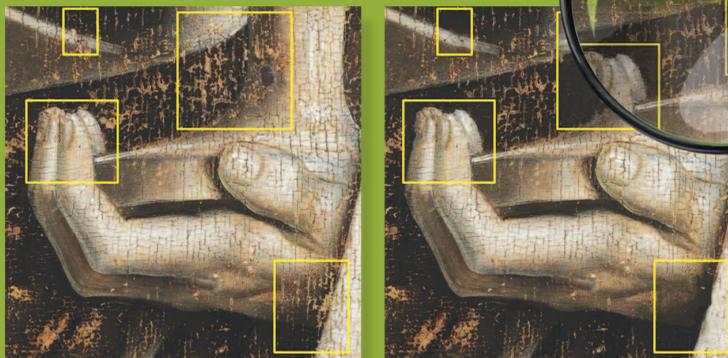


PDE-Based Graph Signal Processing for 3-D Color Point Clouds

© GHEENT KATHEDRALE KERKFABRIEK; PHOTO COURTESY OF KIK-IRPA, BRUSSELS. MAGNIFYING GLASS—IMAGE LICENSED BY INGRAM PUBLISHING.



Signal Processing for Art Investigation

[Opportunities for cultural heritage]

With the advance of three-dimensional (3-D) scanning technology, the cultural heritage community is increasingly interested in 3-D scans of cultural objects such as antiques, artifacts, and heritage sites. Digitization of these objects is commonly aimed at heritage preservation. Since 3-D color scanning has the potential to tackle a variety of traditional documentation challenges, the use of signal processing techniques on such data can be expected to yield new applications

that are feasible for the first time with the aid of captured 3-D color point clouds. Our contributions are twofold. First, we propose a simple method to solve partial differential equations (PDEs) on point clouds using the framework of partial difference equations (PdEs) on graphs. Second, we survey several applications of 3-D color point cloud processing on real examples for which signal processing researchers can develop tools that can be valuable for museum specialists. The results of these methods have been reviewed by experts in the arts and found promising.

INTRODUCTION

Historians, archaeologists, and museum curators are interested in the preservation of pieces of art and want to make them available

to as wide an audience as possible. While direct encounters with the material pieces of art will always have an essential value for museum specialists, one important concern for museum curators is the preparation of catalogs for the purpose of object identification and examination. Traditionally, museum specialists have used two-dimensional (2-D) imaging techniques such as hand drawings, photographs, or document scans; see [1] and [2]. Unfortunately, these standard formats have the limitation of being selective and insufficient to record nuanced information about the complete shape, color, and texture of the object. In addition, some objects might undergo important

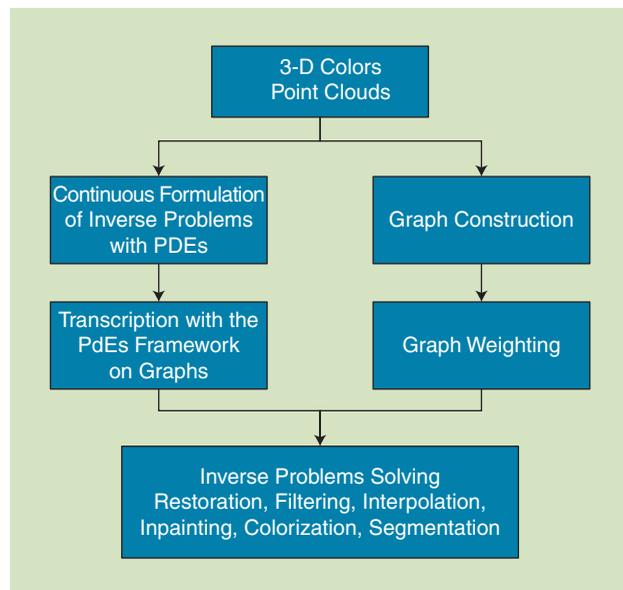
WITH THE ADVANCE OF 3-D SCANNING TECHNOLOGY, THE CULTURAL HERITAGE COMMUNITY IS INCREASINGLY INTERESTED IN 3-D SCANS OF CULTURAL OBJECTS SUCH AS ANTIQUES, ARTIFACTS, AND HERITAGE SITES.

changes over time, and in this case, the preservation of the object state (at a given temporal snapshot) is important. These issues can now be addressed by using 3-D color scanning, which has the ability to record the complete object in great detail. The production of digital 3-D copies of pieces of art has therefore gained considerable attention from the cultural heritage community [3].

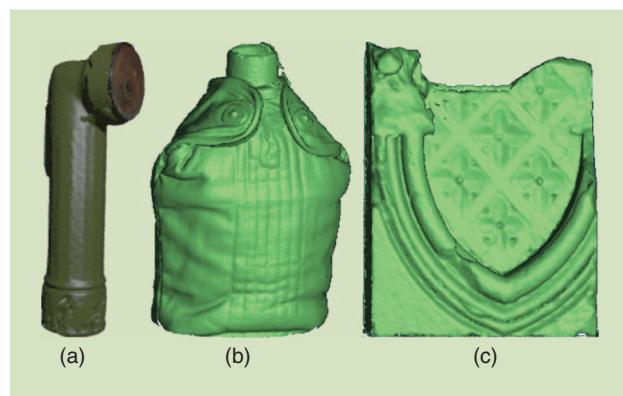
The recent proliferation of commercial 3-D digital scanning devices has led to the establishment of 3-D scanning as a practical reality in the field of cultural heritage preservation [4]. The interest is evident: it provides a high-precision digital reference of a cultural object and makes possible mass distribution and consultation.

Most traditional signal processing methods in art investigation, such as filtering, PDE-based processing, and wavelets, are designed for data defined on regular Euclidean grids; see [2] and [5]. With 3-D color point clouds, the data takes the form of unstructured raw point samples without any attached geometry. Traditional signal processing tools cannot be directly applied on raw 3-D color point clouds since there is no structuring information. In addition, it is not desirable to transform point clouds to meshes since this requires sampling points with a loss of accuracy and this is not compatible with the goal of high-fidelity conservation. Therefore it is essential to investigate a framework for the adaptation of signal processing tools for 3-D point cloud processing.

