EDPs morphologique sur graphes pour le Filtrage et l'Inpainting d'images sur nuages de points 3D.

François Lozes[1] Abderrahim Elmoataz[1,2]

[1] Université de Caen Normandie et de ENSICAEN, Laboratoire GREYC,
Équipe Image, 6 Boulevard Maréchal Juin, F-14050 Caen Cedex, France
[2] Université de Paris-Est Marne-La-Valée, Laboratoire LIGIM, Cité Descartes,
Bâtiment Copernic - 5, bd Descartes, Champs sur Marne 77454 Marne-la-Vallée Cedex 2, France

{francois.lozes,abderrahim.elmoataz-billah}@unicaen.fr

Résumé

Nous proposons de transcrire certains Équations aux Dérivées Partielles (EDPs) morphologiques sur graphes en utilisant le cadre des des Équations aux différences Partielles (EdPs). Ce qui permettra de définir de nouveaux opérateurs morphologiques adaptatifs. Nous montrons ensuite comment ces opérateurs peuvent être utilisés pour résoudre des problèmes de filtrage, et d'interpolation de nuages de points 3D ou d'images sur ces nuages de points. Nous présentons des applications du patrimoine culturel 3D.

Mots Clef

EdP sur graphes, EDP sur nuage de points, traitement non-local, traitement de signal sur graphe, filtrage, filtrage, inpainting.

Abstract

In this paper, we propose an adaptation of morphological Partial Differential Equations (PDEs) on graphs using the framework of Partial difference Equations (PdEs). This enables to define adaptive morphological operators on graphs. We then show how these operators can be used for interpolation and filtering of raw 3D point clouds. To enable a patch-based processing of point clouds, we also show how a weighted graph based on patches can be associated with a point cloud. Finally, we present applications in cultural heritage.

Keywords

PdE on graphs, PDE on points clouds, non-local processing, graph signal processing, filtring, inpainting.

1 Introduction

La récente prolifération des scanneurs 3D commerciaux a permis d'établir la numérisation 3D comme une réalité pratique dans le domaine de la préservation du patrimoine culturel [SBG11]. Les intérêts de la numérisation 3D pour le patrimoine culturel sont multiples : elle fournit une grande précision de l'objet à numériser, et rend possible la distribution et la consultation massive de l'objet. Le projet Cyark (`www.cyark.org`) est un exemple qui a pour objectif le partage numérique et la préservation des objets culturels du monde.

Les scanneurs 3D génèrent des données sous forme de nuages de points bruts. Ces derniers sont ensuite traités pour produire un maillage de l'objet à numériser. Les nuages de points sont constitués d'un ensemble non organisé et non orienté de points munis de leurs coordonnées (x, y, z). Grâce à un dispositif photométrique additionnel, une couleur peut être associée à chaque point. Par opposition, un maillage 3D est constitué de sommets, d'arêtes et de faces qui peuvent être définies sous la forme d'un polyèdre. Les faces sont typiquement constituées de triangles et une texture peut être associée aux faces pour faciliter la visualisation de l'objet. L'étude de maillages polygonaux a été longtemps un champ d'étude en synthèse d'images et en modélisation géométrique [BKP$^+$10].

En effet, les scanneurs 3D permettent d'abord l'acquisition de données 3D sur forme de nuage de points brut, puis un post-traitement est nécessaire pour convertir l'objet en maillage, typiquement un maillage triangulaire. Cependant, avec l'avancée des scanneurs 3D qui ont permis une diminution de l'erreur d'acquisition, on constate un intérêt croissant dans le traitement de nuage de points brut. Un des buts recherchés dans la numérisation d'objets en 3D est de garder une précision suffisamment optimale permettant de préserver, de mettre en évidence des défauts ou de reconstruire des détails ou des textures de l'objet. L'acquisition de nuage de points brut, par exemple avec des scanneurs de type LIDAR (Light Detection and Ranging), trouve un intérêt dans le patrimoine culturel.

Tous ces éléments ont mené au développement de techniques permettant de traiter directement un nuage de points brut [LEL12, Dig12, RC11]. Notre travail est en lien avec ces récents travaux. Notre but est de proposer un formalisme général pour le traitement morphologique de nuage de points en transposant des EDP sur graphes.

Dans cet article, nous présentons une méthode de restauration de nuage de points 3D brut en se reposant sur le cadre des EdPs pour réaliser des traitements basés morphologiques sur graphes [ELB08]. Tous les traitements effectués dans cet article se reposent sur le cadre des EdPs sur graphe sous forme d'opérateurs morphologiques.
Ici, les données manipulées sont des sommets d'un nuage de points 3D, composés de leur coordonnée 3D et de leur couleur.
L'article est organisé de la façon suivante. Dans la première section, nous présentons le cadre des EdPs sur graphe [ELB08, EDL12] et les opérateurs morphologiques qui seront utilisés pour filtrer et interpoler les nuages de points. La seconde section présente la construction d'un graphe pondéré avec des patches à partir d'un nuage de points. La dernière section présente des exemples de filtrage et d'interpolation sur des objets 3D numérisés.

2 Graphes pondérés

Dans cette section, nous rappelons les définitions et les opérateurs sur graphes. Cela constitue la base du cadre des EdPs sur graphe [ELB08] permettant de transposer des EDPs sur graphes.

2.1 Notations et Préliminaires

Soit un graphe pondéré $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ constitué d'un ensemble fini $\mathcal{V} = \{v_1, \ldots, v_N\}$ de N sommets et d'un ensemble fini $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ d'arêtes pondérées. Nous supposons $\mathcal{G}$ non dirigé, sans boucle, et sans arêtes multiples. Soit (u, v) une arête de $\mathcal{E}$ connectant les sommets u et v de $\mathcal{V}$. Son poids, noté par $w(u, v)$, représente la similarité entre ces sommets. Ces similarités sont calculées en utilisant une fonction symétrique positive $w : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$ satisfaisant $w(u, v) = 0$ si $(u, v) \notin \mathcal{E}$. La notation $u \sim v$ est utilisée pour préciser que les sommets u et v sont adjacents. Le degré d'un sommet v_i est noté $\delta(v_i) = card(v_j \sim v_i)$. $\mathcal{H}(\mathcal{V})$ représente un espace de Hilbert des fonctions à valeur réelle définies sur des sommets du graphe, muni de son produit interne. Une fonction $f : \mathcal{V} \to \mathbb{R}$ de $\mathcal{H}(\mathcal{V})$ assigne une valeur réelle $f(u)$ à chaque sommet $u \in \mathcal{V}$.

