tag{18}$$

with the initial condition $\partial_{t=0} f = f^0$. This describes a family of fitted diffusion flows on weighted graphs. This family includes and extends several well-known flows intensively used in image processing and computer graphics. Most of them are formulated without the fitting term (λ=0), and has been analyzed by [2] in the context of image processing. In particular, for the regularization kernel $\phi(s)=s^2$ and p=2, we obtain Laplacian-based diffusion, and if $\phi(s) = s$ it corresponds to mean curvature-based diffusion.

A classical iterative algorithm to get the solution of (18), at a time $t+1$, is the Euler one. An iteration of this algorithm is given by:

$$f(u,t+1) = f(u,t) + \Delta t\, \partial_t f(u,t), \ \forall u \in V, \tag{19}$$

where $f(\cdot,t)$ is the parametrization of f by an artificial time $t > 0$.

3.3. Neighborhood filters

This section describes a second approach to get the solution of (16), that is rewritten as:

$$\left(\lambda + \sum_{v\sim u} \alpha_{uv}^{\phi,p,f} |f(v) - f(u)|^{p-2}\right) f(u) \\ + \sum_{v\sim u} \alpha_{uv}^{\phi,p,f} |f(v) - f(u)|^{p-2} f(v) = \lambda f^0(u)$$

Since this is a nonlinear system, an interesting approximate solution is provided by the linearized Gauss-Jacobi iterative algorithm, an iteration of which is decomposed in the following two steps:

$$\begin{cases} \beta_{uv}^{\phi,p,f} = \sum_{v\sim u} \alpha_{uv}^{\phi,p,f} |f(v,t) - f(u,t)|^{p-2}, \ \forall (u,v) \in E, \\ f(u,t+1) = \dfrac{\lambda f^0(u) + \sum_{v\sim u} \beta_{uv}^{\phi,p,f} f(v,t)}{\lambda + \sum_{v\in V} \beta_{uv}^{\phi,p,f}}, \ \forall u \in V. \end{cases} \tag{20}$$

This describes a family of neighborhood filters. Indeed, at each iteration, the new value of f at a vertex u depends on two quantities: the initial value $f^0(u)$, and a weighted average of the filtered values of f in the neighborhood of u. As in Section 3.2, the choice of the regularization parameters and the choice of the graph allow to retrieve and to extend several well-known filters proposed in the context of image smoothing and denoising.

In particular, for p=2 and $\phi(s)=\frac{1}{2}s^2$, iteration (20) is rewritten as:

$$f(u,t+1) = \frac{\lambda f^0(u) + \sum_{v\sim u} w_{uv} f(v,t)}{\lambda + \sum_{v\sim u} w_{uv}}. \tag{21}$$

Without being exhaustive, when $\lambda = 0$ (no fitting term), one iteration of (21) corresponds to the following filters:

- Gaussian filter if the weight function is

$$w_{uv} = \exp\left(-\frac{\|u-v\|_2^2}{2\sigma^2}\right),$$

- σ-filter [8, 9] if

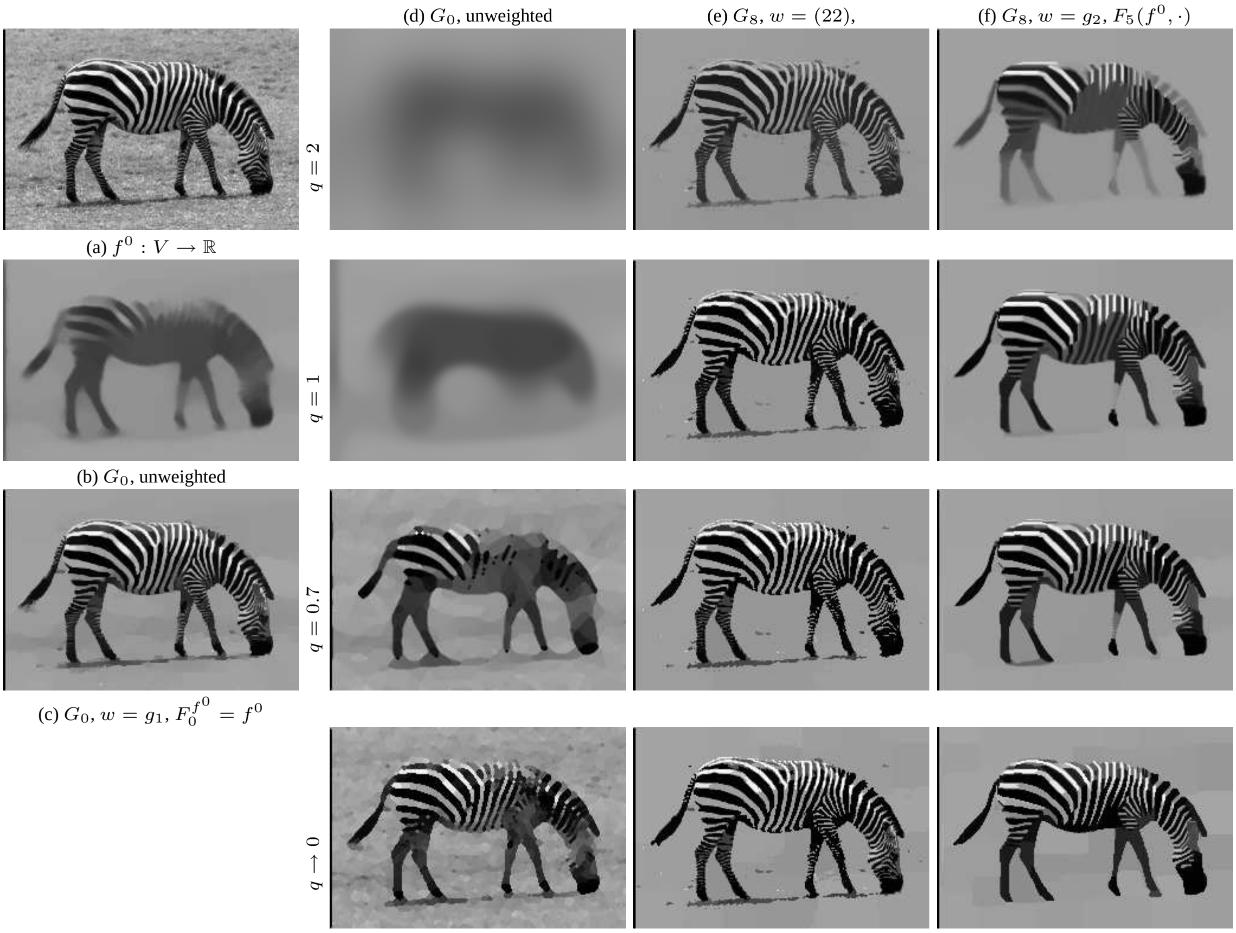
$$w_{uv} = \exp\left(-\frac{\|f^0(u) - f^0(v)\|_2^2}{\sigma^2}\right),$$

- SUSAN [36] or bilateral [10] filters if

$$w_{uv} = \exp\left(-\frac{\|u-v\|_2^2}{2\sigma^2}\right) \exp\left(-\frac{\|f^0(u) - f^0(v)\|_2^2}{\sigma^2}\right), \tag{22}$$

- Nonlocal means filter [11] if w=g_2.

While iterated versions of these three latter filters are related to nonlinear diffusion (since the weights depend on the filtered function f, they need to be updated at each iteration), several iterations of (21) describe a linear filter that is related to Laplacian smoothing (Section 3.2).