Our contributions are twofold. First, we propose a simple method for adapting and solving PDEs on point clouds. This method relies on the framework of PdEs on weighted graphs [6] as shown in Figure 1. Second, we survey several applications in 3-D color point cloud processing where signal processing tools can be applied to images of interest to art and museum specialists. From a signal processing point of view, most of these problems can be formulated as inverse problems for graph signals [7] (signals living on graphs representing 3-D color point clouds as in Figure 1).



[FIG1] The PdEs framework on weighted graphs can be applied to solve PDEs on 3-D point clouds.



[FIG2] Some objects from World War II acquired with our 3-D scanner at the Mémorial of Caen. (a) A lamp. (b) A canteen. (c) Part of a destroyed church.

3-D COLOR POINT CLOUDS

This section details how 3-D images of historic or art items can be acquired. We first provide a review of the benefits and disadvantages of each of the current technologies. In cultural heritage, two main approaches are used to digitize a 3-D object.

The first approach is photogrammetry. This is an image-based modeling technique where multiple photos are taken to retrieve 3-D data coordinates. Photogrammetry requires only cameras and is the least expensive approach. However, it works only with small objects that have regular geometric shapes. The second approach is laser scanning. In this approach, the scanner acquires many 3-D points of the object to digitize. The precision is much better but both the cost and the acquisition time are greater. In this article, we focus on this latter type of 3-D raw point clouds.

Figure 2 shows the acquisition of several World War II items, acquired by a Konica Minolta VIVID-9i noncontact 3-D laser

scanner with a lens of $f = 25$ -mm focal length at a distance of 0.6 m. According to the specification of the scanner, this corresponds to an accuracy of digitizing of ± 0.05 mm.

A point cloud is composed of raw data $\mathcal{D} = \{d_1, \dots, d_n\}$. One associates to each $d_i \in \mathcal{D}$ a 3-D coordinate vector $\mathcal{P} = \{p_1, \dots, p_n\}$ with $p_i (p_i^x, p_i^y, p_i^z)^T \in \mathbb{R}^3$, and a color vector $C = \{c_1, \dots, c_n\}$ with $c_i (c_i^R, c_i^G, c_i^B)^T \in \mathbb{R}^3$. We define for a given point cloud a function $f^*: \mathcal{D} \rightarrow \mathbb{R}^3$, which is either $f^*: \mathcal{D} \rightarrow \mathcal{P}$ or $f^*: \mathcal{D} \rightarrow C$. The function f^* is difficult to process and to analyze with classical signal processing techniques because the underlying point cloud lacks of any topological structure. Therefore, classical 2-D signal processing techniques cannot be considered and new dedicated graph signal processing techniques are needed.

GRAPH CONSTRUCTION FROM POINT CLOUDS

The proposed approach to process point clouds starts by creating a weighted graph from a given point cloud. This section explains how to build such a graph. This latter method is composed of two parts: first the topology of the graph is defined and then the edges of the graph are weighted.

GRAPH CONSTRUCTION

This section discusses how to build a weighted graph from 3-D point clouds. The creation of a graph requires several steps.

- 1) Vertices are created from the raw data.
- 2) The vertices are connected with edges to build a proximity graph based on geometrical structure of the point cloud.
- 3) Weights associated with each edge are calculated.

Weights of edges are deduced from values associated to vertices (the graph signal) and patches can be used to compute a better similarity value that takes account of local neighborhood similarities.

Graph creation from point clouds is challenging. Indeed, the structuring information is lacking, and the data is not naturally organized in a manifold. Therefore, the set of edges cannot be easily determined. Given a point cloud P with associated coordinates $\{\mathbf{p}_1, \dots, \mathbf{p}_n\} \in \mathbb{R}^3$, there are many ways of associating a graph to such a data set. Since point clouds data exhibit a geometrical structure, proximity graphs are preferable: if two data points satisfy certain geometric requirements, the corresponding vertices in the graph are connected by an edge. To each raw data point $d_i \in \mathcal{D}$, one associates a vertex of a proximity graph \mathcal{G} to define a set of vertices $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$. Then, determining the edge set \mathcal{E} of the proximity graph \mathcal{G} requires defining the neighbors of each vertex v_i according to its embedding \mathbf{p}_i using the Euclidean distance. Among many possible choices, we choose to consider the symmetric k -nearest neighbor graph. An undirected edge (v_i, v_j) is added between two vertices v_i and v_j if the distance between \mathbf{p}_i and \mathbf{p}_j is among the k smallest distances from either \mathbf{p}_i or \mathbf{p}_j to all the other data points. The construction of such a graph being computationally expensive for large point clouds, a k -dimensional tree is used [8] to speed up the k -nearest neighbor search.

IN CULTURAL HERITAGE, TWO MAIN APPROACHES ARE USED TO DIGITIZE A 3-D OBJECT.

GRAPH WEIGHTS

Once the graph has been created, it has to be weighted. If one wants to ignore the vertex similarities, the weight function w can be set to $w(v_i, v_j) = 1, \forall (v_i, v_j) \in \mathcal{E}$. To

account for the similarities between the graph signals associated with the vertices, it is possible to use similarity functions based on distances to define the edge weights. Given an initial function $\mathbf{f}^0: \mathcal{V} \rightarrow \mathbb{R}^m$, computing distances between vertices consists of comparing their features (as a function of \mathbf{f}^0). To this end, each vertex v_i is associated with a feature vector $\mathcal{P}(v_i) \in \mathbb{R}^q$. From this, a usual similarity measure is provided by the Gaussian kernel

$$w(v_i, v_j) = \exp\left(-\frac{\|\mathcal{P}(v_i) - \mathcal{P}(v_j)\|_2^2}{\sigma^2}\right). \quad (1)$$

Traditionally, one has simply $\mathcal{P}(v_i) = \mathbf{f}^0(v_i)$. However, in image processing an important feature vector is provided by image patches [9]. In [10], we have proposed a new definition of patches that can be used with any graph representation associated to meshes or point clouds. We detail the latter.

PATCH DEFINITION

Around each vertex we build a 2-D grid (the patch) describing the neighborhood. This grid is defined on the tangent plane of the point (i.e., the vertex). The patch is oriented and finally filled in with a weighted average of the graph-signal values in the local neighborhood. We detail these two steps next.

