Pix2Point : prédiction monoculaire de scènes 3D par réseaux de neurones hybrides et transport optimal

Rémy Leroy Bertrand Le Saux Marcela Carvalho
Pauline Trouvé-Peloux Frédéric Champagnat

DTIS, ONERA, *Université Paris-Saclay*, F-91123 Palaiseau, France

remy.leroy@onera.fr

Résumé

Estimer la géométrie 3D d'une scène est crucial pour la reconstruction et la compréhension de celle-ci. L'information 3D est obtenue traditionnellement par stéréovision, et plus récemment par apprentissage profond avec d'excellents résultats même avec une seule vue. Cependant, il ne s'agit le plus souvent que de cartes de profondeur ou de disparité, alors que de nombreuses applications telles que la conduite autonome, la métrologie ou la robotique ont pour standard des nuages de points 3D, notamment car il s'agit du format natif des capteurs laser. Cet article présente Pix2Point, une approche de prédiction de nuage de points 3D d'une scène complète à partir d'une seule image. Elle repose sur une architecture de réseau de neurones hybride 2D-3D, et d'un apprentissage de bout-en-bout minimisant une distance de transport optimal. Nous montrons que notre approche obtient des résultats prometteurs pour la tâche de reconstruction monoculaire de nuage de points de scènes réelles issue du jeu de données KITTI. N'ayant pas encore été abordée, nous proposons des premières valeurs de référence pour cette tâche.

Mots Clef

Reconstruction 3D, Estimation 3D monoculaire, Nuage de points, Réseaux de neurones, Transport optimal, LiDAR

Abstract

Good quality scene reconstruction and comprehension rely on 3D estimation methods. The 3D information is usually acquired by stereo-photogrammetry, and deep learning has recently provided us with excellent results for monocular depth estimations. However, those estimations keep the image structure, while unstructured point clouds, measured by active laser scanners, are commonly used by many applications such as autonomous driving, metrology or robotics. In this paper, we present the first supervised approach for 3D pointcloud monocular prediction applied to complete and challenging outdoor scenes. Our method relies on a 2D-3D hybrid neural network architecture and efficient use of an optimal transport divergence for training in an end-to-end fashion. We show that this simple promising approach performs reasonably well for a task whose benchmark does not exist yet, that is 3D point clouds reconstruction on the challenging KITTI dataset.

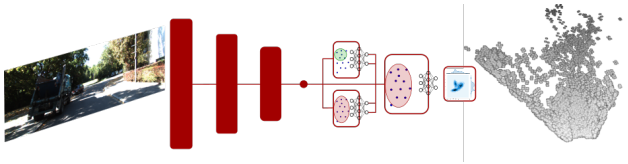


Figure 1 – Pix2Point : prédiction de nuages de points 3D de scènes réelles à partir d'une seule image, par réseau de neurones hybride 2D / 3D et transport optimal.

Keywords

3D Reconstruction, Single-image 3D estimation, Point-cloud, Neural networks, Optimal transport, LiDAR

1 Introduction

Pour comprendre l'environnement et y évoluer, il est fondamental d'en estimer la géométrie, afin de repérer les obstacles et les chemins possibles, reconnaître des objets et s'en approcher pour les saisir, et au final comprendre la scène dans son ensemble. C'est pourquoi de nombreuses recherches ont mené au développement de méthodes d'estimation et de reconstruction 3D. Elles ont longtemps utilisé la parallaxe entre images (6; 11), soit entre deux caméras en mode stéréo, soit entre de multiples acquisitions après déplacement. Récemment, les techniques d'apprentissage profond ont révolutionné l'estimation 3D à partir d'images, permettant même d'obtenir d'excellents résultats avec une seule vue (2; 4; 8; 13; 1). Ces reconstructions 3D sont habituellement représentées par une carte de profondeur, une image indiquant pour chaque pixel la distance du point de vue à la surface de la scène observée. Il est également possible de représenter la 3D avec des nuages de points correspondant à un échantillonnage 3D de la surface de scène considérée, qui peut être acquise à l'aide capteurs 3D natifs tels que les LiDARs. Ce dernier mode de représentation, contrairement aux cartes de profondeur, ne souffre