(d) G_0, unweighted

(e) G_8, $w = (22)$,

(f) G_8, $w = g_2$, $F_5(f^0, \cdot)$

(a) $f^0 : V \to \mathbb{R}$

(b) G_0, unweighted

(c) G_0, $w = g_1$, $F_0^{f^0} = f^0$

Figure 1. Local and nonlocal image smoothing. (a) The initial image f^0 is regularized until convergence of the filter (20), with p=1, $\phi(s)$=s, p=2, and $\lambda = 0.01$: (b) Discrete TV regularization. (c) Discrete weighted-TV regularization. Columns (d), (e) and (f): Behavior of the regularization with 800 iterations of (20), $\lambda = 0$, $\phi(s) = s^q$, $p = 2$. On G_0, it corresponds to the unweighted Laplacian smoothing for $q = 2$ and to the digital TV filter for $q = 1$. On G_8 with w=(22), it is the iterative bilateral filter (without updating the weights) for $q = 2$. On G_8 with $w = g_2$, it is the iterative nonlocal means filter (without updating the weights). The other cases can be considered as extensions of these filters by varying q.

Another particular case of the proposed neighborhood filters is the TV digital filter [18], which is obtained for $\phi(s)$=s, p=2, and w_{uv}=1 for all $(u, v) \in E$. We have extended this filter to weighted graphs and to $\phi(s) = s^q$ in [20, 21, 22]. Figures 1(a) to 1(c) illustrate the difference between the weighted and unweighted cases in the context of image smoothing on the graph G_0. We can observe that for the same value of λ, using a weight function helps to preserve image discontinuities.

The behavior of the regularization for $p = 2$, $\phi(s) = s^q$, $q \in (0, 2]$, is illustrated in Figures 1(d) to 1(f) for several values of q, several graph structures and $\lambda = 0$. The number of iterations is the same in all the cases (800). We can do two principal observations. As the size of the neighborhood increases, sharp edges and image redundancies are better preserved. This is also the case for the use of nonlocal weights based on patches. When $q < 1$ and particularly when $q \to 0$, the regularization behaves like a simplification procedure. This last observation is depicted in the first row of Figure 2, where we can see the effect of the structure of the graph. More examples are given in [21, 22] in the context of image and mesh smoothing/denoising.

In the sequel, the family of filters presented in this section are the one that is used in the applications. More generally, to process vector-valued functions $f : V \to \mathbb{R}^n$, $f = (f_i)_{i=1,\dots,n}$, we use the same filter but the norm of the gradient, in the coefficient $\alpha_{uv}^{\phi,p,f}$, is replaced by its n-dimensional version:

$$\|(\nabla f)(u)\|_p \overset{\text{def.}}{=} \left(\sum_{i=1,\dots,n} \|(\nabla f_i)(u)\|_p^p \right)^{\frac{1}{p}} .$$

4. REGULARIZATION-BASED INTERPOLATION

In image processing, several problems such image inpainting, super-resolution, image colorization or semi-supervised segmentation can be interpreted as interpolation problems. Given a data set where some data are missing, interpolation consists in predicting missing data from existing ones.

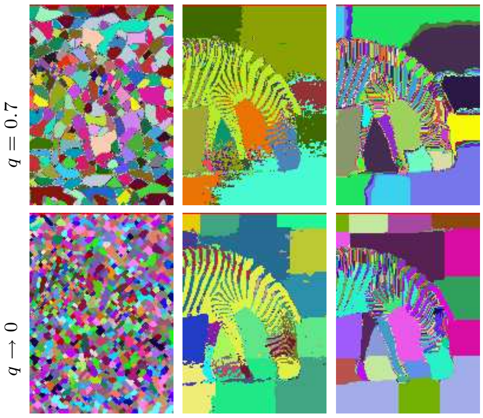


Figure 2. Results presented in Figure 1 for $q < 1$ and rendered here in false colors (each color corresponds to a gray value). First line represents part of 5th row of Figure 1 and second line represents Part of 6^{th} row of Figure 1. We can observe the relation between the size of the neighborhood and the leveling of the image.

Formally, given a knowing function $f^0 \in \mathcal{H}(V^0)$ defined on $V^0 \subset V$, interpolation problem consists in predicting a function $f \in \mathcal{H}(V)$ according to f^0. These problems can be formulated by considering the discrete regularization model (14):

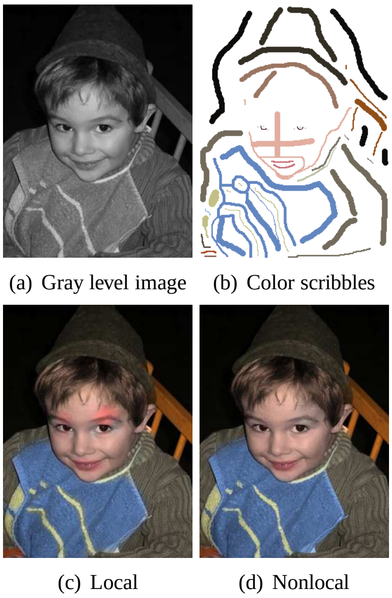
$$\underset{f:V\to\mathbb{R}}{\text{argmin}}\ J(f) + \sum_{u\in V} \frac{\lambda(u)}{2}(f(u) - f^0(u))^2, \qquad (23)$$

where $\lambda : V \to \mathbb{R}$ is a function of the form:

$$\lambda(u) \overset{\text{def.}}{=} \begin{cases} \lambda_u & \text{if } u \in V^0 \\ 0 & \text{otherwise.} \end{cases} \qquad (24)$$

In this section, we focus on two categories of interpolation problems: the semi-supervised image colorization and segmentation.

Image colorization. Colorization is the process of adding colors to monochrome images and is usually made by hand by an expert. Recently, several methods have been proposed for colorization [37, 38] that less require intensive manual efforts. These techniques colorize the image based on the user's input color scribbles and are mainly based on a diffusion process. However, most of these diffusion processes only use local pixel interactions that cannot properly describe complex structures expressed by nonlocal interactions. We propose to address this problem within our framework and we propose to introduce nonlocal configurations in colorization processes [39].


(a) Gray level image (b) Color scribbles

(c) Local (d) Nonlocal

Figure 3. Local versus nonlocal colorization with $\phi(s) = s$, $p = 2$ and $\lambda_u = 0.01$

Figure 3 shows a comparison between local and nonlocal colorization. Figure 3(a) shows a grayscale image $f^l : V \to \mathbb{R}$, on which a user provides an image of color scribbles $f^s : V_0 \subset V \to \mathbb{R}^3$ (Figure 3(b)). The image color scribbles defines a mapping from the vertices to a vector of RGB color channels: $f^s(u) = (f_i^s(u))_{i=R,G,B}$. From these functions, one computes $f^c : V \to \mathbb{R}^3$ that defines a mapping from the vertices to a vector of chrominance:

$$f^0(u) = \begin{cases} \left(f_i^s(u)/f^l(u)\right)_{i=R,G,B} & \text{if } u \in V_0 \\ (0,0,0) & \text{otherwise.} \end{cases}$$