2.2 Opérateurs de différences sur graphes pondérés

Soit $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ un graphe pondéré, soit $f : \mathcal{V} \to \mathbb{R}$ une fonction de $\mathcal{H}(\mathcal{V})$ et $w : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$, une fonction de pondéré qui dépend de l'intéraction entre les sommets. L'*opérateur de différence* [ELB08] de f, noté $d_w : \mathcal{H}(\mathcal{V}) \to \mathcal{H}(\mathcal{E})$ est défini sur une arête $(u, v) \in \mathcal{E}$ par :

$$(d_w f)(v_i, v_j) = \sqrt{w(v_i, v_j)}(f(v_j) - f(v_i)). \quad (1)$$

La *dérivée directionnelle* de f, en un sommet $u \in \mathcal{V}$, le long de l'arête $e = (u, v)$, est définie comme :

$$\partial_v f(u) = (d_w f)(u, v). \quad (2)$$

Les opérateurs de *dérivées partielles directionnelles morphologiques* internes et externes sont respectivement définis comme [TEL11] :

$$\partial^+_{v_j} f(v_i) = \left(\partial_{v_j} f(v_i)\right)^+, \quad (3)$$
$$\partial^-_{v_j} f(v_i) = \left(\partial_{v_j} f(v_i)\right)^-. \quad (4)$$

où $(x)^+ = \max(x, 0)$ et $(x)^- = -\min(x, 0)$. Les *gradients discrets non-locaux* internes et externes sont définis par :

$$(\nabla^\pm_w f)(v_i) = \left((\partial^\pm_{v_j} f)(v_i)\right)^T_{v_j \in \mathcal{V}}. \quad (5)$$

Les normes $\mathcal{L}_p$ et $\mathcal{L}_\infty$ de ces gradients sont définies par :

$$||(\nabla^\pm_w f)(v_i)||_p = \left[\sum_{v_j \sim v_i} w^p(v_i, v_j)\left[(f(v_j) - f(v_i))^\pm\right]^p\right]^{\frac{1}{p}} \quad (6)$$

$$||(\nabla^\pm_w f)(v_i)||_\infty = \max_{v_j \sim v_i}\left(\sqrt{w(v_i, v_j)}|(f(v_j) - f(v_i))^\pm|\right). \quad (7)$$

Dans la suite nous considérons uniquement la norme $\mathcal{L}_\infty$. L'$\infty-$Laplacien peut être défini par ces normes par [EDL12] :

$$(\Delta_{w,\infty} f)(v_i) = \frac{1}{2}\left[||(\nabla^+_w f)(v_i)||_\infty - ||(\nabla^-_w f)(v_i)||_\infty||\right]. \quad (8)$$

2.3 Operateurs Morphologiques sur Graphes

La morphologie continue définie les opérateurs de dilatation δ et d'érosion ϵ d'une fonction $f^0 : \mathbb{R}^m \to \mathbb{R}$ par l'utilisation d'éléments structurants convexes $B = \{x : ||x||_p \leq 1\}$. Les opérateurs de dilatation et d'érosion par EDP sont respectivement définis comme :

$$\frac{\partial f}{\partial t} = +||\nabla f||_p \text{ and } \frac{\partial f}{\partial t} = -||\nabla f||_p, \quad (9)$$

où f est une fonction modifiée de f^0, ∇ est l'opérateur gradient, $||\cdot||_p$ correspond à la norme ℓ_p, avec la condition initiale $f = f^0$ à $t = 0$. En faisant varier la variable p, différents éléments structurants sont obtenus : pour $p = \infty$ un losange, pour $p = 2$ un disque, et pour $p = 1$ un carré (voir [BM94]). La solution au temps n donne une dilatation (avec le signe plus), ou une érosion (avec le signe moins) avec un élément structurant de taille $n\Delta t$. Nous avons proposé dans [TEL11] un analogue discret sur graphes des opérateurs morphologiques reposant sur une EDP. La dilatation sur graphe et l'érosition sur graphe sont respectivement obtenus en remplaçant ∇ par $\nabla^\pm_w$ dans Eq.9, tel que :

$$\begin{aligned}\frac{\partial f}{\partial t}(v_i, t) &= +||(\nabla^+_w f)(v_i)||_p \text{ , and} \\ \frac{\partial f}{\partial t}(v_i, t) &= -||(\nabla^-_w f)(v_i)||_p .\end{aligned} \quad (10)$$

Pour le cas $p = \infty$, ces processus morphologiques peuvent être exprimés par un schéma itératif. Par exemple, la dilatation peut être exprimée par :

$$f^{n+1}(v_i) = f^n(v_i) + \Delta t \max_{v_j \sim v_i} \left(\sqrt{w(v_i, v_j)} (f^n(v_j) - f^n(v_i))^+ \right). \tag{11}$$