pas de l'alignement et d'un échantillonnage rigide sur la grille image. Par ailleurs, les approches SnapNet-R (10) et Pseudo-LiDAR (25; 27) ont montré l'apport significatif de la représentation en nuages de points par rapport aux cartes de profondeur pour les tâches de segmentation sémantique, de détection et localisation d'obstacles dans des scènes.
Dans cet article nous proposons : *(i)* la première approche de reconstruction d'un nuage de point 3D d'une scène entière à partir d'une seule image en utilisant une architecture de réseau de neurones hybride 2D / 3D, illustrée par la Figure 1, *(ii)* un apprentissage de bout-en-bout du modèle hybride, au contraire des apprentissages par étapes des travaux comparables (14) ; *(iii)* nous montrons que l'utilisation d'une fonction de coût issue du transport optimal permet d'obtenir des nuages de points réalistes sur les données KITTI (9), même avec une dizaine de milliers d'éléments.

2 Travaux connexes

2.1 Estimation de profondeur monoculaire

De nombreux travaux proposent des approches, supervisées, semi-supervisées voire non supervisées, pour traiter la tâche d'estimation de profondeur monoculaire sous forme de carte de profondeur. Saxena et al. (20) proposent de déterminer une relation entre une image et une carte de profondeur par apprentissage à l'aide de champs de Markov ; Eigen et al. (4) présentent une architecture convolutive multi-échelles constituée d'un premier réseau permettant une estimation grossière qui sera ensuite raffinée par un second réseau ; Carvalho et al. (2) ont montré l'apport du flou de défocalisation dans l'image pour la tâche d'estimation monoculaire ; Fu et al. (8) abordent cette tâche en reformulant le problème de régression en un problème de régression ordinale. L'état de l'art est atteint par la méthode Big To Small (BTS) de Lee et al. (13) grâce à un guidage efficace de descripteurs denses lors du décodage. Parmi les méthodes par apprentissage semi-supervisé, Amiri et al. (1) proposent semi-Depth, une approche combinant un apprentissage supervisé sur des données LiDAR KITTI et un apprentissage non supervisé à partir de carte de profondeur issue de la stéréo. Toutes ces méthodes estiment la 3D sous la forme de cartes de profondeur, qui une fois représentées en nuages de points induisent une densité de points qui diminue avec la profondeur. Afin d'éviter ce problème, nous proposons de prédire directement des nuages de points en 3D permettant de mieux couvrir spatialement les scènes.

2.2 Génération de nuage de points

La tâche de génération de nuages de points par réseau de neurones est relativement récente. Fan et al. (5) ont présenté PSGN, une méthode de reconstruction 3D d'un objet en nuage de points non ordonné à partir d'une unique vue de celui-ci et de sa localisation dans l'image. Leur approche prédit systématiquement la totalité du nuage de points, donc les contraintes computationnelles la rendent peu viable pour l'estimation de nuages riches en points. Cette limitation est abordée par Mandikal et Babu (14) avec DensePCR, une structure pyramidale permettant de grossir le nombre de points qui auront été préalablement prédits avec PSGN. Ainsi ils arrivent à estimer des nuages denses de 16.384 points après des entraînements réalisés par bloc. Xia et al. (26), abordent également la génération de nuage de points monoculaire d'objets, avec une méthode robuste aux occultations et à différents angles de prises de vues. Récemment, un modèle génératif basé flot permettant la prédiction d'un nuage de points d'un objet à partir d'une unique vue de cet objet a été proposé avec C-flow (17). Une caractéristique majeure de cette approche est la capacité d'inverser la tâche à réaliser : un modèle permettant la prédiction d'un nuage de points à partir d'une vue de l'objet, est aussi capable de synthétiser une vue d'un objet à partir de son nuage de points. Cette inversion est rendue possible notamment par un ordonnancement des points du nuage selon une courbe de Hilbert 3D. Néanmoins cette ordonnancement ôte le caractère non structuré du nuage de points.

