Nonlocal Segmentation of Point Clouds with Graphs
François Lozes, Moncef Hidane, Abderrahim Elmoataz, Olivier Lezoray

To cite this version:
François Lozes, Moncef Hidane, Abderrahim Elmoataz, Olivier Lezoray. Nonlocal Segmentation of Point Clouds with Graphs. Global Conference on Signal and Information Processing (GlobalSIP), Graph Signal Processing Symposium, 2013, Dec 2013, Austin, United States. 4 p. hal-01073146

HAL Id: hal-01073146
https://hal.archives-ouvertes.fr/hal-01073146
Submitted on 9 Oct 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.
Nonlocal Segmentation of Point Clouds with Graphs

François Lozes, Moncef Hidane, Abderrahim Elmoataz, Olivier Lézoray

Normandie Univ., UNICAEN, ENSICAEN, GREYC UMR CNRS 6072, Caen, France
{francois.lozes,moncef.hidane,abderrahim.elmoataz,olivier.lezoray}@unicaen.fr

Abstract—In this paper, we propose a nonlocal approach based on graphs to segment raw point clouds as a particular class of graph signals. Using the framework of Partial difference Equations (PdEs), we propose a transcription on graphs of recent continuous global active contours along with a minimization algorithm. To apply it on point clouds, we show how to represent a point cloud as a graph weighted with patches. Experiments show the benefits of the approach on raw colored point clouds obtained from real scans.

I. INTRODUCTION

There is actually much interest in the development of algorithms that enable to process high-dimensional data that reside on the vertices or edges of a graph, referred to as graph signals [1]. With 3D sensors becoming cheaper, more and more applications in natural and applied sciences involve the segmentation of raw 3D colored point clouds. With point clouds, the data to process is not organized on any Cartesian grid and the neighbors of a point have to be defined. Our previous works aim at representing point clouds as weighted graphs to perform nonlocal graph processing [2], [3]. With such a point of view, raw 3D colored point clouds are considered as a specific class of graph signals where a color vector is associated to a point (i.e., a vertex) located in a 3D Euclidean space. In this paper, we propose a new approach for the nonlocal segmentation of point clouds represented by graphs. First, we consider a convex formulation of active contours on graphs, we propose a transcription of graph signals (functions defined on the vertices of a graph) assigning a real value \( f \) to each vertex \( v \in \mathcal{V} \). The \( \mathcal{H}(\mathcal{V}) \) space is endowed with the usual inner product, denoted \( \langle \cdot, \cdot \rangle_{\mathcal{H}(\mathcal{V})} \) in the sequel.

B. Difference Operators on Weighted Graphs

Let \( \mathcal{G} = (\mathcal{V}, \mathcal{E}, w) \) be a weighted graph and \( w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+ \) a weight function that depends on the interactions between the vertices. The difference operator [4], denoted \( d_w : \mathcal{H}(\mathcal{V}) \rightarrow \mathcal{H}(\mathcal{E}) \), is defined for all \( f \in \mathcal{H}(\mathcal{V}) \) and \( (v_i, v_j) \in \mathcal{E} \) by:

\[
(d_w f)(v_i, v_j) = \sqrt{w(v_i, v_j)}(f(v_j) - f(v_i)).
\]

The adjoint of the difference operator, denoted \( d_w^* : \mathcal{H}(\mathcal{E}) \rightarrow \mathcal{H}(\mathcal{V}) \), is the unique linear operator satisfying \( \langle d_w f, H \rangle_{\mathcal{H}(\mathcal{E})} = \langle f, d_w^* H \rangle_{\mathcal{H}(\mathcal{V})} \) for all \( f \in \mathcal{H}(\mathcal{V}) \) and all \( H \in \mathcal{H}(\mathcal{E}) \). Its expression is given by:

\[
(d_w^* H)(v_i) = \sum_{v_j \sim v_i} \sqrt{w(v_i, v_j)}(H(v_j, v_i) - H(v_i, v_j)).
\]

