Université de Montréal

Calibrage de caméra fisheye et estimation de la profondeur pour la navigation autonome

par

Pierre-André Brousseau

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des arts et des sciences
en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en informatique

août, 2019

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé :

Calibrage de caméra fisheye et estimation de la profondeur pour la navigation autonome

présenté par

Pierre-André Brousseau

a été évalué par un jury composé des personnes suivantes:

_____

Jean Meunier
(Professeur)

_____

Max Mignotte
(Professeur)

_____

Sébastien Roy
(Directeur)

Mémoire accepté le _____

RÉSUMÉ

Ce mémoire s'intéresse aux problématiques du calibrage de caméras grand angles et de l'estimation de la profondeur à partir d'une caméra unique, immobile ou en mouvement. Les travaux effectués se situent à l'intersection entre la vision 3D classique et les nouvelles méthodes par apprentissage profond dans le domaine de la navigation autonome. Ils visent à permettre la détection d'obstacles par un drone en mouvement muni d'une seule caméra à très grand angle de vue. D'abord, une nouvelle méthode de calibrage est proposée pour les caméras fisheyes à très grand angle de vue par calibrage planaire à correspondances denses obtenues par lumière structurée qui peuvent être modélisée par un ensemble de caméras génériques virtuelles centrales. Nous démontrons que cette approche permet de modéliser directement des caméras axiales, et validons sur des données synthétiques et réelles. Ensuite, une méthode est proposée pour estimer la profondeur à partir d'une seule image, à partir uniquement des indices de profondeurs forts, les jonctions en T. Nous démontrons que les méthodes par apprentissage profond sont susceptibles d'apprendre les biais de leurs ensembles de données et présentent des lacunes d'invariance. Finalement, nous proposons une méthode pour estimer la profondeur à partir d'une caméra en mouvement libre à 6 degrés de liberté. Ceci passe par le calibrage de la caméra fisheye sur le drone, l'odométrie visuelle et la résolution de la profondeur. Les méthodes proposées permettent la détection d'obstacle pour un drone.

Mots clés: Vision par ordinateur, Calibrage de Caméra, Fisheye, Caméra axiale, Estimation de la profondeur, Estimation du mouvement, Navigation autonome

ABSTRACT

This thesis focuses on the problems of calibrating wide-angle cameras and estimating depth from a single camera, stationary or in motion. The work carried out is at the intersection between traditional 3D vision and new deep learning methods in the field of autonomous navigation. They are designed to allow the detection of obstacles by a moving drone equipped with a single camera with a very wide field of view. First, a new calibration method is proposed for fisheye cameras with very large field of view by planar calibration with dense correspondences obtained by structured light that can be modelled by a set of central virtual generic cameras. We demonstrate that this approach allows direct modeling of axial cameras, and validate it on synthetic and real data. Then, a method is proposed to estimate the depth from a single image, using only the strong depth cues, the T-junctions. We demonstrate that deep learning methods are likely to learn from the biases of their data sets and have weaknesses to invariance. Finally, we propose a method to estimate the depth from a camera in free 6 DoF motion. This involves calibrating the fisheye camera on the drone, visual odometry and depth resolution. The proposed methods allow the detection of obstacles for a drone.

Keywords: Computer Vision, Camera Calibration, Fisheye, Axial Camera, Depth Estimation, Autonomous Navigation

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE D'ACRONYMES

ABS	Acrylonitrile butadiène styrène (Acrylonitrile Butadiene Styrene)
ADAS	Systèmes avancés d'aide à la conduite (Advanced Driver-Assistance Systems)
CART	Arbres de classification et de régression (Classification And Regression Trees)
CNC	Machine-outil à commande numérique (Computer Numerical Control)
CNN	Réseau de neurones convolutif (Convolutional Neural Network)
FCN	Réseau de neurones complètement convolutif (Fully Convolutional Neural network)
FoV	Angle de vue (Field of View)
GVC	Virtuelle Générique Centrale (Generic Virtual Central)
ILSVRC	Défi de reconnaissance visuelle à grande échelle ImageNet (ImageNet Large Scale Visual Recognition Challenge)
IMU	Unité de mesure Inertielle (Inertial Measurement Unit)
LCD	Affichage à cristaux liquides (Liquid Crystal Display)
LiDAR	Détection et estimation de la distance par la lumière (Light Detection And Ranging)
MRF	Champ aléatoire de Markov (Markov Random Field)
MSER	Régions extrêmes maximalement stables (Maximally Stable Extremal Regions)
PLA	Acide polylactique (PolyLactic Acid)
RCNN	Réseau de neurones convolutifs basés régions (Region-based Convolutional Neural Network)

RPN Réseau de proposition de région (Region Proposal Network)

SLAM Cartographie et localisation simultanées (Simultaneous Localization And Mapping)

SURF Caractéristiques robustes accélérées (Speeded Up Robust Features)

SVM Machine à vecteurs de support (Support-Vector Machine)

TOF Temps de vol (Time Of Flight)

UAV Véhicule aérien sans pilote (Unmanned Aerial Vehicule)

REMERCIEMENTS

Je tiens tout d'abord à remercier mon directeur de recherche, Sébastien Roy, sans qui aucun de ces travaux n'aurait été possible. Je veux souligner la part cruciale qu'il a eue dans le développement de mon sens de la recherche et de mon éthique de travail. Je lui dis merci pour toutes les belles opportunités dont j'ai pu profiter avec son laboratoire V3D.

Je tiens aussi à remercier mes collègues de travail du laboratoire. Un merci spécial à Chaima pour ces trois années de rire, de soirées de travail interminable et de projets farfelus. Son soutien dans ma recherche a été des plus importants. Je souligne aussi la présence d'Emir et de Nicolas qui ont donné vie à mon quotidien.

Je tiens finalement à remercier Martine Simard et Paul Brousseau, mes parents, à qui j'attribue la totalité de ma réussite. Ils m'ont mis en marche sur le chemin de la recherche et ont été avec moi à chaque pas en support et en conseils.

Ce mémoire est dédié à ma maman pour qui les phrases d'éloges seront toujours à court de décrire sa force, sa gentillesse et son humilité.

INTRODUCTION

Ce mémoire s'intéresse à deux problématiques du domaine de la vision tridimensionnelle sous le thème commun de la navigation autonome soient le calibrage de caméras grands-angles et l'estimation de la profondeur à partir d'une seule caméra immobile ou en mouvement. Les présents travaux visent à permettre la détection d'obstacles par un drone en mouvement muni d'une seule caméra très grand-angle de vue. Les travaux effectués se situent à l'intersection entre la vision 3D classique et les nouvelles méthodes par apprentissage profond.

En premier lieu, nous proposons une nouvelle méthode de calibrage pour les caméras ayant un très grand angle de vue. Cette méthode s'appuie en son coeur sur le calibrage planaire à correspondances denses obtenues par lumière structurée, mais en faisant le calibrage du point de vue des plans eux-mêmes. En utilisant les grilles planaires elles-mêmes plutôt que le plan d'image déformé, nous pouvons construire une caméra virtuelle générique centrale (GVC) qui est perspective rectilinéaire. Notre méthode propose une sélection de plusieurs caméras GVC qui peuvent être alignées trivialement et ainsi couvrir n'importe quel champ de vision. Nous démontrons que cette approche permet de modéliser directement des caméras axiales, en supposant que le centre de distorsion est situé sur l'axe de la caméra. La validation expérimentale est fournie sur des caméras fisheye synthétiques et réelles avec un angle de vue allant jusqu'à 280°. À notre connaissance, c'est l'une des seules méthodes pratiques de calibrage de caméras axiales.

En deuxième lieu, nous proposons une nouvelle méthode pour estimer la profondeur à partir d'une unique image. Cette méthode utilise uniquement des indices de profondeur forts soient les jonctions-T ce qui rend la méthode invariante aux divers biais

engendrés par les ensembles de données. Les relations de profondeur sont obtenues grâce à des réseaux de neurones convolutifs sur des imagettes permettant à la fois la robustesse à la variabilité des diverses images et l'invariance aux informations globales qui sont induites par l'ensemble de données. Les relations de profondeur déduites des jonctions-T sont propagées de façon globale par un graphique de flot maximal et sa solution par coupe minimale. Une démonstration du biais chez les méthodes purement par apprentissage profond est fournie en plus d'une validation expérimentale sur des données synthétiques et réelles. Malgré les performances limitées de la méthode, elle se veut comme une étape vers la compréhension des modèles par apprentissage profond et l'exploration de l'application de ces méthodes dans un contexte de la vision tridimensionnelle.

En troisième lieu, nous explorons la détection d'obstacle et l'estimation de la profondeur dans un contexte de drones en mouvement libre. La méthode se base sur de l'apprentissage supervisé dans un secteur où les ensembles de données sont limités voire absent. Une plateforme delta qui permet la translation et la rotation de la caméra sur cinq degrés de liberté est fabriquée afin de permettre l'acquisition d'un ensemble de données fiable. Par réseau de neurones convolutifs, la trajectoire de la caméra est résolue permettant de résoudre directement la profondeur dans la scène pour une caméra perspective rectilinéaire. Ceci fait un rappel à l'algorithme de calibrage qui permet entre autres de convertir la caméra fisheye de drone en caméra perspective. Un exemple de carte de profondeur est montré permettant de faire la distinction entre le proche et le loin et de détecter les obstacles.

Plusieurs perspectives de recherches futures restent à explorer dans le domaine de la navigation autonome. Ce domaine s'inscrit comme un domaine d'avenir qui verra une explosion dans les prochaines années. En facilitant le calibrage, il est envisageable que l'estimation de la profondeur se fasse de plus en plus à travers des systèmes de caméras fisheye notamment sur les voitures où celles-ci sont déjà aujourd'hui posées.

Chapitre 1

CAMÉRAS, MODÈLES ET CALIBRAGE

Dans le domaine de la vision par ordinateur, l'outil d'acquisition des données est la caméra. Celle-ci est un instrument qui génère des images en deux dimensions à travers une étape de projection appliquée au monde en trois dimensions. Comprendre la projection nécessite de comprendre les différents types de caméras, les différents modèles proposés dans la littérature et comment les différents paramètres de ces dit modèles sont retrouvés. Les dernières années ont vu croître l'utilisation des caméras pour des systèmes de décision automatisés. Des systèmes de reconnaissances faciales aux intersections aux systèmes de navigation autonome sur les automobiles, l'importance de maîtriser les systèmes de caméras devient de plus en plus cruciale afin d'éviter les abus et les erreurs. L'abondance des caméras dans les mains de tous au niveau des téléphones mobiles et de l'informatique embarquée permet à tout un chacun de mettre en place son propre système de vision.

Le présent chapitre se veut de définir le processus d'utiliser une caméra comme un outil de mesure. D'abord, il est nécessaire d'identifier la caméra en ce qui a trait à sa fabrication et ses propriétés optiques. Tout simplement, les divers systèmes de lentilles entraînent différents comportements de la lumière et indiquent directement les caractéristiques de l'image résultante. De plus, le type de caméra limite quelles applications peuvent être réalisées. Ensuite, les différents modèles de caméras sont présentés ainsi que les paramètres qui leur sont propres. Il est essentiel de différencier entre une caméra et un modèle de caméra et comprendre les hypothèses simplificatrices impliquées. Finalement, pour un modèle choisi, l'étape du calibrage permet de trouver une estimation aux paramètres grâce à des contraintes sur le monde. Ce processus d'identifier la caméra, de lui choisir un modèle et d'en estimer les paramètres est appelé

le calibrage de caméra.

1.1 La caméra

Une caméra est un instrument optique qui permet la capture d'images. Ces images originalement captées sur des films sont aujourd'hui enregistrées en grande majorité dans des dispositifs digitaux. On définit pour le présent mémoire le terme caméra comme le système de lentilles qui dirigent la lumière de la scène vers le senseur qui réalise la capture de l'intensité lumineuse. De cette façon on définit l'image comme le résultat de cette capture lumineuse qui est enregistré comme une grille de pixels. Pour des fins de simplification, on s'intéresse principalement aux caméras qui capturent le spectre visible de la lumière bien qu'il soit possible de capturer bien d'autres portions du spectre électromagnétique.

Le concept simplifié de la caméra est un système de lentilles convergentes qui guide la lumière vers la surface de captation. Quelques concepts additionnels sont nécessaires toutefois à la capture de l'image. Notamment, l'obturateur qui contrôle la quantité totale de lumière qui est utilisée pour la mesure de l'intensité lumineuse, ce concept qui est appelé le temps d'exposition. L'obturateur se ferme d'ailleurs habituellement de haut en bas dans l'image de façon non instantanée. Ceci mène à des effets de rolling shutter dans des scènes d'objets en grands mouvements. Il y a de plus la notion de plan au focus soit la distance où les éléments de la scène sont à leur pleine résolution.

1.1.1 Systèmes de lentilles

Une lentille de caméra est une lentille de verre ou de plastique ou un assemblage optique de lentilles qui peuvent rassembler la lumière de l'extérieur et la guider sur le capteur. Bien entendu, cette lumière qui se déplace en ligne droite est déviée pour être acheminée et ceci résulte en des artéfacts dans l'image résultante. On peut noter entre autres une présence de courbures dans les lignes supposées droites, une perte du par-

allélisme ou une perte de la perpendicularité entre les lignes. La présence ou l'absence de ces aberrations varie selon l'application et le système de lentilles utilisé.

Selon le système de lentilles utilisé, on obtient des images très différentes en sortie. Ainsi, ces caméras doivent être décrites par des modèles de caméras qui leur correspondent le plus fidèlement possible. Ainsi, il essentiel de comprendre les différents systèmes optiques de caméras ainsi que les différents modèles. Les modèles peuvent varier de très précis s'appliquant qu'à une unique caméra à des modèles très généraux qui peuvent couvrir un groupe de caméras. Le modèle de caméra le plus connu est celui de la caméra sténopé. On pense ici à la "caméra obscura" qui indique une image formée du phénomène naturel lorsque la lumière d'une scène est projetée sur un écran après avoir traversé une surface opaque ayant une ouverture en un point. L'image résultante est une image renversée gauche-droite et haut-bas sur l'écran.

1.1.2 Caméra perspective rectilinéaire

Le plus simple type de caméra est donc une caméra perspective rectilinéaire. En pratique, les caméras perspectives rectilinéaires d'aujourd'hui sont beaucoup plus complexes qu'un simple trou dans une surface opaque. Plutôt, ce sont des systèmes de lentilles qui font apparaître les caractéristiques droites dans le monde comme droites dans l'image. Il n'en demeure pas moins que la lumière doit passer par un petit trou appelé l'ouverture et ceci mène à des limitations au point de vue de l'image. Cette ouverture doit être très petite afin de rendre l'image nette. La taille des pixels ou unités d'images est directement liée à la taille de l'ouverture. Plus celle-ci est petite plus l'image est constituée d'un grand nombre de pixels. En effet, on s'imagine plusieurs rayons de lumière qui quittent un point de l'objet dans le monde et traversent l'ouverture de caméra. Il est espéré que tous ces rayons tombent sur le capteur en un même point et diminuer le diamètre de l'ouverture va dans le sens de ce fait. À noter qu'il existe une limite sur la résolution maximale de l'image qui est proportionnelle à

la longueur d'onde de la lumière utilisée. Ceci correspond de la même façon à la limite sur le diamètre minimal de l'ouverture d'une caméra. Inversement, l'ouverture limite la quantité de lumière perçue par la caméra. Plus l'ouverture est grande, plus la quantité de lumière dans l'image est grande. De ce fait, on obtient deux caractéristiques qui s'opposent soit l'intensité lumineuse qui décroît quand la résolution monte.