Les travaux précédents ne traitent que la reconstruction de modèles d'objets, et montrent des difficultés lorsque des scènes de la vie réelle composées de multiples objets et décors sont considérées. À notre connaissance, nous sommes les premiers à proposer une méthode par apprentissage profond pour la reconstruction de scènes complexes sous forme de nuages de points, directement à partir d'une image monoculaire.

3 Méthode

Nous proposons une méthode permettant la génération de nuages de points pour des scènes à partir d'une seule image aux canaux rouge, vert et bleu (RVB). Similaire à DensePCR, notre architecture est composée de deux modules : (1) un encodeur-décodeur, composé de couches entièrement convolutives précédant une couche dense entièrement connectée, capable de prédire un premier nuage de points épars ; (2) un densifieur permettant d'enrichir un nuage en points d'un facteur donné, construit à partir de couches de perceptrons communs introduits avec PointNet (18) et PointNet++ (19). Contrairement à DensePCR dont l'apprentissage est effectué par bloc, nous réalisons un apprentissage de bout-en-bout.

3.1 Prédiction

Le module de génération de nuage de points est similaire à l'architecture la plus simple proposée par Fan et al. (5). Le réseau utilisé est de type VGG-19 (22) et est composé de 18 couches de convolutions pour encoder les caractéristiques issues de l'image RVB fournie en entrée puis d'une couche entièrement connectée pour les décoder et obtenir les coordonnées des points d'un premier nuage. Le nombre de points prédits N est déterminé par le nombre de neurones composant la dernière couche dense.

3.2 Densification

Le second module réalise une densification d'un facteur k du nuage de points issue du module de génération, permettant d'obtenir un nuage de $k \quad N$ points. La densification repose sur la procédure de reconstruction dense proposée par Mandikal et Babu (14) et utilise des couches de perceptrons communes pour l'extraction de descripteurs locaux et globaux à partir du nuage de points à enrichir.

3.3 Fonction de coût

Les performances de notre approche dépendent entièrement de la fonction de coût utilisée pour l'apprentissage. Contrairement aux méthodes de prédiction de carte de profondeur qui exploitent la structure ordonnée de l'image pour l'évaluation des erreurs, notre méthode nécessite l'utilisation d'outils permettant d'évaluer des distances entre représentations non-ordonnées. Nous exposons deux distances usuelles pour cette tâche, puis nous discutons de l'intérêt et des limitations de chacune d'elles.

Distance de chanfrein (Chamfer distance) La distance de chanfrein est une distance géométrique qui évalue la moyenne des distances euclidiennes au plus proche voisin d'un ensemble vers un autre. Elle est définie entre deux nuages de points S_1 et S_2 comme suit :

$$d_C(S_1; S_2) = \frac{1}{jS_1j} \sum_{x \in S_1} \quad (x; S_2) + \frac{1}{jS_2j} \sum_{y \in S_2} \quad (y; S_1) \tag{1}$$

où $\quad (\ ; S) = \min_{y \in S} k \qquad yk_2^2.$

Distance de Wasserstein (ou EMD pour Earth Mover's Distance) Cette distance établit une relation de comparaison entre les distributions des points dans l'espace, autrement dit, deux ensembles de points ont une distance de Wasserstein faible si et seulement si la distribution de leurs points est sensiblement similaire. Elle est définie comme le résultat du problème d'optimisation suivant :

$$d_{EMD}(S_1; S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} kx \qquad (x)k_2^2 \tag{2}$$

