



TPE :Travail Personnel Encadré

OPTION : Systeme Intelligent et Multi-média

RAPPORT FINAL

THÈME : Segmentation de données laser 3D acquises en environnement urbain par une plate-forme mobile

Auteur :

A.Djalil OUSSEINI HAMZA

Encadrant :

Dr. Damien VIVET

Promotion XXII
3 décembre 2018

Table des matières

1	INTRODUCTION GÉNÉRALE	3
1.1	Contexte	3
1.2	Problématique	3
2	ANALYSE DU SUJET	3
2.1	Introduction	3
2.2	Domaines d'applications	4
2.3	Conclusion	4
3	ETAT DE L'ART	4
3.1	Introduction	4
3.2	Travaux connexes	5
3.3	Présentation et détails des méthodes de segmentation	5
3.3.1	Méthode basé sur les bords (edge)	5
3.3.2	Méthode basé sur les régions (region)	6
3.3.3	Méthode basé sur les attributs (attributes)	6
3.3.4	Méthode basé sur les modèles (model)	7
3.3.5	Méthode basé sur les graphes (graph)	7
3.4	Conclusion	7
4	SOLUTION PROPOSÉE	8
4.1	Introduction	8
4.2	L'algorithme proposée	8
4.3	Implémentation de l'algorithme pour notre cas	10
5	RÉSULTATS DE LA SOLUTION PROPOSÉE	11
5.1	Outils utilisés :	11
5.2	Exécution du programme	11
5.3	Résultats :	11
6	CONCLUSION GÉNÉRALE	18

Table des figures

1	Taxonomie des méthodes de segmentation des nuages de points 3D	5
2	Vue d'ensemble de la méthode proposée.	8
3	Vue d'ensemble de l'algorithme de segmentation rapide au sol.	9
4	l'algorigramme de notre solution	10
5	Visualisation du fichier non segmenté	11
6	Visualisation du fichier non segmenté	12
7	application de l'algorithme de segmentation de sol	12
8	application de l'algorithme de segmentation de sol	13
9	Visualisation de l'algorithme de segmentation de sol	13
10	Visualisation de l'algorithme de segmentation de sol	13
11	Visualisation de l'algorithme de segmentation de sol	14
12	Visualisation de l'algorithme de segmentation de sol	14
13	Visualisation des deux fichiers : ground et object	15
14	Visualisation des deux fichiers : ground et object	15
15	Application du flood-fill	16
16	Application du flood-fill	16
17	Visualisation du flood-fill	16
18	Visualisation du flood-fill	17

1 INTRODUCTION GÉNÉRALE

La segmentation est un procédé de découpage d'éléments en des parties relativement simple pour un objectif précis. Ainsi, on pourrait définir la segmentation de nuages de points 3D comme étant un mécanisme permettant de diviser un nuage de points en des segments pour une identification ultérieure.

L'objectif de notre travail est de pouvoir regrouper les différents éléments d'un nuage de points en des classes homogènes.

1.1 Contexte

Le contexte de notre travail intervient dans l'analyse des scènes, la détection (regroupement en éléments homogènes et non l'identification) des objets urbains, de pouvoir distinguer les objets de la route (par objet, il faut comprendre ici, tout ce qui est non-sol y compris les piétons) dans un nuage de points acquis par des capteurs lasers 3D.

1.2 Problématique

Après la description du contexte dans lequel s'inscrit notre travail, nous abordons dans les lignes qui suivent les différents problèmes aux quels nous pourrions se confronter.

L'objectif de ce travail était de segmenter un nuage de points 3D issue d'un laser panoramique en éléments uniformes pour une analyse de l'environnement. Les question que nous nous posons sont :

1. Après l'étude bibliographique arrivons-nous à nous procurer d'une base de données de nuage de point sur laquelle on testera l'algorithme retenu ?
2. Arrivons nous à implémenter partiellement ou intégralement l'algorithme retenu ?
3. A partir de quels aspects ou sur quoi va-t-on se baser pour valider nos résultats ?

2 ANALYSE DU SUJET

2.1 Introduction

Notre sujet s'inscrit dans le cadre du module TPE (Travail Personnel Encadré) de l'Institut Francophone International. A l'issue de ce cours, chaque étudiant doit choisir un thème parmi les divers thème proposés par différents enseignants (encadrant) qu'il doit développer (analyser, faire l'état de l'art et implémenter la solution retenue) durant tout le temps du Master 1 (6 mois). A cet effet, il nous a été proposé de travailler sur le thème : Segmentation automatique de nuages de points 3D urbain sous l'encadrement du docteur Damien VIVET, professeur-assistant à ISAE-Supaero, Toulouse, France.

2.2 Domaines d'applications

Nous savons depuis longtemps que la segmentation joue un rôle essentielle dans la vie de tous les jours. En réalité notre cerveau avant reconnaître un objet ou une scène quelconque segmente d'abord automatiquement cette scène en élément simple et homogène avant de passer à l'identification. De ce fait, on pourra dire que la segmentation des nuages de points est donc importante voire même capitale pour de nombreuses applications.

Les domaines d'application de notre sujet sont assez vague, ils vont de l'analyse des scènes forestières à celles d'urbaines tels-que la réalisations des cartographies, la détection des sol et non-sol, détection d'objets etc. Le domaine spécifique d'application de notre sujet intervient surtout dans la navigation des robots, des véhicules autonomes afin de distinguer le sol des non-sol pour ne pas se heurter aux obstacles de la route. Nous allons donc nous focaliser ici sur la segmentation des sol et non-sol.

2.3 Conclusion

Le travail demandé consiste à :

- Faire une étude de l'existant des travaux les plus pertinents,
- Comparer les méthodes les plus pertinentes de l'état de l'art,
- Implémenter l'algorithme de segmentation retenu dans l'état de l'art approprié au thème en C++ de préférence,
- Comparer l'algorithme avec les techniques de l'état de l'art.

3 ETAT DE L'ART

3.1 Introduction

L'état de l'art ou recherche bibliographique est l'étude de l'existant sur un sujet donné. En d'autres termes, c'est l'ensemble des méthodes, procédures et techniques ayant pour objet de retrouver les références bibliographiques des documents les plus pertinents. C'est une étape cruciale dans la résolution de problème d'un sujet donné, car, elle permet de mieux connaître les travaux des autres afin de ne pas répéter les mêmes découvertes et de pouvoir apporter sa contribution.

La segmentation est cruciale pour traiter des données, et très souvent, elle est une étape de pré-traitement. Notre sujet traite de la segmentation automatique des nuages de points 3D urbain. Et nous savons que la segmentation 3D est l'ensemble des phénomènes permettant de découper un nuage de points en des groupes d'objets homogènes et donc, pour une meilleur segmentation des données il faut des outils sophistiqués d'acquisition d'images tels les photogrammétries, les lasergrammétries etc. ces outils constituent un moyen rapide et précis de numériser en 3D les nuages de points des environnements urbains [8].

