Utilisation conjointe d'information sémantique et géométrique pour la localisation d'objets par caméra temps-de-vol

Victor Joos[1] Antoine Vanderschueren[1] Christophe De Vleeschouwer[1]

[1] Institut ICTEAM, UCLouvain

{victor.joos, antoine.vanderschueren, christophe.devleeschouwer}@uclouvain.be

Résumé

*Nous proposons une méthode de localisation 3D d'objets à partir d'images d'intensité et de profondeur fournies par un capteur temps-de-vol (*Time-of-Flight*, ToF). Nous calibrons d'abord les paramètres extrinsèques de la caméra en nous appuyant sur l'information de profondeur des pixels du sol, segmentés à partir de l'image d'intensité et de profondeur. Nous localisons ensuite l'objet sur la base de l'alignement de son nuage de points avec un modèle de référence. L'originalité de notre démarche tient au fait que, pour segmenter (le nuage de points de) l'objet, nous proposons de compléter l'image d'intensité par des images d'information géométrique, hauteur et normale, normalisées par rapport à la hauteur du sol. Ces images sont rendues accessibles par la calibration de la caméra, et permettent de fournir la géométrie de la scène en entrée du réseau de segmentation sous la forme de cartes 2D.*

Mots Clef

Localisation, Segmentation, ToF.

Abstract

We propose a novel approach for 3D object localisation from images with both intensity and depth information provided by a Time-of-Flight (ToF) sensor. We first calibrate the extrinsic parameters of the camera, based on the depth information of floor pixels only. The latter were segmented from intensity and depth images. We then locate the object based on pointcloud alignement with a reference model. The originality from our approach comes from the fact that we extend our segmentation framework based on intensity images with geometric information. In particular, we use height and normals, normalized with respect to floor height. These images are made available by camera calibration, and enable the addition of scene geometry as a 2D input to the segmentation network.

Keywords

Localization, Segmentation, ToF.

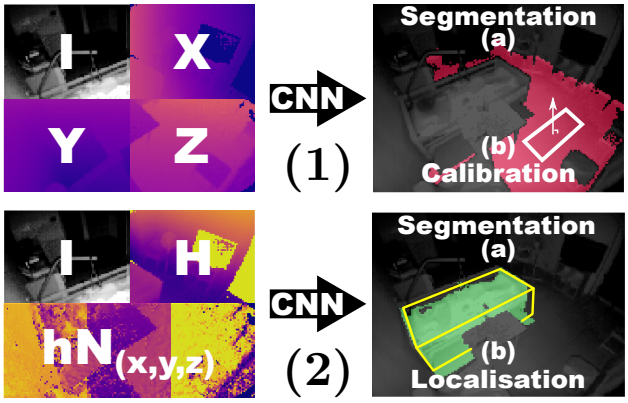


FIGURE 1 – Aperçu de notre méthode. Le premier réseau convolutif (CNN 1) prend en entrée l'intensité (I) et les cartes de profondeur (XYZ). Grâce à la segmentation du sol (1a), le repère peut être fixé au sol (1b), avec la composante en Z pointant vers le haut. Le deuxième CNN utilise l'intensité (I), la hauteur (H), et les normales (hN), ces deux dernières découlant de la calibration, et segmente un objet dans la scène (2a). Ensuite nous effectuons un recalage pour la localisation de l'objet (2b). *Affichage optimal sur écran.*

1 Introduction

La localisation d'objets est une étape préliminaire cruciale dans beaucoup d'applications dédiées à l'interprétation de comportement.

Dans ce contexte, les caméras Temps-de-vol (*ToF*) offrent l'avantage non-négligeables suivant : elles fournissent une carte de distance à la caméra en plus d'une image d'intensité réfléchie, devenant ainsi relativement indépendantes aux conditions de luminosité. Ceci au prix d'une résolution réduite et d'une portée limitée, rendant les ToF particulièrement adaptées au contrôle d'espaces restreints.

Beaucoup de techniques de segmentation sémantique d'images ToF utilisent seulement l'information de profondeur [8, 3], ou ont besoin d'un dispositif où les paramètres extrinsèques sont connus [2], ou travaillent sur le nuage de point résultant de la carte de profondeur [18].

La contribution principale de ce travail est l'utilisation

d'une approche en deux étapes, qui emploie l'information d'intensité 2D et l'information spatiale 3D (la position des pixels dans le repère de la caméra) en *tandem* pour localiser un objet dans le repère associé au sol d'un espace intérieur. Pour ce faire, nous suivons les étapes illustrées à la Figure 1. Dans un premier temps, nous utilisons un réseau de neurones convolutif (CNN) avec des cartes 2D d'intensité et d'information spatiale, dans le repère de la caméra, pour segmenter le sol. Cette segmentation nous permet de calibrer la caméra dans un repère aligné avec le sol. Alors que le repère de la caméra trouve son origine au niveau du dispositif, avec la composante en Z pointant vers la scène, le repère par rapport au sol rend la composante en Z parallèle à la direction de la gravité et l'annule au niveau du sol. Dans un second temps, nous segmentons un objet avec un CNN combinant cette fois-ci l'intensité ainsi que l'information de la hauteur et des normales calculées dans ce nouveau repère. Enfin, nous utilisons la segmentation de l'objet pour le localiser dans la scène.

