



**HAL**  
open science

# Tug-of-War games and PDEs on graphs: simple image and high dimensional data processing

Hamza Ennaji, Yvain Quéau, Abderrahim Elmoataz

► **To cite this version:**

Hamza Ennaji, Yvain Quéau, Abderrahim Elmoataz. Tug-of-War games and PDEs on graphs: simple image and high dimensional data processing. *Scientific Reports*, 2023, 13 (1), pp.6045. 10.1038/s41598-023-32354-5 . hal-03675971v1

**HAL Id: hal-03675971**

**<https://hal.science/hal-03675971v1>**

Submitted on 23 May 2022 (v1), last revised 14 Apr 2023 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Tug-of-War games and PDEs on graphs: simple image and high dimensional data processing

Hamza Ennaji<sup>1,\*</sup>, Yvain Quéau<sup>1</sup>, and Abderrahim Elmoataz<sup>1</sup>

<sup>1</sup>Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

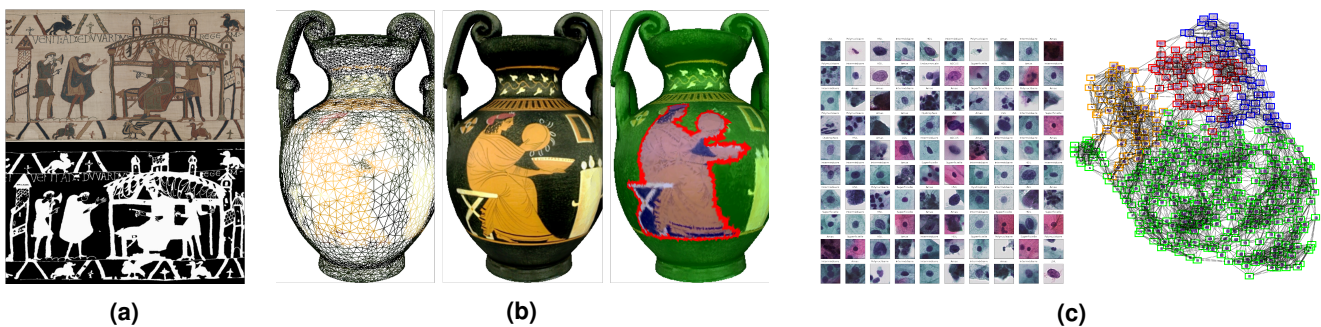
\*hamza.ennaji@unicaen.fr

## ABSTRACT

The aim of this note is to revisit the connections between some stochastic games, namely Tug-of-War games, and a class of nonlocal PDEs on graphs. We consider a general formulation of Tug-of-War games which is shown to be related to many classical PDEs in the continuous setting. We transcribe these equations on graphs using ad hoc differential operators and we show that it covers several nonlocal PDEs on graphs such as  $\infty$ -Laplacian, game  $p$ -Laplacian and the eikonal equation. This unifying mathematical framework allows us to easily design simple algorithms to solve several inverse problems in imaging and data science, with a particular focus on cultural heritage and medical imaging.

## Introduction

Local and nonlocal partial differential equations (PDEs) play a central role in the mathematical analysis and modelling of phenomena arising in physics, biology, economics, image and signal processing, computer vision, etc. Nonlocal PDEs compare favourably to the classical ones, thanks to their ability to preserve geometric and repetitive structures. Yet, the numerical approximation of classical PDEs, based on finite differences (FD), finite elements (FE) or finite volumes (FV) has a certain limitation especially when the computational domain is irregular or is a graph with arbitrary topology. This made discretizing and solving PDEs on graphs and networks gain attention and interest due to numerous applications in imaging, computer vision and machine learning, where data are given in the form of graphs or functions defined on graphs (see Figure 1). Examples of such applications include, but are not limited to, segmentation, filtering, semi-supervised clustering, and classification.



**Figure 1.** Examples of N-dimensional graphs, and of data processing problems which can be applied to them. (a) A 2D grid graph representing a color image, and the 2D segmentation of this image; (b) a 3D triangular graph representing a colored point cloud, and the 3D segmentation of this point cloud; (c) a ND point cloud graph constructed from a database of cells, and the data clustering of this graph. All these problems can be tackled using the same unified mathematical framework described in the present paper.

In such settings, data points are vertices of the graph and are connected by edges if sufficiently close in a certain ground metric. Using discrete vector calculus<sup>1-3</sup>, one defines finite difference operators which are analogous to the differential operators in the continuous setting. This mimetic approach allows recovering many tools and results in the discrete level. This has been applied, e.g., for the  $\infty$ -Laplacian, the eikonal equation and the mean curvature flow<sup>2,4,5</sup>. Yet, each particular application requires resorting to a particular PDE, hence such approaches remain difficult to manipulate for non-experts.

On the other hand, the approximation of PDEs has recently taken advantage of the emergence of techniques based on deterministic or stochastic games such as the Tug-of-War (TOW) game related to the  $\infty$ -Laplacian<sup>6,7</sup> and the  $p$ -Laplacian<sup>8</sup>. In particular, this formalism has been considered for studying the existence and regularity of a class of PDEs<sup>9</sup>. It has also been used for discretizing the game theoretic  $p$ -Laplacian and the eikonal equation on graphs, which finds many applications in image processing and high dimensional clustering<sup>5,10,11</sup>. The main tool to make a link between the TOW games and PDEs is the dynamic programming principle (DPP): at each step of the game, the value function at some point can be obtained by summing up all the possible outcomes.

The aim of this paper is to emphasize that most of the PDEs considered in image and signal processing are particular instances of a generic TOW game, and that this observation provides a simple methodology to solve inverse problems through nonlocal averaging-based algorithms. We will start in Section 1 by providing some theoretical background on the notion of TOW games and PDEs on graphs, and giving the connections between them. Then, we will present in Section 2 a few simple algorithms which can be designed to solve inverse problems, based on the transcriptions of such equations on graphs. In Section 3, we will provide numerical examples to illustrate the proposed method on real-life problems arising from cultural heritage and medical imaging, before drawing our conclusions in Section 4.

