Bibliographie :

Depuis plusieurs années**,** la vision par ordinateur n’a cessé de se développer, d’autant plus avec la multiplication des techniques de Machine Learning et leur accessibilité grandissante. Le plus souvent lorsque l’on entend parler de vision par ordinateur cela à lien avec les voitures autonomes ; l’interprétation des images est donc essentielle et c’est pourquoi les algorithmes de détection et de classification automatique d’objets ou d’obstacles sont de plus en plus divers et deviennent encore plus performantsd’année en année.

Cependant d’autres défis liés à la vision par ordinateur et à l’interprétation des images, reçoivent moins d’attention, notamment l’estimation de distance ou encore le calibrage de caméras à partir d’image amateur unique, alors qu’ils sont pourtant essentiels pour la bonne compréhension et prévision des obstacles dans le cas des véhicules motorisés par exemple.

L’estimation de la distance et le calibrage de caméras sont assez liés de nature et les études que nous avons étudiées répondent souvent aux deux problématiques en même temps.

En effet pour résoudre le problème du calibrage de caméras seulement à partir des informations présentes sur une image quelconque deux choix sont possibles, déterminer ce qu’on appelle la matrice de passage de la caméra **[3]** (composée d’une matrice intrinsèque et une matrice extrinsèque) ou alors déterminer les coordonnées 3D réelles X, Y et Z **[1][2]**.

# Estimation de la profondeur et des coordonnées :

## Avec du machine learning :

Il existe de nombreuse méthode permettant de calculer la profondeur d’un objet par rapport à un observateur, que ça soit en utilisant la géométrie des piétons [12]

La méthode permettant d’estimer les coordonnées qu’ont utilisé Jing Zhu et al. (2019) **[1]**, consiste à utiliser les différentes caractéristiques propres à chaque objet (taille, géométrie, forme) afin d’en déduire leur distance par rapport à la caméra ainsi que leurs coordonnées dans l’espace 3D de la scène. C’est une étude très intéressante car c’est à partir de ces coordonnées 3D que l’on pourra remonter à la matrice de passage de la caméra et ainsi la calibrer totalement.

Leur méthode se base entièrement sur le machine learning et elle consiste d’abord à faire passer les images dans un feature extractor (VGG16 et RES50) afin d’extraire et de mettre en évidence les différents obstacles présents dans l’image puis dans un ROI Pooling afin d’avoir un vecteur de taille fixe en sortie. On injecte ensuite ce vecteur dans un estimateur de distance (RN à 3 couches) qui va nous donner les coordonnées Z, dans un keypoint regressor (RN à 3 couches) pour les coordonnées X et Y et aussi dans un classificateur (RN à 1 couche). Ces différents réseaux de neurones ont ensuite été entrainés et testés à partir de dataset KITTI et nuScenes qui sont composés de différentes vidéos prises en ville contenant donc en général des obstacles tels que des individus, des voitures ou deux roues et il en ressort que leur algorithme d’estimation de distance est en moyenne plus performant et plus rapide que les méthodes d’estimation de distance connues et utilisées telle que l’inverse perspective mapping ou bien encore le support vector regression. Cette étude est très utile car ils calculent les coordonnées 3D et le calibrage est donc théoriquement faisable cependant pour l’entrainement le manque de dataset est un réel problème et notamment pour le keypoint regressor qui calcul les coordonnées X et Y.

La seconde étude**,** elle aussi se concentre sur l’estimation de la distance de la caméra avec des obstacles. Pour cela**,** elle va se baser sur l’étude de la géométrie des bounding box (boîtes d’ancrage en français) des objets présents sur une image. Les bounding box déterminent les coordonnées spatiales minimale et maximale d’un obstacle sur une image. Chaque classe d’objet ayant une taille moyenne de bounding box (chaque humain aura une bounding box plus ou moins équivalente à une distance donnée de la caméra et cela vaut aussi pour chaque catégorie d’objet), la méthode va ensuite utiliser ces informations pour déterminer une relation entre la taille de la bounding box d’un objet avec sa distance par rapport à la caméra.

