Classification dense d'images de profondeur entraînée sans données réelles

Howard Mahe, Denis Marraud, Andrew I. Comport

▶ **To cite this version:**

Howard Mahe, Denis Marraud, Andrew I. Comport. Classification dense d'images de profondeur entraînée sans données réelles. XXVIème colloque GRETSI, Sep 2017, Juan-les-Pins, France. hal-01635500

HAL Id: hal-01635500
https://hal.science/hal-01635500

Submitted on 15 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Classification dense d'images de profondeur entraînée sans données réelles

Howard MAHÉ[12], Denis MARRAUD[2], Andrew I. COMPORT[1],

[1] Université Côte d'Azur, CNRS, I3S (Laboratoire d'Informatique, Signaux et Systèmes)
2000 route des Lucioles, Les Algorithmes - bât. Euclide B, 06900 Sophia Antipolis, France

[2] Airbus Group Innovations - Équipe Image Solutions
12 rue Pasteur, 92150 Suresnes, France

howard.mahe@airbus.com, denis.marraud@airbus.com, andrew.comport@cnrs.fr

Résumé – Nous étudions le problème de la segmentation sémantique d'images de profondeur pour clusteriser les pixels d'une image qui appartiennent à une même classe d'objet afin d'étendre les approches de localisation et cartographie (SLAM) à des cartes labellisées. Nous nous focaliserons sur les approches denses et directes qui exploitent l'ensemble des données fournies par un capteur, tel qu'une caméra RGB-D grand-public. L'originalité de l'approche proposée réside dans l'apprentissage d'un réseau de neurones profond sans une masse de données réelles labellisées. Nous proposons une modélisation du capteur plus complète que les approches existantes pour générer des échantillons d'apprentissage à partir d'un modèle virtuel. Les résultats démontrent que la modélisation du bruit du capteur permet d'obtenir une précision globale de 68.9% pour la classification sémantique éparse d'images de profondeur encodées par représentation HHA.

Abstract – This paper studies the problem of semantic segmentation of depth maps clusterizing pixels of an image together which belong to the same objet class in order to extend SLAM technics with a labelled mapping. We focus on dense and direct approaches that harness whole data generated by a sensor, i.e. consumer RGB-D camera. What is original about this work is the training of a deep neural networks without a mass of real labelled data. A sensor model more complete than existing approachs generates training samples from a virtual model. Results show modeling sensor noise achieve a global precision of 68.9% in the case of sparse semantic segmentation of depth maps encoded with HHA representation.

1 Introduction

En robotique, les techniques de *SLAM* (Simultaneous Localization and Mapping) permettent aujourd'hui aux robots de se déplacer et se localiser en ayant une connaissance de la géométrie de l'environnement en temps-réel. Cependant, une compréhension sémantique de l'environnement permettrait d'améliorer les algorithmes de SLAM actuels mais aussi de robustifier la réalisation de nombreuses tâches de haut niveau qui font intervenir des interactions avec la scène. Dans le cadre du projet COMANOID [1], un robot humanoïde équipé d'une caméra est amené à effectuer des tâches de précision dans un avion civil en cours de construction.

Nous adressons ce problème de reconnaissance d'objets par l'apprentissage d'une tâche de segmentation sémantique d'images. La classification dense d'images ou segmentation sémantique consiste à clusteriser les pixels d'une image qui appartiennent à une même classe d'objet. Notre modèle d'apprentissage s'appuie sur les récents progrès des réseaux de neurones convolutionnels profonds (DCNN) utilisés pour de nombreuses tâches de perception telles que la classification d'images ou la détection d'objets. Notre approche diffère des approches classiques car nous entendons éviter la phase laborieuse de constitution manuelle de la base d'apprentissage en exploitant l'information sémantique contenue dans le modèle CAO disponible. Ce dernier n'étant pas texturé, il n'est pas possible d'extraire d'information photoréaliste. Nous nous intéressons donc à la classification dense d'images de profondeur (D) pour unique donnée d'entrée. Le modèle d'apprentissage devra être robuste au bruit introduit par les caméras de profondeur et capable de généraliser son apprentissage pour s'adapter aux différences entre le modèle virtuel et la scène réelle.