## 1 From TOW to PDEs on graphs

### TOW games and continuous PDEs

Let us briefly recall the notion of TOW as introduced by Peres and Sheffield<sup>8</sup>. The TOW game is a two player random turn zero-sum game played on a domain  $\Omega \subset \mathbb{R}^N$  with a running payoff function  $h : \Omega \rightarrow \mathbb{R}$  and a payoff function  $g$  defined on  $\partial\Omega$ . A token is placed at an initial position  $x_0 \in \Omega$  and each player can move the token to a new position in an  $\epsilon$ -neighborhood of the current position (typically, an open ball  $B_\epsilon(x_k)$  if the token is at position  $x_k$ ), for a fixed step size  $\epsilon > 0$ . A fair coin is tossed, if player I wins, the token is moved to the position  $x_k^I$ , otherwise to the position  $x_k^{II}$ . The process continues until the token reaches some position  $x_f \in \partial\Omega$ . In that case, the game stops and player I's payoff is  $g(x_f) + \epsilon^2 \sum_{i=1}^{f-1} h(x_i)$ . Since the game is zero-sum, player I will try to maximize the payoff while player II will try to minimize it. For this game, the DPP reduces to

$$\begin{cases} u_\epsilon(x) = \frac{1}{2} \left\{ \sup_{y \in B_\epsilon(x)} u_\epsilon(y) + \inf_{y \in B_\epsilon(x)} u_\epsilon(y) \right\} + \epsilon^2 h(x) & \text{in } \Omega, \\ u_\epsilon(x) = g(x) & \text{on } \partial\Omega. \end{cases} \quad (1)$$

Moreover, it is shown<sup>6</sup> that the limit of  $u_\epsilon$  as  $\epsilon \rightarrow 0$  solves the normalized  $\infty$ -Poisson equation

$$\begin{cases} -\Delta_\infty^N u(x) = h(x) & \text{in } \Omega, \\ u(x) = g(x) & \text{on } \partial\Omega, \end{cases} \quad (2)$$

where the normalized infinite Laplacian is given by  $\Delta_\infty^N u = |\nabla u|^{-2} \Delta_\infty u$ ,  $\Delta_\infty u = \sum_{i,j} \partial_i u \partial_{ij} u \partial_j u$ .

There exist several variants of this game. A first example consists in playing the previous game with a certain probability  $\beta \in [0, 1]$ , such that the token is moved to a position  $x_k^I$  or  $x_k^{II}$  with probability  $\beta$ , and to a random position in  $B_\epsilon(x)$  with probability  $1 - \beta$ . This version is called the TOW with noise<sup>12,13</sup>, and it is connected to the DPP

$$\begin{cases} u_\epsilon(x) = \frac{\beta}{2} \left\{ \sup_{y \in B_\epsilon(x)} u_\epsilon(y) + \inf_{y \in B_\epsilon(x)} u_\epsilon(y) \right\} + (1 - \beta) \int_{B_\epsilon(x)} u_\epsilon(y) dy + \epsilon^2 h(x) & \text{in } \Omega, \\ u_\epsilon = g & \text{on } \partial\Omega, \end{cases} \quad (3)$$

with  $\int_B v := \frac{1}{|B|} \int_B v$ . Moreover, setting  $\beta = \frac{p-2}{p+N}$ ,  $p \geq 2$ , and defining the normalized  $p$ -Laplacian according to  $\Delta_p^N u = \frac{1}{p} |\nabla u|^{2-p} \operatorname{div}(|\nabla u|^{p-2} \nabla u)$ , the limit as  $\epsilon \rightarrow 0$  of  $u_\epsilon$  solves the following  $p$ -Laplace equation:

$$\begin{cases} -\Delta_p^N u(x) = h(x) & \text{in } \Omega, \\ u(x) = g(x) & \text{on } \partial\Omega. \end{cases} \quad (4)$$

This variant generalizes the classical TOW game, in the sense that (3) comes down to (1) when  $\beta = 1$ , and similarly, (4) comes down (at least formally) to (2), as  $\lim_{\beta \rightarrow 1} p = \lim_{\beta \rightarrow 1} \frac{2 + \beta N}{1 - \beta} = \infty$ .

It is possible to give an even more general version as follows. Assume that each player chooses a position with probabilities  $\alpha$  and  $\beta$ , respectively, and that the game position moves with a uniform probability  $\gamma$ , with  $\frac{\alpha + \beta}{2} + \gamma = 1$ . In this case, the DPP reads

$$u_\epsilon(x) = \frac{\alpha}{2} \sup_{y \in B_\epsilon(x)} u_\epsilon(y) + \frac{\beta}{2} \inf_{y \in B_\epsilon(x)} u_\epsilon(y) + \gamma \int_{B_\epsilon(x)} u_\epsilon(y) dy + \epsilon^2 h(x), \quad x \in \Omega_X \quad (5)$$

This game can even be extended to the case where the probabilities  $\alpha$ ,  $\beta$  and  $\gamma$  depend on both time and position  $(t, x) \in [0, T] \times \Omega := \Omega_T$ , where  $T > 0$ . Therein, the value function of the game satisfies

$$u_\epsilon(x, t) = \frac{\alpha(x, t)}{2} \sup_{y \in B_\epsilon(x)} u_\epsilon(y, t - \frac{\epsilon^2}{2}) + \frac{\beta(x, t)}{2} \inf_{y \in B_\epsilon(x)} u_\epsilon(y, t - \frac{\epsilon^2}{2}) + \gamma \int_{B_\epsilon(x)} u_\epsilon(y, t - \frac{\epsilon^2}{2}) dy + \epsilon^2 h(x, t), \quad \text{in } \Omega_T. \quad (6)$$

In the particular case where  $\alpha(x, t) = \beta(x, t)$ , we obtain a space and time-dependent extension of (3). Moreover, setting  $\alpha(x, t) = \beta(x, t) = \frac{p(x, t) - 2}{p(x, t) + N}$  and  $\gamma(x, t) = \frac{N + 2}{p(x, t) + N}$ ,  $p \geq 2$ , the game is related to a normalized  $p(x, t)$ -Laplace equation<sup>14</sup>:

$$(N + p(x, t)) \partial_t u(x, t) = \Delta_{p(x, t)}^N u(x, t) + h(x, t),$$

which generalizes (4). However, the interpretation of (5) and (6) in terms of PDES, in the most general case where  $\alpha$ ,  $\beta$  and  $\gamma$  are only required to satisfy  $\frac{\alpha + \beta}{2} + \gamma = 1$ , is less straightforward. In the next paragraph, we prepare the main ingredients for this interpretation, which will also help us transcribing TOW games and the related PDES to the framework of graphs.