Notre évaluation est appliquée sur un cas *pratique* avec des données réelles : localiser le lit dans une chambre de maison de repos ou d'hôpital. Nous montrons deux choses dans cette application réelle. Premièrement, les réseaux convolutifs surpassent les performances des réseaux conçus pour des nuages de points, comme PointNet++[19]. Et deuxièmement, notre segmentation en deux temps, surpasse celle de réseaux qui n'utilisent que les informations brutes de la caméra ToF. En combinant l'information de hauteur, des normales et d'intensité, cela nous permet de gagner entre 1 et 3% d'Intersection-over-Union (IoU) sur la segmentation, en fonction de la complexité calculatoire du réseau envisagé. Ce gain de précision nous permet d'extraire une meilleure localisation d'objets, avec un gain de plus d'1% sur la précision moyenne (AP@[.5,.95]).

Cet article est structuré en quatre parties. Nous établissons d'abord l'état de l'art. Ensuite nous décrivons notre méthode par rapport à l'état de l'art, afin d'en comparer les forces et faiblesses. Nous continuons par une analyse des résultats de notre méthode et terminons avec nos conclusions et perspectives d'avenir.

2 État de l'art

La littérature présente la localisation d'objets en 3D selon deux méthodologies distinctes. D'une part, plusieurs approches récentes travaillent sur une information 2D pour ensuite pouvoir localiser l'objet dans l'espace 3D. D'autre part, de nombreuses approches opèrent directement sur un nuage de points pour estimer la position de l'objet par rapport au repère de la caméra. Notre méthode appartient à la première catégorie. Elle identifie, via une segmentation 2D, les pixels appartenant à l'objet avant d'en dériver ses paramètres de positionnement.

Nous présentons d'abord les approches de segmentation 2D, ensuite celles en 3D et concluons avec les réseaux sur graphes.

Segmentation par réseaux convolutifs 2D Le point commun entre toutes les méthodes de segmentation pour images de couleurs avec information de profondeur (RGB-D), qui se distinguent des images ToF uniquement par la présence de couleurs, est l'importance de la fusion entre les différentes informations pouvant être extraites de l'image RGB-D. [3] et [8] démontrent l'importance de la fusion, l'un avec un simple encodeur-decodeur, l'autre avec des blocs résiduels et des liens-raccourcis (*skip-connections*). Remarquant que l'opération de fusion n'est pas optimale pour garder l'information de profondeur et d'intensité, [7] se concentre sur la fusion, et utilise un module d'attention de type *"Squeeze-and-Excite"* [6]. [24] combine l'information de profondeur non pas avec une fusion au niveau du réseau, mais bien dans les convolutions, avec une technique appellée *"depth-aware"* avec une multiplication pondérée par rapport à la différence de profondeur. Le désavantage de toutes ces approches est qu'elles n'utilisent que la profondeur, ne bénéficiant donc pas d'une information plus saillante comme la hauteur, ou font l'hypothèse d'avoir accès à un nuage de points orienté.

Localisation par réseaux convolutifs 3D Les réseaux convolutifs 3D se basent sur une *voxelisation* du nuage de points et souffrent donc soit de l'utilisation de voxels de trop grande taille, soit d'une sparsité élevée dans l'information qu'apportent les voxels. Un premier exemple de réseau de ce genre est [14] qui adopte la structure d'U-Net [20] (comme ce papier), mais avec des convolutions 3D. Très récemment [5] combine une information 2D à un réseau convolutif 3D. Ils montrent qu'une information 2D améliore d'environ 2-3% la précision de leur modèle. [23] entraîne un réseau sur des données synthétiques pour bénéficier de l'information associée aux voxels non-visibles. Le réseau apprend ainsi à compléter une image de point de vue unique avec ces voxels non-visibles et à localiser plus finement les objets. Nous avons observé que les modèles appris sur des données synthétiques ne généralisaient pas sur des données ToF réelles, à cause de la difficulté de simuler l'intensité de réflection de tous les points d'une scène.

Segmentation et localisation par réseaux sur graphes (GNN) Pour réduire le coût des réseaux convolutifs 3D, les réseaux sur graphes opèrent sur des ensembles de points et non pas sur des matrices (2D ou 3D). Il s'agit donc souvent de combinaisons entre des couches complètement connectés et des couches d'échantillonnage. La plupart des méthodes de segmentation 3D [11, 10], et de localisation 3D [9, 17, 16, 22] se basent sur PointNet [18] et PointNet++ [19]. Malheureusement, celles-ci restent à ce jour inférieures aux approches CNN en terme de précision sur les images RGB-D. Ceci est confirmé dans la section 4, où un réseau de type PointNet++ est comparé à notre méthode.