Orientation

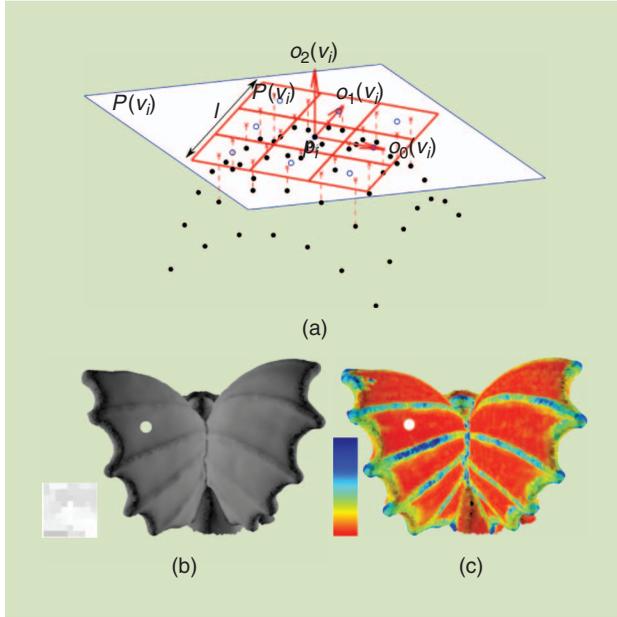
The first step consists of estimating the orientation of each patch. The algorithm first deduces a tangent vector $\mathbf{t}_1(v_i)$ from the normal vector $\mathbf{n}(v_i)$. We use a local principal component analysis on the coordinates p_i to estimate this normal vector: $\mathbf{t}_0(v_i) = \mathbf{n}(v_i)$. Let $\mathbf{x}, \mathbf{y}, \mathbf{z}$ be the three axes of a 3-D space, the first tangent vector $\mathbf{t}_1(v_i)$ is computed with

$$\begin{cases} \mathbf{t}_1(v_i) = \mathbf{z} \times \mathbf{t}_0(v_i) & \text{if } |\mathbf{z} \cdot \mathbf{t}_0(v_i)| \neq 1 \\ \mathbf{t}_1(v_i) = \mathbf{x} \times \mathbf{t}_0(v_i) & \text{otherwise} \end{cases} \quad (2)$$

with \times the cross product and \cdot the dot product. The condition on the first line of (2) checks if the \mathbf{t}_1 and \mathbf{z} axis vector are collinear. If they are not collinear, \mathbf{t}_1 is calculated from the \mathbf{z} axis, otherwise the \mathbf{x} axis is used. Then a bitangent vector $\mathbf{t}_2(v_i)$ is computed by $\mathbf{t}_2(v_i) = \mathbf{t}_0(v_i) \times \mathbf{t}_1(v_i)$. The orientation vectors $\mathbf{o}_0(v_i), \mathbf{o}_1(v_i), \mathbf{o}_2(v_i)$ are then respectively assigned to $\mathbf{t}_1(v_i), \mathbf{t}_2(v_i), \mathbf{t}_0(v_i)$.

Patch Construction

The second step consists of constructing the patches. Given a point \mathbf{p}_i , a patch is defined for this point by constructing a



[FIG3] (a) The interpolation of the content of the patch. l is the patch length. $\mathbf{o}_0(v_i)$ and $\mathbf{o}_1(v_i)$ are the orientation of the patch $\mathcal{P}(v_i)$ at a point \mathbf{p}_i . Elements marked by “ \times ” symbol correspond to the projected neighbors \mathbf{p}'_i of the point \mathbf{p}_i on the patch. These projections are used to deduce values of each patch cell (a “ \circ ” symbol) by a weighted average of the associated graph signal values. (b) A point cloud with a selected vertex (in white), and the patch descriptor of that vertex. (c) A point cloud colored by the patch-based distance between all points and a given selected one, from most similar (in red) to least similar (in blue).

square grid around \mathbf{p}_i on its tangent plane in the orientation of the patch defined by $(\mathbf{o}_0, \mathbf{o}_1)$. We fix the patch length l manually. Let n be the number of cells on a row of the patch. A square lattice of n^2 cells is constructed around \mathbf{p}_i with respect to the basis obtained from the orientation computation. Each cell has a side length of l/n . A local graph is then considered that connects the vertex v_i to all the vertices v_j contained in a sphere of diameter $l\sqrt{2}$. Then, all the neighbors v_j of v_i are projected on the tangent plane of \mathbf{p}_i giving rise to projected points \mathbf{p}'_i . To fill the patch with values, these projected points \mathbf{p}'_i are associated with the cells with the closest center. The value of the cell is then deduced from a weighted average of the values $f^0(v_j)$ associated with the vertices v_j that were projected onto the patch cell. This value is a spectral value (the point's colors). The set of values inside the patch of the vertex v_i are denoted $\mathcal{P}(v_i)$. Let $C_k(v_i)$ denote the k th cell of the constructed patch around v_i with $k \in [1, n^2]$. With the proposed patch construction process, define the set $V_k(v_i) = \{v_j | \mathbf{p}'_j \in C_k(v_i)\}$ as the set of vertices v_j that were associated with the k th patch cell of v_i . Then, the patch vector is defined as :

$$\mathcal{P}(v_i) = \left(\frac{\sum_{v_j \in V_k(v_i)} w_p(\mathbf{c}_k, \mathbf{p}_j) f^0(v_j)}{\sum_{v_j \in V_k(v_i)} w_p(\mathbf{c}_k, \mathbf{p}_j)} \right)_{k \in [1, n^2]}^T, \quad (3)$$

with $w_p(\mathbf{c}_k, \mathbf{p}_i) = \exp(-\|\mathbf{c}_k - \mathbf{p}'_i\|_2^2 / \sigma^2)$ where the \mathbf{c}_k are the coordinate vectors of the k th patch cell center. This weighting takes into account the distribution of the points in the cells of the patch by computing their mean feature vector. Figure 3(a) summarizes the method of patch construction. Figure 3(b) and (c) shows that points with similar geometric configurations are close with respect to the patch distance. With the following (parameters are respectively $\sigma = 0.3$ and $n = 5$).

GRAPH SIGNAL PROCESSING

This section provides the mathematical definitions needed to understand the construction of weighted graphs from point clouds.

PdEs ON WEIGHTED GRAPHS

The definitions presented here provide the basis on which it is possible to translate PDEs on graphs into PdEs on graphs. Most of these definitions are from [6].