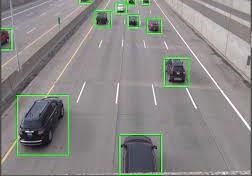


Figure 1 : Bounding Box (boîte d’ancrage) en vert

L’un des intérêts primordiaux pour l’étude de Muhammad Abdul Haseeb et al. (2018) est son utilisation d’images uniques comme seule entrée d’algorithme. En effet, l’une des méthodes habituelles et d’utiliser deux caméras afin de trianguler la position des objets dans un espace 3D. Cette méthode quant à elle va consister dans un premier temps à déterminer les bounding box de tous les obstacles présents dans l’image, pour cela l’image est d’abord injectée dans le réseau de neurones convolutionnels YOLOv3. Une fois la carte des features et des boundings box récupérée, on va venir calculer la géométrie de chaque bounding box (hauteur, largeur et diagonale) et l’injecter dans DisNet, le réseau de neurones développé dans le cadre de leur recherche et dont le but est de déterminer la distance des objets présents dans n’importe quelle image par rapport à la caméra.

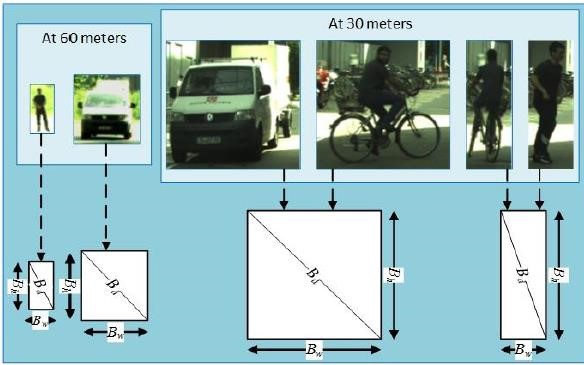


Figure 2 : Caractéristiques géométriques des bounding box

DisNet se base sur le fait que la distance d’un objet est inversement proportionnelle aux caractéristiques géométriques de sa bounding box (largeur, hauteur, diagonale).

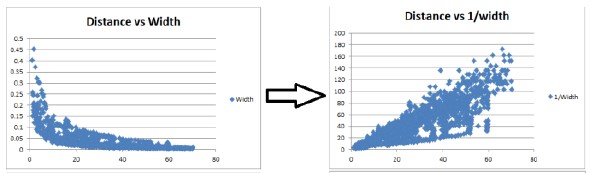


Figure 3 : Relation entre la distance et la largeur de la bounding box a gauche et avec là l’inverse de la largeur à droite

Sachant cela, Disnet a ensuite été entrainé à partir du dataset COCO. Cet algorithme étant d’abord créé pour les chemins de faire il a été testé en partie à l’avant d’un train, il a ensuite été testé dans des rues. Disnet est très précis pour tous les obstacles étant à moins de 100m de la caméra, cependant il a du mal aussi lorsqu’il y a beaucoup d’objets à l’image.

Cette étude est très intéressante car la méthode qu’utilise le réseau de neurones DisNet est applicable à tout type d’image et sans information ou contexte sur l’image.

L’une des études annexes est celle de Cao et Al [5], l’objectif y est de prédire la distance dans une large gamme de mesure. C’est-à-dire qu’au lieu de donner une distance précise ils donnent une gamme de distance dans laquelle est comprise la bonne valeur. Ils traitent ainsi le problème d’évaluation de profondeur comme un problème de classification et non de régression. Cette étude est intéressante car elle utilise une méthode différente de celle que nous allons utiliser cependant elle est loin d’être assez précise pour calibrer une caméra.

L’étude de Pedrazzine [6] se concentre quant à elle sur l'estimation de la position 3D d’un objet en plein vol. Comme pour l’étude de Cao et al. Ils utilisent un dataset basé sur un gradient de profondeur afin de pouvoir entrainer leur modèle. Cependant dans cette étude ils utilisent aussi la calibration de caméra, donc ce n’est pas possible de l’utiliser dans notre étude.

