Segmentation Sémantique à Grande Echelle par Graphe de Superpoints

Loic Landrieu, Martin Simonovsky

Segmentation Sémantique à Grande Échelle par Graphes de Superpoints

Loic Landrieu[1] Martin Simonovsky[2]
[1] Université Paris-Est, LASTIG MATIS IGN, ENSG
[2] Université Paris-Est, Ecole des Ponts ParisTech
loic.landrieu@ign.fr

Résumé

Nous proposons dans cet article une méthode pour la segmentation sémantique de nuages de millions de points basée sur l'apprentissage profond. Nous introduisons une nouvelle structure pour les nuages de points 3D appelée graphe de superpoints (superpoint graph, ou SPG), capable d'encoder de manière compacte l'organisation d'un nuage de points en sous-objets interconnectés. Notre méthode définit un nouvel état de l'art pour la segmentation sémantique de scans LiDAR aussi bien en extérieur ($+11.9$ *et* $+8.8$ *points de mIoU pour les deux ensembles de tests de Semantic3D [13]), ainsi qu'en extérieur (*$+12.4$ *points de mIoU pour les acquisitions S3DIS [2]). Cet article est une traduction de l'article [25].*

Mots Clefs

Segmentation sémantique, nuage de points, LiDAR.

Abstract

We propose a novel deep learning-based framework to tackle the challenge of semantic segmentation of large-scale point clouds of millions of points. We argue that the organization of 3D point clouds as interconnected object parts can be efficiently captured by a structure called superpoint graph (SPG). Our framework sets a new state of the art for segmenting outdoor LiDAR scans ($+11.9$ *and* $+8.8$ *mIoU points for both Semantic3D test sets [13]), as well as indoor scans (*$+12.4$ *mIoU points for the S3DIS dataset [2]). This is a french translation of the article [25].*

Keywords

Semantic segmentation, point cloud, LiDAR.

1 Introduction

La segmentation sémantique de nuages de points 3D de grande taille présente de nombreuses difficultés, dont la plus évidente est la taille des données elle-même. Un autre obstacle important est l'absence de structure aussi bien définie que l'arrangement en grille régulière des images. Ces obstacles sont certainement la raison pour laquelle les réseaux de neurones convolutionnels (CNNs) n'atteignent pas encore sur les nuages de points des performances similaires à celles obtenues pour l'analyse de l'image ou de la voix.

Les premières tentatives d'utiliser l'apprentissage profond pour les nuages de points ont consisté à traduire les architectures convolutionelles utilisées pour la segmentation d'images à la 3D. Par exemple, [5] convertit un nuage de point en une série d'images 2D, dont la segmentation sémantique est projetée sur le nuage de point orignal. SegCloud [40] généralise les convolutions 2D en 3D sur une grille régulière de voxels. Ces structures 2D ou 3D régulières ne sont pas à même de décrire précisément l'organisation particulière des nuages de points 3D, limitant ainsi la performance de ces méthodes.

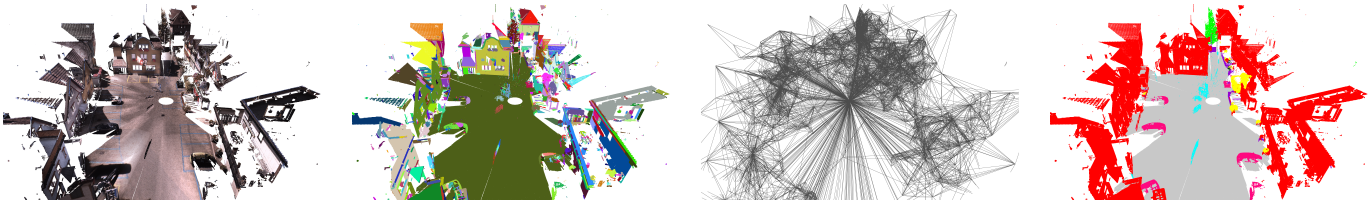
Des architectures profondes récentes, spécifiquement conçues pour l'analyse de points 3D [34, 39, 36, 35, 9], présentent de meilleures performances, mais ne peuvent traiter qu'un nombre limité de points simultanément.

Nous proposons de représenter les nuages de points 3D de grande taille par un ensemble de formes simples interconnectées, que nous appelons *superpoints*. Cette approche transpose les superpixels, populaire pour la segmentation d'images [1] à la 3D. Comme représenté dans la Figure 1, cette structure, appelée graphe de superpoints (SPG), est encodée par un graphe orienté. Les noeuds représentent chacun une forme simple, tandis que les arêtes représentent leur structure d'adjacence. Les arêtes sont enrichie d'attributs charactérisant ces relations d'adjacence.

La représentation d'un nuage de points par un SPG a de nombreux avantages. Premièrement, cela permet de directement classifier des objets (ou partie d'objets), plutôt que des points ou voxels individuellement. Deuxièmement, le SPG décrit richement les relations d'adjacence entre formes, permettant ainsi de prendre en compte finement les informations contextuelles pour la classification. En effet, les voitures sont typiquement situées *au dessus* des routes, tandis que les plafonds sont entourés de murs, etc. Enfin, la taille d'un SPG est définie par le nombre de structures simples dans une scène et non par le nombre de points qu'elle contient, qui est typiquement plusieurs ordres de grandeurs plus grand. Cela permet de mobiliser de puissants outils d'apprentissage profonds, qui ne pourraient pas l'être sur les nuages originaux.

Nos contributions sont les suivantes :

— Nous introduisons le concept de graphe de superpoints, une nouvelle représentation des nuages de points encodant les relations entre partie d'objets par des attributs d'arêtes.