Pour $\Delta t = 1$ and $w = 1$, on retrouve la formulation algébrique classique de la morphologie mathématique (voir [TEL11] pour plus de détails). Dans cet article, nous nous sommes restreints à $\Delta t = 1$ avec $w \neq 1$. Cela permet d'introduire l'adaptabilité aux données et de fournir une formulation de la morphologie mathématique adaptative sur graphe. Dans les traitements, l'élément structurant B (supposé symétrique) est fourni par la configuration du voisinage local et est exprimé par $B(v_i) = \{v_j \sim v_i\} \cup \{v_i\}$. Dans le cas spécial où $\Delta t = 1$, la dilatation EdP peut être interprétée par un processus de dilatation non-local (NLD), et comme un processus d'érosion non-local (NLE). Ces processus peuvent être exprimés par :

$$\begin{aligned} f^{n+1}(v_i) &= NLD(f^n)(v_i) \\ &= f^n(v_i) + ||(\nabla_w^+ f^n)(v_i)||_\infty \\ &= f^n(v_i) + \max_{v_j \sim v_i} \big(w(v_i, v_j)(f^n(v_j) - f^n(v_i))^+ \big) \end{aligned} \tag{12}$$

pour la dilatation, et

$$\begin{aligned} f^{n+1}(v_i) &= NLE(f^n)(v_i) \\ &= f^n(v_i) - ||(\nabla_w^- f^n)(v_i)||_\infty \\ &= f^n(v_i) + \min_{v_j \sim v_i} \big(w(v_i, v_j)(f^n(v_i) - f^n(v_j))^- \big) \end{aligned} \tag{13}$$

pour l'érosion. De plus, les auteurs de [EDL12] ont montré que l'$\infty-$Laplacien non-local (Eq. 8) dépend de ces opérateurs NLD et NLE :

$$(\Delta_{w,\infty} f)(v_i) = NLA(f)(v_i) - f(v_i), \tag{14}$$

où :

$$NLA(f)(v_i) = \frac{1}{2}[NLD(f)(v_i) + NLE(f)(v_i)]. \tag{15}$$

Cette approche peut être utilisée pour définir d'autres opérateurs morphologiques reposants sur les opérateurs d'érosion ϵ ou de dilatation δ, tel que l'ouverture $\gamma = (\delta\epsilon)$, la fermeture $\phi = (\epsilon\delta)$, des gradients morphologiques $(\delta - \epsilon)$. Par exemple, nous proposons une formulation de l'opération de la fermeture non-locale (NLC), qui peut être définie par :

$$\begin{aligned} \frac{\partial f}{\partial t}(v_i, t) = &- sign^+(t - s + 1)||(\nabla_w^- f)(v_i)||_\infty \\ &+ sign^+(s - t)||(\nabla_w^+ f)(v_i)||_\infty, \end{aligned} \tag{16}$$

avec $t \in [0, 2s[$ et

$$sign^+(x) = \begin{cases} 1 \text{ si } x > 0, \\ 0 \text{ sinon.} \end{cases} \tag{17}$$

Cet EdP possède un coefficient qui dépend du temps et peut agir comme une dilatation δ pour $t \in [0, s[$ ou une érosion ϵ pour $t \in [s, 2s[$. Cette formulation est différent de ceux des EDPs classiques [AGLM93] et ne produit pas de discontinuités aux changements du temps t. Cela peut être interprétée par le processus itératif non-local suivant :

$$\begin{aligned} f^{(n+1)}(v_i) &= NLC(f^{(n)})(v_i) \\ &= sign^+(t - s + 1) NLE(f^{(n)})(v_i) \\ &\quad + sign^+(s - t) NLD(f^{(n)})(v_i). \end{aligned} \tag{18}$$

De façon similaire, on peut exprimé l'ouverture non-locale NLO comme un processus itératif.

2.4 Interpolation et Filtrage sur graphes pondérés

Beaucoup de problèmes en traitements d'images peuvent être formulés comme des problèmes inverses. Dans cette section, nous montrons comment notre formulation de l'$\infty-$Laplacien non-local (Eq.14) peut être utilisée pour deux problèmes inverses : l'interpolation et le filtrage.

Interpolation. Beaucoup de taches en traitement d'image et de vision par ordinateur peuvent être formulées comme des problèmes d'interpolation. L'interpolation de données consiste à déduire de nouvelles valeurs pour des données manquantes en cohérence un ensemble de données connues. Les récents travaux sur l'inpainting tendent à unifier les approches locales et non-locales sous une formulation variationnelle (voir [AFCS11] et les références associées pour plus de détails). Nous avons présenté une approche unifiée pour l'inpainting composé de méthodes géométriques locales et de patches non-locaux [GCE09] en utilisant le cadre de régularisation non-local discret sur graphe introduit dans [ELB08]. Nous considérons que les données sont définies sur un domaine général représenté par un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. Soit $f^0 : \mathcal{V} \to \mathbb{R}$ une fonction. Soit $A \subset \mathcal{V}$ le sous-ensemble de sommets de valeurs inconnues et ∂A le sous-ensemble de sommets à valeurs connus. L'objectif de l'interpolation est de trouver la fonction f^* approximant f^0 dans $\mathcal{V}$ qui résoud le problème suivant :

$$\begin{cases} (\Delta_{w,\infty} f)(v_i) = 0 \quad \forall u \in A, \\ f(u) = f^0(v_i) \quad \forall u \in \partial A. \end{cases} \tag{19}$$

La solution f^* est dite être harmonique infinie [GEL11].

Filtrage. Le filtrage est un problème inverse typique, qui consiste à estimer le vrai signal à partir d'une version bruitée mesurée. Nous introduisons l'utilisation de $\infty-$Laplacien sur des graphes pondérés pour la reconstruction de données. Considérons une fonction $f^0 : \mathcal{V} \to \mathbb{R}$ sur un graphe pondéré $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. f^0 est une observation d'une fonction origine f corrompu par un bruit additif n : $f^0 = f^* + n$. L'objectif de la restauration est de retrouver f^* à partir de sa représentation bruité f^0. Ainsi, la régularisation discrète de f^0 en utilisant l'opérateur $\infty-$Laplacien

non-local consiste à chercher une fonction f^* qui est assez lisse sur $\mathcal{G}$. Cela correspond à considérer :

$$\begin{cases} \dfrac{\partial f}{\partial t} & = \Delta_{w,\infty} f, \\ f(.,0) & = f^0. \end{cases} \tag{20}$$