Plusieurs travaux ont introduit des approches et/ou des techniques différentes pour aborder le problème de segmentation en créant, ou en apportant des améliorations aux algorithmes déjà existants. Dans la suite de ce chapitre, nous verrons quelques travaux réalisés sur la segmentation et nous en présenterons les avantages et les limites.

3.2 Travaux connexes

Avec l'avènement des systèmes de balayage laser mobile (MLS) les nuages de points 3D peuvent facilement être acquis. Il faut alors pouvoir les découper en les regroupant en régions semblables tout en assurant une bonne répartition de ces régions. Aujourd'hui de plus en plus de travaux s'intéressent à l'analyse des scènes urbaines ce qui signifie qu'elle reste un domaine très actif. Ainsi, plusieurs travaux ont vu le jour avec différents objectifs pour apporter des solutions automatiques aux segmentation des nuages de point 3D. Les approches de segmentation des nuages de points énoncées dans la littérature nous permet selon [11] de les classer en cinq (5) catégories en fonction de leurs mécanismes de conception. Nous avons (FIGURE 1) :

- Les méthodes basées sur les attributs ;
- Les méthodes basées sur les bords ;
- Les méthodes basées sur les graphes ;
- Les méthodes basées sur les modèles ;
- Et les méthodes basées sur les régions.

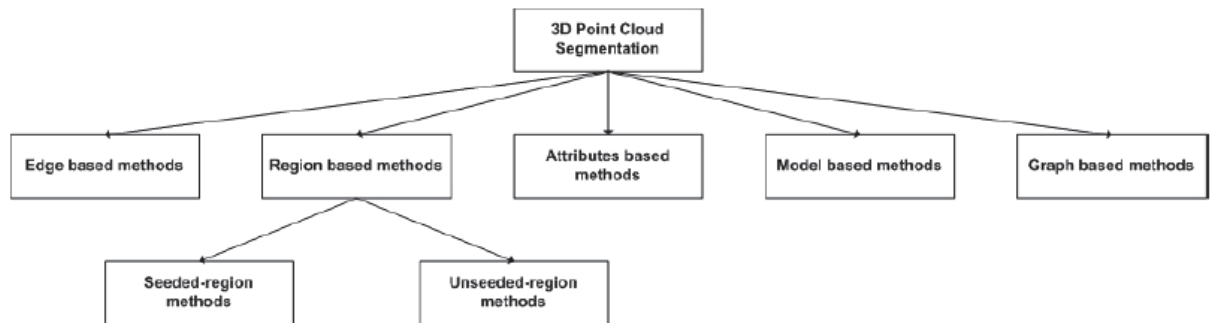


FIGURE 1 – Taxonomie des méthodes de segmentation des nuages de points 3D

Dans la suite du travail, nous apporterons plus de détails sur les méthodes de segmentation citées ci-haut et nous mentionnons les travaux associés.

3.3 Présentation et détails des méthodes de segmentation

3.3.1 Méthode basé sur les bords (edge)

Les bords décrivent les caractéristiques de la forme des objets. Dans [10], nous pouvons remarquer qu'il existe plusieurs façons d'effectuer une détection de contour. Cependant, on peut les regrouper en deux catégories dans le cas d'images 2D, Gradient et Laplacien. La méthode de gradient détecte les bords en recherchant le maximum et le minimum dans la première dérivée

de l'image. La méthode laplacienne recherche des passages par zéro dans la dérivée seconde de l'image pour trouver des bords. Dans le cas des nuages de points 3D, dans [11] on constate que les méthodes basées sur les bords détectent les limites de plusieurs régions dans les nuages de points pour obtenir des régions segmentées. Le principe de ces méthodes est de localiser les points qui ont un changement rapide de l'intensité, à l'instar de [2], de Bhanu et al., qui ont proposé une technique de détection des contours en calculant le gradient, en ajustant les lignes 3D à un point de consigne et en détectant les changements de direction des vecteurs normaux unitaires sur la surface. Bien que les méthodes basées sur les bords permettent une segmentation rapide, elles présentent des problèmes de précision, car elles sont toutes très sensibles aux bruits et à la densité inégale des nuages de points, ce qui se produit fréquemment dans les données de nuages de points.

3.3.2 Méthode basé sur les régions (region)

Dans [11] nous pouvons constater que les méthodes basées sur une région utilisent des informations de voisinage pour combiner des points voisins ayant des propriétés similaires pour obtenir des régions isolées et par conséquent trouver une dissemblance entre les différentes régions. Elles sont plus résistantes aux bruits que les méthodes basées sur les bords. Mais elles ont des problèmes de sur ou sous-segmentation et de déterminer avec précision les frontières de la région. Il est à noter, par ailleurs que les méthodes basées sur les régions sont de deux catégories : les méthodes à régionensemencée (ou ascendante) et les méthodes à région non-ensemencée (ou descendante).

3.3.3 Méthode basé sur les attributs (attributes)

Dans [1], Barnea et Filin (2013) ont proposé une méthode de segmentation pour les données de balayage laser terrestre en intégrant des gammes, des normales et des couleurs dans une représentation panoramique. La méthode de segmentation proposée a donné plus de segments physiquement significatifs. Les objets ont été extraits en formant les segments comme unités significatives selon des règles prédéfinies (Pu et al., 2011, Yang et Dong, 2013). Ces méthodes basées sur les voxels accélèrent la vitesse de calcul. Néanmoins, la qualité de segmentation est soumise à la taille des voxels. Les voxels d'une taille fixe conduisent à des résultats incorrects, en particulier dans les zones d'objets mixtes denses. Dans [12], Bisheng Yang et al., 2014 ont présenté une méthode robuste et efficace pour segmenter un environnement urbain. La méthode extrait des objets des points non-sol des scènes urbaines en supprimant les points au sol (basant sur les travaux de Hernández et Matcotequi) elle génère des supervoxels de deux tailles différentes selon les attributs(couleurs, intensité) et segmente ces supervoxels plutôt que ces points individuels. La méthode proposée génère des supervoxels multi-échelles à partir de nuages de points MLS et segmente ces supervoxels plutôt que des points individuels, ce qui réduit considérablement les coûts de calcul. Cependant, il y a beaucoup d'insuffisance dans l'extraction d'objets urbains plus compliqués, tels que des bâtiments avec des parties non structurées (par exemple, des surfaces de forme libre).

3.3.4 Méthode basé sur les modèles (model)

Les travaux cités dans [11] stipulent que les méthodes basées sur un modèle utilisent des formes primitives géométriques (par exemple sphère, cône, plan et cylindre) pour regrouper des points. Les points qui ont la même représentation mathématique sont regroupés en segments. Dans [7] William Gélard et al., 2017, ont présenté une méthode de segmentation 3D basée modèle. Cette méthode traite de la segmentation 3D d'un nuage de points de tournesol. Les auteurs ont appliqué un algorithme de segmentation dit Euclidien, c'est-à-dire se basant uniquement sur une contrainte géométrique. Cet algorithme leur a permis de regrouper en différents clusters les bases d'insertion des pétioles (partie reliant la tige à une feuille). Mais cette méthode montre des limites quant à la procédure d'acquisition qui se révèle un peu longue.

