



HAL
open science

Détection de la ligne d'horizon et des points de fuite par apprentissage profond

Abdelkarim Ellassam, Marie-Odile Berger, Gilles Simon

► **To cite this version:**

Abdelkarim Ellassam, Marie-Odile Berger, Gilles Simon. Détection de la ligne d'horizon et des points de fuite par apprentissage profond. ORASIS 2021 - 18èmes journées francophones des jeunes chercheurs en vision par ordinateur, Centre National de la Recherche Scientifique [CNRS], Sep 2021, Saint Ferréol, France. hal-03339639

HAL Id: hal-03339639

<https://hal.science/hal-03339639v1>

Submitted on 9 Sep 2021

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.

Détection de la ligne d’horizon et des points de fuite par apprentissage profond

Abdelkarim Ellassam

Gilles Simon

Marie-Odile Berger

Inria, Université de Lorraine, LORIA, CNRS

abdelkarim.ellassam@inria.fr gilles.simon@loria.fr marie-odile.berger@inria.fr

Résumé

Nous étudions l’application des réseaux de neurones convolutifs (CNN) pour estimer la ligne d’horizon (LH) et détecter les points de fuite horizontaux (PFh) à partir d’une seule image. Les méthodes existantes sont généralement basées sur l’extraction de segments de droite et leur regroupement par points de convergence. Dans cet article, nous examinons différentes stratégies pour estimer les paramètres de la ligne d’horizon et détecter les points de fuite horizontaux à l’aide des CNNs. Les principales contributions sont les suivantes : I) Nous prouvons que prédire séquentiellement les paramètres à l’aide de CNN distincts est plus performant que de les prédire simultanément en utilisant un seul CNN. II) Nous démontrons que l’utilisation des paramètres de la ligne d’horizon comme contraintes géométriques sur la détection des PFh améliore la performance globale. III) Nous montrons l’effet de l’utilisation des cartes normales comme caractéristiques supplémentaires pour une meilleure détection des points de fuite. Nos méthodes permettent de détecter de multiples PFh sans être contraint par l’hypothèse d’un monde de Manhattan. Nous entraînons tous les réseaux sur la base de données Holicity, et nous obtenons d’excellents résultats sur la base de données HLW et des résultats compétitifs sur deux bases de données de référence.

Mots Clef

Ligne d’horizon, points de fuite, vision par ordinateur, apprentissage profond

Abstract

We investigate the application of convolutional neural networks (CNNs) to estimate the horizon line (HL) and detect horizontal vanishing points (hVP) from a single RGB image. Existing methods are typically based on extracting converging lines and clustering them to detect candidate vanishing points, then keeping the points that occur on the same line: the horizon line. In this paper, we examine different strategies for estimating the horizon line parameters and detecting the horizontal vanishing points using CNNs. The main contributions are as follows: I) We prove that predicting the parameters sequentially using separate CNNs outmatch using one single CNN to estimate all parameters simultaneously. II) We demonstrate

that using the HL parameters as geometric constraints on the hVP detection improves the overall performance. III) We showcase the effect of using normal maps as additional features for better detection of vanishing points. IV) We point out that all proposed methods enable us to detect multiple hVPs without being constrained to a Manhattan-world assumption. We train all proposed networks on the Holicity dataset, and we achieve state-of-the-art results on the Horizon Lines in the Wild dataset and competitive results on two benchmark datasets.

Keywords

Horizon line, vanishing points, computer vision, deep learning

1 Introduction



Figure 1: Estimation de la ligne d’horizon et détection des points de fuite. En jaune : sortie des réseaux. En bleu : vérité terrain.

La détection de la ligne d’horizon (LH) et des points de fuite (PF) est essentielle dans de nombreux problèmes de vision par ordinateur. La connaissance de ces caractéristiques est fondamentale dans diverses tâches de haut niveau telles que la calibration de la caméra [5, 22], la reconstruction 3D [9], l’estimation de pose [10], l’estimation

de la carte des normales [20, 21], la localisation et la cartographie simultanées (SLAM) [12] et bien d'autres. Des travaux récents ont montré que la LH peut être estimée de manière robuste par des méthodes d'apprentissage profond [4, 16, 24]. La détection par CNN des PF le long de la ligne d'horizon s'est avérée plus problématique, notamment parce que leur nombre attendu est inconnu à l'avance. La plupart des auteurs se sont appuyés sur des méthodes de vision classiques : détection et regroupement de segments de droite [24] ou en fixant le nombre de PF à détecter [4, 16].

Dans cet article, nous présentons différentes méthodes basées sur des CNN pour estimer les paramètres de la ligne d'horizon (pente/offset) et pour prédire les coordonnées d'un nombre quelconque de PF horizontaux (PFh). Comme tous les PFh appartiennent à la LH, leurs coordonnées en x le long de cette droite sont suffisantes pour les localiser ainsi le nombre de paramètres inconnus est réduit à $2+N$ où N est le nombre de PFh dans l'image. Nous proposons plusieurs stratégies pour estimer ces paramètres, simultanément ou en cascade tout en appliquant des contraintes géométriques pour l'estimation des paramètres. Toutes les méthodes proposées nous permettent de prédire les paramètres de la LH et de détecter des PFh multiples sans être contraint à l'hypothèse d'un monde de Manhattan.

Nous étudions également les avantages de l'utilisation des cartes de normales comme caractéristiques supplémentaires pour la détection des PF. Nous avons entraîné un réseau U-net [17] pour estimer une carte des normales à partir d'une seule image RVB. Nous démontrons que l'utilisation des cartes de normales produit des améliorations significatives de la performance globale.

