



HAL
open science

Traitement non-local de nuages de points colorés 3D

François Lozes, Abderrahim Elmoataz, Olivier Lezoray

► **To cite this version:**

François Lozes, Abderrahim Elmoataz, Olivier Lezoray. Traitement non-local de nuages de points colorés 3D. Colloque sur le Traitement du Signal et des Images (GRETSI'13), Sep 2013, Brest, France. pp.ID387. hal-00927581

HAL Id: hal-00927581

<https://hal.science/hal-00927581>

Submitted on 13 Jan 2014

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.

Traitement non-local de nuages de points colorés 3D

François LOZES, Abderrahim ELMOATAZ, Olivier LÉZORAY

Université de Caen Basse-Normandie, GREYC UMR CNRS 6072

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

Résumé – Nous proposons un cadre général pour le traitement de nuages de points colorés 3D. Ce cadre repose sur plusieurs points clés: l'utilisation de graphes pour modéliser les données 3D, une formalisation du traitement reposant sur des équations aux différences partielles sur graphe, et l'introduction de la notion de patch pour des nuages de points 3D. Nous présentons des exemples de débruitage de la couleur ou des coordonnées de nuages de points acquis par des scanners 3D.

Abstract – We propose a method to process colored 3D point clouds. This method is based on several keys points: the use of graphs to represent 3D data, a formalization of the processing with Partial difference Equations (PdEs) on graphs, and the introduction of patches on 3D point clouds. We present exemples of denoising of either coordinate values or color values of point clouds obtained with 3D scanners.

1 Introduction

Les méthodes de traitements non-locales à base de patches ont reçu beaucoup d'intérêts ces dernières années pour le traitement d'images 2D [4]. En effet, celles-ci permettent de mieux préserver les structures fines et les textures présentes dans les images naturelles. Avec l'avènement des scanners 3D, de plus en plus de nuages de points 3D colorés sont disponibles, il semble alors naturel de vouloir disposer de méthodes non-locales pour traiter ces données. Cependant, la notion de patch est difficile à étendre sur un nuage de points 3D. Dans le cadre des images, la notion de patch repose sur une organisation spatiale des pixels sur un grille.

Pour le cas des nuages de points 3D, les données ne sont pas organisées sur une grille Cartésienne, ne permettant pas d'exprimer une notion de voisinage. Ainsi, il n'existe pas d'expression naturelle de la notion de patch sur un nuage de points 3D. Si certains auteurs ont proposé des formulations pour le filtrage non-local de nuages de points [3, 2], celles-ci se restreignent au filtrage des coordonnées des points et ne permettent pas de considérer l'information photométrique de ces points. Dans cet article, nous proposons un cadre général pour le traitement de nuages de points colorés 3D [5]. Ce cadre repose sur : l'utilisation de graphes pour modéliser les données 3D, une formalisation du traitement reposant sur des équations aux différences partielles sur graphe, et l'introduction de la notion de patch pour des nuages de points 3D¹.

2 Graphes pondérés

Soit un graphe pondéré $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ constitué d'un ensemble fini $\mathcal{V} = \{v_1, \dots, v_N\}$ 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 $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$ 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 $\delta(v_i) = \text{card}(v_j \sim v_i)$. $\mathcal{H}(\mathcal{V})$ représente un espace de Hilbert des fonctions à valeur réelle définies sur des sommets du graphe.