3.3.5 Méthode basé sur les graphes (graph)

Les méthodes basées sur les graphes voient les nuages de points comme des graphiques, les techniques présentées dans [9] sont robustes aux bruits et ne nécessitent pas de techniques avancées de réduction de bruit dans les étapes de pré-traitement. Dans [6] les travaux effectués sur les graphes utilisent un prédicat de fusion pour décider si, à un certain paramètre d'échelle, deux régions adjacentes d'une image doivent être fusionnées en une seule, produisant ainsi une image segmentée. Dans [4] EJYC Cahuina et al. ont mis au point un algorithme de segmentation basée sur les graphes en améliorant les travaux de [6]. Les auteurs de [3] proposent une méthode de segmentation d'image basée sur un graphique hiérarchique. La hiérarchie est construite à partir d'une image via une représentation graphique, basée sur la notion de hiérarchie de zone quasi-plate.

3.4 Conclusion

A l'issu de cette étude bibliographique, nous constatons que beaucoup d'approches ont été proposées par les chercheurs. Néanmoins, en général, il existe **deux (2) approches de base**. La première approche utilise un **modèle purement mathématique** et un **raisonnement géométrique**, et la deuxième approche **extraît des entités 3D à partir de données de nuage de points** à l'aide des descripteurs d'entités et utilise des **techniques d'apprentissage automatique** pour apprendre les différentes classes de types d'objets, puis utilise le modèle résultant pour classer les données acquises. Cependant, dans notre cas, nous avons présenté une étude sommaire des méthodes de segmentation des nuages de points 3D. Nous les avons regroupées par ailleurs en 5 catégories comme mentionné dans [11] en fonction de leurs mécanisme de conception. Ensuite, nous avons présenté une étude comparative superficielle de ces méthodes dans un tableau. Il ne s'agit pas de donner une comparaison proprement dite, mais de voir comment se présente les résultats par-rapport aux approches utilisées car les différents problèmes sont abordés séparément ce qui ne nous a pas permis de faire une évaluation globale du problème.

4 SOLUTION PROPOSÉE

4.1 Introduction

Suite à l'étude bibliographique que nous avons menée, le choix de la méthode qui nous a semblé adéquat pour notre sujet s'est porté sur les travaux de [5] de Phuong Minh Chu et al., la méthode proposée dans cet article se compose de deux étapes : une première qui est axée sur un algorithme de **ground-segmentation** (segmentation de sol) et une seconde basé sur l'algorithme de **flood-fill** (remplissage par inondation) de 3D.

4.2 L'algorithme proposée

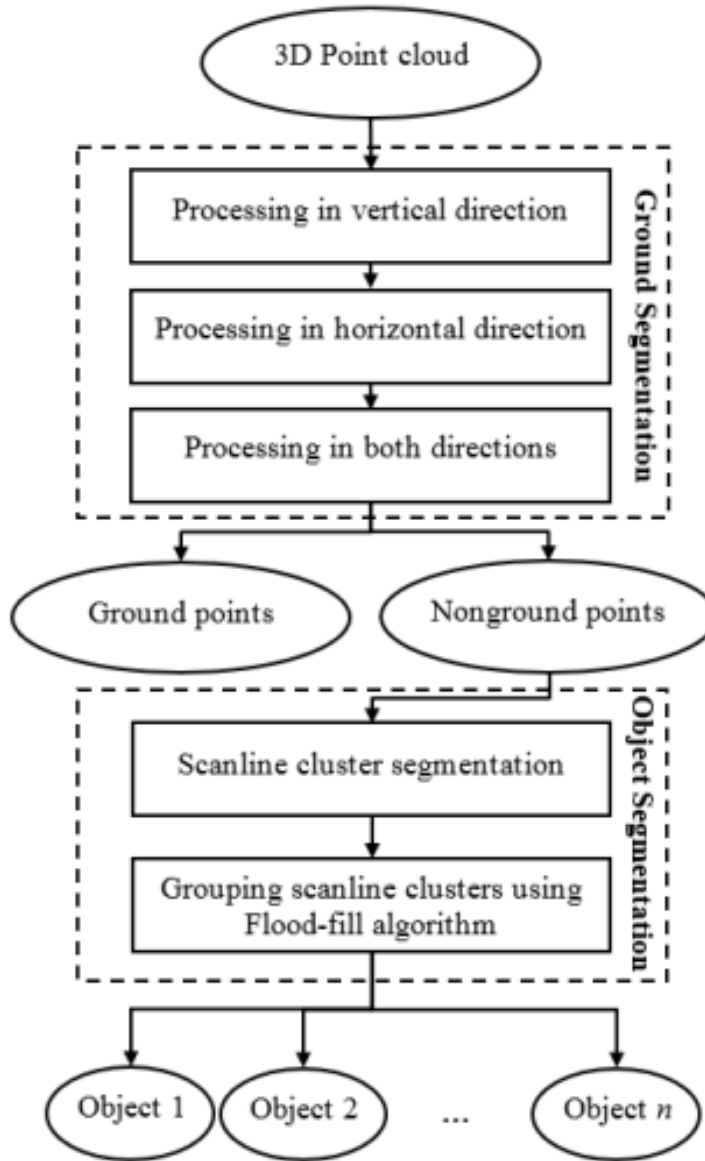


FIGURE 2 – Vue d'ensemble de la méthode proposée.

[5] Phuong Minh Chu et al.

Explication de l'algorithme :

1. Algorithme de détection de sol, art.[5]

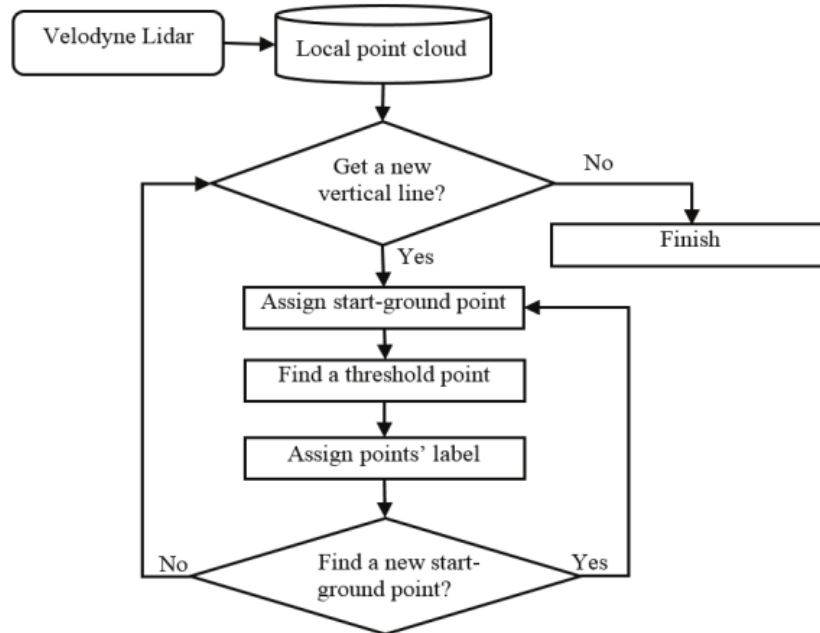


FIGURE 3 – Vue d'ensemble de l'algorithme de segmentation rapide au sol.