The divergence operator, defined by \(-d_w^*\), measures the net outflow of a function \( H \) at each vertex of the graph. The weighted gradient operator of a function \( f \in \mathcal{H}(\mathcal{V}) \), at a vertex \( v_i \in \mathcal{V} \), is the vector defined by:

\[
(\nabla_w f)(v_i) = ((d_w f)(v_i, v_j))_{v_j \in \mathcal{V}}^T.
\]

The \( \ell_p \) norm of this vector is defined, for \( p \geq 1 \), by:

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

III. SEGMENTATION OF POINT CLOUDS

In this section, we propose a framework for the segmentation of graph signals (functions defined on the vertices of graphs). To perform the segmentation, we consider a convex formulation of active contours on graphs. Starting from a continuous formulation, we show how to transcribe the latter on weighted graphs using the framework of PdEs along with a minimization strategy.
A. Convex Segmentation on graphs

The usual drawback of active contours methods is the existence of local minimizers and hence their sensitivity to the initial condition. A recent method, introduced by Bresson and Chan in [5], [6], proposes to redefine the active contour model into a model which gives global minimizers. In the continuous setting, where images are viewed as functions on a continuous domain $\Omega$, this model is given by:

$$\arg\min_{f,v\in(0,1)} \left\{ \int_{\Omega} \|\nabla f(x)\|dx + \lambda \int_{\Omega} g(f(x))v(x)dx \right\}.$$  \hfill (5)

The function $g$ is a region detector function used to force region intensity statistics priors. The transposition of (5) on graphs is obtained using the PDEs framework [4], [7] leading to:

$$f \in \arg \min_{f,v\in(0,1)} \left\{ \sum_{v_i \in V} \|\nabla w_f(v_i)\|^p + \lambda \sum_{v_i \in V} g(f(v_i)\chi(v_i)) \right\}.$$  \hfill (6)

where $f$ is a labeling function (±1 for labeled vertices and 0 for unlabeled ones) and $f^0$ the signal on the graph (the color vectors associated to the vertices). When $\lambda \neq 0$, this energy can be considered as the nonlocal discrete analogue on graphs of the functional introduced in [6]. We now show how such a minimization can be solved. Problem (6) is non-convex and, as shown in [8] for the continuous analogue, can be reformulated through a convex relaxation. Therefore, a new minimization problem is considered:

$$\hat{f} = \arg \min_{f,v\in(0,1)} \left\{ \sum_{v_i \in V} \|\nabla w_f(v_i)\|^p + \lambda \sum_{v_i \in V} g(f(v_i)\chi(v_i)) \right\}.$$  \hfill (7)

Following the approach in [8], one can show that every level-set of a minimizer of (7) is solution of the original optimization problem (6). As a consequence, to obtain a global solution $\hat{f} : V \to \{0,1\}$ to the problem of (6), one first considers any function $\hat{f} : V \to [0,1]$ that is a solution of (7) and $\hat{f}(v_i) = \chi_S(v_i)$, where $S = \{v_i \in V : \hat{f}(v_i) > \tau\}$ with $\tau \in [0,1]$ and $\chi$ is the indicator function defined by $\chi : V \to \{0,1\}$. For a given vertex, if $v_i \in A$, then $\chi_A(v_i) = 1$ and $\chi_A(v_i) = 0$ otherwise. However, to be able to perform such a minimization approach, one has to show that both parts of the energy (7) do verify the co-area formula. This can be easily shown for the second part of the energy (see [8]). We show now that this is also true for the first part.

B. Perimeters and co-area on graphs

Now we show that there is a relation, for the case of a subgraph, between discrete perimeters on graphs and the co-area formula on graphs.

1) Perimeters on graphs: Let $A$ be a set of connected vertices with $A \subset V$. We denote by $\partial^+ A$ and $\partial^- A$, the external and internal boundary sets of $A$, respectively. The set $A^c = V \setminus A$ is the complement of $A$. For a given vertex $v_i \in V$, one has: $\partial^+ A = \{v_i \in A^c : \exists v_j \in A \text{ with } (v_i, v_j) \in E\}$, $\partial^- A = \{v_i \in A : \exists v_j \in A^c \text{ with } (v_i, v_j) \in E\}$, and $\partial A = \{(v_i, v_j) \in E : \exists v_l \in \partial^+ A \text{ and } v_j \in \partial^- A\}$. Let us consider nonlocal regularization functionals based on weighted total variation on graphs $R_{w,p} : \mathcal{H}(V) \to \mathbb{R}$ of a function $f \in \mathcal{H}(V)$: $R_{w,p}(f) = \sum_{v_i \in V} \|\nabla w_f(v_i)\|^p$ with $0 < p < +\infty$. By replacing $f$ by the indicator function $\chi_A$ in these regularization functionals, one has [7]: $R_{w,p}(\chi_A) = \text{vol}(\partial A) = Per_{w,p}(A) = \text{cut}(A, A^c)$.