Algorithmes d'interpolation et de filtrage. Que ce soit pour les problèmes d'interpolation ou de filtrage, les travaux de [EDL12] ont montrés que les deux problèmes ont une unique solution qui peut être obtenue en utilisant l'algorithme itératif suivant :

$$\begin{cases} f^{(0)}(v_i) = f^0(v_i), \\ f^{(n+1)}(v_i) = NLA(f^{(n)})(v_i). \end{cases} \tag{21}$$

La solution est obtenue par l'algorithme itératif ci-dessus et repose sur l'opérateur NLA introduit dans Eq. 15. L'algorithme itératif présenté dans Eq. 21 converge vers une solution unique [EDL12]. L'équation générale décrit une famille de processus de diffusion discrète, paramétrée par la structure du graphe (topologie et fonction de poids w). La modification de la topologie du graphe et des poids du graphe permet de réaliser du filtrage / inpainting local et non-local dans le même cadre des EdPs sur graphes. Pour le cas du filtrage, tous les sommets sont mis à jour en parallèle : $\forall u \in \mathcal{V}$ on applique itérativement Eq. 15, comme montré dans Eq. 21. Pour le cas particulier de l'inpainting, à chaque itération, seule la bordure interne $\partial^- A = \{v_i \in A | \exists v_j \sim v_i, v_j \in \partial A\}$ est mise à jour :

$$\begin{cases} NLA(f)(v_i) = \dfrac{1}{2}[NLD(f)(v_i) + NLE(E)(v_i)] & \forall v_i \in \partial^- A, \\ NLA(f)(v_i) = f^0(v_i) & \forall v_i \in \partial A. \end{cases} \tag{22}$$

À chaque fin d'itération l'ensemble ∂A est mis à jour avec $\partial A^{(n+1)} = \partial A^{(n)} \cup \partial^- A^{(n)}$ et $\partial^- A^{(n+1)}$ est mis à jour avec $\partial A^{(n+1)}$. L'algorithme s'arrête lorsque l'ensemble des sommets à interpoler est vide.

3 Construction d'un graphe pondéré sur un nuage de points

Dans cette section, nous expliquons comment un graphe pondéré reposant sur des patches peut être associé avec un nuage de points. Ceci repose sur trois étapes qui sont détaillées dans la suite de cet article.

3.1 Création d'un graphe à partir d'un nuage de points

La première étape consiste à définir les ensembles $\mathcal{V}$ et $\mathcal{E}$ à partir d'un nuage de points donné. Nous considérons un nuage de points P comme un ensemble de points 3D $\{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \in \mathbb{R}^3$. Il y a plusieurs façons d'associer un nuage de points à un graphe qui encode la proximité entre les points. À chaque point de P nous l'associons à un sommet d'un graphe de proximité $\mathcal{G}$, ceci définit un ensemble de sommets $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$. Ensuite, déterminer l'ensemble des arêtes $\mathcal{E}$ du graphe de proximité $\mathcal{G}$ nécessite de définir des voisins à chaque sommet v_i selon une distance Euclidienne par rapport à sa position 3D $\mathbf{p}_i$. Nous définissons $\mathcal{D}(v_i, v_j) = \|\mathbf{p}_i - \mathbf{p}_j\|_2$ la distance Euclidienne entre deux sommets. Nous considérons le graphe des k plus proches voisins (k-PPV) : $v_j \sim v_i$ si la distance entre $\mathbf{p}_i$ et $\mathbf{p}_j$ est parmi les $k$ième plus petites distances de $\mathbf{p}_i$ à tous les autres points. Pour conclure, la première étape consiste à associé un k-PPV à un nuage de points 3D. La valeur de k sera dénoté k_G pour le graphe $\mathcal{G}$ associé au nuage de points. Pour accélérer l'algorithme du knn, un kD-tree est utilisé [AMN+98].

Une fois le graphe créé, il reste à le pondérer. Si nous ne voulons pas prendre en compte la similarité entre les sommets, la fonction de poids w peut être initialisée à $w = 1$. Une meilleure approche est obtenue en utilisant les patches [BCM10]. Pour les images, un patch $\boldsymbol{\mathcal{P}}(v_i)$ centré en un sommet $v_i \in \mathcal{V}$ est un vecteur de valeurs (e.g., coordonnées, intensitées, ...) défini par $\boldsymbol{\mathcal{P}}(v_i) = \left(f^0(v_j) : v_j \in B(v_i, n)\right)^T$ où $B(v_i, n)$ est un carré de taille n^2 centré en v_i. En utilisant les patches, $w : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ est définie par :

$$w(v_i, v_j) = exp\left(-\frac{||\boldsymbol{\mathcal{P}}(v_i) - \boldsymbol{\mathcal{P}}(v_j)||_2^2}{\sigma^2}\right). \tag{23}$$

3.2 Orientation du patch

Une particularité importante de l'inpainting et la restauration des images reposent sur l'utilisation des patches [BCM10, AFCS11]. Cependant, il n'y a pas d'extension simple à définir un patch sur un nuage de points, que ça soit le contenu ou pour l'orientation. Ainsi, la seconde étape consiste à estimer l'orientation du patch. En effet, vu que deux patches peuvent avoir des orientations très différentes dans le nuage de points, nous avons besoin d'estimer l'orientation pour être capable de comparer les patches.

Dans nos travaux précédents [LEL12], nous avons proposé la stratégie suivante pour estimer l'orientation du patch. Voici un bref rappel de la stratégie proposée. Une analyse en composante connexe (ACP) est était appliqué localement sur les k_n plus proches voisins en chaque point $\mathbf{p}_i$. Cela permet de définir la normale $\mathbf{n}(v_i)$ en chaque point $\mathbf{p}_i$ (associé à chaque sommet v_i). Ensuite, les patches sont orientés à partir des directions principales calculées sur le nuage de points. Cela signifie que les directions du premier et du second axe de la base du patch coïncideront respectivement avec les directions principales majeure et mineure. Pour calculer ces directions principales du point $\mathbf{p}_i$, nous utilisons les arguments de [Dig12] : les directions principales peuvent être estimés à partir des vecteurs propres de l'ACP de la matrice de covariance des normales aux voisins de $\mathbf{p}_i$.