[5] A Fast Ground Segmentation Method for 3D Point Cloud, Phuong Minh Chu et al., Journal of information processing systems

2. On garde que sur les points non "sol".
3. Sur la matrice 2D restante, on regarde chaque ligne, pour chaque point successif on mesure la distance et compare a un seuil (**si seuil > objets différents** ou **si seuil < objet identique**), on met un index à chaque point représentant l'ID de l'objet.
4. A ce stade, pour chaque ligne on a des groupes de points sensé appartenir aux mêmes objets.
5. Il faut relier les lignes entre elles, c'est la que le flood fill intervient. Pour chaque groupe de point on fait grandir les région dans les 4 directions sous la condition que la différence de distance est < au seuil précédent (ou seuil > car la résolution en élévation est sv< que la résolution en azimuth).
6. Si deux régions avec des index différents se touchent on les fusionne car c'est le même objet.

En résumé on a :

- La première étape correspond à la segmentation au sol,
- Ensuite, on sépare chaque objet de la partie non-sol,
- Enfin, on obtient une liste d'objets dans chaque données de trame.

4.3 Implémentation de l'algorithme pour notre cas

En ce qui nous concerne nous avons simplement implémenter l'algorithme de détection de sol c'est-à-dire la première partie de l'algorithme qui a été cité dans la solution proposée. En effet, notre algorithme prendra en entrée un nuage de point de type **.PCD (Point Cloud Data)** et fournis en sorti deux fichiers de type **.PCD** : les sol et les non-sol. Ensuite, pour fichier non-sol, nous avons pris et appliqué l'algorithme flood-fill.

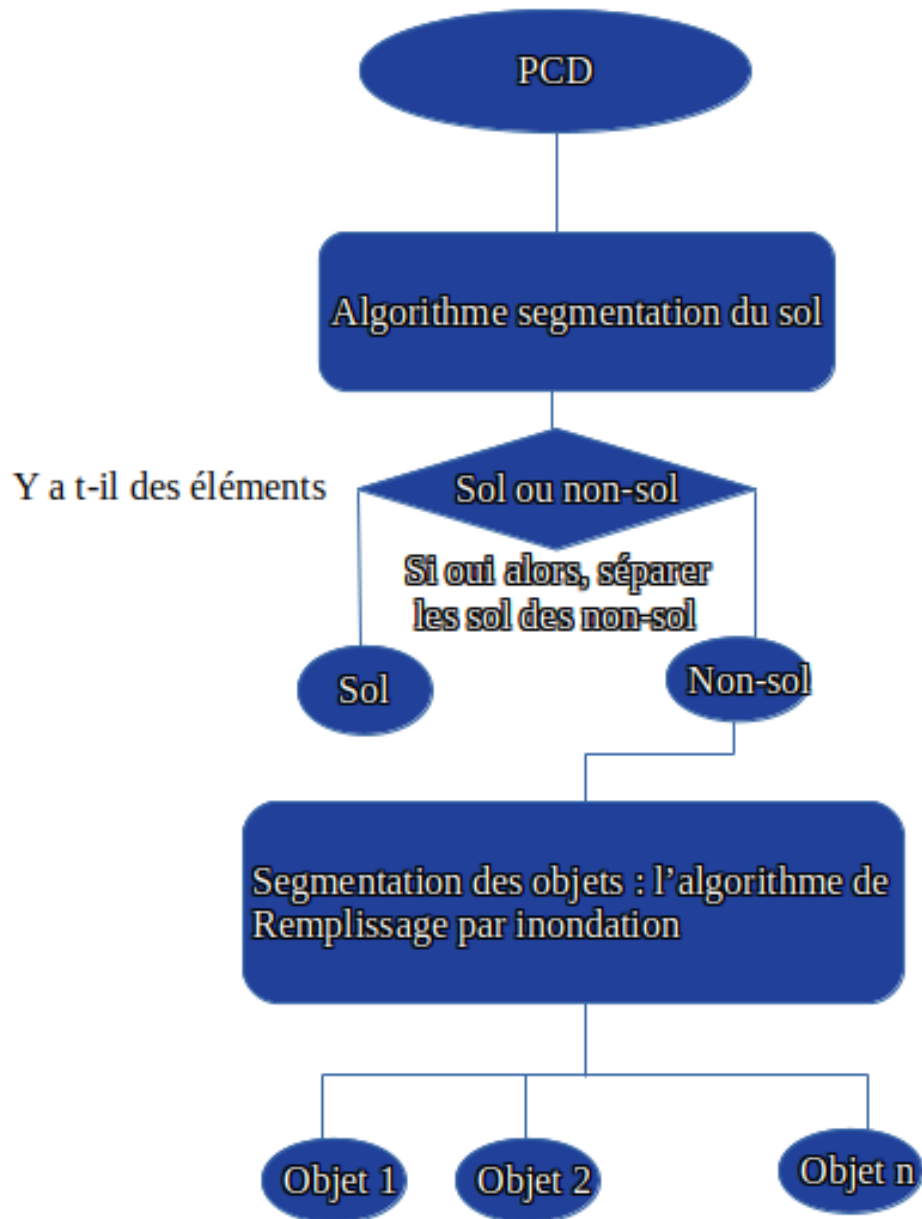


FIGURE 4 – *l'algorithme de notre solution*
Inspirer par les travaux de [5]

5 RÉSULTATS DE LA SOLUTION PROPOSÉE

5.1 Outils utilisés :

- Librairie : **Point Cloud Library (PCL) version pcl-1.8.1**
- Langage de programmation : **C++**
- Compilateur : **g++ avec l'outil Cmake version 3.10.2**
- Système d'exploitation : **Ubuntu 18.04 LTS**

5.2 Exécution du programme

Pour lancer le programme il faut ouvrir un terminal, ensuite il faut se déplacer dans le répertoire se trouvant l'un des deux fichier .cpp. Enfin, on exécute le fichier .cpp au moyen de **cmake** . puis **make** ce qui entraîne la création d'un fichier **exécutable .o**

5.3 Résultats :

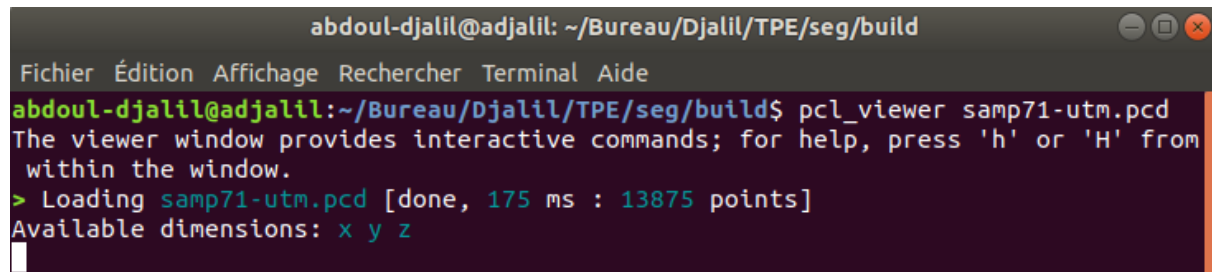
Comme nous l'avions préciser tout haut, les étapes de notre algorithme se font en deux étapes :

1. Il s'agit ici du **ground segmentation** c'est-à-dire on va d'abord segmenter notre fichier .pcd par notre algorithme de segmentation de sol,
2. Ensuite, nous appliquons le **flood-fill** à l'un des fichiers généré par le ground segmentation.