1.1.3 Caméra fisheye

Une limite apparente est perçue chez les caméras perspectives rectilinéaires soit de faire passer en ligne droite la lumière à travers un seul point. En effet, il est simple de voir que lorsque l'objet est suffisamment large, il engendre un angle très grand entre lui et l'ouverture de la caméra. Imaginons un plan infini devant la caméra. L'angle formé entre les extrémités du plan infini et l'ouverture est de 180°. Ce concept d'angle limite qui peut être perçu en un moment par la caméra est appelé angle de vue. Pour une caméra perspective rectilinéaire théorique, l'angle de vue maximal est inférieur à 180°. En pratique, cet angle est beaucoup plus bas et dans les alentours de 90°. Pour des angles de vue plus grands, il existe des systèmes de lentilles fisheye qui peuvent aller à de très grands angles de vue. Toutefois, plus l'angle de vue d'un fisheye est grand plus la distorsion est elle aussi grande. La distorsion mentionnée ici est un type d'aberration qui transforme les caractéristiques droites dans le monde en caractéristiques courbes. De plus, il existe plusieurs types de fisheye qui vont transformer différemment le monde selon leur courbe propre de projection. On note ici les fisheye équisolide, équiangulaire et stéréographique comme les types de fisheye habituellement utilisés [89]. Le chapitre 2 présente une méthode de calibrage pour des caméras fisheye allant jusqu'à 280° d'angle de vue. Des images reccueillies par une telle caméra sont montrées à la figure 1.1.

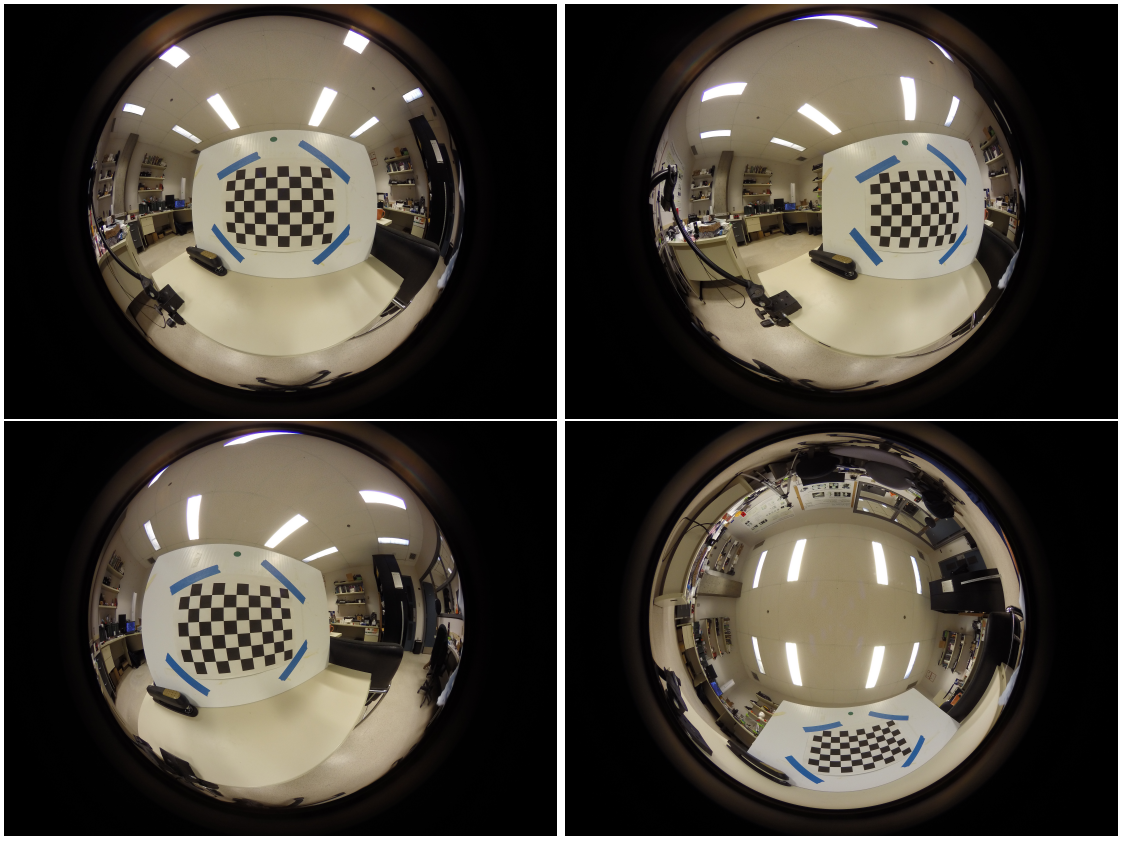


Figure 1.1: Plusieurs photos d'un damier capturées par GoPro HERO 4 avec une lentille fisheye Entaniya 280° montrant la distorsion radiale aux extrémités de la lentille.

1.1.4 Caméra catadioptrique

Dans un ordre d'idées, il est possible de réaliser une caméra de forme tout autre que les caméras vues précédemment. En effet, l'idée d'un système de lentilles est que toutes les lentilles sont ensemble en une succession. Or, il est possible d'ajouter un ou plusieurs miroirs complètement découplés de la caméra pour obtenir une caméra catadioptrique. En utilisant des miroirs externes, il est facile de générer une multitude de caméras aux propriétés différentes. D'abord, il existe les caméras que l'on appelle "single-mirror central catadioptric camera" ou caméra au miroir unique central. Le miroir n'a aucunement besoin d'être une surface plane et bien souvent il peut être un miroir hyperbolique ou

sphérique telle une boule réfléchissante chromée. De façon plus précise, ces caméras sont définies comme ayant un miroir provenant de la révolution d'une conique autour de son propre axe de symétrie. De plus, l'axe de symétrie de la conique et l'axe de la caméra doivent coïncider. Lorsque ceci n'est pas le cas, la caméra devient non centrale et celle-ci nécessite des modèles plus complexes pour la calibrer de façon précise [98]. Comme idée de caméra du présent type, on note les travaux de Finsterwalder qui utilisait trois miroirs perpendiculaires de façon à photographier huit faces simultanément d'un même petit objet. Ceci lui permettait donc de faire une reconstruction stéréo totale de l'objet à partir d'une seule image [18].

Il existe des systèmes de lentilles appelés des lentilles catadioptriques. Ces systèmes de lentilles sont placés de telles sortes à faire rebondir la lumière et ainsi réduire la longueur physique de l'objectif. Ceci est principalement utilisé pour les objectifs aux longues focales puisque l'arrangement des miroirs permet de multiplier la longueur focale. Ceci résulte en un objectif compact à la longue focale qui est moins sensible aux erreurs d'alignement.

1.1.5 Caméra pushbroom

Un autre type de caméra qui peut être à l'étude est la caméra dite pushbroom. Elle est constituée d'une caméra 1D qui est mise en mouvement en rotation ou en translation. L'application initiale de ce type de caméra est d'effectuer des panoramas. Une acquisition d'une ligne de caméra est effectuée pour chaque unité en rotation permettant d'accumuler un panorama entier en haute résolution. L'idée de ce type de caméra est de faire une concaténation d'un grand nombre de captures de faible dimension. La capture d'images par vol d'avion est mise en commun pour générer des cartes. Ce système de capture peut être considéré dans son ensemble comme une caméra pushbroom.

1.2 Modèles de caméras

Comme mentionné plus haut, il existe plusieurs types de caméras chacun ayant son système de lentilles ou de miroirs rendant la représentation de toutes les caméras par un seul modèle très difficile. Dans l'objectif d'effectuer un calibrage d'une caméra, il est essentiel de choisir un modèle. Il importe donc de comprendre comment fonctionne le système d'acquisition, la caméra, afin de déterminer quels facteurs sont importants et lesquels peuvent être ignorés. Il est dit que "chaque hypothèse simplificatrice diminue le réalisme du modèle, mais simplifie le traitement mathématique." Il est désirable d'avoir un modèle qui est simple afin d'en assurer la résolution, mais qui est fonctionnel pour la caméra. Les modèles de caméras qui sont présentés dans cette section s'appliquent principalement aux caméras perspectives rectilinéaires et aux caméras fisheye.

Du point de vue des modèles, considérons la caméra comme une opération mathématique de projection des points 3D dans le monde vers les pixels de la caméra et son opération inverse de déprojection des pixels vers des rayons dans le monde. La caméra peut ainsi être décrite de façon la plus générale possible comme un ensemble de rayons ayant chacun une position et une direction chacune liée à un pixel dans l'image [97]. Ce concept générique peut directement décrire la caméra. Toutefois l'absence de contraintes additionnelles rend le calibrage d'un tel modèle très difficile. En ce sens, selon la caméra des hypothèses simplificatrices sont émises menant à l'apparition d'une multitude de modèles de caméras.

Ces modèles peuvent être séparés selon divers critères. Le premier critère sélectionné est la distribution spatiale des rayons de la caméra. Puisqu'une grande partie des modèles présente un point à travers lequel tous les rayons intersectent, ces modèles sont appelés des modèles centraux et le point d'intersection est appelé le centre optique. Pour ceux-ci, seules la direction des rayons de caméra et la position du centre optique doivent être résolues. Inversement, les modèles non-centraux ne possèdent pas de centre optique unique [98].

Le second critère discriminant entre les modèles fait appel au degré de généralité des paramètres choisis. En effet, les modèles qui présentent un petit nombre de paramètres sont appelés globaux puisque ces quelques paramètres influencent de façon générale tous les rayons de la caméra. Un modèle est dit global si les paramètres sont valides sur la totalité de l'angle de vue. De l'autre côté se trouvent les modèles locaux qui présentent un grand nombre de paramètres. Dans ce cas-ci, chaque petit groupe de paramètres se rapporte plutôt pixel à pixel et chacun est indépendant des voisins.

1.2.1 Projection et déprojection

Un concept qui est nécessaire de décrire est la projection et la déprojection d'un point ou d'un pixel dans le monde. Il est décrit par Sturm [98] que malgré qu'ils soient simplement l'inverse l'un de l'autre, l'un des deux est toujours beaucoup plus difficile à définir. Il continue son exemple en parlant du modèle perspectif avec distorsion polynomiale. En utilisant qu'un seul coefficient de distorsion, le polynôme devient cubique rendant son inverse lourd [98]. La projection est l'étape de prendre un point 3D dans le monde et de trouver la position 2D résultante du point dans l'image. La déprojection est l'étape inverse de prendre un point dans l'image et de trouver le rayon de caméra dans le monde. Lors d'une triangulation, on peut intersecter deux rayons pour retrouver la position d'un point 3D à partir de positions 2D dans deux images.

1.2.2 Modèle sténopé

Le modèle sténopé est un modèle central où tous les rayons de caméra intersectent un seul point unique, le centre optique et où il existe une relation linéaire entre la position 2D d'un point dans l'image et la direction du rayon de caméra qui lui est associée. Cette relation s'exprime par la matrice de paramètres internes suivante:

$$\begin{pmatrix} f_u & s & x_0 \\ 0 & f_v & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.1}$$

Où f_u et f_v sont les distances focales horizontales et verticales en pixels et s est le terme qui décrit le caractère oblique des pixels, par exemple des pixels non rectangulaires. x_0 et y_0 sont les coordonnées sur le plan image du centre optique. Le modèle est davantage simplifié lorsque les pixels sont carrés, $f_u = f_v$ et $s = 1$.

1.2.3 Modèle affine

Le modèle de caméra affine est un modèle où le centre optique est présumé à l'infini. Dans ce cas, on retrouve la caméra orthographique. Ce modèle sert dans les cas où la longueur est très grande ou si la scène imagée présente une faible variation dans la profondeur des points. Il faut noter que l'étape de déprojection est simplifiée puisque la direction des rayons de caméras est la même et est déjà résolue, et ce pour tous les pixels.

1.2.4 Modèle perspectif avec distorsion polynomiale

Tel que décrit précédemment, une projection rectilinéaire est une projection où les lignes droites dans le monde sont préservées droites dans l'image. La distorsion est une forme d'artefact optique qui vient introduire de la courbure dans les lignes droites. Par symétrie du système de lentilles, cette distorsion est habituellement symétrique radialement autour du centre optique et tend à être plus prononcée plus un point s'éloigne du centre optique. On note deux types principaux de distorsion radiale soit les distorsions "barrel" et "pincushion" où les lignes droites sont courbées vers l'extérieur de l'image ou vers l'intérieur respectivement. Il existe d'autres types de distorsion qui peuvent être à leur tour pris ou non en compte. Une description approfondie de la distorsion est présentée en [5, 22, 92]. Lorsque l'on considère que le centre optique coïncide avec le

centre de distorsion, le modèle qui combine à la fois distorsion radiale et tangentielle est:

$$
\begin{aligned}
\bar{x}_d &= x_d - x_0 \\
\bar{y}_d &= y_d - y_0 \\
r_d^2 &= \bar{x}_d^2 + \bar{y}_d^2 \\
x_u &= x_d + p_1(r_d^2 + 2\bar{x}_d^2) + 2p_2\bar{x}_d\bar{y}_d + \bar{x}_d \sum_{i=1}^{n} k_i r_d^{2i} \\
y_u &= y_d + 2p_1\bar{x}_d\bar{y}_d + p_2(r_d^2 + 2\bar{x}_d^2) + \bar{y}_d \sum_{i=1}^{n} k_i r_d^{2i} \\
x_u &= x_d + D_{tx} + \bar{x}_d \sum_{i=1}^{n} k_i r_d^{2i} \\
y_u &= y_d + D_{ty} + \bar{y}_d \sum_{i=1}^{n} k_i r_d^{2i}
\end{aligned}
\tag{1.2}
$$

Pour un point distortionné dans l'image (x_d, y_d) et son centre de distorsion (x_0, y_0), sa postion sans distorsion est le point (x_u, y_u). Ainsi, r_d^2 est la distance entre un point distorsionné et son centre de distorsion, n est le nombre de coefficients de distorsion et D_{tx} et D_{ty} sont les distorsions tangentielles dans la direction x et y respectivement. Les coefficients k_i sont à trouver lors de la résolution du modèle et habituellement la distorsion tangentielle est présumée nulle. Une fois que les coordonnées des points images sont corrigées, on retrouve un modèle sténopé avec une matrice de paramètre interne classique.

1.2.5 Modèle fisheye classique

Le modèle de caméra précédent introduit dans ses termes les caractéristiques de distorsions. Toutefois, les modèles de distorsion classique ne sont pas appropriés pour les caméras aux angles de vus très grands notamment les caméras fisheyes. En reprenant l'équation précédente dans un contexte de caméra perspective rectilinéaire, on obtient

l'équation suivante.

$$\begin{aligned}\theta &= atan(\frac{r_u}{f}) \\ &= atan(\frac{r_d(1+\sum_{i=1}^{n} k_i r^{2i})}{f})\end{aligned} \tag{1.3}$$

On voit que pour obtenir un angle de 90°, le rayon r_u doit être infini. L'ajout du terme de distorsion ne change pas cette limite des caméras perspectives rectilinéaires, elle demeure limitée à un angle de vue inférieur à 180°. Dans ce sens, plusieurs types de fisheye sont vendus par les fabricants de lentilles pour convenir à des modèles facilement évaluables [19, 60] . On parle ici des modèles stéréographique, équidistant et équisolide. En pratique, les fabricants tentent de s'approcher du modèle équidistant aussi nommé équiangulaire.

$$\begin{aligned}r_d &= f\,tan(\frac{\theta}{2}) \\ r_d &= f\,\theta \\ r_d &= f\,sin(\frac{\theta}{2})\end{aligned} \tag{1.4}$$

1.2.6 Modèle polynomial

Le modèle polynomial se veut une façon différente de modéliser la distorsion radiale avec davantage de flexibilité. Ce modèle permet donc de s'ajuster à toute distorsion qui est radialement symétrique autour du centre optique, et ce avec un polynôme. Toutefois, à l'inverse de la méthode précédente, ce n'est pas la relation entre les positions d'un même point avec et sans distorsion qui est pris en contre. Plutôt, la relation est directement celle entre le rayon sur la plan image et l'angle d'incidence du rayon de caméra par rapport à l'axe optique [98]. Il est important de mentionner de nouveau que ce modèle est central et que donc tous les rayons de caméras intersectent à un même point.

$$r_d = \sum_{i=1}^{n} c_i \theta^i \tag{1.5}$$

Ce modèle est suggéré par Kannala et Brandt [47] et ceux-ci ajoutent aussi des termes supplémentaires correspondant à la distorsion tangentielle. Les auteurs présentent d'ailleurs la fonction de déprojection non triviale qui doit en pratique être approximée. Finalement, dans la librairie de code OpenCv, c'est ce modèle qui est utilisé lors de la modélisation d'une caméra de type fisheye.