2) Co-area formulae on graphs:

To solve the optimization problem (7), we propose to use the level sets of $\hat{f}$ to obtain a global solution $\hat{f}$ on $V$. This expression can be considered as a nonlocal discrete perimeter of the sub-graph $A$. Consequently, when $\lambda = 0$ in (7), it is the discrete analogue to the continuous min-cut of [9].

C. Minimization Algorithm on Weighted Graphs

To solve the optimization problem (7), we propose to use the Chan and Pock algorithm [10] on weighted graphs, in a similar manner as in [11]. Let us consider the following general optimization problem:

$$\min_{x \in X} F(Kx) + G(x),$$  \hfill (8)

where $X$ and $Y$ are two general finite-dimensional vector spaces, and $F \in \Gamma_0(Y)$, $G \in \Gamma_0(X)$ and $K : X \to Y$ a linear operators. The set of all proper convex and lower semicontinuous functions from $H$ to $]-\infty, +\infty[$ is denoted by $\Gamma_0(H)$. Recently, Chambolle and Pock have proposed the following iterative algorithm [10] to solve efficiently (8):

$$\begin{align*}
x^0 &= -x^0 = f, \quad y^0 = 0 \\
y^{n+1} &= \text{prox}_{\gamma F^*}(y^n + \sigma K^{n+1}), \\
x^{n+1} &= \text{prox}_{\gamma G^*}(x^n - \tau K^*y^{n+1}), \\
x^{n+1} &= x^{n+1} + \theta(x^{n+1} - x^n),
\end{align*}$$  \hfill (9)

where $F^*$ is the conjugate of $F$ [12], $K^*$ is the adjoint operator of $K$, and $\text{prox}$ the proximity operator defined as:

$$\text{prox}_{\gamma f}(x) = \arg \min_{y \in Y} \left\{ \frac{1}{2}\|y - x\|^2 \right\}.$$  \hfill (10)

The convergence of algorithm (9) is guaranteed if $\theta = 1$ and $0 < \gamma \sigma L^2 < 1$ where $L = \|K\| = \max_{\|x\| \leq 1} \|Kx\|$. The segmentation problem of (7) is formulated with $F = \|\cdot\|_1$, $K = \nabla w$ and $G = \lambda(\cdot, \cdot g(f^0))$, with $\langle \cdot, \cdot \rangle$ the dot product operator. By replacing $F$, $K$ and $G$, in (9), we can simplify the algorithm. For $y \in Y$, as shown in [11], we have:

$$\text{prox}_{\gamma F^*}(y) = \text{prox}_{\gamma \sigma G}(y) = \text{proj}_{B}(y) = \tilde{y},$$
where \( \bar{y}_{ij} = M(y_{ij}) = \frac{y_{ij}}{\max(1, \sqrt{\sum_{x \in C} y_{ij}})} \) and
\[
i_C = \begin{cases} 0 & \text{for } x \in C \\ +\infty & \text{otherwise,} \end{cases} \tag{11}
\]
and \( B \) is the unitary \( \| \cdot \|_{\infty, 2} \) ball.
For \( x \in X \), we can show that:
\[
\text{prox}_{\tau G}(x) = \arg \min_{y \in \mathcal{Y}} \left\{ \tau \lambda(y, g(f^0)) + \frac{1}{2} \| y - x \|^2 \right\}
= x - \tau \lambda g(f^0).
\tag{12}
\]
Thus the algorithm to solve the segmentation problem (7) is reduced to:
\[
\begin{align*}
x^n_0 = \bar{x}^0 = f, & \quad y^n_0 = 0 \\
y^n_{ij} + 1 = M(y^n_{ij} + \sigma(d_w \bar{x}^n)(v_i, v_j)) \\
x^{n+1} = x^n - \tau - \left( d_w \bar{x}^{n+1}(v_i) - \tau \lambda g(f^n)(v_i) \right) + \theta(x^{n+1} - x^n).
\end{align*}
\tag{13}
\]
This algorithm is parametrized by the structure of the graph (topology and weight function \( w \)), the functions \( f, f^0 \) and \( g(f^0) \), and several parameters (\( \lambda, \tau, \theta \) and \( \sigma \)). On has to note that it is the first time that such a solution is proposed to solve (7) on general weighted graphs.