Cependant, cette stratégie n'est pas toujours efficace. En effet, parce que l'orientation des patches est calculée à partir des directions principales, l'orientation obtenue est très

dépendante de la répartition des points dans l'espace 3D. Vu que l'orientation obtenue dépend fortement de l'axe prédominant, on peut trouver différentes orientations de patch pour des répartitions de points similaires et inversement. Ainsi, nous proposons un autre moyen de calculer des patches.

Un autre moyen d'orienter les patches est de déduire une orientation à partir des vecteurs normaux. Par opposition à la méthode des directions principales présentée précédemment, qui dépend de la répartition des points du voisinage, la méthode d'estimation des orientations dépend uniquement des normales en chaque point. En effet, cela produira les mêmes orientations pour des points qui ont des normales similaires. L'algorithme proposé est de déduire en premier le vecteur tangent $\mathbf{t}(v_i)$ à partir du vecteur normal $\mathbf{n}(v_i)$. Soit $\mathbf{x}, \mathbf{y}, \mathbf{z}$ les axes de l'espace 3D, le vecteur tangent $\mathbf{t}(v_i)$ est calcul avec :

$$\begin{cases} \mathbf{t}(v_i) = \mathbf{z} \times \mathbf{n}(v_i) \text{ si } (\mathbf{x} \cdot \mathbf{n}(v_i)) > (\mathbf{z} \cdot \mathbf{n}(v_i)), \\ \mathbf{t}(v_i) = \mathbf{z} \times \mathbf{n}(v_i) \text{ si } (\mathbf{y} \cdot \mathbf{n}(v_i)) > (\mathbf{z} \cdot \mathbf{n}(v_i)), \\ \mathbf{t}(v_i) = \mathbf{x} \times \mathbf{n}(v_i) \text{ sinon}, \end{cases} \quad (24)$$

avec $\times$ l'opérateur produit vectoriel, et $\cdot$ l'opérateur produit scalaire. Ensuite un vecteur bitangent $\mathbf{b}(v_i)$ est calculé par $\mathbf{b}(v_i) = \mathbf{n}(v_i) \times \mathbf{t}(v_i)$. Les orientations des vecteurs $\mathbf{o}_0(v_i), \mathbf{o}_1(v_i), \mathbf{o}_2(v_i)$ sont respectivement assignées à $\mathbf{t}(v_i), \mathbf{b}(v_i), \mathbf{n}(v_i)$.

3.3 Construction du patch

L'étape finale consiste à construire les patches. Étant donné un point $\mathbf{p}_i$, définir un patch pour ce point revient à construire une grille carré autour de $\mathbf{p}_i$ sur son plan tangent. Nous positionnons la longueur du patch l manuellement. Soit n le nombre de cellules sur un ligne du patch. Une grille carrée de n^2 cellules est construite autour de $\mathbf{p}_i$ aligné avec la base de vecteurs obtenue à partir du calcul de l'orientation du patch. Chaque patch a un coé de l/n. Un graphe local est ensuite construit qui connecte le sommet v_i à tous les sommets v_j contenus dans une sphère de diamètre $l\sqrt{2}$. Ensuite, tous les voisins v_j de v_i sont projetés sur le plan tangent de $\mathbf{p}_i$, donnant les points projetés $\mathbf{p}'_i$. Pour remplir le patch avec des valeurs, les points projetés $\mathbf{p}'_i$ sont affectés aux cellules de centre le plus proche. La valeur de la cellule $\boldsymbol{\mathcal{P}}(v_i)$ est ensuite déduite à partir d'une moyenne pondérée des valeurs $f^0(v_j)$ associées aux sommets v_j affectés à la cellule du patch. Cette valeur est une valeur spectrale (la couleur des points). Soit $C_k(v_i)$, la $k$ième cellule du patch construit autour de v_i avec $k \in [1, n^2]$. Avec le processus de construction de patch proposé, on peut définir l'ensemble $V_k(v_i) = \{v_j \mid \mathbf{p}'_j \in C_k(v_i)\}$, comme l'ensemble de sommets v_j qui ont été affectés à la $k$ième cellule du patch de v_i. Ensuite, le vecteur de patch est défini comme $\boldsymbol{\mathcal{P}}(v_i) = \left(\frac{\sum_{v_j \in V_k(v_i)} w(\mathbf{c}_k, \mathbf{p}_j) f^0(v_j)}{|V_k(v_i)|} \right)^T_{k \in [1,n^2]}$, où $w(\mathbf{c}_k, \mathbf{p}_i) = exp(-\frac{||\mathbf{c}_k - \mathbf{p}'_i||_2^2}{\sigma^2})$ où $\mathbf{c_k}$ est les coordonnées du centre du $k$ième cellule du patch. Cette pondération permet de prendre en compte de la répartition des points dans les cellules du patch pour calculer une valeur caractéristique moyenne en chaque cellule du patch.

4 Applications au patrimoine culturel

Cette section montre des exemples de restauration et d'interpolation, avec les filtres morphologiques proposés sur des graphes construits à partir de nuages de points 3D. Il est important de noter que les résultats présentés affichent des nuages de points denses et pas des maillages.

