Détection et localisation d'objets 3D par apprentissage profond en topologie capteur

Pierre BIASUTTI[1,2], Aurélie BUGEAU[1], Jean-François AUJOL[2], Mathieu BRÉDIF[3]

[1]Univ. Bordeaux, LaBRI, INP, CNRS, UMR 5800, F-33400 Talence, France

[2]Univ. Bordeaux, IMB, INP, CNRS, UMR 5251, F-33400 Talence, France

[3]Univ. Paris-Est, LASTIG GEOVIS, IGN, ENSG, F-94160 Saint-Mandé, France

{pierre.biasutti, bugeau}@labri.fr, jaujol@math.u-bordeaux.fr, mathieu.bredif@ign.fr

Résumé – Ce travail présente une nouvelle méthode pour la détection et la localisation d'objets dans des scènes 3D LiDAR acquises par des systèmes de cartographie mobile. Ce problème est généralement traité en discrétisant l'espace 3D en une fine grille de voxels. Nous introduisons une approche alternative ne nécessitant pas de discrétisation. Elle est basée sur la représentation en 2D du nuage de points en topologie capteur. Cette image sert d'entrée à un réseau de neurones convolutionnels qui en extrait les informations 3D des objets. La répresentation en topologie capteur présentant des ambiguïtés dans le fond de la scène, nous améliorerons les résultats de détection en couplant ce modèle avec un réseau de détection 2D d'objets sur une image optique. Les prédictions des deux réseaux sont finalement fusionnées pour obtenir les détections finales.

Abstract – This work proposes a novel approach for detection and localisation of objects in 3D LiDAR scenes aquired via Mobile Mapping Systems. While this task is often treated on a voxel grid representations of the point cloud, our method offers to use the point cloud in sensor topology, thus avoiding a discretisation step. This representation of the point cloud is used as an input for a CNN that extracts 3D positions and dimensions of objects in the scene. As far objects in the scene tends to be mixed with the background when seen in the sensor topology, we offer to enhance the 3D detection by fusing the 3D predictions with 2D object detections performed on optical images.

1 Introduction

Avec l'intérêt grandissant pour les véhicules autonomes, la cartographie 3D et la robotique, la conception de systèmes de perception embarqués est devenue un enjeu capital de la vision par ordinateur. En particulier, la détection et la localisation 3D d'objets est cruciale pour que les systèmes autonomes puissent percevoir les objets présents dans la scène. La plupart des systèmes de conduite autonome sont équipés de différents capteurs (optiques et LiDAR 3D). Tous ces capteurs peuvent être combinés pour permettre une détection précise et robuste des objets en 3D. Ces dernières années, la détection d'objets 2D basée sur des images optiques a vu des améliorations considérables [15, 7, 10, 13, 8]. En comparaison, les systèmes de détection 3D n'atteignent pas encore les mêmes performances en terme de précision, de robustesse et de temps de calcul.

La détection 3D dans des nuages de points LiDAR à récemment été le sujet de nombreux travaux grâce à l'apprentissage profond. La plupart des travaux se basent sur une représentation discrète du nuage de points : par projection verticale des points sur une grille horizontale [11, 17] parfois accompagnée de modalités annexes comme des images optiques [2, 6], ou sous la forme d'une grille de voxels [19]. D'autres articles divisent le nuage de points en régions d'intêret grâce à une première étape de détection 2D optique, puis estiment les informations 3D de chaque objet dans les sous-nuages de points [12] (*e.g.* la portion de nuage dont la projection en domaine image intersecte la détection 2D) ou sur l'ensemble des points [16]. Toutes ces méthodes permettent une détection 3D efficace des objets de la scène, mais requièrent des architectures très lourdes notamment pour gérer le nuage de points à une échelle suffisamment fine (plusieurs millions de voxels sur une scène du dataset KITTI [5] pour une résolution de 0.1m par voxel).

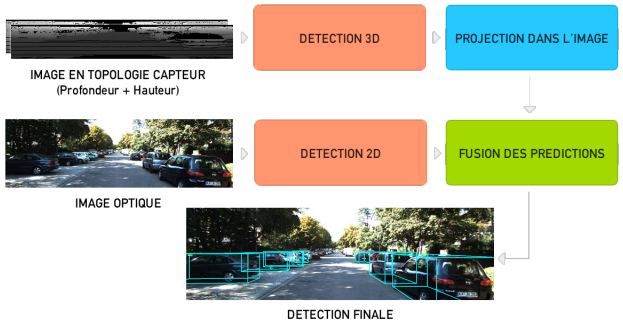

FIGURE 1 – Chaîne de traitement proposée.