La problématique de ce papier consiste donc à effectuer un apprentissage sans données réelles à partir de rendus d'images de profondeur synthétiques issues d'un modèle CAO pour inférer la segmentation sémantique d'images de profondeur réelles bruitées et à trous.

1. Ce travail est supporté par la Commission Européenne, dans le cadre du projet H2020 COMANOID (http ://www.comanoid.eu) RIA No 645097

2 État de l'art

Les récents progrès du *deep learning* ont grandement amélioré les performances des algorithmes de segmentation sémantique. Le réseau FCN [13] fut le premier à introduire la structure en encodeur-décodeur des DCNN pour la segmentation sémantique. L'encodeur est un DCNN entraîné pour la classification d'images qui transforme l'image en entrée en une représentation de plus faible dimension (*features*). Le décodeur se charge de déconvoluer le bloc de *features* en sortie de l'encodeur pour obtenir une classification pixellique dense de l'image en entrée. Badrinarayanan [2] compare plusieurs variantes de décodeur. Les approches [2, 13, 15] sont comparables et se différencient principalement par un léger compromis entre temps d'apprentissage, temps d'inférence, mémoire nécessaire à l'inférence et précision de la segmentation.

Ces techniques de classification dense ont été appliquées à différentes modalités d'images : RGB, RGB-D, D, EVI [2, 13, 15, 19, 21]. Cependant, rares sont les méthodes de segmentation sémantique basées uniquement sur des images de profondeur (D) denses mais dont certaines données sont manquantes [16, 13, 8] et que nous appellerons donc *denses à trous* . Pourtant, les problèmes liés à l'usage des cartes de profondeur obtenues à partir de caméras grand-public ont fait l'objet de nombreuses études : d'une part pour modéliser les effets de bruit et de trous [4, 3, 20, 9, 7] et d'autre part pour les filtrer [19, 7, 21]. Les approches multimodales [13, 14, 21] ont l'avantage de pouvoir compter sur d'autres modalités que la profondeur lorsque celle-ci est manquante.

La plupart des approches d'apprentissage par ordinateur et plus encore celles de *deep learning* nécessitent l'élaboration d'une masse de données importante. Pour éviter l'acquisition et la labellisation manuelle d'une base d'apprentissage, des images de pro-

fondeur rendues à partir de modèles 3D ont été utilisées pour la reconnaissance et l'estimation de pose d'**objets** dans des images RGB [12, 1] mais aussi dans des images de profondeur (D) [22, 20, 5]. Récemment, Handa a montré l'intérêt de la création automatique de **scènes** synthétiques à partir de bases de données de modèles 3D d'objets pour le pré-apprentissage de DCNN [8].

Pour apprendre à réaliser la tâche de segmentation sémantique d'images de profondeur réelles $\{D_j\}$ par *deep learning*, nous constituons une base d'apprentissage (Sec. 3) en générant à partir d'un modèle $\mathcal{M}$ (Sec. 3.1) des rendus synthétiques $\{\overline{D_i}\}$ dans l'espace des données du capteur (Sec. 3.2). Nous introduisons un nouveau modèle pour le bruit du capteur (Sec. 3.3) afin de convertir les données synthétiques en données bruitées $\{\widehat{D_i}\}$ de sorte qu'elles soient représentatives des données réelles. Nous proposons ensuite plusieurs stratégies de prise en compte des pixels sans mesure (Sec. 3.4). Une dernière étape d'encodage (Sec. 3.5) des données vise à faciliter l'apprentissage. Pour finir, nous interprétons les résultats (Sec. 4) de ces contributions.