4.1 Opérateurs morphologiques de nuages points colorés

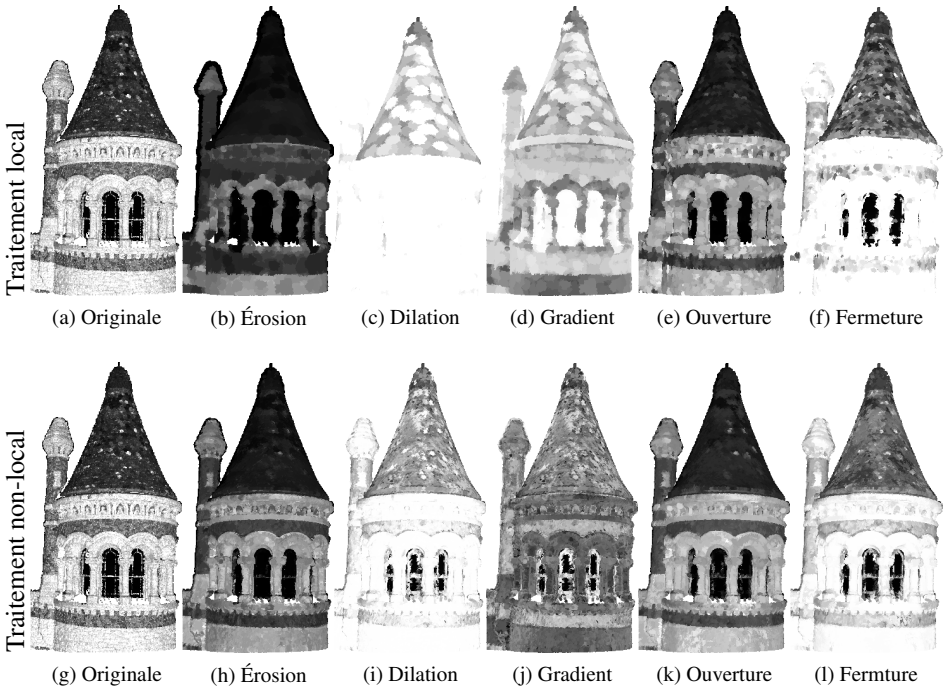
Soit P un nuage de points, qui associe une intensité à chaque point $p \in P$. À partir de ce dernier nuage de points, un graphe pondéré $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ est d'abord créé en utilisant la méthode présentée dans Sec. 3 (i.e., un k_G plus proche voisin graph). Ensuite, ce graphe est filtré en utilisant quelques opérateurs morphologiques, comme expliqué dans Sec. 2.3. Fig. 1 montre quelques traitements morphologiques sur une tour du nuage de points "bishop castle" [1] obtenu avec un scanneur Optech Lidar. Les résultats du filtrage non-local correspond au cas où w est calcul à partir des patches présentés dans Sec. 3.3. Les cellules de chaque patches sont construites à partir de la moyenne des couleurs associées aux voisins des points. Le filtrage local correspond au cas où $w = 1$ avec un petit voisinage, tandis que le filtrage non-local utilise des patches avec un large voisinage. Les résultats dans Fig. 1 montre qu'avec le filtrage non-local les bords sont mieux préservés.

Considérons maintenant un nuage de point bruité. La version filtrée de ce nuage de point peut être obtenue en utilisant l'algorithme itératif présenté dans Eq. 21. Ce dernier algorithme repose sur l'utilisation de l'$\infty-$Laplacien, qui est une combinaison des processus d'érosion et de dilatation. Fig. 2 montre l'utilisation de l'$\infty-$Laplacien pour filtrer le nuage de point "bishop castle" en entier. Nous pouvons voir que l'opérateur $\infty-$Laplacien non-local produit une version simplifiée du nuage de points : les détails correspondants aux hautes fréquences ont été retirés, tandis que les frontières correspondant aux basses fréquences ont été préservées.

4.2 Inpainting couleurs sur nuage de points

Soit P un nuage de points avec quelques couleurs à restaurer. L'ensemble de points à restaurer est noté A et est donné par l'utilisateur. À partir du nuage de points P, un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ est construit en utilisant la méthode présentée dans Sec. 3. La fonction de similarité $w : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ est calculée à partir des patches, et le patch en un sommet $v_i \in \mathcal{V}$ contient la moyenne des couleurs des voisins

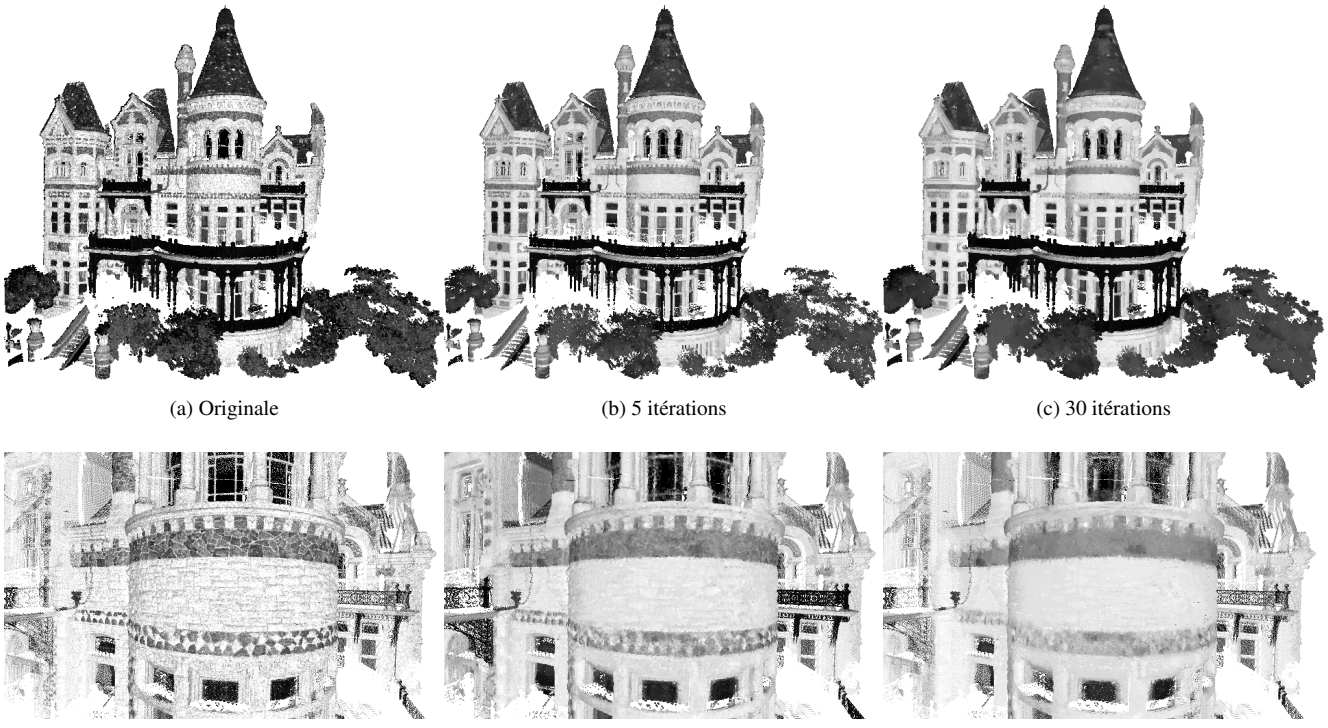
1. fourni gracieusement par `www.cloudcasterlite.com`



