Partial Difference Equations (PdE) on graphs for image and data processing

Olivier Lézoray

Université de Caen Basse Normandie
olivier.lezoray@unicaen.fr
http://www.info.unicaen.fr/~lezoray/

May 18, 2011

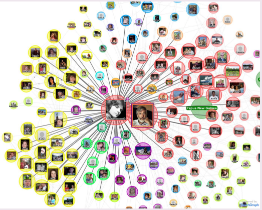
Joint work with A. Elmoataz.

A deluge of many different kinds of data

Different organization

- Images
- Videos
- Meshes
- Social, complex, biological Networks
- ...
- And databases of them



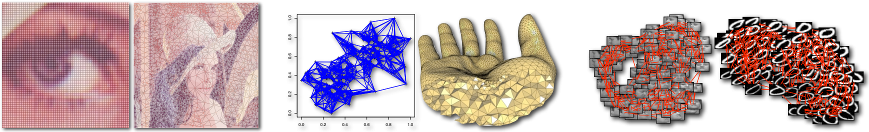


Different nature

- High dimensions,
- Non-linear,
- Heterogeneous,
- Noisy, redundant, incomplete

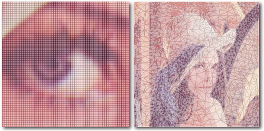
From image and data to graphs

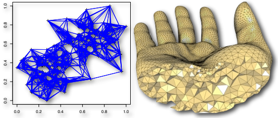
Graphs occur as a the most natural of representing arbitrary data by modeling the neighborhood properties between these data (organized or not).

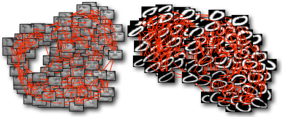


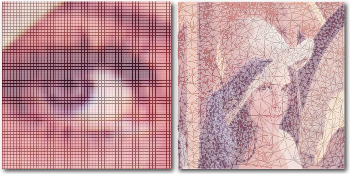
From image and data to graphs

Graphs occur as a the most natural of representing arbitrary data by modeling the neighborhood properties between these data (organized or not).





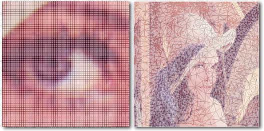


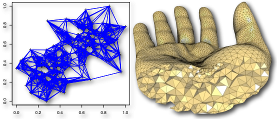


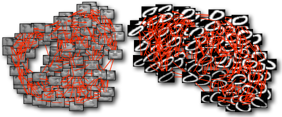
Images, Region adjacency graphs

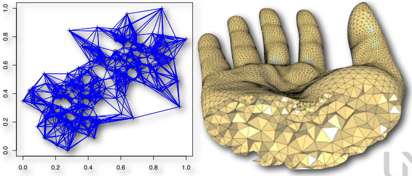
From image and data to graphs

Graphs occur as a the most natural of representing arbitrary data by modeling the neighborhood properties between these data (organized or not).





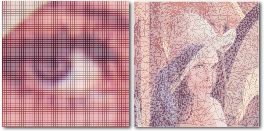


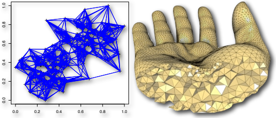


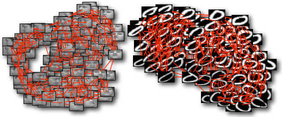
Points clouds, meshes

From image and data to graphs

Graphs occur as a the most natural of representing arbitrary data by modeling the neighborhood properties between these data (organized or not).







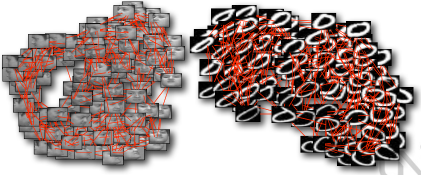


Image databases

Typical problems in image and data processing

Given some input Data $\mathbf{X}_0$, we want to conceive a processing operator Υ that outputs the processed data $\mathbf{X}_1$.

Typical processing operators

- Restoration, denoising, interpolation
- Smoothing, simplification
- Segmentation, classification
- Dimensionality reduction
- Visualization, exploration

université de Caen
Basse-Normandie

Typical problems in image and data processing

Given some input Data $\mathbf{X}_0$, we want to conceive a processing operator Υ that outputs the processed data $\mathbf{X}_1$.

Typical processing operators

- Restoration, denoising, interpolation
- Smoothing, simplification
- Segmentation, classification
- Dimensionality reduction
- Visualization, exploration

Two problems arise

- How to model and represent the input and output data sources ?
- How to model and formalize the processing operator ?

Typical problems in image and data processing

Given some input Data $\mathbf{X}_0$, we want to conceive a processing operator Υ that outputs the processed data $\mathbf{X}_1$.

Typical processing operators

- Restoration, denoising, interpolation
- Smoothing, simplification
- Segmentation, classification
- Dimensionality reduction
- Visualization, exploration

Two problems arise

- How to model and represent the input and output data sources ?
- How to model and formalize the processing operator ?

Different methods to do this

- Graph theory, spectral analysis **(Mainly for data processing)**
- Continuous variational methods **(Mainly for image processing)**
- A lot of new works aim at extending signal processing for data processing (e.g, diffusion wavelets)
- Partial difference Equations on graphs **(a unified framework for image and data processing)**

Continuous variational methods

Benefits

- Provide a formal framework for the resolution of problems in image processing, computer vision, etc.
- Solutions are obtained by the minimization of appropriate energy functions
- The minimization is usually performed with Partial Differential Equations (PDEs)
- PDEs are discretized to obtain a numerical solution

Limitations

- PDE-based methods are difficult to adapt for data that live on non Euclidean domains
- Indeed, their discretization is difficult for high dimensional data
- Not easy to extend them to advanced representations of data, i.e., graphs
- It is essential to be able to transcript PDEs on graphs

Partial difference Equations on graphs

Motivations

- Problems involving PDEs can be reduced to ones of a very much simpler structure by replacing the differentials by difference equations on graphs.
  R. Courant, K. Friedrichs, H. Lewy, On the partial difference equations of mathematical physics, Math. Ann. 100 (1928) 32-74.
- Our goal is to provide methods that mimic on graphs well-known PDE variational formulations under a functional analysis point of view.
- To do this we use Partial difference Equations (PdE) over graphs.
- PdEs mimic PDEs in domains having a graph structure.

Interest of our proposals

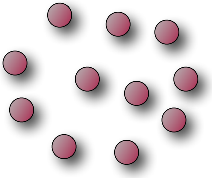
- To dispose of discrete analogues of differential geometry operators (integral, derivation, gradient, divergence, p-Laplacian, etc.)
- To use the framework of PdEs to transcribe PDEs on graphs,
- Provides a natural extension of variational methods on graphs,
- Provides a unification of local and nonlocal processing on images.
- Using weighted graphs provides Adaptive PDEs according to data geometry

What we will talk about

- a nonlocal discrete regularization on graphs as a framework for data simplification and interpolation,
- a formulation of mathematical morphology that considers a discrete version of PDEs-based approaches over weighted graphs,
- an adaptation of the Eikonal equation for data clustering and image segmentation.

Weighted graphs Basics

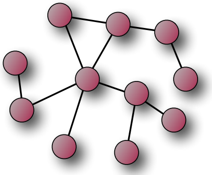
- A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ consists in a finite set $\mathcal{V} = \{v_1, \dots, v_N\}$ of N vertices



université de Caen
Basse-Normandie

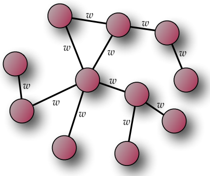
Weighted graphs Basics

- A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ consists in a finite set $\mathcal{V} = \{v_1, \ldots, v_N\}$ of N vertices
- and a finite set $\mathcal{E} = \{e_1, \ldots, e_{N'}\} \subset \mathcal{V} \times \mathcal{V}$ of N' weighted edges.
- We assume $\mathcal{G}$ to be simple, undirected, with no self-loops and no multiple edges.



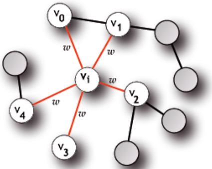
Weighted graphs Basics

- A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ consists in a finite set $\mathcal{V} = \{v_1, \ldots, v_N\}$ of N vertices
- and a finite set $\mathcal{E} = \{e_1, \ldots, e_{N'}\} \subset \mathcal{V} \times \mathcal{V}$ of N' weighted edges.
- We assume $\mathcal{G}$ to be simple, undirected, with no self-loops and no multiple edges.
- $e_{ij} = (v_i, v_j)$ is the edge of $\mathcal{E}$ that connects vertices v_i and v_j of $\mathcal{V}$. Its weight, denoted by $w_{ij} = w(v_i, v_j)$, represents the similarity between its vertices.
- Similarities are usually computed by using a positive symmetric function $w : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$ satisfying $w(v_i, v_j) = 0$ if $(v_i, v_j) \notin \mathcal{E}$.