3 Apprentissage sans données réelles

3.1 Modèle virtuel

Soit N le nombre de vertices et M le nombre de triangles, le modèle $\mathcal{M}=\{\mathcal{V},\mathcal{N},\mathcal{T}\}$ (Fig. 1) est défini par un ensemble de vertices $\mathcal{V}=\{\mathbf{V_1},...,\mathbf{V_N}/\mathbf{V_i}\in\mathbb{R}^3\}$, de normales $\mathcal{N}=\{\mathbf{N_1},...,\mathbf{N_N}/\mathbf{N_i}\in\mathbb{R}^3\}$ et de triangles $\mathcal{T}=\{\{\mathbf{V_i^1},\mathbf{V_j^1},\mathbf{V_k^1}\},...,\{\mathbf{V_{i'}^M},\mathbf{V_{j'}^M},\mathbf{V_{k'}^M}\}\}$.

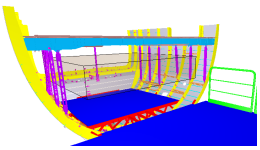


FIGURE 1 – Le maillage est représentatif d'un atelier d'assemblage sur lequel des robots assistants pourraient être déployés. La scène est divisée en 11 classes $\mathcal{C} = \{c_1,...,c_{11}\}$ décrivant les structures de l'avion et une classe *void* en noir dans les exemples.

3.2 Transformation modèle → espace capteur

Le rendu des images de profondeur est obtenu par un moteur de rendu classique par rastérisation pour P poses $\{\mathbf{T_1},...,\mathbf{T_P}\}$ dans un volume défini (Fig. 1).

L'Équation 1 projette les vertices, faces et normales du maillage dans le repère de la caméra à la pose $\overline{T_i}$:

$$\mathcal{V}^{P_i}=\overline{\mathbf{T_i}}\mathcal{V},\quad \mathcal{N}^{P_i}=\mathbf{R_i}\mathcal{N},\quad \mathcal{T}^{P_i}=\mathcal{T} \tag{1}$$

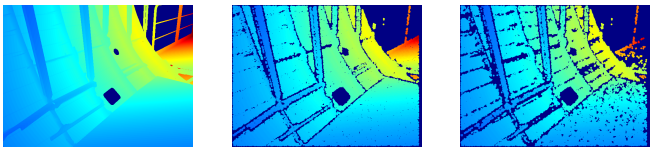
Pour chaque pose $\mathbf{T_i}=\begin{bmatrix}\mathbf{R_i} & \mathbf{t_i}\\ 0 & 1\end{bmatrix}\in\mathbb{SE}(3), \mathbf{R_i}\in\mathbb{SO}(3), \mathbf{t_i}\in\mathbb{R}^3$ et une résolution d'image (h,w), l'algorithme de rastérisation non linéaire R génère l'échantillon d'apprentissage $\{\overline{D_i}, {L_i}^*\}$ composé d'une image de profondeur synthétique $\overline{D_i}\in\mathbb{R}^{h\times w}$ en mètres (Fig. 2a) et de sa segmentation de référence ${L_i}^*\in\mathcal{C}^{h\times w}$ (Fig. 5d).

$$\{\overline{D_i}, {L_i}^*\}=R(\mathcal{V}^{P_i},\mathcal{N}^{P_i},\mathcal{T}^{P_i}) \tag{2}$$

3.3 Modélisation du bruit du capteur

Les images acquises à partir d'une caméra de profondeur qui utilise la technologie de projection d'un motif de lumière structuré sont par conception bruitées et présentent des trous sans profondeur (profondeur=0). Or, pour entraîner un algorithme de segmentation sémantique, il est primordial que les données d'apprentissage soient représentatives des données réelles.

Pour simuler ces effets sur les cartes de profondeur synthétiques, notre modélisation probabilistique du bruit d'une caméra de profondeur est inspirée de l'implémentation de la base de données RGB-D ICL-NUIM [9].



(a) Image de profondeur synthétique (b) Image de profondeur bruitée (modèle partiel) (c) Image de profondeur bruitée (modèle complet)

FIGURE 2 – Visualisation des deux modèles de bruit appliqués à une image de profondeur synthétique.