Soit $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ un graphe pondéré, soit $f : \mathcal{V} \rightarrow \mathbb{R}$ une fonction de $\mathcal{H}(\mathcal{V})$. Les auteurs de [1] ont défini des opérateurs sur graphes pondérés, tels que l'opérateur de différence, l'opérateur de divergence, le gradient sur graphe pondéré, etc. Soit une fonction générale $f^0 : \mathcal{V} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ définie sur un graphe de topologie arbitraire. La régularisation de cette fonction correspond à un problème d'optimisation qui peut être formalisé par la minimisation d'une énergie représentée par la somme pondérée d'un terme de régularisation avec un terme d'attache aux données :

$$E = \frac{1}{2} \sum_{v_i \in \mathcal{V}} \|\nabla_{\mathbf{w}} \mathbf{f}(v_i)\|_2^2 + \frac{\lambda}{2} \|f - f^0\|_2^2.$$

Les travaux présentés dans [1] ont montré que la solution minimisant cette énergie pouvait être obtenue par l'algorithme itératif suivant :

$$\begin{cases} f^{(0)}(v_i) = f^0(v_i) \\ f^{(n+1)}(v_i) = \frac{\lambda f^0(v_i) + \sum_{v_j \sim v_i} w(v_i, v_j) f^{(n)}(v_j)}{\lambda + \sum_{v_j \sim v_i} w(v_i, v_j)} \end{cases} \quad (1)$$

1. Ces travaux ont été financés par une bourse de thèse du conseil régional de Basse-Normandie.

L'équation 1 décrit une famille de traitements de diffusion sur graphe, paramétrée par la structure du graphe (sa topologie et la fonction de poids w) et le paramètre λ (paramètre de fidélité). Dans la suite nous allons considérer notre nuage de points comme un graphe pondéré où la fonction $w(u, v)$ est une fonction d'interaction entre 2 patchs que nous allons définir par la suite.

3 Traitement de nuages de points 3D

L'approche que nous proposons se décline en quatre étapes : la construction du graphe, l'estimation du plan tangent en chaque sommet, l'orientation puis le calcul du patch.

3.1 Construction du graphe

Considérons un nuage de points P comme un ensemble de vecteurs $\{\mathbf{p}(v_1), \dots, \mathbf{p}(v_n)\} \in \mathbb{R}^3$. Chaque point $p \in P$, est associé à un sommet d'un graphe de proximité \mathcal{G} afin de définir un ensemble de sommets $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$. Ensuite, pour déterminer l'ensemble des arêtes \mathcal{E} du graphe de proximité \mathcal{G} , nous définissons le voisinage de chaque sommet v_i selon l'appartenance à une boule de rayon ϵ et de centre $\mathbf{p}(v_i)$. Nous notons $\mathcal{D}(v_i, v_j) = \|\mathbf{p}(v_i) - \mathbf{p}(v_j)\|_2$, la distance Euclidienne entre sommets et $\mathcal{B}(v_i; r) = \{\mathbf{p}(v_j) \in \mathbb{R}^3 \mid \mathcal{D}(v_i, v_j) \leq r\}$ la boule fermée de rayon r centrée en $\mathbf{p}(v_i)$. Nous considérons deux types de graphes : i) le ϵ -graphe : $v_i \sim v_j$ si $\mathbf{p}(v_j) \in \mathcal{B}(v_i; \epsilon)$, ii) le graphe des k plus proches voisins (k -NNG) : $v_i \sim v_j$ si la distance entre $\mathbf{p}(v_i)$ et $\mathbf{p}(v_j)$ est parmi les k plus proches distances de $\mathbf{p}(v_i)$ aux autres points. La première étape consiste à associer un k -NNG au nuage de points 3D.

3.2 Estimation du plan tangent

L'approximation du plan tangent en chaque sommet résulte d'une ACP de la matrice de covariance des voisins de v_i dans un ϵ -graphe local autour du sommet v_i . Soit $\bar{\mathbf{p}}(v_i)$ le centroïde des voisins de v_i :

$$\bar{\mathbf{p}}(v_i) = \frac{1}{1 + \delta(v_i)} \sum_{v_k \in S(v_i)} \mathbf{p}(v_k)$$

avec $S(v_i) = \{v_j \sim v_i\} \cup \{v_i\}$. La matrice de covariance de la fenêtre locale est

$$\mathbf{C}(v_i) = [\mathbf{p}(v_j) - \bar{\mathbf{p}}(v_i)] \cdot [\mathbf{p}(v_j) - \bar{\mathbf{p}}(v_i)]^T, v_j \sim v_i.$$