3 Méthode proposée

Pour chaque pixel, le dispositif ToF nous fournit à la fois une information d'intensité, réfléchie par la scène, et

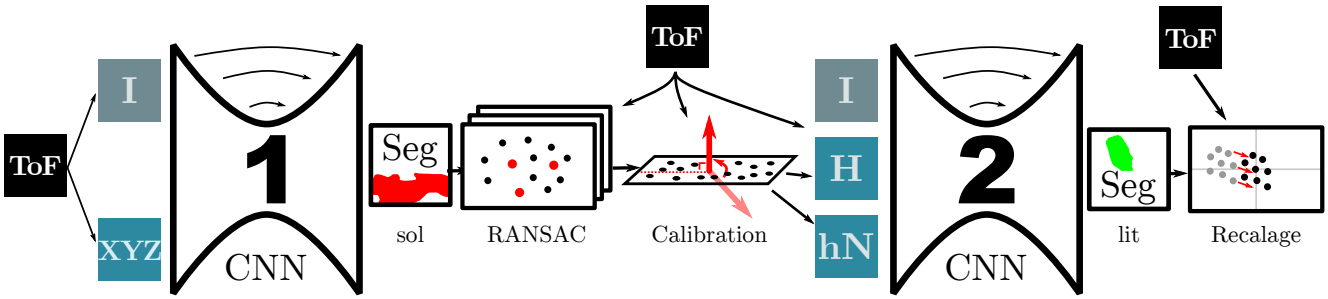


FIGURE 2 – Vue d'ensemble de notre méthode. Le premier réseau de segmentation utilise une image d'intensité (I) et la position spatiale (XYZ) dans le repère de la caméra ToF. De la segmentation du sol nous dérivons le repère du sol. Ce repère nous permet ensuite d'extraire les cartes de hauteur (H) et de normales (hN). Le second réseau combine ces cartes avec l'image d'intensité pour segmenter l'objet d'intérêt. Les points segmentés sont ensuite alignés avec un modèle de référence afin de déterminer les paramètres de localisation de l'objet d'intérêt.

de distance à la caméra. Disposant des paramètres intrinsèques de la caméra, nous pouvons exprimer la profondeur en terme de position dans un repère fixé à la caméra. Nous pouvons en outre estimer la normale en chaque point dans ce même repère.

Aperçu Général L'illustration à la Figure 2 montre les quatre sous-parties de notre méthode :

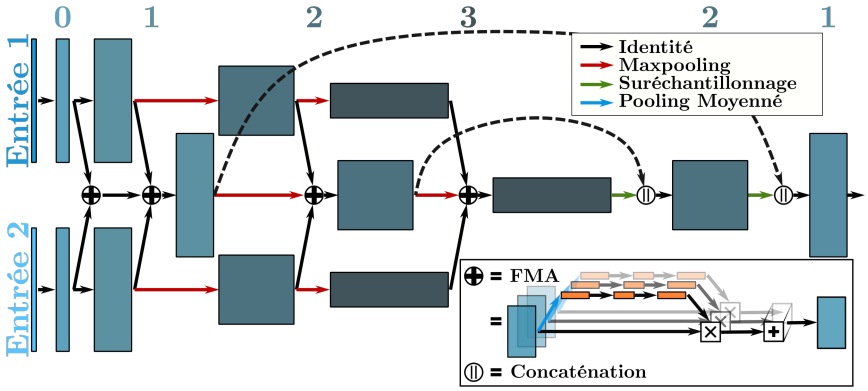
1. Un premier réseau de segmentation exploite à la fois les images d'intensité et de profondeur pour segmenter les pixels du sol.
2. Nous déterminons ensuite l'équation du sol par décomposition en valeurs singulières (SVD) des coordonnées 3D des points du sol segmentés. La normale au sol est fournie par le vecteur singulier correspondant à la plus petite valeur singulière. Pour augmenter la robustesse à d'éventuelles valeurs aberrantes, nous embarquons l'estimation des directions principales dans un calcul de consensus par échantillonnage aléatoire (RANSAC).
3. Le deuxième réseau convolutif segmente les points de l'objet d'intérêt sur base de l'information d'intensité, augmentée de quatre images définissant, en chaque pixel, la hauteur et les trois composantes de la normale dans un repère fixé au sol.
4. Les coordonnées 3D des points identifiés comme appartenant à l'objet sont fournies à un algorithme de recalage, en charge de les aligner avec un modèle de référence. Une méthode simple de recalage rigide consiste à optimiser la fonction de coût dérivée de l'IoU entre les coordonnées projetées sur le sol des points du lit et un modèle 2D rectangulaire de référence. Nous calculons le centroïde et l'angle du vecteur principal, après décomposition par SVD, des points pour initialiser la translation et la rotation.