3.3.1 Bruit de profondeur et de quantification

L'Équation 3 du **modèle partiel** (Fig. 2b) convertit la carte de profondeur en carte de disparité. Les disparités sont perturbés par un bruit Gaussien $n_d{\sim}\mathcal{N}(0,\sigma_d^2)$ [3] puis arrondies à 2^{-3} près [4]. Enfin, la carte de disparité bruitée est reconvertie en carte de profondeur bruitée. Soit b l'écart entre le projecteur IR et la caméra IR, f la focale en pixels.

$$\widehat{D}(x,y)=\frac{b.f}{2^{-3}\lfloor 2^3\left(b.f/\overline{D}(x,y)+n_d\right)+0.5\rfloor} \tag{3}$$

Le bruit sur les disparités n_d illustre l'augmentation avec la profondeur du rapport signal sur bruit de la mesure de profondeur.

3.3.2 Bruit de mise en correspondance

Le **modèle complet** [9] (Fig. 2c) ajoute des effets de bruit supplémentaires en amont du modèle partiel. A chaque point 3D $\mathbf{P_i}$ correspondant à un pixel (x,y) dans l'image de profondeur D, est associée une normale locale $\mathbf{N}(x,y)$. Soit $\mathbf{K}\in\mathbb{R}^{3x3}$ la matrice de calibration intrinsèque de la caméra de profondeur tel que $\mathbf{P_i}(x,y)=D(x,y)\mathbf{K}^{-1}[x,y,1]^T$ et θ l'angle entre la normale au point et le lancer de rayon de la caméra passant par P_i (Équ. 4).

$$\theta=\left|cos^{-1}\left(-\mathbf{N}(x,y).\frac{\overline{P_i}(x,y)}{\left\|\overline{P_i}(x,y)\right\|_2}\right)\right| \tag{4}$$

Les points 3D en coordonnées homogènes $\overline{\mathbf{P_i}}$ sont pertubés le long de la normale locale par un bruit Gaussien $n_\theta{\sim}\mathcal{N}(0,\sigma_\theta^2)$ (Équ. 5) qui évolue linéairement avec la profondeur et quasi quadratiquement avec l'angle θ.

$$\sigma_\theta=\overline{D}(x,y)/f*\left(\sigma_{\theta_1}+\sigma_{\theta_2}\frac{\theta}{\pi/2-\theta}\right) \tag{5}$$

Les points 3D perturbés sont alors projetés et interpolés afin d'obtenir une profondeur latéralement décalée voire manquante (Équ. 6).

$$\widehat{D_1}(x,y)=\begin{cases}0 \quad \text{si} \quad degre(\theta)\in[82°,98°]\\ interp\Big(\mathbf{K}\big(\overline{\mathbf{P_i}}(x,y)+n_\theta.\mathbf{N}(x,y)\big)\Big) \quad \text{sinon}\end{cases} \tag{6}$$

Ce bruit illustre la difficulté à mesurer les disparités par mise en correspondances des points du motif projeté lorsque des surfaces sont observées avec un angle d'incidence important.

Par ailleurs, les pixels (x,y) de la carte de profondeur sont décalés aléatoirement suivant $(n_x,n_y)\sim\mathcal{N}(0,\sigma_s^2.\mathbf{I})$ le long des axes $\vec{x}$ et $\vec{y}$. La nouvelle carte de profondeur est obtenue par interpolation bilinéaire de cette grille de pixels bruités (Équ. 7).

$$\widehat{D_2}(x,y)=interp\big(\widehat{D_1}(x+n_x,y+n_y)\big) \tag{7}$$

Ce brassage des pixels simule la mauvaise localisation des mesures de profondeur, particulièrement observable aux frontières des objets dans les images de profondeur réelles.

3.4 Prise en compte des pixels sans profondeur

Si le modèle de bruit proposé était parfaitement identique au bruit réel du capteur, alors le classifieur apprendrait directement à prendre en compte les pixels sans mesure. Compte-tenu des limites de notre modèle, nous proposons une approche directe basée capteur (Sec. 3.4.1) que nous confrontons à une approche indirecte qui vise à améliorer les données réelles à classifier (Sec. 3.4.2).