où $\quad$ est une application bijective dans le cas où $jS_1j = jS_2j$.
Le calcul de la distance de Wasserstein étant très coûteux, nous considérons dans notre cas une approximation de la distance de Wasserstein régularisée par un terme entropique. Cette approximation est obtenue au moyen de l'algorithme de Sinkhorn-Knopp (3; 16), qui permet également de traiter des nuages de points aux nombre d'éléments différents. Nous ferons par la suite référence à cette approximation par le terme distance de transport optimal, ou distance OT. L'apprentissage de bout-en-bout minimisant la distance OT pour près de 10,000 éléments est rendu possible grâce à l'implémentation rapide et efficace de celle-ci par Feydy et al.(7) dans la librairie Python `geomloss`. Cette implémentation permet le traitement de scènes de la vie réelle avec un gain d'un facteur 100 par rapport aux modèles graphiques d'un millier de points manipulés par (5; 17).

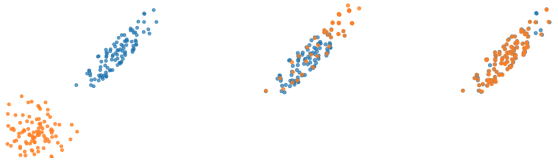


Figure 2 – Résultats du problème de minimisation des différentes fonctions de coût pour la même initialisation. La mise jour des positions des 100 points source (orange) est obtenue par descente de gradient. Le résultat obtenu par minimisation de la distance OT est fidèle au nuage cible (bleu), contrairement à celui issu de chanfrein.

Comparaison des distances Nous faisons le choix de ne conserver que la distance OT comme fonction à minimiser pour les raisons suivantes : minimiser la distance OT revient à ce que le nuage prédit (nuage source) ait des points suivant la même distribution que les points du nuage cible, même pour des nuages aux nombre d'éléments différents. Cela n'est pas assuré par la distance de chanfrein. De plus, la résolution du problème de minimisation de la distance de chanfrein par descente de gradient est sensible à la position initiale des points et peut mener à des minima locaux comme le montre la Figure 2.

4 Expériences

Nous présentons d'abord dans cette partie le protocole expérimental. Nous commençons par définir en partie 4.1 les jeux de données utilisés lors des apprentissages, puis nous détaillons notre implémentation en 4.2. Nous analysons ensuite les résultats obtenus pour la reconstruction de diverses données 3D en partie 4.3, et nous les comparons au méthodes d'estimation de profondeur 2D en partie 4.4. Nous terminons avec une discussion de ces résultats en 4.5

4.1 Jeux de données

La 3D d'une scène peut être acquise de plusieurs façons, avec un LiDAR procurant une représentation parcimonieuse de la scène sous la forme d'un nuage de points ou encore avec un capteur RVB-D. Cette dernière méthode permet une acquisition dense, sous la forme de carte de profondeur correspondant à l'image, et assez fidèle de la scène, idéale en intérieur, mais peu adaptée pour des scènes extérieures dont les conditions environnementales sont fortement changeantes et difficilement contrôlables. Il s'agit exactement du type de scènes que nous souhaitons traiter, et le LiDAR est insensible à ce genre de phénomènes. C'est la raison pour laquelle nous considérons le jeu de données KITTI (9) qui contient des séquences d'images de scènes urbaines ainsi que les nuages de points correspondants acquis par LiDAR.

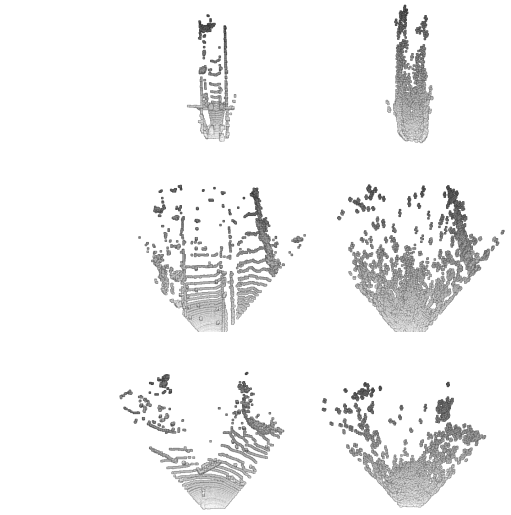
Nous décrivons par la suite les deux variantes de nuages de points de KITTI que nous allons utiliser pour nos expériences. Dans tous les cas, les évaluations sont réalisées sur la partition d'entraînement (22600 scènes) et de test (697 scènes) définie par Eigen (4).