Toutes les méthodes proposées ont été évaluées sur trois bases de données de référence standard. Nous obtenons les meilleurs résultats sur la base de données *Horizon Lines in the Wild* [23] et des résultats compétitifs sur deux bases de données de référence [1, 6]. Nous avons également évalué notre approche sur la base de données Holicity récemment introduite [25].

2 État de l'Art

Pour une caméra à sténopé, un point de fuite est un point du plan de l'image où se croisent les projections bidimensionnelles d'un ensemble de lignes parallèles dans l'espace tridimensionnel.

Détection directe des PF. La détection de points de fuite est un problème fondamental et difficile de la vision par ordinateur. Les méthodes existantes peuvent être divisées en deux catégories : Les méthodes basées sur les lignes et les méthodes d'apprentissage supervisé. L'approche dominante est celle où des segments de droite sont extraits [8] puis regroupés pour produire des candidats aux PF à l'aide d'algorithmes robustes tels que RANSAC [3], J-linkage [2] et la transformée de Hough [11]. L'algorithme Expectation-Maximization (EM) [14] est souvent utilisé

pour affiner la position des PF.

Plus récemment, de nombreux chercheurs ont tenté de résoudre ce problème en utilisant une approche basée sur les CNN. Borji [4] a proposé de diviser l'image en cellules et a utilisé un CNN pour prédire si un PF est présent dans chaque cellule. L'inconvénient de cette approche est qu'un seul PF à l'intérieur du cadre de l'image est détecté. Kluger et al. [13] ont proposé un CNN qui fonctionne sur la représentation en sphère gaussienne des lignes détectées dans l'image plutôt que sur l'image naturelle. Zhou et al. [26] ont présenté un nouveau CNN avec des opérateurs convolutifs inspirés de la géométrie pour détecter des PF. Pour exploiter les propriétés géométriques des PF en tant qu'intersections de lignes parallèles, les auteurs ont développé un nouvel opérateur appelé convolution conique : une convolution régulière mise en œuvre dans un espace conique canonique dans lequel les lignes structurelles locales sont toujours horizontales.

Dans la plupart des travaux, une contrainte sur le nombre de PF attendus est imposée. Dans [4, 16], seul le PF dominant est détecté. D'autres travaux [2, 14, 22] fonctionnent sous l'hypothèse d'un monde de Manhattan où ils supposent trois points de fuite mutuellement orthogonaux : le PF vertical (zénith) et deux PF horizontaux. La ligne d'horizon est simplement la ligne passant par les PF horizontaux. L'hypothèse d'un monde de Manhattan est en général valable pour les scènes urbaines simples. Cependant, dans les scènes plus complexes, cette hypothèse n'est plus valable. Une hypothèse moins stricte est l'hypothèse d'un monde d'Atlanta où l'on suppose l'existence d'un PF vertical et d'un nombre inconnu de PF horizontaux qui sont reliés par une LH commune. De nombreux travaux [24, 26], dont le nôtre, reposent sur cette hypothèse.

Détection préalable de la ligne d'horizon. La détection préalable de la LH a été introduite dans deux travaux [18, 24], tous deux basés sur la même idée : proposer un ensemble de lignes d'horizon candidates et les noter en fonction des PF qu'elles contiennent. La méthode proposée dans [18] repose sur une observation simple : toute ligne horizontale à la hauteur du centre optique de la caméra se projette sur la ligne d'horizon, quelle que soit sa direction. Ainsi, la LH peut être estimée comme une accumulation de segments de droite horizontale détectés dans l'image. Zhai et al. [24] ont utilisé un CNN pour extraire le contexte global de l'image et guider la génération d'un ensemble de candidats pour la LH. Et pour chaque candidat, les auteurs identifient les PF en résolvant un processus d'optimisation discrète. Le score final de chaque candidat à la LH est basé sur la cohérence des lignes de l'image avec les PFs sélectionnés. Dans [19], une approche basée sur les détections de points de fuite *a-contrario* est utilisée. Les PFh significatifs sont détectés en les contraignant sur la LH, ce qui réduit les fausses détections. Cette méthode obtient les meilleures performances sur deux ensembles de données de référence.

Estimation directe de la ligne d'horizon. Workman et

al. [23] dans la continuité des travaux de [24] ont proposé un seul CNN pour estimer directement la LH. Le réseau proposé est construit sur l'architecture GoogleNet pour estimer simultanément la pente et le décalage de la LH. J. Lee et al. [15] ont proposé un détecteur de lignes sémantiques utilisant un CNN. Les droites sémantiques sont détectées dans une image en classifiant et en régressant les lignes candidates à l'aide du réseau proposé. Pour l'estimation de la LH, cette méthode a atteint des performances de pointe sur le jeu de données HLW [23].

3 Vue générale de la méthode

Nous étudions plusieurs stratégies pour détecter la LH et prédire l'emplacement des PFh. Dans ce qui suit, la LH est paramétrée par sa pente $\theta \in [-\pi/2, \pi/2]$ et son décalage $r \in [-\infty, +\infty]$, où θ est l'angle entre la ligne d'horizon et l'axe horizontal de l'image et r la distance du point principal à la ligne d'horizon. Pour traiter les PF infinis, nous réduisons l'espace infini de r à l'intervalle $[-\pi/2, \pi/2]$ en appliquant l'inverse de la fonction tangente. Nous traitons le problème de détection de la LH et des PFh comme un problème de classification. La pente, le décalage et les coordonnées des PF sont convertis en classes catégorielles indépendantes, en divisant uniformément leurs domaines respectifs en 500 cases. Deux types de stratégies sont étudiées pour inférer la distribution des paramètres à estimer :