(a) Originale (b) Érosion (c) Dilation (d) Gradient (e) Ouverture (f) Fermeture

(g) Originale (h) Érosion (i) Dilation (j) Gradient (k) Ouverture (l) Fermture

FIGURE 1 – Opérateurs morphologiques sur un nuage de points coloré (avec 219 699 points) après 10 itérations avec $k_G = 1000$ et $n^2 = 25$. Filtrage local : $w = 1$ et $k_G = 8$. Filtrage non-local : w est une mesure de similarité entre patches.



(a) Originale (b) 5 itérations (c) 30 itérations

(d) Originale (e) 5 itérations (f) 30 itérations

FIGURE 2 – Opérateur $\infty-$Laplacien non-local sur le nuage de point “bishop castle” (avec 1 815 044 points) après plusieurs itérations avec $k_G = 1000$ et $n^2 = 25$. De haut en bas, visualisation du château en entier, zoom sur un coin de la tour du château.

projetés en v_i. Fig. 3 montre l'interpolation sur une personne réelle numérisée en utilisant l'algorithme présenté dans Eq.21. Une partie du T-shirt a été retiré par l'utilisateur puis interpolé avec succès.

4.3 Inpainting géométrique et couleur sur nuage de points

Beaucoup d'objets numérisés, en patrimoine culturel, possèdent des défauts, typiquement des parties de l'objet sont manquantes. L'objectif de l'inpainting n'est pas seulement de restaurer la couleur, mais aussi de restaurer la géométrie des parties manquantes. Nous proposons ainsi, une méthode pour restaurer ces parties manquantes. Soit $f : \mathcal{V} \rightarrow \mathbb{R}^3$ une fonction qui associe à chaque noeud $v_i \in \mathcal{V}$ ces coordonnées 3D $\mathbf{p}_i$ dans un nuage de point P_1. Un utilisateur délimite manuellement la partie de l'objet qui doit être réparé virtuellement. L'algorithme est composé de deux étapes. Premièrement, le plan tangent à la partie manquante est estimé. La frontière de la partie manquante est projetée sur le plan tangent. L'enveloppe convexe est calculée et des points sont rajoutés de telle façon que l'échantillonnage permet de couvrir une version dilatée de l'enveloppe convexe. Soit P_2, l'ensemble des points générés. Un nuage de points P est ensuite obtenu à partir de l'union de P_2 et du nuage de points initial P_1. Un graphe des plus proches voisins est construit à partir de P. Ce graphe est pondéré avec des patches construits d'une façon différente qu'avec l'inpainting couleur. En effet, pour permettre de générer des points qui reposent correctement sur les parties réelles de l'objet, les patches sont construits à partir de P et remplis avec la distance $||\mathbf{p}_i - \mathbf{p}'_i||_2$. Ensuite, l'algorithme présenté dans Eq.21 est utilisé sur les points de P_2, sauf que les patches (et aussi les poids) sont mis à jour à chaque itération pour permettre la déformation itérative des points de P_2. Fig. 4 montre le processus complet de la restauration virtuelle d'un vase antique cassé. D'abord, le trou est rebouché avec des points pour couvrir la partie manquante, ensuite la couleur de la partie restaurée géométriquement est interpolée comme dans Sec. 4.2.

5 Conclusion

Dans cet article, nous avons montré comment des EDPs morphologiques peuvent être adaptées sur des nuages de points bruts. L'approche proposée repose sur plusieurs composants clés : l'utilisation du cadre des EdPs pour adapter les EDPs sur des graphes, une formulation adaptative des opérateurs morphologiques sur des graphes, une formulation unifiant l'interpolation et le filtrage avec l'opérateur $\infty-$Laplacien, et une méthode spécifique pour construire des patches sur un nuage de points (voir [LEL14, LEL15, TELM15]). Nous avons présenté des résultats qui montrent les avantages de l'approche pour des applications au patrimoine culturel.

Références

[AFCS11] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro. A variational framework for exemplar-based image inpainting. *Int. J. Comput. Vision*, 93(3) :319–347, 2011.

[AGLM93] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and Fundamental Equations of Image Processing. *Arch. Ration. Mech. An.*, 123(3) :199–257, 1993.

[AMN+98] 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*, 45(6) :891–923, November 1998.

[BCM10] A. Buades, B. Coll, and J.-M. Morel. Image denoising methods. a new nonlocal principle. *SIAM Review*, 52(1) :113–147, 2010.

[BKP+10] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy. *Polygon Mesh Processing*. AK Peters, 2010.

[BM94] R. W. Brockett and P. Maragos. Evolution equations for continuous-scale morphological filtering. *IEEE T. Signal. Proces.*, 42(12) :3377–3386, 1994.

[Dig12] J. Digne. Similarity based filtering of point clouds. In *CVPR Workshops*, pages 73 –79, june 2012.