## Sans machine learning :

L’objectif du stage est de proposer une méthode d’estimation de profondeur et de calibrage de caméra en utilisant le machine learning principalement cependant il existe aussi de nombreux algorithmes et transformations mathématiques qui peuvent permettre d’arriver au moins à une estimation de mouvement d’un obstacle dans l’espace et cela sans utiliser de machine learning [11] . C’est aussi le cas pour l’étude de Mallot et al. [4] datant de 1991.

Tout d’abord il est important d’avoir une idée précise d’un obstacle dans le cadre de leur étude. L'hypothèse de base sous-jacente peut être énoncée comme suit :

L'observateur ne dispose que de trois degrés de liberté pour mouvement : les translations avant-arrière et gauche-droite, et la rotation autour de l'axe orthogonal à ces deux translations. De cette hypothèse, il découle que le mouvement est confiné au plan horizontal défini par les deux translations possibles. Tout ce qui se trouve en dehors de ce plan est un obstacle, puisque la trajectoire de l'observateur ne peut le traverser. En d'autres termes, nous avons une définition de l'obstacle qui n'exige pas qui ne nécessite pas de reconnaissance de formes d'aucune sorte et qui, d'autre part, ne sera pas confondu par des ombres portées ou des contours dans le plan horizontal.

En ayant cette idée d’un obstacle en tête on peut d’ores et déjà catégoriser ce qui est un obstacle et ce qui ne l’est pas sans avoir recours à des algorithmes de détection d’obstacle complexes se basant sur la reconnaissance de motif par exemple.

Ils utilisent ensuite l’inverse perspective mapping afin de déterminer le flux optique (le mouvement d’un objet par rapport à l’observateur). L’inverse perspective mapping est utilisé afin de réduire les effets de la perspective sur l’image (et ainsi pouvoir en tirer des informations plus facilement).

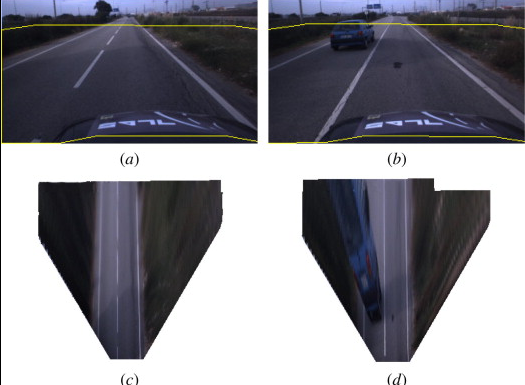


Figure 4 : Utilisation de l'IPM

Cela permet de passer d’un plan 3D à un plan horizontal 2D, il est ainsi plus aisé de travailler et ainsi pouvoir en déterminer les informations que l’on souhaite tel que le mouvement des objets dans la scène. Une fois le flux optique des objets dans la scène déterminée ils en déduisent si c’est un obstacle ou non. Cette méthode est très intéressante cependant elle nécessite beaucoup de prérequis et ne peut donc pas être utilisé efficacement dans le cadre du stage.

# Estimation des paramètres intrinsèque

Comme il a été dit auparavant, pour le calibrage de caméras deux choix peuvent être faits, connaitre les coordonnées 3D et 2D afin de retrouver la matrice de passage de la caméra ou alors déterminer la matrice de passage afin de calculer les coordonnées 3D

L’une des manières les plus rapides pour calibrer les caméras sans mire de calibrage comme ont pu le faire André Vitor et Al. [13] est d’utiliser deux caméras [10], cependant ce n’est pas l’objectif de ce stage.

Afin de calibrer une caméra sans avoir besoin de deux images, Oleksandre et al (2018) **[2]** ont mis au point Deepcalib, un réseau de neurone convolutif donc l’objectif est de calibrer une caméra de manière automatique sans aucun support ou données préalable (type de caméra, ligne d’horizon, etc..). L’objectif de Deepcalib est de déterminer la matrice intrinsèque de la caméra (la distorsion et la focale) seulement à partir d’une image amateure ou professionnelle.