(a) Nuage de points (b) Partition géometrique (c) Graphe de superpoints (d) Segmentation sémantique

FIGURE 1 – Visualisation des différentes étapes de notre méthode. Un nuage de points (a) est partitionné en formes géométriquement simples, appelées superpoints (b). A partir de ce prétraitement, un graphe de superpoint (SPG) est construit en reliant les superpoints adjacents par des arêtes dont des attributs décrivent leur connection (c). Une représentation profonde de chaque superpoint est ensuite calculée puis fournie à un réseau de convolutions modélisant l'information contextuelle, et finalement assignant à chaque superpoint un label sémantique.

— Grâce à cette représentation, nous sommes capables d'appliquer des méthodes d'apprentissage profond sur des nuages de points de très grande tailles tout en conservant leurs détails. Notre réseau est composé de PointNets [34] pour la représentation de chaque forme simple, et d'un réseau de convolution pour la segmentation contextuelle. Ce dernier est muni d'une forme nouvelle de filtrage des entrées (*input gating*) pour les réseaux récurrents à portes (GRU) [7].
— Nous définissons un nouvel état de l'art sur deux jeux de données publics : Semantic3D [13] et S3DIS [2]. Nous améliorons la moyenne de l'indice de Jaccard (mIoU) de 11.9 points pour le jeu de test réduit de Semantic3D, et de 8.8 points pour le jeu complet, et de plus de 12.4 pour S3DIS.

1.1 État de l'Art

L'approche classique de la segmentation sémantique de nuages de points à grande échelle consiste à classifier chaque point individuellement à partir de descripteurs de leur géométrie locale [41]. Cette classification est ensuite régularisée spatialement par des modèles graphiques [32, 21, 38, 19, 33, 30, 42] ou un problème d'optimisation structurée [24]. L'usage de partition comme prétraitement [15, 12] ou post-traitement est aussi employé pour améliorer la précision de la classification. Voir [4] pour une étude des différentes méthodes de partition actuelles.

Méthode d'apprentissage profond pour les nuages de points. Plusieurs approches ont été proposées récemment pour traiter directement les nuages de points 3D. En particulier, on distingue l'approche structurée par ensemble de points [34, 35], par arbres [36, 20], et par graphes [39]. Cependant, aucune de ces méthodes n'est capable de traiter en une fois de grands volumes de points 3D. PointNet [34] a recours à une fenêtre glissante, limitant de fait l'information contextuelle disponible au réseau. [9] y remedie partiellement à l'aide de fenêtres multi-échelles et en prenant en compte le résultat des fenêtres voisines. SEGCloud [40] traite les grands nuages de points en sous-échantillonnant le nuage original, puis en interpolant le résultat à l'aide de champs aléatoires conditionnels (CRF).

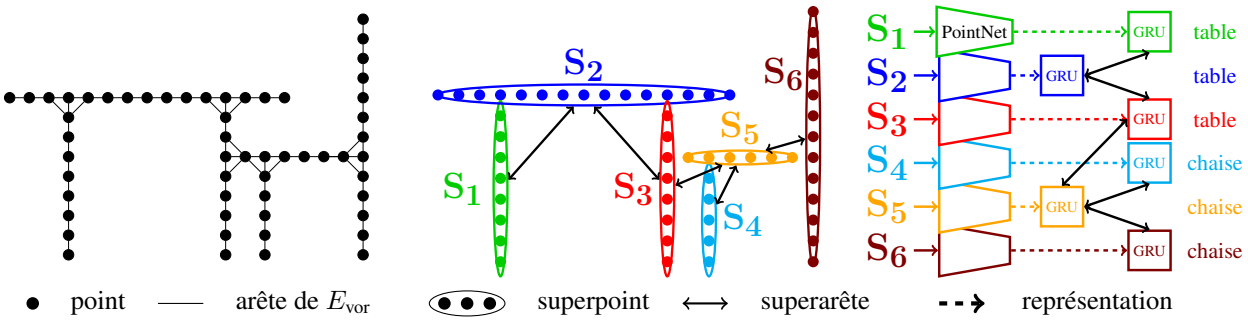
Notre méthode, en partitionant le nuage de manière adaptative à sa complexité locale, est capable simultanément de conserver les détails locaux et de prendre en compte le contexte global.

Réseau de Convolutions. Une étape critique de notre approche est l'utilisation d'un réseau de convolutions pour prendre en compte l'information contextuelle. Un tel réseau doit être capable de considérer des graphes de tailles arbitraires [11], et avec des attributs d'arêtes continus [39, 31]. Notre approche est similaire en pratique à l'approche profonde de l'inférence dans les CRFs [43], à laquelle nous nous comparons dans la Section 2.4.

2 Méthodologie

La principale contrainte que notre méthode tente de résoudre est la taille des scans liDAR, qui peut atteindre des centaines de millions de points, rendant une approche profonde directe impraticable. Notre représentation du nuage à l'aide d'un SPG permet de réduire le problème de la segmentation sémantique en trois problèmes distincts, illustrés par la Figure 2. Chacun de ces problèmes opère à une échelle propre, et peut par conséquent être résolu par une méthode à la complexité adaptée.

