

# Segmentation sémantique de grands nuages de points avec Superpoint Graph

FOSS4G 2018

Loic Landrieu

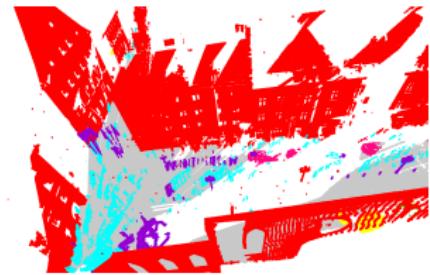
MATIS - IGN

Mai 2018

# Segmentation sémantique à grande échelle

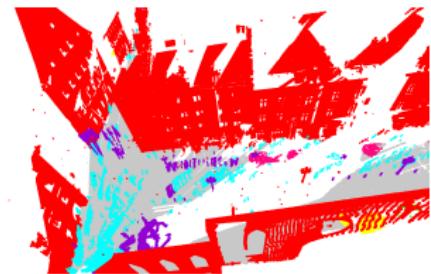
---

- **Objectif:** prédiction dense, un label par point 3D



# Segmentation sémantique à grande échelle

- **Objectif:** prédiction dense, un label par point 3D
- **Difficultés:** nuages de très grandes tailles, structure complexe



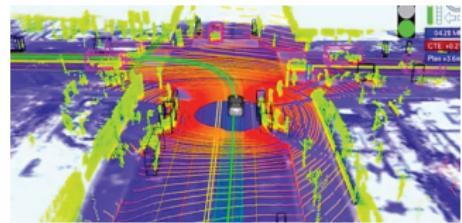
# Segmentation sémantique à grande échelle

- **Objectif:** prédiction dense, un label par point 3D
- **Difficultés:** nuages de très grandes tailles, structure complexe
- **Acquisitions:** LiDAR fixe, mobile, drone, aérien, intérieur, extérieur



# Segmentation sémantique à grande échelle

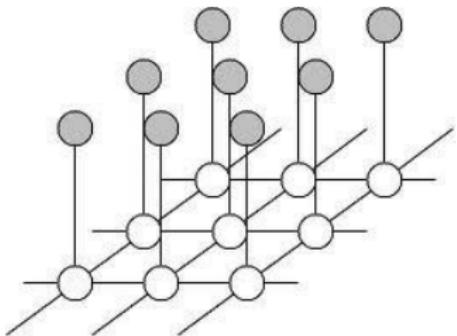
- **Objectif:** prédiction dense, un label par point 3D
- **Difficultés:** nuages de très grandes tailles, structure complexe
- **Acquisitions:** LiDAR fixe, mobile, drone, aérien, intérieur, extérieur
- **Applications:** conduite autonome, maquette virtuelle, reconstruction 3D, occupation des sols



# Etat de l'art

---

- Approche traditionnelle: prédition point par point + regularisation



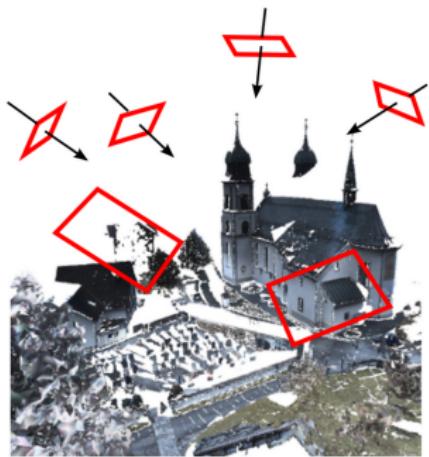
# Etat de l'art

---

- Approche traditionnelle: prédition point par point + regularisation
- La révolution Deep Learning:

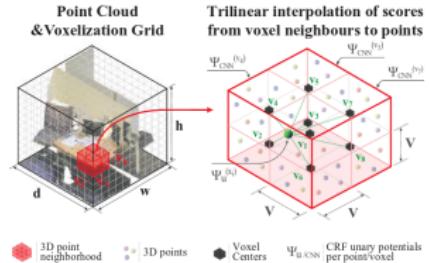
# Etat de l'art

- Approche traditionnelle: prédiction point par point + régularisation
- La révolution Deep Learning:
  - SnapNet: **efficace** mais  $3D \rightarrow 2D$



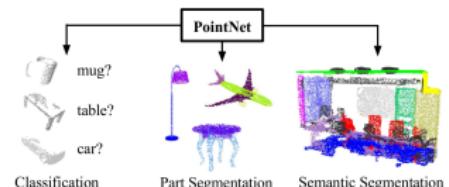
# Etat de l'art

- Approche traditionnelle: prédition point par point + régularisation
- La révolution Deep Learning:
  - SnapNet: efficace mais  $3D \rightarrow 2D$
  - SegCloud: efficace mais perd détails



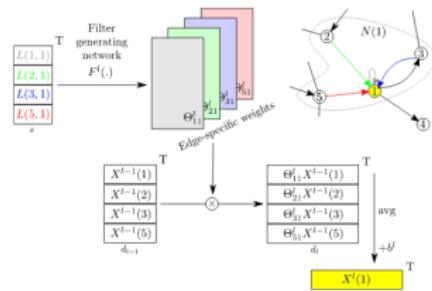
# Etat de l'art

- Approche traditionnelle: prédiction point par point + régularisation
- La révolution Deep Learning:
  - SnapNet: efficace mais  $3D \rightarrow 2D$
  - SegCloud: efficace mais perd détails
  - PointNet: précis mais coûteux