### Reminders on differential operators on graphs

**Definition 1** (Weighted graph). A weighted graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$  is a collection of vertices  $\mathcal{V}$ , edges  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$  and edges weights  $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$  with  $w(x, y) > 0$  if  $(x, y) \in \mathcal{E}$  and  $w(x, y) = 0$  otherwise. We assume that  $\mathcal{G}$  is finite (i.e.,  $\mathcal{V}$  consists of a finite number of points), simple (i.e., without loops and multiple edges), connected and undirected, that is  $w(x, y) = w(y, x)$  if  $(x, y) \in \mathcal{E}$ . We write  $x \sim y$  if  $(x, y) \in \mathcal{E}$  and we denote by  $I(x) := \{y \in \mathcal{V} : x \sim y\}$  the set of neighbors of  $x$ .

Given a function  $u : \mathcal{V} \rightarrow \mathbb{R}$ , we recall the definition of its gradients.

### Definition 2 ( $p$ -eikonal operators).

The discrete upwind nonlocal weighted gradients of  $u$  are defined by

$$\nabla_w^\pm u(x) = (\partial_y^\pm u(x))_{y \in \mathcal{V}},$$

where  $\partial_y^\pm u(x) = \left( \sqrt{w(x, y)} (u(y) - u(x)) \right)^\pm$ ,  $a^+ = \max(a, 0)$  and  $a^- = \max(-a, 0)$ .

Its  $\mathcal{L}_p$  norm is defined as

$$\|\nabla_w^\pm u(x)\|_p = \begin{cases} \max_{y \in I(x)} \left( \sqrt{w(x, y)} (u(y) - u(x))^\pm \right) & \text{for } p = \infty \\ \left[ \sum_{y \in I(x)} w(x, y)^{p/2} (u(y) - u(x))^\pm \right]^{\frac{1}{p}} & \text{for } 1 \leq p < \infty. \end{cases}$$

### Definition 3 (Laplacian operators on graphs).

1. The 2-Laplacian on graph is defined by

$$(\Delta_{w,2} u)(x) = \frac{\sum_{y \in I(x)} w(x, y) u(y)}{\sum_{y \in I(x)} w(x, y)} - u(x).$$

2. The  $\infty$ -Laplacian on graph is defined by

$$(\Delta_{w,\infty}u)(x) = \frac{1}{2} \left( \|\nabla_w^+ u(x)\|_\infty - \|\nabla_w^- u(x)\|_\infty \right).$$

3. For  $2 \leq p < \infty$ , the game  $p$ -Laplacian on graph is defined by

$$(\Delta_{w,p}^G u)(x) = \frac{p-2}{p} \Delta_{w,\infty}u(x) + \frac{2}{p} \Delta_{w,2}u(x).$$

### Connections between TOW games and PDEs on Euclidean graphs

We are now ready to introduce the transcription of the TOW games (5) and (6) in terms of PDEs on graphs. Let us consider an Euclidean graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$  with  $\mathcal{V} = \Omega \subset \mathbb{R}^N$  and a weight function

$$w(x, y) = \begin{cases} 1 & \text{if } |x - y| \leq \epsilon, \\ 0 & \text{otherwise.} \end{cases}$$

Then, using Definition 2, we easily get

$$\begin{aligned} \max_{y \in I(x)} u(y) &= \|\nabla_w^+ u(x)\|_\infty + u(x), \\ \min_{y \in I(x)} u(y) &= u(x) - \|\nabla_w^- u(x)\|_\infty. \end{aligned} \tag{7}$$

Plugging (7) in (5), we obtain the following interpretation of the generalized TOW game (5) in terms of a PDE:

$$-\Delta_{\alpha,\beta,\gamma}u(x) = h(x), \tag{8}$$

where  $\Delta_{\alpha,\beta,\gamma}u(x) = \frac{\alpha}{2} \|\nabla_w^+ u(x)\|_\infty - \frac{\beta}{2} \|\nabla_w^- u(x)\|_\infty + \gamma \Delta_{w,2}u(x)$ .

Similarly, in the case of the time-dependent TOW game (6), we obtain the following parabolic PDE:

$$\partial_t u(x, t) = \Delta_{\alpha(x,t), \beta(x,t), \gamma(x,t)} u(x, t) + h(x, t). \tag{9}$$

As can be seen in Table 1, by taking different values of  $\alpha, \beta$  and  $\gamma$  in the TOW games (5) and (6), we are able to recover different well-known PDEs on graphs. In order to apply this unifying mathematical framework to real-world imaging or data processing problems, it only remains to have at hand a practical way to discretize such PDEs, on graphs which are not necessarily Euclidean. This is discussed in the next paragraph.

**Table 1.** A large class of PDEs on graphs recovered from (8)-(9), which are obtained by taking particular parameter values in the TOW games (5)-(6).

TOW game parameters	Elliptic PDE	Parabolic PDE
$\alpha = \gamma = 0$ and $\beta = 1$	eikonal equation: $\ \nabla_w^- u(x, t)\ _\infty = h(x)$	$\partial_t u(x, t) = \ \nabla_w^- u(x)\ _\infty + h(x, t)$
$\alpha = \beta = 1$ and $\gamma = 0$	$\infty$ -Laplacian: $-\Delta_{w,\infty}u(x) = h(x)$	$\partial_t u(x, t) = \Delta_{w,\infty}u(x, t) + h(x, t)$
$\alpha = \beta = 0$ and $\gamma = 1$	Laplace equation: $-\Delta_{w,2}u(x) = h(x)$	$\partial_t u(x, t) = \Delta_{w,2}u(x, t) + h(x, t)$
$\alpha = \beta = \frac{p-2}{p}$ and $\gamma = \frac{2}{p}$	game $p$ -Laplace equation: $-\Delta_{w,p}^G u(x) = h(x)$	$\partial_t u(x, t) = \Delta_{w,p}^G u(x, t) + h(x, t)$

### Extension to General Weighted Graphs

We have seen that on Euclidean graphs, PDEs of the form (8) are related to some classical TOW games. A natural question that one may ask is what can be said in the case of general weighted graphs, e.g., images or 3D point clouds, where the weights are not induced by a metric. To this end, let us consider a general weighted graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ , and recall<sup>10</sup> the following notations for nonlocal dilation, nonlocal erosion and nonlocal mean, respectively:

$$\begin{aligned} \text{NLD}(u)(x) &= \|\nabla_w^+ u(x)\|_\infty + u(x) = u(x) + \max_{y \in I(x)} \left( \sqrt{w(x, y)} (u(y) - u(x))^+ \right), \\ \text{NLE}(u)(x) &= u(x) - \|\nabla_w^- u(x)\|_\infty = u(x) - \max_{y \in I(x)} \left( \sqrt{w(x, y)} (u(y) - u(x))^- \right), \\ \text{NLM}(u)(x) &= u(x) + \Delta_{w,2}u(x) = \frac{\sum_{y \in I(x)} w(x, y) u(y)}{\sum_{y \in I(x)} w(x, y)}, \end{aligned}$$

which come down to the classical morphological dilation, erosion and mean when  $w \equiv 1$ . Then, defining the following nonlocal averaging operator:

$$\text{NLA}(u) := \frac{\alpha}{2}\text{NLD}(u) + \frac{\beta}{2}\text{NLE}(u) + \gamma\text{NLM}(u),$$

Equations (8) and (9) can be rewritten as

$$u(x) - \text{NLA}(u)(x) = h(x), \text{ and } \partial_t u = u(x) - \text{NLA}(u)(x) + h(x), \quad (10)$$

This rewriting provides a practical way to discretize such nonlocal PDEs on general weighted graphs, as it suffices in practice to implement the nonlocal mathematical morphology operators above.