- **La stratégie directe** (Section 4) vise à inférer directement la pente, l'offset et les PFh. Cette stratégie s'inscrit dans la continuité des travaux de Zhai et al. [24] qui ont utilisé un CNN pour estimer simultanément la pente et le décalage de la LH. Les auteurs ont proposé un CNN avec deux branches disjointes entièrement connectées. De même, nous proposons un CNN unique avec trois branches disjointes entièrement connectées. Les deux premières branches estiment la pente et le décalage de la LH, la troisième branche prédit les coordonnées x des PFh.
- **Stratégies en cascade** (Section 5) : Ayant observé que le décalage estimé par la méthode directe est plus précis pour les images où la LH est horizontale, nous avons envisagé deux stratégies en cascade où la pente est d'abord estimée seule puis utilisée pour rectifier l'image, rendant la ligne d'horizon parallèle à l'axe x . Le décalage et les PF sont ensuite déduits de l'image rectifiée à partir de deux CNNs indépendants (méthode ROTATED). Pour mieux contraindre la détection des PFh à appartenir à la LH, nous proposons enfin la méthode FULL où l'image pivotée et le décalage prédit sont en entrée de la détection des PFh.

Les distributions des paramètres sont les sorties de ces modèles. L'étape finale de nos méthodes consiste alors à extraire les valeurs ayant le meilleur score pour la pente et le décalage. Pour les PFh, les modes significatifs maximaux sont détectés grâce à une méthode *a-contrario* [7].

Cette méthode probabiliste présente l'avantage principal de pouvoir détecter des pics constitués de cases adjacentes, tout en ne nécessitant pas un réglage délicat des paramètres.

Dans les sections suivantes, nous discutons en détail toutes les méthodes proposées et nous présentons une évaluation approfondie sur des jeux de données de référence existants.

4 Détection directe de la ligne d'horizon et des points de fuite



Figure 2: Détection directe de la LH et des PF (DIRECT).

La méthode DIRECT consiste à estimer les paramètres de la ligne d'horizon ainsi qu'un nombre quelconque de PFh à l'aide d'un seul CNN (Figure 2). L'architecture AlexNet est utilisée comme colonne vertébrale de notre CNN (Figure 3). Les couches convolutionnelles sont laissées intactes tandis que les couches entièrement connectées d'origine sont remplacées par trois ensembles disjointes de couches entièrement connectées, chaque branche estimant un paramètre.

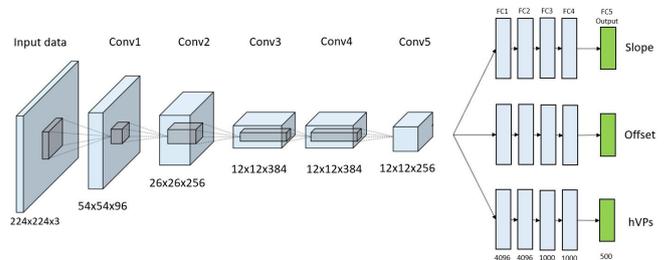


Figure 3: L'architecture AlexNet modifiée (AlexNet3).

Les deux premières branches estiment la pente et le décalage de la LH. La troisième branche prédit la coordonnée x du PFh le long de la LH. Le choix d'utiliser une approche de classification est crucial : bien que l'entraînement ne porte que sur les PFh dominants, nous pouvons obtenir plusieurs pics dans l'histogramme de sortie correspondant à des PFh multiples dans l'image (Figure 5). La méthode *a-contrario* [7] permet de détecter les modes significatifs maximaux de l'histogramme correspondant aux coordonnées x des PFh le long de la LH. Pour la pente et l'offset, nous extrayons simplement la valeur ayant le meilleur score.

La figure 4 illustre un exemple de résultat obtenu en sortie du réseau. La LH de la vérité terrain (GT) est représen-



Figure 4: Exemple de résultat de la stratégie DIRECT. En jaune : sortie du réseau. En bleu : vérité terrain.

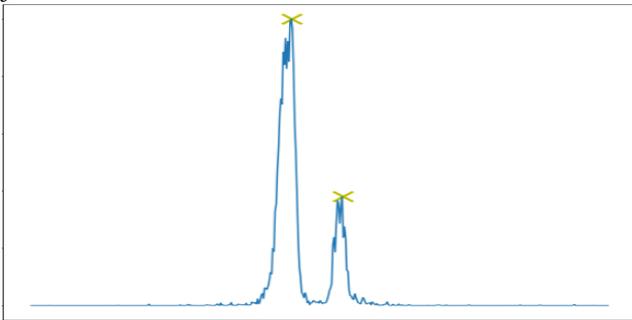


Figure 5: Détection *a-contrario* des modes de la distribution des PFh.

tée en bleu et la LH prédite en jaune. La figure 5 montre la détection *a-contrario* des modes de la distribution des PFh : deux modes sont détectés, correspondant aux deux PFh présents dans l'image.

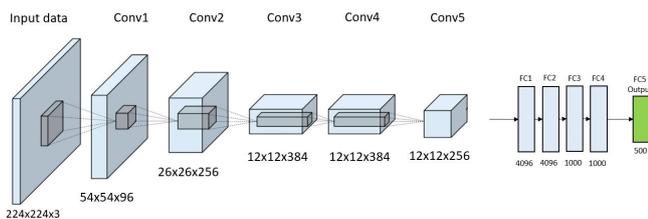


Figure 6: L'architecture AlexNet modifiée (AlexNet1).