À partir de cette matrice, les valeurs propres $\lambda_0 < \lambda_1 < \lambda_2$ et les vecteurs propres $\mathbf{t}_0(v_i), \mathbf{t}_1(v_i), \mathbf{t}_2(v_i)$ sont calculés. Les vecteurs propres $\mathbf{t}_1(v_i)$ et $\mathbf{t}_2(v_i)$ forment une base orthonormée sur le plan tangent, et $\mathbf{t}_0(v_i)$ est la normale au plan tangent.

3.3 Orientation du patch

On oriente la base orthonormée du patch de manière à ce que les directions des premiers et des deuxièmes axes de la base

du patch ($\mathbf{t}_1(v_i)$ et $\mathbf{t}_2(v_i)$) coïncident respectivement avec les directions principales majeure et mineure. Ces dernières sont calculées à partir d'une ACP sur la matrice de covariance des normales des voisins de $\mathbf{p}(v_i)$ projetées sur le plan tangent de $\mathbf{p}(v_i)$. Soit $\mathbf{n}(v_j)$ la normale associée au nœud v_j et

$$\boldsymbol{\xi}(v_j, v_i) = [\mathbf{n}(v_j) \cdot \mathbf{t}_1(v_i); \mathbf{n}(v_j) \cdot \mathbf{t}_2(v_i)]^T,$$

la projection de la normale $\mathbf{n}(v_j)$ sur le plan tangent ($\mathbf{t}_1(v_i), \mathbf{t}_2(v_i)$). Ensuite, la matrice de covariance $\mathbf{C}_n(v_i)$ des normales projetées de tous les voisins en un point donné v_i (dans le ϵ -graphe local) est calculée avec

$$\mathbf{C}_n(v_i) = [\boldsymbol{\xi}(v_j, v_i) - \bar{\boldsymbol{\xi}}(v_i)] \cdot [\boldsymbol{\xi}(v_j, v_i) - \bar{\boldsymbol{\xi}}(v_i)]^T, v_j \sim v_i$$

et $\bar{\boldsymbol{\xi}}(v_i)$ le centroïde des normales projetées :

$$\bar{\boldsymbol{\xi}}(v_i) = \frac{1}{1 + \delta(v_i)} \sum_{v_k \in S(v_i)} \boldsymbol{\xi}(v_k, v_i).$$

À partir de cette matrice, les valeurs propres $\lambda_0 < \lambda_1$ et les vecteurs propres $\mathbf{c}_0(v_i), \mathbf{c}_1(v_i)$ sont calculés. Alors $\mathbf{c}_0(v_i), \mathbf{c}_1(v_i)$ sont respectivement l'estimation des directions principales mineure et majeure.

3.4 Construction du patch

Soit un point $\mathbf{p}(v_i)$, définir un patch en ce point revient à construire une grille carrée autour de $\mathbf{p}(v_i)$ sur le plan tangent. Nous fixons la longueur du patch à $l = 2 \times \max_{v_j \sim v_i} \|\mathbf{p}(v_j) - \mathbf{p}(v_i)\|$. Une grille carrée de n^2 cellules est ensuite construite autour de $\mathbf{p}(v_i)$ selon la base obtenue par les directions principales $\mathbf{c}_0(v_i), \mathbf{c}_1(v_i)$. Ensuite, chaque voisin v_j de v_i est projeté sur le plan tangent de $\mathbf{p}(v_i)$, le point résultant de la projection est noté $\mathbf{p}'(v_j)$. Pour remplir le patch de valeurs, les points projetés $\mathbf{p}'(v_j)$ sont affectés au centre de la cellule la plus proche. La valeur de chaque cellule est ensuite calculée à partir de la moyenne des valeurs $f^0(v_j)$ associées aux sommets v_j des centres des cellules. Cette valeur peut être une valeur spatiale, ou une valeur spectrale (couleurs des points). L'ensemble des valeurs à l'intérieur d'un patch de sommet v_i est noté $\mathcal{P}(v_i)$.