Ce travail propose une nouvelle méthode pour la détection et la localisation d'objets en 3D basé sur la représentation 2D du nuage de points en topologie capteur. Cette représentation permet l'utilisation de réseaux de neurones convolutionels simples, en adaptant une architecture developpée pour la détection 2D sur images optiques pour la détection et la localisation 3D LiDAR. Pour affiner la détection 3D, notre méthode fusionne les prédictions 3D LiDAR avec des prédictions 2D issues d'un détecteur 2D optique. Notre modèle, résumé en Figure 1, permet

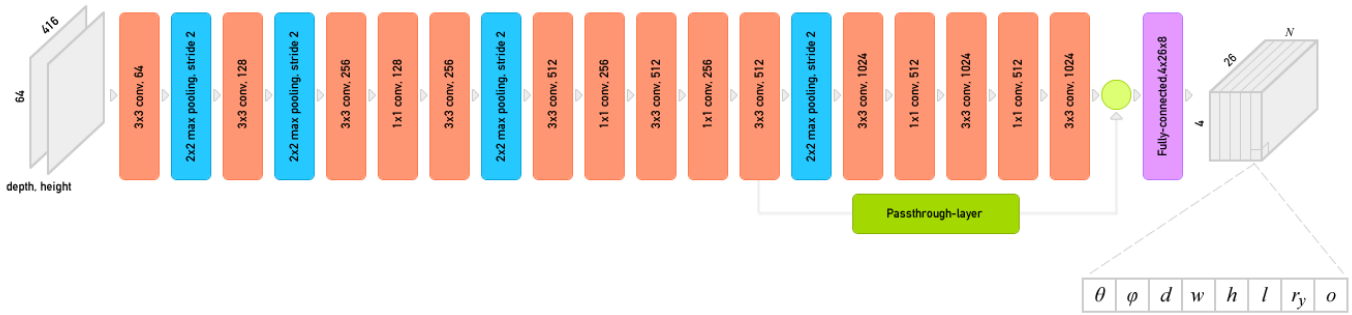

FIGURE 2 – Architecture du réseau de détection et localisation 3D.

d'effectuer la détection et la localisation 3D en temps-réel tout en utilisant peu de ressources, ce qui est indispensable pour des systèmes embarqués.

2 Méthodologie

Chaque étape de la chaîne de traitement (Figure 1) est détaillée dans la suite de cette section.

2.1 Détection et localisation 3D

Les capteurs LiDAR modernes font généralement l'acquisition de points 3D en suivant une structure régulière, de laquelle on peut dériver une image dense [1]. En effet, chaque point étant défini par deux angles et une distance, resp. (θ, ϕ, d), avec un pas de $(\Delta\theta, \Delta\phi)$ entre deux positions successives du capteur, on peut associer une modalité de chaque point p du nuage avec un pixel aux coordonnées (x, y) où $x = \left\lfloor \frac{\theta}{\Delta\theta} \right\rfloor, y = \left\lfloor \frac{\phi}{\Delta\phi} \right\rfloor$. On obtient une image dans laquelle chaque pixel contient, par canal, une modalité du point 3D qu'il représente. La Figure 3 montre un exemple de nuage de points (haut) issue du dataset KITTI [5] ainsi que sa représentation en topologie capteur (TC) pour la distance (milieu) et l'élevation (bas).

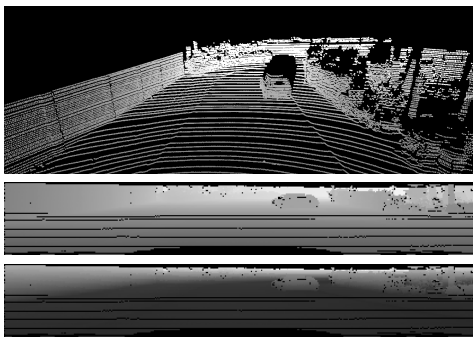

FIGURE 3 – Nuage de points vu en 3D (haut) et en topologie capteur (distance au capteur, au milieu et élévation, en bas).

La première étape de notre chaîne de traitement consiste à prédire les coordonnées 3D ainsi que les dimensions et l'orientation de boîtes englobant les objets à partir d'une image TC à deux canaux : distance logarithmique au capteur (pour compenser l'écart entre les lignes de scan) et élévation. Nous adaptons pour cela le modèle YOLO9000 [13], initialement prévu pour la détection d'objets 2D dans une image optique. L'architecture en pyramide de ce réseau, comparable à un encodeur, estime la présence d'objets dans chaque case de la couche la plus basse. Chaque case est divisée en un nombre N d'objets potentiels, initialisés à différentes dimensions. Pour chaque objet potentiel, on estime l'*objectness* $o \in [0, 1]$ qui indique la probabilité que la case contienne un objet.