5 Détection en cascade de la ligne d'horizon et des points de fuite

En utilisant la méthode DIRECT, nous avons observé que l'estimation de l'offset est plus précise pour les images où la LH est horizontale. Dans l'idée d'améliorer la détection de l'offset et des PFh, nous proposons de détecter d'abord la pente θ de la LH et d'effectuer ensuite la détection de l'offset et des PFh sur les images pivotées par une rotation

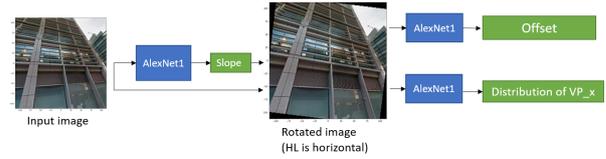


Figure 7: Détection de la ligne d'horizon et des points de fuite basée sur la géométrie (ROTATED).

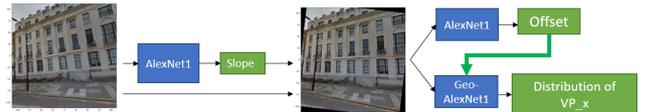


Figure 8: Détection de la ligne d'horizon et des points de fuite basée sur la géométrie (FULL).

$-\theta$. En procédant ainsi, nous limitons la recherche des PFh sur les lignes parallèles à l'axe des x et introduisons plus de spécificité dans le CNN. Cette idée est dans un esprit similaire à ce qui a été fait dans [26], où un système de coordonnées locales a été utilisé dans lequel la direction du pixel au point de fuite est horizontale. Nous proposons donc deux méthodes qui intègrent la rotation de l'image comme première étape. Dans l'approche ROTATED, deux branches distinctes sont utilisées pour prédire le décalage et les PFh. Dans l'approche FULL, la distribution du décalage et l'image pivotée sont considérées comme des entrées pour l'estimation des PFh.

Méthode ROTATED. Étant donnée une image, la pente de la ligne d'horizon est estimée à l'aide du premier CNN. Ensuite, l'image d'entrée est tournée de façon à ce que la ligne d'horizon soit horizontale dans l'image. L'image pivotée est envoyée à deux CNN pour estimer le décalage de la LH et les coordonnées des PFh dans cette image (Figure 7). La figure 10 montre un exemple de résultat de cette approche : l'estimation du décalage est plus précise que le résultat de la méthode DIRECT (Figure 9).

Méthode FULL. Étant donnée une image, nous estimons la pente de la LH en utilisant le premier CNN. Ensuite, nous faisons pivoter l'image d'entrée de sorte que la LH soit horizontale dans l'espace image. L'image pivotée est envoyée au deuxième CNN pour estimer le décalage de la LH. Enfin, l'image pivotée et le décalage prédit sont transmis au troisième CNN pour estimer les coordonnées x des PFh. La figure 8 montre la stratégie FULL. Nous utilisons la même architecture (Figure 6) pour estimer la pente puis le décalage de la LH. Le réseau est modifié pour intégrer le décalage comme une contrainte géométrique sur la détection PFh comme suit. Nous ajoutons le décalage comme une caractéristique extraite au vecteur de sortie des couches convolutionnelles avant les couches entièrement connec-

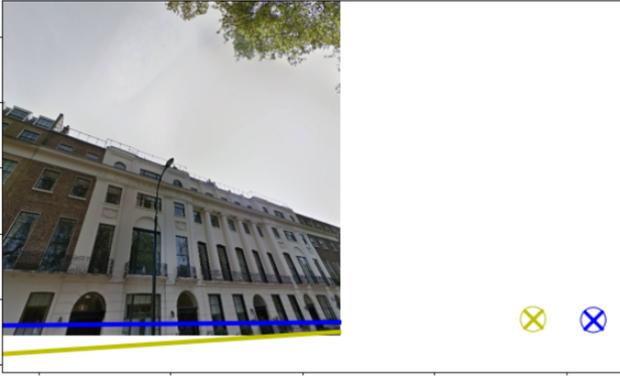


Figure 9: Exemple de sortie de la stratégie DIRECT. En jaune : sortie du réseau. En bleu : vérité terrain.

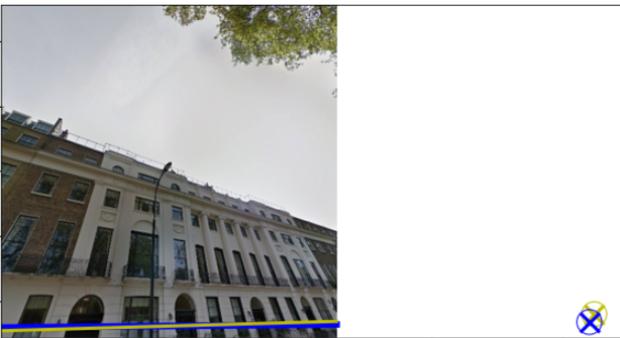


Figure 10: Exemple de sortie de la stratégie FULL. En jaune : sortie du réseau. En bleu : vérité terrain.

tées. Nous considérons le décalage comme un vecteur de 500 cases représentant une distribution de probabilité sur les LH possibles dans l'image. La figure 11 illustre la nouvelle architecture pour la détection des PF basée sur la géométrie. Dans la figure 10, nous montrons un exemple du résultat de l'approche proposée. La détection des PFh est plus précise après l'intégration du paramètre de la LH dans le réseau.