DEFINITIONS

A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ consists of a finite set $\mathcal{V} = \{v_1, \dots, v_N\}$ of N vertices and a finite set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ of weighted edges. Assume \mathcal{G} to be undirected, with no self-loops and no multiple edges. Let (v_i, v_j) be the edge of \mathcal{E} that connects two vertices v_i and v_j of \mathcal{V} . Its weight, denoted by $w(v_i, v_j)$, represents the similarity between its vertices. Similarities are usually computed by using a positive symmetric function $w: \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$ satisfying $w(v_i, v_j) = 0$ if $(v_i, v_j) \notin \mathcal{E}$. The notation $v_i \sim v_j$ is also used to denote two adjacent vertices. The degree of a vertex v_i is defined as $\delta_w(v_i) = \sum_{v_j \sim v_i} w(v_i, v_j)$. Let $\mathcal{H}(\mathcal{V})$ be the Hilbert space of real-valued functions defined on the vertices of a graph. A function $f \in \mathcal{H}(\mathcal{V})$ assigns a real value $f(v_i)$ to each vertex $v_i \in \mathcal{V}$. We define the internal border of a set $A \subset \mathcal{V}$ as $\partial^- A = \{v_i \in A | \exists v_j \sim v_i, v_j \in \partial A\}$.

Similarly, we define the space $\mathcal{H}(\mathcal{E})$ of functions that are defined on the set \mathcal{E} of edges. Given a function $f: \mathcal{V} \rightarrow \mathbb{R}$, its ℓ_p and ℓ_∞ norms are given by

$$\|f\|_p = \left(\sum_{v_i \in \mathcal{V}} |f(v_i)|^p \right)^{1/p}, \quad \text{with } 1 \leq p < \infty, \quad (4)$$

$$\|f\|_\infty = \max_{v_i \in \mathcal{V}} |f(v_i)|, \quad \text{for } p = \infty. \quad (5)$$

DIFFERENCE OPERATORS ON WEIGHTED GRAPHS

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ be a weighted graph and $w: \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$ a weight function that depends on the interactions between the vertices. The difference operator [6], denoted $d_w: \mathcal{H}(\mathcal{V}) \rightarrow \mathcal{H}(\mathcal{E})$, is defined for all $f \in \mathcal{H}(\mathcal{V})$ and $(v_i, v_j) \in \mathcal{E}$ by

$$(d_w f)(v_i, v_j) = \sqrt{w(v_i, v_j)} (f(v_j) - f(v_i)). \quad (6)$$

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

$$(\nabla_w f)(v_i) = ((d_w f)(v_i, v_j))_{v_j \in \mathcal{V}}^T. \quad (7)$$

The ℓ_p norm of this vector is defined, for $p \geq 1$, as

$$\|(\nabla_w f)(v_i)\|_p = \left(\sum_{v_j \sim v_i} w(v_i, v_j)^{p/2} |f(v_j) - f(v_i)|^p \right)^{1/p}. \quad (8)$$

The external and internal morphological directional difference operators are defined as in [11] to be $((d_w f)(v_i, v_j))^\pm$, with $(x)^+ = \max(x, 0)$ and $(x)^- = -\min(x, 0) = \max(-x, 0) = (-x)^+$. The corresponding discrete upwind weighted gradients are

$$(\nabla_w^\pm f)(v_i) = \left(((d_w f)(v_i, v_j))^\pm \right)_{v_j \in \mathcal{V}}^T. \quad (9)$$

The ℓ_p and the ℓ_∞ norms of these gradients are

$$\|(\nabla_w^\pm f)(v_i)\|_p = \left[\sum_{v_j \sim v_i} w(v_i, v_j)^{p/2} \left((f(v_j) - f(v_i))^\pm \right)^p \right]^{1/p}, \quad (10)$$

$$\|(\nabla_w^\pm f)(v_i)\|_\infty = \max_{v_j \sim v_i} \sqrt{w(v_i, v_j)} (f(v_j) - f(v_i))^\pm. \quad (11)$$

p -LAPLACE OPERATORS ON WEIGHTED GRAPHS

The isotropic weighted p -Laplace operator, with $p \in [1, +\infty[$, at a vertex $v_i \in \mathcal{V}$ is defined on $\mathcal{H}(\mathcal{V})$ by [6] as

$$(\Delta_{w,p}^i f)(v_i) = \sum_{v_j \sim v_i} \psi_{w,p}^i(v_i, v_j) (f(v_i) - f(v_j)), \quad (12)$$

where

$$\psi_{w,p}^i(v_i, v_j) = \frac{1}{2} w(v_i, v_j) (\|\nabla_w f(v_i)\|_2^{p-2} + \|\nabla_w f(v_j)\|_2^{p-2}). \quad (13)$$

The ∞ -Laplacian is defined by [12] as

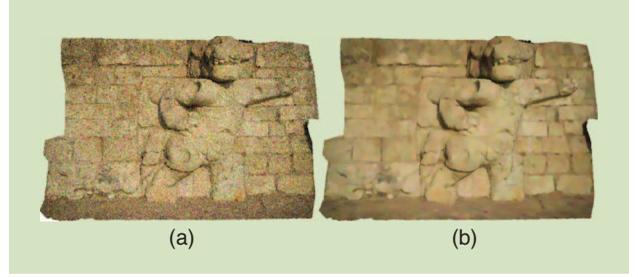
$$(\Delta_{w,\infty} f)(v_i) = \frac{1}{2} \left[\|(\nabla_w^+ f)(v_i)\|_\infty - \|(\nabla_w^- f)(v_i)\|_\infty \right]. \quad (14)$$

From the definitions of these discrete difference operators on graphs, we are now in position to translate any PDEs that involves gradient, p -Laplacian or ∞ -Laplacian in their continuous formulation onto Euclidean domains. In the sequel, we will consider directly the discrete formulation on graphs; see [6] for further details.

REGULARIZATION ON GRAPHS

Here we present some PDEs on graphs and show a methodology to regularize the functions defined on the vertices of graphs. Let $f^0 \in \mathcal{H}(\mathcal{V})$ be a given function defined on the vertices of a weighted graph $G = (\mathcal{V}, \mathcal{E}, w)$. In a given context, this function represents an observation of a clean function $h \in \mathcal{H}(\mathcal{V})$ corrupted by an additive noise $n \in \mathcal{H}(\mathcal{V})$ such that $f^0 = h + n$. Regularizing functions on graphs using either isotropic or anisotropic p -Laplacian, was proposed in [6] and [13]. Recently, a new family of p -Laplace operators based on a divergence formulation, which unifies both the isotropic and anisotropic p -Laplacian on graphs, has been proposed in [10].