# Etat de l'art

- Approche traditionnelle: prédition point par point + regularisation
- La révolution Deep Learning:
  - SnapNet: efficace mais  $3D \rightarrow 2D$
  - SegCloud: efficace mais perd détails
  - PointNet: précis mais coûteux
  - ECC: structuré mais coûteux



# Principe de SPG

---

- **Observation:** les méthodes les plus précises ne sont pas applicables pour les grands nuages de points

# Principe de SPG

---

- **Observation:** les méthodes les plus précises ne sont pas applicables pour les grands nuages de points
- **Idée:** découper le nuage de point en formes simples

# Principe de SPG

---

- **Observation:** les méthodes les plus précises ne sont pas applicables pour les grands nuages de points
- **Idée:** découper le nuage de point en formes simples
- **Avantages:**

# Principe de SPG

---

- **Observation:** les méthodes les plus précises ne sont pas applicables pour les grands nuages de points
- **Idée:** découper le nuage de point en formes simples
- **Avantages:**
  - Formes simples plus faciles à classifier que des points

# Principe de SPG

---

- **Observation:** les méthodes les plus précises ne sont pas applicables pour les grands nuages de points
- **Idée:** découper le nuage de point en formes simples
- **Avantages:**
  - Formes simples plus faciles à classifier que des points
  - Relations entre formes plus riches qu'entre points

# Principe de SPG

---

- **Observation:** les méthodes les plus précises ne sont pas applicables pour les grands nuages de points
- **Idée:** découper le nuage de point en formes simples
- **Avantages:**
  - Formes simples plus faciles à classifier que des points
  - Relations entre formes plus riches qu'entre points
  - Beaucoup moins de formes que de points

# Principe de SPG

---

- **Observation:** les méthodes les plus précises ne sont pas applicables pour les grands nuages de points
- **Idée:** découper le nuage de point en formes simples
- **Avantages:**
  - Formes simples plus faciles à classifier que des points
  - Relations entre formes plus riches qu'entre points
  - Beaucoup moins de formes que de points
  - Permet de modéliser les interactions à longues distances

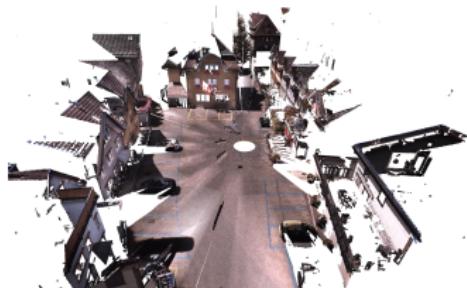
# Principe de SPG

---

- **Observation:** les méthodes les plus précises ne sont pas applicables pour les grands nuages de points
- **Idée:** découper le nuage de point en formes simples
- **Avantages:**
  - Formes simples plus faciles à classifier que des points
  - Relations entre formes plus riches qu'entre points
  - Beaucoup moins de formes que de points
  - Permet de modéliser les interactions à longues distances

Landrieu Loic, and Martin Simonovsky. *Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs*. CVPR 2018

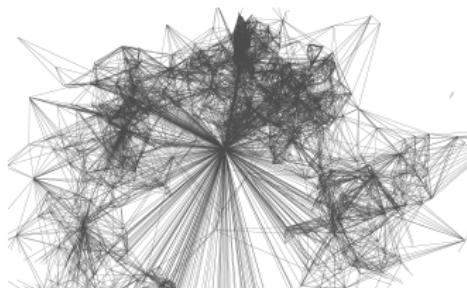
# Etapes de superpoint graph



(a) Nuage de points



(b) Formes simples



(c) Superpoint graph



(d) Segmentation

# Les 3 sous-problèmes

---

- Segmentation semantique réduite à 3 sub-problèmes:

# Les 3 sous-problèmes

---

- Segmentation semantique réduite à 3 sub-problèmes:
  - **Partition géométrique** : en formes simples.  
Complexité: très hautes (nuages de  $10^8$  points)  
Algorithme:  $\ell_0$ -cut pursuit

# Les 3 sous-problèmes

---

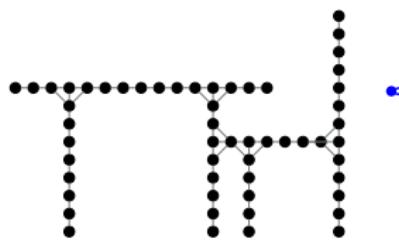
- Segmentation semantique réduite à 3 sub-problèmes:
  - **Partition géométrique :** en formes simples.  
Complexité: très hautes (nuages de  $10^8$  points)  
Algorithme:  $\ell_0$ -cut pursuit
  - **Représentation de formes:** apprentissage de descripteurs  
Complexité: très basse (formes subsampliée à 128 points)  
Algorithme: PointNet

# Les 3 sous-problèmes

---

- Segmentation semantique réduite à 3 sub-problèmes:
  - **Partition géométrique :** en formes simples.  
Complexité: très hautes (nuages de  $10^8$  points)  
Algorithme:  $\ell_0$ -cut pursuit
  - **Représentation de formes:** apprentissage de descripteurs  
Complexité: très basse (formes subsampliée à 128 points)  
Algorithme: PointNet
  - **Segmentation contextuelle:** utilisation du contexte  
Complexité: faible (superpoint graph  $\sim 1000$  noeuds)  
Algorithme: ECC avec Gated Recurrent Unit (GRU)