1 **Partition Géométriquement Homogène :** La première étape de notre méthode est la partition non-supervisée du nuage en formes simples mais identifiables, appelées superpoints. Cette étape considère le nuage en entrée en entier, et doit donc être particulièrement efficace. Le SPG s'obtient naturellement à partir de cette partition.
2 **Représentation Profonde des Superpoints :** Chaque noeud du SPG coorespond à une petite partie du nuage de point initial, et donc à une forme géométriquement simple, dont on peut supposer l'homogénéité sémantique. De par leur simplicité, de telles formes peuvent être représentées fidèlement par quelques centaines de points seulement, nous autorisant à utiliser le réseau PointNet pour calculer leur représentation [34].
3 **Segmentation Contextuelle :** Le graphe des super-



(a) Nuage de point (b) Graphe de superpoints (c) Réseau de convolution

FIGURE 2 – Illustration de notre méthode sur un scan simulé d'une table et d'une chaise. La partition géométrique est calculée à partir du nuage de points (a), menant à la construction du graphe de superpoints (b). Une représentation de chaque superpoint est calculée par un réseau de type PointNet, puis traitée par un réseau de GRUs par passage de messages le long des superarêtes pour finalement classifier chaque superpoint (c).

points est plus petit de plusieurs ordres de grandeurs que tout graphe qui serait construit à partir du nuage original. Ceci autorise d'employer des méthodes d'apprentissage profond basées sur les réseaux de convolutions, tirant ainsi parti des riches attributs des superarêtes.

Notre méthode permet l'apprentissage de bout en bout des paramètres des deux dernières étapes. Nous décrivons dans la suite de cette section chacune de ces étapes plus en détail.

2.1 Partition Géométriquement Homogène

Dans cette sous-section, nous présentons notre méthode de partitionnement de nuage de points en formes simples. Notre objectif n'est pas de délimiter des objets individuels tels qu'une voiture ou une chaise, mais d'obtenir un découpage en formes simples, comme nous l'illustrons à la Figure 3. En revanche, de par l'hypothèse de simplicité géométrique, les formes retrouvées devraient être sémantiquement homogènes, *i.e.* ne recouvrant pas d'objets de classes distinctes. Nous insistons sur le fait que cette étape est purement non-supervisée et n'utilise uniquement les labels des points que pour la validation.

Nous suivons le modèle de partition par énergie globale présenté par [12] pour sa rapidité, et pour la nature adaptative de la partition par rapport à la complexité géométrique locale. En d'autres termes, les éléments de la partition peuvent aussi bien être de grandes formes simples comme une façade ou une route, que de petites formes représentant les détails d'une chaise ou d'une voiture.

Nous considérons C un ensemble de n points 3D. Chaque point $i \in C$ est défini par sa position p_i, ainsi qu'éventuellement des observations radiométriques o_i (couleur, intensité, etc.). A chaque point nous associons d_g descripteurs géométriques locaux $f_i \in \mathbb{R}^{d_g}$ caractérisant son voisinage. Dans ce papier, nous considérons les trois descripteurs de la dimensionnalité introduits par [8] : linéarité, planarité et volumétrie, ainsi que le descripteur de verticalité, introduit par [12]. Nous associons également à chaque point son élévation, obtenue en normalisant la coordonnée z des points à l'échelle du nuage tout entier.

L'énergie globale proposée par [12] est définie par rapport au graphe d'adjacence obtenu par 10-voisinage $G_{\text{nn}} = (C, E_{\text{nn}})$ sur le nuage de point initial (notons que ce graphe n'est *pas* le SPG). La partition géométriquement homogène est définie par les composantes connexes constantes de la solution du problème d'optimisation suivant :

$$\arg\min_{g \in \mathbb{R}^{dn}} \sum_{i \in C} \|g_i - f_i\|^2 + \mu \sum_{(i,j) \in E_{\text{nn}}} w_{i,j} \left[g_i - g_j \neq 0\right], \tag{1}$$

avec $[\cdot \neq 0]$ la fonction de $\mathbb{R}^{d_g} \mapsto \{0, 1\}$ qui vaut 0 en 0 et 1 ailleurs. Le poids des arêtes est défini comme linéairement décroissant par rapport à leur longueur euclidienne. Le scalaire μ définit la force de régularisation et détermine la finesse de la partition.

L'Equation 1 définit un *problème de partition minimale généralisé*, qui peut être vu comme une version du modèle d'énergie de Potts dans un espace continu, ou comme une variante ℓ_0 de la variation totale sur graphe.

La fonctionelle à minimiser n'est ni convexe, ni continue, rendant ce problème impossible à résoudre exactement pour de grands nuages de points en pratique. En revanche, une solution approchée peut être rapidement obtenue en quelques coupes de graphes avec l'algorithme ℓ_0-*cut pursuit* introduit par [23]. Les composantes connexes constantes $\mathcal{S} = \{S_1, \cdots, S_k\}$ de la solution de Equation 1 définissent notre partition en formes géométriquement simples, que nous appelons *superpoints* (*i.e.* ensemble de points) dans la suite de cet article.