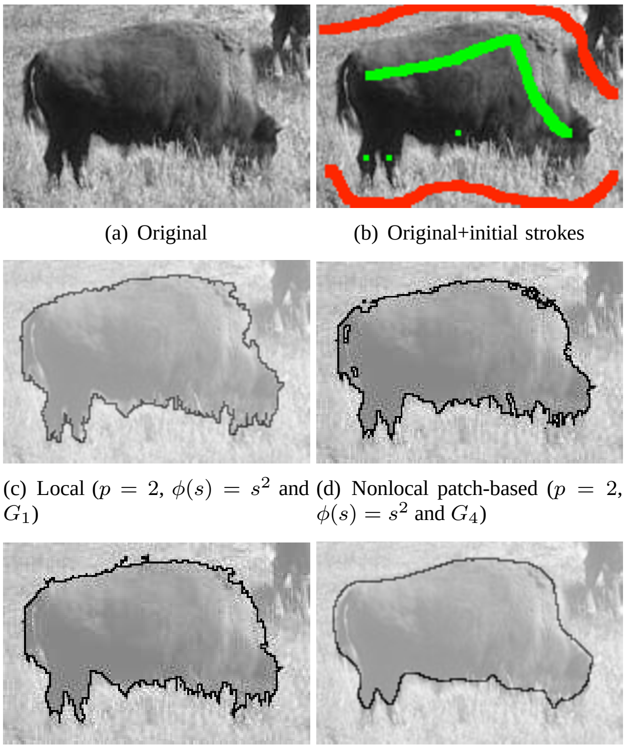
The colorization process is performed according to Equation (23), where $\lambda_u = 0.01$ ensures that the original color scribbles can change during the process. At convergence of the process, the final function is defined as $f : V \to \mathbb{R}^3$ and final colors are obtained by

$$f^l(u)\,(f_i(u, t \to +\infty))_{i=R,G,B} \quad .$$

Figures 3(c) and 3(d) show the obtained colorization, respectively in local and nonlocal schemes. The graph associated with the local processing is the graph G_1 associated with the weight function $w = g_1$ where $F(f^l, u) = f^l(u)$ for a vertex u. For the nonlocal colorization, the associated graph is the graph G_5 associated with the weight function $w = g_1$ where $F_2(f^l, u)$ is used as a feature vector.

One can view the benefits of nonlocal processing as compared to local one: the eyes and several areas of the bib are not properly colored and have diffused over straight edges. On the opposite, nonlocal colorization has successfully colored these areas thanks to its ability to discover similar textures and fine details.

Semi-supervised image segmentation. Numerous automatic segmentation schemes have been proposed in literature and they have shown their efficiency. But sometimes, automatic segmentation results are not accurate when images are much more complex. Meanwhile, recent interactive image segmentation approaches have been proposed. They reformulate image segmentation into semi-supervised classification by label propagation strategies [40, 41]. Other applications of these label diffusion methods can be found in [19, 42]. We propose to address this learning



(a) Original

(b) Original+initial strokes

(c) Local ($p = 2$, $\phi(s) = s^2$ and G_1)

(d) Nonlocal patch-based ($p = 2$, $\phi(s) = s^2$ and G_4)

(e) Local ($p = 1$, $\phi(s) = s$ and G_1)

(f) Nonlocal patch-based ($p = 2$, $\phi(s) = s^2$ and 16-NNG $\cup$ G_1 graph)

Figure 4. Local versus nonlocal patch-based image semi-supervised segmentation. All the result images were whitened in order to accentuate the user labels and the segmented boundary.

problem as an interpolation problem within our regularization framework.

The semi-supervised clustering of the set V consists in grouping the set V into k classes where the number k of classes is given. For this, the set V is composed of labeled data sets and unlabeled ones. The objective is then to estimate the unlabeled data from labeled ones. Let c_i be the set of vertices which belong to the ith class. The set $V^0 {=} \{c_i\}_{i=1,\dots,k}$ is the initial set of labeled data, and the initial unlabeled data is the set $V \setminus V^0$. This is equivalent to consider k label functions $f_i^0{:}V{\rightarrow}\mathbb{R}$ such as

$$f_i^0(u) = \begin{cases} 1 & \text{if } u \in c_i \text{ with } i = 1, \dots, k, \forall\, c \in V^0 \\ 0 & \text{otherwise} \end{cases},$$

where each f_i^0, with $i = 1, \dots, k$, corresponds to a given class. Starting from the labeled data (the f_i^0's), the vertex clustering is accomplished by k regularizations defined in (23) where $\lambda_u = +\infty$. At convergence of the processes, one can estimate the class membership probabilities and assign to a vertex u the most plausible one. For all $i \in 1, \dots, k$, we have

$$\operatorname*{argmax}_i \Big\{ f_i(u, t \rightarrow +\infty) / \sum_i f_i(u, t \rightarrow +\infty) \Big\} \ . \quad (25)$$

To obtain a final image segmentation, a connected image components labeling can be performed on classified elements.

Figure 4 shows the behavior of our semi-supervised image segmentation method for local and nonlocal configurations, graph structures and p values.

Figure 4(b) shows original image with initial labels superimposed. Figures 4(c) and 4(e) show the final segmentation performed on the graph G_1 with the weight function $w = g_1$. Figure 4(c) is obtained with $p = 2$ and Figure 4(e) with $p = 1$. In the latter case, we use the anisotropic version of our regularization. Both results show a suitable segmentation. When we use nonlocal configuration (Figure 4(d)), the segmentation captures more fine image structures and details. The associated graph for this nonlocal processing is a graph G_4 with a weight function $w = g_1$ where the feature vector $F_2(f^0, u)$ is used. In these three latter segmentations, one can note that the boundaries are not smooth. By using a modified nonlocal configuration graph, Figure 4(f) shows a better segmentation where the boundaries are more smoother. In this case, we use a graph defined as 16-NNG $\cup$ G_1. The nearest neighbors are selected with a patch distance where the feature vector is $F_5(f^0, u)$ within a $15{\times}15$ neighborhood search window. Finally, the weight function associated with this graph is $w = 1$.

5. MATHEMATICAL MORPHOLOGY

The two fundamental operators in Mathematical Morphology are dilation and erosion. They form the basis of many other morphological processes such as opening, closing, reconstruction, leveling, etc [26].

These two operations are commonly defined in terms of algebraic set operators but alternative formulations, based on PDEs was also proposed by [23, 24] and references therein. For a unit disc $B = \{z \in \mathbb{R}^2 : \|z\|_p \leq 1\}$, PDEs-based methods generate flat dilation and erosion of a scalar function $f^0 : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ by B with the following diffusion equations: $\delta_t(f) = \partial_t f = +|\nabla f|$ and $\epsilon_t(f) = \partial_t f = -|\nabla f|$, where $\nabla = (\partial x, \partial y)^T$ is the spatial gradient operator and f is the transformed version of f^0. If one assumes that the evolution at time $t{=}0$ is initialized with $f(x, y, 0) = f^0(x, y)$, solution of $f(x, y, t)$ at time $t{>}0$ provides dilation (with the plus sign) or erosion (with the minus sign) within a disc of radius t. These PDEs produce continuous scale morphology and have the advantages of offering excellent results for non-digitally scalable structuring elements whose shapes cannot be correctly represented on a discrete grid; allowing sub-pixel accuracy and can be adaptive by introducing a local speed evolution term [43].

In this section, we present our morphological framework based on discrete derivatives and PdEs. The proposed formulation extends local PDEs-based approaches to nonlocal configuration in context of image processing. In the sequel, we introduce our dilation and erosion processes based on previously defined discrete operators. Then, links with well-known MM morphology methods are discussed and we show that formulations are special cases of our methodology. Finally experiments in image processing show the benefits of weighted and nonlocal operations for image morphological processing that better preserve edges, fine and repetitive image structures.

5.1. Dilatation and erosion processes

In this section, we define the discrete analogue of the continuous PDEs-based dilation and erosion formulations. One wants to obtained the two following dilation and erosion processes over graphs:

$$\begin{aligned} \delta_{p,t}(f) &= \frac{\partial f}{\partial t} = +\left\|\nabla^+ f\right\|_p \text{ and} \\ \epsilon_{p,t}(f) &= \frac{\partial f}{\partial t} = -\left\|\nabla^- f\right\|_p \ , \end{aligned} \tag{26}$$

where $\|.\|_p$ corresponds to the L^p-norm.

To establish these two morphological processes, we use on the one hand, the decomposition of a function $f : V \to \mathbb{R}$ into its level sets $f^k = H(f-k)$ where H is the Heaviside function and, on the other hand, the graph boundaries notion defined in (1). Then, one can interpret dilation process over A as a growth process that adds vertices from $\partial^+ A$ to A. By duality, erosion process can be interpreted over A as a contraction process that removes vertices from $\partial^- A$.