To conclude this section, let us remark that just as their counterparts on Euclidean graphs, nonlocal PDEs on general weighted graphs can also be interpreted in terms of TOW games. To simplify the presentation, let us take  $\alpha = \beta = 1$  and  $\gamma = 0$ , i.e., we are considering the equation  $-\Delta_{w,\infty}u = h(x)$ . Then, using Definitions 2-3, we easily see that this equation can be rewritten as<sup>10</sup>

$$u(x) = \max_{y \in \mathcal{V}} \left( \min_{z \in \mathcal{V}} \left( P(x,y,z)u(y) + (1 - P(x,y,z))u(z) \right) \right), \quad (11)$$

where  $P(x,y,z) = \frac{\sqrt{w(x,y)}}{\sqrt{w(x,y)} + \sqrt{w(x,z)}}$ . Now, think about the same configuration as the classical TOW game but with nonlocal neighbours, i.e., the token's displacement takes place in  $I(x_{k-1}) = \{x \in \mathcal{V} : w(x, x_{k-1}) > 0\} \cup \{x_{k-1}\}$  instead of  $\epsilon$ -balls with weight-dependent probabilities as follows. If the player I wins the toss, the token will move to a position  $x_k^I$  with probability  $P(x_{k-1}, x_k^I, x_k^II)$ , and to a position  $x_k^II$  with probability  $1 - P(x_{k-1}, x_k^I, x_k^II)$ . This can be interpreted as a nonlocal TOW game, whose value function satisfies (11).

To summarize, so far we have established a unifying mathematical framework based on the TOW games (5) and (6), which allows one to recover a series of well-known PDEs on graphs, as depicted in Table 1. We have then extended this analogy to nonlocal PDEs on general weighted graphs, and shown that they could be implemented in a simple manner by resorting to nonlocal mathematical morphology operators. In the next section, we describe a few simple algorithms for inverse problems, which follow directly from this theoretical discussion.

## 2 Simple algorithms for inverse problems

In this section we give simple algorithms based on the transcription of the TOW game on graphs, which will be used to solve a few inverse problems. As we shall see, the main features of these algorithms are their simplicity and ease of implementation, and the fact that many tasks can be achieved just by taking particular values of the TOW game parameters  $\alpha, \beta$  and  $\gamma$ .

### Unified interpolation for image and high dimensional data processing

We have seen that we are able to recover a bunch of local and nonlocal PDEs on graphs. We will use this as a methodology to solve several interpolation problems in image processing and machine learning. In particular, in the next section we will show examples in medical imaging and in cultural heritage corresponding to problems on graphs using PDEs of the form (8)-(9) with particular attention on colorization, inpainting and data classification. To do so we consider a subset  $A \subset \mathcal{V}$  consisting of vertices with the missing information. To simplify the presentation suppose that  $h \equiv 0$ . Then, many problems we encounter in image processing and computer vision can be recast in the form of interpolation problems, i.e., one seeks constructing new values starting from known values, which amounts to solve the Dirichlet problem

$$\begin{cases} -\Delta_{\alpha,\beta,\gamma}u = 0 & \text{in } A, \\ u = g & \text{on } \partial A, \end{cases} \quad (12)$$

where  $g : \partial A \rightarrow \mathbb{R}$  is the boundary data which depends on the considered application. To solve (12), the strategy is the following. We first consider the associated evolution problem:

$$\begin{cases} \partial_t u = \Delta_{\alpha,\beta,\gamma}u & \text{in } A, \\ u = g & \text{on } \partial A, \\ u|_{t=0} = u^0 & \text{in } A, \end{cases} \quad (13)$$

with some initial value  $u^0 : A \rightarrow \mathbb{R}$  chosen suitably for each application. As usual, (13) is solved using a Euler discretization by taking  $\partial_t u \approx \frac{u_{n+1} - u_n}{\Delta t}$ , where  $u_n(x) = u(x, n\Delta t)$ . Then (13) can be approximated via the following iterations:

$$\begin{cases} u_0 = u^0 & \text{in } A, \\ u_{n+1} = u_n + \Delta t \Delta_{\alpha, \beta, \gamma} u & \text{in } A, \\ u_{n+1} = g & \text{on } \partial A. \end{cases}$$

Taking  $\Delta t = 1$  and  $\Delta_{\alpha, \beta, \gamma} = \text{NLA}(u) - u$ , we get a nonlocal average filter consisting of convex combination of the nonlocal, dilation, erosion and mean terms introduced in (10):

$$\begin{cases} u_0 = u^0 & \text{in } A, \\ u_{n+1} = \text{NLA}(u_n) & \text{in } A, \\ u_{n+1} = g & \text{on } \partial A. \end{cases} \quad (14)$$

Thus, Problem (12) can be approximated via the iterative scheme (14) which consists of simple algebraic operations. This makes the approach convenient for users with no deep knowledge on PDEs to solve interpolation problems. Notice that this can also be applied to parabolic variants of (12).