3.4.1 Pixels sans profondeur ignorés à l'apprentissage

Selon le mode de prise en compte des pixels sans profondeur, nous définissons deux modalités d'apprentissage.

Dans la modalité nominale, l'erreur de prédiction du classifieur est calculée sur l'ensemble des pixels (avec ou sans profondeur) pour mettre à jour les paramètres du classifieur.

Dans la modalité **pixels sans profondeur ignorés**, les pixels sans profondeur sont ignorés dans la fonction de coût de l'apprentissage et n'influent donc pas sur l'apprentissage des paramètres du classifieur.

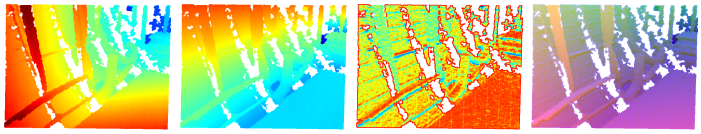
3.4.2 Inpainting des données réelles

L'***inpainting*** est une technique qui permet de remplir les pixels sans profondeur afin d'obtenir une image réellement dense. Nous évaluerons différentes techniques d'*inpainting* de la littérature :

1. interpolation au **plus proche voisin** [2] [6].
2. interpolation par **dilatation** morphologique **récursive** [2] [6].
3. **colorisation** de l'image de profondeur supervisée par l'image RGB [11], utilisée par la base de données NYUDv2 [19].

3.5 Encodage

L'encodage HHA [7] (Fig. 3) permet d'exploiter au maximum l'information contenue dans les images de profondeur. Cette représentation encode la disparité horizontale (Fig. 3a), la hauteur par rapport au sol (Fig. 3b) et l'angle entre la normale à la surface et la direction de la gravité (Fig. 3c). Cet encodage nécessite la connaissance du vecteur gravité (inféré ou estimé à partir d'une centrale inertielle) et la détection du sol dans l'image. Cette représentation géocentrique introduit une invariance à la hauteur par rapport au sol et une invariance au point de vue (illustrée dans [8]).



(a) Disparité (b) Hauteur (c) Angle (d) Image HHA

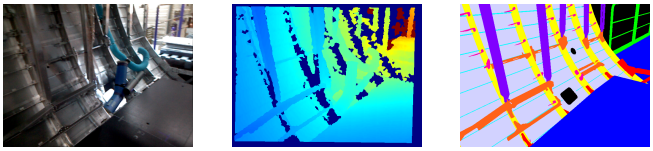
FIGURE 3 – Canaux et représentation RGB de l'image HHA.

4 Expérimentations et résultats

Rendu du modèle Nous avons rendu $P = 30000$ échantillons $\{\overline{D_i}, {L_i}^*\}$ à partir du maillage $\mathcal{M}$, pour des poses $\mathbf{T} = [\mathbf{R}, \mathbf{t}]$ où la translation $\mathbf{t} = [x, y, z]^T \in \mathbb{R}^3$ de la caméra est échantillonnée suivant une distribution uniforme $\mathcal{U}$ au sein de l'espace de travail du robot (Fig. 1) et la rotation $R \in SO(3)$ est définie par les angles d'Euler (α, β, γ) avec le tangage $\alpha \sim \mathcal{N}(\mu_\alpha = 90°\text{(horizon)}, \sigma_\alpha = 30°)$, le lacet $\beta \sim \mathcal{U}(0°, 360°)$ et le roulis $\gamma \sim \mathcal{U}(-10°, 10°)$. La distribution gaussienne sur le tangage exprime le fait que le robot passe plus de temps à regarder l'horizon que le sol ou le plafond.

2. https://github.com/s-gupta/rgbdutils

Données Ces 30000 échantillons sont répartis en une base d'apprentissage de 29000 échantillons et 1000 échantillons pour la base de validation. La base de **test** est constituée d'un lot $\{D_j\}$ de 25 images de profondeur réelles (Fig. 4b) acquises dans l'atelier d'assemblage réel avec une ASUS Xtion PRO LIVE. La segmentation de référence (Fig. 4c), qui approxime la vérité terrain, est obtenue par recalage de l'image RGB-D sur le modèle CAO.