1.2.7 Modèle central discret par rayon

Le modèle central discret par rayon diffère grandement des modèles précédents puisque celui-ci ne limite plus la caméra à avoir des caractéristiques de symétrie radiale autour du point principal. Très simplement, ce modèle permet à chaque rayon de caméra d'être indépendant des autres rayons de caméra. De cette façon ce modèle possède deux ou trois paramètres par rayon de caméra, soit une description de la direction du rayon en deux angles ou trois coordonnées dans un système euclidien. Il ne demeure donc qu'une seule hypothèse simplificatrice sur la description générale de ce qu'est une caméra: tous les rayons de caméra intersectent un centre optique unique.

Dans leur publication, Gremban et al. [31] indiquent qu'une généralisation de leur modèle de caméra serait possible. Or, selon eux, "Une table de correspondance des données de calibrage pour chaque pixel aurait un coût prohibitif". Pour des questions de limites de mémoire ceux-ci n'ont pas continué d'explorer davantage cette avenue. Toutefois, les techniques par lumière structurée et l'amélioration des technologies informatiques ont permis des travaux qui réalisent exactement ce qui est dit. En faisant directement la correspondance entre les points 3Ds dans le monde, on retrouve directement les rayons dans le monde [32, 15, 83, 101].

1.2.8 Modèle non-centraux

Les modèles les plus généraux sont les modèles non-centraux. Simplement, ceux-ci ne font aucune hypothèse simplificatrice sur le modèle général de la caméra. La caméra est un ensemble de pixels avec pour chacun un rayon de caméra dans une direction et position quelconque [97, 33]. Ces modèles se voient très difficiles à calibrer puisqu'il requiert six paramètres euclidiens par pixel, trois pour l'origine du rayon de caméra et trois pour la direction.

1.3 Calibrage

Pour une caméra donnée, l'étape du calibrage combine habituellement l'étape de la sélection du modèle et l'estimation des paramètres du modèle. Toutefois, ces étapes sont en fait distinctes et la présente section ne réfère qu'aux différentes méthodes de recueillir de l'information sur une caméra dans un environnement contrôlé, et ce dans l'objectif d'estimer les paramètres d'un modèle choisi pour cette caméra. Ce calibrage s'effectue avec des images provenant de la caméra et les relations établies entre les positions des points projetés dans les images.

1.3.1 Calibrage avec objet 3D connu

Les termes "objets 3D connus" désignent un objet aux dimensions réelles connues. De cette façon, la position 3D dans le monde de certains des points sur cet objet est connue. Pensons simplement à un cube aux dimensions connues. La position de ses coins est du même fait connue. Des images prises par une caméra de ce cube connu présenteraient en elle des points dont les coordonnées 2Ds dans l'image ainsi que les coordonnées 3Ds sont connues. Pour un modèle simple comme le modèle perspectif rectilinéaire, il est possible de construire un système d'équations linéaires homogènes et de le résoudre. En pratique, les objets doivent être plus complexes qu'un cube afin de présenter suffisamment de points saillants pour obtenir un maximum d'informations par image unique.

Chaque image additionnelle doit avoir un terme de rotation et translation ajouté à la position 3D connue des points qui doit être séparément résolu. Dans un cas idéal, l'objet 3D connu photographié doit être suffisamment complexe pour proposer suffisamment de correspondance 2D-3D pour résoudre le modèle en une seule image [67, 93, 102].

1.3.2 Calibrage avec objet planaire

Le calibrage avec objet planaire aussi nommé calibrage planaire et calibrage par plans, est la méthode du calibrage la plus connue pour dû principalement à la facilité de se procurer le matériel de calibrage soit un plan. En pratique, ces plans de calibrage peuvent être des murs, des planches de cartons, une façade de bâtiment ou même le sol. La seconde caractéristique requise est que sur le plan de calibrage se trouvent des points ou des marques aux positions connues. Puisque la position de ces points est une position sur un plan, on appelle ces positions des positions 2D sur un plan. Le damier est l'objet de calibrage planaire classique puisque la distance entre les points est connue, mais des marqueurs placés sur un mur sont tout aussi valides. Cette méthode de calibrage est attribuée à Zhang qui l'utilise pour calibrer le modèle sténopé et même un modèle avec distorsion radiale [113]. Puisque les correspondances obtenues sont 2D-2D, plusieurs vues du même objet planaire sont nécessaires. Toutefois, puisque l'objet de calibrage est un plan, il est possible de relier les différentes images du plan entre elles par des homographies dans le cas où le plan image est bel et bien un plan et que le modèle de caméra est central. Ces restrictions sont d'ailleurs les raisons pourquoi les modèles non-centraux et non globaux sont moins utilisés.

1.3.3 Calibrage par rotation connue

Le calibrage par rotation connue est une méthode de calibrage qui inverse la méthodologie des méthodes précédentes. Plutôt que d'avoir des contraintes sur la position de certains points dans le monde et leur correspondance dans l'image, nous avons des

contraintes sur le mouvement qui sépare deux images. Il est important de savoir que ce mouvement doit être une rotation pure de sorte à ne pas introduire de notion de parallaxe entre les points des images qui fausseraient les mesures de déplacement des points. Stevenson et Fleck ont réalisé cette approche avec une caméra et une source lumineuse ponctuelle. En tournant la caméra autour de son centre optique, il est possible d'échantillonner la position de la source lumineuse ponctuelle et donc d'échantillonner la fonction de distorsion radiale selon l'angle d'incidence [94]. Cette méthode peut être généralisé pour n sources de lumières ponctuelles où n est le nombre de pixels dans l'image. Cette méthode est en pratique très peu utilisée puisqu'elle nécessite de connaître la position du centre optique préalablement au calibrage. Ceci est souvent antithétique puisque l'étape de calibrage est habituellement utilisée pour retrouver la position du centre optique.

1.3.4 Calibrage par rayons

Les méthodes de calibrage par rayons sont aussi souvent appelées calibrage basé raxel ou calibrage basé rayon. Cette méthode vise à retrouver deux positions 3D d'un point 2D dans l'image de sorte à pouvoir tracer un rayon traversant les deux points en 3D. Cette méthode de calibrage utilise aussi bien des objets planaires connus et le mouvement qui sépare les images lui aussi est connu [31, 32, 33]. En utilisant plusieurs images, il est possible de retrouver des correspondances entre plusieurs points 2D sur un plan dans le monde et puisque le mouvement est connu, les points deviennent 3D. Il est possible de tracer une ligne les traversant, et ce pour chaque pixel de l'image. Ceci correspond en une combinaison de la méthode de calibrage avec objet planaire et de calibrage par rotation connue. Cette approche de calibrage est une version dense de l'approche de calibrage surnommée "deux-plans" [58].

Toutefois, cette approche pour le calibrage par rayons présente la faiblesse de requérir des mouvements connus autour d'un centre optique connu. Heureusement,

Sturm et Ramalingam ont pu généraliser le calibrage pour les situations où le mouvement de la caméra est inconnu [97]. Ceci s'effectue en ajoutant une troisième image du même plan cette méthode devenant en quelque sorte "trois plans". Les mouvements effectués sont ainsi retrouvés à travers la résolution d'un tenseur trifocal et, avec le mouvement connu, il est possible de retrouver le rayon de caméra associé aux différents pixels.

Dunne et al. ont poursuivi les travaux débutés en effectuant des scans en lumière structurée d'écrans afin d'avoir des plans de calibrage parfaitement denses. De plus, en utilisant la carte de correspondance il est possible de déformer les images de façon à obtenir des images capturées par une caméra perspective rectilinéaire. De ce fait, ces images peuvent être simplement calibrées par un calibrage planaire simple ou une autre méthode au choix [15].

1.4 Systèmes multicaméras et stéréo

Les caméras, les modèles et les méthodes de calibrage présentés jusqu'ici s'appliquent à des caméras uniques. Il est toutefois possible d'imaginer des systèmes de caméras comportant plusieurs caméras. La géométrie épipolaire est une relation qui relie deux images d'une scène 3D du point de vue de deux caméras et permet d'obtenir des contraintes entre les images. Cette relation s'exprime par une matrice essentielle pour des caméras perspectives rectilinéaires préalablement calibrées. Dans le cas de caméras perspectives rectilinéaires non calibrées, cette relation s'exprime sous la forme d'une matrice fondamentale. Dans le cas de caméras avec modèle perspectif avec distorsion polynomiale, Zhang propose une méthode d'optimisation non-linéaire pour retrouver les coefficients de distorsion et la matrice fondamentale [112]. Pour le cas par modèle polynomial, Claus et Fitzgibbon décrivent la méthode pour retrouver la matrice fondamentale [12].

Chapitre 2

(ARTICLE) CALIBRAGE

Cet article [4] a été publié tel qu'indiqué dans la bibliographie :

> Pierre-André Brousseau et Sébastien Roy. Calibration of axial fisheye cameras through generic virtual central models. Dans Proceedings of the IEEE International Conference on Computer Vision. Institute of Electrical and Electronics Engineers(IEEE), 2019.

Dans cet article, nous proposons une nouvelle méthode de calibrage pour les caméras fisheyes ayant un très grand angle de vue par calibrage planaire à correspondances denses obtenues par lumière structurée. En utilisant les grilles planaires elles-mêmes plutôt que le plan d'image déformé, nous pouvons construire une caméra virtuelle générique centrale (GVC) qui est perspective rectilinéaire. Notre méthode propose une sélection de plusieurs caméras GVC qui peuvent être alignées trivialement et ainsi couvrir n'importe quel champ de vision. Nous validons cette approche sur des données réelles et synthétiques avec un angle de vue allant jusqu'à 280°. À notre connaissance, c'est l'une des seules méthodes pratiques de calibrage de caméras axiales.

L'article est présenté dans sa version originale.

Abstract

Fisheye cameras are notoriously hard to calibrate using traditional plane-based methods. This paper proposes a new calibration method for large field of view cameras. Similarly to planar calibration, it relies on multiple images of a planar calibration grid with dense correspondences, typically obtained using structured light. By relying on the grids themselves instead of the distorted image plane, we can build a rectilinear Generic Virtual Central (GVC) camera. Instead of relying on a single GVC camera, our method proposes a selection of multiple GVC cameras which can cover any field of view and be trivially aligned to provide a very accurate generic central model. We demonstrate that this approach can directly model axial cameras, assuming the distortion center is located on the camera axis. Experimental validation is provided on both synthetic and real fisheye cameras featuring up to a 280° field of view. To our knowledge, this is one of the only practical methods to calibrate axial cameras.

2.1 Introduction

Fisheye lens calibration is a type of calibration which is becoming very common since the emergence of low cost high quality fisheye lenses for applications such as immersive imaging as well as industrial and automotive applications. These lenses can feature very large fields of view (FoV). Modern fisheyes can see as much as 280° (see Fig. 2.7), with large amount of radial distortion and significant axial displacement of the optical center, essentially making them non-single viewpoint, which do not comply with standard lens models [44, 96]. Simple OpenCv tools fail completely at calibrating these kinds of cameras and current methods are either planar and limited to 180° or non-planar which makes them impractical. This paper proposes a calibration method adapted to these kinds of lenses, which are described as generic axial cameras. We tackle a slightly more constrained version of the axial model, which we identify as *Generic Quasi-Central* camera.

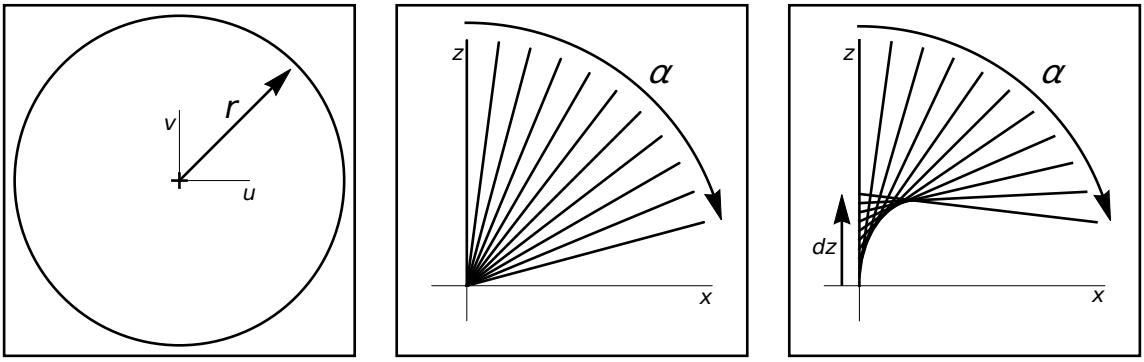


Figure 2.1: Central and Axial models. In the fisheye image (left), points at increased radius r correspond to increased ray angles α in a central camera (middle) as well as increased center displacement dz in a quasi-central camera (right).

Similarly to traditional planar calibration, the new method makes use of multiple dense correspondences of planar grids. It is known that a grid can be used as a *virtual image* to build a *Generic Virtual Central* cameras, which is free of distortion [15]. We propose to build multiple such virtual cameras, then calibrate and align them to provide an accurate solution to the full central camera model.

In order to recover the axial displacement of the optical center, an approach is presented for the generalization of the generic central model into the quasi-central model. This approach also relies on multiple GVC models, confirming their high accuracy. The paper is organized as follows. In Section 2.2, previous work and generic camera models are presented. The Generic Quasi-Central camera model is presented in Section 2.3. The calibration method for multiple Generic Virtual Central cameras is presented in Section 2.4, followed by experimental validation and results in Section 2.5.

2.2 Previous Work

There has been a lot of work on the calibration of fisheye lenses [65, 98]. Most of this work relies on global single viewpoint models. In this paper, the focus is on generic and

non-single viewpoint models.

Camera models can be divided in various ways. One of such way is to classify models according to how broad the impact of a set of parameters is in the image. The broader the impact the more global a model is. In this way, models can be classified as global, local and discrete classes. Discrete models are models where each individual pixel of the image requires its individual parameters. One such model is the *ray-based* model where it is required to explicitly represent each ray of a camera [98]. In this way, the generic imaging model as defined in [69, 97] is a discrete ray-based model where the camera captures images defined as a set of individual pixels each associated to a single ray of light in the 3D world. This unconstrained set of projection rays, each with its own direction and position in 3D, and associated image pixels, constitute the camera model and the expected result for the calibration algorithm [69, 95]. One possible calibration method is to solve 3D points along projection rays using 3D calibration objects. In the case of unknown pose of the calibration object, it is possible with 3 views of the object to solve a system of linear equations for the unknown poses and the projection rays in the case of a generic non-central camera. This system relies only on the collinearity constraints of points between the three views [69, 97]. The term non-central refers to the fact that the projection rays are unconstrained. Depending on the spatial distribution of the projection rays, a hierarchy of camera models emerges. Three of them are: central cameras where all projection rays intersect a single point (the optical center), axial cameras where all projection rays intersect a single line (the camera axis), and non-central cameras where the projection are entirely unconstrained [69, 95].

2.2.1 Generic Central Cameras

In the case of generic central cameras, calibration using planar calibration grids was demonstrated to work in practical cases [2, 68]. Using three calibration grids that are dense, the optical center as well as the pose of all three grids are solved. A 183°

field of view fisheye camera was successfully calibrated using a total of 12 calibration grids to cover the whole field of view [68]. It is shown that a minimum of four points each matching three planes are required to solve plane poses. Furthermore, the linear equation system relies on collinearity of matching projection rays as well as cheirality (solved 3d points are located on the same side of the optical center) [68]. Our approach requires two planes per match, its linear system is more straightforward, and can generalize to axial models.