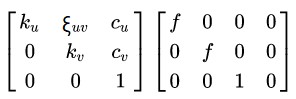


Figure 5 : Matrice intrinsèque de la caméra

Pour répondre à cette problématique, l’étude propose trois modèles de réseaux de neurones possible pour Deepcalib, les trois sont toujours composés d’un ou plusieurs CNN avec en sortie la distorsion et la focal**e** (réseau de neurone**s** convolutif).

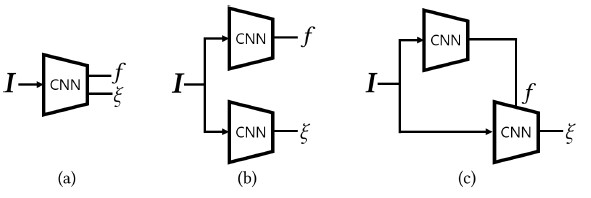


Figure 6 : Architecture de deepcalib (3 possibilités)

Où **ξ** est la distorsion de l’image et **f** la focal de la caméra.

L’un des challenges pour leur modèle a été de trouver un dataset adéquat où la distorsion et la focale pour chaque image est annotée.

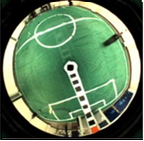


Figure 7 : exemple d'image omnidirectionnelle

Pour cela ils ont créé leur propre dataset à partir d’images omnidirectionnelles issues du dataset SUN360 grâce à ce qu’on appelle une projection stéréographique qui consiste à projeter une sphère 3D sur un plan 2D image.

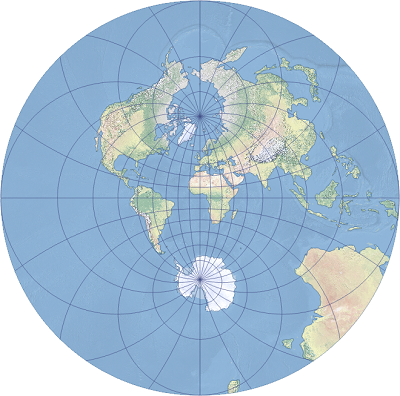
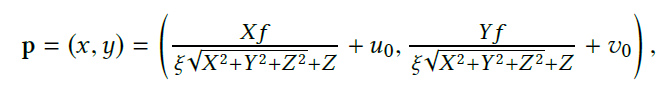
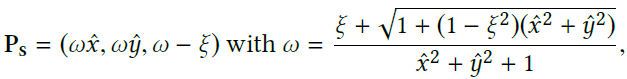


Figure 8 : Projection stéréographique de la terre

Cette projection permet d’avoir un modèle de distorsion connu mathématiquement et facile à calculer avec un algorithme. Au moyen de ce modèle mathématique il est possible de générer une large quantité d’images pour entrainer et valider leur modèle de CNN. Ils ont ainsi généré des images avec diverses distorsions et distance focale afin d’entrainer efficacement leur réseau de neurones.





Pour évaluer l’efficacité de Deepcalib, ils l’ont comparé face à des méthodes decalibrage de caméra plus traditionnelle qui contrairement à deepcalib, utilise diverses méthodes en plus de l’image pour calibrer une caméra telle qu’une mire de calibrage ou encore les lignes d’horizon sur l’image. Les résultats montrent que Deepcalib est utilisable avec n’importe quel type de caméra, quelle que soit sa distance focale ou sa distorsion alors que l’algorithme précédent fonctionne principalement avec la gamme de caméras pour lesquelles ils ont été créé et ne sont donc pas polyvalent.

La méthode Deepcalib est très utile pour ce stage car elle propose directement une méthode de calibrage de caméras ainsi qu’un procédé permettant de créer un dataset important.

Une autre manière de déterminer la profondeur des objets présents sur une image est l’utilisation du principe de Structure from Motion, qui est une technique d'imagerie par intervalle [photogrammétrique](https://fr.wikipedia.org/wiki/Photogramm%C3%A9trie) destinée à estimer la structure 3D de quelque chose à partir d'images 2D. C’est-à-dire qu’on peut reconstruire un objet en 3D lorsque l’on a plusieurs points de vue du même objet sous différent angles.