### Unified segmentation and data clustering

For segmentation and data clustering, we make use of an adaptation of the eikonal equation on graphs. First, let us say that in this case one could make use of the iterative scheme (10) with  $\alpha = \gamma = 0$  and  $\beta = 1$  to solve the eikonal equation, however, for faster computations, it is preferable to use a monotone algorithm where we have an explicit formula for the solution<sup>4</sup>. Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$  be a weighted graph and consider the following eikonal equation:

$$\begin{cases} \|(\nabla_{\bar{w}} u)(x)\|_{\infty} = h(x) & \text{in } \mathcal{V} \setminus S_0, \\ u = 0 & \text{on } S_0, \end{cases} \quad (15)$$

with  $S_0 \subset \mathcal{V}$  the set of initial seed vertices. Using the operators defined in Definition 2, Equation (15) becomes

$$\max_{y \in I(x)} \left( \sqrt{w(x, y)} \max(0, u(x) - u(y)) \right) = h(x). \quad (16)$$

Then, setting  $C = h(x)$ ,  $k_i = \sqrt{1/w(x, y)}$ ,  $a_i = \{f(y_i) : y_i \in I(x_i)\}$  and  $n = \text{Card}(I(x))$ , Equation (16) becomes

$$\max_i \left( \frac{(x - a_i)^+}{k_i} \right) = C,$$

whose unique solution is explicitly given by

$$x^* = \min_{i=1, \dots, n} (a_i + k_i C),$$

which gives rise to a Dijkstra-like algorithm. Then, for semi-supervised graph clustering, this algorithm is extended by enabling label propagation on a general weighted graph (see [4, Algorithm 2]). To recapitulate the process, we denote by  $L = \{l_1, \dots, l_n\}$  the set of labels and by  $S^0 = \cup_{i=1}^n S_i^0$  the set of seeds, i.e., each  $S_i^0$  is a set of vertices marked by a label  $l_i$ . Then, we mark each vertex  $x \in \mathcal{V}$  with a label  $l_i$  provided  $x$  is closer to a vertex of  $S_i^0$  than other vertices of  $S^0$  which decomposes the graph into  $n$  clusters. Each label  $l_i$  is induced by a front  $\mathcal{F}_i$  initialized at  $\partial S_i^0$ , and the final configuration  $S_f$  is given by the set of vertices reached by  $\mathcal{F}_i$  until it is stopped by the boundary of the domain or by another front. Note that this is a particular case ( $p = \infty$ ) of the method to solve the eikonal equation with  $\mathcal{L}_p$  norm<sup>4</sup>. For  $p = 2$ , it is worth mentioning that the obtained scheme on a  $n$ -dimensional grid in  $R^n$  reduces to the celebrated Osher-Sethian scheme which can be solved using a Fast Marching method (FMM)<sup>15</sup>.

### 3 Applications

In all the sequel, we are given a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$  constructed as described below. We will provide examples of nonlocal inpainting, colorization, semi-supervised segmentation using the iteration (14) as well as the Fast Marching-like algorithm described above.

The results of this section will mainly concern applications on cultural heritage and medical imaging. Cultural heritage applications will essentially focus on scenes taken from the Bayeux Tapestry. The Tapestry is a unique record of the Norman conquest of England and the battle of Hastings. It is an eleventh-century medieval embroidery of 70m long and 50cm tall, which has been digitized during a project gathering the City of Bayeux, the University of Caen-Normandy, the CNRS and the GREYC. Concerning medical imaging, we shall focus on computer-aided cytology. Cytopathologists usually have to study the morphology and texture of cytoplasm and nuclei as imaged in microscopy to make a diagnosis decision. This task can be tedious since every slide supporting sample contains millions of cells and the presence of infected cells is rare. The use of PDEs on graphs for computer-aided cytology has been shown to be efficient for segmentation and semi-supervised classification of cells<sup>16</sup>. All the results concerning cytology are produced using the database of the Cherbourg University Hospital.

#### Construction of graphs

The first step is to construct an appropriate graph for the given data to apply the obtained algorithms. There are several well-known techniques to construct a weighted graph from a given discrete data set depending on whether the data is unorganized or structured. For unstructured data, we can mention  $\epsilon$ -neighbourhood graphs where we connect two elements  $x, y \in \mathcal{V}$  by an edge if  $d(x, y) \leq \epsilon$  for some  $\epsilon > 0$ , where  $d$  is some distance on  $\mathcal{G}$ . We also find the  $k$ -nearest neighborhood graph ( $k$ -NNG) where each vertex is connected with its  $k$  nearest neighbours with respect to the distance  $d$  (this will give rise to directed graphs due to the nonsymmetry of this neighborhood relationship, however undirected graph can still be obtained using a modification in the construction<sup>17</sup>).

Structured data (e.g., images, meshes), can naturally be described by grid graphs. In the case of images, each pixel is connected by an edge to adjacent vertices. Region adjacency graphs (RAG) were also designed for images where vertices correspond to image regions and edges are obtained by considering an adjacency distance. This can be generalized for other data, where a region  $R_i$  is defined as a set of vertices such that  $\mathcal{V} = \cup_i R_i$  and  $\cap_i R_i = \emptyset$ . Then,  $R_i$  and  $R_j$  are adjacent if there exists  $x \in R_i, y \in R_j$  such that  $x \sim y$  (see Figure 1).

As for the weight function  $w$ , it is generally computed via a similarity map  $s : \mathcal{E} \rightarrow \mathbb{R}^+$ , so that  $w(x, y) = s(x, y)$  if  $(x, y) \in \mathcal{E}$  and 0 otherwise. In general we take

$$s(x, y) = 1, \quad s(x, y) = \exp\left(\frac{-d(u^0(x), u^0(y))}{\sigma^2}\right), \quad \text{or } s(x, y) = (d(u^0(x), u^0(y)) + \epsilon)^{-1}, \quad \epsilon \rightarrow 0,$$

where  $\sigma > 0$ . The choice of the parameter  $\sigma$  can be done globally or locally, i.e., by assigning to each vertex a parameter  $\sigma_x$  depending on the application. For images, we generally make use of patches as follows. Pick some vertex  $x \in \mathcal{V}$  and consider a square  $C(x, n)$  of size  $n^2$  centred at  $x$ . Then  $\mathcal{P}(x) = \left(u^0(y)\right)_{y \in C(x, n)}^T$  is called a patch centred at  $x$ . Using this definition, one can define a similarity map via  $s(x, y) = \exp\left(\frac{-\|\mathcal{P}(x) - \mathcal{P}(y)\|_2^2}{\sigma^2}\right)$ . This notion can be extended to the case of 3-dimensional points clouds<sup>18</sup>.

#### Nonlocal inpainting

Inpainting is a virtual restoration processes. It consists in reconstructing a damaged or incomplete part of an image. Most of the proposed algorithms to perform this task are based on PDEs and variational formulations<sup>19,20</sup>. Nonlocal approaches for image inpainting have been widely considered since the work on nonlocal filtering<sup>21</sup>. This approach was particularly successful for the treatment of textured images, for which local methods had many limitations. To formulate the problem within our framework, we consider  $A \subset \mathcal{V}$  the set of vertices with missing data and prescribe boundary condition on  $\partial A$  via a function  $g : \partial A \rightarrow \mathbb{R}^c$ , with  $c$  the number of color channels of the image ( $c = 3$  in the examples below). We apply the iterative scheme (14) with  $\alpha = \beta = 1, \gamma = 0$ , i.e., the  $\infty$ -Laplacian is used for the inpainting of 2D images and 3D point clouds from scribbles giving reference colors.