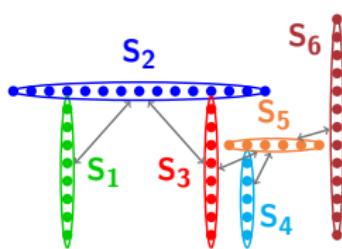
# Les 3 sous-problèmes



point



arête de Voronoi



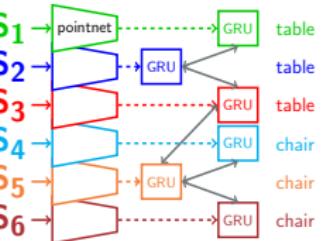
superpoint



superarête

(a) Nuage de point

(b) Superpoint graph



représentation

(c) Réseau

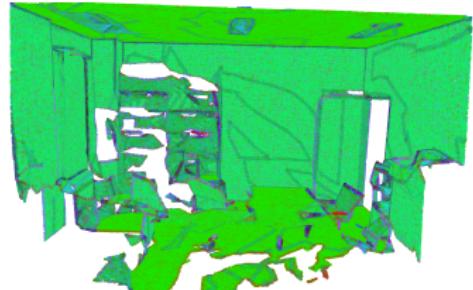
# Partition géométrique

- Nuage de points  $C$ , position  $p_i$ ,  
RGB:  $o_i$



# Partition géométrique

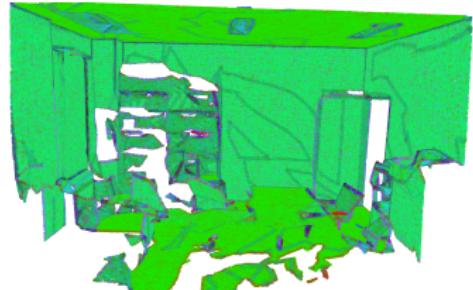
- Nuage de points  $C$ , position  $p_i$ ,  
RGB:  $o_i$ ;
- **Step1:** Descripteurs  
géométriques  $f_i$ ;



linearity, planarity, verticality

# Partition géométrique

- Nuage de points  $C$ , position  $p_i$ ,  
RGB:  $o_i$ ;
- **Step1:** Descripteurs  
géométriques  $f_i$ ;

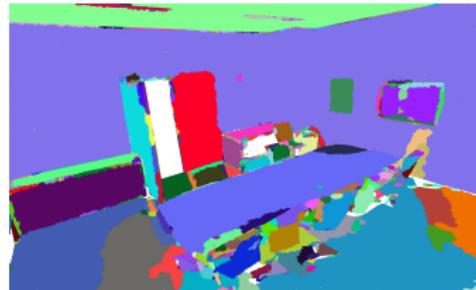


linearity, planarity, verticality

# Partition géométrique

- Nuage de points  $C$ , position  $p_i$ ,  
RGB:  $o_i$
- **Step1:** Descripteurs  
géométriques  $f_i$
- **Step3:** Approximation  
constantes par morceaux  
( $\ell_0$ -cut pursuit):

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

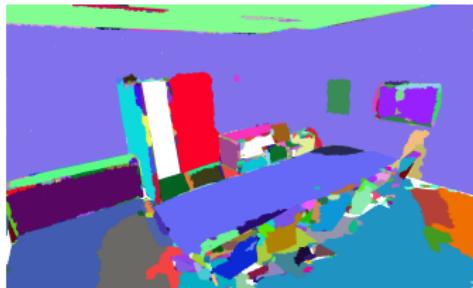


# Partition géométrique

- Nuage de points  $C$ , position  $p_i$ ,  
RGB:  $o_i$
- **Step1:** Descripteurs  
géométriques  $f_i$
- **Step3:** Approximation  
constantes par morceaux  
( $\ell_0$ -cut pursuit):

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

- Composantes constantes:  
formes simples.



# Construction du Superpoint Graph

---

- Superpoint Graph  $\mathcal{G} = (\mathcal{S}, \mathcal{E}, F)$

# Construction du Superpoint Graph

---

- Superpoint Graph  $\mathcal{G} = (\mathcal{S}, \mathcal{E}, F)$
- Superpoints  $\mathcal{S}$ : formes simples

# Construction du Superpoint Graph

---

- Superpoint Graph  $\mathcal{G} = (\mathcal{S}, \mathcal{E}, F)$
- Superpoints  $\mathcal{S}$ : formes simples
- Superedges  $\mathcal{E}$ : relation d'adjacence entre superpoints

# Construction du Superpoint Graph

---

- Superpoint Graph  $\mathcal{G} = (\mathcal{S}, \mathcal{E}, F)$
- Superpoints  $\mathcal{S}$ : formes simples
- Superedges  $\mathcal{E}$ : relation d'adjacence entre superpoints
- $G_{\text{vor}} = (C, E_{\text{vor}})$  graphe de Voronoi

# Construction du Superpoint Graph

- Superpoint Graph  $\mathcal{G} = (\mathcal{S}, \mathcal{E}, F)$
- Superpoints  $\mathcal{S}$ : formes simples
- Superedges  $\mathcal{E}$ : relation d'adjacence entre superpoints
- $G_{\text{vor}} = (C, E_{\text{vor}})$  graphe de Voronoi
- $F$ : descripteurs des superedges:

Descripteurs	Taille	Description
mean offset	3	$\text{mean}_{m \in \delta(S, T)} \delta_m$
offset deviation	3	$\text{std}_{m \in \delta(S, T)} \delta_m$
centroid offset	3	$\text{mean}_{i \in S} p_i - \text{mean}_{j \in T} p_j$
length ratio	1	$\log \text{length}(S) / \text{length}(T)$
surface ratio	1	$\log \text{surface}(S) / \text{surface}(T)$
volume ratio	1	$\log \text{volume}(S) / \text{volume}(T)$
point count ratio	1	$\log  S  /  T $