Il est à noter que l'étape de calibration (étape 2) nous permet de transformer le repère de la caméra en un repère fixé au sol. Plus précisément, nous réalisons une rotation sur les axes x et y, pour amener l'axe z dans l'orientation de la gravité. Ainsi au lieu d'exprimer les données dans un repère fort dépendant du placement de la ToF, nous les exprimons dans un repère indépendant. Cela nous apporte deux avantages :

1. Le réseau de segmentation apprend plus facilement la relation entre l'information spatiale et le masque de l'objet (comme nous le montrons dans la section 4.3).
2. Le nombre de degrés de liberté lors de l'étape de recalage est réduit de 6 (3 angles, 3 distances) à 3 (1 angle, 2 distances).

Le recalage (étape 4) est sommaire, pour mettre en avant notre approche en deux-temps. Comme nous le verrons dans la section 4.3 il s'avère néanmoins suffisant.
Nous détaillons dans le reste de cette section comment les réseaux de segmentation impliqués aux étapes 1 et 3 ont été mis en oeuvre.

Segmentation Pour obtenir une segmentation de bonne qualité, nous nous tournons vers des réseaux de neurones convolutifs. Nous avons opté pour un réseau de type U-Net [20] pour notre architecture finale. U-Net adopte une structure d'auto-encodeur en y intégrant des liens-raccourcis qui relient les cartes de caractéristiques de même résolution de la partie encodeur, à celles de la partie décodeur. Ceci permet le transfert direct d'information haute-résolution, sans passer par les représentations de résolutions plus faibles. Par la suite nous ferons référence aux cartes de caractéristiques ayant la même résolution comme faisant partie d'un même *étage*. Les étages sont numérotés en ordre croissant de sous-échantillonnage (étage 1 = résolution maximale).
Pour la structure de chaque étage, nous nous sommes basés sur deux types de blocs différents : les blocs résiduels classiques [4] et les blocs *bottlenecks* résiduels inversés (ou '*mobiles*') [21]. Nous utilisons toujours (sauf indication contraire) les blocs résiduels classiques, car plus précis, mais les comparons tout de même dans un souci d'efficacité dans des dispositifs embarqués, aux blocs mobiles



(a)

	Étage	#CA	#répét.
ENC	0	32	1
	1	64	2
	2	72	2
	3	96	3
	4	128	4
DEC	3	72	2
	2	64	2
	1	64	1

(b)

FIGURE 3 – Architecture du réseau. (a) Représentation d'un exemple à 2 entrées et 3 étages (avec 1 bloc par étage) de notre architecture fusion ainsi qu'un zoom sur la fusion par module d'attention (FMA). (b) Détail pour chaque étage du nombre de cartes de caractéristiques et du nombre de fois que chaque bloc est répété.

dans la section 4.3. Ils sont tous deux combinés avec un module *Squeeze-and-Excite*. Ce dernier a pour but de pondérer chaque carte d'activation individuellement avant la sommation. [6] montre que cette optimisation d'architecture, peu chère en matière de paramètres et temps de calcul, apporte des améliorations notables d'un point de vue performance.

Nous changeons le nombre de répétition de chaque bloc aux différents étages pour mieux refléter les architectures de réseaux modernes de classification, souvent utilisées comme structures dorsales pour les réseaux de segmentation [26, 8, 1]. Le nombre de répétitions et de cartes de caractéristiques par étage est détaillé dans le Tableau 3b de la Figure 3.

En entrée, comme expliqué dans l'aperçu général de notre méthode, nos réseaux reçoivent une combinaison des types suivants : l'intensité (1D), la normale (3D), la profondeur (3D), et la hauteur (1D). Pour gérer des informations de nature différentes, [3] a montré l'intérêt d'une fusion après chaque étage d'un point de vue de la précision au détriment du nombre de paramètres et de la complexité calculatoire du réseau. Afin d'évaluer ce compromis, notre section expérimentale compare 2 approches différentes :

— Une fusion directe (FD) : chaque type d'entrée est convoluée une fois pour avoir un nombre de cartes d'activations égales. Elles sont ensuite sommées et passées à travers le réseau, simulant ainsi une entrée 'simple'.
— Une fusion progressive par module d'attention (FMA) : chaque entrée a sa propre branche d'encodage. À chaque étage ces branches sont fusionnées dans un encodeur central comme illustré à la figure 3a. Ce module de fusion ajoute un mécanisme d'attention sous la forme d'un Squeeze-and-Excite (SE). Cette architecture est inspirée de [7] qui utilise une version plus calculatoire du SE.