In Figure 2a we use our approach for the virtual restoration of a 2D image of King Edward taken from the Bayeux Tapestry. A part of this scene was damaged due to wax stains caused by candles when the Tapestry was at the cathedral. As one sees, the proposed algorithm allows “virtually restoring” the part which has been damaged.





**Figure 2.** Virtual restoration and colorization. (a) 2D restoration (original images, images with areas to be restored indicated in red, and restoration results). (b) 2D colorization, where the left column represents the backside of a scene from the Bayeux Tapestry, and the second column represents the initial image, the image with seeds, and the colored result. (c-d) 3D colorization, with input models on the left and colored ones on the right.

### 2D-3D colorization

Colorization consists in adding colors to monochromatic or colorless images. It can be particularly helpful in cultural heritage and movie industry applications, or in conjunction with 3D acquisition techniques. Colorization of monochromatic images is based on the intensity channel to determine similarities between pixels and proceed to color diffusion from scribbles. For 3D data, the intensity channel is missing and one needs to proceed differently. An optimization-based approach was proposed for 3D meshes colorization<sup>22</sup> relying on similarities between mesh vertices via the spin image descriptor<sup>23</sup>. Yet, this approach seems not suitable for 3D point clouds as they have no intrinsic connectivity. On the other hand, formulating 3D colorization as an interpolation problem allows us exploiting the simple nonlocal averaging iterations (14), starting from a few user-specified color scribbles.

For 2D images, we apply our schemes to recover the original colors (at least vivid ones) of the Bayeux Tapestry. This is done using a "reflection" trick as follows. In fact, the colors in the exposed side of the Tapestry tend to bleed and fade over time due to sunlight, humidity and reaction with chemicals such as oxygen, ozone etc. Colors such as blue, red, orange, dark green and brown are more susceptible to this. Since the backside of the Tapestry is less exposed, one can guess the original colors from the yarns in the back of the embroidery. We exploit this to use appropriate seeds for the colorization process. This is illustrated in the right column of Figure 2b where we find the initial image, image with seeds and the colored images where the seeds are taken from the left column of Figure 2b. The top row in Figure 2c illustrates virtual colorization of a 3D model representing the "Betrayal of Judas"<sup>24</sup> using the proposed scheme (more particularly, the  $\infty$ -Laplacian). A detailed comparison between the obtained results (with different values of  $\alpha$ ,  $\beta$  and  $\gamma$ ) and an alternative approach<sup>22</sup> can be found in a dedicated paper<sup>10</sup>. The bottom row in Figure 2d illustrates the colorization of a 3D model from the chimney of the old Abbaye du voeu at Cherbourg<sup>25</sup> and in particular, we see that the method can be applied to change colors.

### Semi-supervised segmentation and data clustering

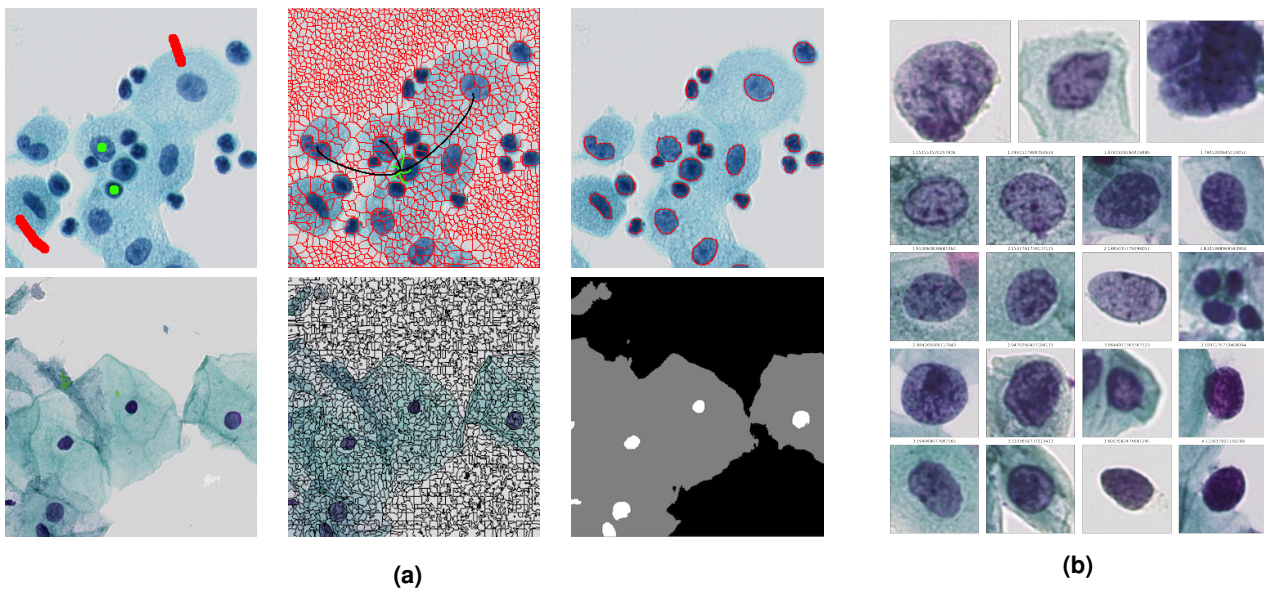
Image segmentation consists in partitioning images into multiple regions to make them easier to analyse. In particular, one wishes to detect objects such as roads in satellite images, tumors in medical imaging, etc. As for data clustering, it consists in splitting data into different families of objects sharing similar properties, such as infected cells in a medical database, etc. All these tasks can be achieved using the eikonal equation (15), with  $h \equiv 1$  and labels propagation as described before. One of the advantages of this formulation is that it allows segmentation on any graph representing images. In the examples below, we use a 4-adjacency grid graph to build an image partition. The obtained region map is then turned into a RAG. To allow labels growing beyond local neighbors, each vertex neighbor is extended by a  $k$ -NN based on mean color value.

In the following examples, we perform segmentation using superpixels decomposition which allows reducing image complexity by regrouping pixels in a region map while preserving contours. This can be done by dilating a regular grid of seeds by the label propagation method using the eikonal equation<sup>4</sup> in such a way to preserve the local structure of the image. This is illustrated in Figure 3, left, where we show the segmentation of 2D and 3D scenes from the Bayeux Tapestry.