La prédiction des coordonnées 3D d'un objet depuis l'image TC est similaire à la prédiction 2D, accompagnée de la prédiction de la profondeur (Figure 2). En effet, comme présenté plus haut, la position d'un pixel de l'image TC correspond directement aux angles d'acquisition du capteur LiDAR. On cherche donc à prédire les angles (θ, ϕ) ainsi que d la distance au capteur pour chaque objet.

En 2D, la perspective oblige à initialiser N objets potentiels à différentes échelles pour compenser les variations de taille en fonction de la distance. En 3D, la dimension d'un objet ne varie pas selon sa distance au centre d'acquisition. Pour chaque classe d'objet, on définit (H, W, L) comme les dimensions moyennes des objets. On cherche ensuite à prédire les coefficients (h, w, l) tels que $(h * H, w * W, l * L)$ soient égaux aux dimensions de l'objet à détecter.

La plupart des challenges de détection 3D en milieu urbain [4, 5] ne considèrent que la rotation des objets sur l'axe vertical (yaw) et ignorent la rotation de l'objet sur les autres axes (pitch, roll) car l'objet est supposé se situer sur le sol proche du plan horizontal. La représentation du nuage de points en topologie capteur correspond à une projection panoramique de la scène. Ainsi, deux objets de même rotation dans la scène 3D vont apparaître différemment dans l'image TC. Il est donc nécessaire de prendre en compte l'angle θ lors de la prédiction d'un objet. On définit r_y la rotation d'un objet sur l'axe vertical vu depuis l'image TC, comme illustré Figure 4.

Entrainement Soit $\mathcal{F} = \{\theta, \phi, d, w, h, l, o\}$ l'ensemble des caractéristiques représentant un objet, à l'exception de la rotation. Pour chaque case $c \in H \times W \times A$ contenant un objet dans la vérité terrain, on cherche à optimiser la fonction de coût suivante :

$$\sum_{f \in \mathcal{F}} \lambda_f \|f(c) - \hat{f}(c)\|_2^2 + \lambda_{r_y}[1 - cos(r_y(c) - \hat{r}_y(c)]$$

où $\hat{x}$ dénote la valeur dans la vérité terrain, λ_x la pondération de chaque caractéristique. Le réseau est entrainé avec l'algorithme

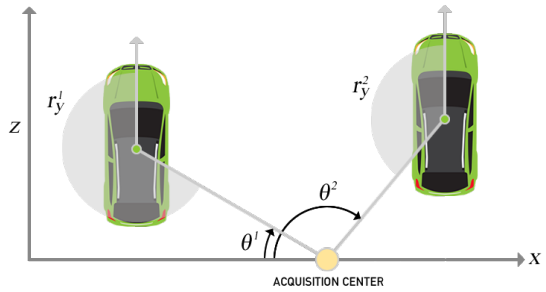

FIGURE 4 – Illustration de la rotation de deux objets dans une scène pour deux angles θ différents.

d'optimisation Adam et un pas de temps de 0.001, avec des lots de taille 64 et 10 époques. On utilise la base de détection 3D KITTI [5] contenant 7481 exemples d'entrainement.

Ambiguïté des objets distants Du fait de la faible résolution du capteur, peu de points sont acquis sur les objets situés loin du capteur. Par conséquent, la différence entre ces objets et le fond de la scène est ambigüe en topologie capteur. Un exemple de d'ambiguïté est montré Figure 5. Pour palier ce problème d'ambiguïté, on propose de coupler la détection 3D avec une détection 2D optique comme présenté ci-après.

FIGURE 5 – Exemple d'ambiguïté dans le fond de la scène. Les deux prédictions sont très similaires (en rouge), mais une seule correspond réellement à un objet (ici, une voiture, en vert).

2.2 Détection 2D optique

Les méthodes récentes de détection d'objets dans des images optiques 2D [15, 3, 7, 13, 10, 8] atteignent d'excellents scores sur les challenges de référence. Certaines ont été développées spécifiquement pour la détection en milieu urbain [18, 14]. Pour rendre la détection 3D plus robuste aux ambiguïtés du fond de la scène, nous détectons également les objets en 2D dans des images optiques associées au nuage de points. On utilise ici la version pré-entrainée de YOLO9000 [13] sur la base de données COCO [9], du fait de la disponibilité du code et des poids pré-entrainés, et de ses performances comme montré Figure 6 sur une image de la base KITTI.