4 Résultats et Analyses

4.1 Méthodologie de validation

Cas d'usage Nous évaluons notre méthode sur une base de données de nuages de points, capturés dans des chambres d'hôpital ou de maison de repos. Notre objectif est de positionner le lit dans la chambre, sans autre information que la calibration intrinsèque de la caméra. Pour évaluer la performance de notre système lorsqu'il est confronté à un nouveau type de chambres, nous divisons nos données en 6 sous-ensembles tout en regroupant au sein d'un même sous-ensemble les données relatives à une même institution (hôpital ou maison de repos). Nous appliquons une validation croisée afin de systématiquement tester les modèles sur des chambres d'institutions qui n'ont pas été vues lors de l'entraînement ou la validation.
Notre base de données contient 2245 images, d'une résolution de 160×120 pixels, correspondant à 8 institutions et 51 chambres. Entre 40 et 50 images sont disponibles par chambre. Ainsi nous préservons des cas variés d'occlusion, de changements (mineurs et majeurs) de position du lit, d'illumination, et d'erreurs de mesures du senseur ToF, permettant aux réseaux de mieux généraliser.
Pour entraîner et évaluer nos modèles et comparer nos résultats, nous utilisons des données annotées manuellement, tant pour la calibration de la caméra que pour la localisation du lit.

Entraînement Nous sélectionnons dans notre validation croisée, 1 sous-ensemble pour les données de test, 1 pour la validation et 4 pour l'entraînement. Nous augmentons nos données d'entraînement en appliquant une symétrie verticale et un redimensionnement aléatoire. Ensuite, nous rognons ou comblons les données transformées pour qu'elles aient une résolution de 128×128 pixels.
Pour ce qui est de l'estimation des normales, nous prenons en chaque point les 10 voisins les plus proches à une distance maximale de 10 cm. Elles sont exprimées par 3 coor-

Méthode	Entrées	Fusion	Segmentation du **sol**			Diff. angle (°)
			IoU (%)	Rappel (%)	Préc. (%)	
RANSAC	XYZ	-	-	-	-	13.2
PointNet++[19]	XYZ+I	-	72.7	85.7	82 .8	3.37
ToF-Net	I	-	71.6	82.2	78.6	4.74
ToF-Net	I+XYZ	FD	**82.5**	**90.7**	**90.4**	**1.87**
ToF-Net	I+XYZ	FMA	81.5	**90.7**	89.4	1.90
ToF-Net	I+XYZ+N	FMA	80.6	89.0	89.9	1.89

TABLE 1 – Comparaison entre différentes méthodes de segmentation du sol. I désigne l'intensité, XYZ dénote les cartes de coordonnées spatiales, et N dénote les cartes des coordonnées des normales. Toutes les données sont dans le repère de la caméra. Une fusion directe est indiqué par FD et une fusion par module d'attention à chaque étage par FMA.

données spatiales soit dans le repère de la caméra (N), soit dans le repère par rapport au sol (hN).
Tous nos réseaux de segmentation résultent d'une sélection d'hyper-paramètres évalués sur la validation. Nous présentons finalement la moyenne des résultats obtenus sur les données de test. Les réseaux sont entraînés avec AdamW [13] en choisissant le taux d'apprentissage dans $\{1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}\}$ et le *weight decay* dans $\{1 \times 10^{-4}, 1 \times 10^{-5}, 1 \times 10^{-6}\}$. Le taux d'apprentissage est divisé par 10 aux époques 30 et 55 et l'entraînement dure pendant 70 époques avec une taille de *batch* fixe de 32. Nous implémentons notre réseau de segmentation avec PyTorch [15], et avons publié notre code à l'adresse https://github.com/ispgroupucl/tof2net.
Pour le réseau PointNet++, considéré comme base de comparaison, nous utilisons la structure du réseau décrite par [19] et implémentée par [25]. Nous ne démarrons pas de paramètres pré-entraînés, car les tâches disponibles sont forts différentes de notre cas d'usage.

Métriques Nous pouvons séparer les métriques d'évaluation en deux catégories : celles qui nous permettent d'évaluer la performance des réseaux de segmentation, et celles qui servent à évaluer les résultats de calibration de la caméra et de la localisation du lit.
Les réseaux de segmentation sont le plus souvent évalués avec l'indice de Jaccard (ou *IoU*), qu'on définit comme étant

$$\text{IoU} = \frac{|\text{GT} \cap \text{PRED}|}{|\text{GT} \cup \text{PRED}|} \tag{1}$$