Nous avons :

GROUND SEGMENTATION :

On va d'abord visualiser le fichier **samp71-utm.pcd non segmenté** à l'aide de l'outil **pcl_viewer** de la librairie PCL.



```
abdoul-djalil@adjalil: ~/Bureau/Djalil/TPE/seg/build
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
abdoul-djalil@adjalil:~/Bureau/Djalil/TPE/seg/build$ pcl_viewer samp71-utm.pcd
The viewer window provides interactive commands; for help, press 'h' or 'H' from
within the window.
> Loading samp71-utm.pcd [done, 175 ms : 13875 points]
Available dimensions: x y z
```

FIGURE 5 – Visualisation du fichier non segmenté

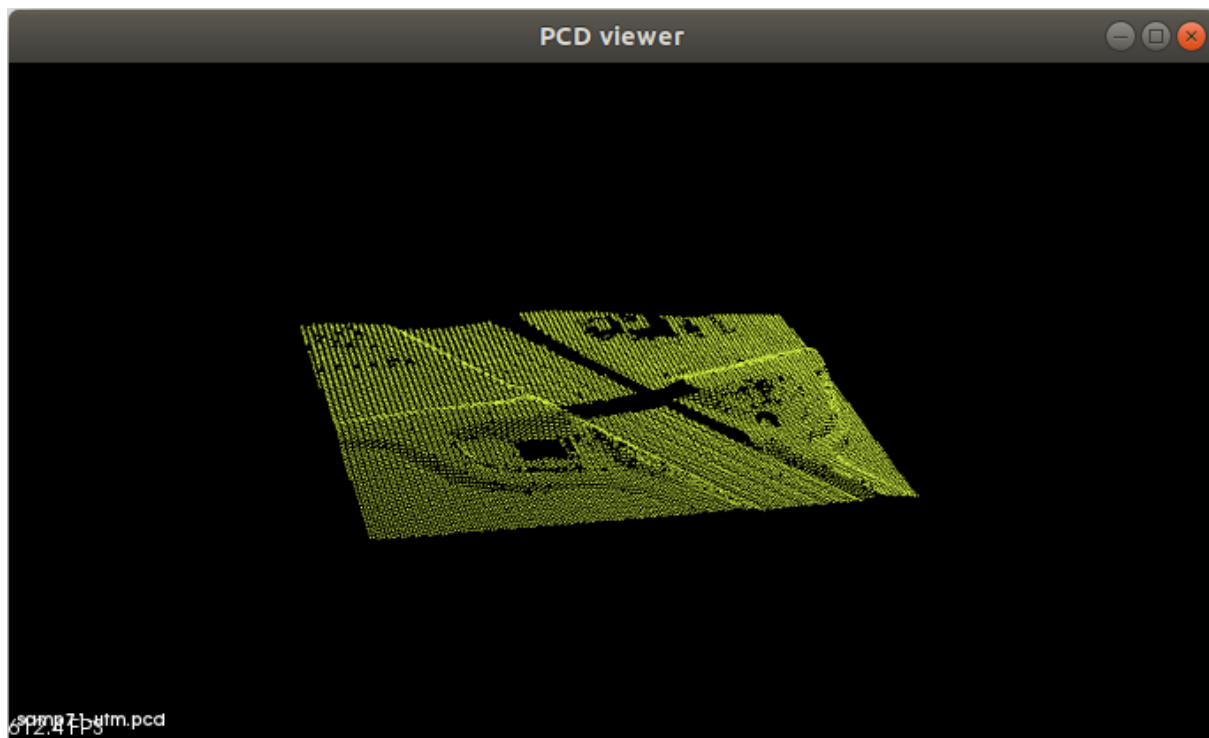


FIGURE 6 – Visualisation du fichier non segmenté

Maintenant, on applique notre algorithme de segmentation de sol, le **Ground segmentation**

```

abdoul-djalil@adjalil: ~/Bureau/Djalil/TPE/seg/build
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
abdoul-djalil@adjalil:~/Bureau/Djalil/TPE/seg/build$ ./seg samp71-utm.pcd
Cloud before filtering:
  points[]: 13875
width: 13875
height: 1
is_dense: 1
sensor origin (xyz): [0, 0, 0] / orientation (xyzw): [0, 0, 0, 1]

Ground cloud after filtering:
points[]: 12933
width: 12933
height: 1
is_dense: 1
sensor origin (xyz): [0, 0, 0] / orientation (xyzw): [0, 0, 0, 1]

Object cloud after filtering:
points[]: 942
width: 942
height: 1
is_dense: 1
sensor origin (xyz): [0, 0, 0] / orientation (xyzw): [0, 0, 0, 1]

fin 758 ms
abdoul-djalil@adjalil:~/Bureau/Djalil/TPE/seg/build$

```

FIGURE 7 – application de l'algorithme de segmentation de sol

Nous pouvons remarquer ici que l'algorithme a bien fonctionné, il a généré deux fichiers aux quels nous y attendons : un fichiers dénommé **ground_samp71-utm** et un autre du nom de **object_samp71-utm**

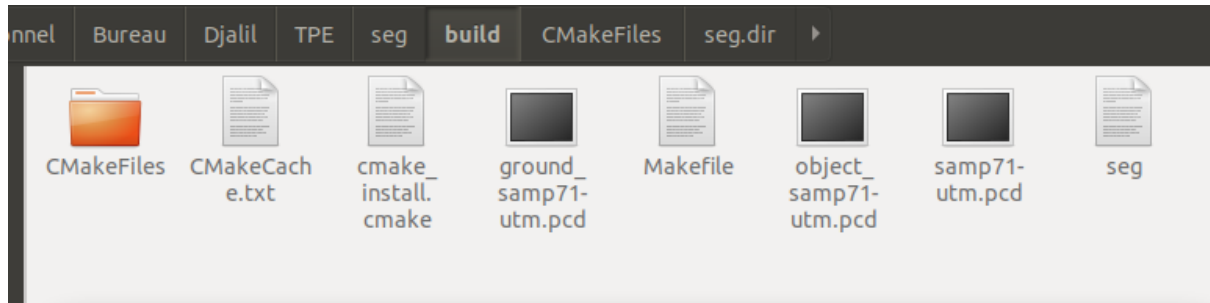


FIGURE 8 – application de l'algorithme de segmentation de sol

Passons à la visualisation des fichiers générés :