IV. CONSTRUCTION OF A WEIGHTED GRAPH FROM A POINT CLOUD

In this section, we explain how a weighted graph based on patches can be associated with a point cloud. This relies on three steps that we detail in the sequel.

A. Graph Creation from Data Points

First step consists in defining the sets \( \mathcal{V} \) and \( \mathcal{E} \) from a given point cloud. Let us consider a point cloud \( P \) as a set of data points \( \{p_1, \ldots, p_n\} \subset \mathbb{R}^3 \). To each data point we first associate a vertex of a proximity graph \( \mathcal{G} \) to define a set of vertices \( \mathcal{V} = \{v_1, v_2, \ldots, v_n\} \). Then, determining the edge set \( \mathcal{E} \) of the proximity graph \( \mathcal{G} \) requires defining the neighbors of each vertex \( v_i \) according to its embedding \( p_i \) using the Euclidean distance. We will denote as \( D(v_i, v_j) = \|p_i - p_j\|_2 \) the Euclidean distance between vertices. We consider the \( k \)-Nearest Neighbors Graph (k-NNG); \( v_j \sim v_i \) if the distance between \( p_i \) and \( p_j \) is among the \( k \)-th smallest distances from \( p_i \) to all the other data points. To conclude, the first step consists in associating a k-NNG to the 3D point cloud. The value of \( k \) will be denoted \( k_G \) for the graph \( \mathcal{G} \) associated with the point cloud. To speed up the kNN algorithm, a kD-tree is used. Once the graph has been created, it has to be weighted. If one does not want to take care of the vertices similarities, the weight function \( w \) will be set to \( w = 1 \). A better one can be obtained using patches [13].

For images, a patch \( \mathcal{P}(v_i) \) centered at a vertex \( v_i \in \mathcal{V} \) is a vector of values (e.g., coordinates, intensities, ...) defined by \( \mathcal{P}(v_i) = (f^0(v_j) : v_j \in B(v_i, n)) \) where \( B(v_i, n) \) is a square of size \( n^2 \) centered at \( v_i \). Using patches, \( w: \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R} \) is defined by \( w(v_i, v_j) = \exp(-\frac{\|\mathcal{P}(v_i) - \mathcal{P}(v_j)\|^2}{\sigma^2}) \).

B. Patch Orientation

Patches enable to compute a similarity between two nodes of the graph. It is extensively used with images for inpainting, and restoration. For point clouds, the patch orientation is usually estimated from principal directions computed on a smoothed point cloud [3]. Unfortunately, the obtained orientations of patches are unstable. Because the obtained orientation depends highly on the most predominant axis, one can find different patches orientations for similar points repartitions, and conversely. Recently we have proposed another strategy (see [2]). We have proposed to estimate the patch orientation from the normals. Indeed, this will produce the same orientations for points that have similar normals. The proposed algorithm is therefore to first deduce a tangent vector \( t(v_i) \) from the normal vector \( n(v_i) \). Let \( x, y, z \) be the three axis of a 3D space, the tangent vector \( t(v_i) \) is computed with:
\[
\begin{align*}
t(v_i) = z \times n(v_i) & \text{ if } (x \cdot n(v_i)) > (z \cdot n(v_i)), \\
t(v_i) = z \times n(v_i) & \text{ if } (y \cdot n(v_i)) > (z \cdot n(v_i)), \\
t(v_i) = x \times n(v_i) & \text{ otherwise,}
\end{align*}
\tag{14}
\]
with \( \times \) the cross product operator, and \( \cdot \) the dot product operator. Then a bitangent vector \( b(v_i) = n(v_i) \times t(v_i) \). The orientations vectors \( o_0(v_i), o_1(v_i), o_2(v_i) \) are then respectively assigned to \( t(v_i), b(v_i), n(v_i) \).

