

Nuage de points & Modélisation 3D

-

PolarMix: A General Data Augmentation Technique for LiDAR Point Clouds

Rida Lali

IASD 2022/2023 - 2e Semestre

Table des matières

1	Introduction	1
2	PolarMix	1
2.1	Scene level swapping	1
2.2	Instance-level rotate-pasting	2
3	Sur l'article	3
4	Expériences	3
5	Conclusion	4
	Références	5

1 Introduction

L'idée de la segmentation sémantique est d'associer différents nuages de points à une entité ou à un objet que l'on souhaite reconnaître. Faisant appel à des relevés de LiDAR et donc un nombre important de nuage de points, on se retrouve limité par la taille des Dataset ce qui peut limiter le nombre de relevés LiDAR qu'on peut avoir.

L'augmentation de données (ou "Data Augmentation") est une technique couramment utilisée dans l'apprentissage automatique pour augmenter la quantité et la variété des données d'entraînement disponibles pour un modèle. Cette technique consiste à appliquer des transformations aléatoires aux données existantes, telles que des rotations, des translations, des zooms ou des changements de couleurs. En appliquant ces transformations, on peut créer de nouvelles données d'entraînement qui sont similaires aux données d'origine mais qui présentent des variations mineures. Cela permet au modèle d'apprendre à mieux généraliser et à être plus robuste aux variations dans les données.

Utilisée dans la vision par ordinateur, elle permet d'augmenter le nombre d'images disponibles pour l'entraînement d'un modèle. On peut, par exemple, appliquer une rotation aléatoire à une image existante pour créer une nouvelle image qui représente le même objet sous un angle différent. Ainsi, on peut entraîner un modèle à reconnaître l'objet indépendamment de son orientation.

Ainsi, l'article présenté¹ introduit une méthode d'augmentation de données appliquée à des nuages de points capturés par un LiDAR.

2 PolarMix

L'idée apportée par l'article est un algorithme qui permet de découper une scène relevée par un LiDAR et de le combiner avec un second relevé de LiDAR. Deux méthodes sont présentées.

En partant de deux relevés de LiDAR avec leurs labels (s_A, y_A) , (s_B, y_B) avec s_* qui correspond aux coordonnées de chaque point dans chaque relevé et y_* leur label respectif.

Les méthodes utilisées exploitent les coordonnées circulaires, ainsi lors des opérations, on convertit $s_{i,j} = (x, y, z)_{cart} \mapsto (\theta, r, \phi)_{circ}$ avec :

$$\begin{array}{ll} \theta = \arctan(y/x) & \text{l'azimuth} \\ r = \sqrt{x^2 + y^2 + z^2} & \text{la profondeur} \\ \phi = \arccos(z/r) & \text{l'inclinaison} \end{array}$$

Ainsi, suivant la méthode, nous aurons à utiliser potentiellement l'une des coordonnées, ainsi il sera inutile de calculer les 3 autres.

2.1 Scene level swapping

L'idée de cette méthode est de construire un nouveau relevé LiDAR, en se basant sur deux relevés appartenant au Dataset. Ainsi avec une probabilité qui est fixée ainsi que deux bornes α et β d'azimut, on inverse les scènes des deux relevés LiDAR appartenant à un angle entre α et β .

Algorithm 1: Algorithme d'échange de niveau de scène (Scene-level Swapping)

Données: $(s_A, y_A), (s_B, y_B)$ relevés du LiDAR, α, β bornes pour l'azimut

Sorties: (\tilde{s}, \tilde{y}) : nouvelle pièce de donnée

- 1 $\tilde{s}, \tilde{y} = s_A, y_A$;
 - 2 Calcul de l'azimut θ pour $\tilde{s}, s_B \rightarrow \tilde{\theta}, \theta_B$;
 - 3 **Pour chaque** $\tilde{\theta}_i \in \tilde{\theta}$ **faire**
 - 4 **Si** $\alpha \leq \tilde{\theta}_i \leq \beta$ **alors**
 - 5 Supprimer $(\tilde{s}_i, \tilde{y}_i)$ de (\tilde{s}, \tilde{y}) ;
 - 6 **Pour chaque** $\theta_{B,i} \in \theta_B$ **faire**
 - 7 **Si** $\alpha \leq \theta_{B,i} \leq \beta$ **alors**
 - 8 Ajouter $(s_{B,i}, y_{B,i})$ à (\tilde{s}, \tilde{y}) ;
-