An example of image undistorsion is provided in [2] by intersecting a single plane with the previously computed projection rays. With a lens of only 60° FoV, the camera is probably very close to central, and its comparison to our axial and large FoV examples is not applicable.

The concept of using a grid as a virtual image for the purpose of a rectilinear planar calibration has been proposed in [15]. Once a grid is selected as the virtual image, if can be calibrated with all other grids, effectively providing the pose of all grids. However, using a single virtual image is limited to grids inside a small FoV. All grids not included must have their pose solved separately with an alternate algorithm which tends to be unstable and accumulate errors toward the edge of the FoV. This limitation is removed in our approach by introducing multiple generic virtual cameras and allows any field of view to be calibrated, up to a full sphere.

2.2.2 Generic Axial Cameras

In the context of generic axial cameras, calibration using planar calibration grids was demonstrated to work on simulated data [69, 71]. In this model, one has to solve for the camera axis and the poses of all grids. To establish the trifocal tensor to be solved, each image point must match three calibration grids and satisfy additional constraints: the camera axis must intersect all the projection rays, and the principal point, located at the distortion center, must be known to allow a prealignment of the planes. Subse-

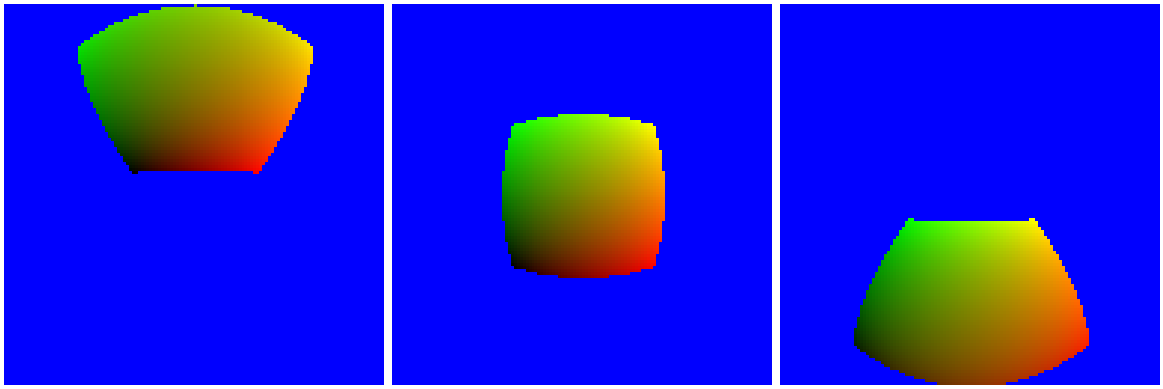


Figure 2.2: Examples of dense correspondence maps of planar grids obtained with structured light. In the 16-bit image, red and green encode x and y coordinates, respectively.

quently, a bundle adjustment is used to minimize the distance between 3D points and their projection on their respective projection rays[71]. In the case of the spherical catadioptric camera, even though this camera seems to be axial, the solved rays are distributed in a very small region approximated as the optical center. The resolved poses seem to show some inconsistencies with the actual sequence of capture [70].

2.2.3 Dense planar correspondences

Calibration of generic cameras, since they are ray-based, require dense matches between the image and calibration grids. In practice, dense correspondences can be obtained by interpolating positions from closely spaced targets [68, 97], or structured light matching on a flat LCD monitor [2, 16, 83, 84]. The correspondences are provided in the form of a lookup table, where each pixel (u, v) of the fisheye image provides a match (x, y) on the planar grid. As illustrated in Fig. 2.2, (x, y) matches are represented as red and green, respectively. Blue is used as a mask.

2.3 Generic Quasi-Central Cameras

The goal of our method is to calibrate the generic axial model, as defined in [71], but this model is underconstrained. It allows a single 3d point to have two or more rays projecting to the image, making it impossible to inverse the image formation model. In [88], the term quasi-central is introduced for models where light rays intersect near a single optical center, as a way to approximate non-central cameras with central cameras.

In this paper, the term *quasi-central* defines a model where camera rays intersect on a common camera axis, and where the displacement along this axis is a monotonic function of the ray angle with the axis. This ensures the convergence of our projection model as well as the inversibility of the image formation model. The quasi-central model, depicted in Fig. 2.1(right), is thus somewhere between central and fully axial and applies to real fisheye cameras.

2.3.1 Image Formation Model

A generic axial camera model can be represented as a lookup table (LUT) assigning to each fisheye image position (u, v) a projection ray with parameters (α, β, dz). As illustrated in Fig. 2.3, the relationship between (u, v) and the ray (α, β, dz) is simple. A central camera features $dz = 0$ while an axial or quasi-central camera provides an additional parameter dz, the vertical displacement of the optical center, which more accurately represents real fisheye lenses. A quasi-central camera requires a monotonic relation between dz and α, and full axial camera has no restriction on dz.

Typically, an equidistant fisheye has $r \propto \alpha$, and a rectilinear lens has $r \propto \tan\alpha$. Our model does not enforce a particular relationship between r and α or dz as it relies on a LUT to explicitly store individual ray information. Furthermore, it does not impose radial symmetry around the principal point.

As illustrated in Fig. 2.3, the projection of a world point (x, y, z) to a camera ray

(α, β, dz) is

$$\begin{aligned}
(\alpha, \beta, dz) &= M(u, v) \\
(u, v, dz) &= M^{-1}(\alpha, \beta) \\
(\alpha, \beta) &= f(x, y, z) \\
(u, v, dz)^k &= \underset{k=0,1,\ldots}{\mathbf{FP}} \left| \begin{array}{ll} M^{-1}(f(x, y, z)) & k = 0 \\ M^{-1}(f(x, y, z - dz^{k-1})) & k > 0 \end{array} \right.
\end{aligned} \tag{2.1}$$

where M is the image formation model, M^{-1} is its inverse, f is the projection model such that

$$\begin{aligned}
(\alpha, \beta) = f(x, y, z) = (&\Theta(\langle x, y, z \rangle, \langle 0, 0, 1 \rangle), \\
&\Theta(\langle x, y, 0 \rangle, \langle 1, 0, 0 \rangle))
\end{aligned} \tag{2.2}$$

and **FP** is the fixed point function starting at $k = 0$ which iterates until the result stops changing. We define $\Theta(\mathbf{a}, \mathbf{b})$ as the angle between vectors $\mathbf{a}$ and $\mathbf{b}$. Notice that for a central camera, dz is always 0 so the iteration stops at $k = 0$. For quasi-central, dz is unknown so it is initialized at 0. Its value will converge to the correct dz, and thus the correct (u, v) in the image, after a few iterations, assuming dz is a monotonic function of α. This assumption is realistic in practice for real lenses.

2.4 Calibrating multiple GVC cameras

Fig. 2.4 illustrates generic central and non-central camera models, where the image plane is non-planar to represent radial distortion. These models are incompatible with a linear planar calibration approach. As seen at the right of Fig. 2.4, [15] proposes to use one of the calibration grids, here depicted in red, as a *virtual image plane*, which we refer to as a Generic Virtual Central camera.

Instead of solving a single GVC for a full fisheye as in [15], we propose to use multiple GVC cameras, each using a minimal number of calibration grids, to represent

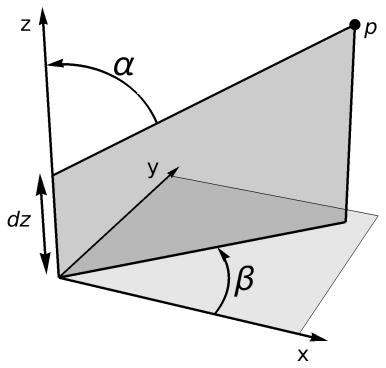


Figure 2.3: Generic quasi-central model. The projection ray of a 3D point $p = (x, y, z)$ is represented as (α, β, dz). We have $(\alpha, \beta) = f(x, y, z - dz)$.

the full generic central camera. This central camera calibration serves as the basis for the generalization to the generic quasi-central model.

2.4.1 Planar calibration for GVC model

As described in [15], *active grids* provide dense correspondence maps for the purpose of calibration. Given three grids, it is possible to find their respective poses using planar calibration [113]. Fig. 2.5 illustrates this for three grids C_0, C_1 and C_2, from which C_0 is designated *virtual image*. Notice that it is important to ensure that the relative pose of a grid with respect to the virtual image is never a pure translation, as this would yield a degenerate configuration and a failed calibration [57].

From the homographies relating C_0 to C_1 and C_2, planar calibration will provide the pose of C_1 and C_2 as well as the internal parameters $\mathbf{K}_0$. Assuming the camera is central, grid C_0 shares its optical center with the real camera, but has its own pose in the world. Because it features a known pixel ratio and size, we can derive its pose from

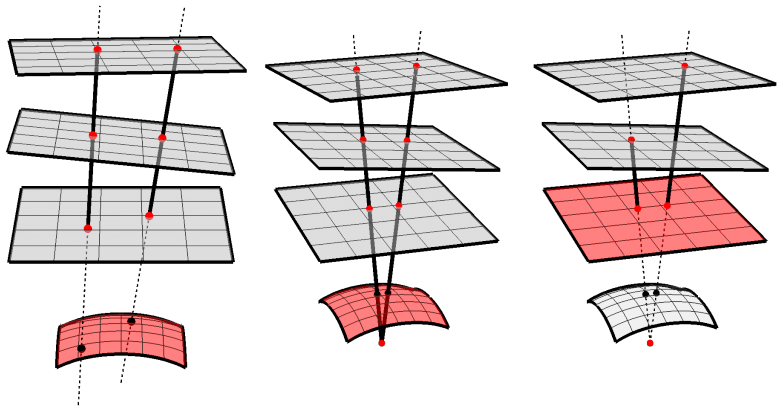


Figure 2.4: Left) Generic Non-Central camera. Middle) Generic Central camera (rays intersect at optical center). Right) Generic Virtual Central camera, where the plane in red takes the role of a virtual image plane.

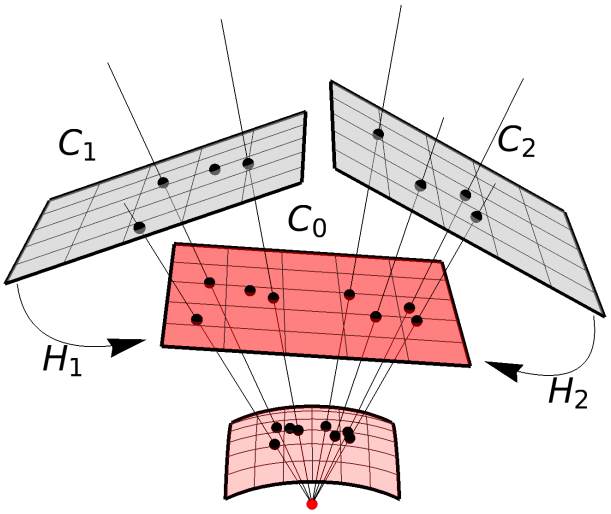


Figure 2.5: Calibration grid C_0 will act as the image plane in a planar calibration step with checkerboards C_1 and C_2, yielding the pose of C_1 and C_2 and the internal parameters ($\mathbf{K}$) for C_0.

the internal parameters of $\mathbf{K}_0$, which is of the form

$$\mathbf{K}_0 = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

Because the projection model assumes that the optical center is at the origin of the world $(0,0,0)^\top$, we can infer its origin to be at $(-c_x,-c_y,f)^\top$ and its orientation aligned with the world.

2.4.2 Grouping grids into GVC

In this planar calibration scheme, three planes are the minimum required as one of them is the virtual image plane and the two others define the homographies. More planes could be used, as it is the case in [15], but one must realize that rectilinear cameras are not well defined at large field of views (and certainly cannot reach 180° FoV), even without radial distortion. Moreover, it is most probable that grids oriented closer to the far edges of the fisheye FoV will have few correspondences with the virtual image plane, making the homography unstable. This is why we propose to always use three planes for a GVC model, and then solve multiple GVC models which are subsequently aligned together. This ensures optimal planar calibration with reasonable FoV for each GVC camera, while allowing very large FoV once grouped together.

The correspondence maps need to be separated into triplets and some correspondence maps must be part of more than one triplet. In order to do that, we construct a fully connected graph between every correspondence map and every other correspondence map were the edge weights are the negative number of matching correspondences between both maps. Subsequently, this graph can be reduced by taking its minimum spanning tree. The plane with the most edges (or number of matches in the case of a tie) is determined as the global reference plane and is marked as a visited node. From the global reference plane, two neighboring nodes are selected and this

triplet of calibration planes is assembled. The selected nodes are then marked as visited. In further iterations, each new triplet needs to count one visited and two unvisited neighbors. Each new visited node is marked as such. Special cases can happen, most notably when a visited node has only a single unvisited neighboring node. In such a case, the triplet will be composed of the visited node, the unvisited neighboring node and a previously visited neighboring node. In this way, all triplets of planes are assembled into a collection of GVC cameras which are then calibrated independently. For each triplet, any grid can be used as the virtual image plane with the one exception being the first triplet where the global reference plane needs to be the virtual image.

2.4.3 Merging multiple GVC cameras

By calibrating all GVC cameras independently, we obtain the positions and orientations of all the grids but in their respective GVC triplet reference. Some planes belong to more than one triplet and were calibrated multiple times thus have multiple poses. Since each plane has only a single pose in the world, they can be rigidly moved into alignment. The reference systems are merged using only these planes as landmarks, not their centers of projection. This provides a way to align all triplets in the world. For a central camera, the multiple centers of projections will be a single point. For a quasi-central camera, each center of projection will be axially displaced along the camera axis, as each relates to a local region of the FoV. After the merge, a single grid is kept as the reference plane. The optical axis of the merged GVC camera is not yet defined in that reference system. Although, the displacement of the centers of projection describes the camera axis, in practice these displacements are small, even nonexistent for a central camera, they are not reliable to estimate the camera axis. Finding this axis is addressed in the following section.

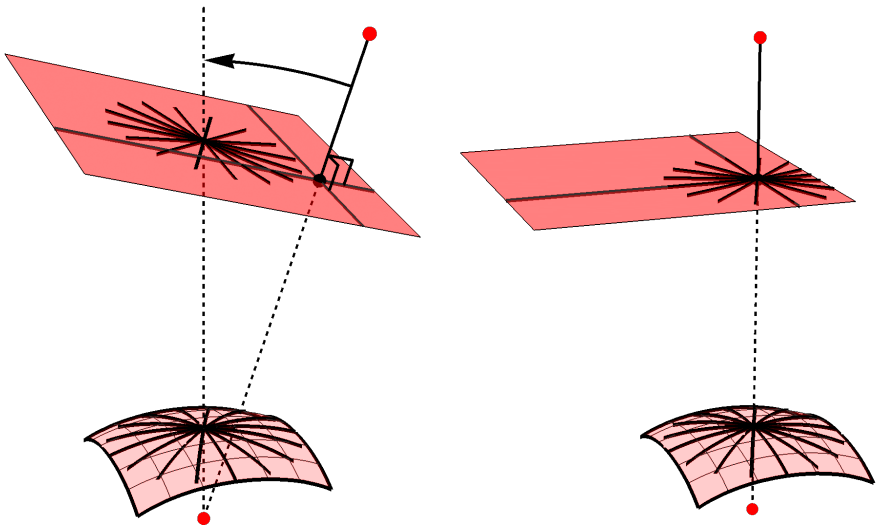


Figure 2.6: Principal point alignment. Left) The optical axis of the GVC is not aligned with the fisheye axis. Right) The principal point allows to align the GVC axis.

2.4.4 Aligning with the Principal point

By finding the center of distortion, it is possible to align the optical axis of the merged GVC camera to the optical axis of the fisheye camera. This point could be different than the principal point [36], but in practice we consider them to be the same [113].