$$= \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{2}$$

avec GT et PRED l'ensemble des points de l'annotation manuelle et de la prédiction respectivement, et dénotant les vrais positifs, faux positifs et faux négatifs avec TP, FP et FN. Ceci nous offre un bon résumé des erreurs de rappel $\left(\frac{\text{TP}}{\text{TP+FN}}\right)$ et de précision $\left(\frac{\text{TP}}{\text{TP+FP}}\right)$. Nous évaluons aussi ces dernières séparément, ce qui va nous permettre de départager plus précisément les réseaux avec un indice de Jaccard similaire. Pour évaluer la performance de nos réseaux de segmentation par rapport à leur complexité calculatoire, nous introduisons aussi la notion de GFLOP, ou le nombre d'instructions en virgule flottante que le réseau doit exécuter pour une image 128x128. Cette métrique donne une meilleure idée du temps de calcul nécessaire pour un réseau que le nombre de paramètres utilisés. Elle nous permet de paramétrer l'équilibre entre temps de calcul et précision qui sera différent pour chaque application.
Nous évaluons la calibration extrinsèque de la caméra en comparant l'angle absolu entre le vecteur normal au plan de la prédiction et celui de l'annotation manuelle.
Dans notre cas applicatif, le recalage est lui aussi évalué avec l'équation (1) de l'indice de Jaccard en comparant la projection des boites en 2D pour juger du bon alignement de celles-ci (IoU^b). Nous calculons aussi la précision moyenne (AP) à différents paliers d'IoU (de 0.5 à 0.95 par incréments de 0.05), comme il est courant de faire pour la détection de boites 2D [12]. Cette métrique permet de mieux différencier les détecteurs, car elle évite le biais vers un seuil d'IoU fixe.

4.2 Segmentation du sol et calibration

Dans un premier temps, nous comparons différentes méthodes de segmentation du sol ainsi que la calibration de la caméra qui s'ensuit.
La Table 1 résume les différentes valeurs des métriques des méthodes utilisées. Nous pouvons voir que PointNet++ est meilleur qu'un réseau convolutif qui utilise seulement une image d'intensité, probablement grâce à la structure inhérente du sol. Cependant, l'ajout de l'information XYZ dans le réseau convolutif suffit à dépasser PointNet++. Nous remarquons aussi que le réseau à fusion par module d'attention (FMA) estime légèrement moins bien le sol que celui à fusion direct (FD). Nous expliquons ceci par le grand nombre de paramètres ajoutés, pas nécessaires pour la segmentation de la structure simple du sol.
La dernière colonne de la Table 1 compare les résultats de la calibration dérivée des différentes sorties de réseaux, et on voit là aussi la même tendance que pour les résultats de segmentation.
Pour les réseaux que nous proposons, qui prennent les coordonnées XYZ en entrée, la calibration et normale au sol

Méthode	Entrées	Fusion	Segmentation du **lit**			Recalage	
			IoU (%)	Rappel (%)	Préc. (%)	AP@[.5, .95] (%)	IoU^b (%)
PointNet++[19]	XYZ+I	-	43.3	53.6	69.9	16.23	41.6
ToF-Net	I	-	67.2	82.2	78.6	35.6	58.7
ToF-Net	I+XYZ	FD	69.2	84.7	79.0	51.3	70.5
ToF-Net	I+XYZ	FMA	71.7	86.1	80.8	<u>52.5</u>	<u>71.6</u>
ToF-Net	I+XYZ+N	FMA	69.7	84.8	79.6	48.2	68.6
ToF^2-Net	I+H	FD	<u>71.9</u>	<u>86.2</u>	<u>81.4</u>	51.7	70.8
ToF^2-Net	I+H+hN	FMA	**74.1**	**86.9**	**83.4**	**53.6**	**72.2**

TABLE 2 – Comparaison entre différentes méthodes de segmentation du lit. I dénote l'intensité, XYZ et N dénotent les cartes de coordonnées spatiales et de normales locales, toutes deux dans les repère de la caméra, H et hN dénotent la carte de hauteur et des normales locales, toutes deux dans le repère fixé au sol. (le **meilleur** en gras, le <u>second</u> en souligné).

sont estimées avec une erreur inférieure à 2° par rapport à la calibration manuelle.

Pour évaluer la nécessité d'utiliser un réseau de segmentation pour trouver le plan du sol, nous comparons nos réseaux à une utilisation directe de RANSAC sur les données brutes. Nous avons observé que RANSAC marche convenablement sur certaines chambres, avec une erreur inférieure à 10° sur 45% des chambres. Cependant RANSAC se trompe systématiquement sur 55% des chambres, ce qui conduit à une erreur moyenne globale de l'ordre de 13°. Ceci confirme l'intérêt de baser l'estimation de la normale au sol sur une segmentation préalable des pixels du sol.

Les cas les plus difficiles pour la segmentation du sol correspondent à des problèmes de mauvaise réflection des signaux ToF, comme le montre la Figure 4. Nous ne proposons pas de solution à ce type de problème, car ils sont rares (1 chambre sur 51) dans l'environnement étudié.

4.3 Segmentation du lit et recalage

La Table 2 résume les résultats de segmentation et recalage du lit pour différentes modalités en entrée du réseau.

FIGURE 4 – Cas d'erreur : la lumière réfléchie par le revêtement du sol perturbe la ToF. En rouge, nous voyons les points pour lesquels la ToF ne fournit aucune mesure de distance et qui ne servent donc pas pour la calibration du sol.

