Détection de véhicules en temps réel sur grilles d'occupation par des méthodes d'apprentissage profond

Nils DEFAUW[1], Olivier ANTONI[1], Marielle MALFANTE[1], Tiana RAKOTOVAO[1], Suzanne LESECQ[2]

[1]Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France

[2]Univ. Grenoble Alpes, CEA, Leti, F-38000 Grenoble, France

Firstname.Lastname@cea.fr

Résumé – Les grilles d'occupation sont un modèle d'environnement communément utilisé pour représenter l'environnement direct autour d'un véhicule sous forme d'une grille. Cette étude évalue la possibilité de détecter des véhicules sur des grilles d'occupation en s'inspirant de l'état de l'art en traitement d'images. Les architectures développées produisent des résultats de détection précis avec un temps de prédiction permettant une exécution en temps réel.

Abstract – Occupancy grids are a common environment model for representing as a grid the environment around a vehicle. This study tests the possibility of detecting vehicles on occupancy grids by using state of the art models for image processing. The developed architectures produce precise detection results in a prediction time allowing real time execution.

1 Introduction

L'automatisation de la conduite dans le cadre du véhicule autonome est un défi de recherche majeur de notre temps. Un tel système doit conjuguer d'une part très haute précision car une erreur peut mener à des accidents graves et d'autre part être en mesure de s'exécuter en temps réel sur des calculateurs embarqués à puissance de calcul limitée. Dans ce contexte, la détection des autres véhicules présents sur la chaussée revêt une importance particulière.

Pour permettre la localisation précise dans l'espace des obstacles, un véhicule autonome est équipé de capteurs de distance (« range sensors ») tels que des lidars, des radars, des sonars ou encore des caméras stéréoscopiques. Ces capteurs ont pour particularité leur capacité à positionner les obstacles dans l'espace via la mesure de leur distance et de leur position angulaire par rapport au capteur.

Chacun de ces types de capteur a ses propres avantages et ses propres limites, les caméras stéréoscopiques par exemple ne fonctionnent que de jour tandis que les lidars voient leur acuité dégradée en cas de pluie. Pour permettre une fiabilité maximale en toute circonstance, ces capteurs sont généralement utilisés en association sur les véhicules autonomes. Un procédé de fusion de données est ainsi nécessaire pour assembler les mesures provenant des différents capteurs et produire un modèle unifié de l'environnement autour du véhicule. Le modèle d'environnement de la grille d'occupation [3] développé dans ce cadre permet de fusionner les mesures de différents capteurs et prend en compte leurs bruits et incertitudes via l'usage d'un mécanisme probabiliste.

Le cadre dans lequel se place cette étude est celui de la conception de détecteurs d'objets précis et temps réel prenant en entrée des grilles d'occupation. En particulier, nous proposons d'exploiter le format matriciel des grilles d'occupation pour le développement de détecteurs d'objets sur ces grilles inspirés de l'état de l'art en détection d'objets sur image.

Des modèles existants de détection d'objets utilisent des grilles en vue de dessus construites à partir de données lidar [6, 2]. Ces grilles qui sont spécifiques aux capteurs lidar sont différentes des grilles d'occupation utilisées ici qui peuvent être construites à partir de divers capteurs de distance.

Les détecteurs proposés permettent la détection précise des véhicules présents sur la grille d'occupation via la prédiction de leur position, dimensions et orientation le tout en temps-réel par rapport à la fréquence de la production des données au niveau des capteurs.

2 État de l'art

2.1 Grilles d'occupation

Le modèle d'environnement de la grille d'occupation a été introduit en 1985 [3] et est progressivement devenu un standard dans les applications de robotique. Ce formalisme (un exemple est donné sur la Figure 1) permet de représenter l'environnement de manière discrétisée en vue de dessus (« bird's eye view ») sous la forme d'une matrice dont chaque cellule, de taille $10\text{cm} \times 10\text{cm}$ par exemple, contient une probabilité d'occupation dans l'intervalle $[0, 1]$. Cette probabilité d'occupation quantifie la certitude quant à l'occupation de la cellule entre 0 signifiant que la cellule est vide et 1 qu'elle est occupée.