# Représentation de formes simples

---

- Chaque forme est subsamplée à 128 points

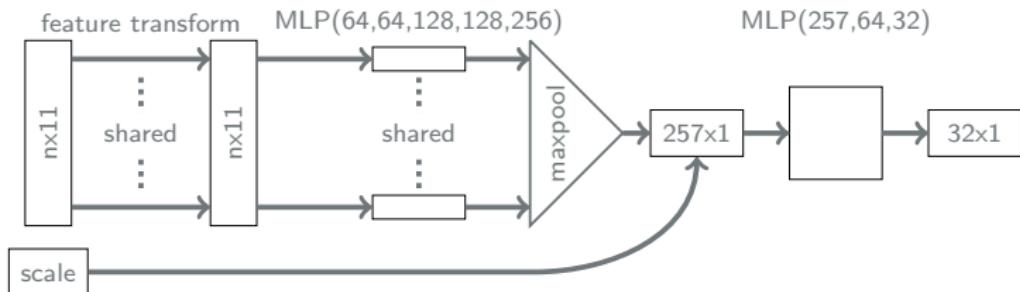
# Représentation de formes simples

---

- Chaque forme est subsamplée à 128 points
- Chaque point:  $(p_i, o_i, f_i)$

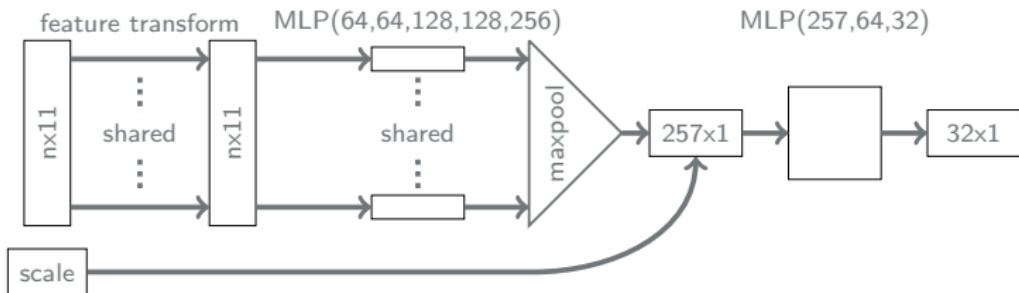
# Représentation de formes simples

- Chaque forme est subsamplée à 128 points
- Chaque point:  $(p_i, o_i, f_i)$
- PointNet réduit pour représenter les formes simples



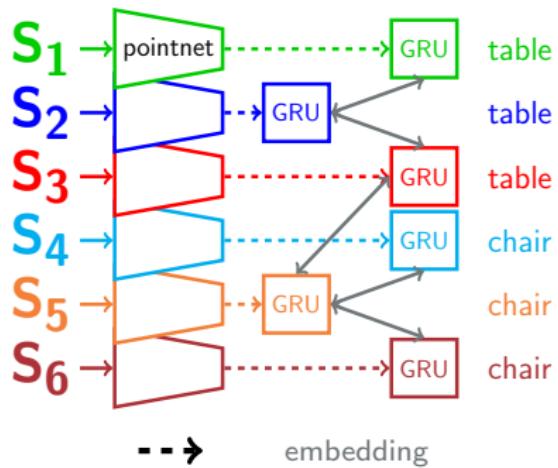
# Représentation de formes simples

- Chaque forme est subsamplée à 128 points
- Chaque point:  $(p_i, o_i, f_i)$
- PointNet réduit pour représenter les formes simples
- Une représentation de taille 32 par forme



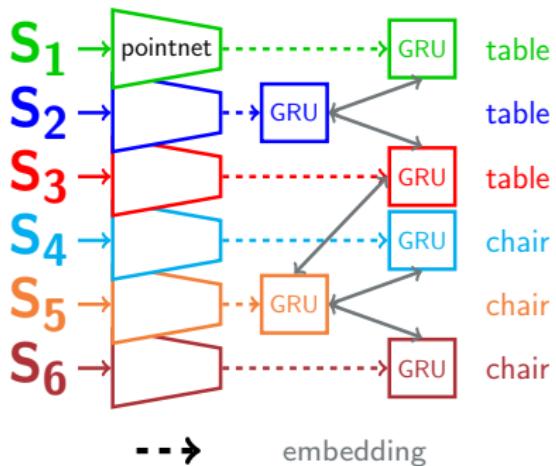
# Segmentation contextuelle

- ECC: passage de messages entre voisins dans  $\mathcal{G}$



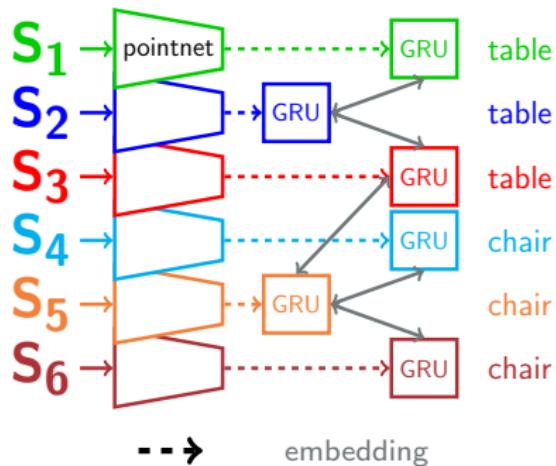
# Segmentation contextuelle

- ECC: passage de messages entre voisins dans  $\mathcal{G}$
- Etat interne + descripteurs d'arêtes → message