LiDAR Ce premier jeu de référence est constitué à partir des nuages de points issus du LiDAR fournis pour chaque scène. Il s'agit des coordonnées quasi-exactes des points échantillonnés par le LiDAR à un instant donné. Cependant, seuls les points visibles dans le cône de vision de la caméra sont conservés, ce qui produit des nuages de points de taille variable, 18000 en moyenne. L'apprentissage comme l'évaluation compareront donc des nuages prédits de taille fixée avec ces nuages cibles dont le nombre de points varie significativement.

KITTI-Depth Le second jeu de données est issu des cartes de profondeurs de référence pour la tâche d'estimation de la profondeur du challenge. Ces cartes sont construites par accumulation d'acquisitions LiDAR consécutives, ré-alignées sur la grille image, aboutissant à 40000 pixels avec information de profondeur. Pour notre tâche de prédiction 3D, nous créons à nouveau les nuages de points correspondants en nous limitant toutefois à 10000 points répartis uniformément sur la totalité de la scène. KITTI-Depth donne une représentation plus complète de la scène que la base de donnée LiDAR et permet également la comparaison avec les approches 2D denses présentée en 4.4.

4.2 Détails d'implémentations

L'implémentation de notre réseau est réalisée grâce à la bibliothèque logicielle *PyTorch* (15). Nous prédisons des nuages de $N = 2500$ points en sortie du module de génération (module comportant près de 293M de paramètres dont 94% issues de la dernière couche entièrement connectée) qui seront enrichis d'un facteur $k = 4$ par le module de densification (de 20k paramètres), menant à des nuages de 10000 éléments. Tous les nuages de points se trouvent dans le champ de vue de la caméra 2 (RVB), et sont exprimés avec les coordonnées rectifiées. Les images utilisées en entrée du réseau sont toutes rognées à la résolution 1224 370, ainsi nous travaillons avec la définition originale des images, contrairement à toutes les approches d'estimation de profondeur, qui utilisent des versions sous-résolues des images.
L'apprentissage de bout-en-bout est rendu possible grâce à la bibliothèque Python `geomloss` (7), permettant le calcul d'une approximation de la distance de Wasserstein pour autant d'éléments. Pour des raisons de contraintes computationnelles, causées par une inconsistance entre les différents exemples du nombre de points par nuage, lors de l'entraînement les exemples sont présentés individuellement et non par batch. Nous considérons donc dans notre architecture d'encodeur des couches de normalisation par instance (24) pour remplacer la normalisation par batch. Tous les entraînements sont réalisés avec une optimisation de type Adam, pendant 40 époques sur la partition d'entraînement standard Eigen. Un entraînement complet réalisé sur carte graphique TITAN X prend 80 heures.



(a) (b) (c)

Figure 3 – Reconstruction en nuage de points de la scène, vue de dessus, après un entraînement sur les données LiDAR. (a) image RVB de la scène test ; (b) nuage de points LiDAR de référence et (c) notre prédiction de 10000 éléments.

4.3 Résultats

Nous abordons à présent des résultats d'apprentissage réalisés pour chacune des variantes de nuage de points KITTI. Nous commençons par une analyse qualitative des prédictions de notre approche, puis nous définissons des outils de mesures de performances issues du problème de reconstruction de scène afin d'évaluer qualitativement nos résultats.