The center of distortion is defined as the point in the fisheye camera with the least distortion, where straight lines in the world should remain straight in the camera image, as illustrated in Fig. 2.6. We find the center of distortion by locating straight lines in the fisheye image which correspond to collinear matches in their planar grid. The merged GVC camera is then rotated such that the virtual principal point, corresponding to the fisheye image principal point, lies on its Z-axis. In theory, this alignment is not absolutely necessary, except to simplify the representation of radial symmetry and axial displacement.

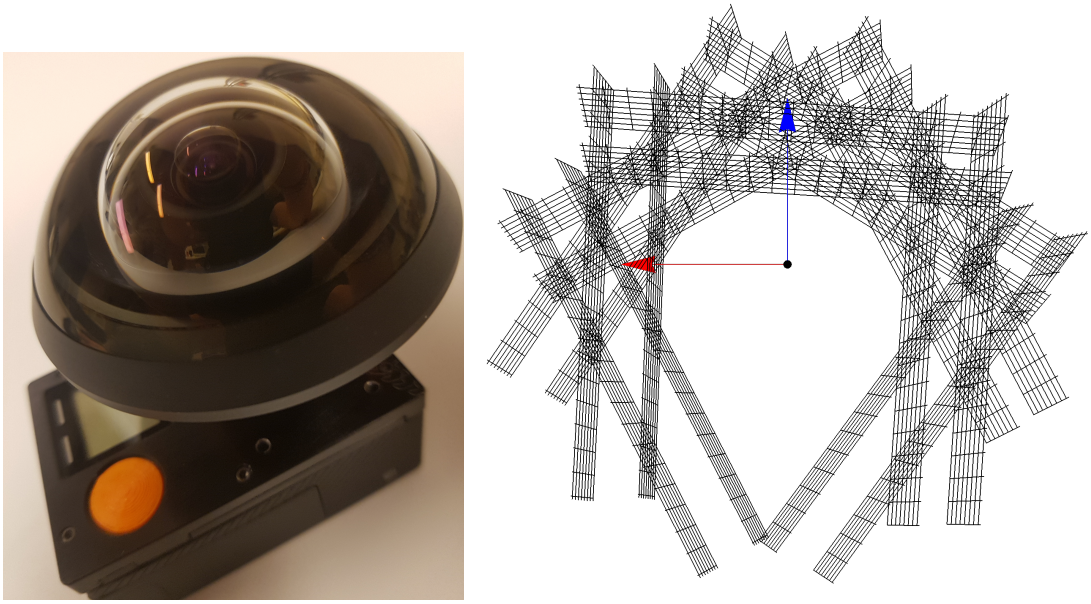


Figure 2.7: Left) Entaniya 280° Fisheye Lens mounted on a GoPro HERO 4. Right) Recovered poses of the calibration grids around the fisheye. Notice the near and far sets of planes.

2.4.5 Generalizing to quasi-central camera model

For the axial fisheye cameras, the above method has a severe weakness because homographies assume that all rays intersect at a single point. In the case of axial cameras that is in fact false, since there will be multiple optical centers distributed along a common axis. In this situation, the homographies will in fact provide a single optical center representing the average of all these optical centers.

In order to generalize from central to quasi-central, an optimization procedure is devised. Let us consider the poses of the planar calibration grids $C_i : (\mathbf{R}_i, \mathbf{T}_i)$ and the calibration grid set defined as $\Pi = \left\{C_1, \ldots, C_n\right\}$. For given coordinates (u, v) in the fisheye camera image, the 3D point $\mathbf{p}^i_{u,v}$ located on plane i corresponding to the fisheye image point (u, v) is extracted directly from the lookup tables $LUT^i(u, v)$.

In this way the 3D point in the world centered around the optical center is defined

as

$$\mathbf{q}_{u,v}^{i} = \mathbf{R}_i \mathbf{p}_{u,v}^{i} + \mathbf{T}_i \tag{2.4}$$

This leads on to defining a line between two points i and j such as

$$L_{u,v}^{i,j} = \left\langle \mathbf{q}_{u,v}^{i}, \mathbf{q}_{u,v}^{j} \right\rangle \tag{2.5}$$

One special line is the camera axis L_z defined as the Z-axis. Furthermore, we rely on two functions, $\Theta(L_a, L_b)$ (see Eq. 2.2) and $d(L_a, L_b)$, to provide the angle and distance, respectively, between two lines L_a and L_b. The minimization problem is formulated as

$$\arg\min_{\Pi} \sum_{\{i,j\}}^{|\Omega|} \sum_{u,v} d(L_{u,v}^{i,j}, L_Z) + \gamma \sum_{\{i,j\}}^{|\Omega|} \sum_{\{a,b\}}^{|\Omega|} \sum_{u,v} \Theta(L_{u,v}^{i,j}, L_{u,v}^{a,b}) \tag{2.6}$$

where Ω is the set of pairs of planes such that planes i and j have correspondences and are separated by a sufficient distance and where γ is a weighting parameter. This distance is required to ensure stability because lines generated between overlapping planes are very unstable. This has an important consequence in practice as it requires two sets of calibration grids, one closer and one further away from the camera (see Fig. 2.7(right)). The cost function is illustrated in Fig. 2.8 for two pairs of planes. The planes i and j must be positioned so the distance d between line $L_{u,v}^{i,j}$ and the Z-axis is 0. Also, the angle Θ must be minimized to ensure parallelism with a line $L_{u,v}^{a,b}$ related to the same fisheye image point (u, v).

This above minimization problem is presented in two separate terms which minimize different objectives. The first aims to enforce that all lines intersect the camera axis. The second aims to enforce parallelism between lines corresponding to a common (u, v) point in the fisheye image, since this is the basic assumption of the generic camera model. This second term is required since the optimization does not solve all parameters globally, so multiple conflicting solutions for plane poses are possible.

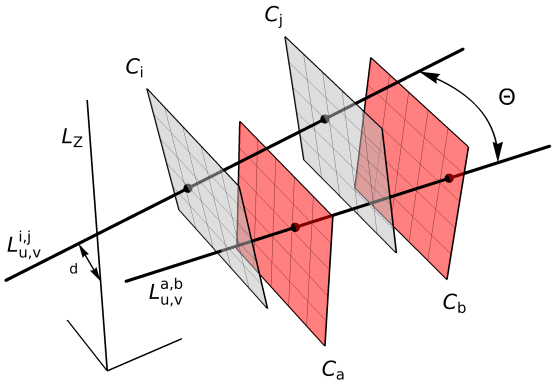


Figure 2.8: Minimization of planes i and j must reduce distance d with the Z-axis, as well as Θ for parallelism with the matching line from the same (u, v).

2.5 Experimental results

We provide two different types of experimental results. First, synthetically generated fisheye cameras, both central and quasi-central, are used to validate our calibration method. Second, calibration results of a real large FoV camera are provided demonstrating practical use of the quasi-central model and calibration.

Comparing experimental results with other methods proved to be challenging. In [70, 71], they calibrate an axial camera model where the camera is in fact two central pinhole cameras in a stereo configuration which is not quasi-central. Since our method requires quasi-central, no comparison can be performed. Their further work in [68] constrains their model to a central camera but its 183° FoV cannot be compared to ultra-wide. Finally, the method in [97] describes a framework for generic calibration and is theoretical in nature. Although they establish the groundwork for truly generic camera models, in the case of fisheye cameras, the results shown are preliminary as

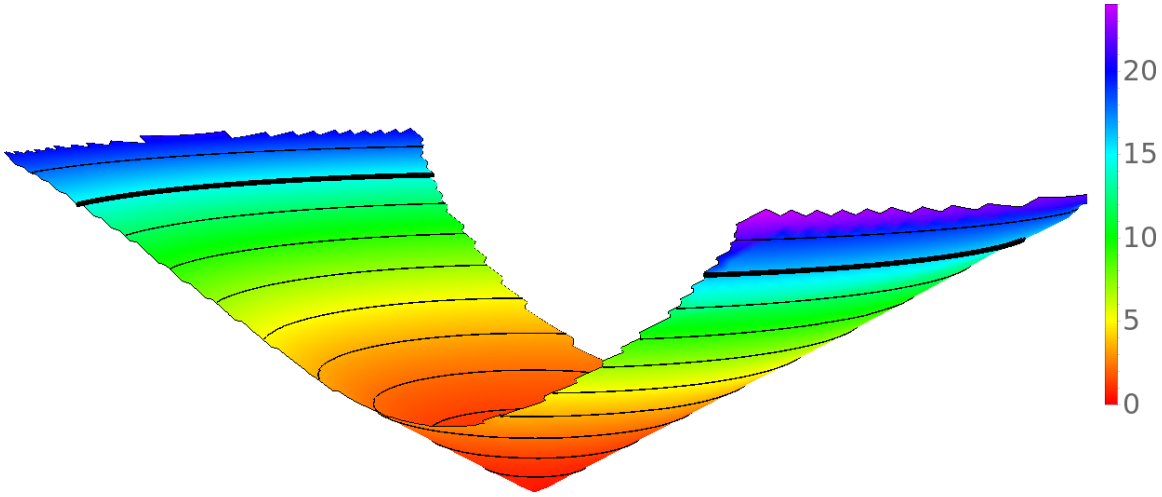


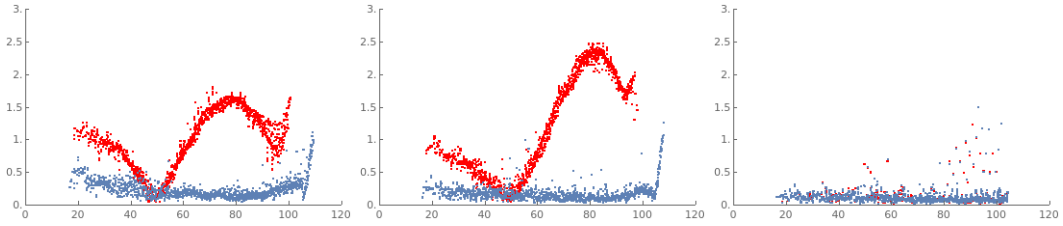
Figure 2.9: 3D view of the lookup table for the Synthetic Axial camera with parabolic displacement. Height direction is α with isocontours at every 10° (90° is in bold). Axial displacement dz is color coded from 0 to 24 plane pixels (see legend).

only a very limited FoV is calibrated.

2.5.1 Synthetic Calibration

The synthetic fisheyes have a 220° field of view with three different axial displacement modalities, respectively a linear axial displacement of $\frac{20}{r}$, a parabolic displacement of $\frac{20}{r^2}$ and no axial displacement (a central camera). For each fisheye cameras, the projection rays associated with each pixel were generated. Theses rays are then intersected with various planar grids at various poses to generate dense correspondence maps (see Fig. 2.2). The calibration is done using only these correspondence maps.

Image formation model. The result of Fig. 2.9 is the recovered image formation model $M(u, v)$ (see Eq. 2.1) for the synthetic camera with the parabolic axial displacement. We observe that the result closely matches the camera specification. The cone shape indicates an equidistant fisheye and the axial displacement is recovered accurately as parabolic.



(a) (b) (c)

Figure 2.10: Reprojection error in pixels, as a function of α, for calibrating a central model (in red) and a quasi-central model (in blue). Cameras are fisheye 220° with various axial displacements: A) linear, B) parabolic, C) none.

Reprojection error. Fig. 2.10 presents the reprojection error for all three synthetic cameras, obtained by reprojecting all the points of a single calibration grid into the fisheye image. The selected grid is the same for all 3 cameras and was chosen because it fills a large portion of the FoV and includes areas where the radial distortion and axial displacement are highest.

When applying central model calibration on axial cameras (red curves in Fig. 2.10a and 2.10b) we observe high reprojection errors with two dipping points, indicative of the *average pose* recovered by this inadequate model. This is similar to the error observed when fitting a line to a portion of a second-degree curve. The fit will be wrong, except maybe at two points.

When applying quasi-central model on the same cameras, shown as blue curves in Fig. 2.10a and 2.10b, the curves are near constant at a low reprojection error (0.5 pixels) which also corresponds to the lowest point on the red curve. This demonstrates that our proposed quasi-central calibration method performs as intended.

For the central camera, depicted in Fig. 2.10c, both central and quasi-central calibration feature low reprojection errors, demonstrating that both methods perform as expected.

Table. 2.1 provides a global performance estimate at a glance. For synthetic as well

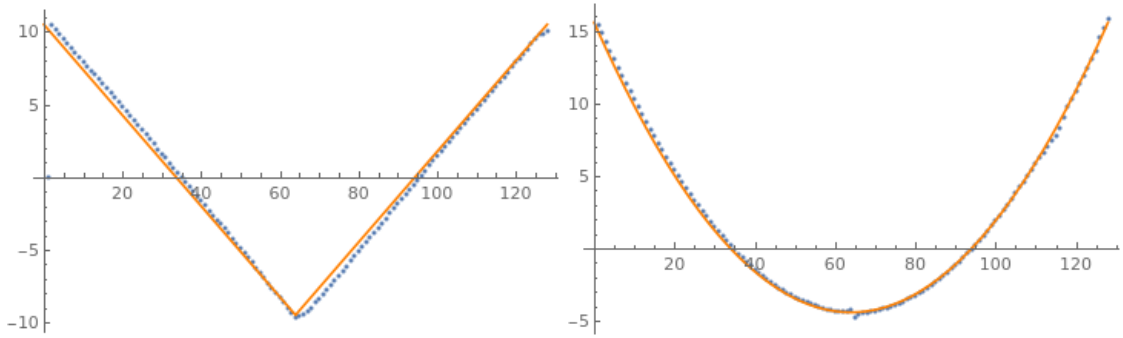


Figure 2.11: Axial displacement for the synthetic axial cameras, linear (left) and parabolic (right). In orange are the true curves and in blue are the axial displacements recovered across a vertical slice of the fisheye image. Units are in image pixels.

as real cameras, calibrations with low reprojection errors are achieved. Moreover, using the axial model on axial cameras further reduce the error.

Axial displacement. Fig. 2.11 illustrates that the axial displacement is correctly recovered by the quasi-central model.

2.5.2 Real Camera Calibration

The experimental setup for the fisheye calibration uses a LG29UM57-P LCD monitor with a resolution of 2560x1080 pixels and a 0.2651mm dot pitch. The camera is a GoPro HERO 4, with an Entaniya 280° Fisheye lens attached, in 2.7K 4:3 Ultra Wide video resolution mode which results in a 2704 × 2028 full frame image (see Fig. 2.7(left)). The camera is mounted on a pan-tilt unit located at about 20cm from a monitor acting as an active calibration grid, and rotated to provide multiple calibration grids for different points of view. A second set of acquisitions is performed at 30cm from the monitor. Note that although the amount of rotation between planes and the approximate distances between the camera and the monitor are known, the calibration is done solely using the correspondence maps.

Image formation model. Fig. 2.12 is the recovered image formation model $M(u, v)$

		Field of View Region			
Camera	**Calib**	**Center**	**Middle**	**Outer**	**Full**
Synthetic Central	Central	0.2	0.1	0.1	0.1
	Axial	0.2	0.1	0.1	0.1
Synthetic Axial Linear	Central	0.5	0.8	1.2	0.8
	Axial	0.4	0.2	0.2	0.3
Synthetic Axial Parabolic	Central	0.3	0.9	1.9	1.1
	Axial	0.2	0.4	0.2	0.4
Real Entaniya 280°	Central	4.2	4.9	6.4	5.0
	Axial	4.5	3.0	4.4	3.8

Table 2.1: Reprojection error in pixels, averaged for different regions of the field of view, for central and axial calibrations of the synthetic and real cameras. We observe that axial calibration yields lower error for axial cameras.