Segmentation Nous comparons ici les réseaux de segmentation un-temps (PointNet++, ToF-Net), qui prédisent le sol et le lit en une seule inférence, à la structure deux-temps (ToF^2-Net) que nous proposons.

Nous voyons ici que PointNet++ perd beaucoup en *IoU*, même par rapport à un réseau de segmentation opérant uniquement sur l'intensité de réflection. Nous expliquons cela par le fait que le lit est un modèle 3D plus complexe que le sol.

Nous notons qu'en rajoutant l'entrée XYZ, c'est-à-dire les coordonnées 3D dans le repère de la caméra, nous obtenons une augmentation de 2% d'IoU. Ce bénéfice passe à près de 5% si nous encodons l'information spatiale en terme de la hauteur par rapport au sol (ToF^2-Net, I+H-FD). Un bénéfice similaire est obtenu en utilisant les modules d'attention, mais cela exige de multiplier la complexité calculatoire du réseau par 2.

Ceci confirme que la structure du lit est difficile à reconnaître sur base des coordonnées brutes et que l'encodage en hauteur améliore la qualité des modèles entraînés.

Le bénéfice de travailler dans le repère du sol est confirmé par l'apport des normales locales. L'ajout de l'information des normales est nuisible lorsqu'elle est exprimée dans le repère de la caméra (N) alors que celui-ci est bénéfique dans le repère normalisé par rapport au sol (hN). Nous obtenons ainsi notre meilleur résultat en combinant la FMA avec l'information de hauteur et des normales. Nous illustrons dans la Figure 5 quelques cas compliqués de segmentation. Ils confirment les avantages d'utiliser une segmentation en deux temps. En particulier, l'approche en deux-temps gère mieux les bords du lit et offre une segmentation plus précise dans les recoins éloignés de la scène.

Recalage Les conclusions de la partie recalage suivent presque mot pour mot celles de la segmentation. PointNet++ n'est pas utilisable en pratique, et se contenter de l'image d'intensité ne suffit pas non plus à obtenir un résultat satisfaisant. La méthode en deux temps bat à complexité calculatoire égale, celle en un temps, et la meilleure approche reste ToF^2-Net I+H+hN, avec FMA. Néanmoins,

(a) PointNet++ (b) ToF-Net FMA, `I+XYZ` (c) ToF^2-Net FMA, `I+H+hN`

FIGURE 5 – *(A regarder en couleurs)* Comparaison de trois réseaux de segmentation différents. En jaune, l'annotation manuelle du lit, en vert et rouge la segmentation automatique du lit et du sol respectivement. PointNet++ a beaucoup de mal avec ces cas difficiles. Le réseau un-temps montre de belles segmentations, mais fait souvent face à des problèmes de sur-segmentations (rangées 3 & 4) ou de sous-segmentations (2 & 5). Le réseau un-temps a même parfois plus de mal à trouver le lit que PointNet++ (rangée 1). Des cas difficiles avec des objets ressemblant à des lits (rangée 5) restent malgré tout difficiles pour le réseau deux-temps, mais la segmentation du lit est plus complète.

en l'absence de normales locales, nous constatons qu'en termes de performance de recalage, la segmentation en un temps (ToF-Net `I+XYZ`) avec FMA surpasse la segmentation en deux temps (ToF^2-Net `I+H`) avec FD, malgré son léger déficit en terme de qualité de segmentation. Nous expliquons ce phénomène par la simplicité de notre approche de recalage. Cela est illustré dans la dernière rangée de la Figure 5 où un canapé est segmenté en plus du lit. Même si le *rappel* de la segmentation du lit est plus élevé dans le cas à deux temps, notre méthode de recalage trouve dans ce cas une très mauvaise solution, englobant à la fois les pixels du lit et du canapé. Dans nos travaux futurs nous voulons étendre notre segmentation pour supporter la segmentation d'autres classes d'objets et ainsi réduire ce type d'erreur.

Méthode	Fusion	IoU_{lit} (%)	GFLOP
ToF^2-Net Standard[20]	FD	72.5	44.72
ToF^2-Net Mobile	FD	71.4	**5.27**
ToF^2-Net Résiduel	FD	71.9	16.18
ToF^2-Net Mobile	FMA	<u>73.3</u>	<u>8.29</u>
ToF^2-Net Résiduel	FMA	**74.1**	36.64

TABLE 3 – Comparaison de l'IoU de segmentation du lit en fonction du nombre d'opérations de différents réseaux. Tous les réseaux ont en entrée `I+H+hN`. Le réseau "Standard" utilise les paramètres originaux de [20].