C. Patch Construction

Final step consists in constructing the patches. Given a point \( p_i \), defining a patch for this point comes to construct a square grid around \( p_i \), on its tangent plane. We fix the patch length \( l \) manually. Let \( n \) be the number of cells on a row of the patch. A square lattice of \( n^2 \) cells is constructed around \( p_i \) with respect to the basis obtained from orientation computation. Each cell has a side length of \( l/n \). A local graph is then considered that connects the vertex \( v_i \) to all the vertices \( v_j \) contained in a sphere of diameter \( l\sqrt{2} \). Then, all the neighbors \( v_j \) of \( v_i \) are projected onto the tangent plane of \( p_i \), giving rise to projected points \( p'_i \). To fill the patch with values, these projected points \( p'_i \) are associated to the cells the center of which is the closest. The value of the cell is then deduced from a weighted average of the values \( f^0(v_j) \) associated with the vertices \( v_j \) that were affected to the patch cell. This value is a spectral value (the points' colors). The set of values inside the patch of the vertex \( v_i \) are denoted as \( \mathcal{P}(v_i) \). Let \( C_k(v_i) \) denotes the \( k \)-th cell of the constructed patch around \( v_i \) with \( k \in [1, n^2] \). With the proposed patch construction process, one can define the set \( V_k(v_i) = \{v_j \mid p'_j \in C_k(v_i)\} \) as the set of vertices \( v_j \) that were affected to the \( k \)-th patch cell of \( v_i \). Then, the patch vector is defined as \( \mathcal{P}(v_i) = \left( \frac{\sum_{v_j \in V_k(v_i)} o(c_k, p_j) f^0(v_j)}{|V_k(v_i)|} \right)_k \in [1, n^2] \), with
\[
w(c_k, p_j) = \exp(-\frac{\|c_k - p'_j\|^2}{\sigma^2}) \text{ and } c_k \text{ is the coordinates' vector of the } k \text{th patch cell center. This weighting enables to take into account the repartition of the points in the cells’ patch to compute their mean feature vectors.}
This section shows some segmentation results of colored 3D point clouds in two classes using the algorithm presented in (13). The raw data consist in sets of 3D points \( p_i \) (i.e., vertices \( v_i \)) that are associated with CIELAB colors vectors \( f^0 : \mathbb{V} \to \mathbb{R}^3 \). In all our experiments, the parameters are \( \theta = 1, \tau = 1, \sigma = 0.25 / (w_{\text{max}} \times \tau) \) with \( w_{\text{max}} = \max_{v_i \in \mathbb{V}} \delta_w(v_i) \). The shown results are raw 3D colored point clouds and not meshes, thus some holes can appear in the rendering. Figures 1 and 2 show segmentation results on real and artificial examples with \( n = 37,161 \) points and \( n^2 = 9 \). From left to right respectively: colored raw point cloud, the initial labels, the final labeling.

V. EXPERIMENTS

This section shows some segmentation results of colored 3D point clouds. In our experiments, the parameters are \( \theta = 1, \tau = 1, \sigma = 0.25 / (w_{\text{max}} \times \tau) \) with \( w_{\text{max}} = \max_{v_i \in \mathbb{V}} \delta_w(v_i) \). The shown results are raw 3D colored point clouds and not meshes, thus some holes can appear in the rendering. Figures 1 and 2 show segmentation results on real and artificial examples with \( n = 37,161 \) points and \( n^2 = 9 \). From left to right respectively: colored raw point cloud, the initial labels, the final labeling.

ONCLUSION

In this paper, we have proposed a nonlocal approach based on weighted graphs to segment raw point clouds. We used the framework of PDEs to adapt PDEs on graphs, next we solved the optimization problem of segmentation with the Chambolle and Pock iterative algorithm. We have presented results that show the benefits of the approach on real scanned people and artificial points clouds.

REFERENCES