[EDL12] A. Elmoataz, X. Desquesnes, and O. Lézoray. Non-local morphological pdes and p-laplacian equation on graphs with applications in image processing and machine learning. *IEEE J. Sel. Top. Signa.*, 6 :764–779, 2012.

[ELB08] A. Elmoataz, O. Lezoray, and S. Bougleux. Nonlocal discrete regularization on weighted graphs : A framework for image and manifold processing. *IEEE T. Image. Process.*, 17(7) :1047–1060, 2008.

[GCE09] M. Ghoniem, Y. Chahir, and A. Elmoataz. Geometric and texture inpainting based on discrete regularization on graphs. In *ICIP*, pages 1349–1352, 2009.

[GEL11] M. Ghoniem, A. Elmoataz, and O. Lézoray. Discrete infinity harmonic functions : towards a unified interpolation framework on graphs. In *ICIP*, pages 1361–1364, 2011.

[LEL12] F. Lozes, A. Elmoataz, and O. Lézoray. Non-local processing of 3d colored point clouds. In *ICPR*, pages 1968–1971, 2012.

[LEL14] F. Lozes, A Elmoataz, and O. Lezoray. Partial difference operators on weighted graphs for image processing on surfaces and point clouds. *Image Processing, IEEE Transactions on*, 23(9) :3896–3909, Sept 2014.

FIGURE 3 – Objet réel (avec 127,039 points) interpolé avec $k_G = 30000$ et $n^2 = 81$. De gauche à droite, respectivement : homme original (obtenu à partir de `www.cyberware.com`), la partie à interpoler est affichée en jaune, homme interpolé.

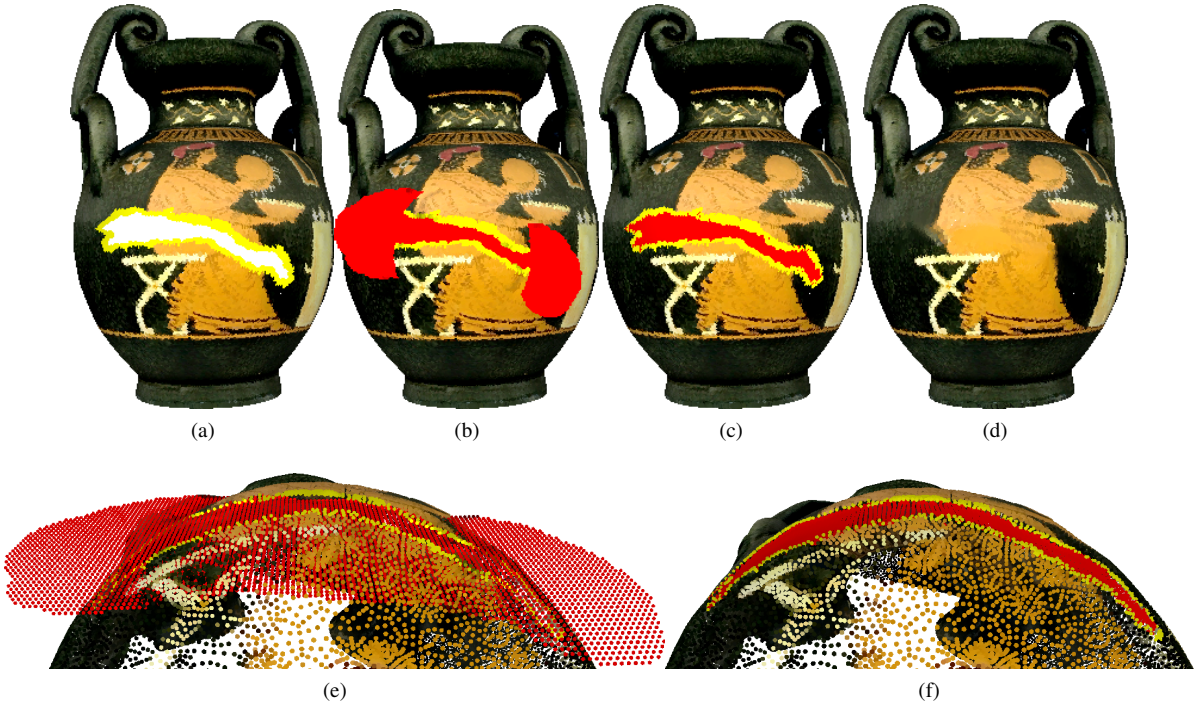


FIGURE 4 – Restauration virtuel d'un vase antique cassé (avec 220,994 points) avec $k_G = 4000$ et $n^2 = 81$. (a) vase cassé avec des parties labélisés pour la restauration, (b) échantillonnage de points sur le plan tangent, (c) résultat du filtrage géométrique, (d) résultat de l'interpolation couleur, (e) zoom sur une partie de (b), (f) zoom sur une partie de (c).

[LEL15] F. Lozes, A. Elmoataz, and O. Lezoray. Pde-based graph signal processing for 3-d color point clouds : Opportunities for cultural heritage. *Signal Processing Magazine, IEEE*, 32(4) :103–111, July 2015.

[RC11] Radu Bogdan Rusu and Steve Cousins. 3d is here : Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.

[SBG11] F. Stanco, S. Battiato, and G. Gallo. *Digital Imaging for Cultural Heritage Preservation : Analysis, Restoration, and Reconstruction of Ancient Artworks*. Digital Imaging and Computer Vision Series. Taylor and Francis, 2011.

[TEL11] V.-T. Ta, A. Elmoataz, and O. Lézoray. Non-local pdes-based morphology on weighted graphs for image and data processing. *IEEE T. Image. Process.*, 26(2) :1504–1516, june 2011.

[TELM15] Matthieu Toutain, Abderrahim Elmoataz, François Lozes, and Alamin Mansouri. Non-local discrete\ infty-poisson and hamilton jacobi equations. *Journal of Mathematical Imaging and Vision*, pages 1–13, 2015.