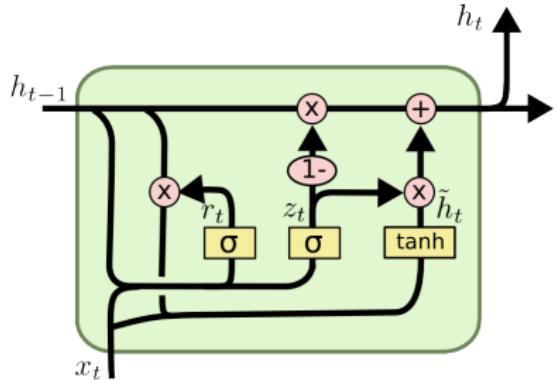
# Segmentation contextuelle

- ECC: passage de messages entre voisins dans  $\mathcal{G}$
- Etat interne + descripteurs d'arêtes → message
- Messages de taille 32



# Segmentation contextuelle

- ECC: passage de messages entre voisins dans  $\mathcal{G}$
- Etat interne + descripteurs d'arêtes → message
- Messages de taille 32
- Gated Recurrent Unit avec Input Gating



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

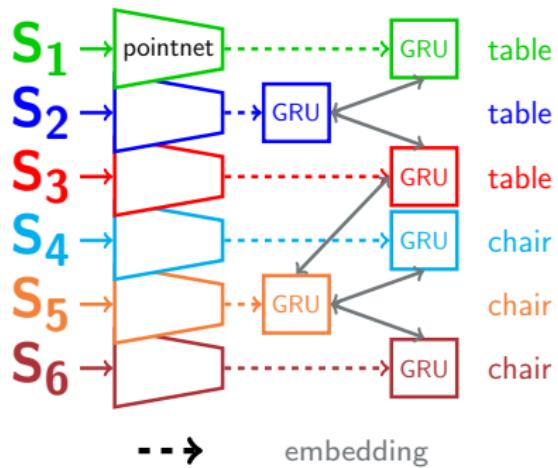
$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Image credit: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

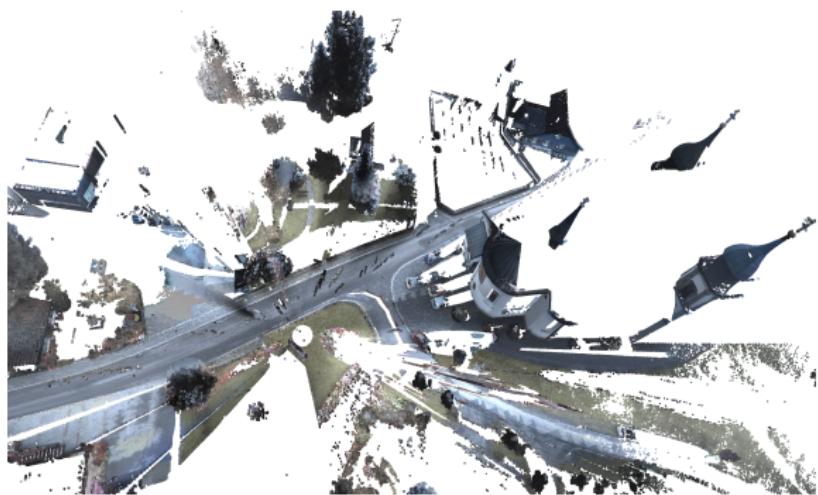
# Segmentation contextuelle

- ECC: passage de messages entre voisins dans  $\mathcal{G}$
- Etat interne + descripteurs d'arêtes → message
- Messages de taille 32
- Gated Recurrent Unit avec Input Gating
- Concaténation de 10 iterations = analyse multi-échelle



# Résultat qualitatif results: Semantic3D

Semantic3D: 3 milliards de points sur 30 nuages



- route
- herbe
- arbre
- buisson
- bâtiment
- aménagement
- artefact
- voiture

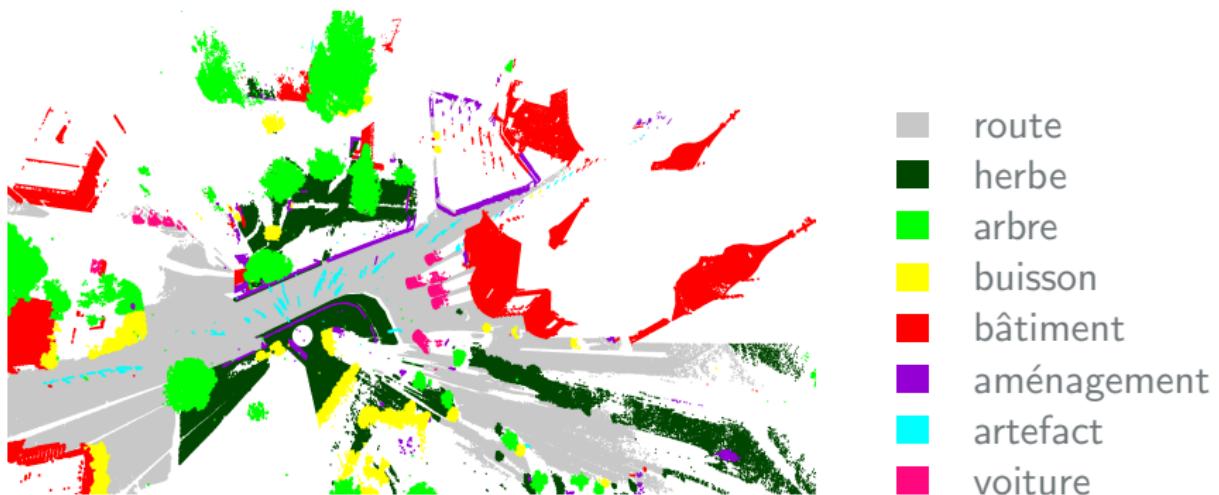
# Résultat qualitatif results: Semantic3D