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 a function $f \in \mathcal{H}(\mathcal{V})$, which is regular enough on G , and



[FIG4] The filtering of a noisy Mayan wall with a preservation of edges: (a) a noisy point cloud and (b) a denoised point cloud.

also close enough to f^0 . This can be formalized by the minimization of an energy functional, that involves a regularization term (or penalty term) plus an approximation one (or fitting term). This article considers the following model:

$$h \approx \operatorname{argmin}_{f: \mathcal{V} \rightarrow \mathbb{R}} J_{w,p}^\phi(f) + \frac{\lambda}{2} \|f - f^0\|_2^2 \quad (15)$$

$$\text{where } J_{w,p}^\phi(f) = \sum_{v_i \in \mathcal{V}} \phi(\|(\nabla_w f)(v_i)\|_p) \quad (16)$$

is a gradient-based functional, and $\lambda \in \mathbb{R}$ is a regularization parameter, called the Lagrange multiplier, that controls the tradeoff between the penalty term and the fitting term. The function $\phi(\cdot)$ is a positive convex function that penalizes large variations of f in the neighborhood of each vertex.

To solve (16), we consider the Euler–Lagrange equations

$$\frac{\partial J_{w,p}^\phi(f)}{\partial f(v_i)} + \lambda (f(v_i) - f^0(v_i)) = 0, \forall v_i \in \mathcal{V}, \quad (17)$$

where the first term denotes the variation of (16) with respect to f at a vertex v_i .

In [10], we have proven that the solution of (17) can be obtained with the following iterative algorithm:

$$f^{n+1}(v_i) = \frac{\lambda f^0(v_i) + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f} f^n(v_j)}{\lambda + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f}} \quad (18)$$

with

$$\beta_{v_i, v_j}^{\phi, p, f} = \alpha_{v_i, v_j}^{\phi, p, f} |f(v_j) - f(v_i)|^{p-2} \quad (19)$$

and

$$\alpha_{v_i, v_j}^{\phi, p, f} = w(v_i, v_j)^{p/2} \left(\frac{\phi'(\|(\nabla_w f)(v_i)\|_p)}{\|(\nabla_w f)(v_i)\|_p^{p-1}} + \frac{\phi'(\|(\nabla_w f)(v_j)\|_p)}{\|(\nabla_w f)(v_j)\|_p^{p-1}} \right). \quad (20)$$

Figure 4 shows such color filtering of a given 3-D colored point cloud (obtained from a laser scan of a Mayan temple wall).

APPLICATIONS

This section gives an overview of 3-D color point cloud applications. We illustrate the abilities of the proposed methods and algorithms

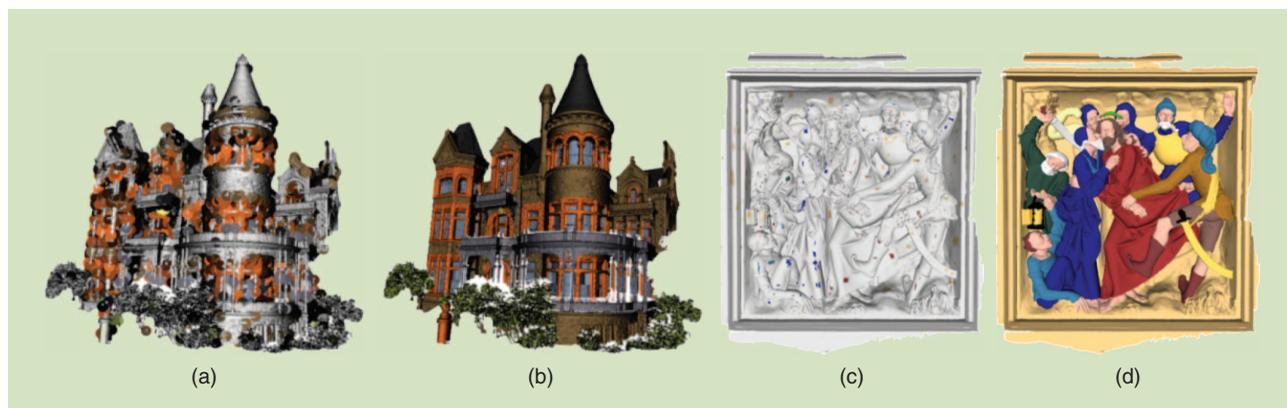
for signal processing on point clouds. (All of the 3-D point clouds we used are available at <https://lozes.users.greyc.fr/>.) The typical graph signals are point (respectively vertices) coordinates or colors. Given a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ associated to a point cloud, consider an initial graph signal $\mathbf{f}^0: \mathcal{V} \rightarrow \mathbb{R}^m$ with $m = 3$. This signal will be either the vertices' coordinates, in which $\mathbf{f}^0(v_i) = \mathbf{p}_i$, or the vertices' colors, in which $\mathbf{f}^0(v_i) = (f^R(v_i), f^G(v_i), f^B(v_i))^T$, where $f^X(v_i)$ denotes the X color component of the color at the vertex v_i .

The approach we developed can be interesting within a cultural heritage documentation, analysis, and dissemination perspective (according to the evaluation done by art expert Livio De Luca, as mentioned in the "Acknowledgments" section). First,