Qualitativement La Figure 3 (respectivement Figure 4) montre des exemples de nuages de points prédits à partir d'une image RVB donnée après un entraînement sur la variante LiDAR (resp. variante carte de profondeur KITTI-Depth). Chacune des scènes possède des difficultés qui lui est propre. Un premier constat est que la forme générale du nuage prédit est fidèle à la vérité terrain, nous retrouvons une distribution ressemblant à un couloir lorsque que l'image présente une rue parcourant un lot de bâtiments ; certains détails sont également retrouvés comme une rangée d'arbres sur le bord de la route dans le cas du deuxième exemple.

Quantitativement Il n'existe pas de valeur ni de méthode de référence concernant la tâche de reconstruction de scène KITTI. Nous proposons donc des mesures de performances provenant du défi de reconstruction de scène 3D, *3D Reconstruction meets Semantics 2018* (23). Ces mesures sont définies de la manière suivante :

Référence	Précision # en m	Complétude ' (en %) 1 m	50cm	10cm
LiDAR	2.29	86.11	68.16	10.23
KITTI-Depth	2.18	87.39	70.88	15.28

Tableau 1 – Mesures de performances pour chaque variante, évaluées sur la partition de test Eigen.

(a) (b) (c)

Figure 4 – Reconstruction en nuage de points de la scène, vue de dessus, après un entraînement sur les données KITTI-Depth. (a) correspond à l'image RVB de la scène test ; (b) le nuage de points de référence de 10000 éléments et (c) notre prédiction de 10000 éléments.

La précision est la distance d, en mètre, du n-ième percentile des distances au plus proche voisin, du nuage prédit vers le nuage vérité terrain. Elle mesure la pire distance au plus proche voisin parmi les points prédits les plus proches de la vérité terrain.

La complétude est le pourcentage de points cibles ayant un voisin dans le nuage source à une distance inférieure à une distance seuil préalablement dé nie. Nous évaluons des valeurs de complétude pour des seuils de 1m, 50cm et 10cm. Cette mesure de performance re ète la couverture du nuage cible par le nuage source.

Nous rassemblons dans le Tableau 1 les valeurs de performances pour chacune des variantes de jeu de données, pour plusieurs seuils de complétude (du moins n au plus n) et la précision est évaluée avec un taux de 90%.
Nous remarquons que la disparité des résultats est assez faible entre les variantes, avec une légère supériorité pour celle utilisant les cartes de profondeur, notamment au niveau de la complétude à petite échelle (10cm). Il est important de préciser que les scènes font jusqu'à 80 mètres de profondeurs, ainsi la précision de 2.18 mètres pour cette portée est encourageante pour les futures approches appliquées à cette tâche.

4.4 Comparaison aux approches d'estimation de la profondeur 2D

La comparaison des méthodes de prédiction de profondeur avec notre approche qui prédit directement le nuage n'est pas triviale. Nous proposons le protocole illustré par la Figure 5 permettant de construire un nuage de points possédant autant d'éléments qu'une prédiction Pix2Point. Pour cela nous projetons les points de la prédiction Pix2Point dans le plan carte de profondeur estimée par l'approche dense à comparer, a n d'obtenir un masque indiquant les pixels de la carte dense à conserver. Le nuage de points issu de la carte dense est déduit de ce masque et des paramètres caméra. À partir des nuages de points ainsi obtenus, il est possible de comparer Pix2Point à diverses approches d'estimation de profondeur monoculaire, dense et 2D issues de l'état de l'art du challenge KITTI [1] avec les mêmes critères de performances énoncés en section 4.3. Nous référençons ces valeurs dans le Tableau 2.

Figure 5 – Méthode de comparaison aux méthodes denses 2D. (1) Projection du nuage de points prédit par Pix2Point sur la carte de profondeur pour déterminer un masque. (2) Construction du nuage issu de la carte de profondeur à partir du masque.