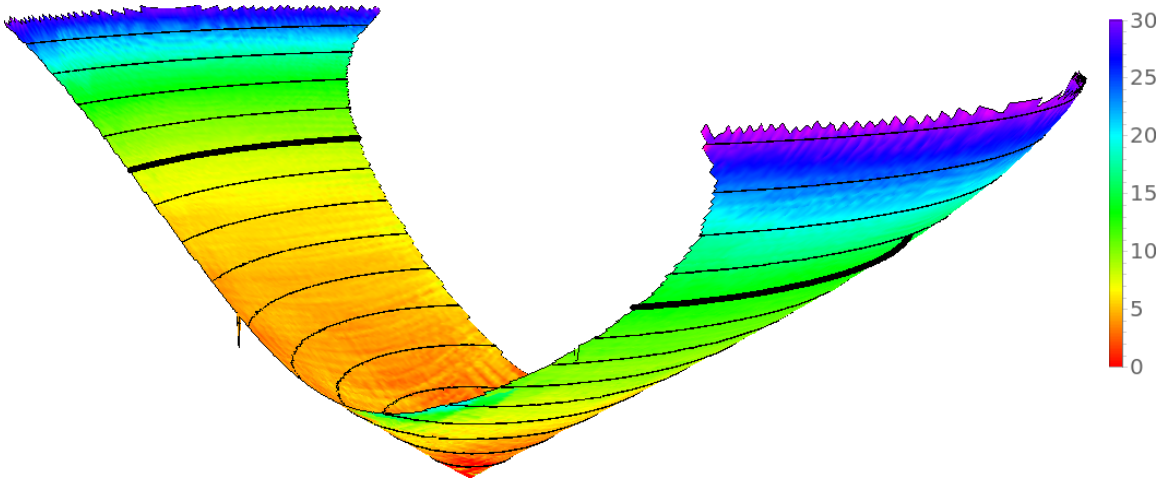


Figure 2.12: 3D view of the lookup table for the Entaniya 280°. Height direction is α with isocontours at every 10° (90° is in bold). Axial displacement dz is color coded from 0 to 35mm (see legend).

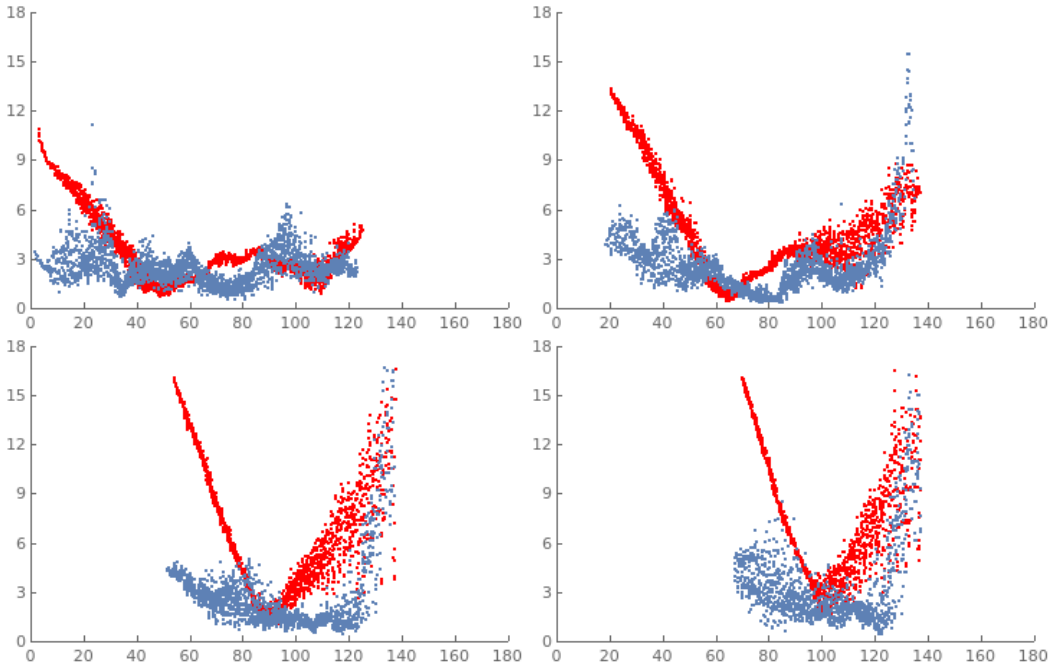


Figure 2.13: Reprojection error in pixels, as a function of α, of four calibration grids. Curves results from calibrating a central model (in red) and a quasi-central model (in blue) for a 280° fisheye.

(see Eq. 2.1) for the Entaniya 280°. We observe that the α angles are recovered well over the full field of view and indicate near equisolid geometry. The axial displacement is significant and also recovered well. It is compatible with the lens diameter of 64mm and the maximum incident angle of 50° above the horizon.

Reprojection error. Similarly to the synthetic camera cases, the reprojection error is computed on different calibration grids along the field of view. Four of them were chosen as representative of the expected results and together they cover the whole FoV.

Overall, we observe in Fig. 2.13 that for very wide FoV fisheyes, modeling the axial displacement greatly improves calibration. In all cases, the reprojection error for the quasi-central calibration can be averaged to 3 pixels or less, while the central calibration is up to 15 pixels. Notice the "V" shapes in the red curves which are steeper with

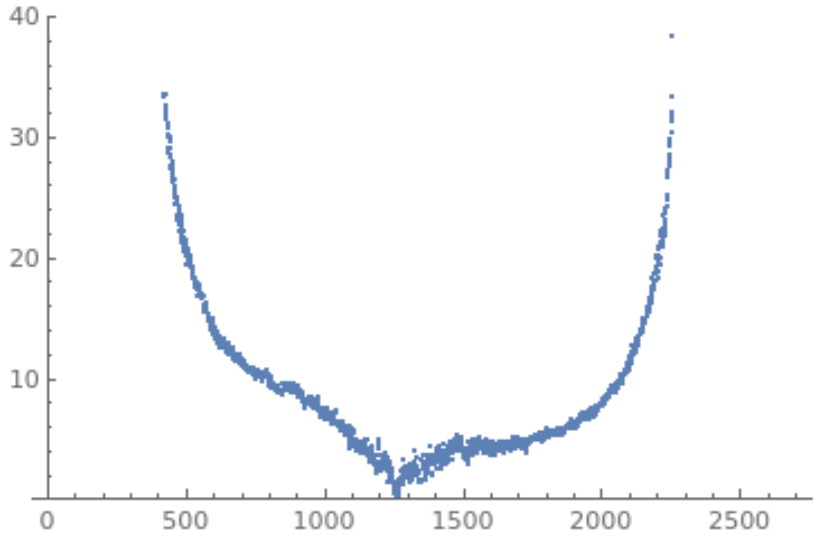


Figure 2.14: Axial displacement in mm for the 280° camera, recovered across a horizontal slice of the fisheye image.

increasing angle due to the averaging effect of the central calibration. This means that the farther away from the camera axis you are, the worst a central calibration performs, and the more essential quasi-central calibration becomes.

Axial displacement. To better assess the quality of the quasi-central calibration, Fig. 2.14 shows the recovered axial displacement in mm across the camera image. This illustrates that near the edges, the optical center has moved by up to 30mm from its initial position and confirms why a central approximation to such a camera is inaccurate. Notice that the curve is not symmetric around the principal point. We believe that this is a result of reaching an inaccurate local minimum during the optimization step. This was observed only for specific grids, suggesting some inaccuracy in the structured light matching.

2.6 Conclusion

This paper presented a new calibration method for fisheye cameras, with an underlying Generic Quasi-Central camera model. It can bypass any image distortion by using calibration planes as virtual images, the Generic Virtual Central cameras, which are perfectly rectilinear and can be solved with simple planar calibrations. The resulting calibration allows very large fields of view. This model is then generalized to recover axial optical center displacements. Although the optimization scheme which recovers grids poses is simple and could be further improved, results on a real lens demonstrate not only that axial calibration works, but that it significantly improves the reprojection error. In the future, we expect that the Quasi-Central model and multiple GVC cameras will find more uses such as generating perfect image rectifications, generalize to other models, improve fisheye image stitching for immersive imaging and enhance 3D reconstruction from the triangulation of fisheye images.

Chapitre 3

APPRENTISSAGE PROFOND EN VISION PAR ORDINATEUR

L'apprentissage profond est une technique d'apprentissage machine qui vise à simuler les méthodes humaines d'apprentissages par l'intermédiaire d'un réseau de neurones. L'arrivée de la rétropropagation en 1986 [81] a permis l'entraînement de tels modèles. Toutefois, les performances étaient très limitées dues à l'absence de données et de systèmes informatiques suffisamment puissants. Ceci a changé après l'arrivée des cartes graphiques au début des années 2000 et des percés en reconnaissance vocale par l'équipe d'Hinton [40] utilisant des techniques d'apprentissage profond. Les années 2010 ont été marquées par les réseaux sociaux et les géants de l'informatique alors que la population mondiale se retrouve connectée en permanence à travers leurs téléphones intelligents. Ceci entraîne une hausse massive de la quantité de données générée par chaque individu, allant des photos quotidiennes et les données de navigation GPS jusqu'aux contenus regardés et l'historique de navigation sur le web.

Cette nouvelle masse de données recoupe le monde physique et le monde digital de telle sorte qu'il devient possible d'en extraire des indications fortes sur la façon de penser de ses utilisateurs. Dans l'autre direction, il devient possible d'utiliser ces données pour prendre des décisions dans le monde physique. Le présent chapitre se concentre sur les systèmes d'apprentissage profond qui utilisent les données qui se rapportent à la vision. La vision est un concept très important pour l'humain puisqu'elle représente la plus grande partie de l'information qui provient du monde extérieur. D'abord, ce chapitre présente les fondements des notions en apprentissage profond appliqués à la vision notamment les réseaux de neurones profonds à convolution. Puis il présente les architectures de réseaux de neurones typiquement utilisés dans la littérature avant de conclure sur les grandes tâches en vision telles qu'elles sont résolues par l'apprentissage

profond.

3.1 Fondements de l'apprentissage profond en vision

Le domaine de l'apprentissage profond propose des méthodes efficaces pour obtenir des systèmes intelligents pour effectuer des tâches de classification, de régression ou de regroupement. En augmentant le nombre de neurones et le nombre de couches dans un réseau, la fonction de décision augmente elle-même en complexité. Spécifiquement dans le domaine de l'apprentissage supervisé, les tâches qui nécessitent de réaliser une correspondance entre une donnée en entrée et une représentation choisie en sortie peuvent être accomplies à l'aide des réseaux de neurones. La précision et l'exactitude de cette fonction de correspondance retrouvée demeurent limitées par la quantité et la qualité des données d'entraînement. Il est mentionné par Goodfellow et al. [1] que les tâches qui ne peuvent être décrites comme la transformation d'un vecteur vers un autre demeurent des tâches difficiles à résoudre malgré l'apprentissage profond.

3.1.1 Réseau de neurones

Un réseau complètement connecté connu aussi sous le nom de perceptron multicouche est le réseau classique en apprentissage profond. Le réseau est appelé en anglais "feed-forward" puisqu'il propage le signal vers l'avant avec aucune connexion vers les neurones voisins ou les neurones précédents comme on pourrait voir chez les graphiques markoviens. Bien que les réseaux complètement connectés soient peu utilisés dans le domaine de la vision par ordinateur, il est pertinent de comprendre comment ceux-ci sont entraînés puisque ces concepts s'appliquent généralement à tous les réseaux de neurones.

Tel que décrit par Goodfellow et al. [1] l'idée d'un réseau est d'approximer une fonction f^* qui transforme une donnée x en sa sortie y à l'aide de $f^*(x;\theta)$. La fonction approximative $f(x;\theta)$ peut être composée de plusieurs fonctions f_1 jusqu'à f_n et la mise en

chaîne de ces fonctions entraîne $y = f(x;\theta) = f_n(\theta_n) \bullet f_{n-1}(\theta_{n-1}) \bullet ... \bullet f_2(\theta_2) \bullet f_1(x, \theta_1)$. Cette écriture indique simplement d'où le terme apprentissage profond tire son nom. En effet, chaque couche d'un réseau de neurones se rapporte à une des fonctions f_i qui applique une transformation au résultat de la fonction, de la couche, précédente. Ainsi, plus un réseau est dit profond plus cette chaîne de fonction est longue.

L'étape d'entraînement d'un réseau de neurones est l'étape où l'on fait tendre la fonction f vers la fonction f^*. Dans l'apprentissage supervisé, un jeu de données est fourni qui représente un ensemble d'exemples x_i et son $y_i^* = f^*(x_i)$ respectif. Il est espéré que la fonction f provenant du réseau de neurones donnerait à sa sortie un $y_i \approx y_i^*$. Le problème devient simplement de minimiser la distance entre $f(x_i;\theta)$ et $f^*(x_i)$ pour les paramètres θ. La définition même de cette "distance" est toutefois à déterminer selon l'application et selon les données. Des données de meilleure qualité contribuent donc très grandement lors de l'apprentissage. Ces données présentent des paires x_i et y_i^* qui sont le plus fortement corrélés par f^*. En d'autres mots, dans les jeux de données $y_i^* = f_i^*(x_i)$ où f_i^* est en moyenne égal à f^*. Plus la quantité de données est grande et plus la qualité des données est bonne, plus l'entraînement sera efficace.

La métrique de distance est appelée la fonction de coût. Lors de tâches de classification, la fonction choisie est l'entropie croisée entre les sorties dites "prédites" et les sorties cibles. Dans d'autres cas, la distance est tout simplement une distance euclidienne. Évidemment, en plus d'avoir un jeu de données qui est bon, il est essentiel d'avoir une fonction de coût qui décrit bien le problème. Puisque l'apprentissage est décrit comme un problème de minimisation, ceci indique que lors de la minimisation, les valeurs des paramètres vont varier suivant la variété (manifold) décrite par la fonction de coût. Ainsi, si la fonction de coût est discontinue en certains points ou n'est pas lisse, il est possible que la minimisation reste piégée dans un minimum local. La minimisation est habituellement effectuée à l'aide d'une méthode par descente de gradient. Les paramètres sont initialisés à des valeurs aléatoires et à chaque itération ils sont modifiés afin de satisfaire à la direction de descente du gradient. Il est important de

mentionner qu'une telle descente de gradient ne garantit pas la convergence du problème de minimisation. De nombreuses techniques comme la régularisation, le dropout et le "batch normalization" ont été développées afin d'éviter les minimums locaux et le surapprentissage durant l'entraînement.

L'étape de rétropropagation est l'étape du calcul du gradient dans un réseau de neurones. Cette étape vient après la propagation où l'on estime des sorties pour un certain sous-ensemble des données et après l'étape du calcul de la distance entre ces sorties approximées et les sorties cibles. Tel que décrit par Goodfellow et al. [30], durant la rétropropagation on calcule le gradient d'un scalaire z en fonction d'un certain ancêtre. Il est possible de calculer le gradient en z puisque $\frac{dz}{dz} = 1$. Il est ensuite possible de calculer le gradient des parents de z en multipliant le gradient courant par le Jacobien de la fonction qui engendre z. On remonte ainsi jusqu'à l'ancêtre en multipliant à la chaîne le gradient par les Jacobien successifs en se promenant à reculons à travers le réseau de neurones. Ce gradient est calculé symboliquement par dérivation en chaîne au début de l'entraînement pour tous les neurones du réseau rendant l'entraînement très rapide.

3.1.2 Réseau à convolution

Les réseaux à convolutions sont des réseaux de neurones qui sont spécialement appropriés pour les problèmes où les données présentent une structure en grille. Ces réseaux ont d'ailleurs été développés pour la reconnaissance des codes postaux écrits à la main sur les lettres [50]. Plus spécifiquement, les réseaux de neurones à convolution (CNN) sont des réseaux qui utilisent une ou plusieurs couches de convolution. Cette couche remplace la couche complètement connectée et donc remplace une fonction linéaire par une fonction mathématique de convolution. Cette étape de convolution se veut en quelque sorte une façon de contraindre les fonctions apprises à être d'un type qui s'applique mieux aux données.