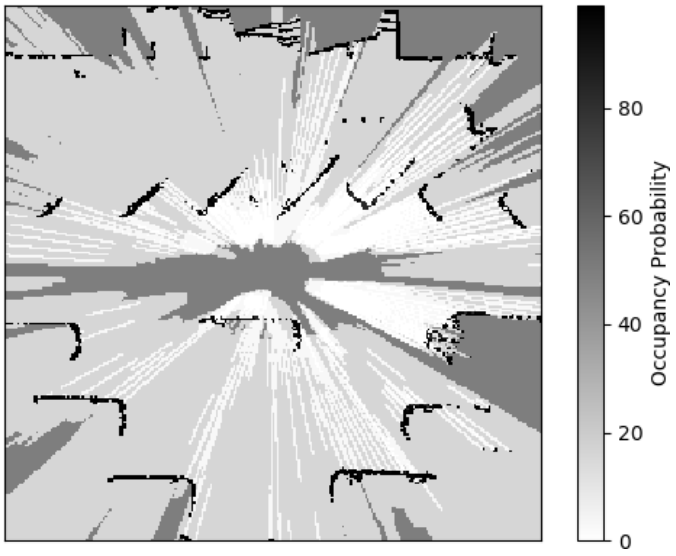

FIGURE 1 – Grille d'occupation dont les probabilités d'occupation sont représentées en niveaux de gris

La fusion des mesures de multiples capteurs via les grilles d'occupation se fait en deux étapes. D'abord, une grille d'occupation est construite pour chaque capteur en utilisant un modèle de bruits de mesure adapté [1, 3]. Ensuite, les grilles d'occupation par capteur sont fusionnées en une unique grille en appliquant un procédé de fusion bayésienne. Pour une cellule donnée, le processus de fusion bayésienne vient renforcer ou mitiger les probabilités d'occupation des différentes grilles fusionnées selon que ces valeurs sont concordantes ou divergentes, ce qui permet de synthétiser les informations extraites des différents capteurs tout en prenant en compte leur incertitude.

En pratique, une grille d'occupation peut contenir des millions de cellules. L'implémentation se fait généralement sur des architectures parallèles comme des GPUs pour atteindre une exécution en temps réel. Il a été montré [1] qu'il est possible de fusionner des capteurs en grille d'occupation de manière très efficace en utilisant uniquement de l'arithmétique entière.

2.2 Détecteurs d'objets sur image

Du point de vue informatique, une grille d'occupation est une matrice 2D dont les éléments sont les probabilités d'occupation de chaque cellule. Cette structure de données est très similaire à la structure matricielle des pixels d'une image. La présente étude propose ainsi de s'inspirer des détecteurs d'objets sur image pour développer un détecteur d'objets sur grilles d'occupation en exploitant l'aspect matriciel commun à ces deux objets.

Le champ scientifique de la détection d'objets sur image a été largement étudié lors de la dernière décennie. Les détecteurs d'objets sur image peuvent être catégorisés en deux classes. Les détecteurs à deux étapes (« two-stage detectors ») produisent les résultats les plus précis mais de par leur architecture en deux étapes ont un temps d'inférence élevé ne répondant pas à l'objectif temps réel de cette étude. Les détecteurs à une étape (« one-stage detectors ») sont composés d'un unique réseau de neurones exécuté une seule fois par image et offrent une précision élevée tout en ayant démontré la possibilité de les exécuter en temps réel sur un flux vidéo. Un représentant important de cette famille de détecteurs est YOLOv2 [4] qui permet de détecter des objets via l'utilisation d'un réseau convolutionnel (« fully convolutional network »).

Ce réseau de neurones prend en entrée une image et produit en sortie une matrice 8×8 qui représente une subdivision de l'image d'entrée. Chaque cellule de cette matrice contient un *tuple* $(s, x, y, \log w, \log h)$ qui représente une boîte englobante : x et y sont les coordonnées du centre de cette boîte englobante dans la cellule, w et h sont ses dimensions et s est un score de confiance dans l'intervalle $[0, 1]$ qui représente le niveau de confiance associé à cette détection. Un seuillage des 64 boîtes englobantes prédites selon leur score de confiance permet d'obtenir les détections d'objets sur l'image. En parallèle, une classe est prédite pour chaque cellule de sortie du détecteur pour catégoriser l'objet détecté. L'architecture utilisée est une succession de couches convolutionnelles dont le nombre de filtres augmente avec la profondeur de la couche. Ces couches sont entrecoupées de couches de sous-échantillonnage par valeur maximale (« max pooling ») diminuant les dimensions des tenseurs d'un facteur 2 à chaque passage. La fonction de coût utilisée est $\mathcal{L} = \mathcal{L}_{classification} + \mathcal{L}_{regression}$ avec $\mathcal{L}_{classification}$ l'entropie croisée binaire sur le score de confiance s et $\mathcal{L}_{regression}$ le coût L1 lissé (« smooth L1 ») sur $(x, y, \log w, \log h)$ pour les cellules contenant une boîte englobante vérité terrain.