Le premier fichier ==> **ground_samp71-utm** :

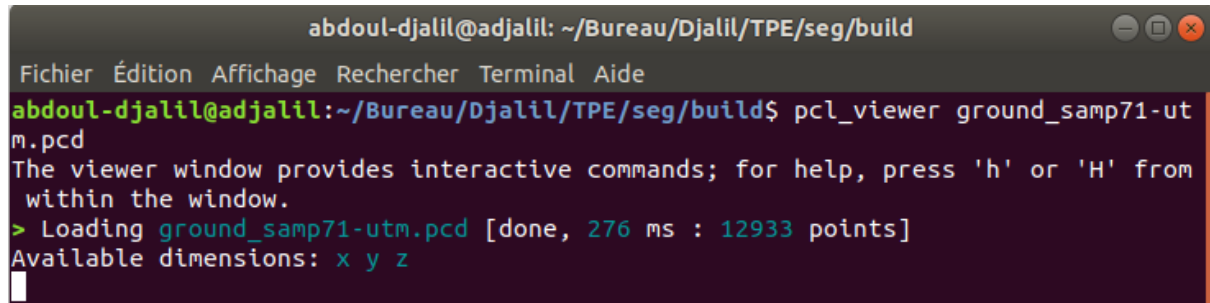


FIGURE 9 – Visualisation de l'algorithme de segmentation de sol

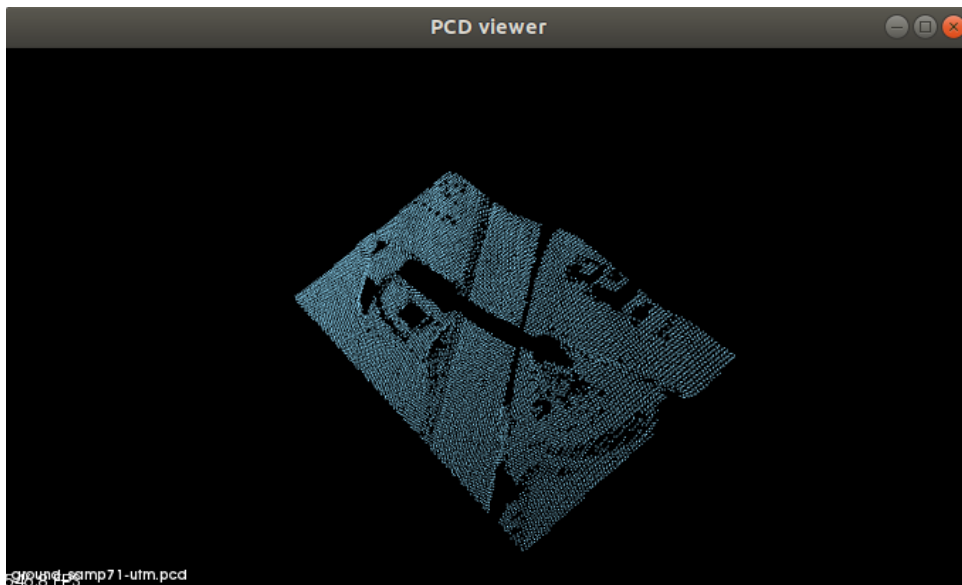


FIGURE 10 – Visualisation de l'algorithme de segmentation de sol

Le second fichier ==> object_samp71-utm :

```
abdoul-djalil@adjalil: ~/Bureau/Djalil/TPE/seg/build
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
abdoul-djalil@adjalil:~/Bureau/Djalil/TPE/seg/build$ pcl_viewer object_samp71-ut
m.pcd
The viewer window provides interactive commands; for help, press 'h' or 'H' from
within the window.
> Loading object_samp71-utm.pcd [done, 182 ms : 942 points]
Available dimensions: x y z
```

FIGURE 11 – Visualisation de l'algorithme de segmentation de sol

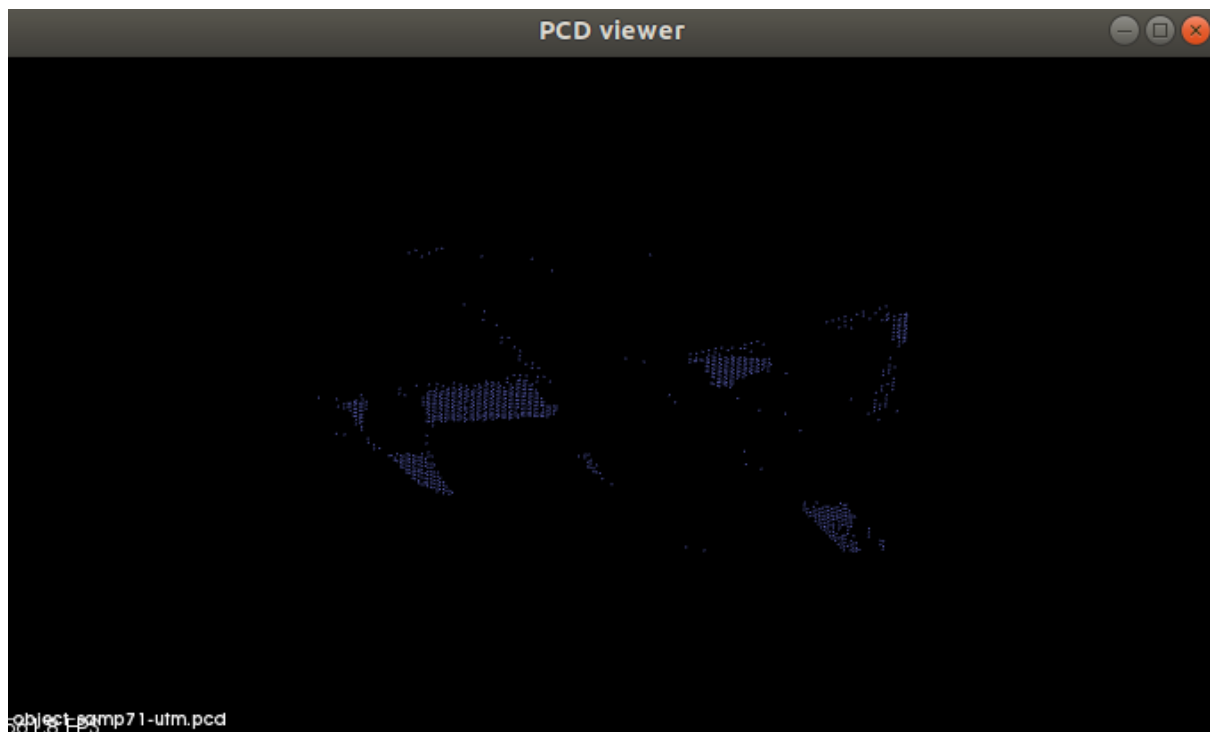


FIGURE 12 – Visualisation de l'algorithme de segmentation de sol

On remarque très bien ici que les objets ont été bien mis en évidence ce qui montre le bon fonctionnement de notre algorithme.