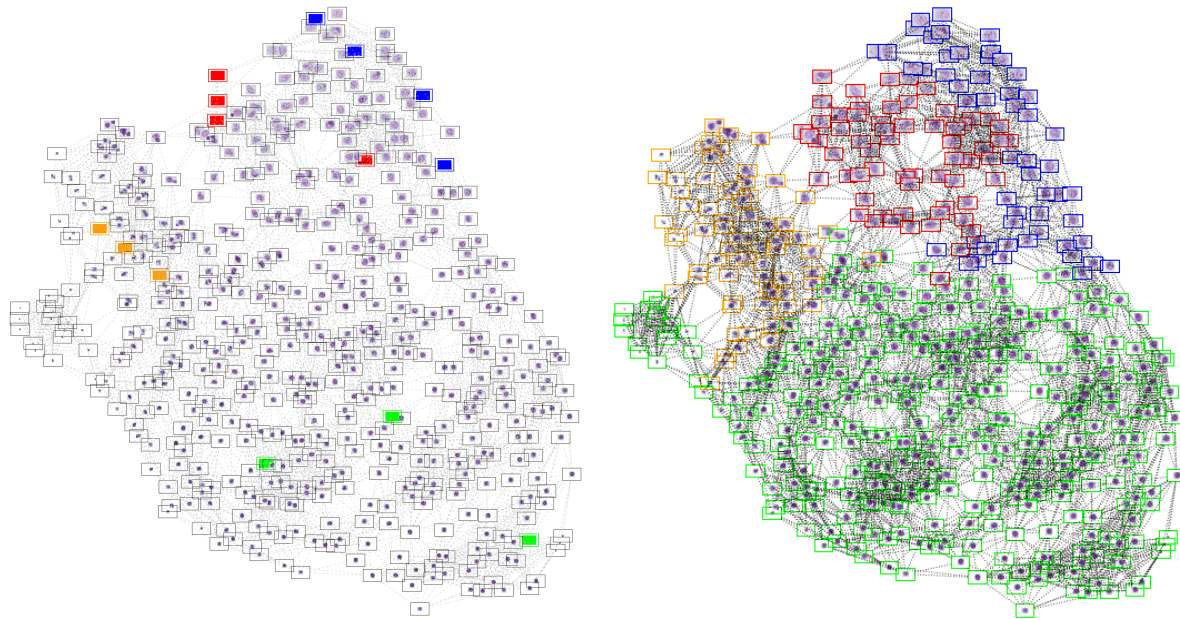
**Figure 3.** Semi-supervised segmentation of the Bayeux Tapestry. Left: superpixel-based segmentation of a single scene (initial image, image with seeds, image partition with supervertices using a regular grid of seeds, and segmentation result). Right, first and second rows: segmentation of the foreground on 2D images (initial image and segmentation result). Right, third row: 3D point cloud segmentation (from left to right: initial image, image with initial labels, segmentation result).

Figure 4a shows how our framework can be used for segmenting medical images. In fact, each image can contain various objects to be segmented. In particular, nuclei are numerous and not adjacent, but some are still concentrated in some regions and are separated by the cytoplasm or the background. This suggests considering nuclei as a first class and the background and cytoplasm as the second class. The nonlocal graph structure is needed to allow labels grow beyond local neighbours. To this end, the eikonal equation is used twice: first, for the superpixel decomposition, and then for segmentation. Thanks to this approach, only few seeds are needed to perform efficient segmentation of images containing multiple and non-adjacent objects to detect. Another application we can consider is indexation. In fact, after segmenting cells using the eikonal equation, we get a database of cells which can be represented in the form of a graph. The top of Figure 4b illustrates some abnormal cells that will be used as seeds. Then, using the eikonal equation one can compute the distance map and thus, find the closest cells to the abnormal cells, as illustrated in the bottom of Figure 4b where we can see 16 cells amongst the database that are the closest to the seeds.



**Figure 4.** Applications in medical imaging. (a) Semi-supervised segmentation of cell images (from left to right: images with initial labels, superpixel segmentation and segmentation results). (b) Indexation using the eikonal equation (top: abnormal cells used as seeds; bottom: some indexed cells from the database).

We conclude this section by applying our approach to semi-supervised classification of cells in a cytological slide into normal and abnormal classes. We first perform segmentation using the eikonal equation as discussed above to extract nuclei. We used 10% seeds on a dataset of 3956 cells extracted from a cytological slide divided by the cytologists into 4 classes where nuclei are described by characteristics concerning the form, surface, color, and the texture of cells, etc. Then we perform label propagation using the eikonal equation (with  $p = 1, 2$ ) on a nearest neighbour graph as described in the previous section. This gives the result illustrated in Figure 5 which are promising since we get 98.47% classification accuracy for  $p = 1$  and 98.27% for  $p = 2$ . These algorithms are present in the Antarctic software, which was developed during the phd thesis of X. Desquesnes<sup>26</sup> to provide an accessible support to pathologists for the classification and the segmentation of cytological slides. The software comes with a database of millions of cells and allows pathologists to create and modify characteristics.



**Figure 5.** Illustration of semi-supervised data clustering using the eikonal equation. Left: graph built on a database of cytological cells, where some cells are initially labelled (blue, green, red). Right: final clustering.

## 4 Discussion

In this note, we revisited the notion of tug-of-war games and related PDEs on graphs. We showed that the transcription of such games on graphs allows recovering a bunch of nonlocal elliptic and parabolic PDEs on graphs. We used this observation as a unified methodology to solve interpolation problems. This gives rises to simple algorithms for image and high dimensional data processing. We illustrated this approach by examples from cultural heritage and medical imaging. In future work we are planing to consider other real life applications and compare results with the related works. In addition, the continuum limit of the considered PDEs, as the number of vertices goes to infinity, will be investigated in depth.

## References

1. Bertozzi, A. L. & Flenner, A. Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Model. Simul.* **10**, 1090–1118, DOI: [10.1137/11083109X](https://doi.org/10.1137/11083109X) (2012).
2. Elmoataz, A., Lezoray, O. & Bogleux, S. Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. *IEEE Transactions on Image Process.* **17**, 1047–1060, DOI: [10.1109/TIP.2008.924284](https://doi.org/10.1109/TIP.2008.924284) (2008).
3. Elmoataz, A., Desquesnes, X., Lakhdari, Z. & Lézoray, O. Nonlocal infinity laplacian equation on graphs with applications in image processing and machine learning. *Math. Comput. Simul.* **102**, 153–163 (2014).