Semantic3D: 3 milliards de points sur 30 nuages



# Résultat qualitatif results: Semantic3D

Semantic3D: 3 milliards de points sur 30 nuages



# Résultat qualitatif results: Semantic3D

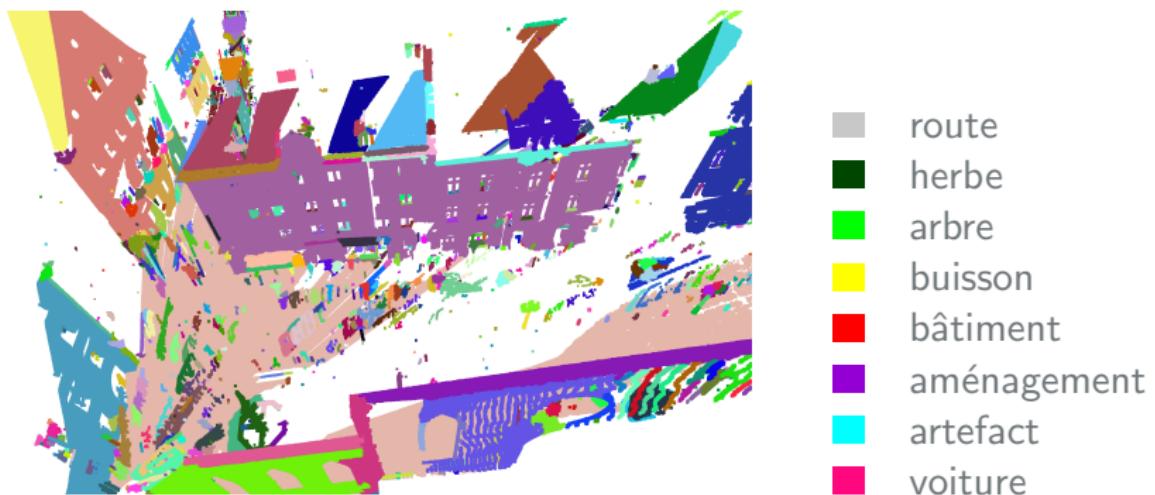
Semantic3D: 3 milliards de points sur 30 nuages