Weighted graphs Basics

- A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ consists in a finite set $\mathcal{V} = \{v_1, \ldots, v_N\}$ of N vertices
- and a finite set $\mathcal{E} = \{e_1, \ldots, e_{N'}\} \subset \mathcal{V} \times \mathcal{V}$ of N' weighted edges.
- We assume $\mathcal{G}$ to be simple, undirected, with no self-loops and no multiple edges.
- $e_{ij} = (v_i, v_j)$ is the edge of $\mathcal{E}$ that connects vertices v_i and v_j of $\mathcal{V}$. Its weight, denoted by $w_{ij} = w(v_i, v_j)$, represents the similarity between its vertices.
- Similarities are usually computed by using a positive symmetric function $w : \ \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$ satisfying $w(v_i, v_j) = 0$ if $(v_i, v_j) \notin \mathcal{E}$.
- The notation $v_i \sim v_j$ is used to denote two adjacent vertices.



Space of functions on Graphs

- Let $\mathcal{H}(\mathcal{V})$ be the Hilbert space of real-valued functions defined on the vertices of a graph.
- A function $f: \mathcal{V} \to \mathbb{R}$ of $\mathcal{H}(\mathcal{V})$ assigns a real value $x_i = f(v_i)$ to each vertex $v_i \in \mathcal{V}$.
- By analogy with functional analysis on continuous spaces, the integral of a function $f \in \mathcal{H}(\mathcal{V})$, over the set of vertices $\mathcal{V}$, is defined as $\int_{\mathcal{V}} f = \sum_{\mathcal{V}} f$.
- The space $\mathcal{H}(\mathcal{V})$ is endowed with the usual inner product $\langle f, h \rangle_{\mathcal{H}(\mathcal{V})} = \sum_{v_i \in \mathcal{V}} f(v_i)h(v_i)$, where $f, h : \mathcal{V} \to \mathbb{R}$.
- Similarly, let $\mathcal{H}(\mathcal{E})$ be the space of real-valued functions defined on the edges of $\mathcal{G}$.
- It is endowed with the inner product $\langle F, H \rangle_{\mathcal{H}(\mathcal{E})} \sum_{v_i \in \mathcal{V}} \sum_{v_j \sim v_i} F(v_i, v_j)H(v_i, v_j)$, where $F, H: \mathcal{E} \to \mathbb{R}$ are two functions of $\mathcal{H}(\mathcal{E})$.

➲ **Discretization of classical continuous differential geometry**.
The **difference operator** of f, $d_w : \mathcal{H}(\mathcal{V}) \to \mathcal{H}(\mathcal{E})$, is defined on an edge $e_{ij} = (v_i, v_j) \in \mathcal{E}$ by:

$$(d_w f)(e_{ij}) = (d_w f)(v_i, v_j) = w(v_i, v_j)^{1/2}(f(v_j) - f(v_i)) \ . \tag{1}$$

The **adjoint** of the difference operator, noted $d_w^* : \mathcal{H}(\mathcal{E}) \to \mathcal{H}(\mathcal{V})$, is a linear operator defined by $\langle d_w f, H \rangle_{\mathcal{H}(\mathcal{E})} = \langle f, d_w^* H \rangle_{\mathcal{H}(\mathcal{V})}$ for all $f \in \mathcal{H}(\mathcal{V})$ and all $H \in \mathcal{H}(\mathcal{E})$.
The adjoint operator d_w^*, of a function $H \in \mathcal{H}(\mathcal{E})$, can by expressed at a vertex $v_i \in \mathcal{V}$ by the following expression:

$$(d_w^* H)(v_i) = -div_w(H)(v_i) = \sum_{v_j \sim v_i} w(v_i, v_j)^{1/2}(H(v_j, v_i) - H(v_i, v_j)) \ . \tag{2}$$

Each function $H \in \mathcal{H}(\mathcal{E})$ has a null divergence over the entire set of vertices: $\sum_{v_i \in \mathcal{V}} (d_w^* H)(v_i) = 0$.

Difference operators on weighted graphs

The **directional derivative** (or *edge derivative*) of f, at a vertex $v_i \in \mathcal{V}$, along an edge $e_{ij} = (v_i, v_j)$, is defined as $\left.\frac{\partial f}{\partial e_{ij}}\right|_{v_i} = \partial_{v_j} f(v_i) = (d_w f)(v_i, v_j)$.
This definition is consistent with the continuous definition of the derivative of a function: $\partial_{v_j} f(v_i) = -\partial_{v_i} f(v_j)$, $\partial_{v_i} f(v_i) = 0$, and if $f(v_j) = f(v_i)$ then $\partial_{v_j} f(v_i) = 0$.
We also introduce **morphological difference operators**:

$$\begin{aligned} (d_w^+ f)(v_i, v_j) &= w(v_i, v_j)^{1/2} \big(\max\big(f(v_i), f(v_j)\big) - f(v_i)\big) \text{ and} \\ (d_w^- f)(v_i, v_j) &= w(v_i, v_j)^{1/2} \big(f(v_i) - \min\big(f(v_i), f(v_j)\big)\big) \ , \end{aligned} \tag{3}$$

with the following properties

$$(d_w^+ f)(v_i, v) = \max\big(0, (d_w f)(v_i, v_j)\big) \text{ (always positive)}$$
$$(d_w^- f)(v_i, v) = -\min\big(0, (d_w f)(v_i, v_j)\big) \text{ (always negative)}$$

The corresponding external and internal partial derivatives are $\partial_{v_j}^+ f(v_i) = (d_w^+ f)(v_i, v_j)$ and $\partial_{v_j}^- f(v_i) = (d_w^- f)(v_i, v_j)$.

A. Elmoataz, O. Lezoray, S. Bougleux, Nonlocal Discrete Regularization on Weighted Graphs: a framework for Image and Manifold Processing, IEEE transactions on Image Processing, Vol. 17, n7, pp. 1047-1060, 2008.

Weighted gradient operator

The **weighted gradient operator** of a function $f \in \mathcal{H}(\mathcal{V})$, at a vertex $v_i \in \mathcal{V}$, is the vector operator defined by

$$(\nabla_{\mathbf{w}}\mathbf{f})(\mathbf{v_i}) = [\partial_{v_j} f(v_i) \, : \, v_j \sim v_i]^T = [\partial_{v_1} f(v_i), \ldots, \partial_{v_k} f(v_i)]^T, \; \forall (v_i, v_j) \in \mathcal{E}. \quad (4)$$

The $\mathcal{L}_p$ norm of this vector represents the *local variation* of the function f at a vertex of the graph (It is a semi-norm for $p \geq 1$):

$$\|(\nabla_{\mathbf{w}}\mathbf{f})(\mathbf{v_i})\|_p = \Big[\sum_{v_j \sim v_i} w_{ij}^{p/2} \big|f(v_j) - f(v_i)\big|^p\Big]^{1/p} \quad . \quad (5)$$

Similarly, we have with $M^+ = \max$ and $M^- = \min$

$$(\nabla^{\pm}_{\mathbf{w}}\mathbf{f})(\mathbf{v_i}) = \big(\partial^{\pm}_{v_j} f(v_i)\big)^T_{(v_i, v_j) \in \mathcal{E}} \quad .$$

$$\|(\nabla^{\pm}_{\mathbf{w}}\mathbf{f})(\mathbf{v_i})\|_p = \Big[\sum_{v_j \sim v_i} w(v_i, v_j)^{p/2} \big|M^{\pm}\big(0, f(v_j) - f(v_i)\big)\big|^p\Big]^{1/p} \quad . \quad (6)$$

Isotropic p-Laplacian

The *weighted p-Laplace isotropic operator* of a function $f \in \mathcal{H}(\mathcal{V})$, noted $\Delta^i_{w,p} : \mathcal{H}(\mathcal{V}) \to \mathcal{H}(\mathcal{V})$, is defined by:

$$(\Delta^i_{w,p} f)(v_i) = \tfrac{1}{2} d^*_w (\|\mathbf{(\nabla_w f)(v_i)}\|_2^{p-2} (d_w f)(v_i, v_j)) \ . \tag{7}$$

The isotropic p-Laplace operator of $f \in \mathcal{H}(\mathcal{V})$, at a vertex $v_i \in \mathcal{V}$, can be computed by:

$$(\Delta^i_{w,p} f)(v_i) = \tfrac{1}{2} \sum_{v_j \sim v_i} (\gamma^i_{w,p} f)(v_i, v_j)(f(v_i) - f(v_j)) \ , \tag{8}$$

with

$$(\gamma^i_{w,p} f)(v_i, v_j) = w_{ij} \left(\|\mathbf{(\nabla_w f)(v_j)}\|_2^{p-2} + \|\mathbf{(\nabla_w f)(v_i)}\|_2^{p-2} \right) \ . \tag{9}$$

The p-Laplace isotropic operator is nonlinear, except for $p = 2$ (corresponds to the combinatorial Laplacian). For $p = 1$, it corresponds to the *weighted curvature* of the function f on the graph.