4 Expérimentations

Le cadre proposé est utilisé soit pour filtrer les coordonnées des sommets, soit pour filtrer un nuage de points colorés. Soit $f^0 : \mathcal{V} \rightarrow \mathbb{R}^3$, la fonction associant soit une coordonnée 3D ou une couleur RGB à chaque nœud $u \in \mathcal{V}$. À partir de cette fonction, un graphe $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ est tout d'abord créé. Pour chaque expérimentation, le nombre des plus proches voisins est fixé à 10. Le graphe est pondéré avec

$$w(v_i, v_j) = \exp\left(-\frac{\|\mathbf{F}(f^0, \mathbf{v}_i) - \mathbf{F}(f^0, \mathbf{v}_j)\|^2}{\sigma^2}\right).$$

Si on considère une pondération locale alors $\mathbf{F}(f^0, \mathbf{v}_i) = f^0(v_i)$. Dans le cas des poids non-locaux (donc reposant sur des patchs),

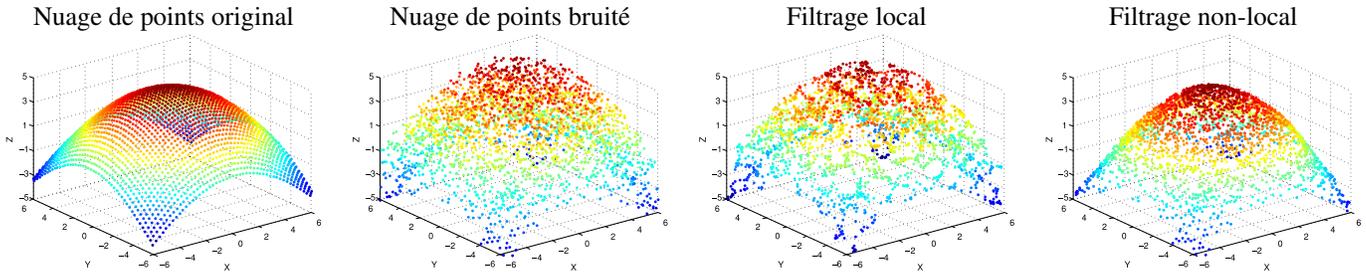


FIGURE 1 – Débruitage des coordonnées d’un nuage de points (en lignes). De gauche à droite, le nuage de points : original, bruité (avec un bruit uniforme), résultat du filtrage local, et le résultat du filtrage non-local (avec des patches 3×3).

nous avons $\mathbf{F}(\mathbf{f}^0, \mathbf{v}_i) = \mathcal{P}(v_i)$. Dans un premier exemple, nous considérons le cas d’un nuage de points 3D où $f^0(v_i) = \mathbf{p}(v_i)$.

La Figure 1 présente des résultats obtenus à partir d’un nuage de points bruité avec un bruit uniforme et compare l’effet de la régularisation locale par rapport au non-local. Le bénéfice obtenu avec la formulation du filtrage non-local est bien mis en évidence : le nuage de points final est beaucoup plus proche de l’original. L’erreur quadratique moyenne pour le filtrage local et le filtrage non-local, sont respectivement 0,65 et 0,49.

Considérons maintenant le cas d’un nuage de points 3D coloré où $f^0(v_i) = [R_i, G_i, B_i]^T$. La Figure 2 présente des résultats de débruitage de la couleur d’un nuage de points 3D coloré. Pour le filtrage local, et non-local, l’erreur quadratique moyenne sont respectivement 1938,95 et 1424,82. Il est important de prendre en compte le fait que l’on travaille sur un nuage de point coloré et non sur un maillage triangulaire, de ce fait des trous peuvent apparaître dans les nuages de points présentés. Comme prévu, le filtrage local tend à retirer beaucoup de détails en ajoutant du flou au niveau des bords. Par opposition, la formulation du filtrage non-local arrive à un meilleur résultat en préservant les contours et en générant des zones uniformes pour les textures similaires.