Analyse de la complexité calculatoire La Table 3 montre la précision par rapport au temps de calcul de différents modèles. Nous réintroduisons ici les blocs mobiles, expliqués dans la section 3, pour leur faible complexité calculatoire. Nous voyons que les réseaux mobiles ont de très bonnes caractéristiques avec un bon équilibre entre précision et GFLOPs. À noter que le réseau mobile FMA surpasse même un réseau 5 fois plus calculatoire. Pour obtenir la plus grande performance possible, il faut par contre utiliser un réseau plus complexe, ce qui limite le déploiement en dispositif embarqué à temps réel.

5 Conclusion

Nous avons présenté une méthode de calibration de caméra et de localisation d'objets s'appuyant sur des outils de segmentation d'images ToF. Notre méthode estime les paramètres extrinsèques de la caméra avec un erreur moyenne inférieure à 2° à une mesure sur site. En utilisant cette information de calibration, nous segmentons un objet dans la scène grâce aux cartes de hauteur et de normales. Une méthode de recalage conventionelle est ensuite utilisée pour aligner un modèle de référence avec les points segmentés.
Nos expériences démontrent que notre approche en deux étapes (estimation de l'orientation du sol, suivie par la localisation de l'objet) améliore la prédiction de l'empreinte du lit au sol par rapport à une approche segmentant directement les points du lit à partir de l'information ToF. En d'autres mots, notre travail montre qu'exprimer les informations dans un repère aligné par rapport à la normale au sol plutôt que par rapport à la caméra bénéficie à l'interprétation de la scène.
Des travaux futurs peuvent être effectués sur une méthode capable de traiter des instances multiples d'un ou plusieurs objets d'intérêts dans une pièce.

Références

[1] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Lecture Notes in Computer Science*, volume 11211 LNCS, pages 833–851, 2018.

[2] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. *Lecture Notes in Computer Science*, 8695 LNCS(PART 7) :345–360, 2014.

[3] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. FuseNet : Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture. *ACCV*, pages 213–228, 2016.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *CVPR*, 2016.

[5] J. Hou, A. Dai, and M. NieBner. 3D-SIS : 3D Semantic Instance Segmentation of RGB-D Scans. *CVPR*, pages 4416–4425, 2019.

[6] J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. *CVPR*, pages 7132–7141, 2018.

[7] X. Hu, K. Yang, L. Fei, and K. Wang. ACNET : Attention Based Network to Exploit Complementary Features for RGBD Semantic Segmentation. *ICIP*, pages 1440–1444, 2019.

[8] J. Jiang, L. Zheng, F. Luo, and Z. Zhang. RedNet : Residual Encoder-Decoder Network for indoor RGB-D Semantic Segmentation. *arXiv preprint arXiv :1806.01054*, pages 1–14, 2018.

[9] L. Landrieu and M. Simonovsky. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. *CVPR*, pages 4558–4567, 2018.

[10] G. Li, M. Müller, A. Thabet, and B. Ghanem. DeepGCNs : Can GCNs Go as Deep as CNNs ? *ICCV*, 2019.

[11] Z. Liang, M. Yang, L. Deng, C. Wang, and B. Wang. Hierarchical depthwise graph convolutional neural network for 3D semantic segmentation of point clouds. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May :8152–8158, 2019.

[12] 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. *Lecture Notes in Computer Science*, pages 740–755, 2014.

[13] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *ICLR*, 2019.

[14] F. Milletari, N. Navab, and S. A. Ahmadi. V-Net : Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, pages 565–571, 2016.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch : An Imperative Style, High-Performance Deep Learning Library. *NeurIPS*, 2019.

[16] C. R. Qi, O. Litany, K. He, and L. J. Guibas. Deep Hough Voting for 3D Object Detection in Point Clouds. *ICCV*, 4 2019.

[17] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. *CVPR*, 2018.

[18] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet : Deep Learning on Point Sets for 3D Classification and Segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, pages 601–610, 2016.

[19] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++ : Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances In Neural Information Processing Systems*, 2017.

[20] O. Ronneberger, P. Fischer, and T. Brox. U-net : Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science*, 9351 :234–241, 2015.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen. MobileNetV2 : Inverted Residuals and Linear Bottlenecks. *CVPR*, pages 4510–4520, 2018.

[22] S. Shi, X. Wang, and H. Li. PointRCNN : 3D Object Proposal Generation and Detection From Point Cloud. In *CVPR*, pages 770–779. IEEE, 6 2019.

[23] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic Scene Completion from a Single Depth Image. *CVPR*, 2017.

[24] W. Wang and U. Neumann. Depth-Aware CNN for RGB-D Segmentation. *Lecture Notes in Computer Science*, 11215 LNCS :144–161, 2018.

[25] E. Wijmans. PointNet++ Pytorch. *https ://github.com/erikwijmans/Pointnet2_PyTorch*, 2018.

[26] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. BiSeNet : Bilateral segmentation network for real-time semantic segmentation. *Lecture Notes in Computer Science*, 11217 LNCS :334–349, 2018.