A. Elmoataz, O. Lezoray, S. Bougleux, Nonlocal Discrete Regularization on Weighted Graphs: a framework for Image and Manifold Processing, IEEE transactions on Image Processing, Vol. 17, n7, pp. 1047-1060, 2008.

Anisotropic p-Laplacian

The *weighted p-Laplace anisotropic operator* of a function $f \in \mathcal{H}(\mathcal{V})$, noted $\Delta^a_{w,p} : \mathcal{H}(\mathcal{V}) \rightarrow \mathcal{H}(\mathcal{V})$, is defined by:

$$(\Delta^a_{w,p} f)(v_i) = \frac{1}{2} d^*_w (|(d_w f)(v_i, v_j)|^{p-2} (d_w f)(v_i, v_j)) \ . \tag{10}$$

The anisotropic p-Laplace operator of $f \in \mathcal{H}(\mathcal{V})$, at a vertex $v_i \in \mathcal{V}$, can be computed by:

$$(\Delta^a_{w,p} f)(v_i) = \sum_{v_j \sim v_i} (\gamma^a_{w,p} f)(v_i, v_j)(f(v_i) - f(v_j)) \ . \tag{11}$$

with

$$(\gamma^a_{w,p} f)(v_i, v_j) = w_{ij}^{p/2} |f(v_i) - f(v_j)|^{p-2} \ . \tag{12}$$

O. Lezoray, V.T. Ta, A. Elmoataz, Partial differences as tools for filtering data on graphs, Pattern Recognition Letters, Vol. 31, n14, pp. 2201-2213, 2010.

Constructing graphs

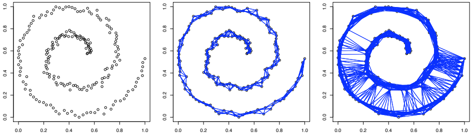
Any discrete domain can be modeled by a weighted graph where each data point is represented by a vertex $v_i \in \mathcal{V}$.

Unorganized data

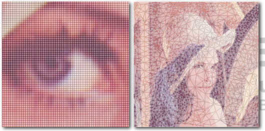
An unorganized set of points $\mathcal{V} \subset \mathbb{R}^n$ can be seen as a function $f^0 : \mathcal{V} \to \mathbb{R}^m$.
The set of edges is defined by in modeling the neighborhood of each vertex based on similarity relationships between feature vectors.
Typical graphs: k-nearest neighbors graphs and τ-neighborhood graphs.

Organized data

Typical cases of organized data are signals, gray-scale or color images (in 2D or 3D).
The set of edge is defined by spatial relationships.
Such data can be seen as functions $f^0 : \mathcal{V} \subset \mathbb{Z}^n \to \mathbb{R}^m$.
Typical graphs: pixel or region graphs.



(a) (b) k=3 (c) k=15



Weighting graphs

For an initial function $f^0 : \mathcal{V} \to \mathbb{R}^m$, similarity relationship between data can be incorporated within edges weights according to a measure of similarity $g : \ \mathcal{E} \to [0,1]$ with $w(e_{ij}) = g(e_{ij})$, $\forall e_{ij} \in \mathcal{E}$.
Each vertex v_i is associated with a feature vector $\mathbf{F}_{\boldsymbol{\tau}}^{\mathbf{f^0}} : \mathcal{V} \to \mathbb{R}^{m \times q}$ where q corresponds to this vector size:

$$\mathbf{F}_{\boldsymbol{\tau}}^{\mathbf{f^0}}(v_i) = \Big(f^0(v_j) : v_j \in \mathcal{N}_\tau(v_i) \cup \{v_i\} \Big)^T \tag{13}$$

with $\mathcal{N}_\tau(v_i) = \big\{ v_j \in \mathcal{V} \setminus \{v_i\} : \mu(v_i, v_j) \le \tau \big\}$.
For an edge e_{ij} and a distance measure $\rho : \mathbb{R}^{m \times q} \times \mathbb{R}^{m \times q} \to \mathbb{R}$ associated to $\mathbf{F}_{\boldsymbol{\tau}}^{\mathbf{f^0}}$, we can have:

$$\begin{aligned} g_1(e_{ij}) &= 1 \text{ (unweighted case) } , \\ g_2(e_{ij}) &= \exp\Big(-\rho\big(\mathbf{F}_{\boldsymbol{\tau}}^{\mathbf{f^0}}(v_i), \mathbf{F}_{\boldsymbol{\tau}}^{\mathbf{f^0}}(v_j) \big)^2 / \sigma^2 \Big) \text{ with } \sigma > 0 \ , \\ g_3(e_{ij}) &= 1 / \Big(1 + \rho\big(\mathbf{F}_{\boldsymbol{\tau}}^{\mathbf{f^0}}(v_i), \mathbf{F}_{\boldsymbol{\tau}}^{\mathbf{f^0}}(v_j) \big) \Big) \end{aligned} \tag{14}$$

Local to Non Local to Graphs

In Image Processing, we can divide methods according to three different models:

- **Local Processing**: usual model where local interactions around one pixel are taken into account (Vector Median Filter, Anisotropic Filtering, Wavelets, Total Variation minimization with PDE, etc.),

Local to Non Local to Graphs

In Image Processing, we can divide methods according to three different models:

- **Local Processing**: usual model where local interactions around one pixel are taken into account (Vector Median Filter, Anisotropic Filtering, Wavelets, Total Variation minimization with PDE, etc.),
- **Semi Local Processing**: one takes into account larger neighborhood interactions favored by the image geometry (Yaroslavsky and Bilateral filters),

Local to Non Local to Graphs

In Image Processing, we can divide methods according to three different models:

- **Local Processing**: usual model where local interactions around one pixel are taken into account (Vector Median Filter, Anisotropic Filtering, Wavelets, Total Variation minimization with PDE, etc.),
- **Semi Local Processing**: one takes into account larger neighborhood interactions favored by the image geometry (Yaroslavsky and Bilateral filters),
- **Non Local Processing**: model recently proposed by Buades and Morel which replaces spatial constraints by pixel blocks (i.e. patchs) constraints in a large neighborhood.

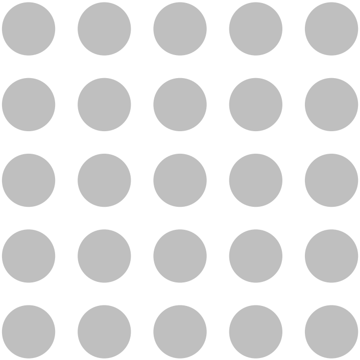
Local to Non Local to Graphs

In Image Processing, we can divide methods according to three different models:

- **Local Processing**: usual model where local interactions around one pixel are taken into account (Vector Median Filter, Anisotropic Filtering, Wavelets, Total Variation minimization with PDE, etc.),
- **Semi Local Processing**: one takes into account larger neighborhood interactions favored by the image geometry (Yaroslavsky and Bilateral filters),
- **Non Local Processing**: model recently proposed by Buades and Morel which replaces spatial constraints by pixel blocks (i.e. patchs) constraints in a large neighborhood.

Graph topology

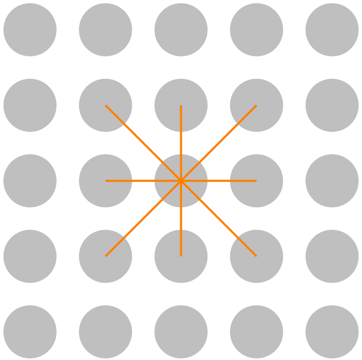
Digital Image



Graph topology

Digital Image

8-neighborhood : 3×3

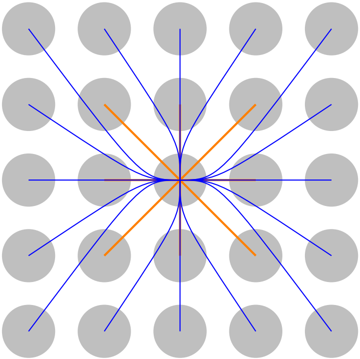


Graph topology

Digital Image

8-neighborhood : 3×3

24-neighborhood : 5×5



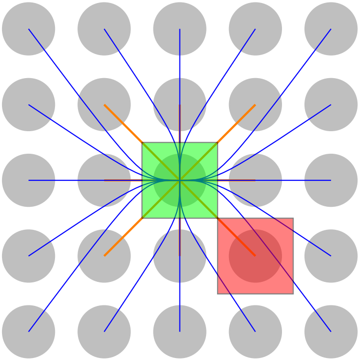
Graph topology

Digital Image

8-neighborhood : 3×3

24-neighborhood : 5×5

Local: a value is associated to vertices

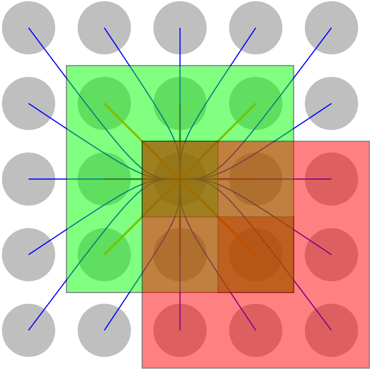


Digital Image

8-neighborhood : 3×3

24-neighborhood : 5×5

Nonlocal: a patch (vector of values in a given neighborhood) is associated to vertices.