La figure 3 montre elle-aussi, que de meilleurs résultats sont obtenus avec le filtrage non-local. Pour le filtrage local, et non-local, l’erreur quadratique moyenne sont respectivement 219,04 et 212,81.

5 Conclusion

Dans cet article, nous avons proposé un cadre général pour le traitement de nuages de points colorés 3D. Nous utilisons le formalisme des équations aux différences partielles sur graphe pour transposer et résoudre le problème sur graphe. La méthode est divisée en plusieurs étapes : un graphe est créé à partir du nuage de points à analyser, puis les poids des arêtes sont calculés à partir de patches. Enfin nous procédons à une régularisation sur le graphe. La méthode d’orientation des patches présentée dans cet article dépend fortement de la répartition des voisins en chaque point. Une amélioration possible est de calculer l’orientation des patches à partir des normales uniquement (voir [6]).

Références

- [1] A. Elmoataz, O. Lezoray, et S. Bougleux. *Nonlocal discrete regularization on weighted graphs : a framework for image and manifold processing.* IEEE Trans. Im. Proc., 17(7) :1047–1060, 2008.
- [2] J. Digne. *Similarity based filtering of point clouds.* IEEE Computer Society Conference. CVPR Workshops. 73 - 79, 2012.
- [3] R.-F. Wang, W.-Z. Chen, S.-Y. Zhang, Y. Zhang et X.-Z. Ye. *Similarity-based denoising of point-sampled surfaces.* J. of Zhejiang. Univ. -Science A, 807–815, 2008
- [4] A. Buades, B. Coll et J.-M. Morel. *A Non-Local Algorithm for Image Denoising.* CVPR, 60-65, 2005
- [5] F. Lozes, A. Elmoataz et O. Lézoray. *Nonlocal processing of 3D colored point clouds.* International Conference on Pattern Recognition (ICPR), 2012
- [6] F. Lozes, A. Elmoataz et O. Lézoray. *Morphological PDEs on Graphs for Filtering and Inpainting of Point Clouds,* International Symposium on Image and Signal Processing and Analysis (ISPA), 2013