3 Méthode

Cette section présente trois architectures de détection de véhicules sur grilles d'occupation inspirées de l'état de l'art.

3.1 Détecteur 8x8

L'architecture interne de ce détecteur (Figure 2) est identique à celle de YOLOv2 à la différence que le nombre de filtres de chaque couche convolutionnelle est divisé par deux par rapport au détecteur original ce qui permet d'obtenir un réseau de neurones réduit contenant 4 961 463 paramètres entraînables et qui devrait par conséquent offrir un temps d'inférence faible. En revanche la tâche de détection n'étant pas la même il est nécessaire de modifier la sortie du détecteur. Étant donné que le seul type d'objet à détecter sont les véhicules, il n'est plus nécessaire de prédire une classe par cellule de sortie. Ensuite, contrairement aux objets génériques observables sur image les véhicules présents sur une grille d'occupation ont une orientation bien définie qu'il est utile de prédire. Il est donc rajouté deux filtres de sortie à la dernière couche convolutionnelle qui prédisent $(\cos\theta, \sin\theta)$, dont on extrait l'angle d'orientation θ à valeurs dans l'intervalle $[-\pi, \pi]$ grâce à la fonction $\mathrm{atan2}$.

3.2 Détecteur 16x16

La subdivision 8×8 force le détecteur à ne proposer au maximum qu'une seule boîte englobante par cellule. Les grilles d'occupations utilisées dans ce travail couvrent une surface de 25.6 mètres de côté, ce qui donne au maximum un véhicule détecté par zone de 3.2 mètres de côté. Pour prévenir l'absence de détection de véhicules rapprochés dans des contextes de trafic dense, nous introduisons une modification du détecteur 8x8 pour produire une matrice de détection en sortie 16×16 via la suppression de la dernière couche de sous-échantillonnage par valeur maximale dans l'architecture du réseau (indiquée dans la Figure 2). Cette modification permet de détecter au maximum un véhicule par zone de 1.6 mètres de côté, ce qui réduit la possibilité de non-détection de véhicules rapprochés.

3.3 Détecteur 16x16-NMS

Le passage à une matrice de sortie 16×16 permet la détection de plus de véhicules mais peut introduire des doublons dans la détection, c'est-à-dire plusieurs boîtes englobantes prédites pour le même véhicule. Pour corriger ce problème, nous proposons l'introduction d'un post-traitement pour supprimer les doublons connu sous le nom de « Non-Maximum Suppression » (NMS).

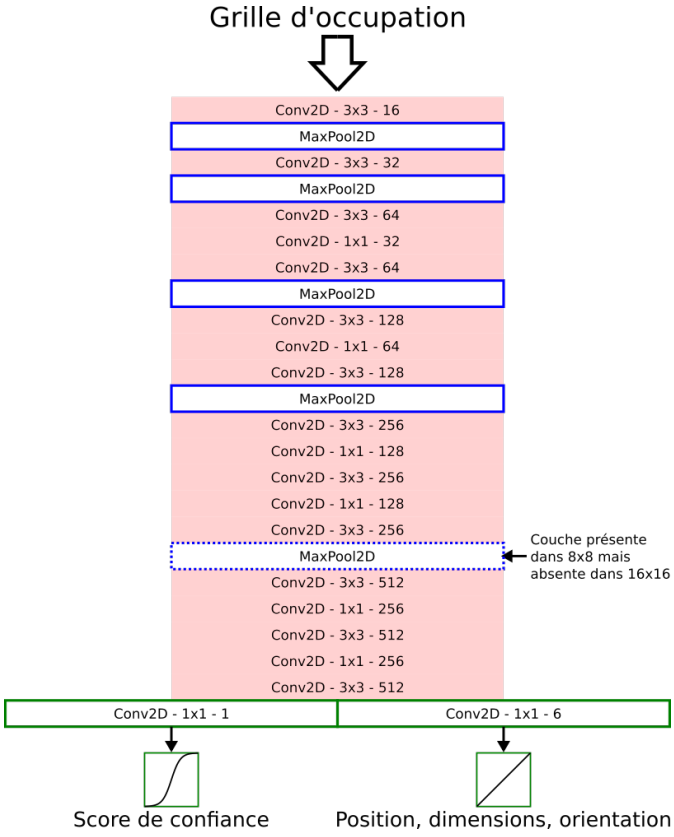

FIGURE 2 – Architecture des détecteurs ; Le post-traitement NMS intervenant après l'inférence n'est pas représenté

4 Résultats

Les architectures étudiées sont entraînées sur le jeu de données Waymo Open [5] à l'aide de l'algorithme d'optimisation Adam avec un taux d'apprentissage de 0.001.