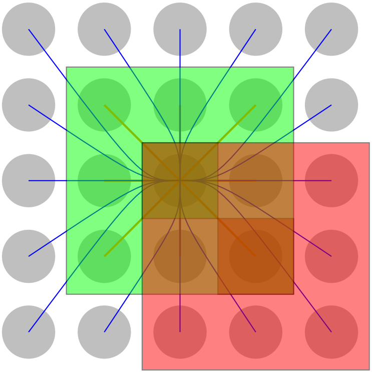
Graph topology

Digital Image

8-neighborhood : 3×3

24-neighborhood : 5×5

Nonlocal: a patch (vector of values in a given neighborhood) is associated to vertices.



With Graphs

Nonlocal behavior is directly expressed by the graph topology.
Patches are used to measure similarity between vertices.

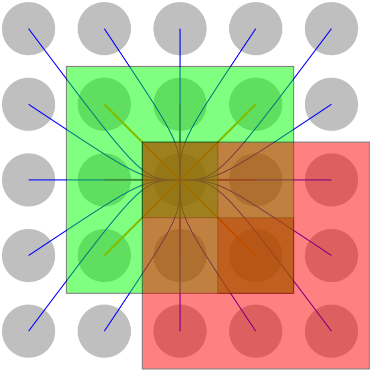
Graph topology

Digital Image

8-neighborhood : 3×3

24-neighborhood : 5×5

Nonlocal: a patch (vector of values in a given neighborhood) is associated to vertices.



Consequences

- Nonlocal processing of images becomes local processing on similarity graphs.
- Our difference operators on graphs naturally enable local and nonlocal configurations (with the weight function and the graph topology)

p-Laplacian nonlocal regularization on graphs

Let $f^0 : \mathcal{V} \to \mathbb{R}$ be a given (noisy) function defined on the vertices of a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$.
f^0 represents an observation of a clean function $g : \mathcal{V} \to \mathbb{R}$ corrupted by a given noise n such that $f^0 = g + n$.
Recovering the uncorrupted function g is an inverse problem: a commonly used method is to seek for a function $f : \mathcal{V} \to \mathbb{R}$ which is regular enough on $\mathcal{G}$, and also close enough to f^0.
We consider the following variational problem:

$$g \approx \min_{f:\mathcal{V}\to\mathbb{R}} \left\{ E^*_{w,p}(f, f^0, \lambda) = R^*_{w,p}(f) + \frac{\lambda}{2} \|f - f^0\|_2^2 \right\}, \tag{15}$$

where the regularization functional $R^*_{w,p} : \mathcal{H}(\mathcal{V}) \to \mathbb{R}$ can correspond to an isotropic $R^i_{w,p}$ or an anisotropic $R^a_{w,p}$ functionnal.

A. Elmoataz, O. Lezoray, S. Bougleux, Nonlocal Discrete Regularization on Weighted Graphs: a framework for Image and Manifold Processing, IEEE transactions on Image Processing, Vol. 17, n7, pp. 1047-1060, 2008.

Isotropic an anisotropic regularization terms

The isotropic regularization functionnal $R^i_{w,p}$ is defined by the $\mathcal{L}_2$ norm of the gradient and is the discrete p-Dirichlet form of the function $f \in \mathcal{H}(\mathcal{V})$:

$$\begin{aligned} R^i_{w,p}(f) &= \frac{1}{p} \sum_{v_i \in \mathcal{V}} \|(\nabla_{\mathbf{w}} \mathbf{f})(\mathbf{v_i})\|_2^p = \frac{1}{p} \langle f, \Delta^i_{w,p} f \rangle_{\mathcal{H}(\mathcal{V})} \\ &= \frac{1}{p} \sum_{v_i \in \mathcal{V}} \left[\sum_{v_j \sim v_i} w_{ij} (f(v_j) - f(v_i))^2 \right]^{\frac{p}{2}} . \end{aligned} \tag{16}$$

The anisotropic regularization functionnal $R^a_{w,p}$ is defined by the $\mathcal{L}_p$ norm of the gradient:

$$\begin{aligned} R^a_{w,p}(f) &= \frac{1}{p} \sum_{v_i \in \mathcal{V}} \|(\nabla_{\mathbf{w}} \mathbf{f})(\mathbf{v_i})\|_p^p = \frac{1}{p} \langle f, \Delta^a_{w,p} f \rangle_{\mathcal{H}(\mathcal{V})} \\ &= \frac{1}{p} \sum_{v_i \in \mathcal{V}} \sum_{v_j \sim v_i} w_{ij}^{p/2} |f(v_j) - f(v_i)|^p . \end{aligned} \tag{17}$$

When $p \geq 1$, the energy $E^*_{w,p}$ is a convex functional of functions of $\mathcal{H}(\mathcal{V})$.

Isotropic diffusion process

To get the solution of the minimizer, we consider the following system of equations:

$$\frac{\partial E^i_{w,p}(f,f^0,\lambda)}{\partial f(v_i)} = 0, \forall v_i \in \mathcal{V} \tag{18}$$

which is rewritten as:

$$\frac{\partial R^i_{w,p}(f)}{\partial f(v_i)} + \lambda(f(v_i) - f^0(v_i)) = 0, \quad \forall v_i \in \mathcal{V}. \tag{19}$$

Moreover, we can prove that

$$\frac{\partial R^i_{w,p}(f)}{\partial f(v_i)} = 2(\Delta^i_{w,p} f)(v_i) \ . \tag{20}$$

The system of equations is then rewritten as

$$2(\Delta^i_{w,p} f)(v_i) + \lambda(f(v_i) - f^0(v_i)) = 0, \quad \forall v_i \in \mathcal{V}, \tag{21}$$

which is equivalent to the following system of equations:

$$\left(\lambda + \sum_{v_j \sim v_i} (\gamma^i_{w,p} f)(v_i, v_j)\right) f(v_i) - \sum_{v_j \sim v_i} (\gamma^i_{w,p} f)(v_i, v_j) f(v_j) = \lambda f^0(v_i). \tag{22}$$

Isotropic diffusion process

We use the linearized Gauss-Jacobi iterative method to solve the previous system. Let n be an iteration step, and let $f^{(n)}$ be the solution at the step n. Then, the method is given by the following algorithm:

$$\begin{cases} f^{(0)} = f^0 \\ f^{(n+1)}(v_i) = \dfrac{\lambda f^0(v_i) + \sum_{v_j \sim v_i} (\gamma^i_{w,p} f^{(n)})(v_i, v_j) f^{(n)}(v_j)}{\lambda + \sum_{v_j \sim v_i} (\gamma^i_{w,p} f^{(n)})(v_i, v_j)}, \forall v_i \in \mathcal{V}. \end{cases} \tag{23}$$

with

$$(\gamma^i_{w,p} f)(v_i, v_j) = w_{ij} \left(\|(\nabla_{\mathbf{w}} \mathbf{f})(\mathbf{v_j})\|_2^{p-2} + \|(\nabla_{\mathbf{w}} \mathbf{f})(\mathbf{v_i})\|_2^{p-2} \right) . \tag{24}$$

It describes a family of discrete diffusion processes, which is parameterized by the structure of the graph (topology and weight function), the parameter p, and the parameter λ.

λ	w	Graph	$p = 1$	$p = 2$	$p \in]0,1[$
0	$exp()$	semi-local	Our	Bilateral	Our
0	$exp()$	nonlocal	Our	NLMeans	Our
$\neq 0$	constant	local	TV Digital	L_2 Digital	Our
$\neq 0$	any	nonlocal	Our	Our	Our

Table: Works related to our framework in image processing.