One can demonstrate, the relation between the graph boundary and the gradient norm of the level set function at vertex $u \in V$:

$$\left\|\nabla^+ f^k(u)\right\|_p \text{ and } \left\|\nabla^- f^k(u)\right\|_p \ ,$$

by studying cases where $u \in A^k$ or $u \notin A^k$ and similarly for $v \sim u$ (see [44] for more details). Then, for any level set f^k, the L^p-norm (with $0 < p < +\infty$) of the directional gradients $\left\|\nabla^+ f^k(u)\right\|_p$ and $\left\|\nabla^- f^k(u)\right\|_p$ at a vertex $u \in V$ are

$$\begin{aligned} \left\|\nabla^+ f^k(u)\right\|_p &= \left(\sum_{v\sim u, v\in A^k} w_{uv}^{p/2}\right)^{1/p} \chi_{\partial^+ A^k}(u) \text{ and} \\ \left\|\nabla^- f^k(u)\right\|_p &= \left(\sum_{v\sim u, v\in A^k} w_{uv}^{p/2}\right)^{1/p} \chi_{\partial^- A^k}(u) \ , \end{aligned} \tag{27}$$

where $\chi : V \to \{0,1\}$ is the indicator function and $A^k \subset V$ is the set such that $f^k = \chi_{A^k}$.

Directly from (27) and by using the inner and outer boundaries $\partial^+ A^k$ and $\partial^- A^k$ (see [44] for the proof), one obtains the following relation or any level set f^k with $0 < p < +\infty$:

$$\left\|(\nabla f^k)(u)\right\|_p = \left\|(\nabla^+ f^k)(u)\right\|_p + \left\|(\nabla^- f^k)(u)\right\|_p \ . \tag{28}$$

Equations (27) and (28) only consider the L^p-norm when $0 < p < +\infty$. For the case where $p = \infty$ one can demonstrate and obtain same results by using L^∞-norms expressions.

Dilation and erosion processes. A simple variational definition of dilation applied to f^k can be interpreted as maximizing a surface gain proportional to the gradient norm $+\left\|(\nabla f^k)(u)\right\|_p$. Similarly, erosion is a surface gain minimization proportional to $-\left\|(\nabla f^k)(u)\right\|_p$.

Dilation of f^k on A^k can be expressed by the following evolution equation:

$$\partial f^k(u)/\partial t = +\left\|(\nabla^+ f^k)(u)\right\|_p \ ,$$

where $\left\|(\nabla f^k)(u)\right\|_p$ is reduced to $\left\|(\nabla^+ f^k)(u)\right\|_p$ for $u \in \partial^+ A^k$ by using Equations (27) and (28). Similarly erosion process can be expressed by:

$$\partial f^k(u)/\partial t = -\left\|(\nabla^- f^k)(u)\right\|_p \ .$$

Finally, by extending these two processes for all the levels of f, we can obtain the two processes expressed by Equations (26) and parametrized by p and w, over any weighted graph $G = (V, E, w)$:

$$\begin{aligned} \delta_{p,t}(f) &= \frac{\partial f}{\partial t} = +\left\|\nabla^+ f\right\|_p \text{ and} \\ \epsilon_{p,t}(f) &= \frac{\partial f}{\partial t} = -\left\|\nabla^- f\right\|_p \ . \end{aligned}$$

5.2. Dilation algorithm

To solve the partial difference equations of dilation and erosion processes, on the contrary to the PDEs case, no spatial discretization is needed thanks to derivatives directly expressed in a discrete form. Then, one obtains the general iterative scheme for dilation, at time $t+1$, for all $u \in V$,

$$f^{t+1}(u) = f^t(u) + \Delta t\left\|(\nabla^+ f^t)(u)\right\|_p \ . \tag{29}$$

With the corresponding p values, the iterative scheme becomes for $0 < p < +\infty$,

$$\begin{aligned} &f^{t+1}(u) = f^t(u) + \\ &\qquad \Delta t\left(\sum_{v\sim u} w_{uv}^{\frac{p}{2}} \left(\max\bigl(0, f^t(v) - f^t(u)\bigr)\right)^p\right)^{\frac{1}{p}} \ , \end{aligned} \tag{30}$$

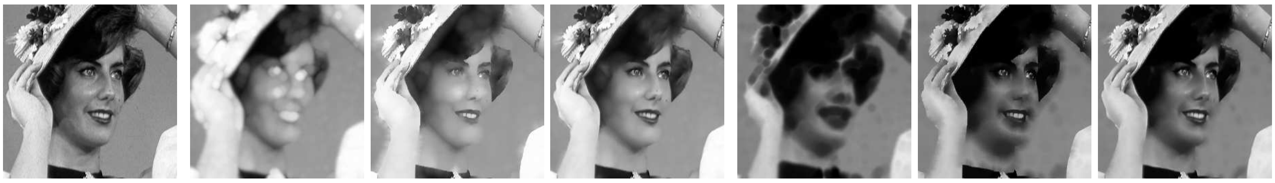
and for $p = \infty$

$$\begin{aligned} &f^{t+1}(u) = f^t(u) + \\ &\qquad \Delta t \max_{v\sim u}\Bigl(\sqrt{w_{uv}} \max\bigl(0, f^t(v) - f^t(u)\bigr)\Bigr) \ , \end{aligned} \tag{31}$$

where $f^0 \in \mathcal{H}(V)$ is the initial function defined on the graph vertices, $f^{(0)}(u) = f^0(u)$ is the initial condition, t the iteration step, and Δt is the time discretization. The extension to erosion process case can be established by using the corresponding gradient $\nabla^- f$.

5.3. Related schemes in image processing

With an adapted graph structure and an appropriated weight function, our propose morphological framework recovers well-known morphological methods in image processing. For clarity, we only consider dilation but same



(a) Original (b) Local $w = 1$ (c) Local weighted (d) Nonlocal (e) Local $w = 1$ (f) Local weighted (g) Nonlocal

Figure 5. Image morphological processing with different graph topologies and weight functions. First row: dilation. Second row: erosion.

remarks can be obtained for erosion.

Osher-Sethian discretization scheme. When $p = 2$ and the weight function is constant ($w = 1$), Equation (30) recovers the exact Osher-Sethian first order upwind discretization scheme [45] for a grayscale image defined as $f^0 : V \subset \mathbb{R}^2 \to \mathbb{R}$. If the associated graph is G_0 then, with Equation (30) and the following property $\big(\max(0, a-b)\big)^2 = \big(\min(0, b-a)\big)^2$, we have:

$$\begin{aligned} f^{t+1}(x,y) &= f^t(x,y) \\ &+ \Delta t\Big(\big(\min\big(0, f^t(x,y) - f^t(x-1,y)\big)\big)^2 + \\ &\big(\max\big(0, f^t(x+1,y) - f^t(x,y)\big)\big)^2 + \\ &\big(\min\big(0, f^t(x,y) - f^t(x,y-1)\big)\big)^2 + \\ &\big(\max\big(0, f^t(x,y+1) - f^t(x,y)\big)\big)^2\Big)^{\frac{1}{2}} \quad , \end{aligned} \tag{32}$$

where vertex $u \in V$ and its neighborhood $u \sim v$ are replaced by their spatial image coordinates (x, y).

It corresponds exactly to the Osher and Sethian discretization scheme [45] of the PDEs-based dilation process. Using this expression, the proposed morphological framework can perform a sub-pixel approximation and recovers the notion of structuring elements [23]. For a unit ball $B = \big\{z \in \mathbb{R}^2 : \|z\|_p \leq 1\big\}$, if we consider the three special cases of $p = 1, 2, \infty$, one obtains an approximation of a square, circle and diamond.