Les approches denses héritent des progrès issus de plusieurs années de recherche sur l'estimation de profondeur 2D, et bénéficient d'un apprentissage sur des cartes 2D plus complètes que les nuages de points KITTI-Depth. Pix2Point obtient des résultats d'une précision inférieure, mais néanmoins du même ordre de grandeur, tout en étant le premier essai d'estimation directement en 3D de la profondeur. La Figure 6 montre notamment que même si moins précises, les estimations de Pix2Point sont toutes à proximité des points de référence, avec une erreur inférieure à 2.50m. Il est notable que Pix2Point utilise une armature VGG-19 pour l'encodage de l'information 2D tandis que les autres approches sont fondées sur un ResNet-50. L'utilisation d'armatures performantes constitue une piste de choix de l'amélioration de Pix2Point.

1. http://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_prediction

Référence	Précision en m	Complétude (en %) 1 m	50cm	10cm
point d'intérêts	6.02	51.39	27.0	0.75

Tableau 3 – Mesures de performance pour des données de référence issues de nuages de points d'intérêt obtenus par stéréovision, fortement non-structurés et répartis de manière non-uniforme dans l'espace (à comparer avec le Tableau 1).

Figure 6 – Densités de probabilité des distances au point le plus proche.

4.5 Discussion et perspectives

KITTI stereo Les jeux de référence précédemment exposés en 4.1 utilisent des données acquises par LiDAR pour connaître la géométrie de la scène. Cependant cette mesure n'est pas toujours disponible. Nous nous intéressons également à la prédiction de nuages de points construits par stéréophotogrammétrie. Cette approche est plus abordable que par un capteur actif. Le jeu de données 3D est construit de la façon suivante : (1) Nous évaluons une carte de disparité à partir d'une méthode Semi-Global Block Matching (SGBM) (12) ; (2) À l'aide d'une méthode Good Features To Track (21), nous réalisons une détection de 10000 points d'intérêts dans les zones de l'image présentant une disparité ; (3) Nous plaçons les points d'intérêts obtenus dans la scène en utilisant les paramètres intrinsèques de la caméra et la carte de disparité.

Comme le montre le Tableau 3, les résultats sont cohérents, mais beaucoup moins précis que pour la prédiction de nuages de points ressemblants à du LiDAR. Un explication possible est la forte variation de répartition des points qui sont concentrés sur les zones texturées et les coins des objets de l'image. La densité des points 3D à estimer est alors particulièrement multi-modale et piquée. Apporter des solutions à ce problème constituera l'une des suites de ces travaux et passera par une modélisation plus ne de la distribution spatiale cible.

Approche	Précision# en m	Complétude' (en %) 1 m	50cm	10cm
BTS (13)	0.94	96.11	87.94	32.84
semiDepth (1)	1,18	91.37	79.95	24.14
Pix2Point	2.18	87.39	70.88	15.28

Tableau 2 – Comparaison de l'approche proposée à des méthodes denses d'estimation de profondeur 2D.

Une autre limite actuelle de l'approche est le manque de précision locale des prédictions. En effet, si la forme globale des scènes est bien estimée sur les Figures 3 et 4, les détails sont confus. Des modèles d'encodage 2D plus performants (réseaux résiduels, denses, etc) ainsi que des décodeurs plus exibles spatialement que l'approche DensePCR (14) seront étudiés.

5 Conclusion

Dans ce papier, nous avons présenté Pix2Point, la première méthode d'estimation monoculaire de nuage de points 3D de scènes urbaine. Nous avons utilisé une architecture hybride 2D-3D, entraînée de bout-en-bout à l'aide d'une distance provenant du transport optimal. Nous avons introduit un protocole permettant la comparaison d'approches 2D et 3D. Nous avons montré que l'utilisation d'une distance de transport optimal permet à Pix2Point une reconstruction - dèle des nuages de points et d'obtenir des résultats, certes inférieurs, mais du même ordre de grandeur que des méthodes issues de l'état de l'art pour la prédiction de profondeur en 2D. Plusieurs pistes d'améliorations sont possibles, depuis l'utilisation d'une armature plus performante pour l'encodage des images jusqu'à des modèles de densi cation multi-échelles permettant des résultats plus ns spatialement. Ces nouvelles approches seront développées dans nos futurs travaux.