La Figure 3 montre un exemple de prédictions faites par les trois détecteurs sur une même grille d'occupation. On y observe des détections plus précises pour le détecteur 16x16 comparativement au détecteur 8x8 au prix d'un véhicule détecté deux fois, qui est éliminé par le post-traitement NMS du détecteur 16x16-NMS.

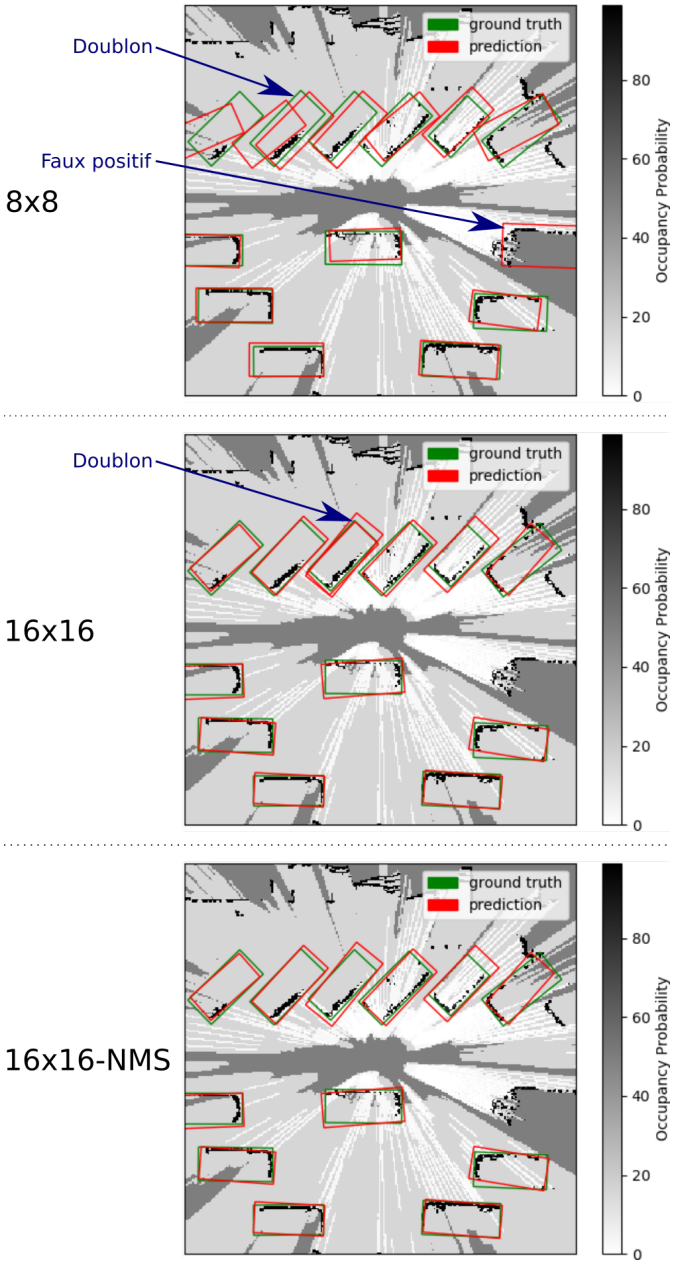

FIGURE 3 – Exemple de détections sur une même grille d'occupation

Les détecteurs d'objets sont classiquement évalués par l'étude de leur courbe de précision en fonction du rappel (« recall »). Ces courbes représentent le taux de détections correctes sur l'ensemble des détections (la précision) en fonction du taux de détections correctes sur l'ensemble des véhicules présents sur la grille d'occupation (le rappel). L'objectif étant de maximiser ces deux valeurs, la courbe d'un bon détecteur sera proche de la fonction constante valant 1 partout. Ces courbes sont par ailleurs paramétrées par un seuil d'association entre les détections et les vérités terrain appelé IoU (« Intersection over Union »). Plus ce seuil est élevé plus le taux de recouvrement géométrique entre les détections et les vérités terrain nécessaire pour qu'une détection soit considérée comme correcte est élevé. Nous utilisons les taux 0.5 et 0.7 dans cette étude dont les courbes sont représentées Figure 4.

La figure montre un net avantage de 16x16 et 16x16-NMS sur 8x8 car ces courbes y sont supérieures sur tout l'intervalle $[0, 1]$. La précision de 16x16 baisse plus rapidement que 16x16-NMS mais atteint un meilleur rappel maximal. On peut en déduire que l'ajout du post-traitement NMS à 16x16 lui permet de maintenir une meilleure précision en évitant des doublons mais a aussi pour effet de supprimer des détections valides.