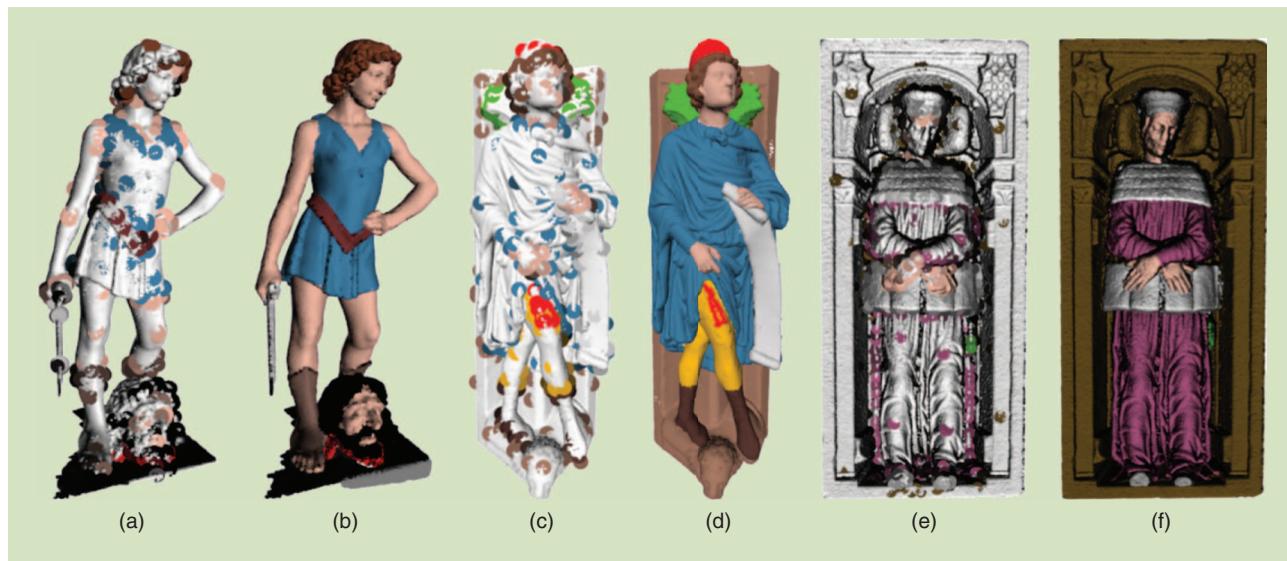
THE USE OF ADAPTIVE GEOMETRICAL WEIGHTS RELYING ON A GEOMETRIC FEATURE MAPS PATCHES ENABLES TO BETTER TAKE INTO ACCOUNT THE GEOMETRY OF THE POINT CLOUD DURING THE COLOR DIFFUSION AND BLENDING.

the idea to introduce a method for structuring color/texture information within a point-based 3-D representation is particularly relevant within the 3-D digitization and documentation purposes. Points clouds (especially if enriched by color information) include the essential geometric information useful for several applications scenarios

related to the heritage artifacts analysis (measurement, visualization, semantic annotation, etc.). The point clouds processing methods we suggest (colorization, filtering and simplification, and inpainting of missing data and segmentation) can find several effective applications within the digital driven process for documenting heritage buildings, archaeological sites, and museums' objects.



[FIG5] The colorization of the bishop castle (approximately 1.5 million points) and a Bas-relief (with 506,000 points): (a) the original point cloud with annotations, (b) the colored point cloud, (c) the colorless Bas-relief with annotations, and (d) the colored Bas-relief.



[FIG6] The colorization of some cultural heritage statues [sizes of point clouds: (a) and (b) 253,000, (c) and (d) 254,000, (e) and (f) 791,000 points]. (a) Scribbles, (b) result, (c) scribbles, (d) result, (e) scribbles, and (f) result.

COLORIZATION

Colorization is the process of adding colors on an uncolored object. Let $f^0: \mathcal{V} \rightarrow \mathcal{C}$ be a function that assigns colors to vertices. Let $A \subset \mathcal{V}$ be the subset of vertices with unknown colors and ∂A the subset of vertices with known values. The purpose of interpolation is to find a function \hat{f} approximating f^0 in \mathcal{V} minimizing the following isotropic total variation energy for $v_i \in A$:

$$\min_{f \in H(\mathcal{V})} \left\{ R_{w,p}(f) = \sum_{v_i \in A} \|\nabla_w f(v_i)\|_p^p \right\} \quad (21)$$

with $f(v_i) = f^0(v_i)$, for $v_i \in \partial A$. This can again be solved using the Euler–Lagrange equations

$$\begin{cases} (\Delta_{w,p}^i f)(v_i) = 0 & \forall v_i \in A, \\ f(v_i) = f^0(v_i) & \forall v_i \in \partial A. \end{cases} \quad (22)$$

The solution of (22) can be obtained with the iterative algorithm in (18) with $\lambda = 0, \forall v_i \in \partial^- A$. The similarity function $w: \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ is computed from the comparison of patches of a geometric feature. This geometric feature is computed from the degree at a vertex using local height weights. These later weights are obtained from the similarity of height patches. Since the colorization starts from initial color annotations, not all the points are considered simultaneously. The colorization starts from the points that are neighbors to the annotated colors and the set of points to be colorized grows as the algorithm iterates. The colorization process stops when the set of vertices to colorize is empty and has converged to a stable solution. The use of adaptive geometrical weights relying on a geometric feature maps patches enables to better take into account the geometry of the point cloud during the color diffusion and blending.

Figure 5 shows the colorization of the bishop castle and of a Bas-relief (a type of sculpture in which shapes are carved). Figure 6 shows the colorization on some uncolored statues. Both cases show that the colorization allows restoration of colors not captured by the 3-D scanner (the bishop castle case), but also allows placement of colors on an initially uncolored object (statues cases) for a more realistic rendering.

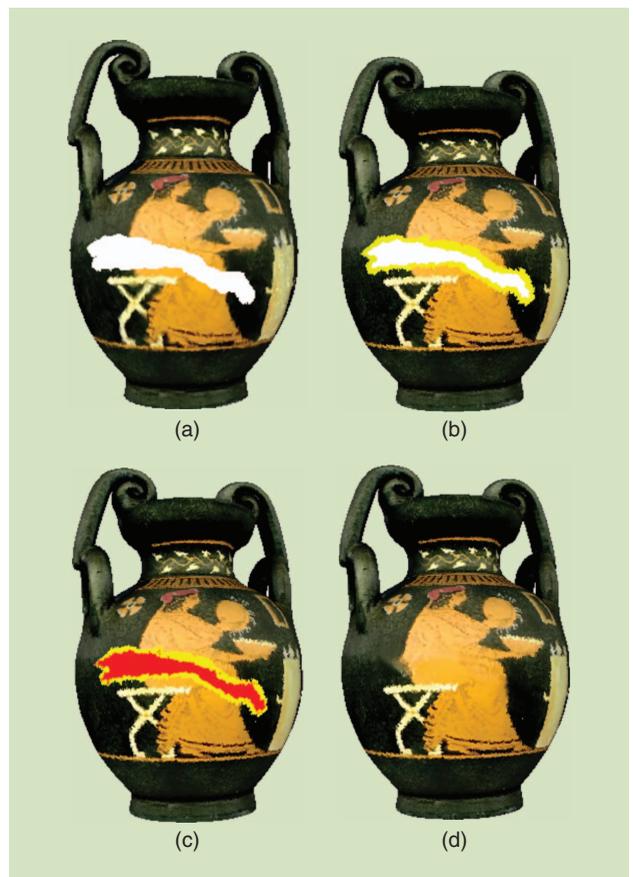
FILTERING AND SIMPLIFICATION

Modern 3-D scanners can generate large point clouds with several million or billion points. The processing of such data is difficult. Instead of downscaling the point clouds and losing details, we propose a way to remove fine details without modify the appearance of the point cloud. It can be interesting to filter and simplify complex and redundant point clouds for Web publication purposes.