Original image

Noisy image (Gaussian noise with $\sigma = 15$)

$f^0 : \mathcal{V} \to \mathbb{R}^3$ PSNR=29.38dB

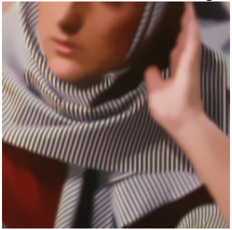
Examples: Image denoising

Isotropic $\mathcal{G}_1$, $\mathbf{F}_{\mathbf{0}}^{\mathbf{f^0}} = f^0$ | Isotropic $\mathcal{G}_7$, $\mathbf{F}_{\mathbf{3}}^{\mathbf{f^0}}$ | Anisotropic $\mathcal{G}_7$, $\mathbf{F}_{\mathbf{3}}^{\mathbf{f^0}}$

$p = 2$

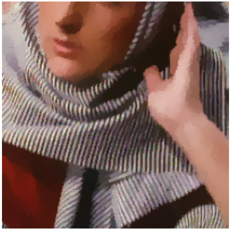
PSNR=28.52db

PSNR=31.79dB



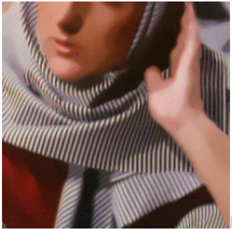
PSNR=31.79dB

$p = 1$



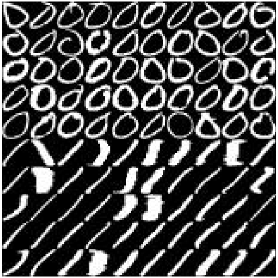
PSNR=31.25dB

PSNR=34.74dB



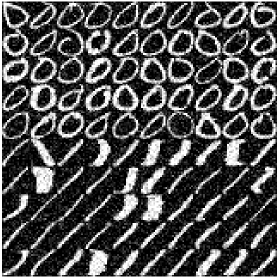
PSNR=31.81dB

Examples: Image Database denoising

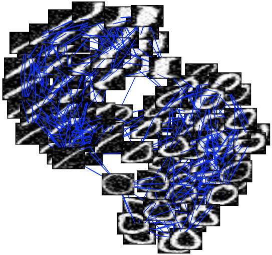


Initial data

$f^0 : \mathcal{V} \rightarrow \mathbb{R}^{16\times16}$



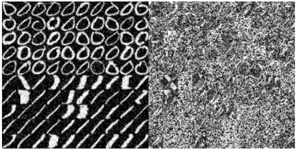
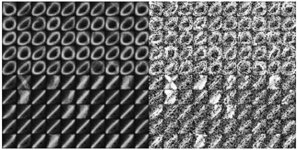
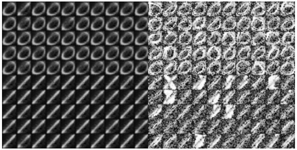
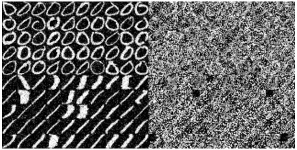
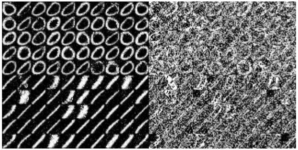
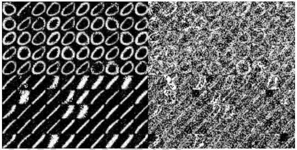
Noisy data



10-NNG

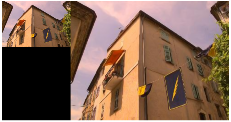
université de Caen Basse-Normandie

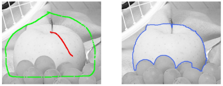
Examples: Image Database denoising

	$\lambda = 1$	$\lambda = 0.01$	$\lambda = 0$
		Isotropic	
$p = 1$			
	PSNR=18.80dB	PSNR=13.54dB	PSNR=10.52dB
		Anisotropic	
$p = 1$			
	PSNR=18.96dB	PSNR=15.19dB	PSNR=14.41dB

Interpolation of missing data on graphs

Let $f^0 : \mathcal{V}_0 \to \mathbb{R}$ be a function with $\mathcal{V}_0 \subset \mathcal{V}$ be the subset of vertices from the whole graph with known values.





The interpolation consists in recovering values of f for the vertices of $\mathcal{V} \setminus \mathcal{V}_0$ given values for vertices of $\mathcal{V}_0$ formulated by:

$$\min_{f:\mathcal{V}\to\mathbb{R}} R^*_{w,p}(f) + \lambda(v_i)\|f(v_i) - f^0(v_i)\|_2^2 \ . \tag{25}$$

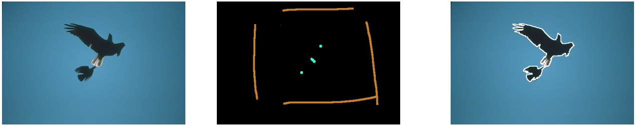
Since $f^0(v_i)$ is known only for vertices of $\mathcal{V}_0$, the Lagrange parameter is defined as $\lambda : \mathcal{V} \to \mathbb{R}$:

$$\lambda(v_i) = \begin{cases} \lambda & \text{if } v_i \in \mathcal{V}_0 \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

This comes to consider $\Delta^*_{w,p} f(v_i) = 0$ on $\mathcal{V} \setminus \mathcal{V}_0$.
Our isotropic and anisotropic diffusion processes can be directly used to perform the interpolation.

Examples: Image segmentation

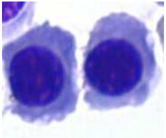
Original Image | User label input | Segmentation result



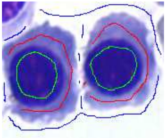
$\mathcal{G}_0$, $\mathbf{F_0^{f^0}} = f^0$, $w = g_2$, $p = 2$, $\lambda = 1$

$\mathcal{G}_0 \cup$ 4-NNG$_3$, $\mathbf{F_3^{f^0}}$, $w = g_2$, $p = 2$, $\lambda = 1$

Examples: Image segmentation

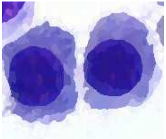


(a) 27 512 pixels

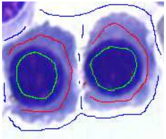


(b) Original+Labels

(c) $t = 50$ (11 seconds)



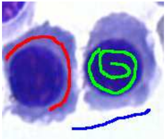
(d) 639 zones (98% of reduction)



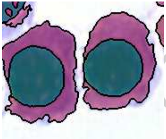
(e) Original+Labels

(f) $t = 5$ (< 1 second)

(g) 639 zones (98% of reduc-

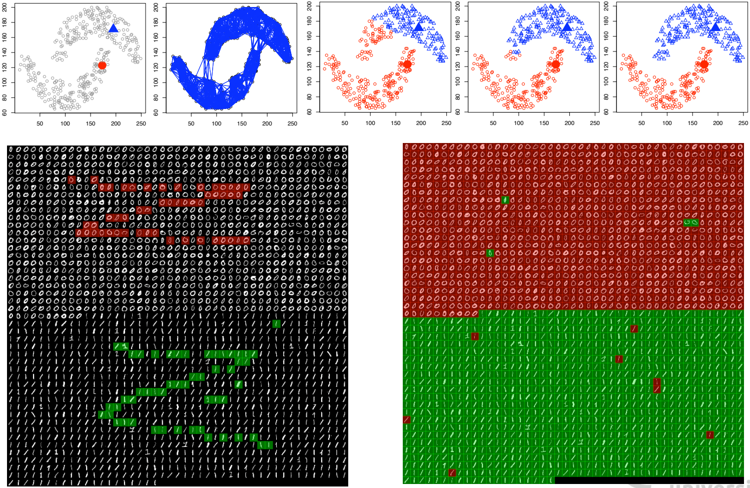


(h) Original+Labels



(i) $t = 2$ (< 1 second)

Examples: Data base clustering



Examples: Image colorization

Gray level image

Color scribbles

Compute Weights from the gray-level image, interpolation is performed in a chrominance color space from the seeds: $\mathbf{f^c(v_i)} = \left[\frac{f_1^s(v_i)}{f^l(v_i)}, \frac{f_2^s(v_i)}{f^l(v_i)}, \frac{f_3^s(v_i)}{f^l(v_i)}\right]^T$

O. Lezoray, A. Elmoataz, V.T. Ta, Nonlocal graph regularization for image colorization, International Conference on Pattern Recognition (ICPR), 2008.