(a) Image RGB (b) Image de profondeur réelle (c) Segmentation de référence

FIGURE 4 – Échantillon de données issues de la base de **test**.

Implémentation du classifieur dense Nous avons opté pour le réseau de neurones **fcn-8s** [13] qui est le plus répandu dans la communauté à l'heure actuelle, pour des images en entrée D ou HHA de taille 640x480. Le réseau est implémenté avec la librairie Caffe [10] et entraîné suivant la stratégie *heavy learning* [18] par une descente de gradient stochastique (SGD). L'apprentissage du réseau de neurones converge sans surapprentissage, en atteste une précision globale sur la base de validation supérieure à 95%. L'inférence prend en moyenne 60ms par image.

Métriques Soit n_{ij} le nombre de pixels de la classe i prédits comme appartenant à la classe j, $t_i = \sum_j n_{ij}$ le nombre total de pixels de la classe i et n_{cl} le nombre de classes différentes. Nous calculons la précision et l'indice de Jaccard moyen [2] tels que :

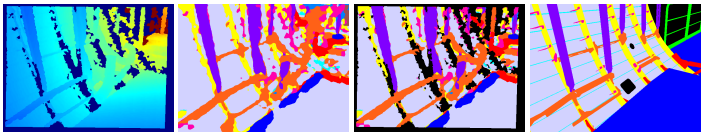
- la précision globale : $\mathbf{G} = \sum_i n_{ii} / \sum_i t_i$
- l'indice de Jaccard moyen : $\mathbf{mIoU} = \frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}$

Pour une classe donnée, l'indice de Jaccard représente le ratio de l'intersection sur l'union entre la vérité terrain et la prédiction. Cette métrique permet ainsi de pénaliser à la fois les faux positifs et les faux négatifs, alors que la précision globale fournit une information pixellique, indépendante de la répartition des classes.

Classification sémantique dense et éparse Nous distinguons deux tâches distinctes de classification sémantique :

- la **classification** sémantique **dense** : les performances sont évaluées sur l'intégralité des pixels des images de test, que l'information de profondeur soit présente ou non.
- la **classification** sémantique **éparse** : les performances ne sont évaluées que sur les pixels des images de test associées à une mesure de profondeur. Les labels en noir (Fig. 5c) sont ignorés par les métriques.

Cette distinction est intéressante pour des applications comme la reconstruction 3D sémantique [17, 14] où la qualité de la segmentation sémantique est primordiale et où une classification dense des pixels de l'image n'est pas nécessaire.



(a) Image de profondeur réelle (b) Classification sémantique dense (c) Classification sémantique éparse (d) Segmentation de référence

FIGURE 5 – Résultats de classification pour le réseau D0-C.

Pour le réseau D0-C, la classification sémantique dense obtient une précision globale de 56.8% contre 63.1% (**+6.3%**) pour la version éparse (Fig. 5).

Évaluations Dans le Tableau 1, nous évaluons la combinaison des modèles de bruit avec les stratégies à l'apprentissage de gestion des pixels sans profondeur pour la classification sémantique éparse, avec des entrées de type D (profondeur) et HHA.
Dans le Tableau 2, nous évaluons les différents techniques d'*inpainting* sur les données de profondeur (D).

4.1 Modélisation du bruit de la caméra

En pratique $b.f = 35.130$ et noux fixons $\sigma_d = 0.1, \sigma_s = 0.25, \sigma_{\theta_1} = 0.138, \sigma_{\theta_2} = 0.035$ par une évaluation qualitative du bruit généré.
Le Tableau 1 est sans appel concernant l'importance de la modélisation du buit du capteur lorsque l'on réalise un apprentissage sur des données simulées avec validation sur données réelles non *inpaintées*. Comparé à l'absence de modèle de bruit, le **modèle complet** apporte **+31.7%** avec entrée de type image de profondeur (D) et le **modèle partiel** apporte **+35.8%** pour les entrées HHA.