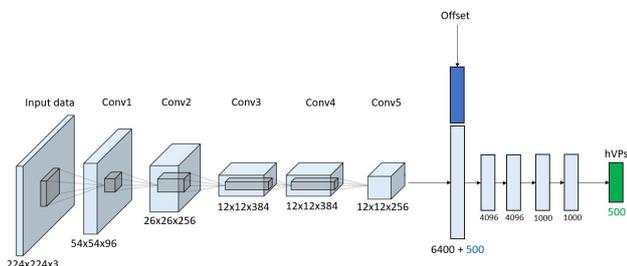


Figure 11: Geometric based vanishing point detection (Geo-AlexNet1)

6 Utilisation des cartes de normales

Avec les approches classiques basées sur les contours, des PF erronés peuvent être générés par des arêtes qui convergent vers un point sans représenter d'information structurelle. Des câbles suspendus ou des branches d'arbres peuvent générer de telles erreurs. Bien que n'étant pas explicitement basée sur les contours, l'approche CNN est sensible à ce problème comme le montre la figure 14 où nous constatons que la distribution obtenue pour le paramètre de décalage est très bruitée. Pour renforcer la détection des PF structurels, nous proposons d'utiliser des cartes normales comme caractéristiques supplémentaires de l'image d'entrée.

Dans ce nouveau modèle, l'entrée des CNN est obtenue en concaténant l'image avec la carte des normales correspondante. Bien que certains jeux de données, comme Holicity [25], fournissent des cartes normales pour chaque image, cette information n'est généralement pas disponible. Comme dans [25], nous avons donc entraîné un réseau U-net [17] à inférer de telles cartes. La figure 12 présente un exemple de carte des normales générée sur une image extraite du jeu de données Holicity que l'on peut comparer à la carte de vérité terrain. Notez que les auteurs de [25] ont généré des cartes normales à partir des modèles CAO 3D, par conséquent seules les surfaces structurelles sont présentes dans la carte, ce qui exclut des arbres et autres fils électriques.

7 Expérimentations

7.1 Implémentation

Nous avons implémenté tous les réseaux proposés en Python en utilisant Tensorflow et Keras et nous les avons exécutés sur un CPU Intel Xeon W-2133 @3.60GHz avec un GPU Nvidia Quadro RTX 4000. Nous entraînons chaque réseau en utilisant un optimiseur Adam avec une fonction de perte d'entropie croisée catégorielle. Nous initialisons les paramètres des réseaux de manière aléatoire. Nous entraînons les réseaux pendant 30 époques, en commençant par un taux d'apprentissage de 10^{-3} et nous utilisons la technique d'arrêt précoce pour éviter le sur-apprentissage. Toutes les images d'entrée ont été redimensionnées en $224 \times 224 \times 3$ pour l'entraînement et le test.

7.2 Jeux de données

Données d'entraînement. Toutes les méthodes considérées dans cet article ont été entraînées sur le jeu de données Holicity, un jeu de données 3D à l'échelle d'une ville contenant de riches informations structurelles. Il se compose de 6300 images du monde réel couvrant le centre-ville de Londres et fournit des étiquettes pour différentes structures 3D dans les environnements urbains, comme les cartes de normales et les points de fuite. Pour chaque image, nous avons calculé les paramètres de la LH directement à partir des coordonnées des PH. Puisque la LH est horizontale dans toutes les images fournies par Holicity,

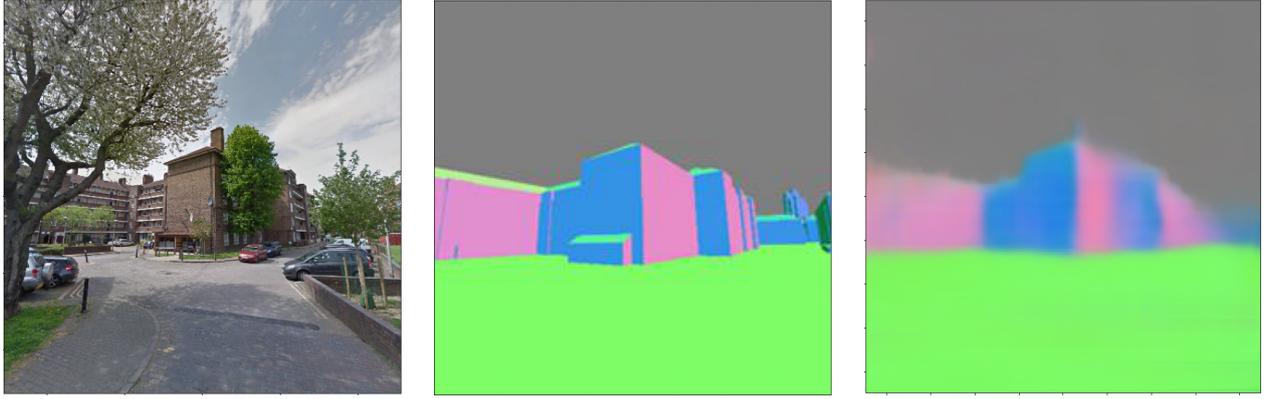


Figure 12: Un exemple de la sortie de l’U-Net (de gauche à droite : image d’entrée, carte GT, carte prédite).

nous avons fait pivoter les images de façon aléatoire en suivant une distribution normale avec un écart type de 5 degrés. Le jeu de données Holicity se compose de 37910 images étiquetées. Pour obtenir un jeu de données beaucoup plus important, nous avons utilisé des techniques d’augmentation des données en inversant aléatoirement l’image horizontalement avec une probabilité de 50%.