- route
- herbe
- arbre
- buisson
- bâtiment
- aménagement
- artefact
- voiture

# Résultat qualitatif results: Semantic3D

Semantic3D: 3 milliards de points sur 30 nuages

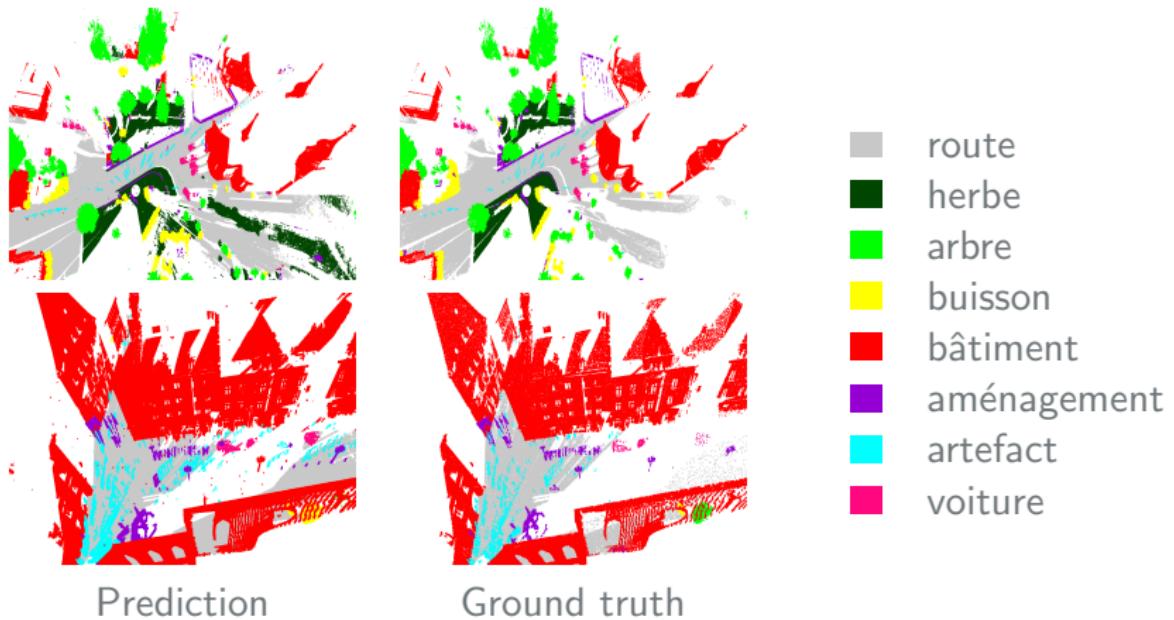


# Résultat qualitatif results: Semantic3D

Semantic3D: 3 milliards de points sur 30 nuages



# Résultat qualitatif results: Semantic3D

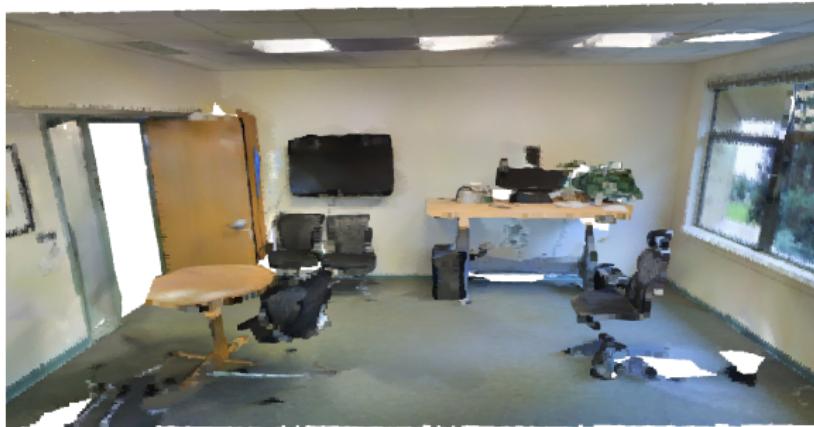


# Résultats quantitatifs: Semantic3D

Methode	OA	mIoU	route	herbe	arbre	buisson	bâti-ment	aména-ge-ment	arte-fact	voiture
reduced test set: 78 699 329 points										
TMLC-MSR	86.2	54.2	89.8	74.5	53.7	26.8	88.8	18.9	36.4	44.7
DeePr3SS	88.9	58.5	85.6	83.2	74.2	32.4	89.7	18.5	25.1	59.2
SnapNet	88.6	59.1	82.0	77.3	79.7	22.9	91.1	18.4	37.3	64.4
SegCloud	88.1	61.3	83.9	66.0	86.0	40.5	91.1	30.9	27.5	64.3
SPG (Ours)	<b>94.0</b>	<b>73.2</b>	<b>97.4</b>	<b>92.6</b>	<b>87.9</b>	<b>44.0</b>	<b>93.2</b>	<b>31.0</b>	<b>63.5</b>	<b>76.2</b>
full test set: 2 091 952 018 points										
TMLC-MS	85.0	49.4	91.1	69.5	32.8	21.6	87.6	25.9	11.3	55.3
SnapNet	91.0	67.4	89.6	<b>79.5</b>	74.8	56.1	90.9	36.5	34.3	77.2
SPG (Ours)	<b>92.9</b>	<b>76.2</b>	<b>91.5</b>	75.6	<b>78.3</b>	<b>71.7</b>	<b>94.4</b>	<b>56.8</b>	<b>52.9</b>	<b>88.4</b>

# Résultat qualitatif results: S3DIS

Intérieur, 3 bâtiments, 6 étages, 200+ pièces, 600 000 000+ points



- plafond
- sol
- mur
- colonne
- poutre
- fenêtre
- porte
- table
- chaise
- meuble
- tableau
- sofa
- autre

# Résultat qualitatif results: S3DIS

Intérieur, 3 bâtiments, 6 étages, 200+ pièces, 600 000 000+ points



- plafond
- sol
- mur
- colonne
- poutre
- fenêtre
- porte
- table
- chaise
- meuble
- tableau
- sofa
- autre

# Résultat qualitatif results: S3DIS

Intérieur, 3 bâtiments, 6 étages, 200+ pièces, 600 000 000+ points



- plafond
- sol
- mur
- colonne
- poutre
- fenêtre
- porte
- table
- chaise
- meuble
- tableau
- sofa
- autre

# Résultat qualitatif results: S3DIS

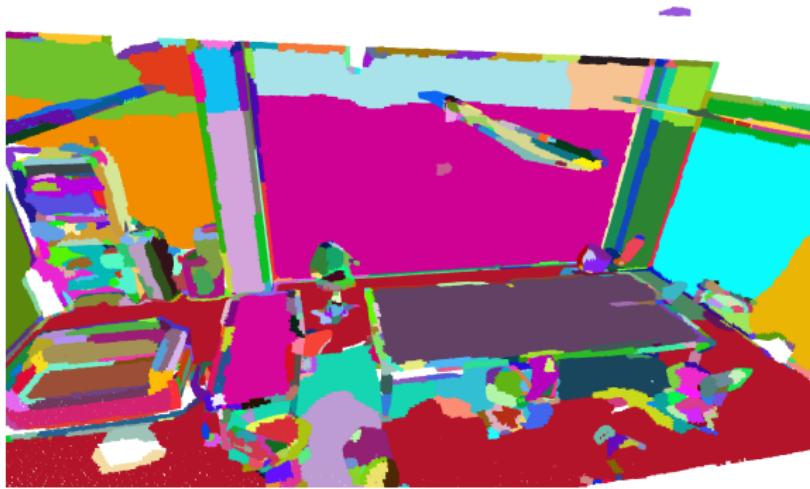
Intérieur, 3 bâtiments, 6 étages, 200+ pièces, 600 000 000+ points



- plafond
- sol
- mur
- colonne
- poutre
- fenêtre
- porte
- table
- chaise
- meuble
- tableau
- sofa
- autre

# Résultat qualitatif results: S3DIS

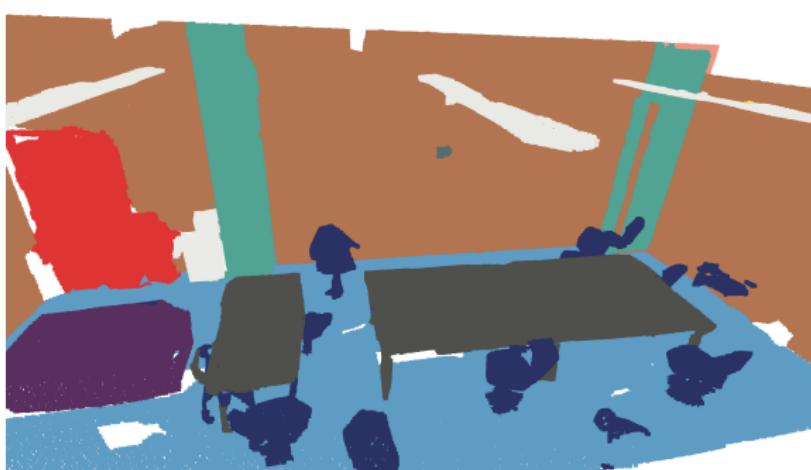
Intérieur, 3 bâtiments, 6 étages, 200+ pièces, 600 000 000+ points