2.2 Construction du Graphe de Superpoints

Le SPG encode une représentation structurée d'un nuage de points, définie par un graphe orienté $\mathcal{G} = (\mathcal{S}, \mathcal{E}, F)$, dont

Descripteur	Taille	Description
offset moyen	3	$\mathrm{mean}_{m\in\delta(S,T)}\ \delta_m$
deviation d'offset	3	$\mathrm{std}_{m\in\delta(S,T)}\ \delta_m$
offset de centroïdes	3	$\mathrm{mean}_{i\in S}\ p_i - \mathrm{mean}_{j\in T}\ p_j$
ratio de longueur	1	$\log \mathrm{length}\,(S)\,/\mathrm{length}\,(T)$
ratios de surface	1	$\log \mathrm{surface}\,(S)\,/\mathrm{surface}\,(T)$
ratio de volume	1	$\log \mathrm{volume}\,(S)\,/\mathrm{volume}\,(T)$
ratio de cardinal	1	$\log |S|/|T|$

TABLE 1 – Liste des $d_f = 13$ descripteurs de superarête characterisant l'adjacence entre deux superpoints S et T.

les noeuds sont l'ensemble des superpoints $\mathcal{S}$, et les arêtes $\mathcal{E}$ (appelées *superarêtes*) représentent leur adjacence. Un ensemble de d_f valeurs : $F \in \mathbb{R}^{\mathcal{E}\times d_f}$ sont associées à chaque superarête, charactérisant la relation d'adjacence.
Nous considérons $G_{\text{vor}} = (C, E_{\text{vor}})$ le graphe d'adjacence des cellules de Voronoi obtenu à partir des positions des points du nuage initial [18]. Deux superpoints S et T sont adjacents dans $\mathcal{G}$ s'il existe au moins une arête de E_{vor} avec une extrémité dans S et l'autre dans T :

$$\mathcal{E} = \left\{(S,T) \in \mathcal{S}^2 \mid \exists\,(i,j) \in E_{\text{vor}} \cap (S \times T)\right\}. \quad (2)$$

Les descripteurs des superarêtes (S,T) sont obtenus notamment à partir de l'ensemble des vecteurs d'*offset* $\delta(S,T)$ définis par les arêtes de E_{vor} liant les deux superpoints :

$$\delta\,(S,T) = \{(p_i - p_j) \mid (i,j) \in E_{\text{vor}} \cap (S \times T)\}. \quad (3)$$

Les autres descripteurs de superarêtes sont obtenus en comparant la forme et la taille des superpoints adjacents. Dans cette perspective, nous calculons pour chaque superpoint $|S|$ le nombre de points le composant, ainsi que les descripteurs de formes $\mathrm{length}\,(S) = \lambda_1$, $\mathrm{surface}\,(S) = \sqrt{\lambda_1\lambda_2}$, $\mathrm{volume}\,(S) = \sqrt[3]{\lambda_1\lambda_2\lambda_3}$ définis à partir des valeurs propres $\lambda_1, \lambda_2, \lambda_3$ de la matrice de covariance des positions des points composant le superpoint, triées par ordre décroissant. Nous listons à la Table 1 les différents descripteurs des superarêtes utilisés dans cet article. Nous attirons l'attention du lecteur sur la dissymétrie de ces descripteurs, qui fait du SPG un graphe orienté.

2.3 Représentation des Superpoints

Le but de cette étape est d'associer à chaque superpoint S_i une représentation vectorielle $\mathbf{z_i}$ de taille d_z. La représentation de chaque superpoint est calculée indépendamment, le contexte n'étant pris en compte qu'à l'étape suivante.
Plusieurs architectures de représentation profondes de nuage de points ont été proposées récemment dans la littérature. Nous choisissons ici PointNet [34] pour sa grande simplicité et son efficacité. Dans PointNet, les nuages de points sont tout d'abord alignés par un réseau de transformation spatial [17], puis chaque point est traité indépendamment par un perceptron multi-couches (MLP), dont les sorties sont ensuite réduites (*pooled*) afin de décrire la forme entière.
Dans notre cas d'application, les nuages à traiter sont des formes géométriquement simples par construction, qui peuvent donc être fidèlement résumées par un petit nombre de points, et être ainsi représentées par un pointNet miniature. Ceci s'avère crucial pour limiter les besoins en mémoire lors du traitement de nombreux superpoints en parallèle. Concrètement, nous sous-échantillonnons les superpoints à la volée en $n_p = 128$ points. Ceci à pour conséquence de limiter la mémoire requise, mais aussi d'augmenter artificiellement la variété des formes apprises. Les points des superpoints avec moins de n_p points sont dupliqués jusqu'à atteindre n_p points, ce qui n'affecte pas leur représentation par PointNet. En revanche, conserver les superpoints constitués de moins de $n_{\text{minp}} = 40$ points réduit la qualité de la prédiction. En conséquence, nous faisons le choix de représenter ces superpoints par un vecteur de zéros, et nous reposons donc entièrement sur leur information contextuelle pour les classifier.
Pour faciliter l'apprentissage par PointNet de la distribution spatiale des différentes formes, nous normalisons le diamètre de chaque superpoint à 1. Chaque point est alors représenté par sa position normalisée p'_i, ses observations radiométriques o_i ainsi que les descripteurs de sa géométrie locale f_i. Finalement, pour que la représentation des formes prenne leur taille en compte, le diamètre des superpoints avant normalisation est ajouté comme descripteur supplémentaire après l'étape de pooling de PointNet

2.4 Segmentation Contextuelle

La dernière étape de notre méthode est la classification des superpoints à partir de leur représentation profonde et de leur voisinage dans le SPG à l'aide d'un réseau de convolutions. Notre approche s'inspire des concepts de Graphes de Réseaux de Neurones à Portes [28] et des convolutions conditionnées par arêtes (ECC) [39]. Le principe général est que les superpoints adaptent leur représentation aux informations reçues par leur superarêtes. En pratique, chaque superpoint S_i dispose d'un état interne caché dans un réseau récurrent à porte (GRU) [7]. Cet état interne, initialisé par la représentation calculée à l'étape précédente, est ensuite modifié lors de plusieurs itérations $t = 1 \dots T$. A chacune de ses itérations t, les GRU calculent leur nouvel état interne $\mathbf{h}_i^{(t+1)}$ à partir de leur état interne courant $\mathbf{h}_i^{(t)}$ et de la synthèse $\mathbf{m}_i^{(t)}$ des messages transmis par leurs voisins. Pour chaque superpoint j, cette synthèse est simplement effectuée en calculant la moyenne des messages transmis par ses voisins, pondérée par la sorties d'un perceptron multi-couches Θ qui prends les descripteurs $F_{ji,\cdot}$ en entrée. Formellement :