Données de test. Toutes les méthodes proposées sont testées sur des images pivotées provenant de l’ensemble de données Holicity. 20% de ce jeu de données est utilisé pour l’évaluation, les images étant pivotées de manière aléatoire selon une distribution normale. De plus, trois jeux de données de référence sont utilisés pour évaluer la détection de la ligne d’horizon : Eurasian Cities (EC), York Urban (York) et Horizon Line in The Wild (HLW). EC comprend 103 images urbaines extérieures contenant de nombreuses scènes non Manhattan. York comprend 102 images de scènes extérieures et intérieures, principalement des scènes du campus de l’Université de York et du centre-ville de Toronto, au Canada. Pour chaque image, les paramètres de la caméra et les PF GT sont fournis. Toutes les images satisfont à l’hypothèse du monde de Manhattan et trois PF orthogonaux sont donc fournis. HLW contient des images du monde réel avec des lignes d’horizon étiquetées. Ce jeu de données est beaucoup plus grand et plus difficile que les précédents. Beaucoup de ses images ne satisfont pas à l’hypothèse du monde de Manhattan, et sont capturées dans un ensemble diversifié d’environnements. La plupart des images sont des photos de vacances prises par des touristes, montrant des environnements artificiels. Pour certaines images, des points de repère et des statues occupent une grande partie de l’image, etc. Toutes les méthodes sont évaluées sur l’ensemble de test fourni, qui contient 2018 images étiquetées.

7.3 Métriques

Nous utilisons deux métriques : l’erreur de détection de la HL et l’erreur de détection des PF.

L’**erreur de détection de la LH** est définie comme la distance euclidienne maximale entre la LH estimée et la LH

GT dans les limites de l’image, divisée par la hauteur de l’image. Le pourcentage de l’aire sous la courbe (AUC) de l’histogramme cumulatif des erreurs dans le domaine $[0, 0.25] \times [0, 1]$ est calculé.

Erreur de détection des PF. Pour chaque PF prédit, nous considérons le PF GT le plus proche. Nous calculons l’erreur d’angle entre les vecteurs reliant le centre optique au PF GT et, resp., au PF prédit. Nous comptons enfin le pourcentage de prédictions dont l’erreur d’angle est inférieure à un seuil prédéfini. En faisant varier le seuil, une courbe de précision de l’angle peut être tracée. *AA* est défini comme l’aire sous la courbe entre $[0, 20^\circ]$ divisée par 20° . Pour la détection des PF dominants, une seule erreur d’angle est calculée par image (*AA – DomPF*). Pour la détection de PFs multiples, nous considérons l’erreur d’angle associée à chaque PF prédit (*AA – MultPF*).

7.4 Évaluation

Dans le tableau 1, nous comparons nos méthodes avec les approches proposées dans [19, 23, 24]. Dans cette première expérience, nos méthodes ont été entraînées sur HLW pour une comparaison équitable avec [23, 24] qui a été entraîné sur ce jeu de données. La comparaison est effectuée avant l’étape de raffinement et d’optimisation : en effet, la plupart des méthodes utilisent un algorithme EM pour raffiner les PFs sur la base des segments de droite détectés dans l’image. Nous obtenons la meilleure performance sur le jeu de données HLW en utilisant toutes les méthodes proposées. Nous obtenons une AUC de 71,62% avec DIRECT et une AUC de 75,07% avec FULL, améliorant ainsi de 16,98% la meilleure performance précédente de Workman et al. ¹.

Dans toutes les expériences suivantes, l’entraînement est effectué sur le jeu de données Holicity. L’évaluation est présentée en table 3. Nous obtenons une AUC de 87,52% en utilisant FULL. Afin d’apprécier l’impact de la précision du décalage sur la détection des PF, nous présentons également les performances de FULL si le décalage est parfaitement prédit. La figure 13 illustre les performances

¹Amélioration relative $= (AUC_{new} - AUC_{old}) / (1 - AUC_{old})$

Method	HLW	EC	York
Zhai [24] (CNN + EMPTY)	49.03%	67.64%	73.67%
Workman [23] (CNN)	69.97%	83.96%	85.33%
Simon et al. [19]	45.88%	72.78%	70.6%
Ours (DIRECT)	71.62%	66.22%	60.26%
Ours (ROTATED/FULL)	75.07%	75.68%	68.05%

Table 1: Erreurs (AUC) sur la LH (entraîné sur HLW).

Method	Holicity	EC	York
Ours (DIRECT)	86.46%	57.15%	66.54%
Ours (ROTATED/FULL)	87.52%	72.71%	81.03%

Table 2: Erreurs (AUC) sur la LH (entraîné sur Holicity).

des méthodes proposées sur quelques exemples du jeu de données Holicity. Nous démontrons que l'utilisation des paramètres de la LH comme contraintes géométriques sur la détection PFh améliore la performance globale. Nous obtenons une amélioration relative de 8,64% dans la détection des PF multiples et de 3,26% dans la détection des PF dominants avec la méthode FULL par rapport à la méthode DIRECT.

La table 4 montre l'effet de l'utilisation des cartes de normales comme caractéristiques supplémentaires. En utilisant une carte des normales GT, nous améliorons considérablement les performances de détection de la LH et des PF. En utilisant les cartes de normales prédites en conjonction avec la méthode DIRECT, nous obtenons de meilleurs résultats que toutes les méthodes proposées. Nous obtenons une amélioration relative de 24,74% pour la détection des PF multiples et de 21,71% pour la détection la LH. De plus, les PF dans les images avec des branches d'arbres sont prédits correctement alors qu'ils étaient très perturbés auparavant. La figure 14 illustre un exemple de sortie de la stratégie DIRECT pour une image avec des arbres. Nous remarquons que la sortie pour la détection des PFs est très bruitée. Cependant, après avoir ajouté la carte des normales GT en entrée, la sortie devient propre et nous pouvons détecter correctement les pics dans l'histogramme (Figure 15) et extraire les PFs significatifs. En utilisant la carte des normales prédite, un seul PF est détecté (Figure 16). Cependant, la détection est très précise, ce qui est prometteur. Des travaux sont en cours pour affiner la carte des normales prédite afin d'obtenir de meilleures performances.