Références

[1] A. J. Amiri, S. Y. Loo, and H. Zhang. Semi-supervised monocular depth estimation with left-right consistency using deep neural network. CoRR, abs/1905.07542, 2019. 1, 2, 6

[2] M. Carvalho, B. Le Saux, P. Trouvé-Peloux, A. Almansa, and F. Champagnat. Deep depth from defocus : how can defocus blur improve 3D estimation using dense neural networks ? CoRR, abs/1809.01567, 2018. 1, 2

[3] M. Cuturi. Sinkhorn distances : Lightspeed computation of optimal transport. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, NIPS, pages 2292–2300, 2013. 3

[4] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 27, pages 2366–2374. Curran Associates, Inc., 2014. 1, 2, 4

[5] H. Fan, H. Su, and L. J. Guibas. A point set generation

network for 3D object reconstruction from a single image. *CoRR*, abs/1612.00603, 2016. 2, 3

[6] O. Faugeras. *Three-Dimensional Computer Vision : A Geometric Viewpoint*. MIT Press, Cambridge, USA, 1993. 1

[7] J. Feydy, T. Séjourné, F.-X. Vialard, S.-I. Amari, A. Trouvé, and G. Peyré. Interpolating between optimal transport and MMD using Sinkhorn divergences. *arXiv preprint arXiv :1810.08278*, 2018. 3, 4

[8] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. *CoRR*, abs/1806.02446, 2018. 1, 2

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics : The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2, 3

[10] J. Guerry, A. Boulch, B. Le Saux, J. Moras, A. Plyer, and D. Filliat. SnapNet-R : Consistent 3D multi-view semantic labeling for robotics. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017. 2

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN : 0521540518, second edition, 2004. 1

[12] H. Hirschmueller. Stereo processing by semiglobal matching and mutual information. *Trans. Pattern Analysis and Machine Intelligence*, 30 :328–41, 2008. 6

[13] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh. From Big to Small : Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv :1907.10326*, 2019. 1, 2, 6

[14] P. Mandikal and R. V. Babu. Dense 3D point cloud reconstruction using a deep pyramid network. *CoRR*, abs/1901.08906, 2019. 2, 3, 6

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, 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. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 4

[16] G. Peyré and M. Cuturi. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6) :355–607, 2019. 3

[17] A. Pumarola, S. Popov, F. Moreno-Noguer, and V. Ferrari. C-Flow : Conditional generative flow models for images and 3D point clouds, 2019. 2, 3

[18] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet : Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 77–85, 2017. 2

[19] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++ : Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5105–5114, 2017. 2

[20] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems 18*, pages 1161–1168. MIT Press, 2006. 2

[21] J. Shi and C. Tomasi. Good features to track. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 593–600. IEEE, 1994. 6

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 2

[23] R. Tylecek, T. Sattler, H.-A. Le, T. Brox, M. Pollefeys, R. B. Fisher, and T. Gevers. The second workshop on 3D Reconstruction Meets Semantics : Challenge results discussion. In *ECCV 2018 Workshops*, pages 631–644, Cham, 2019. Springer. 4

[24] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization : The missing ingredient for fast stylization. *arXiv preprint arXiv :1607.08022*, 2016. 4

[25] Y. Wang, W. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger. Pseudo-LiDAR from visual depth estimation : Bridging the gap in 3D object detection for autonomous driving. *CoRR*, abs/1812.07179, 2018. 2

[26] Y. Xia, Y. Zhang, D. Zhou, X. Huang, C. Wang, and R. Yang. Realpoint3d : Point cloud generation from a single image with complex background. *CoRR*, abs/1809.02743, 2018. 2

[27] Y. You, Y. Wang, W. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger. Pseudo-LiDAR++ : Accurate depth for 3D object detection in autonomous driving. *CoRR*, abs/1906.06310, 2019. 2