$$\begin{aligned} \mathbf{h}_i^{(t+1)} &= (1 - \mathbf{u}_i^{(t)}) \odot \mathbf{q}_i^{(t)} + \mathbf{u}_i^{(t)} \odot \mathbf{h}_i^{(t)} \\ \mathbf{q}_i^{(t)} &= \tanh(\mathbf{x}_{1,i}^{(t)} + \mathbf{r}_i^{(t)} \odot \mathbf{h}_{1,i}^{(t)}) \\ \mathbf{u}_i^{(t)} = \sigma(\mathbf{x}_{2,i}^{(t)} + \mathbf{h}_{2,i}^{(t)}), &\qquad \mathbf{r}_i^{(t)} = \sigma(\mathbf{x}_{3,i}^{(t)} + \mathbf{h}_{3,i}^{(t)}) \end{aligned} \quad (4)$$

$$
\begin{aligned}
(\mathbf{h}_{1,i}^{(t)}, \mathbf{h}_{2,i}^{(t)}, \mathbf{h}_{3,i}^{(t)})^T &= \rho(W_h \mathbf{h}_i^{(t)} + b_h) \\
(\mathbf{x}_{1,i}^{(t)}, \mathbf{x}_{2,i}^{(t)}, \mathbf{x}_{3,i}^{(t)})^T &= \rho(W_x \mathbf{x}_i^{(t)} + b_x)
\end{aligned} \tag{5}
$$

$$
\mathbf{x}_i^{(t)} = \sigma(W_g \mathbf{h}_i^{(t)} + b_g) \odot \mathbf{m}_i^{(t)} \tag{6}
$$

$$
\mathbf{m}_i^{(t)} = \mathrm{mean}_{j|(j,i)\in\mathcal{E}}\ \Theta(F_{ji,\cdot}; W_e) \times \mathbf{h}_j^{(t)} \tag{7}
$$

$$
\mathbf{h}_i^{(1)} = \mathbf{z}_i, \qquad \mathbf{y}_i = W_o(\mathbf{h}_i^{(1)}, \ldots, \mathbf{h}_i^{(T+1)})^T, \tag{8}
$$

avec $\odot$ la multiplication terme à terme, $\sigma(\cdot)$ la fonction sigmoïde, et $W_\cdot, b_\cdot$ des paramètres appris partagés par tout les GRUs. Equation 4 liste les règles standards des GRUs [7], avec $\mathbf{u}_i^{(t)}$ la porte de mise à jour et $\mathbf{r}_i^{(t)}$ la porte de réinitialisation. Nous appliquons une normalisation par couche [3], $\rho(\mathbf{a}) := (\mathbf{a} - \mathrm{mean}(\mathbf{a}))/(\mathrm{std}(\mathbf{a}) + \epsilon)$, à la fois sur les entrées des GRUs $\mathbf{x}_i^{(t)}$ et les états internes $\mathbf{h}_i^{(t)}$ indépendamment, avec ϵ une petite valeur constante fixe. Notre modèle inclut également deux extensions dans les Équations 6 et 8, que nous détaillons ci-dessous.

Filtrage des Entrées : Il est désirable pour les GRUs d'avoir la possibilité de filtrer en partie leurs entrées selon leur état interne. Par exemple, les GRUs pourraient apprendre à ignorer leur contexte quand leur état interne est hautement certain, ou de diriger toute leur attention sur certaines parties des messages, typiquement pour les parties d'objets comme les pieds de table ou de chaise. Ceci est rendu possible par l'Equation 6 en filtrant le message d'entrée $\mathbf{m}_i^{(t)}$ selon l'état interne pour obtenir $\mathbf{x}_i^{(t)}$.

Convolution Conditonnée par Arête. ECC est un élément central de notre méthode, permettant de générer dynamiquement des filtres conditionnés par les attributs des arêtes $F_{ji,\cdot}$ à partir du réseau Θ, comme présenté à l'Equation 8. Ce réseau est partagé par toutes les itérations, et notons également que les boucles auto-référentes qui étaient nécessaires dans [39] ne le sont plus grâce à l'existence de l'état interne des GRUs.

Concatenation des États Internes. Inspirés par DenseNet [16], nous concaténons les états internes de toutes les itérations puis apprenons une transformation linéaire Equation 8 pour obtenir les *logits* $\mathbf{y}_i$ définissant la segmentation. Comme le champ récepteur des GRUs augmente au fur et à mesure des itérations, ceci s'apparente à une analyse multi-échelle du contexte.

Relation avec les CRFs. L'usage des CRFs comme post-traitement des sorties des réseaux convolutionnels est une approche très populaire en segmentation sémantique d'images. Certains algorithmes d'inférence dans ces modèles graphiques peuvent se formuler comme des réseaux de neurones récurrents [43, 37], permettant d'apprendre des paramètres binaires arbitraires [29, 6, 26]. Si ces propriétes sont bien retrouvées dans notre méthodes, elle permet de surcroit d'opérer dans un espace de représentation de dimension d_z, pouvant être potentiellement beaucoup plus grands que celui des distributions de labels par classes [10]. Cela permet ainsi au message de contenir une information plus riche que la simple compatibilité entre classes adjacentes, ce qui s'avère cruciale pour distinguer les parties d'objets.