	DIRECT	ROTATED	FULL	GT offset
AUC	86.46%	87.52%	87.52%	93.30%
AA - Dom PF	70.24%	70.78%	71.21%	73.01%
AA - Mult PF	67.83%	68.90%	70.61%	70.87%

Table 3: Comparaison entre les stratégies proposées (testées sur le jeu de données Holicity).

	without normal maps	with GT normal maps	with pred normal maps
AUC	86.46%	91.84%	89.40%
AA - Dom PF	70.24%	83.45%	72.73%
AA - Mult PFs	67.83%	78.02%	75.79%

Table 4: The effect of using normal maps as additional input features (tested on Holicity dataset)

8 Conclusion

Dans ce papier, nous avons présenté différentes stratégies pour estimer les paramètres de la ligne d'horizon et détecter les points de fuite horizontaux sur la LH en utilisant des CNN. La méthode FULL tire parti des paramètres de la ligne d'horizon en tant que contraintes géométriques pour améliorer la performance globale. Nous avons montré l'importance de l'utilisation des cartes de normales comme caractéristiques supplémentaires pour une meilleure détection des points de fuite. L'utilisation d'une carte prédite est très prometteuse et n'a été testée qu'avec la méthode DIRECT. Les travaux en cours portent sur l'évaluation de l'utilisation de la carte avec la méthode FULL. Toutes les méthodes proposées peuvent détecter de multiples points de fuite horizontaux, même en dehors de l'image, sans être contraintes à l'hypothèse d'un monde de Manhattan. Nos méthodes sont les premières, à notre connaissance, à être capables d'estimer un nombre quelconque de points de fuite horizontaux en utilisant uniquement des réseaux neuronaux.²

References

- [1] Olga Barinova, Victor Lempitsky, Elena Tretyak, and Pushmeet Kohli. Geometric image parsing in man-made environments. *International Journal of Computer Vision*, 97:57–70, 07 2010.
- [2] J.-C Bazin, Yongduek Seo, Cédric Demonceaux, Pascal Vasseur, Katsushi Ikeuchi, In-So Kweon, and Marc Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–645, 06 2012.
- [3] R. Bolles and M. Fischler. A ransac-based approach to model fitting and its application to finding cylinders in range data. In *IJCAI*, 1981.
- [4] Ali Borji. Vanishing point detection with convolutional neural networks. *CoRR*, abs/1609.00967, 2016.
- [5] William Chen and Bernard C. Jiang. 3-d camera calibration using vanishing point concept. *Pattern Recognition*, 24(1):57–67, 1991.

²Ce travail fait partie du projet MoveOn entre l'INRIA et le Centre allemand de recherche en intelligence artificielle (DFKI).

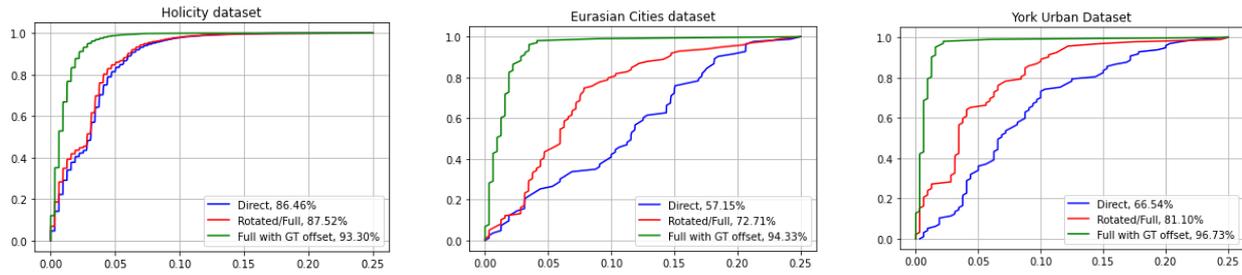


Figure 13: Erreurs (AUC) obtenues sur la ligne d'horizon dans les trois jeux de données en fonction de la stratégie considérée.

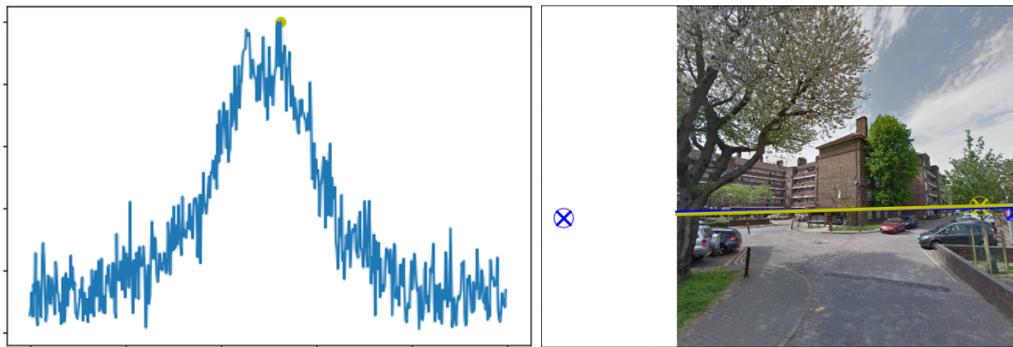


Figure 14: Exemples de résultats obtenus avec la stratégie DIRECT sans la carte des normales. Gauche: distribution des PFh et mode significatif maximal. Droite: PF et LH prédits (en jaune) et GT (en bleu).