Les CNN sont inspirés du système visuel et plus spécifiquement du cortex visuel des chats et son modèle présenté par Hubel et al. [42]. La couche de convolution est une couche qui va appliquer une opération de convolution avec un certain nombre restreint de paramètres sur une petite fenêtre de l'entrée. La fenêtre est ensuite balayée sur l'ensemble de l'entrée. De cette façon chaque région de l'image va partager localement les mêmes paramètres. Les paramètres organisés dans une fenêtre créent ensemble un masque de convolution. Cette contrainte vient réduire grandement le nombre total de paramètres dans l'image et inversement augmente grandement les performances pour les tâches où l'information est structurée en régions comme les images. En pratique, chaque couche de convolution comporte plusieurs masques et donc à chaque couche, l'entrée est transformée en un volume de sortie. Dans un réseau pleinement convolutif, on peut réduire la dimensionnalité de la sortie jusqu'à un volume 1D soit un vecteur ou bien plutôt un volume qui représente une image. Ce sont ces masques qui sont appris lors de l'entraînement [105].

Tel que décrit par Voulodimos et al. [105] un réseau de neurones s'appuie sur des concepts clés soient les champs réceptifs locaux, le partage de poids et le sous-échantillonnage spatial. Le champ réceptif local indique qu'un neurone va recevoir en entrée de l'information provenant d'unités qui sont obligatoirement voisines. Ceci permet l'apprentissage de masques qui sont capables de représenter des notions comme la détection de contours. De plus, ces masques, qui s'activent à un endroit de l'entrée, sont très probables de s'activer à un autre endroit de l'entrée, et ce spécialement dans le contexte d'images. Le concept du partage de poids permet de réutiliser le masque partout sur l'entrée et ainsi d'apprendre les masques qui s'appliquent le mieux partout. La sortie produite par chaque masque à chaque couche est appelée carte de caractéristiques (feature map). Les réseaux de neurones à convolution réussissent à surpasser les performances des techniques classiques dans plusieurs problèmes de vision par ordinateur l'une d'elles étant la compétition ImageNet [48]. Leurs excellentes performances ainsi que la facilité avec laquelle ils sont entraînés sont la raison de leur popularité

[105].

3.1.3 Réseaux de neurones profonds

La représentation hiérarchique des caractéristiques, soit les multiples représentations des caractéristiques des pixels à différents niveaux, peut être apprise automatiquement à partir des données. De plus, les facteurs cachés qui influencent les sorties peuvent être démêlés à travers un réseau de neurones profond [6]. Les réseaux qui sont profonds offrent une capacité d'exprimer des fonctions complexes. Un CNN plus profond offre entre autres la possibilité d'optimiser plusieurs tâches simultanément. D'un autre côté, un très large CNN permet de repenser plusieurs tâches de vision par ordinateur comme des tâches de correspondance d'une donnée en très haute dimension vers une cible [114].

Une idée évidente pour augmenter la performance des CNNs est donc d'augmenter leur profondeur, nombre de couches, et d'augmenter leur largeur, nombre de masques par couche [100]. Prenons GoogLeNet[100] et VGG [91] qui ont chacun leur tour été l'état de l'art dans leur tâche respective. Ils démontrent qu'augmenter la taille du réseau est bénéfique et permet d'augmenter les performances. Certains autres vont d'ailleurs réaliser des réseaux très profonds en mettant bout à bout plusieurs réseaux. La sortie d'un réseau est ainsi reprise en entrée par un réseau subséquent [35]. Ceci permet d'entraîner séparément les réseaux et de produire des réseaux qui produisent plusieurs sorties différentes voire même des sorties intermédiaires spécifiques à la tâche. Wang et al. [107] ont d'ailleurs connectés deux CNN, le premier ayant comme objectif de localiser les objets dans la scène et le second permettant de les segmenter. De cette façon, bien que ce réseau en soit un de segmentation, le réseau initial retourne des coordonnées dans l'image. Sun et al. [99] ont pour leur part réalisé un réseau de neurones pour détecter les points clés dans un visage tels les yeux, la bouche et le nez. Leur architecture est par contre en trois réseaux bout à bout où chaque réseau

propose progressivement une version plus précise de l'emplacement des points clés. Alors que le premier réseau offre un estimé vague des différentes positions, le dernier retourne des points concordant avec la position précise. Finalement, Ouyang et al.[62] présente une approche en deux réseaux qui se suivent où le second va traiter que les cas de classification où le premier réseau est incertain. Les deux réseaux sont entraînés conjointement dans des tâches complémentaires.

Une autre façon de réaliser ceci est d'exécuter des réseaux en parallèle plutôt qu'en série. De cette façon chaque réseau peut être entraîné indépendamment. Ciresan et al.[11] ont d'ailleurs proposé d'entraîner plusieurs réseaux de neurones indépendamment et de faire la moyenne de leurs résultats. Ceci est une forme de "Bagging" ou bootstrap aggregating où l'on construit un classifieur fort à partir de plusieurs classifieurs faibles. En combinant les architectures en séries et les architectures en parallèle, il est apparent que les réseaux à convolution plus profonds puissent maintenant s'adapter à des tâches complexes en mettant à profit les connaissances à priori sur les caractéristiques de la solution recherchée.

3.2 Architectures CNN typiques

Les récents résultats des CNN dans le domaine de la vision par ordinateur ont permis l'émergence de quelques architectures qui se démarquent. Cette section présente les architectures CNN couramment utilisées, leurs caractéristiques et leurs applications.

AlexNet [48] présenté lors de la compétition ILSVRC2012 (ImageNet Large Scale Visual Recognition Challenge) est un réseau de neurones avec dans l'ordre cinq couches de convolution et 3 couches complètement connectées. De plus, après chaque couche de convolution se trouve une couche de "pooling" ou mise en commun. Il prend en entrée des images de taille 224x244 et est entraîné sur l'ensemble de données ImageNet avec de l'augmentation de donnée et du dropout. Tel que mentionné par Russakovsky et al. [82] ce réseau est responsable du mouvement de la communauté en vision par

ordinateur vers l'apprentissage profond. Par contre, les auteurs mentionnent que trois ans après la publication originale, il demeure incertain pourquoi ce réseau atteint des performances si hautes. On note dès lors une faiblesse importante des réseaux de neurones soit leur caractère inexplicable.

À l'inverse d'un réseau large, VGG [91] est un réseau beaucoup plus profond, mais très mince à chaque niveau. Les auteurs mentionnent que les performances sont explicables par la profondeur uniquement. Ceci permet d'obtenir un réseau qui s'attarde à des caractéristiques dans l'image qui sont très complexes. Chaque niveau de profondeur dans un réseau de neurones ajoute un niveau d'abstraction aux caractéristiques identifiées. De même, Szegedy et al.[100] ont proposé un modèle, GoogLeNet, qui a une profondeur de 22 couches et qui surpasse les performances réalisées par AlexNet et ce lors de la compétition ILSVRC2014.

Grâce à ces modèles, deux grandes approches par réseaux de neurones ont vu le jour dans le domaine de la vision par ordinateur. Ces approches sont les réseaux de neurones à convolution basés régions (RCNN) (voir 3.3.2) qui se spécialisent dans les tâches de détection d'objets et les réseaux de neurones complètement convolutif (FCN) qui se spécialisent dans les tâches de segmentation [35] (voir la section suivante).

3.2.1 Réseaux de neurones complètement convolutif (FCN)

Avant de parler des applications, il est nécessaire de souligner les contributions que l'apparition du domaine des réseaux complètement convolutif sur l'ensemble du domaine des réseaux de neurones en vision par ordinateur.

D'abord, une problématique bien réelle est la disparition d'un gradient dans un réseau de neurones le "vanishing gradient problem". À mesure que le gradient est propagé lors de l'étape de rétropropagation, celui-ci est multiplié par un jacobien. Ainsi, pour les jacobiens inférieurs à 1, la valeur du gradient est décroissante. En pratique, les gradients sont pratiquement toujours décroissants pour un réseau initialisé avec des

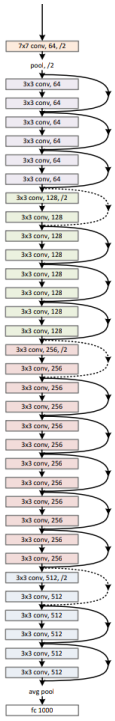


Figure 3.1: Un expleme de réseau résiduel (ResNet [39]) de 34 couches. Les raccourcis permettent de court-circuiter des couches.

poids entre -1 et 1. De cette façon le gradient tend vers 0 pour les premières couches d'un réseau très profond lors de la rétropropagation. Ceci limite la profondeur maximale d'un réseau puisqu'un gradient nul n'entraîne aucun apprentissage au niveau d'une couche. Le réseau ResNet [39] présenté à la figure 3.1 vient introduire une connexion entre une couche et une autre plus loin dans le réseau. Cette connexion est un raccourci qui vient court-circuiter certaines couches en multipliant une des entrées par l'identité. De cette façon, le gradient peut passer par les raccourcis lors de la rétropropagation et ne pas disparaître.

Une autre problématique des réseaux de neurones à convolution est que l'opération de convolution est une opération d'agglomération. Voilà pourquoi la sortie d'une couche de convolution présente des dimensions plus petites que l'entrée. Il est pos-

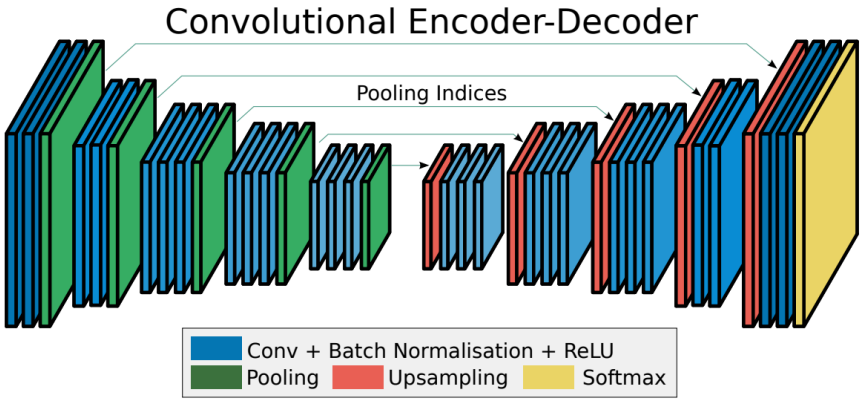


Figure 3.2: Un exemple de réseau Encodeur-Décodeur SegNet [1]. Un décodeur suréchantillonne son entrée en utilisant les indices transférés de son encodeur.

sible de contrer ce problème avec du “padding” ou du rembourrage, mais il est très difficile d’avoir une sortie aux dimensions plus grandes que l’entrée. La couche de déconvolution ou convolution transpose est une couche qui permet du suréchantillonnage de son entrée, mais dont les poids sont appris. SegNet [1], qui est montré à la figure 3.2, introduit l’idée d’une convolution transpose qui prend en entrée les valeurs des indices des couches de “pooling” utilisées en amont dans le réseau. Ceci permet un suréchantillonnage qui n’est pas appris, mais est plutôt l’inverse direct de l’étape de “pooling” hautement non linéaire.

Finalement, les réseaux de neurones sont entraînés sur une seule fonction de coût. Celle-ci tend à minimiser globalement l’erreur. Il n’est pas clair que le réseau de neurones va apprendre une représentation détaillée de la distribution. Notamment, dans les cas de segmentation, le réseau identifie globalement la région recherchée, mais est incapable de retourner un résultat qui segmente les détails. La figure 3.3 montre U-Net [79] qui est un réseau qui présente une architecture en échelle où l’image d’entrée peut emprunter plusieurs chemins lors de la traversé du réseau. Grâce aux connexions raccourcies, il est possible de réaliser un réseau où l’entrée traverse simultanément plusieurs réseaux avec différentes profondeurs qui partagent des poids. Ceci permet

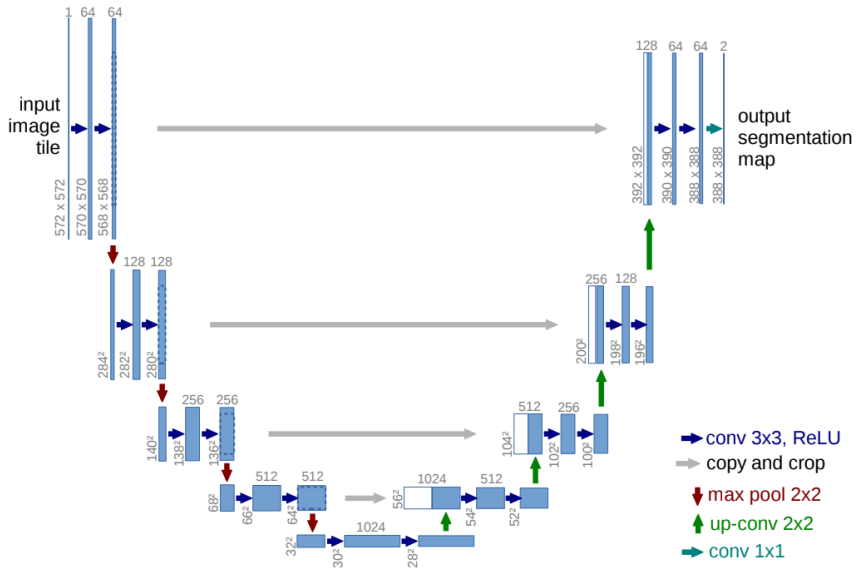


Figure 3.3: Un exemple de réseau U-Net [79]. Cette architecture permet cinq réseaux en parallèle de longueur croissante qui partagent des poids.

un réseau de segmentation parallélisé qui entraîne à la fois un réseau qui se concentre sur le caractère général et un réseau qui se concentre sur les détails. Dans leur cas, le réseau proposé fonctionne à cinq niveaux de détails en parallèle.

3.3 Problèmes classiques

Le domaine de la vision par ordinateur s'attaque aux tâches qui nécessitent une caméra ou qui utilisent des images comme sources de données principales. Bien entendu, les données vidéo sont comptées comme une série d'images organisées temporellement. Diviser le domaine en application est une tâche ardue qui laisse place à beaucoup de subjectivité. Dans cette section, les taches de la classification d'images, de la segmentation sémantique et de la détection d'objets sont analysées puisque celle-ci a été particulièrement affectée par l'arrivée des techniques d'apprentissage profond. Les tâches de Cartographie et localisation simultanées (SLAM), de correspondance stéréo et de cali-

brage sont explorées dans d'autres chapitres et jusqu'à ce jour se prêtent difficilement à être résolue par apprentissage profond.

3.3.1 Classification d'images

La tâche de classifier des images implique de retrouver l'étiquette appropriée pour une image donnée selon une banque d'étiquette correspondant au cas traité. L'étiquette retournée est donc un indice de classe signifiant que la sortie lors d'une tâche de classification est de très basse dimensionnalité. Krizhevsky et al. [48] avec le réseau AlexNet ont obtenu une erreur top-5[1] de 15.3% ce qui représente une amélioration significative par rapport aux années précédentes. Le réseau de Zeiler et al.[110] a baissé l'erreur top-5 à 11.7%. Ces derniers ont de plus présenté la méthode de visualisation à travers un second réseau DeconvNet greffé sur chaque couche du réseau qui réalise la tâche. Il est ainsi possible de donner les masques appris par le réseau de classification au DeconvNet afin que celui-ci génère une image. Wu et al. [109] ont ensuite réalisé une méthode pour de l'augmentation des données et ont créé des superordinateurs spécifiques aux algorithmes d'apprentissage profond. Ils ont obtenu une erreur top-5 de 5.33%. La tâche de classification d'images est ici étudiée du point de vue de l'ensemble de données ImageNet et de la compétition ILSVRC qui nécessite de classer une image parmi 1000 classes possibles. Bien entendu, plusieurs autres ensembles de données existent, mais ces quelques réseaux résument bien les avancées significatives. D'ailleurs, en 2017 29 des 38 équipes participantes à cette compétition ont obtenu des erreurs inférieures à 5% et ceux-ci se différencient au centième de pour cent. La compétition a donc changé considérablement pour inclure des notions de langage naturel et d'images 3D. Plus récemment, Mahajan et al. [56] ont utilisé un ensemble de données de 3.5 milliards d'images recueillies d'Instagram pour préentraîner le réseau et obtenir une

[1] L'erreur top-5 définie que le modèle a correctement classé une image donnée si l'étiquette cible est l'une des cinq principales prédictions du modèle.

erreur top-5 de 2.4%.