2.5 Détails d'Implémentation

Voxelisation. Nous effectuons un pré-traitement en sous-échantillonnant le nuage par une grille de voxels régulière, puis en calculant la moyenne des positions et des observations radiométriques par voxel. Ceci a pour conséquence d'accélérer la segmentation, de diminuer son impact mémoire et améliore également la qualité de la segmentation. En effet, en diminuant le bruit géométrique et radiométrique, ce traitement améliore la qualité de la partition, sans toutefois affecter les autres étapes car les superpoints sont eux même déjà fortement sous-échantillonnés pendant leur représentation (voir la Section 2.3).

Apprentissage. Contrairement à l'étape de partition géometrique qui est non-supervisée, la représentation et la classification contextuelle des superpoints sont entraînés ensemble de façon supervisée par l'optimisation d'une seule fonction de perte basée sur l'entropie croisée. Nous assignons à chaque superpoint, supposé par hypothèse sémantiquement homogène, un label unique correspondant au label majoritaire des points qu'il contient. Nous avons également essayé d'associer à chaque superpoint la distribution empirique de ses labels plutôt qu'un unique label, puis d'utiliser une fonction de perte basée sur la comparaison de distributions comme la divergence de Kullback-Leibler [22]. Les résultats s'avérèrent significativement moins bons.

Calculer directement toutes les représentations des superpoints d'un SPG pour un grand nuage de point est problématique au vu des limitations de mémoire des GPUs actuels. Ce problème peut être contourné en ne considérant lors de l'apprentissage qu'une partie du SPG, choisie aléatoirement. En pratique, un ensemble de noeuds est sélectionné ainsi que leur voisinage jusqu'à ce qu'un maximum de 512 superpoints de plus de n_{minp} points soient sélectionnés(les superpoints plus petits n'ont pas de représentation apprise). Notons que de par la haute connectivité typiquement observée dans les SPGs, ces voisinages sont souvent interconnectés, ce qui permet donc d'apprendre les relations contextuelles à grande distance. Cette stratégie a également le bénéfice d'augmenter artificiellement le nombre de configurations de SPGs servant d'exemple à notre réseau. Pour renforcer encore cette régularisation, les superpoints sont aléatoirement pivotés selon l'axe vertical et les descripteurs de points sont altérés par un bruit Gaussien $\mathcal{N}(0, 0.01)$ borné par $[-0.05, 0.05]$.

Inférence. Pour compenser la stochasticité du sous-échantillonnage des superpoints fournis aux PointNets, et comme l'inférence est particulièrement rapide sur GPU, nous faisons la moyenne des logits sur 10 prédictions avec des graines aléatoires distinctes.

Method	OA	mIoU
reduced test set : 78 699 329 points		
TMLC-MSR [14]	86.2	54.2
DeePr3SS [27]	88.9	58.5
SnapNet [5]	88.6	59.1
SegCloud [40]	88.1	61.3
SPG (Ours)	**94.0**	**73.2**
full test set : 2 091 952 018 points		
TMLC-MS [14]	85.0	49.4
SnapNet [5]	91.0	67.4
SPG (Ours)	**92.9**	**76.2**

TABLE 2 – Performance de notre algorithme comparé à l'état de l'art sur Semantic3D. OA est le taux de bonnes classifcations, mIoU et mAcc respectivement la moyenne par classe de l'indice de Jaccard et du rappel.

3 Éxperiences numériques

Notre méthode établit un nouvel état de l'art par une large marge pour les deux plus grands jeux de données publiques de nuages de points, Semantic3D [13] et Stanford Large-Scale 3D Indoor Spaces (S3DIS) [2]. En dépit de leurs natures différentes (intérieur et extérieur), le même modèle parvient à segmenter les deux jeux avec succès. Notre modèle, avec moins de 200000 paramètres, tient sur une carte graphique de 6 GB de mémoire vidéo.

Nous évaluons les performance avec trois mesures : la moyenne par classe de l'indice de Jaccard par classe (mIoU) et du rappel (mAcc), ainsi que le taux de classifications correctes (OA). Ces valeurs sont mesurées sur les points des nuages originaux, et non pas sur les superpoints.

3.1 Semantic3D

Semantic3D [13] est la plus grande base de données annotée de nuage de points, avec plus de 3 milliards de points en couleur acquis sur une variété de scènes urbaines et rurales. Cette base est composée de trois jeux de données : un ensemble d'apprentissage annoté de 15 nuages, un ensemble de test de 15 nuages et un ensemble réduit de 4 nuages. Pour ces deux derniers jeux, les labels ne sont pas communiqués aux participants et la validation se fait uniquement côté serveur.

Nous fournissons nos résultats quantitatifs à la Table 2, ainsi que des illustrations qualitatives à la Figure 3. Notre méthode est considérablement plus précise que l'état de l'art de la segmentation sémantique avec une avance de plus de 12 points de mIoU points sur l'ensemble de test réduit, et de presque 9 points sur l'ensemble complet. En particulier, notre méthode se démarque sur la classe *artefact*. Ceci s'explique par le fait que la partition les individualise facilement de par leur forme particulière, alors que leur voxelisation ou projection en 2D est délicate.