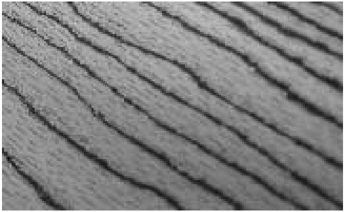
Examples: Image colorization

$p = 1,\ \mathcal{G}_1,\ \mathbf{F}_{\mathbf{0}}^{\mathbf{f^0}} = f^0$

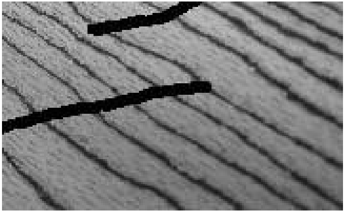
$p = 1,\ \mathcal{G}_5,\ \mathbf{F}_{\mathbf{2}}^{\mathbf{f^0}}$

université de Caen
Basse-Normandie

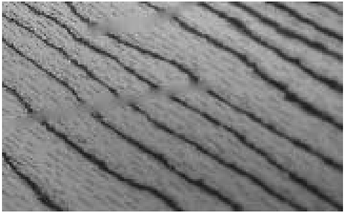
Examples: image inpainting



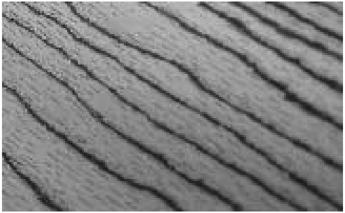
Original image



Damaged image to inpaint



$\mathcal{G}_1$, $\mathbf{F}_{\mathbf{0}}^{\mathbf{f}^0} = f^0$



$\mathcal{G}_{15}$, $\mathbf{F}_{\mathbf{6}}^{\mathbf{f}^0}$

Using our nonlocal interpolation regularization-based functional unifies geometric and texture based techniques: geometric aspect is expressed by graph topology and texture by graph weights.

Nonlinear scale-space approaches based on Mathematical Morphology (MM) operators are one of the most important tools in image processing.

The two fundamental operators in Mathematical Morphology are dilation and erosion.

Dilation δ of a function $f^0 : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ consists in replacing the function value by the maximum value within a structuring element B such that:

$$\delta_B\big(f^0(x,y)\big) = \max\big\{f^0(x+x', y+y') | (x',y') \in B\big\}$$

Erosion ϵ is computed by:

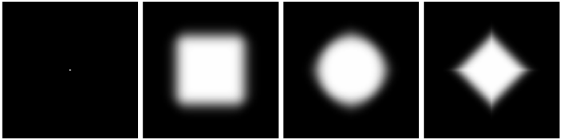
$$\epsilon_B\big(f^0(x,y)\big) = \min\big\{f^0(x+x', y+y') | (x',y') \in B\big\}$$

For convex structuring elements, an alternative formulation in terms of Partial Differential Equations (PDE) has also been proposed.
Given an initial function $f^0 : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, a disc $B = \{z \in \mathbb{R}^2 : \|z\|_p \leq 1\}$, one considers the following evolution equation

$$\frac{\partial f}{\partial t} = \partial_t f = \pm \|\mathbf{\nabla f}\|_p$$

Solution of $f(x, y, t)$ at time $t > 0$ provides dilation (with the plus sign) or erosion (with the minus sign) within a structuring element of size $n\Delta t$:

$$\delta(f) = \partial_t f = +\|\mathbf{\nabla f}\|_p \quad \text{and} \quad \epsilon(f) = \partial_t f = -\|\mathbf{\nabla f}\|_p$$



with a size of $100\Delta t$, $\Delta t = 0.25$ and $p = 1$, $p = 2$, and $p = \infty$.

Adaptive mathematical morphology on graphs

Our proposal

- Transcription of PDE MM on arbitrary graphs
- Introduction of nonlocal schemes for images
- Extend MM to the processing of arbitrary data (point clouds, databases, etc.)

V.T. Ta, A. Elmoataz, O. Lézoray, Nonlocal PDEs-based Morphology on Weighted Graphs for Image and Data Processing, IEEE transactions on Image Processing, 2011. to appear

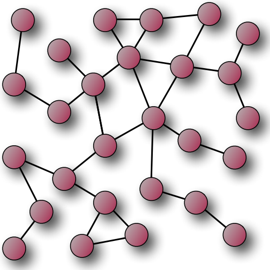
V.T. Ta, A. Elmoataz, O. Lézoray, Nonlocal Graph Morphology, International Symposium on Mathematical Morphology - Abstract Book, pp. 5-9, 2009.

V.T. Ta, A. Elmoataz, O. Lézoray, Partial difference equations on graphs for mathematical morphology operators over images and manifolds, International Conference on Image Processing (IEEE), pp. 801-804, 2008. **Winner of the IBM Student-Paper Award**.

V.T. Ta, A. Elmoataz, O. Lézoray, Partial Difference Equations over Graphs: Morphological Processing of Arbitrary Discrete Data, European Conference on Computer Vision, Vol. LNCS 5304, pp. 668-680, 2008.

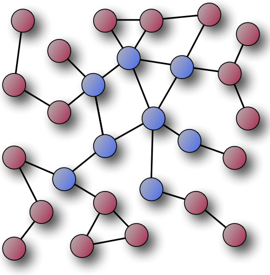
Transcription on graphs

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, $f^0 : \mathcal{V} \to \mathbb{R}$, $f(.,0) = f$, $\forall v_i \in \mathcal{V}$, we define:
$\delta : \partial_t f(v_i, t) = +\|(\boldsymbol{\nabla}^+_{\mathbf{w}}\mathbf{f})(\mathbf{v_i}, \mathbf{t})\|_p$ $\quad\epsilon : \partial_t f(v_i, t) = -\|(\boldsymbol{\nabla}^-_{\mathbf{w}}\mathbf{f})(\mathbf{v_i}, \mathbf{t})\|_p$



Transcription on graphs

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, $f^0 : \mathcal{V} \to \mathbb{R}$, $f(., 0) = f$, $\forall v_i \in \mathcal{V}$, we define:
$\delta : \partial_t f(v_i, t) = +\|(\nabla_w^+ \mathbf{f})(\mathbf{v_i}, \mathbf{t})\|_p$ $\quad\epsilon : \partial_t f(v_i, t) = -\|(\nabla_w^- \mathbf{f})(\mathbf{v_i}, \mathbf{t})\|_p$

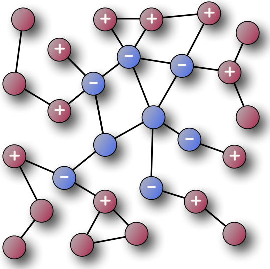


$\mathcal{A} \subset \mathcal{V}$

Transcription on graphs

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, $f^0 : \mathcal{V} \to \mathbb{R}$, $f(.,0) = f$, $\forall v_i \in \mathcal{V}$, we define:

$\delta : \partial_t f(v_i, t) = +\|(\nabla_w^+ \mathbf{f})(\mathbf{v_i}, \mathbf{t})\|_p \qquad \epsilon : \partial_t f(v_i, t) = -\|(\nabla_w^- \mathbf{f})(\mathbf{v_i}, \mathbf{t})\|_p$



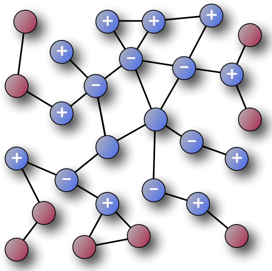
$\mathcal{A} \subset \mathcal{V}$

$\partial^+ \mathcal{A} = \{v_i \notin \mathcal{A} : \exists v_j \in \mathcal{A} \text{ with } e_{ij} \in \mathcal{E}\}$

$\partial^- \mathcal{A} = \{v_i \in \mathcal{A} : \exists v_j \notin \mathcal{A} \text{ with } e_{ij} \in \mathcal{E}\}$

Transcription on graphs

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, $f^0 : \mathcal{V} \to \mathbb{R}$, $f(.,0) = f$, $\forall v_i \in \mathcal{V}$, we define:
$\delta : \partial_t f(v_i, t) = +\|(\nabla^+_{\mathbf{w}} \mathbf{f})(\mathbf{v_i}, \mathbf{t})\|_p$ $\quad$ $\epsilon : \partial_t f(v_i, t) = -\|(\nabla^-_{\mathbf{w}} \mathbf{f})(\mathbf{v_i}, \mathbf{t})\|_p$



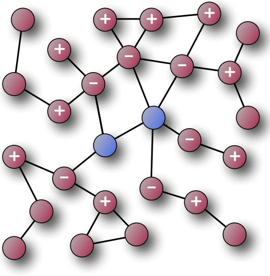
$\mathcal{A} \subset \mathcal{V}$
$\partial^+ \mathcal{A} = \{v_i \notin \mathcal{A} : \exists v_j \in \mathcal{A}$ with $e_{ij} \in \mathcal{E}\}$
$\partial^- \mathcal{A} = \{v_i \in \mathcal{A} : \exists v_j \notin \mathcal{A}$ with $e_{ij} \in \mathcal{E}\}$
Dilation: adding vertices from $\partial^+ \mathcal{A}$ to $\mathcal{A}$

université de Caen Basse-Normandie

Transcription on graphs

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, $f^0 : \mathcal{V} \to \mathbb{R}$, $f(.,0) = f$, $\forall v_i \in \mathcal{V}$, we define:

$\delta : \partial_t f(v_i, t) = +\|(\nabla_w^+ f)(v_i, t)\|_p$ $\quad\quad$ $\epsilon : \partial_t f(v_i, t) = -\|(\nabla_w^- f)(v_i, t)\|_p$



$\mathcal{A} \subset \mathcal{V}$

$\partial^+\mathcal{A} = \{v_i \notin \mathcal{A} : \exists v_j \in \mathcal{A} \text{ with } e_{ij} \in \mathcal{E}\}$

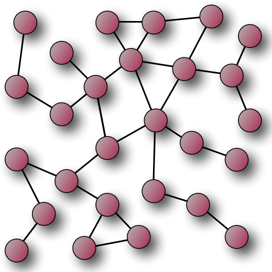
$\partial^-\mathcal{A} = \{v_i \in \mathcal{A} : \exists v_j \notin \mathcal{A} \text{ with } e_{ij} \in \mathcal{E}\}$

Dilation: adding vertices from $\partial^+\mathcal{A}$ to $\mathcal{A}$

Erosion: removing vertices from $\partial^-\mathcal{A}$ to $\mathcal{A}$

Transcription on graphs

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, $f^0 : \mathcal{V} \to \mathbb{R}$, $f(.,0) = f$, $\forall v_i \in \mathcal{V}$, we define:
$\delta : \partial_t f(v_i, t) = +\|(\boldsymbol{\nabla}_{\mathbf{w}}^{+}\mathbf{f})(\mathbf{v_i}, \mathbf{t})\|_p$ $\quad$ $\epsilon : \partial_t f(v_i, t) = -\|(\boldsymbol{\nabla}_{\mathbf{w}}^{-}\mathbf{f})(\mathbf{v_i}, \mathbf{t})\|_p$



$\mathcal{A} \subset \mathcal{V}$
$\partial^+ \mathcal{A} = \{v_i \notin \mathcal{A} : \exists v_j \in \mathcal{A} \text{ with } e_{ij} \in \mathcal{E}\}$
$\partial^- \mathcal{A} = \{v_i \in \mathcal{A} : \exists v_j \notin \mathcal{A} \text{ with } e_{ij} \in \mathcal{E}\}$
Dilation: adding vertices from $\partial^+ \mathcal{A}$ to $\mathcal{A}$
Erosion: removing vertices from $\partial^- \mathcal{A}$ to $\mathcal{A}$

Dilation: maximizing a surface gain proportionally to $\|(\boldsymbol{\nabla}_{\mathbf{w}}^{+}\mathbf{f})(\mathbf{v_i})\|_p$
Erosion: minimizing a surface gain proportionally to $\|(\boldsymbol{\nabla}_{\mathbf{w}}^{-}\mathbf{f})(\mathbf{v_i})\|_p$

PDE MM: $\delta : \partial_t f(x,t) = +\|\mathbf{\nabla f(x,t)}\|_p \qquad \epsilon : \partial_t f(x,t) = -\|\mathbf{\nabla f(x,t)}\|_p$

Transcription on graphs

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, $f^0 : \mathcal{V} \to \mathbb{R}$, $f(.,0) = f$, $\forall v_i \in \mathcal{V}$, we define:
$\delta : \partial_t f(v_i,t) = +\|(\mathbf{\nabla^+_w f})(\mathbf{v_i, t})\|_p \qquad \epsilon : \partial_t f(v_i,t) = -\|(\mathbf{\nabla^-_w f})(\mathbf{v_i, t})\|_p$
Since we can prove that for any level f^l of f, we have:

$$\|(\mathbf{\nabla_w f^l})(\mathbf{v_i})\|_p = \begin{cases} \|(\mathbf{\nabla^+_w f^l})(\mathbf{v_i})\|_p \text{ if } v_i \in \partial^+\mathcal{A}^l, \\ \|(\mathbf{\nabla^-_w f^l})(\mathbf{v_i})\|_p \text{ if } v_i \in \partial^-\mathcal{A}^l. \end{cases} \qquad (27)$$

$\mathcal{L}_p$ norm:

$$\|(\mathbf{\nabla^\pm_w f})(\mathbf{v_i})\|_p = \left[\sum_{v_j \sim v_i} w(v_i,v_j)^{p/2} \left| M^\pm\big(0, f(v_j) - f(v_i)\big)\right|^p\right]^{1/p}, \ 0 < p < \infty$$

$\mathcal{L}_\infty$ norm:

$$\|(\mathbf{\nabla^\pm_w f})(\mathbf{v_i})\|_\infty = \max_{v_j \sim v_i}\left(w(v_i,v_j)^{1/2} \left| M^\pm\big(0, f(v_j) - f(v_i)\big)\right|\right)$$

with $M^+ = \max$ and $M^- = \min$

Numerical resolution

Iterative algorithms with discretization in time: $f^0 : \mathcal{V} \to \mathbb{R}$, $f^{(n)}(v_i) \approx f(v_i, n\Delta t)$

$$\begin{cases} f^{(n+1)}(v_i) = f^{(n)}(v_i) \pm \Delta t \|(\nabla_{\mathbf{w}}^{\pm} \mathbf{f}^{(\mathbf{n})})(\mathbf{v_i})\|_p \\ f^0(v_i) = f^0(v_i) \end{cases}$$

$\mathcal{L}_p$ norm:

$$f^{(n+1)}(v_i) = f^{(n)}(v_i) \pm \Delta t \left[\sum_{v_j \sim v_i} w(v_i, v_j)^{p/2} \left| M^{\pm}\left(0, f(v_j) - f(v_i)\right) \right|^p \right]^{1/p}$$

$\mathcal{L}_\infty$ norm:

$$f^{(n+1)}(v_i) = f^{(n)}(v_i) \pm \Delta t \max_{v_j \sim v_i} \left(w(v_i, v_j)^{1/2} \left| M^{\pm}\left(0, f(v_j) - f(v_i)\right) \right| \right)$$

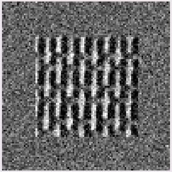
For $p = 2$ and $w = 1$ on a grid, we recover the PDE numerical scheme of Osher & Sethian.
For $p = \infty$, $\Delta t = 1$, and $w = 1$, we recover the algebraic formulation with the structuring element expressed by the graph topology.

Basse-Normandie

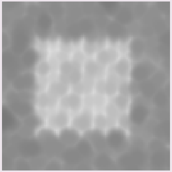
Why **Adaptive** mathematical morphology ?

Varying w and graph topology, we obtain adaptivity.

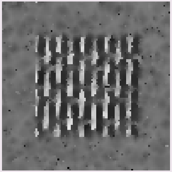
Adaptivity with graph weights: example of a closing $\phi(f) = \epsilon(\delta(f))$



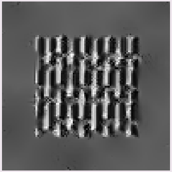
Initial



Unweighted



Weighted

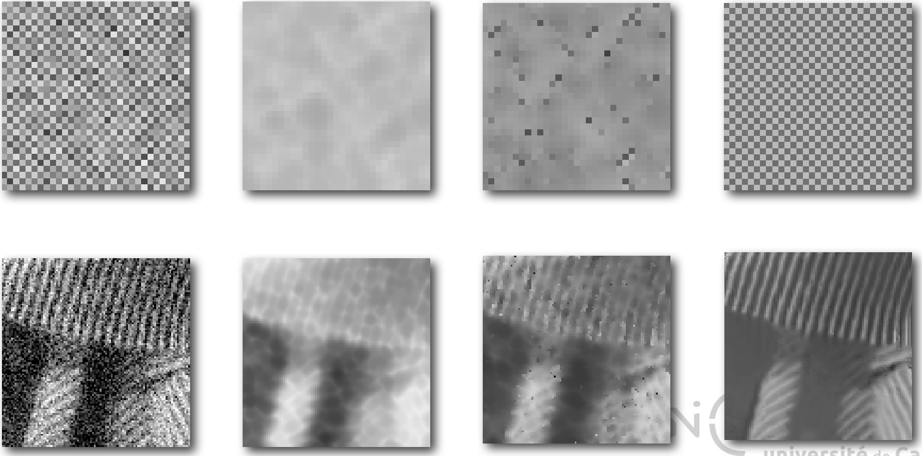


Non local with patchs

université de Caen Basse-Normandie

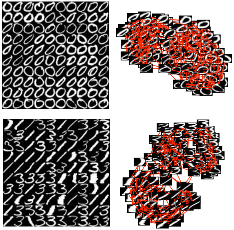
Examples: closing $\phi(f) = \epsilon(\delta(f))$

Initial | Local | Weighted | Non local / patchs



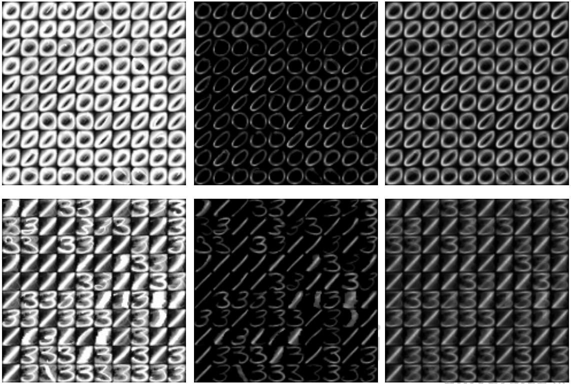
Examples: image databases

Initial k-NNG



$f^0 : V \rightarrow \mathbb{R}^{256}$

Dilation Erosion Opening



Eikonal equation

Our proposal

- Transcription of the Eikonal equation on graphs
- Introduction of nonlocal schemes for images
- Applications to any data and graphs

V.T. Ta, A. Elmoataz, O. Lézoray, Adaptation of Eikonal Equation over Weighted Graphs, International Conference on Scale Space Methods and Variational Methods in Computer Vision (SSVM), Vol. LNCS 5567, pp. 187-199, 2009.