Algebraic formulation. If we consider the neighborhood of a vertex $u \in V$ with the vertex itself and by studying the sign of the quantity $f^t(v) - f^t(u)$; when $p = \infty$, with a constant time discretization (i.e. $\Delta t = 1$) and a constant weight function ($w = 1$), Equation (31) recovers the algebraic formulation of dilation over graphs.

$$f^{t+1}(u) = \max_{v \sim u}\big(f^t(v)\big) \quad . \tag{33}$$

In this case, the structuring element is provided by the graph structure and the neighborhood of the vertices. For instance, if we consider a 8-adjacency image grid graph, it is equivalent to a dilation by a square structuring element of size 3×3.

5.4. Experimentations in image processing

The following experimentations show the potentialities of the proposed morphological framework for image processing. The examples illustrate the flexibility and the ability of our method to perform different morphological processing within a same formulation.

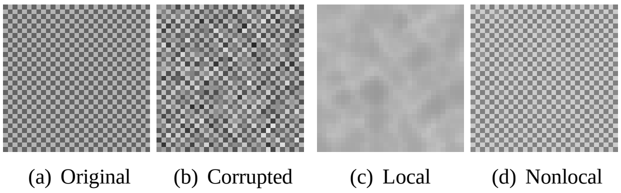
Image morphological processing. Figure 5 shows a comparison between local unweighted, local weighted and nonlocal patch-based dilation and erosion. The graph associated with local processing is the 4-adjacency grid graph G_0, where for the weighted case, the weight function is $w = g_1$ with $F(f^0, u) = f^0(u)$. For the nonlocal case, the graph is G_3 associated with the weight function $w = g_1$ where the feature vector is $F_1(f^0, u)$. These results show that by using non constant weights, the proposed dilation and erosion better preserve edges as compared to classical approaches. When a nonlocal configuration is used fine structures and repetitive elements are better preserved.

Morphological processing for textured images. Figure 6 illustrates one of the novelties of our framework, the application of nonlocal approach for morphological processing. Figure 6 shows a comparison between local and nonlocal closing. Closing can be defined as a serial composition of dilation (δ) and erosion (ε) operations. The closing of a function f is $\varepsilon\left(\delta(f)\right)$. Figure 6 shows closing of an corrupted image (Figure 6(b)) from the initial Figure 6(a) with a Gaussian noise of $\sigma = 20$. The local closing is performed with the graph G_0 and the associated weight function $w = 1$. For the nonlocal morphological closing, the associated graph is 10-NNG $\cup\, G_1$ (same construction is defined in Figure 4(f)) where the feature vector is $F_3(f^0, u)$ and the patch distance is computed within a 21×21 search window. This example clearly demonstrates the efficiency of nonlocal patch-based methods to better preserve frequent features during the morphological process. Contrary to local ones that destroy fine structures and repetitive elements.

6. DATA PROCESSING ON ARBITRARY GRAPH

Our regularization and MM frameworks work on graphs of arbitrary topology. One of the advantages is that we can use our methodology on any discrete data that can be represented by a weighted graph. As a result, our formulation provides a natural extension of PDEs-based methods to process any discrete data even if they are defined in a high dimensional domain.

In the sequel, through different experimentations and



(a) Original (b) Corrupted (c) Local (d) Nonlocal

Figure 6. Local versus nonlocal patch-based texture image closing. First row: original and corrupted image with Gaussian noise ($\sigma = 20$). Second row: local and nonlocal closing results (See text for more details).

applications in regularization and morphological processing, we show the potentialities of our approaches to process unorganized high dimensional data set. Moreover, we also show that another graph-based image representation can be used instead of usual pixel-based grid graph leading to fast image processing.

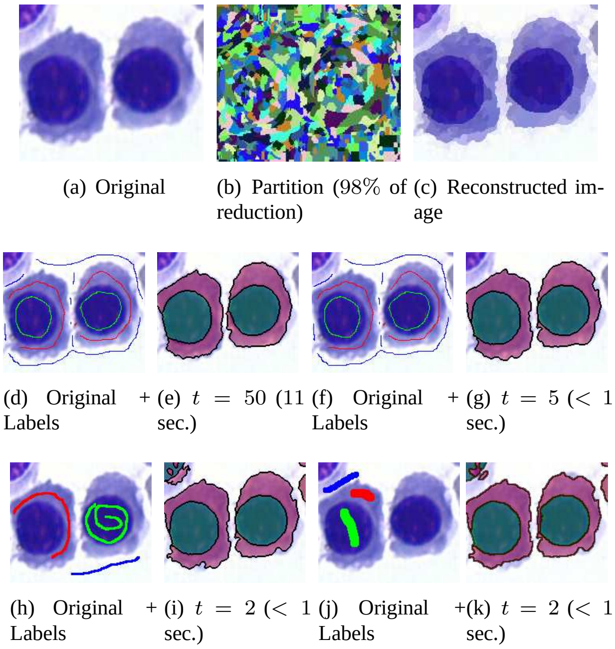
The regularization process used in this section is the neighborhood filter of Section 3.3 with $p{=}2$ and $\phi(s){=}s^2$, i.e. Laplacian smoothing.

6.1. Fast image processing on partitions

If we consider that image pixels are not the only relevant elements, then more abstract structures can be used such as image regions or superpixels [46]. We suggest to work directly with reduced versions of images: image partitions. Constructing image partitions can be viewed as an image simplification or a data reduction process. To obtain such image partitions, any well known image pre-segmentation can be performed such as watershed techniques.

In this paper, we use generalized Voronoi diagrams (for more details see [47]). One of the advantages of this method is the low computing time to obtain a complete image partition. Indeed, the amortized time complexity of a such method is $\mathcal{O}(E{+}V\log V)$ with Dikjstra algorithm and Fibonacci heap structure. Then, the obtained partition can be associated with any graph topology such as Region Adjacency Graph (RAG), proximity graphs or a fully connected graph (G_∞) where vertices represent image regions. The function to be processed on such graphs are defined at each vertex of V as the mean value of its associated region.

In the sequel, we show that with this image representation, one obtains similar processing behaviors than pixel-based processing, but with a drastically decreasing computation complexity. Due to the low computing time to create an image partition, it can be neglected in the following experiments (e.g. to obtain partitions of an image of size $256{\times}256$ take less than 1 sec. on a modern computer [1]).



(a) Original (b) Partition (98% of reduction) (c) Reconstructed image

(d) Original + Labels (e) $t = 50$ (11 sec.) (f) Original + Labels (g) $t = 5$ (< 1 sec.)

(h) Original + Labels (i) $t = 2$ (< 1 sec.) (j) Original + Labels (k) $t = 2$ (< 1 sec.)

Figure 7. Semi-supervised image segmentation with $p{=}2$, $\lambda{=}1$, t iterations for different graph structures and user input strokes. (a), (b) and (c): original ($152{\times}181$ pixels), partition, reconstructed images. (d), (f), (h) and (j): user input labels. (e), (g), (i) and (k): original image with the obtained segmented regions superimposed: cytoplasm (red), nuclei (green) and regions boundaries (black); the segmentation is performed with the specified iteration steps t and the corresponding computation time. The images (e), (g), (i) and (k) are respectively obtained from label images (d), (f), (h) and (j). Graph structures used to obtain results (e): G_1,(g): RAG,(i) and (k):G_∞.

Fast semi-supervised image segmentation. Image semi-supervised segmentation are usually based on label diffusion strategies on grid graphs [40, 41], such as the one presented in Section 4. The drawback of this method is that when the considered image is large, the label propagation method is inefficient due to the great mass of data to analyze. To avoid this computational problem, we propose to use image partitions [48].