FIGURE 6 – Détection 2D sur une image de KITTI avec YOLO9000 pré-entrainé sur la base [9]. On voit que la détection (rouge) est très proche de la vérité terrain (vert).

2.3 Projection et fusion des détections

La détection 3D permet ensuite de calculer les coordonnées 3D des 8 coins de la boîte englobante de chaque objet. La plupart des dispositifs de scan urbains fournissent les paramètres de calibration précis du système d'acquisition. À partir de ces informations, les 8 coins de chaque objet détecté sont projetés dans l'image optique, puis le rectangle de taille minimale b_{3D} contenant ces 8 coins projetés est estimé. On définit aussi b_{2D} le rectangle prédit par le détecteur 2D sur l'image optique. On considère alors qu'une détection 3D est valide si sa projection dans l'image optique intersecte suffisamment une des détections issues du détecteur 2D. Une détection 3D et sa projection b_{3D} dans l'image sont donc valides si $\mathrm{valid}(\mathrm{b}_{3D}) > 0$ avec :

$$\mathrm{valid}(\mathrm{b}_{3D}) = \sum_{\mathrm{b}_{2D} \in B_{2D}} S(\mathrm{b}_{3D}, \mathrm{b}_{2D})$$

$$S(\mathrm{b}_{3D}, \mathrm{b}_{2D}) = \begin{cases} 1 & \text{si } \frac{|\mathrm{b}_{3D} \cap \mathrm{b}_{2D}|}{|\mathrm{b}_{3D} \cup \mathrm{b}_{2D}|} > \mathrm{t} \\ 0 & \text{sinon.} \end{cases}$$

où B_{2D} est l'ensemble des détections issues du détecteur 2D et t le seuil d'intersection sur union pour considérer qu'une détection 3D et une détection 2D correspondent au même objet.

3 Résultats

La Figure 7 montre des résultats de notre méthode pour la détection 3D de voitures. On observe que les détections 3D (projetées en bleu dans l'image optique) coïncident bien avec la vérité terrain (en vert). De plus, on observe que notre méthode peut détecter des objets proches les uns des autres ainsi que loin du capteur, grâce au couplage avec la détection 2D. Le Tableau 1 présente différents scores de précision des détections par rapport à la vérité terrain. On constate que pour les objets détectés, la précision moyenne selon chaque indicateur est très élevée.

	Score
Distance moyenne (3D)	0.53m
Distance moyenne (Profondeur)	0.51m
IoU 3D moyenne	63%
Erreur angulaire moyenne	9.13 deg

TABLE 1 – Précision moyenne des détections 3D par rapport à la vérité terrain.

Néanmoins, comme le montre le Tableau 2, certains objets ne sont pas détectés par notre méthode alors qu'ils le sont par des méthodes de l'état-de-l'art. La méthode d'évaluation utilisée est celle de [4]. Cela est dû à la difficulté d'estimer à la fois les informations 3D de chaque objet et d'estimer l'*objectness o* sur l'image TC. De plus, le détecteur 2D utilisé dans nos expériences n'atteint pas des scores aussi élevés que l'état-de-l'art [16], ce qui affecte directement les performances de notre méthode. Lorsque plusieurs objets ne sont pas détectés, le score de

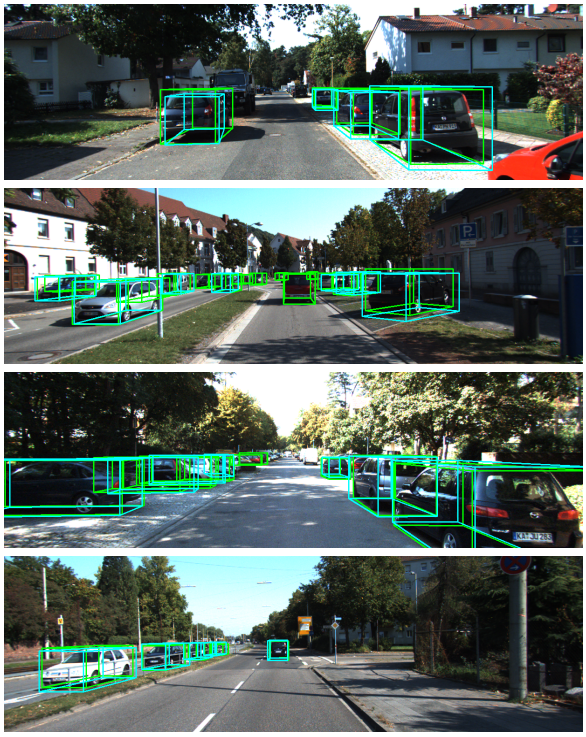

FIGURE 7 – Exemple de détections 3D (en bleu) par notre pipeline sur des scènes de la base KITTI et vérité terrain (en vert).

mAP pour la détection 3D est grandement affecté. En revanche, on peut voir que la fréquence de détection de notre méthode est bien plus élevée que celle de l'état-de-l'art.