Let $f^0: \mathcal{V} \rightarrow \mathcal{P}$ be the vertices coordinates. The iterative algorithm of (18) allows filtering the geometry of 3-D objects using $\lambda = 0$. Figure 7 shows the simplification of a point cloud of the Saint Eligius statue with a reduction in the number of points by 77.6% using parameters $p = 0.1$ and $\phi(s) = s^p$. With such a simplified point cloud, the processing of any data on this point cloud, like colors, becomes computationally feasible. Indeed, reducing the quantity of data to process makes algorithms and renderings faster. Finally, the simplification could be leveraged in



[FIG7] The simplification of a point cloud of the Saint Eligius statue after 10,000 iterations. (a) The original point cloud (201,000 points) and (b) the point cloud simplified (45,000 points).



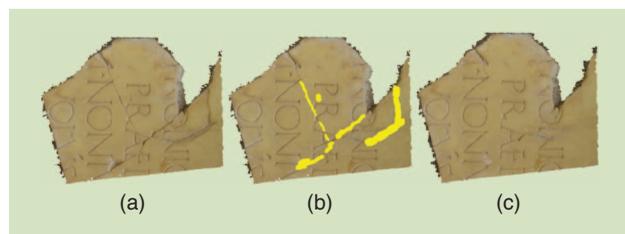
[FIG8] The restoration pipeline of a vase. The geometric part is first filled, then the missing color is inpainted: (a) the denoised vase, (b) the hole to fill in (boundary in yellow), (c) the filled-in hole, and (d) the inpainted hole.

a meshing process by producing more regular polygons.

INPAINTING OF MISSING DATA

Inpainting consists of constructing new values for missing data in coherence with a set of known data. This can be the reconstruction of deteriorated parts of a 3-D object represented as a point cloud. This can be of interest for extending color information to missing parts (e.g., for virtual restoration purposes). Recent inpainting work tends to unify local and nonlocal approaches under a variational formulation (see [14] and references therein for more details). We consider that data are defined on a general domain represented on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. Let $f^0: \mathcal{V} \rightarrow \mathcal{C}$ be a function that assigns colors to vertices. Let $\mathcal{A} \subset \mathcal{V}$ be the subset of vertices with unknown values and $\partial\mathcal{A}$ the subset of vertices with known values. The purpose of the interpolation is to find a function \hat{f} approximating f^0 in \mathcal{V} and that corresponds to solving

$$\begin{cases} (\Delta_{w, \infty} f)(v_i) = 0 & \forall v_i \in \mathcal{A}, \\ f(v_i) = f^0(v_i) & \forall v_i \in \partial\mathcal{A}. \end{cases} \quad (23)$$



[FIG9] The inpainting of a part of a damaged wall: (a) the object to inpaint, (b) the inpainting marker, and (c) the inpainting result.

INPAINTING CONSISTS OF CONSTRUCTING NEW VALUES FOR MISSING DATA IN COHERENCE WITH A SET OF KNOWN DATA.

The infinity Laplacian is used here for interpolation since it provides better inpainting results than the isotropic p -Laplacian interpolation (see [15]). Works in [10] have shown that this interpolation problem has a unique solution that can

be obtained using the iterative algorithm presented in [12]. At the end of each iteration the set $\partial\mathcal{A}$ is updated by $\partial\mathcal{A}^{(n+1)} = \partial\mathcal{A}^{(n)} \cup \partial^-\mathcal{A}^{(n)}$, and $\partial^-\mathcal{A}^{(n+1)}$ is updated from $\partial\mathcal{A}^{(n+1)}$. The algorithm stops when the set of vertices to inpaint is empty. Figure 8 shows the restoration of a broken vase. The geometric part is corrected first, then the hole is filled; finally, the color is inpainted. One major objective of the cultural heritage is the preservation of an object. This inpainting algorithm provides a plausible virtual reconstruction of the original state of an object. The algorithm can also be used to remove and subvert parts of a damaged object, as in Figure 9.

SEGMENTATION

Segmentation is the process of partitioning a 3-D object into multiple regions. Let $f: \mathcal{V} \rightarrow \mathbb{R}$ be the function on a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ to segment. The segmentation problem can be formulated as a PDE to be solved on weighted graphs. This latter formulation is based on front propagation using the eikonal equation to compute general distances on graphs. Let $L = \{l_1, \dots, l_n\}$ be the set of labels and let $S^0 = S_1^0 \cup \dots \cup S_n^0$ be the corresponding set of labeled vertices. The goal of label propagation is to label each vertex $u \in \mathcal{V}$ under the condition that u is a neighbor of S_i^0 . Such a label propagation can be formulated as solving the eikonal equation on a graph \mathcal{G} as

$$\begin{cases} \|(\nabla_w \phi)(v_i, t)\|_p = P_i(x) & v_i \in \mathcal{V} \\ f(v_i) = \phi(v_i) & x \in \gamma \subset \mathcal{V}, \end{cases} \quad (24)$$



[FIG10] Segmentation of a vase by resolving the eikonal equation on graphs [770,000 points for (a) and 200,000 points for (d)]: (a) a vase with labels, (b) the result, (c) a vase with labels, and (d) the result.

where $F_i: \mathcal{V} \rightarrow \mathbb{R}^3$ is a speed function, $P_i(v_i) = (1/F_i(v_i))$, and $\phi: \mathcal{V} \rightarrow \mathbb{R}$ represent the set of initials labels. The resolution of the eikonal equation (24) on a weighted graph has been investigated in [16].