X. Desquesnes, A. Elmoataz, O. Lézoray, V.T. Ta, Efficient Algorithms for Image and High Dimensional Data Processing using Eikonal Equation on Graphs, International Symposium on Visual Computing, Vol. LNCS 6454, pp. 647-658, 2010.

Adaptation of the Eikonal equation on graphs

We consider the time marching approach of the Eikonal equation:

$$\begin{cases} \partial_t f(x,t) = P(x) - \|\boldsymbol{\nabla f(x,t)}\|_2 & x \in \Omega \backslash \Gamma \\ f(x,t) = \phi(x) & x \in \Gamma \\ f(.,0) = \phi_0(.) & x \in \Omega \end{cases}$$

At $t \to \infty$, the solution satisfies the Eikonal equation.

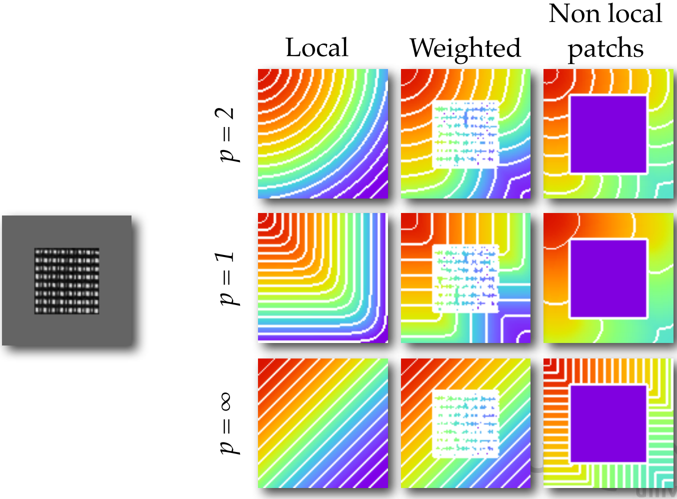
Transcription on arbitrary graphs

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ and source vertices $\mathcal{V}_0$,

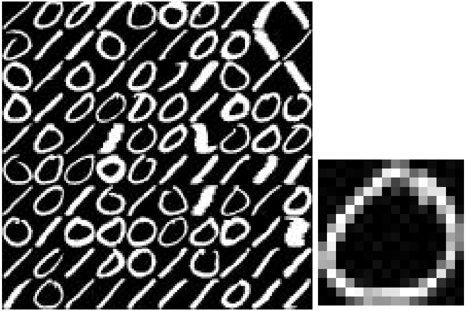
$$\begin{cases} \partial_t f(v_i,t) = P(v_i) - \|(\boldsymbol{\nabla_w^- f})(\mathbf{v_i})\|_p & v_i \in \mathcal{V} \backslash \mathcal{V}_0 \\ f(v_i,t) = \phi(v_i) & v_i \in \mathcal{V}_0 \\ f(v_i,0) = \phi_0(v_i) & v_i \in \mathcal{V} \end{cases}$$

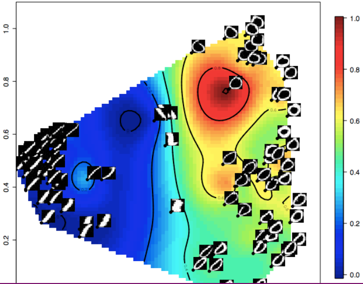
solved by $f^{(n+1)}(v_i) = f^{(n)}(v_i) - \Delta t \left(\|(\boldsymbol{\nabla_w^- f^n})(\mathbf{v_i})\|_p - P(v_i) \right)$.

- With $p = 2$, 4-grid, and $w = 1$: Osher-Sethian upwind first order Hamiltonian discretization scheme
- With $\Delta t = 1$ and $\mathcal{L}_\infty$: shortest path on a graph.
- With other p and w values: a difference equation with adaptive coefficients.

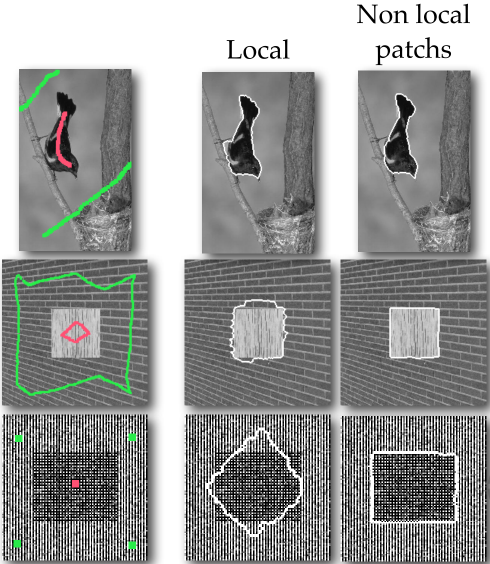

Local
Weighted
Non local
patchs
$p = 2$
$p = 1$
$p = \infty$

Examples: distances computation on databases

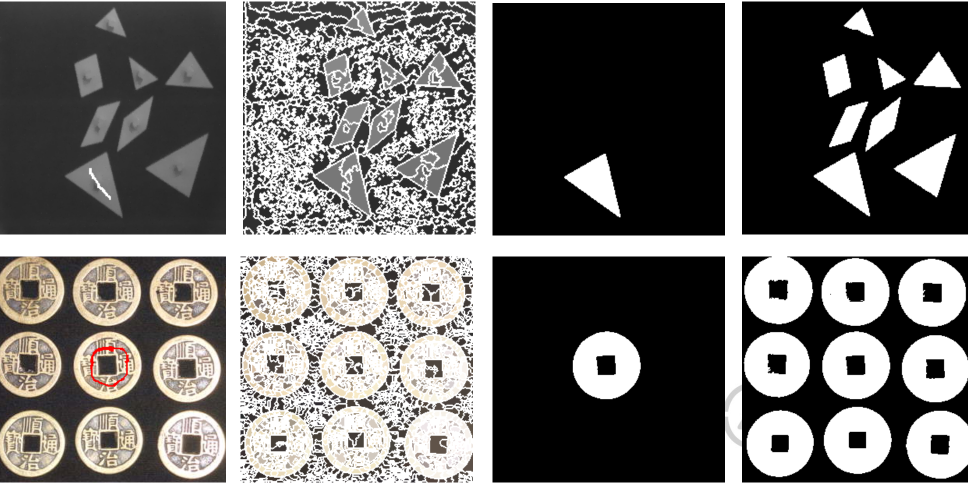




Examples: segmentation of images



Partitions RAG RAG + NNG



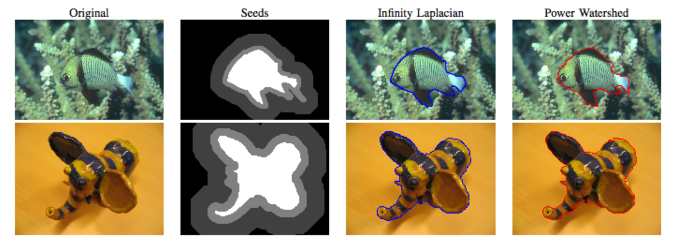
Conclusions

- PdE provide a framework for the transcription of PDE on graphs
- Recovers many approaches in literature
- Extends them to the processing of arbitrary data on graphs
- Naturally enables local and nonlocal processing

Next Works

Study the limit as p tend to ∞ of minimizers of p harmonic functions on graphs.
Discrete nonlocal ∞-Laplacian equation for interpolation:

$$\Delta_{w,\infty} f(u) = \|(\nabla_w^- f)(u)\|_\infty - \|(\nabla_w^+ f)(u)\|_\infty = 0$$



The End. Thanks.

Publications available at :
`http://www.info.unicaen.fr/~lezoray`

université de Caen
Basse-Normandie