Score KITTI [5]	**Notre méthode**	**Etat de l'art [16]**
Détection 2D	63.67%	89.32%
Détection 3D	10.43%	75.76%
Orientation	29.87%	89.22 %
FPS *(GPU)*	312fps	10fps

TABLE 2 – Score de notre méthode contre l'état de l'art [16] sur le challenge KITTI [5]. Les mesures sont données en mAP avec un seuil de 0.7 d'intersection entre prédiction et vérité terrain.

4 Conclusion

Cet article présente une nouvelle méthode pour la détection et la localisation d'objets en 3D à partir d'images TC. La combinaison avec une méthode de détection 2D optique permet de lever l'ambiguïté dans les détections 3D dans le fond de la scène. Notre méthode prédit très précisément la localisation et les dimensions des objets. La détection d'objets en 3D donne des résultats satisfaisant mais imparfaits du fait de la difficulté de détecter des objets sur les images TC. À l'avenir, nous souhaitons tester une version multi-tâches de la sortie du notre réseau pour séparer la détection de la localisation, permettant ainsi d'allouer plus de neurones pour la détection d'objets sur l'image en TC avec un faible impact sur le temps de calcul.

Remerciement

Ce travail a bénéficié d'une aide du programme de Recherche et Innovation European Union's Horizon 2020 au titre de la bourse Marie Skłodowska-Curie (No 777826).

Références

[1] P. Biasutti, J-F. Aujol, M. Brédif, and A. Bugeau. Range-Image : Incorporating sensor topology for LiDAR point cloud processing. *Photogram. Eng. & Remote Sensing*, 84(6).

[2] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3D object detection network for autonomous driving. In *IEEE Conf on Comp Vis and Pat Rec*, 2017.

[3] J. Dai, Y. Li, K. He, and J. Sun. R-FCN : Object detection via region-based fully convolutional networks. In *Advances in Neural Inf. Proc. Sys.*, 2016.

[4] M. Everingham, Luc Van G., C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) challenge. *Int Jour of Comp Vis*, 88(2).

[5] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving ? The KITTI Vision Benchmark Suite. In *IEEE Conf on Comp Vis and Pat Rec*, 2012.

[6] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander. Joint 3D proposal generation and object detection from view aggregation. In *IEEE Int. Conf. on Intel. Robots and Systems*, 2018.

[7] T-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *IEEE Conf on Comp Vis and Pat Rec*, 2017.

[8] T-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *IEEE Conf on Comp Vis and Pat Rec*, 2017.

[9] T-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO : Common objects in context. In *Euro Conf on Comp Vis*, 2014.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C-Y. Fu, and A. C. Berg. SSD : Single shot multibox detector. In *Euro Conf on Comp Vis*, 2016.

[11] W. Luo, B. Yang, and R. Urtasun. Fast and furious : Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In *IEEE Conf on Comp Vis and Pat Rec*, 2018.

[12] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3D object detection from RGB-D data. In *IEEE Conf on Comp Vis and Pat Rec*, 2018.

[13] J. Redmon and A. Farhadi. YOLO9000 : Better, Faster, Stronger. In *IEEE Conf on Comp Vis and Pat Rec*, 2017.

[14] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y-W. Tai, and L. Xu. Accurate single stage detector using recurrent rolling convolution. In *IEEE Conf on Comp Vis and Pat Rec*, 2017.

[15] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN : Towards real-time object detection with region proposal networks. *IEEE trans on Pat Anal and Mach Intel*, 39(6).

[16] S. Shi, X. Wang, and H. Li. PointRCNN : 3D object proposal generation and detection from point cloud. *arXiv preprint :1812.04244*, 2018.

[17] Y. Yan, Y. Mao, and B. Li. Second : Sparsely embedded convolutional detection. *Sensors*, 18(10).

[18] F. Yang, W. Choi, and Y. Lin. Exploit all the layers : Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. In *IEEE Conf on Comp Vis and Pat Rec*, 2016.

[19] Y. Zhou and O. Tuzel. Voxelnet : End-to-end learning for point cloud based 3D object detection. In *IEEE Conf on Comp Vis and Pat Rec*, 2018.