- plafond
- sol
- mur
- colonne
- poutre
- fenêtre
- porte
- table
- chaise
- meuble
- tableau
- sofa
- autre

# Résultat qualitatif results: S3DIS

Intérieur, 3 bâtiments, 6 étages, 200+ pièces, 600 000 000+ points



- plafond
- sol
- mur
- colonne
- poutre
- fenêtre
- porte
- table
- chaise
- meuble
- tableau
- sofa
- autre

# Résultats qualitatif: S3DIS



- █ plafond
- █ sol
- █ mur
- █ colonne
- █ poutre
- █ fenêtre
- █ porte
- █ table
- █ chaise
- █ meuble
- █ tableau
- █ sofa
- █ autre

# Résultats quantitatif: S3DIS

Méthode	OA	mAcc	mIoU	porte	tableau
A5 PointNet	–	48.5	41.1	10.7	<b>26.3</b>
A5 SEGCloud	–	57.3	48.9	23.1	13.0
<b>A5 SPG</b>	86.4	<b>66.5</b>	<b>58.0</b>	<b>61.5</b>	2.1
PointNet	78.5	66.2	47.6	51.6	29.4
Engelmann	81.1	66.4	49.7	51.2	<b>30.0</b>
<b>SPG</b>	<b>85.5</b>	<b>73.0</b>	<b>62.1</b>	<b>68.4</b>	8.7

# Résultats quantitatif: S3DIS

Méthode	OA	mAcc	mIoU	porte	tableau
A5 PointNet	–	48.5	41.1	10.7	<b>26.3</b>
A5 SEGCloud	–	57.3	48.9	23.1	13.0
<b>A5 SPG</b>	86.4	<b>66.5</b>	<b>58.0</b>	<b>61.5</b>	2.1
PointNet	78.5	66.2	47.6	51.6	29.4
Engelmann	81.1	66.4	49.7	51.2	<b>30.0</b>
<b>SPG</b>	<b>85.5</b>	<b>73.0</b>	<b>62.1</b>	<b>68.4</b>	8.7

Étape	Nuage entier	2 cm	3 cm	4 cm
Voxelisation	0	40	24	16
Calcul des descripteurs	439	194	88	43
Partition géométrique	3428	1013	447	238
Calcul du SPG	3800	958	436	252
Inférence	240	110	60	50
Total	7907	2315	1055	599
mIoU 6-fold	54.1	60.2	62.1	57.1

# Perspectives

---

- De nouveaux data sets: architecture très versatile

# Perspectives

---

- **De nouveaux data sets:** architecture très versatile
- **Améliorer la partition:** vers une partition sémantique apprise

# Perspectives

---

- **De nouveaux data sets:** architecture très versatile
- **Améliorer la partition:** vers une partition sémantique apprise
- **Applications à d'autres problèmes:** très grandes images, videos, etc...

# Perspectives

---

- **De nouveaux data sets:** architecture très versatile
- **Améliorer la partition:** vers une partition sémantique apprise
- **Applications à d'autres problèmes:** très grandes images, vidéos, etc...
- Dépôt sur GitHub:  
[github.com/loicland/superpoint-graph](https://github.com/loicland/superpoint-graph)

# Perspectives

---

- **De nouveaux data sets:** architecture très versatile
- **Améliorer la partition:** vers une partition sémantique apprise
- **Applications à d'autres problèmes:** très grandes images, vidéos, etc...
- Dépôt sur GitHub:  
[github.com/loicland/superpoint-graph](https://github.com/loicland/superpoint-graph)
- Licence MIT

# Perspectives

---

- **De nouveaux data sets:** architecture très versatile
- **Améliorer la partition:** vers une partition sémantique apprise
- **Applications à d'autres problèmes:** très grandes images, vidéos, etc...
- Dépôt sur GitHub:  
[github.com/loicland/superpoint-graph](https://github.com/loicland/superpoint-graph)
- Licence MIT
- Modèles pré-entraînés disponibles

# Etude d'ablation

---

Modèle	mAcc	mIoU
Best	73.0	62.1
Perfect	92.0	88.2
Unary	50.8	40.0
iCRF	51.5	40.7
CRF – ECC	66.4	56.4
GRU13	69.1	58.5
NoInputGate	68.6	57.5
NoConcat	69.3	57.7
NoEdgeFeat	50.1	39.9
ECC – mem	70.2	59.4

**Table:** Étude d'ablation des différents choix et apports de cet article (validation croisée sur S3DIS)

# Ablation study

<i>Superpoints max</i>	256	<b>512</b>	1024	
mIoU	57.9	62.1	60.4	
<i>Limit des superarêtes</i>	1 m	5 m	<b>10 m</b>	
mIoU	61.0	61.3	62.1	
<i>Regularisation <math>\mu</math></i>	0.01	0.02	<b>0.3</b>	0.4
# superpoints	785 010	385 091	251 266	186 108
perfect mIoU	90.6	88.2	86.6	85.2
mIoU	59.1	59.2	62.1	58.8
<i>Modèle</i>	LSTM	LSTM+IG	GRU	<b>GRU+IG</b>
mIoU	61.0	59.6	57.5	62.1
<i>Taille des superpoint</i>	1-40	40-128	128-1000	$\geq 1000$
proportion	25.2%	34.7%	31%	9.2%

**Table:** Évaluation sur S3DIS par validation croisée de différentes configurations. Les choix retenus dans notre méthode sont en gras.