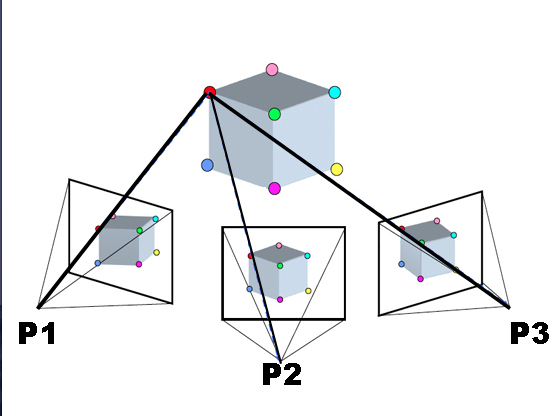


Figure 9 : Procédé de structure acquise à partir d'un mouvement (SfM)

C’est exactement le principe sur lequel se base la méthode d’estimation de profondeur de H Ha et al. [8], qui a pour objectif de déterminer une carte complète de la profondeur d’une vidéo ou d’une image de la manière suivante :

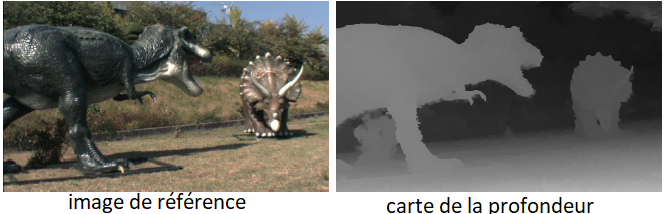


Figure 10 : carte de la profondeur

L’objectif de cette étude est de pouvoir calibrer n’importe quelle caméra (estimer sa distorsion radiale ainsi que sa distance focale) à partir de clip vidéo en s’aidant du mouvement de la caméra. Contrairement au procédé classique du SfM qui nécessite de prendre des images de l’objet sous différent angle assez large, cette méthode ne nécessite pas plus qu’un extrait d’une seconde pour pouvoir déterminer les paramètres intrinsèques de la caméra. Cependant son inconvénient est qu’il nécessite d’avoir une vidéo ou plusieurs images de la même scène, le bon fonctionnement du processus dépend donc de la qualité de la vidéo, son nombre d’image et du mouvement effectué par la caméra (si la translation de la caméra est trop faible cela ne peut fonctionner).

# La détection d’objet :

**Faster R-CNN :**

Dans le monde de la vision par ordinateur la détection d’objet est de plus en plus efficace, cela est dû au développement de diverses méthodes telles que l’utilisation de point clé notamment avec une des méthodes récentes et prometteuses concerne les Region Proposal Network (réseau de propositions de région), et plus particulièrement les RPN qui utilisent des CNN. Ces réseaux prennent en entrée une image quelconque (sans aucune information sur l’image ou la caméra) et donnent en sortie la boîte d’ancrage des objets ainsi que leur classification. C’est ce que propose l’étude de Shaoqing Ren et al. [7] avec leur Faster R-CNN qui permet d’analyser des vidéos en temps réel. Le Faster R-CNN est un algorithme de deep learning basé sur les CNN qui est donc un Region Proposal Network. Comme pour YOLO la différence avec des algorithmes de classification et de détection d’objet, l’une des méthodes des RPN est de diviser l’image en entrée en un grand nombre de régions et de regarder si elles contiennent ou non un objet. Une fois les régions déterminées, on va ensuite chercher la position de chaque objet en plaçant dans chaque région une ancre qui va indiquer la position des objets possible. Dans l’étude proposée [7] ils ont utilisé deux modules afin de pouvoir déterminer précisément la classe et la position des objets présents dans la scène. Le premier module détermine les différentes régions qui contiennent des objets et le second module (Fast R-CNN la version antérieure à Faster R-CNN) va quant à lui détecter les objets et les classifier. Leur méthode montre des résultats très satisfaisant et une utilisation du CPU modérée cependant dans le cadre du stage nous avons décidé d’utiliser le modèle YOLOV3 en tant que region proposal network car il permet de détecter et classifier les objets en une étape contrairement à Faster R-CNN qui fait cela en deux étapes et peut donc avoir du mal à suivre pour une détection en temps réel. YOLO est aussi le RPN utilisé dans l’étude Disnet [2] qui est une des études principales lors de ce stage.