Essayons maintenant de voir le comportement des deux fichiers en les affichant en même temps :

```
abdoul-djalil@adjalil: ~/Bureau/Djalil/TPE/seg/build
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
abdoul-djalil@adjalil:~/Bureau/Djalil/TPE/seg/build$ pcl_viewer ground_samp71-ut
m.pcd object_samp71-utm.pcd
The viewer window provides interactive commands; for help, press 'h' or 'H' from
within the window.
> Loading ground_samp71-utm.pcd [done, 284 ms : 12933 points]
Available dimensions: x y z
> Loading object_samp71-utm.pcd [done, 11 ms : 942 points]
Available dimensions: x y z
```

FIGURE 13 – Visualisation des deux fichiers : *ground et object*

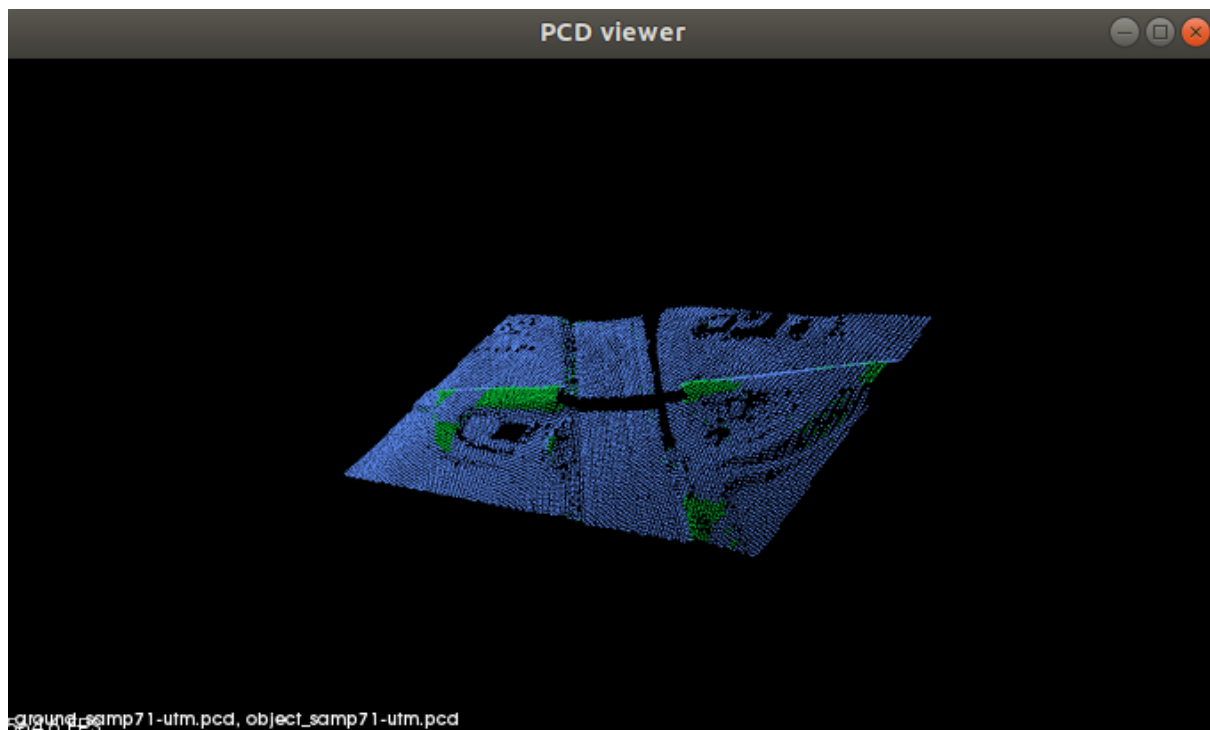


FIGURE 14 – Visualisation des deux fichiers : *ground et object*

Le résultat est tout simplement extraordinaire, on voit très bien le contraste entre le sol et le non-sol. Ce qui stipule que la segmentation a marché.

FLOOD-FILL :

On va appliquer l'algorithme de remplissage par inondation au deuxième fichier : `object_samp71-utm` afin de remplir les zones connectées de même couleur avec une couleur différente.


```
abdoul-djalil@adjalil: ~/Bureau/Djalil/TPE/flood_fill/build
Fichier Édition Affichage Rechercher Terminal Aide
abdoul-djalil@adjalil:~/Bureau/Djalil/TPE/flood_fill/build$ ./flood object_samp71-utm.pcd
Loading...
Failed to find match for field 'intensity'.
>> Done: 8 ms, 942 points
Downsampling...
>> Done: 0 ms, 16 points
Computing normals...
>> Done: 0 ms
Segmenting to clusters...
>> Done: 0 ms
Saving...
>> Done: 0 ms
abdoul-djalil@adjalil:~/Bureau/Djalil/TPE/flood_fill/build$
```

FIGURE 15 – Application du flood-fill

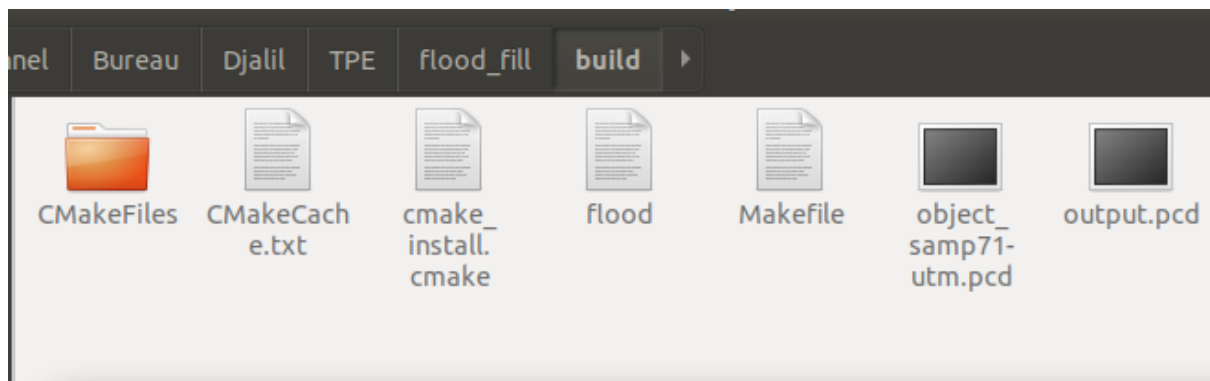


FIGURE 16 – Application du flood-fill

On peut remarquer ici à la fin du traitement de l'algorithme flood-fill la génération d'un fichier **output.pcd**

En les visualisant on aura :

```
abdoul-djalil@adjalil: ~/Bureau/Djalil/TPE/flood_fill/build
Fichier Édition Affichage Rechercher Terminal Aide
abdoul-djalil@adjalil:~/Bureau/Djalil/TPE/flood_fill/build$ pcl_viewer output.pcd
The viewer window provides interactive commands; for help, press 'h' or 'H' from within the window.
> Loading output.pcd [done, 176 ms : 16 points]
Available dimensions: x y z intensity
```