Figure 7 shows the proposed semi-supervised clustering method applying to segment cytological images into 3 classes (nuclei, cytoplasm and background). This experiment also show how partitions in addition of nonlocal scheme can provide an efficient image segmentation method. To this aim, the following experiment compares on the one hand, computation time and the segmentation results between a pixel-based grid graph, and two region-based proximity graphs (RAG and fully connected graph); and one the other hand, it shows the robustness of our approach regarding to initial user input labels. Figure 7(e) is the semi-supervised segmentation result obtained from labels of Figure 7(d) and an 8-adjacency grid graph (G_1) associated with the initial image (Figure 7(a)). In this case, one can observe the number and the precise location of the initial labels, in particular, the necessary labels between

[1] All the results are obtained with a standard Linux computer equipped with quadri 2.4 GHz Intel Xeon processors and 16 GB of RAM, and the mentioned computing times include the graph construction itself.

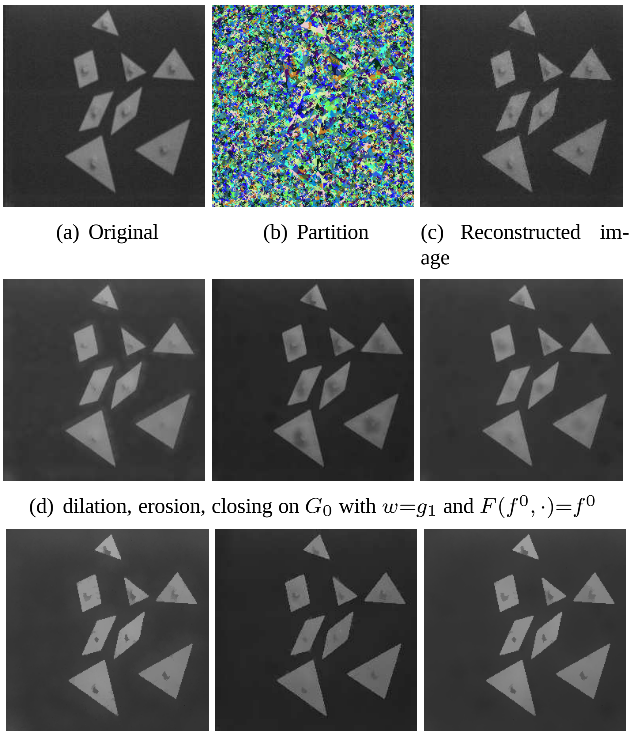
the two cells. Figure 7(b) is a partition of Figure 7(a). One can observe the important rate of reduction (98%) in term of graph vertices. Figure 7(c) is a reconstructed image from the partition where the pixel values of each region of the partition are replaced by the mean pixel value of its regions. With this simplified version, we construct two proximity graphs: the RAG and the fully connected graph. Figure 7(g) shows the segmentation result obtain from the RAG with the same initial labels (Figure 7(f)) as in the grid graph case. We can observe that the two results (Figures 7(g) and 7(e)) are similar but in the RAG-based segmentation case, the computation time is significantly reduced. Figures 7(i) and 7(k) show the segmentation result obtained from the fully connected graph. Using this graph topology has several advantages. First, the graph contains all the image information within the edge weights. Second, a minimal number of labels is needed to obtain correct results as compared to the case of the grid-graph or of the RAG. Third, this nonlocal approach has the important property to quickly label objects in the same class, even if they are not spatially adjacent or close. In Figures 7(i) and 7(k), the two main nuclei and cytoplasm are segmented even if they have no initially been labeled, and the two pieces of cytoplasm on the left and the piece of cells on the top-left corner of the image are also found. Finally, the robustness of our approach is shown by two similar results (Figures 7(i) and 7(k)) with two different user input labels (Figures 7(h) and 7(j)).

Fast image morphological processing. Figure 8 compares the behavior of our image morphological processing between pixel-based and partitions-based graphs.

Figure 8(b) shows an image partition obtained from the initial image of Figure 8(a), and Figure 8(b) is a reconstructed image from the partition. The initial image has size 256×256, and the partition is a significant reduced version (82% of reduction in term of vertices) as compared to the original one. Figures 8(d) and 8(e) show dilation, erosion and closing respectively performed on the 4-adjacency grid graph associated to the original image and on the RAG associated to the partition. Both cases exhibit similar behaviors while the case of RAG reduces drastically the computation complexity. This is due to the reduced number of vertices to consider.

6.2. Processing of high dimensional unorganized data

In this section, we show one of the advantages of our formulation, the application of regularization and mathematical morphology on high dimensional unorganized sets of data. In the sequel, different experiments show the potentialities of our methodology to smooth discrete data with regularization or mathematical morphology, or to classify data set by semi-supervised clustering.



(a) Original

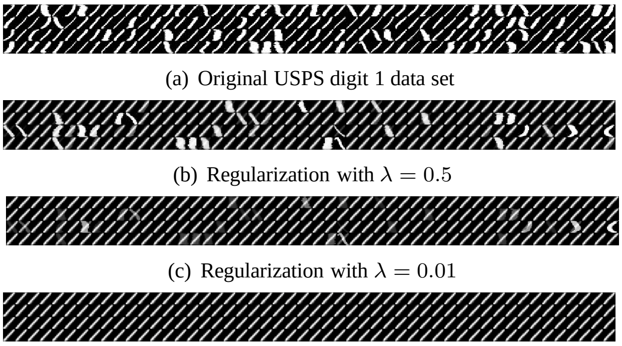
(b) Partition

(c) Reconstructed image

(d) dilation, erosion, closing on G_0 with $w{=}g_1$ and $F(f^0,\cdot){=}f^0$

(e) dilation, erosion, closing on the RAG with $w{=}g_1$ and $F(f^0,\cdot){=}f^0$

Figure 8. Fast morphological image processing. (d): image pixel based grid graph processing. (e): image partitions based RAG processing.



(a) Original USPS digit 1 data set

(b) Regularization with $\lambda = 0.5$

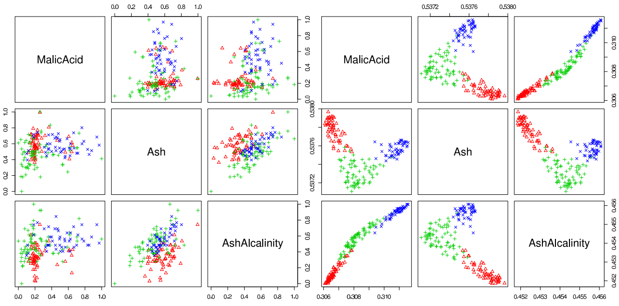
(c) Regularization with $\lambda = 0.01$

(d) Regularization with $\lambda = 0$

Figure 9. USPS data set regularization. (b), (c) and (d): results obtained with the corresponding λ parameter.

Unorganized data set regularization. In the following experiments, we consider two real-world high dimensional data set. On the one hand, the United States Postal Service (USPS) handwritten digits database and, on the other hand the UCI Wine database. USPS database contains grayscale handwritten digit images scanned from digit 0 to 9 where each image is of size 16×16 pixels. Wine database contains 3 classes of samples in 14-dimensions and for each class 59, 71, and 48 samples. Coming from the real-world, the data sets naturally contain noise, and one wants to recover a denoised sub-data sets. To perform this task, we use the proposed regularization process. Figures 9 and 10 show the regularization results obtain respectively for the USPS and Wine data sets.