4. Desquesnes, X., Elmoataz, A. & L  zoray, O. Eikonal equation adaptation on weighted graphs: fast geometric diffusion process for local and non-local image and data processing. *J. Math. Imaging Vis.* **46**, 238–257, DOI: [10.1007/s10851-012-0380-9](https://doi.org/10.1007/s10851-012-0380-9) (2013).
5. Elmoataz, A., Toutain, M. & Tenbrinck, D. On the  $p$ -Laplacian and  $\infty$ -Laplacian on graphs with applications in image and data processing. *SIAM J. Imaging Sci.* **8**, 2412–2451, DOI: [10.1137/15M1022793](https://doi.org/10.1137/15M1022793) (2015).
6. Peres, Y., Schramm, O., Sheffield, S. & Wilson, D. B. Tug-of-war and the infinity Laplacian. *J. Am. Math. Soc.* **22**, 167–210, DOI: [10.1090/S0894-0347-08-00606-1](https://doi.org/10.1090/S0894-0347-08-00606-1) (2009).
7. Peres, Y., Pete, G. & Somersille, S. Biased tug-of-war, the biased infinity Laplacian, and comparison with exponential cones. *Calc. Var. Partial Differ. Equ.* **38**, 541–564, DOI: [10.1007/s00526-009-0298-2](https://doi.org/10.1007/s00526-009-0298-2) (2010).
8. Peres, Y. & Sheffield, S. Tug-of-war with noise: a game-theoretic view of the  $p$ -Laplacian. *Duke Math. J.* **145**, 91–120, DOI: [10.1215/00127094-2008-048](https://doi.org/10.1215/00127094-2008-048) (2008).
9. Ruosteenoja, E. Local regularity results for value functions of tug-of-war with noise and running payoff. *Adv. Calc. Var.* **9**, 1–17, DOI: [10.1515/acv-2014-0021](https://doi.org/10.1515/acv-2014-0021) (2016).
10. Elmoataz, A., Lozes, F. & Toutain, M. Nonlocal PDEs on graphs: from tug-of-war games to unified interpolation on images and point clouds. *J. Math. Imaging Vis.* **57**, 381–401, DOI: [10.1007/s10851-016-0683-3](https://doi.org/10.1007/s10851-016-0683-3) (2017).
11. Elmoataz, A., Desquesnes, X. & Toutain, M. On the game  $p$ -Laplacian on weighted graphs with applications in image processing and data clustering. *Eur. J. Appl. Math.* **28**, 922–948, DOI: [10.1017/S0956792517000122](https://doi.org/10.1017/S0956792517000122) (2017).
12. Manfredi, J. J., Parviainen, M. & Rossi, J. D. On the definition and properties of  $p$ -harmonious functions. *Ann. Sc. Norm. Super. Pisa, Cl. Sci. (5)* **11**, 215–241, DOI: [10.2422/2036-2145.201005\\_003](https://doi.org/10.2422/2036-2145.201005_003) (2012).
13. Manfredi, J. J., Parviainen, M. & Rossi, J. D. Dynamic programming principle for tug-of-war games with noise. *ESAIM, Control Optim. Calc. Var.* **18**, 81–90, DOI: [10.1051/cocv/2010046](https://doi.org/10.1051/cocv/2010046) (2012).
14. Parviainen, M. & Ruosteenoja, E. Local regularity for time-dependent tug-of-war games with varying probabilities. *J. Differ. Equations* **261**, 1357–1398, DOI: [10.1016/j.jde.2016.04.001](https://doi.org/10.1016/j.jde.2016.04.001) (2016).
15. Sethian, J. A. A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci. USA* **93**, 1591–1595, DOI: [10.1073/pnas.93.4.1591](https://doi.org/10.1073/pnas.93.4.1591) (1996).
16. Toutain, M., Elmoataz, A., Desquesnes, X. & Pruvot, J.-H. A unified geometric model for virtual slide image processing and classification. *IEEE J. Sel. Top. Signal Process.* **10**, 151–160, DOI: [10.1109/JSTSP.2015.2482941](https://doi.org/10.1109/JSTSP.2015.2482941) (2016).
17. Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R. & Wu, A. Y. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. ACM* **45**, 891–923, DOI: [10.1145/293347.293348](https://doi.org/10.1145/293347.293348) (1998).
18. Lozes, F., Elmoataz, A. & L  zoray, O. Partial difference operators on weighted graphs for image processing on surfaces and point clouds. *IEEE Trans. Image Process.* **23**, 3896–3909, DOI: [10.1109/TIP.2014.2336548](https://doi.org/10.1109/TIP.2014.2336548) (2014).
19. Ghoniem, M., Chahir, Y. & Elmoataz, A. Nonlocal video denoising, simplification and inpainting using discrete regularization on graphs. *Signal Process.* **90**, 2445–2455, DOI: [10.1016/j.sigpro.2009.09.004](https://doi.org/10.1016/j.sigpro.2009.09.004) (2010).
20. Arias, P., Facciolo, G., Caselles, V. & Sapiro, G. A variational framework for exemplar-based image inpainting. *Int. J. Comput. Vis.* **93**, 319–347, DOI: [10.1007/s11263-010-0418-7](https://doi.org/10.1007/s11263-010-0418-7) (2011).
21. Buades, A., Coll, B. & Morel, J.-M. La chasse au bruit dans les images. In *Math  matiques et vision*, 1–38 (Paris: Soci  t   Math  matique de France, 2006).
22. Leifman, G. & Tal, A. Mesh Colorization. *Comput. Graph. Forum* DOI: [10.1111/j.1467-8659.2012.03021.x](https://doi.org/10.1111/j.1467-8659.2012.03021.x) (2012).
23. Johnson, A. & Hebert, M. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis Mach. Intell.* **21**, 433–449, DOI: [10.1109/34.765655](https://doi.org/10.1109/34.765655) (1999).
24. Web archive. <https://web.archive.org/web/20101228172600/http://www.artec3d.com/gallery/3d-models>.
25. Inventaire g  n  ral du patrimoine culturel de normandie. <https://inventaire-patrimoine.normandie.fr/dossier/cheminee-de-l-ancienne-abbaye-du-voeu/117cac38-f30e-49aa-a38a-2ee7401b9f47>.
26. Desquesnes, X. Propagation de fronts et  $p$ -laplacien normalis   sur graphes : algorithmes et applications au traitement d’images et de donn  es. (2012).

## **Acknowledgements**

This work was supported by the ANR grant "Inclusive Museum Guide" (ANR-20-CE38-0007). AE is partially supported by the ANR grant "SUMUM" (ANR-17-CE38-0004).

## **Author contributions statement**

All authors contributed to the writing, discussions and revisions of the manuscript.

## **Competing interests**

The authors declare no competing interests.