FIGURE 17 – Visualisation du flood-fill

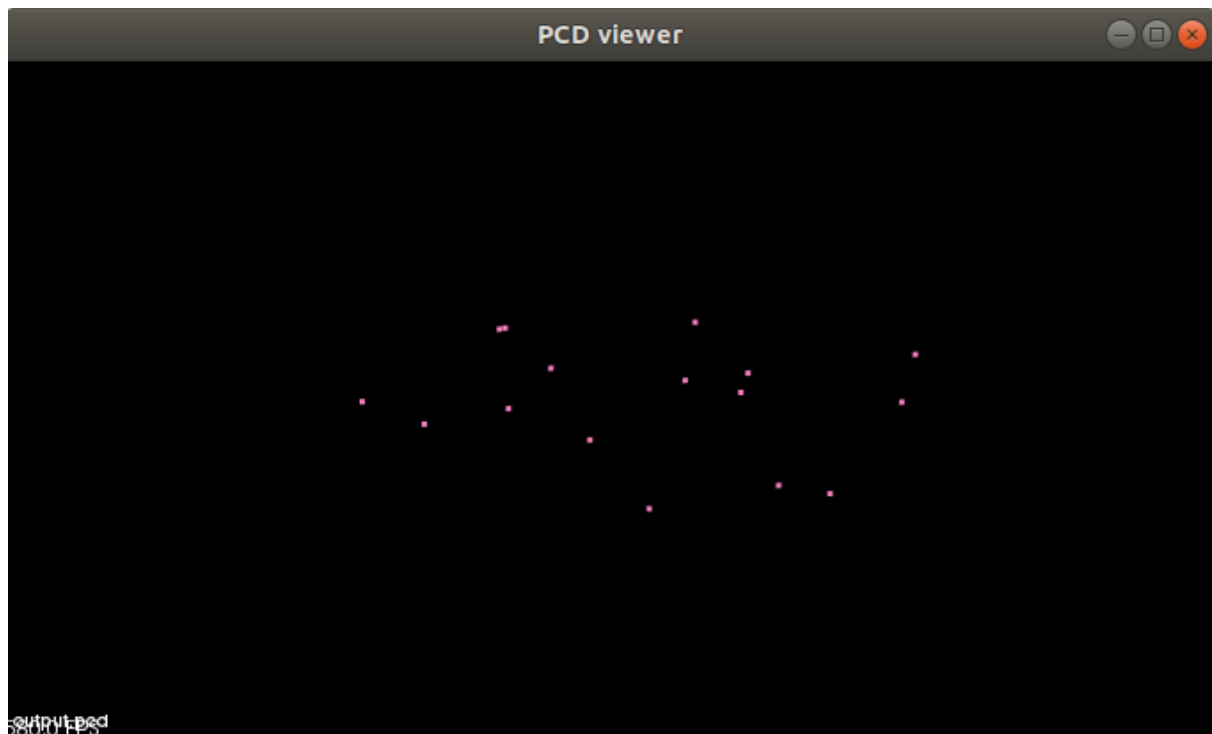


FIGURE 18 – *Visualisation du flood-fill*

On voit que clairement que l'algorithme de remplissant a inspecté les lignes voisines (précédente et suivante) pour rechercher les segments adjacents susceptibles d'être remplis lors d'une prochaine passe.

6 CONCLUSION GÉNÉRALE

Un travail important a été effectué pour segmenter les nuage de points 3D urbains au cours des dernières années. Différentes méthodes ont été développées pour segmenter efficacement ces nuages de points en temps réel. Il ressort clairement des articles examinés que beaucoup d'efforts restent à faire. Cependant, nous nous sommes concentrés uniquement sur la segmentation des sol et des non sol puisqu'il s'agissait des données urbaines. Nous avons, pour cela, implémenté un algorithme de segmentation de sol et ensuite nous avons appliqué l'algorithme flood-fill pour corriger les segments non bien formés. Ce qui nous a donné des résultats assez conséquent bien que nous n'ayons pas pu mesurer l'efficacité de notre algorithme par-rapport à la méthode retenue.

Au terme de ce laborieux travail, nous avons réalisé une analyse superflue du sujet puis nous avons identifié quelques domaines d'application dans lesquels s'inscrit notre sujet ensuite, nous avons fait une étude bibliographique dans laquelle nous avons énuméré quelques travaux existant et enfin à l'issue de cette étude, nous avons pu retenir une méthode qui nous a semblé adéquate à notre thème puis nous avons implémenté la première partie de l'algorithme retenu.

Références

- [1] Shahar Barnea and Sagi Filin. Segmentation of terrestrial laser scanning data using geometry and image information. *ISPRS journal of photogrammetry and remote sensing*, 76 :33–48, 2013.
- [2] Bir Bhanu, Sungkee Lee, Chih-Cheng Ho, and Tom Henderson. Range data processing : Representation of surfaces by edges. In *Proceedings of the Eighth International Conference on Pattern Recognition*, pages 236–238, 1986.
- [3] Edward Jorge Yuri Cayllahua Cahuina, Jean Cousty, Yukiko Kenmochi, Arnaldo De Albuquerque Araujo, and Guillermo Cámara-Chávez. Algorithms for hierarchical segmentation based on the felzenszwalb-huttenlocher dissimilarity. 2018.
- [4] Edward JY Cayllahua Cahuina and Guillermo Camara Chavez. A new method for static video summarization using local descriptors and video temporal segmentation. In *2013 XXVI Conference on Graphics, Patterns and Images*, pages 226–233. IEEE, 2013.
- [5] Phuong Minh Chu, Seoungjae Cho, Yong Woon Park, and Kyungeun Cho. Fast point cloud segmentation based on flood-fill algorithm. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2017 IEEE International Conference on*, pages 656–659. IEEE, 2017.
- [6] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2) :167–181, 2004.
- [7] William Gélard, Ariane Herbulot, Michel Devy, and Philippe Burger. Segmentation de nuages de points 3d pour le phénotypage de tournesols. In *16ième Journées Francophones des Jeunes Chercheurs en Vision par Ordinateur (ORASIS) 2017*, 2017.
- [8] Jorge Hernandez and Beatriz Marcotegui. Segmentation et interprétation de nuages de points pour la modélisation d’environnements urbains. *Revue française de photogrammétrie et de télédétection*, 191 :28–35, 2008.
- [9] Raheleh Kafieh, Hossein Rabbani, and Saeed Kermani. A review of algorithms for segmentation of optical coherence tomography from retina. *Journal of medical signals and sensors*, 3(1) :45, 2013.
- [10] Debasree Mitra, Rupashri Barik, Sanjoy Roy, and S Bhattacharyya. A survey on image segmentation and image registration. In *ACEEE-CPS, International Conference on Computing, Communication & Manufacturing, ISBN*, pages 978–0. Citeseer.
- [11] Anh Nguyen and Bac Le. 3d point cloud segmentation : A survey. In *Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on*, pages 225–230. IEEE, 2013.
- [12] Bisheng Yang, Zhen Dong, Gang Zhao, and Wenxia Dai. Hierarchical extraction of urban objects from mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 99 :45–57, 2015.