To perform regularization of USPS data set shown in Figure 9, we use a randomly subsampled set of 200 samples from the original 1 digit set. Figure 9(a) shows the test data set. The fully connected graph G_∞ is built with



(a) Original data set (b) Regularization results

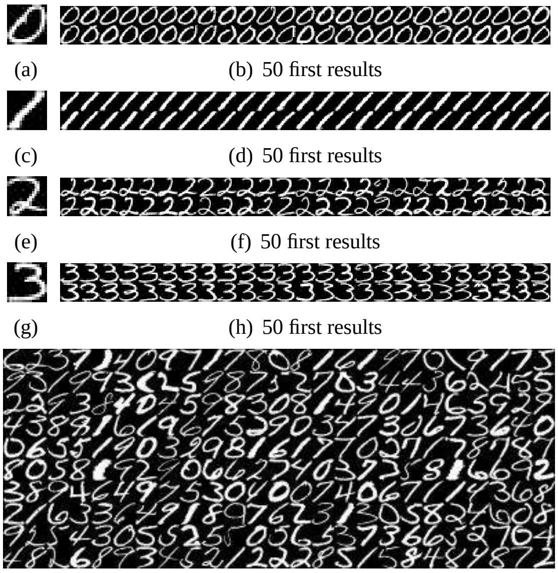
Figure 10. UCI Wine data set regularization with $p{=}2$, $\lambda{=}0$, and $t{=}1$. For simplicity, only Malic Acid, Ash, and Ash Alkalinity features pairs projections are shown.

the weight function $w{=}g_1$. Each vertex of the graph corresponds to an image sample and is described by a 256-dimensions ($\mathbb{R}^{16\times16}$) feature vector where each feature is a pixel grayscale value. Figure 9 shows several regularization results. One can note in these figures, that all samples are strongly transformed, in particular when the fidelity term $\lambda{=}0$ (Figure 9(d)). All samples become uniformly identical and converge to an artificial mean digit sample. When $\lambda{\neq}0$, the samples are smoothed but the more dissimilar ones preserve their main shape as shown in Figures 9(b) and 9(c). Figure 10 shows the regularization of Wine database. Due to the high dimensionality of the data set only few relevant feature pair projections are shown. Each different color corresponds to a specific class for the represented data set. The fully connected graph is built on the data with the weight function $w{=}g_2$. Each vertex of the graph corresponds to a data point and is described by a 14-dimensions feature vector. Figure 10(b) illustrates that the data regularization has the interesting behavior to naturally group all the samples in different parts of the feature space in comparison with the initial organization where they are completely mixed (Figure 10(a)).

These experiments show the potentialities of our method to process unorganized data. Finally, this data regularization can be viewed as a data pre-processing that can be used to improve the efficiency of final data classification or machine learning methods.

Semi-supervised clustering for unorganized data information retrieval. Starting from a user query, a classical Information Retrieval (IR) task consists in matching objects stored in a database. Then, the system presents to the user an ordered result depending on the relevance with the initial query. In semi-supervised learning term, user query is an unlabeled object, and the retrieved objects are the more similar labeled objects contained in the database.

Figure 11 shows an IR application performed on the USPS handwritten digits data set. For convenience, a subset of the original database is used. The labeled set consists into 800 randomly selected elements from original ones for all the digits (from 0 to 9), i.e. the initial labeled set is composed of 8000 samples. All the elements are labeled with a corresponding digit class. Figure 11(i) shows



(a) (b) 50 first results

(c) (d) 50 first results

(e) (f) 50 first results

(g) (h) 50 first results

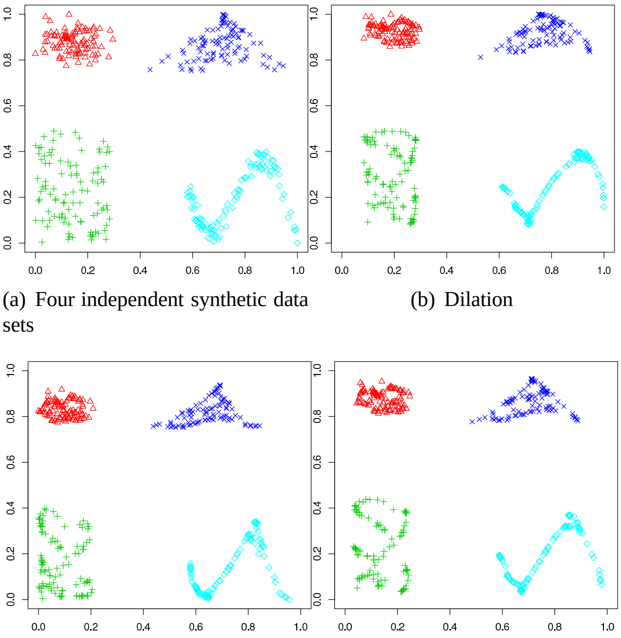
(i) Sample of the initial labeled data set

Figure 11. USPS image retrieval based on semi-supervised clustering. (a), (c), (e) and (g): user input query (unlabeled point). (b), (d), (f) and (h): 50 first obtained results for the corresponding query. (i): sample of the initial labeled data set, original one contains $8\,000$ points.

250 elements of the initial labeled set. The sample query is randomly selected from the unused elements. In this example, we test the digit from 0 to 3. Figures 11(a), 11(c), 11(e) and 11(g) show the initial input query: the unlabeled data. The IR task consists in estimating and ranking the more similar images from the labeled samples according to the user query. In our experiment, we show the 50 first samples (Figures 11(b), 11(d), 11(f) and 11(h)) founded by the semi-supervised clustering method and classified according to Equation (25). The fully connected graph is computed in connection with the weight function $w{=}g_2$. The general parameters are $\lambda{=}1$, and $t{=}1$. As shown by the results, one can notice the correctness of retrieved images from initial user query even with the use of simple Euclidean distance.

Mathematical morphology processing for unorganized data set. In the following experiments, as for the regularization case, we show how our morphological framework can be applied to process high dimensional unorganized data sets. Figures 12 and 13 shows morphological processing: dilation, erosion and opening on respectively four independent synthetic unorganized data sets and the USPS database. As for closing case, opening can be defined as a serial composition of dilation (δ) and erosion (ε) operations. The opening of a function f is $\delta\left(\varepsilon(f)\right)$.

Figure 12 shows morphological openings performed on four independent synthetic unorganized data sets. For each set, we compute the 8-NNG from the original data with weight the function $w{=}g_2$. Figures 12(b) and 12(c) show the results of dilation and erosion. Figure 12(d) the opening operation. One can note that opening acts as fil-



(a) Four independent synthetic data sets

(b) Dilation

(c) Erosion

(d) Opening

Figure 12. Synthetic data set morphological processing

tering or denoising processes on the data, and works as grouping operation where points tend to shrink into main data structures.

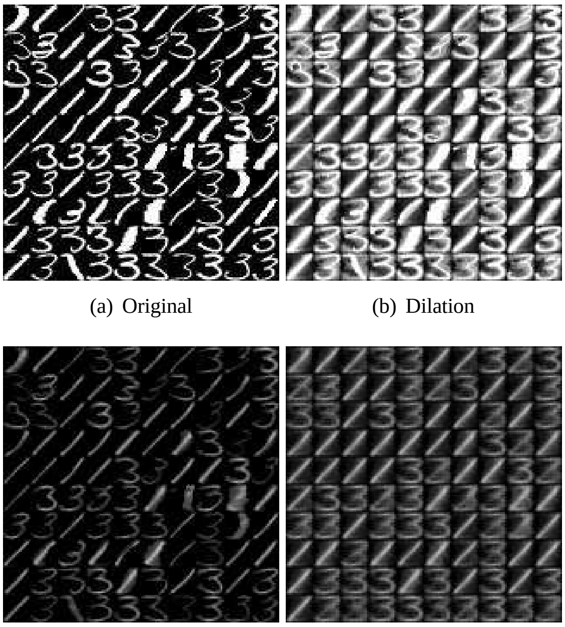
Figure 13 shows the processing of USPS images database. This experiment consists of 100 samples randomly selected and mixed from digits digit 1 and 3. Figure 13(a) shows the original test set. From the original data, a 30-NNG is computed associated with the weight function $w{=}g_1$, where each vertex of the graph corresponds to an image sample and is described by a 256-dimensions ($\mathbb{R}^{16\times 16}$) feature vector where each feature is an image pixel grayscale value. Figures 13(b) and 13(c) shows the dilatation and erosion results. Figure 13(d) presents the opening result. It shows that the opening tends to reduce the data set to the main artificial digits.