Method	OA	mAcc	mIoU
A5 PointNet [34]	–	48.98	41.09
A5 SEGCloud [40]	–	57.35	48.92
A5 SPG (Ours)	86.38	**66.50**	**58.04**
PointNet [34] in [9]	78.5	66.2	47.6
Engelmann *et al.* [9]	81.1	66.4	49.7
SPG (Ours)	**85.5**	**73.0**	**62.1**

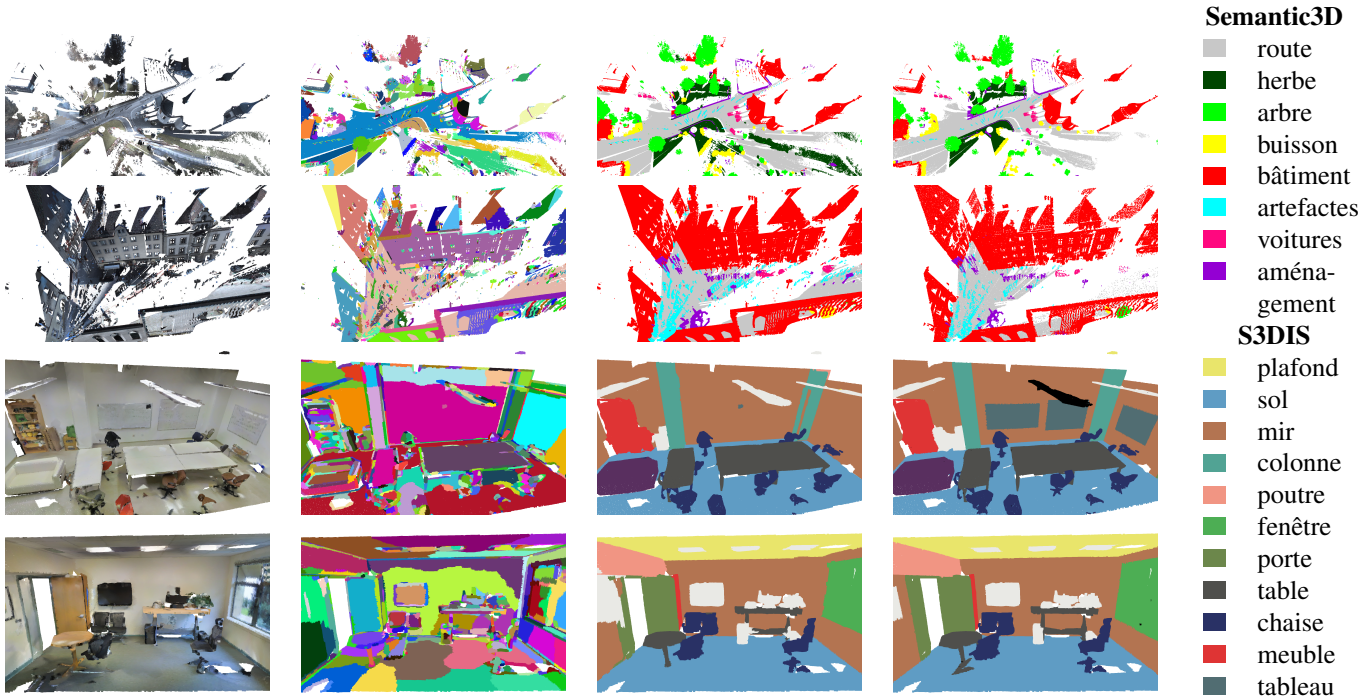
TABLE 3 – Résultats sur S3DIS pour l'étage 5 (haut) et par validation croisée sur les 6 étages.

3.2 Stanford Large-Scale 3D Indoor Spaces

La base de données S3DIS [2] est composée de scans de pièces situées dans six étages de trois bâtiments différents. Nous évaluons notre méthode en suivant deux stratégies de validation utilisées dans la littérature. Comme suggéré par [34, 9], nous calculons la performance globale de la prediction obtenue quand chaque étage est segmenté par un reseau entraîné sur les cinq autres. Comme préconisé par [40], nous donnons également la performance sur l'étage 5 de notre méthode entraînée sur les autres étages. De par la difficulté de distinguer certaines classes purement par leur géométrie (telles que les tableaux sur les murs), nous aggrégeons les mesures radiométriques aux descripteurs géométriques lors du calcul de la partition. Nous présentons nos résultats quantitatifs à la Table 3, et nos illustrations qualitatives à la Figure 3, ainsi qu'à l'adresse `https://youtu.be/vyFrRIFlZu4`.
S3DIS est un jeu de données rendu particulièrement difficile par la présence de classes difficiles à distinguer même en combinant leur géométrie et leur radiométrie, telle que les tableaux blancs sur murs blancs.
Néanmoins, notre méthode améliore significativement l'état de l'art. En particulier, notre méthode est capable de distinguer les portes, tant qu'elle sont ouvertes, de par leur position par rapport aux murs. En revanche, la partition est incapable de distinguer les tableaux des murs, et par conséquent le réseau n'a même pas la possibilité d'apprendre à les distinguer : l'indice de Jaccard (IoU) des tableaux pour la classification théorique qui classeraient parfaitement tous les superpoints est inférieur à 50.
Vitesse d'exécution. La vitesse d'exécution est dépendante du taux de voxelisation effectué en prétraitement. Avec nos choix de paramètres, notre algorithme est capable de traiter les 80 millions de points des 68 pièces de l'étage 5 en moins de 1000s avec un processeur à huit coeurs cadencés à 4Ghz et une Titan Black.