2.2 Instance-level rotate-pasting

La deuxième méthode employée dans PolarMix consiste à sélectionner les points d'un relevé LiDAR appartenant à une même classe, de tourner cet ensemble de points avec un angle défini et de l'insérer dans un autre relevé LiDAR pour en créer un nouveau. Ainsi, on s'appuie sur C , une liste de classes sur lesquels on souhaite exercer une rotation et Ω , la liste des angles de rotations associée. On a alors l'algorithme suivant :

Algorithm 2: Algorithme de rotation-collage de niveau d'instance (Instance-level rotate-pasting)

Données: $(\tilde{s}, \tilde{y}), (s_B, y_B)$ relevés du LiDAR, C, Ω listes de classes et d'angles

Sorties: (\tilde{s}, \tilde{y}) : nouvelle pièce de donnée

- 1 $(s_C, y_C) \leftarrow \{(s_{B,i}, y_{B,i}) | y_{B,i} \in C\}$;
 - 2 **Pour chaque** $\omega_j \in \Omega$ **faire**
 - 3 $(\tilde{s}, \tilde{y}) \mapsto (\text{Concat}(\tilde{s}, \mathcal{R}_{\omega_j} s_C), \text{Concat}(\tilde{y}, y_C))$;
-

En résumé, on peut voir les deux méthodes qui constituent PolarMix comme ceci :

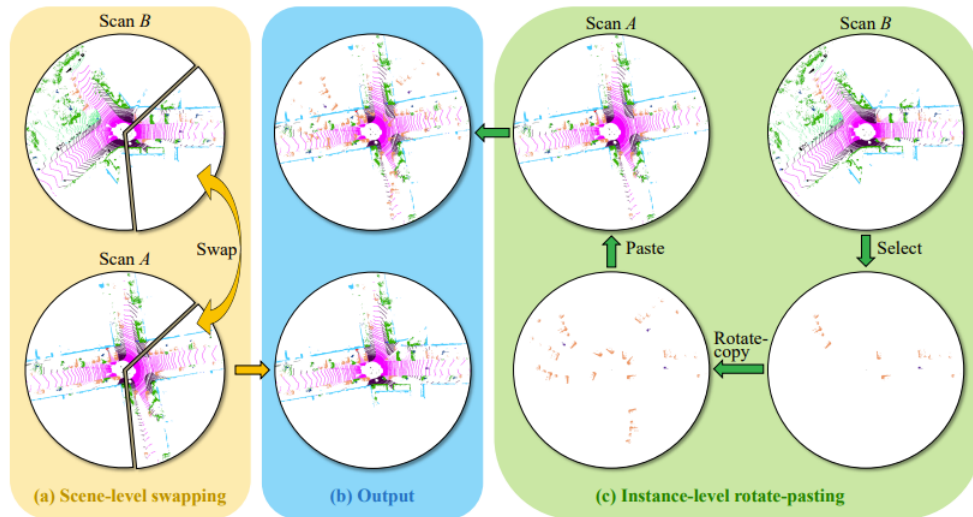


FIGURE 1 – Image venant de l'article PolarMix illustrant les deux méthodes de Data Augmentation en utilisant deux scans issus du Dataset

3 Sur l'article

L'article présente alors une nouvelle méthode d'augmentation de données en exploitant le domaine des relevés LiDAR par découpage et collage, ainsi que superposition de classes issus d'un relevé sur d'autres relevés.

Des tests ont été réalisés par les auteurs dans différents contextes. Le premier consiste à comparer leur méthode avec d'autres méthodes de Data Augmentation pour différents algorithmes MinkNet et SPVCNN. Mais aussi en découpant PolarMix en exploitant les différentes méthodes qui la composent (la section "4.4 Ablation study") montrant ainsi comment les deux méthodes de PolarMix permettent d'améliorer la Data Augmentation.

L'article fournit par ailleurs dans son annexe ([Lien](#)) une étude paramétrique afin de déterminer les paramètres δ et l'écart d'angle qui doit être utilisé pour l'azimut ($\beta - \alpha$).

Cependant, malgré une formalisation rigoureuse de l'algorithme facilitant la compréhension de l'article ainsi que des tests mettant en valeur les résultats de l'algorithme, le repository GitHub vers lequel l'article pointe ne détaille pas les étapes pour bien installer le programme, les versions utilisées demeurent obsolètes et nécessite une réadaptation avec les nouvelles versions de bibliothèques même si le fichier de configuration permet d'avoir des résultats reproductibles (avec une seed définie pour l'entraînement, des critères définies et indiquées pour l'optimisation, ainsi que le nombre d'époques et la taille du batch).

Par ailleurs, l'article se contente de terminer sur un résumé de PolarMix et n'indique pas de pistes éventuelles pour une amélioration ou un changement de paradigme.