TABLE 1 – Évaluation sur données réelles (test) des modèles de bruit et de la modalité 'pixels sans profondeur ignorés'.

Réseaux	Modèle de bruit	Pixels sans prof. ignorés	Classification éparse G	Classification éparse mIoU
D0			24.3%	7.8%
D1	partiel		51.5%	14.3%
D2	partiel	✓	53.2%	16.2%
D3	complet		52.7%	14.4%
D4	complet	✓	**56.0%**	**16.6%**
HHA0			33.1%	11.7%
HHA1	partiel		67.5%	24.7%
HHA2	partiel	✓	**68.9%**	**26.9%**
HHA3	complet		68.1%	26.3%
HHA4	complet	✓	67.2%	24.2%

4.2 Prise en compte des pixels sans profondeur

4.2.1 Pixels sans profondeur ignorés à l'apprentissage

D'un point de vue quantitatif, pour la **classifiation éparse**, le Tableau 1 montre que la modalité 'pixels sans profondeur ignorés'améliore systématiquement les résultats par rapport à la modalité nominale, quelque soit le modèle de bruit, sauf pour les réseaux HH3/HH4.

Dans la modalité nominale, les réseaux (D1/D3) ont été entraînés à prédire la classe *void* pour l'*arrière-plan* et la classe correcte pour les pixels *sans mesure de profondeur*. Pourtant en pratique, ce comportement idéal n'est pas appris et le classifieur a tendance à prédire la classe *void* pour des zones sans profondeur.

En revanche dans la modalité 'pixels sans profondeur ignorés', les réseaux (D2/D4) ne sont pas capables de prédire la classe *void* car il n'y a aucun exemple positif de cette classe à l'apprentissage. Cette modalité est particulièrement adaptée à la **classification** sémantique **éparse**, par définition.

4.2.2 Inpainting des données réelles

Dans le Tableau 2, l'*inpainting* est évalué sur les réseaux entrainés **sans** modélisation du bruit lors de l'apprentissage (D0/HHA0) car ceux-ci sont plus performants sur des données sans trous, comme les images *inpaintées*, que les réseaux D1-4/HHA1-4.

Nos essais montrent que la méthode d'*inpainting* par **colorisation** surpasse largement les autres approches d'*inpainting*, quel que soit le type d'entrée : D (Tab. 2) ou HHA.
Nous nous attendions à ce résultat dans la mesure où c'est la seule méthode qui introduit de l'information des canaux RGB pour améliorer la qualité des cartes de profondeur.

Tout comme la modalité concurrente 'pixels sans profondeur ignorés', les techniques d'*inpainting* ne permettent pas de prédire les pixels correspondant à l'*arrière-plan*.

TABLE 2 – Évaluation sur données réelles (test) de l'inpainting, sans modélisation du bruit à l'apprentissage.

Réseaux	Inpainting	Classification éparse G	Classification éparse mIoU
D0		24.3%	7.8%
D0-A	plus proche voisin [6]	57.3%	20.2%
D0-B	dilatation récursive [6]	60.0%	22.2%
D0-C	colorisation [19]	**63.1%**	**24.2%**
HHA0		33.1%	11.7%
HHA0-C	colorisation [19]	42.3%	16.3%

4.2.3 Comparaison des stratégies

L'*inpainting* par **colorisation** surpasse (de **7.1%** sur la précision globale) le meilleur des modèles de bruit probabilistique pour des images de profondeur (D) en entrée.
En revanche, les résultats de l'*inpainting* sont très mauvais pour les données HHA. Une étude qualitative des images permet de constater que les prédictions à partir de données HHA sont particulièrement sensibles au bruit sur les images de profondeur réelles qui est propagé par l'encodage HHA. Alors que l'apprentissage a été fait sur des images HHA parfaites.

4.3 Encodage HHA

L'encodage HHA améliore de **13.4%** en moyenne (Tab. 1) la précision globale pour tous les réseaux excepté ceux utilisant une gestion des pixels sans profondeur par *inpainting* (Tab. 2).
L'indice de Jaccard (IoU) par classe permet d'identifier que les classes *plancher* et *pont supérieur* profitent grandement des invariances introduites par l'encodage HHA, au profit des classes décrivant des objets fins comme *poutre verticale*, qui souffrent des imprécisions du calcul des normales aux frontières des objets.