**YOLO :**

YOLO [9] est un algorithme qui utilise des réseaux neuronaux pour assurer la détection d'objets en temps réel. Cet algorithme est populaire en raison de sa rapidité et de sa précision. Il a été utilisé dans diverses applications pour détecter les feux de circulation, les personnes ou encore les animaux. Nous utilisons YOLO lors de ce stage avec DisNet afin de détecter la classe des objets et de déterminer les boites d’ancrage de ces derniers.

YOLO est très intéressant car il est plus rapide que son confrère Faster R-CNN et permet donc d’analyser une image en temps réel. YOLO prend en entrée une image et la divise en un nombre fixe de grille

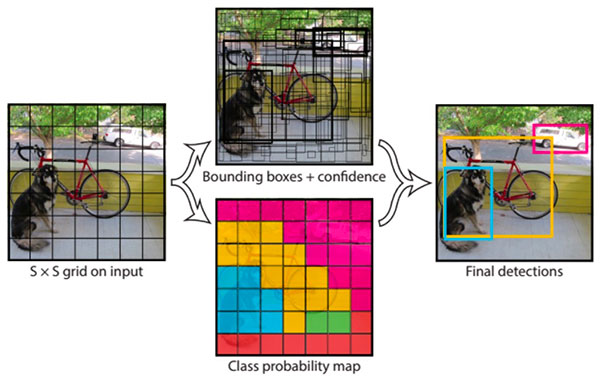


Figure 11 : Procédé de YOLO

L’algorithme va ensuite faire passer chaque grille dans un classificateur d’image et va détecter dans chaque zone s’il y a un objet, comme pour Faster R-CNN [7]. Une fois les objets identifiés, YOLO va ensuite déterminer leurs boites d’ancrages ainsi que leurs coordonnées dans l’image.

**Références :**

[1] Zhu, J., & Fang, Y. (2019). Learning object-specific distance from a monocular image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 3839-3848).

[2] Haseeb, M. A., Guan, J., Ristic-Durrant, D., & Gräser, A. (2018). DisNet: a novel method for distance estimation from monocular camera. *10th Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS*.

[3] Bogdan, O., Eckstein, V., Rameau, F., & Bazin, J. C. (2018, December). DeepCalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production* (pp. 1-10).

[4] Mallot, H. A., Bülthoff, H. H., Little, J. J., & Bohrer, S. (1991**).** Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, *64*(3), 177-185.

[5] Cao, Y., Wu, Z., & Shen, C. (2017). Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(11), 3174-3182.

[6] Pedrazzini, F. (2018). 3D Position Estimation using Deep Learning.

[7] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*, 91-99.

[8] Ha, H., Im, S., Park, J., Jeon, H. G., & Kweon, I. S. (2016). High-quality depth from uncalibrated small motion clip. In *Proceedings of the IEEE conference on computer vision and pattern Recognition* (pp. 5413-5421).

[9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

[10] Memon, Q., & Khan, S. (2001). Camera calibration and three-dimensional world reconstruction of stereo-vision using neural networks. *International Journal of Systems Science*, *32*(9), 1155-1159.

[11] Ali, A. A., & Hussein, H. A. (2016, May). Distance estimation and vehicle position detection based on monocular camera. In *2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)* (pp. 1-4). IEEE.

[12] Bertoni, L., Kreiss, S., & Alahi, A. (2019). Monoloco: Monocular 3d pedestrian localization and uncertainty estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 6861-6871).

[13] Pedra, A. V. B. M., Mendonça, M., Finocchio, M. A. F., de Arruda, L. V. R., & Castanho, J. E. C. (2013). Camera calibration using detection and neural networks. *IFAC Proceedings Volumes*, *46*(7), 245-250.