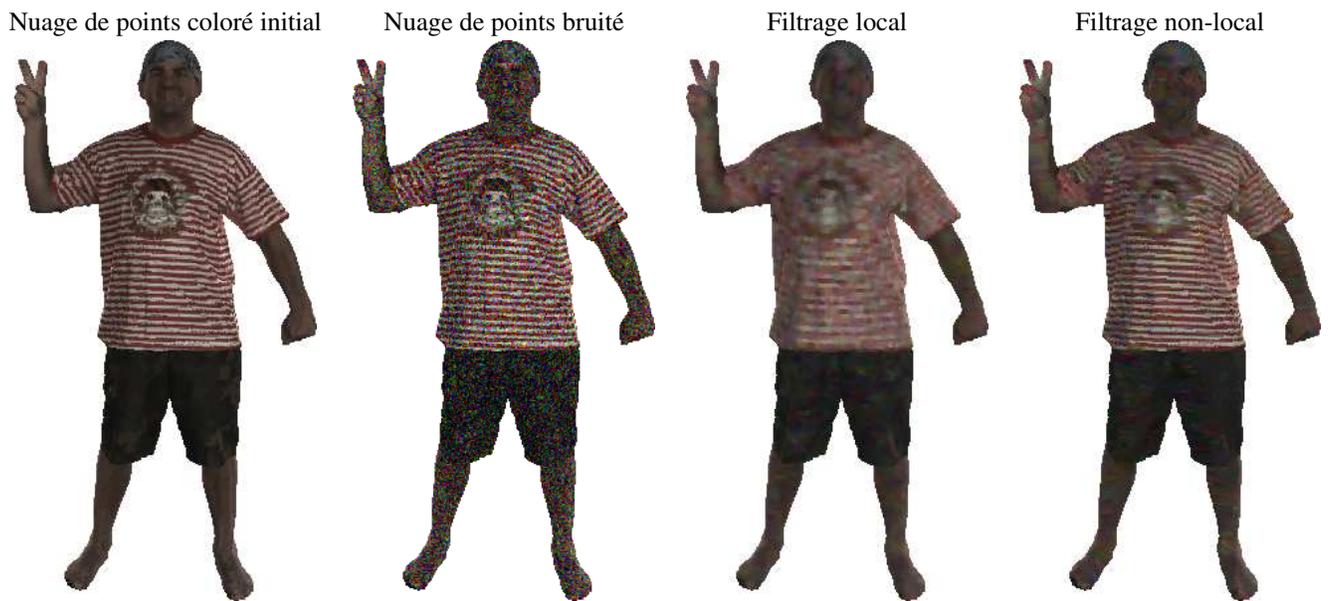


FIGURE 2 – Filtrage local et non-local (avec des patches 9×9) d'un nuage de points 3D colorés d'une personne numérisée avec un scanner 3D et préalablement bruité par un bruit gaussien ($\sigma = 40$).

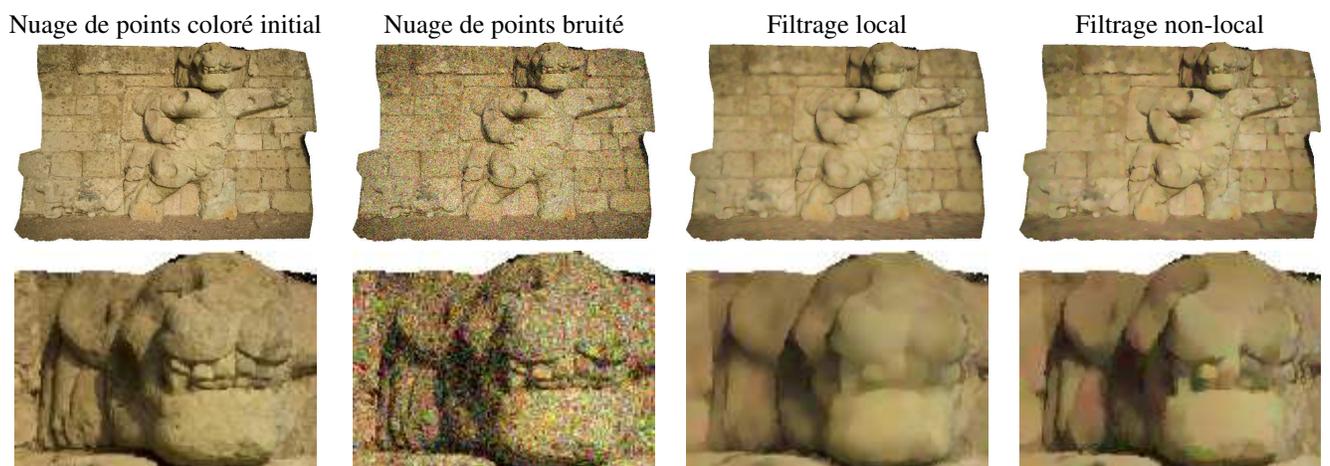


FIGURE 3 – Filtrage local et non-local (avec des patches 3×3) d'un nuage de points 3D colorés représentant le mur d'un temple Maya numérisé avec un scanner 3D et préalablement bruité par un bruit gaussien ($\sigma = 40$). La première ligne présente le nuage de points entier. La seconde ligne affiche un zoom du nuage de points sur une partie spécifique.