5 Conclusion

Dans ce papier, nous avons proposé une méthode d'apprentissage pour la segmentation semantique d'images de profondeur à partir d'un modèle virtuel. L'originalité de l'approche réside dans la conception (a) d'un nouveau modèle de bruit pour simuler les données issues d'un capteur de profondeur réel et (b) d'une méthode de traitement des données réelles pour qu'elles ressemblent davantage aux données simulées. Le **modèle probabiliste de bruit** tient compte du bruit proportionnel à la profondeur, du bruit du mise en correspondance interne au capteur et du bruit de quantification. Dans ce contexte, nous avons mise en concurrence un modèle de bruit de capteur avec trous contre l'amélioration des images réelles à classifier par une methode d'*inpainting*.
Les résulats sont obtenues pour la tâche de classification sémantique éparse d'images de profondeur réelles issues d'un capteur RGB-D grand-public. Les essais mettent en évidence l'importance

de la modélisation du bruit. En effet, les meilleurs résultats sont obtenus pour la segmentation d'images de profondeur encodées par représentation HHA avec apprentissage sur des images synthéthiques HHA bruitées (réseau **HHA2**) avec **68.9%** de précision globale. Notons cependant que l'*inpainting* par colorisation est la meilleure méthode pour la segmentation d'images de profondeur (réseau **D0-C**) avec **63.1%** de précision globale .
Dans de futurs travaux, nous tâcherons d'améliorer la modélisation du bruit du capteur avec des méthodes d'apprentissage.

Références

[1] Mathieu Aubry, Daniel Maturana, Alexei A Efros, Bryan C Russell, and Josef Sivic. Seeing 3d chairs : exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3762–3769, 2014. 2

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet : A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv :1511.00561*, 2015. 1, 3

[3] Jonathan T Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2013. 1, 2

[4] Michael Gschwandtner, Roland Kwitt, Andreas Uhl, and Wolfgang Pree. Blensor : blender sensor simulation toolbox. In *International Symposium on Visual Computing*, pages 199–208. Springer, 2011. 1, 2

[5] Saurabh Gupta, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Aligning 3d models to rgb-d images of cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4731–4740, 2015. 2

[6] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *CVPR*. GitHub, 2013. 3, 4

[7] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European Conference on Computer Vision*, pages 345–360. Springer, 2014. 1, 3

[8] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Scenenet : Understanding real world indoor scenes with synthetic data. *arXiv preprint arXiv :1511.07041*, 2015. 1, 2, 3

[9] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1524–1531. IEEE, 2014. 1, 2

[10] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe : Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 3

[11] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 689–694. ACM, 2004. 3

[12] Joseph J Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing ikea objects : Fine pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2992–2999, 2013. 2

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 1, 3

[14] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion : Dense 3d semantic mapping with convolutional neural networks. *arXiv preprint arXiv :1609.05130*, 2016. 1, 3

[15] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015. 1

[16] Xiaofeng Ren, Liefeng Bo, and Dieter Fox. Rgb-(d) scene labeling : Features and algorithms. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2759–2766. IEEE, 2012. 1

[17] Sunando Sengupta, Eric Greveson, Ali Shahrokni, and Philip HS Torr. Urban 3d semantic modelling using stereo vision. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 580–585. IEEE, 2013. 3

[18] Evan Shelhamer, Jonathon Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 2016. 3

[19] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1, 3, 4

[20] Shuran Song and Jianxiong Xiao. Sliding shapes for 3d object detection in depth images. In *European conference on computer vision*, pages 634–651. Springer, 2014. 1, 2

[21] Abhinav Valada, Johan Vertens, Ankit Dhall, and Wolfram Burgard. Adapnet : Adaptive semantic segmentation in adverse environmental conditions. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017. 1

[22] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets : A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 2