4 Conclusion

Nous avons présenté une méthode d'apprentissage profond pour la segmentation sémantique de grands nuages de points à partir d'une partition en formes simples. Notre approche permet de mobiliser des outils d'apprentissage pro-



(a) Nuage de points RGB (b) Partition géométrique (c) Prédiction (d) Vérité terrain

FIGURE 3 – Illustration de la segmentation sur les duex jeux de données considérés. Les couleurs dans (b) sont choisies aléatoirement pour chaque élément de la partition.

fonds qui n'auraient pas pu gérer le volume considérable des données autrement, et nous permet d'améliorer significativement l'état de l'art sur deux jeux de données publics. Nos perspectives d'amélioration sont centrées sur l'étape de partition, dont l'efficacité et la précision peuvent être augmentées. Le code, ainsi que les modèles pré-entrainés, sont disponibles à l'adresse suivante :`https://github.com/loicland/superpoint_graph`.

Références

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11) :2274–2282, 2012.

[2] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, 2016.

[3] L. J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[4] Y. Ben-Shabat, T. Avraham, M. Lindenbaum, and A. Fischer. Graph based over-segmentation methods for 3d point clouds. *arXiv preprint arXiv :1702.04114*, 2017.

[5] A. Boulch, B. L. Saux, and N. Audebert. Unstructured point cloud semantic labeling using deep segmentation networks. In *Eurographics Workshop on 3D Object Retrieval*, volume 2, 2017.

[6] S. Chandra and I. Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. In *ECCV*, 2016.

[7] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, Oct. 2014.

[8] J. Demantké, C. Mallet, N. David, and B. Vallet. Dimensionality based scale selection in 3d lidar point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-5/W12 :97–102, 2011.

[9] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe. Exploring spatial context for 3d semantic segmentation of point clouds. In *IEEE ICCV, 3DRMS Workshop,*, 2017.

[10] R. Gadde, V. Jampani, M. Kiefel, D. Kappler, and P. Gehler. Superpixel convolutional networks using bilateral inceptions. In *ECCV*, 2016.

[11] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272, 2017.

[12] S. Guinard and L. Landrieu. Weakly supervised segmentation-aided classification of urban scenes

from 3d LiDAR point clouds. In *ISPRS 2017*, 2017.

[13] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys. Semantic3d. net : A new large-scale point cloud classification benchmark. *arXiv preprint arXiv :1704.03847*, 2017.

[14] T. Hackel, J. D. Wegner, and K. Schindler. Fast semantic segmentation of 3d point clouds with strongly varying density. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 3(3), 2016.

[15] H. Hu, D. Munoz, J. A. Bagnell, and M. Hebert. Efficient 3-d scene analysis from streaming data. In *ICRA*. IEEE, 2013.

[16] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, volume abs/1511.05493, 2017.

[17] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *NIPS*, pages 2017–2025. Curran Associates, Inc., 2015.

[18] J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80(9) :1502–1517, 1992.

[19] B.-s. Kim, P. Kohli, and S. Savarese. 3d scene understanding by voxel-CRF. In *ICCV*, 2013.

[20] R. Klokov and V. S. Lempitsky. Escape from cells : Deep kd-networks for the recognition of 3d point cloud models. *CoRR*, abs/1704.01222, 2017.

[21] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *NIPS*, pages 244–252, 2011.

[22] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22 (1) :79–86, 1951.

[23] L. Landrieu and G. Obozinski. Cut pursuit : fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM Journal on Imaging Sciences*, 10(4) :1724–1766, 2017.

[24] L. Landrieu, H. Raguet, B. Vallet, C. Mallet, and M. Weinmann. A structured regularization framework for spatially smoothing semantic labelings of 3d point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 132 :102 – 118, 2017.

[25] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[26] M. Larsson, F. Kahl, S. Zheng, A. Arnab, P. H. S. Torr, and R. I. Hartley. Learning arbitrary potentials in CRFs with gradient descent. *CoRR*, abs/1701.06805, 2017.

[27] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg. Deep projective 3d semantic segmentation. *arXiv preprint arXiv :1705.03428*, 2017.

[28] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel. Densely connected convolutional networks. In *ICLR*, volume abs/1511.05493, 2016.

[29] G. Lin, C. Shen, A. van den Hengel, and I. D. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016.

[30] A. Martinovic, J. Knopp, H. Riemenschneider, and L. Van Gool. 3d all the way : Semantic segmentation of urban scenes from start to end in 3d. In *CVPR*, 2015.

[31] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, pages 5425–5434, 2017.

[32] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert. Contextual classification with functional max-margin markov networks. In *CVPR*. IEEE, 2009.

[33] J. Niemeyer, F. Rottensteiner, and U. Soergel. Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87 :152–165, 2014.

[34] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet : Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv :1612.00593*, 2016.

[35] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++ : Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017.

[36] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet : Learning deep 3d representations at high resolutions. In *CVPR*, 2017.

[37] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *CoRR*, abs/1503.02351, 2015.

[38] R. Shapovalov, D. Vetrov, and P. Kohli. Spatial inference machines. In *CVPR*, 2013.

[39] M. Simonovsky and N. Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *CVPR*, 2017.

[40] L. P. Tchapmi, C. B. Choy, I. Armeni, J. Gwak, and S. Savarese. Segcloud : Semantic segmentation of 3d point clouds. *arXiv preprint arXiv :1710.07563*, 2017.

[41] M. Weinmann, A. Schmidt, C. Mallet, S. Hinz, F. Rottensteiner, and B. Jutzi. Contextual classification of point cloud data by exploiting individual 3d neighborhoods. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W4 :271–278, 2015.

[42] D. Wolf, J. Prankl, and M. Vincze. Fast semantic segmentation of 3d point clouds using a dense crf with learned parameters. In *ICRA*. IEEE, 2015.

[43] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.