This two experiments show that the application of morphology on data sets can be useful for classification purpose by extracting noiseless sub-data sets from noisy ones.

7. CONCLUSION

In this paper, we have presented a graph-based framework that unifies local and nonlocal processing in the context of gradient-based regularization and mathematical morphology. This unification is achieved by defining explicitly discrete derivatives over weighted graphs, and by choosing the graph topology and geometry.

Through several experiments, we have shown the efficiency of the proposed nonlocal regularizations and mathematical morphology processing. In particular, they better preserve sharp edges, as well as fine and repetitive structures than local ones. The application of our methodology to process unorganized data sets leads to a set of tools that can be useful to denoise, smooth or simplify these data. It can be used as pre-processing steps in classification processes.



(a) Original

(b) Dilation

(c) Erosion

(d) Opening

Figure 13. USPS data set morphological processing.

8. ACKNOWLEDGMENTS

The work of Vinh-Thong Ta was partially supported under a research grant of the ANR Foundation (ANR-06-MDCA-008-01/FOGRIMMI) and a doctoral grant of the Conseil Régional de Basse-Normandie and of the Cœur et Cancer association in collaboration with the Department of Anatomical and Cytological Pathology from Cotentin Hospital Center.

The work of Sébastien Bougleux was partially supported by ANR Grant SURF-NT05-2 45825.

9. REFERENCES

[1] 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.

[2] J. Weickert, *Anisotropic Diffusion in Image Processing*, ECMI series. Teubner-Verlag, 1998.

[3] N. Paragios, Y. Chen, and O. Faugeras, Eds., *Handbook of Mathematical Models in Computer Vision*, Springer, 2005.

[4] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing, Partial Differential Equations and the Calculus of Variations*, Number 147 in Applied Mathematical Sciences. Springer, 2nd edition, 2006.

[5] G. Gilboa and S. Osher, “Nonlocal operators with applications to image processing,” Report CAM 07-23, UCLA, Los Angeles, 2007.

[6] G. Gilboa and S. Osher, "Nonlocal linear image regularization and supervised segmentation," *SIAM Multiscale Modeling and Simulation*, vol. 6, no. 2, pp. 595–630, 2007.

[7] S. Kinderman, S. Osher, and S. Jones, "Deblurring and denoising of images by nonlocal functionals," *SIAM Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1091–1115, 2005.

[8] L.P. Yaroslavsky, *Digital picture processing–an introduction*, Springer, 1985.

[9] J.S. Lee, "Digital image smoothing and the sigma filter," *Computer Vision, Graphics, and Image Processing*, vol. 24, no. 2, pp. 255–269, 1983.

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

[11] 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.

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

[13] 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, pp. 1325–1368, 2007.

[14] 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*, 2006, vol. 6232.

[15] A. Szlam, M. Maggioni, and R. Coifman, "A general framework for adaptive regularization based on diffusion processes on graphs," Tech. Rep. YALE/DCS/TR1365, YALE, 2006.

[16] L.I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1-4, pp. 259–268, 1992.

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

[18] T. Chan, S. Osher, and J. Shen, "The digital TV filter and nonlinear denoising," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 231–241, 2001.

[19] D.Y. Zhou and B. Scholkopf, "Regularization on discrete spaces," in *German Pattern Recognition Symposium*. 2005, vol. 3663 of *LNCS*, pp. 361–368, Springer.

[20] S. Bougleux, A. Elmoataz, and M. Melkemi, "Discrete regularization on weighted graphs for image and mesh filtering," in *1st Int. Conf. on Scale Space and Variational Methods in Computer Vision (SSVM)*. 2007, vol. 4485 of *LNCS*, pp. 128–139, Springer.

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

[22] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1047–1060, 2008.

[23] R.W. Brockett and P. Maragos, "Evolution equations for continuous-scale morphology," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1992, vol. 3, pp. 125–128.

[24] P. Maragos, "PDEs for morphology scale-spaces and eikonal applications," in *The Image and Video Processing Handbook*, chapter 4.16, pp. 587–612. Elsevier Academic Press, second edition, 2004.

[25] H. Heijmans, P. Nacken, A. Toet, and L. Vincent, "Graph morphology," *Journal of Visual Communication and Image Representation*, vol. 3, no. 1, pp. 24–38, March 1992.

[26] P. Soille, *Morphological Image Analysis, Principles and Applications*, Springer, second edition, 2002.

[27] F. Meyer and R. Lerallut, "Morphological operators for flooding, leveling and filtering images using grpahs," in *In Proceedings of the 6th IAPR-TC-15 GbRPR*, 2007, vol. 4538 of *LNCS*, pp. 158–167.

[28] M. Requardt, "A new approach to functional analysis on graphs, the connes-spectral triple and its distance function," 1997.

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

[30] J. O'Rourke and G. Toussaint, "Pattern recognition," in *Handbook of discrete and computational geometry*, chapter 51, pp. 1135–1162. Chapman & Hall/CRC, 2004.

[31] A. Efros and T. Leung, "Texture synthesis by nonparametric sampling," in *Proc. of the International Conference on Computer Vision (ICCV)*. 1999, pp. 1033–1038, IEEE Computer Society.

[32] C. Kervrann, J. Boulanger, and P. Coupé, "Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal," in *Proc. of the*

1st Int. Conf. on Scale Space and Variational Methods in Computer Vision (SSVM). 2007, vol. 4485 of *LNCS*, pp. 520–532, Springer.

[33] S. Bougleux, *Reconstruction, Détection et Régularisation de Données Discrètes*, Ph.D. thesis, Université de Caen Basse-Normandie, 2007.

[34] A.N. Tikhonov and V.Y. Arsenin, *Solutions of ill-posed problems*, Winston & Sons, 1977.

[35] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, 1990.

[36] S.M. Smith and J.M. Brady, "SUSAN-a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[37] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

[38] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1120–1129, 2006.

[39] O. Lézoray, V.-T. Ta, and A. Elmoataz, "Nonlocal graph regularization for image colorization," in *ICPR 2008*, 2008, p. to appear.

[40] F. Wang, C. Zhang, H.C. Shen, and J. Wang, "Semi-supervised classification using linear neighborhood propagation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. June 2006, vol. 1 of *IEEE Conference Proceedings*, pp. 160–167, IEEE Computer Society.

[41] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1778, 2006.

[42] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[43] M. Breuß, B. Burgeth, and J. Weickert, "Anisotropic continuous-scale morphology," in *In Proceedings of the 3rd IbPRIA*. 2007, vol. 4478 of *LNCS*, pp. 512–522, Springer.

[44] V.-T Ta, A. Elmoataz, and O. Lézoray, "Partial difference equations over graphs: Morphological processing of arbitrary discrete data," in *The 10th ECCV*, 2008, to appear.

[45] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[46] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings Ninth IEEE International Conference on Computer Vision*, October 2003, vol. 1, pp. 10–17.

[47] P.A. Arbeláez and L.D. Cohen, "Energy partitions and image segmentation," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1–2, pp. 43–57, January–March 2004.

[48] V.-T. Ta, O. Lézoray, and A. Elmoataz, "Graph based semi and unsupervised classification and segmentation of microscopic images," in *The 7th ISSPIT 2007*, December 2007, pp. 1177–1182.