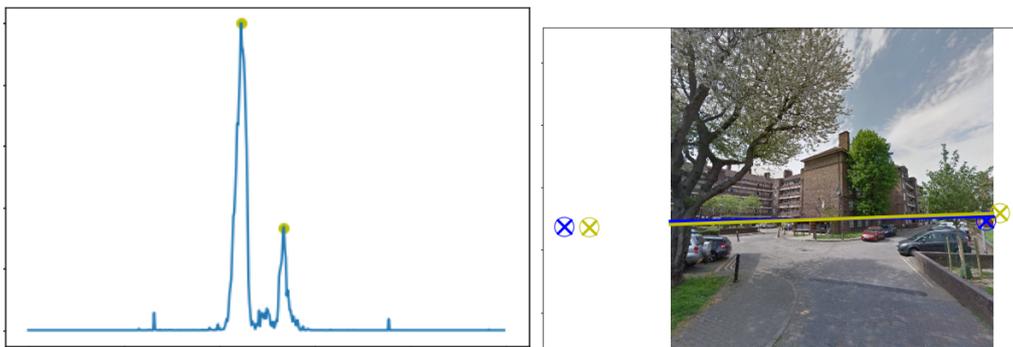


Figure 15: Exemples de résultats obtenus avec la stratégie DIRECT et la carte des normales GT (voir Figure 14).

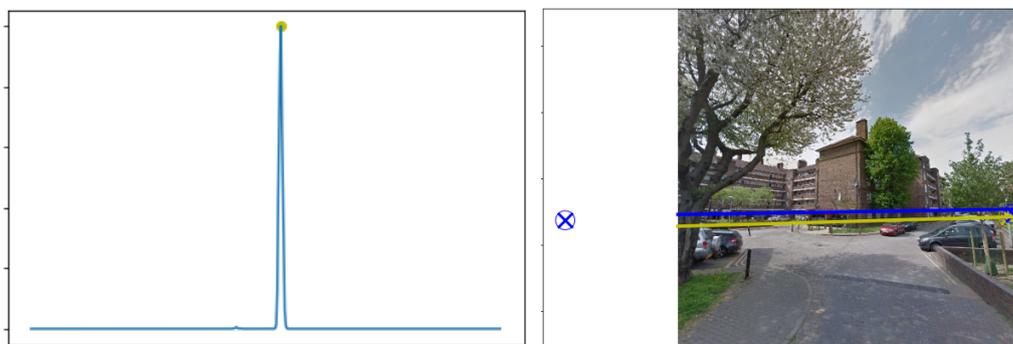


Figure 16: Exemples de résultats obtenus avec la stratégie DIRECT et la carte des normales prédite (voir Figure 14).

- [6] Patrick Denis, James Elder, and Francisco Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *European Conference in Computer Vision (ECCV)*, pages 197–210, 10 2008.
- [7] A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis: A Probabilistic Approach*. Springer Publishing Company, Incorporated, 1st edition, 2007.
- [8] Rafael Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence*, 32:722–32, 04 2010.
- [9] E. Guillou, Daniel Méneveaux, E. Maisel, and K. Bouatouch. Using vanishing points for camera calibration and coarse 3d reconstruction from a single image. *The Visual Computer*, 16:396–410, 2000.
- [10] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.
- [11] P. Hough. Machine analysis of bubble chamber pictures. 1959.
- [12] Yonghoon Ji, A. Yamashita, and H. Asama. Rgb-d slam using vanishing point and door plate information in corridor environment. *Intelligent Service Robotics*, 8:105–114, 2015.
- [13] Florian Kluger, Hanno Ackermann, Michael Ying Yang, and Bodo Rosenhahn. Deep learning for vanishing point detection using an inverse gnomonic projection. *CoRR*, abs/1707.02427, 2017.
- [14] Jana Košecká and Wei Zhang. Video compass. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision — ECCV 2002*, pages 476–490, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
- [15] Jun-Tae Lee, Han-Ui Kim, Chul Lee, and Chang-Su Kim. Semantic line detection and its applications. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3249–3257, 2017.
- [16] Yin-Bo Liu, Ming Zeng, and Qing-Hao Meng. D-vpnet: A network for real-time dominant vanishing point detection in natural scenes. *CoRR*, abs/2006.05407, 2020.
- [17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.
- [18] Gilles Simon, Antoine Fond, and Marie-Odile Berger. A Simple and Effective Method to Detect Orthogonal Vanishing Points in Uncalibrated Images of Man-Made Environments. In *Eurographics 2016*, Lisbon, Portugal, May 2016.
- [19] Gilles Simon, Antoine Fond, and Marie-Odile Berger. A-Contrario Horizon-First Vanishing Point Detection Using Second-Order Grouping Laws. In *ECCV 2018 - European Conference on Computer Vision*, pages 323–338, Munich, Germany, September 2018.
- [20] Rui Wang, David Geraghty, Kevin Matzen, Richard Szeliski, and Jan-Michael Frahm. Vplnet: Deep single view normal estimation with vanishing points and lines. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 686–695, 2020.
- [21] Xiaolong Wang, David F. Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. *CoRR*, abs/1411.4958, 2014.
- [22] Horst Wildenauer and Allan Hanbury. Robust camera self-calibration from monocular images of manhattan worlds. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2831–2838, 2012.
- [23] Scott Workman, Menghua Zhai, and Nathan Jacobs. Horizon lines in the wild. *CoRR*, abs/1604.02129, 2016.
- [24] Menghua Zhai, Scott Workman, and Nathan Jacobs. Detecting vanishing points using global image context in a non-manhattan world. *CoRR*, abs/1608.05684, 2016.
- [25] Yichao Zhou, Jingwei Huang, Xili Dai, Linjie Luo, Zhili Chen, and Yi Ma. HoliCity: A city-scale data platform for learning holistic 3D structures. 2020. arXiv:2008.03286 [cs.CV].
- [26] Yichao Zhou, Haozhi Qi, Jingwei Huang, and Yi Ma. NeurVPS: Neural Vanishing Point Scanning via Conic Convolution. *CoRR*, abs/1910.06316, 2019.