4 Expériences

Le texte propose une implémentation de PolarMix agrégée à la méthode de segmentation des nuages de points par Deep Learning.

On a testé alors les programmes cités dans l'article : MinkNet, SPVCNN, RandLA-Net, Cylinder3D³. En raison des problèmes de compatibilité des logiciels proposés par l'article causées par l'obsolescence (versions de CUDA et de Tensorflow inadaptées), on se rabat alors sur une version de Cylinder3D implémentée pour les versions récentes de CUDA ([Lien GitHub](#)).

SemanticKITTI L'article propose un Dataset contenant des relevés LiDAR de capteurs circulant dans une rue nommé SemanticKITTI². Ce Dataset classe les nuages de points selon les objets qu'ils représentent. Ici, nous avons 19 classes d'objets allant du simple panneau sur la route aux bâtiments. Ce dernier est accessible via ce [lien](#).

En raison de la taille du Dataset (80 Go) proposé par l'article, on a réduit la taille de ce dernier en extrayant une partie du Dataset dédiée à l'entraînement et une partie dédiée au test, ce qui peut expliquer un pourcentage de 0% pour certaines classes. La partie du Dataset servant à la validation est conservée.

mIOU Afin d'évaluer les performances d'un modèle en segmentation sémantique, on définit la notion de mIoU (Intersection-Over-Union) qui fait le rapport suivant :

$$IoU = \frac{\text{Aire de superposition correcte}}{\text{Aire de l'union entre l'aire prédite et l'aire réelle}}$$

ce qui permet de mieux quantifier quelles sont les parties les mieux reconnues sémantiquement. Cette unité est utilisée dans de nombreux articles dédiés au domaine.

Expérimentations faites Par ailleurs, en raison des temps de calculs disponibles, on a pu entraîner le modèle sur quelques epochs (ici 5). Néanmoins, on constate déjà des résultats intéressants (cf. Table 1) avec l’utilisation de PolarMix comparé à la méthode basique d’augmentation de données de Cylinder3D.

	Cylinder3D + Basic DA	Cylinder3D + PolarMix
Car	94.10	91.45
Bicycle	17.33	34.40
Motorcycle	31.23	35.50
Truck	0.68	17.26
Bus	30.22	4.33
Person	43.13	37.60
Bicyclist	9.77	18.34
Motorcyclist	0.00	0.00
Road	89.81	91.48
Parking	21.65	6.97
Sidewalk	72.05	71.97
Other-ground	0.03	0.00
Building	86.87	83.83
Fence	49.09	34.85
Vegetation	88.50	85.69
Trunk	59.73	63.03
Terrain	74.88	67.80
Pole	61.44	63.49
Traffic-sign	43.41	41.21
mIoU	45.996	44.695
Loss	0.827	0.920

TABLE 1 – Tableau des résultats des modèles pour chaque catégorie.

Avec plus d’épisodes, on pourrait comparer les résultats pour chacune des méthodes afin de confronter entre les deux méthodes d’augmentation de données.

5 Conclusion

Ainsi, ce projet nous a permis de nous familiariser avec une méthode d’augmentation de données pour la segmentation et la classification de nuages de points. PolarMix étant une méthode en deux grandes étapes : l’échange de scène et le coupage/rotation d’instances, elle permet d’exploiter les caractéristiques des datasets sur lesquels les méthodes de segmentation par Deep Learning (tels que SPVCNN, MinkNet, Cylinder3D) sont entraînées afin de permettre à ces modèles de mieux cerner les caractéristiques des objets appartenant à une même classe (notamment via l’Instance-level rotate-pasting) dans des contextes différents (via le Scene-level Swapping).

Références

- [1] Aoran Xiao, Jiaying Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, Ling Shao, *PolarMix : A General Data Augmentation Technique for LiDAR Point Clouds*. arXiv, [2208.00223](#), 2022.
- [2] J. Behley and M. Garbade and A. Milioto and J. Quenzel and S. Behnke and C. Stachniss and J. Gall, *SemanticKITTI : A Dataset for Semantic Scene Understanding of LiDAR Sequences*, [Proc. of the IEEE/CVF International Conf. on Computer Vision \(ICCV\)](#), 2019.
- [3] Hui Zhou and Xinge Zhu and Xiao Song and Yuexin Ma and Zhe Wang and Hongsheng Li and Dahua Lin, *Cylinder3D : An Effective 3D Framework for Driving-scene LiDAR Semantic Segmentation*, arXiv, [2008.01550](#), 2020.