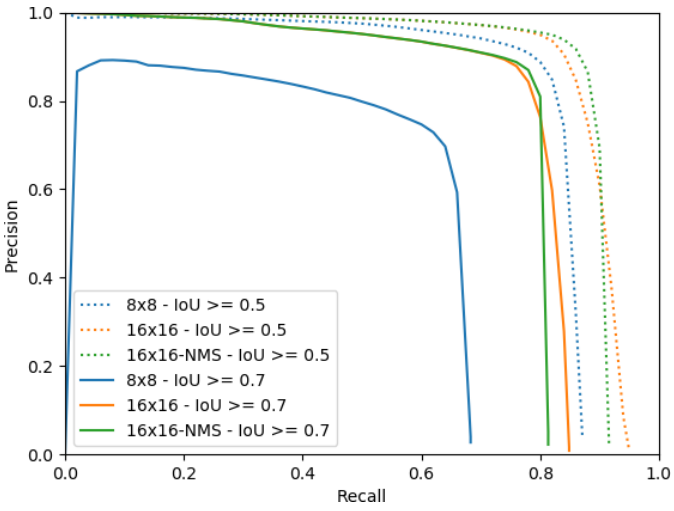

FIGURE 4 – Courbes de la précision en fonction du rappel

Ces courbes sont généralement résumées dans une métrique appelée précision moyenne (« Average Precision ») qui est la moyenne de cette précision quand le rappel varie dans $[0, 1]$. Cette métrique calculée sur la courbe avec un seuil d'IoU t est notée AP_t. Par ailleurs, les temps d'inférence et de post-traitement (dans le cas du détecteur 16x16-NMS) sont mesurés pour évaluer l'aspect temps réel de ces détecteurs. Les résultats sont résumés dans la Table 1.

On peut y observer un même temps d'inférence pour les trois détecteurs ainsi qu'un temps de post-traitement négligeable pour 16x16-NMS. En revanche, des différences s'observent sur la précision moyenne. Le détecteur 8x8 a une $AP_{0.5}$ et une $AP_{0.7}$ toutes deux inférieures à celles des détecteurs 16x16. Ces deux réseaux ne diffèrent que par la valeur de l'$AP_{0.7}$ pour laquelle 16x16-NMS est légèrement moins bon que 16x16.

TABLE 1 – Comparatif des détecteurs selon différentes métriques. Le GPU utilisé pour l'inférence est un Nvidia Quadro RTX 3000 Mobile.

Détecteur	$AP_{0.7}$	$AP_{0.5}$	Inférence	Post-traitement
8x8	0.55	0.82	**11ms**	**N/A**
16x16	**0.79**	**0.89**	**11ms**	**N/A**
16x16-NMS	0.77	**0.89**	**11ms**	**$\ll$1ms**

5 Conclusion

Cette étude propose trois architectures de détection de véhicules sur grilles d'occupation via la prédiction de boîtes englobantes orientées. Ces architectures inspirées de l'état de l'art en traitement d'images sont comparées selon leur précision moyenne et leur temps de prédiction.

Les valeurs de métrique ainsi que les exemples de détections présentés permettent de valider l'approche. Par ailleurs, les temps de prédiction des trois détecteurs sont faibles et permettent une exécution en temps réel de ces détecteurs. Des trois détecteurs implémentés, 16x16 semble le plus indiqué avec la possibilité en cas de doublons gênants de les éliminer avec un post-traitement NMS au prix d'une très légère perte d'$AP_{0.7}$.

Références

[1] T. R. Andriamahefa. Integer occupancy grids : a probabilistic multi-sensor fusion framework for embedded perception.

[2] A. Barrera, J. Beltrán, C. Guindel, J. A. Iglesias, and F. García. BirdNet+ : Two-stage 3d object detection in LiDAR through a sparsity-invariant bird's eye view. 9 :160299–160316. Conference Name : IEEE Access.

[3] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *1985 IEEE International Conference on Robotics and Automation Proceedings*, volume 2, pages 116–121.

[4] J. Redmon and A. Farhadi. YOLO9000 : Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525. ISSN : 1063-6919.

[5] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in perception for autonomous driving : Waymo open dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2443–2451. ISSN : 2575-7075.

[6] S. Wirges, T. Fischer, C. Stiller, and J. B. Frias. Object detection and classification in occupancy grid maps using deep convolutional networks. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3530–3535. ISSN : 2153-0017.