To segment a mesh or a point cloud according to the vertices colors, compute the similarity function $w: \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$ by taking into account the colorimetric distance as $w = e^{-\|f^0(v_i) - f^0(v_j)\|/\sigma^2}$, where $f^0: \mathcal{V} \rightarrow \mathcal{C}$ are the colors associated at the node $v_i \in \mathcal{V}$. The function $\phi: \mathcal{V} \rightarrow \mathbb{R}$ associates an initial label to each vertex v_i . The parameters are set as follows: $p = 2$, $F_i(v_i) = 1$, $\forall v_i \in \mathcal{V}$. Figure 10 shows some segmentation results on point clouds after the resolution of the eikonal equation. Segmentation results are 3-D subsets of the original points clouds. These subsets can then be processed by applying the previous graph signal techniques such as filtering, simplification, inpainting, and colorization.

CONCLUSIONS

This article has proposed an approach for the processing of functions on point clouds represented as graphs. We have shown how to translate PDEs using the framework of PdEs. The approach allows processing of signal data on point clouds (e.g., spectral data, colors, coordinates, and curvatures). We have applied this approach for cultural heritage purposes on examples aimed at restoration, denoising, hole-filling, inpainting, object extraction, and object colorization. The results demonstrate the many potentials of the point cloud approach to the processing of cultural heritage 3-D scanned objects.

ACKNOWLEDGMENTS

Livio De Luca is the director of the Models and Simulations for Architecture and Cultural Heritage Laboratory, a French research unit of the National Centre for Scientific Research and the Ministry of Culture and Communication working on the application of informatics technologies for the study of architecture considered in its heritage, constructive, urban, and landscape dimensions. We thank him for his evaluation of the potential applications of these methods mentioned in the “Applications” section.

AUTHORS

François Lozes (francois.lozes@unicaen.fr) received his Ph.D. degree in computer science at the Université de Caen Basse-Normandie, France. His research mainly concerns the analysis of data on 3-D meshes and 3-D point clouds using graph signal processing.

Abderrahim Elmoataz (abderrahim.elmoataz-billah@unicaen.fr) is a full professor of computer science at the Université de Caen Basse-Normandie, France. His research concerns partial differential equations (PDEs) on 3-D surfaces and points clouds, PDEs on graphs of arbitrary topology with applications in image processing, and machine learning.

Olivier Lézoray (olivier.lezoray@unicaen.fr) received the M.Sc. and doctoral degrees in computer science from the Université de Caen Basse-Normandie, France, in 1996 and 2000, respectively. From 1999 to 2000, he was an assistant professor in the Computer Science Department at the same university. From 2000 to 2009, he was an associate professor at the Cherbourg Institute of Technology in the Communication Networks and Services Department. Since 2010, he has been a full professor at the Cherbourg Institute of Technology. His research is focused on color image segmentation and filtering (graph-based variational and morphological methods) and machine-learning techniques for image mining (neural networks and support vector machines).

REFERENCES

- [1] L. J. Van der Maaten and E. O. Postma, “Texton-based analysis of paintings,” in *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 2010, pp. 77980H–77980H.
- [2] P. Abry, S. Jaffard, and H. Wendt, “When van Gogh meets Mandelbrot: Multifractal classification of painting’s texture,” *Signal Process.*, vol. 93, no. 3, pp. 554–572, 2013.
- [3] M. Hess and S. Robson, “3D colour imaging for cultural heritage artefacts,” in *ISPRS*, vol. 5, 2010, pp. 288–292.
- [4] *Digital Imaging for Cultural Heritage Preservation: Analysis, Restoration, and Reconstruction of Ancient Artworks*, (Digital Imaging and Computer Vision Series), F. Stanco, S. Battiato, and G. Gallo, Eds. New York: Taylor and Francis, 2011.
- [5] B. Cornelis, T. Ružić, E. Gezels, A. Doms, A. Pižurica, L. Platiša, J. Cornelis, M. Martens, M. D. Mey, and I. Daubechies, “Crack detection and in painting for virtual restoration of paintings: The case of the Ghent Altarpiece,” *Signal Process.*, vol. 93, no. 3, pp. 605–619, 2013.
- [6] A. Elmoataz, O. Lezoray, and S. Bougleux, “Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing,” *IEEE Trans. Image Processing*, vol. 17, no. 7, pp. 1047–1060, 2008.
- [7] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Processing Mag.*, vol. 30, no. 3, pp. 83–98, 2013.
- [8] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, “An optimal algorithm for approximate nearest neighbor searching fixed dimensions,” *J. ACM*, vol. 45, no. 6, pp. 891–923, Nov. 1998.
- [9] A. Buades, B. Coll, and J.-M. Morel, “Image denoising methods. A new nonlocal principle,” *SIAM Rev.*, vol. 52, no. 1, pp. 113–147, 2010.
- [10] F. Lozes, A. Elmoataz, and O. Lezoray, “Partial difference operators on weighted graphs for image processing on surfaces and point clouds,” *IEEE Trans. Image Processing*, vol. 23, no. 9, pp. 3896–3909, Sept. 2014.
- [11] V.-T. Ta, A. Elmoataz, and O. Lézoray, “Nonlocal PDEs-based morphology on weighted graphs for image and data processing,” *IEEE Trans. Image Processing*, vol. 26, no. 2, pp. 1504–1516, June 2011.
- [12] A. Elmoataz, X. Desquesnes, and O. Lezoray, “Non-local morphological pdes and p-Laplacian equation on graphs with applications in image processing and machine learning,” *IEEE J. Sel. Top. Signal.*, vol. 6, no. 7, pp. 764–779, 2012.
- [13] O. Lezoray, V.-T. Ta, and A. Elmoataz, “Partial differences as tools for filtering data on graphs,” *Pattern Recogn. Lett.*, vol. 31, no. 14, pp. 2201–2213, 2010.
- [14] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro, “A variational framework for exemplar-based image inpainting,” *Int. J. Comput. Vision*, vol. 93, no. 3, pp. 319–347, 2011.
- [15] M. Ghoniem, A. Elmoataz, and O. Lézoray, “Discrete infinity harmonic functions: towards a unified interpolation framework on graphs,” in *Proc. IEEE Int. Conf. Image Processing*, 2011, pp. 1361–1364.
- [16] X. Desquesnes, A. Elmoataz, and O. Lézoray, “Eikonal equation adaptation on weighted graphs: Fast geometric diffusion process for local and non-local image and data processing,” *J. Math. Imaging Vis.*, vol. 46, no. 2, pp. 238–257, 2013.