3.3.2 Détection d'objets dans les images

La tâche de détection d'objets se veut une tâche de localisation puis de classification des objets dans une image. La localisation signifie de délimiter par un cadre la région qui contient un objet donné. Ceci indique donc qu'il est nécessaire de retrouver la taille approximative de l'objet. La classification est réalisée comme à la tâche précédente en donnant une étiquette à la région obtenue. La tâche de détection réfère aussi à plusieurs sous-tâches telles que la détection de visages et des points clés, la détection des piétons et la détection d'un squelette [114]. La difficulté de la tâche de détection d'objets est de suggérer un cadre qui englobe l'objet. Bien entendu, mettre un cadre autour de toute l'image engloberait nécessairement l'objet recherché, mais serait de faible intérêt scientifique. De cette façon, il est pensable d'envoyer une imagette extraite de l'image et de demander à un réseau si cette imagette contient l'objet recherché et rien d'autre. Toutefois, cette méthode naïve requiert de tester toutes les sous-images d'une dimension donnée, et ce pour toute l'image.

Tel qu'indiqué précédemment deux grandes approches utilisant les CNN peuvent être utilisées. La première est l'approche basée régions qui effectue d'abord une recherche dans l'image afin de trouver des régions d'intérêts. Le réseau de neurones R-CNN [27] sélectionne d'abord 2000 régions d'intérêts dans l'image avec l'algorithme "Selective Search" [104]. Ces régions sont ensuite données individuellement à un CNN encodeur afin qu'il réalise la classification à l'aide d'un SVM à sa sortie. R-CNN est très naïf puisqu'il suggère 2000 régions avant de décider parmi eux. Les auteurs de SPP-net [38] tentent de régler le problème de lenteur qui surgit de valider 2000 régions. En exécutant un CNN sur l'image d'entrée, ils obtiennent une carte de caractéristique valable sur toute l'image. Ensuite, le selective search est exécuté, mais puisque la carte de caractéristique est valide partout, la classe d'une certaine région est automatiquement

résolue. Fast-RCNN [26] est un réseau qui combine les idées des deux réseaux précédents. L'image est fournie à un réseau et une carte de caractérisque est générée pour toute l'image. Puis, une couche de régions d'intérêt parcourt la carte pour trouver les cadres et détecter les objets. L'avantage principal est que le réseau peut maintenant être entraîné au complet puisque le gradient peut passer de la cible vers l'entrée. Un autre jalon important est le réseau Faster-R-CNN [77] qui utilise des "Region Proposal Network (RPN)" qui sont des réseaux mieux adaptés pour générer les régions désirées à partir de la carte de caractéristique. Finalement, Mask R-CNN [37] est un réseau qui ajoute une tâche au réseau précédent. En effet, pour chaque région proposée, un masque de l'objet de la classe dominante est calculé. En d'autres mots, ce masque est une segmenation obtenue comme sous-produit. Ceci offre une avenue de coût supplémentaire et permet une amélioration des performances à travers d'un apprentissage multitâches.

La seconde approche est d'utiliser un FCN. Avec les RCNN, les méthodes suggèrent d'abord des régions puis ces régions sont classifiées ou rejetées selon leurs valeurs. Avec YOLO [73], l'image en entrée est divisée en superpixels ce qui diminue grandement le nombre d'imagettes rectangulaires possible. Toutes les imagettes sont envoyées une à une au réseau et un seuillage est effectué afin de sélectionner que les cadres à la confiance la plus grande. Ce réseau a donc une faiblesse importante quant aux objets plus petits. Liu et al. ont proposé le SSD [53], qui est un réseau de neurones qui se place en aval d'un autre réseau de neurones qui va lui servir d'encodeur, dans le cas de l'article, le réseau encodeur est VGG16. SDD ajoute deux fonctions de coût de prédiction l'une sur la position du cadre et l'autre sur la forme du cadre. Ainsi, lors de l'entraînement les cadres vont tendre vers les cadres cibles. Ceci est d'ailleurs ajouté dans les successeurs YOLOv2 [74] et YOLOv3 [75].

3.3.3 Segmentation sémantique d'images

Une des tâches difficiles en vision par ordinateur est la segmentation sémantique des images en composantes. Elle est décrite comme étant différente des tâches de classification et de détection d'objets puisqu'il n'est pas nécessaire de connaître au préalable les concepts visuels et donc que l'espace de la distribution de la solution est beaucoup plus grand [35]. Pour une image donnée, le réseau de segmentation doit retourner en sortie quels pixels sont ensembles dans un sens sémantique. En d'autre mots, quels pixels sont du même objet. Présentement, il est incertain comment le système visuel humain fonctionne pour segmenter efficacement les images perçues. Bien entendu, le cerveau dispose d'information de profondeur, mais le tout nécessite tout de même une compréhension approfondie du monde et de chaque objets qui le compose de façon à identifier ce qui est commun de ce qui est distinct [34]. Les algorithmes traditionnels de vision par ordinateur approchent habituellement le problème de la segmentation à travers la détection de contours et les opérations effectuées sur ceux-ci [108]. L'utilisation des CNNs sur cette tâche a mené à des résultats intéressants notamment dus au domaine de la navigation autonome et simplement parce que les réseaux de segmentation sont maintenant utilisés en amont dans plusieurs types de systèmes de vision par ordinateur [35].

Les FCNs ont comme mode de fonctionnement de prendre une image en entrée et de retourner en sortie une image binaire dans le cas de segmentation à une classe ou une série de n images binaires ayant sur chacune d'elle un groupe de pixel allumé pour marquer le groupe de pixel qui appartiennent au même objet [34]. Long et al. [54] décrivent leur réseau de segmentation comme un réseau de classification pixel à pixel. En d'autres mots, les étapes successives de convolutions vont faire une agrégation locale de l'information pour prendre une décision sur la classe d'une région. De cette façon, le résultat obtenu pour un pixel est une classification sur une imagette centrée autour du pixel. Ils notent que plus le réseau est profond moins celui-ci garde les

détails et donc plus la segmentation finale est lisse. Cela se comprend en disant que plus l'imagette est grande, plus il est probable qu'elle contienne de l'information appartenant à deux classes différentes. DeepLabv3+ [8] continue sur cette idée avec un réseau encodeur-décodeur où l'entrée est donnée à la fois à l'encodeur et au décodeur. L'encodeur utilise l'entrée pour identifier grossièrement les régions et le décodeur utilise la sortie de l'encodeur en plus de l'image pour identifier les contours détaillés des régions. Ce réseau à la fois parallèle et en série permet une segmentation puissante et est l'état de l'art sur les ensembles de données PASCAL VOC 2012 [17] et Cityscapes [13].

Chapitre 4

ESTIMATION DE LA PROFONDEUR À PARTIR D'UNE SEULE IMAGE

Résoudre la profondeur à partir d'images est un problème classique en vision par ordinateur qui se répercute dans le domaine de la robotique aujourd'hui avec les avancées en navigation autonome. Ce problème est habituellement approché en vision par ordinateur avec des méthodes comme la stéréo ou du repérage de points-clés. Autrement dit, plusieurs images sont requises pour résoudre la profondeur. De ce fait, résoudre la profondeur dans une unique image est une tâche à priori impossible. Toutefois, avec de l'information à priori il est certainement possible d'effectuer des raisonnements logiques qui permettent de l'estimer.

Le présent chapitre vise à s'attarder au problème de l'estimation de la profondeur à partir d'une seule image dans des scènes non contraintes. Les algorithmes par apprentissages profonds sont très puissants et ont permis d'apprendre les relations statistiques entre les images et leur carte de profondeur. Ces méthodes requièrent par contre des données qui sont sujettes à des biais voir même des erreurs lors de leur acquisition qui rendent la mise en application pratique de ces algorithmes difficile. En utilisant des réseaux de neurones comme détecteur de jonctions-T, des indices de profondeurs forts, il est possible de réaliser un système d'équations et de le résoudre par méthode de flot maximal.

4.1 Approches par apprentissage profond

L'apprentissage machine qui prédate l'apprentissage profond s'appuyait entre autres sur les champs aléatoires de Markov. Ceux-ci sont adéquats pour répondre aux problématiques qui nécessitent d'inférer à partir d'information contextuelle et non seulement

d'information locale. Ceci caractérise le domaine de la vision par ordinateur en ce qui a trait à la compréhension contextuelle [1]. Les termes "information contextuelle" sont d'ailleurs définis comme de l'information qui ne peut être raisonnée à partir d'une imagette. Bien qu'il existe des indices sur la profondeur dans une imagette, ces indices sont insuffisants à eux seuls et nécessitent une accumulation de l'information sur une échelle globale.

Saxena et al. [86] sont parmi les premiers à s'attaquer à l'estimation de la profondeur à partir d'une unique image en utilisant des techniques d'apprentissage machine. Leur approche est d'utiliser un scanner 3-D SICK laser pour se générer une base de données d'images et de cartes de profondeur. Ainsi, avec un champ aléatoire de Markov, il est possible de modéliser la distribution a posteriori des profondeurs en fonction des caractéristiques d'une image d'entrée. Ces caractéristiques sont construites à partir de convolutions avec divers filtres et visent à représenter les variations dans la texture, les occultations, la mise au point de la caméra et la taille des objets.

4.1.1 Réseaux de neurones complètement convolutifs (FCN)

La tâche d'estimer la profondeur dans une unique image en est une qui se prête bien à l'utilisation de connaissances à priori. Un exemple concret est une caméra de recul sur une voiture. Il est raisonnable d'estimer la distance par rapport à la route à partir d'une mesure similaire prise en laboratoire. En effet, la caméra ne bouge pas et demeure à une distance quasi constante de la route. Même des changements importants dans la route affecteraient peu sa capacité d'exercer sa fonction. Pour des scènes plus variées, la situation est plus complexe et ne peu être contournée par des calibrages en laboratoire. Avec l'arrivée des réseaux de neurones profonds qui peuvent représenter des distributions et des caractéristiques complexes, le problème de l'estimation de la profondeur à partir d'une seule image devient plus accessible.

Notamment, Roy et al. [80] ont créé un "neural regression forest" qui combine la

puissance des réseaux de neurones avec les forêts d'arbres décisionnels. Ils permettent à un réseau convolutif d'apprendre les meilleures caractéristiques des images et utilisent ensuite un classifieur CART [3], des arbres décisionnels pour la classification. Ceci met à profit les fortes capacités des réseaux de neurones de modéliser la distribution des caractéristiques de l'image. Toutefois, ceci n'est pas encore un modèle complètement convolutif. Kuznietsov et al. [4] sont de ceux qui s'attaquent à ce problème en utilisant un réseau de neurones complètement convolutif. Ils permettent donc au réseau de neurones de représenter directement la distribution des profondeurs en fonction de l'image d'entrée.

Kuznietsov et al. [49] présentent toutefois une lacune des méthodes d'estimation de la profondeur. Ils mentionnent que les résultats des scanneurs laser sont bruités, un bruit qui sera appris par le réseau de neurones et que le scanner requiert un calibrage sans quoi la carte de profondeur cible est faussée. De plus, le résultat d'un scanneur laser est beaucoup moins dense que ses images correspondantes ce qui ne permet pas de capturer les détails dans la carte de profondeur générée. Ils compensent les faiblesses dans l'ensemble de données avec une fonction de coût supplémentaire qui est particulière. Simplement, ils utilisent l'image d'entrée pour générer la carte de profondeur avec un réseau de neurones. Ensuite, la carte de profondeur est transformée de façon à engendrer l'image complémentaire stéréo avec l'image d'entrée. Un coût est calculé entre cette image générée et la vraie image de la paire stéréo. L'ensemble de données KITTI offre des paires d'images stéréo et les cartes de profondeur acquises par scanneur laser leur permettant donc d'utiliser cette méthode. Godart et al. [28] poursuivent d'ailleurs cette approche en se débarrassant de la partie supervisée de l'entraînement soit la carte de profondeur générée par un scanner 3D laser. Ils entraînent un réseau de neurones qui prédit uniquement l'image paire stéréo. Durant l'entraînement, ils disposent des images gauches et droites. Ainsi, pour une image gauche donnée en entrée, le réseau est entraîné à générer l'image droite. Si les données de calibrage sont connues, le problème d'estimation de profondeur est une simple triangulation avec deux

images stéréo. Pour des raisons évidentes, ceci ne fonctionne que sur l'ensemble de données KITTI avec une séparation de caméra précise. Les auteurs raisonnent que le réseau de neurones apprend implicitement à résoudre la profondeur lorsqu'il apprend à générer l'image de droite.

Dans une autre approche, se trouvent les réseaux de neurones qui sont entraînés de manière complètement supervisée. Fu et al. [23] utilisent une architecture simple qui remplace les couches de convolution traditionnelles par des convolutions dilatées de manière à couvrir une surface plus grande avec chaque masque et ainsi recueillir de l'information de façon plus globale. Leur encodeur prend une image complète en entrée et retourne en sortie une carte de profondeur. Toutefois, la nature même des fonctions de coûts tend à rendre les résultats des cartes de profondeur floues. Ramamonjisoa et al. [72] améliorent donc les résultats en corrigeant cette problématique. Ils affirment que de manière concrète la profondeur aux bordures des objets est discontinue de façon nette. Ils optimisent donc pour que la profondeur soit égale de chaque côté des contours d'occultations. Ils montrent de meilleures estimations le long des contours sans dégrader la précision du reste de l'estimation.

4.1.2 Les limitations de la programmation basée données

Les dernières années ont permis un progrès important sur cette tâche grâce aux approches par apprentissage profond et aux grands ensembles de données. Toutefois, Chen et al. [9] mentionnent que l'estimation de profondeur à partir d'une seule image demeure un problème. Ils expliquent que la raison est que l'estimation de la profondeur doit être faite sur une image quelconque. Les ensembles de données ne sont pas des images quelconques et les réseaux de neurones s'appuient fortement sur les particularités mêmes de l'ensemble de données pour résoudre la tâche. Ceci rend le réseau entraîné non portable à une autre caméra et donc limite grandement l'impact de la technologie.

Un survol des ensembles de données qui permettent de résoudre cette tâche montre

d'emblée leurs limitations. L'ensemble de données NYU depth [90] est un ensemble de scènes intérieures principalement des chambres à coucher, des salles de bains et des bureaux. L'ensemble de données Make3D [87] est un ensemble de scènes extérieures et KITTI[25] est un ensemble de données prise sur les routes par une caméra fixée à une voiture. Ces trois ensembles de données représentent la majeure partie de ce qui est utilisé par les publications pour l'estimation de la profondeur à partir d'une seule image. Bien que chacun présente très peu de variétés internes dans les scènes et les objets présents dans chacune d'elles, c'est sur ces ensembles que se décide l'état de l'art. Peu de publications s'attardent à l'entrainement sur de multiples ensembles de données puisque ceci entraîne une baisse des performance sur chaque ensemble de donnée respectif. De plus, chaque ensemble de données se veut une différente sous-tâche de l'estimation de la profondeur. Une critique qui revient est que la très grande majorité des images dans ces ensembles de données n'ont pas d'humains dans la scène. Pour un réseau de neurones entraîné sur un unique ensemble de données, il est très difficile de dire que celui-ci va généraliser à des images différentes.

La tâche même d'estimer la profondeur à partir d'une unique image est une tâche mal contrainte. Par exemple, une augmentation de la profondeur d'un objet peut être compensée par une augmentation de sa taille et ainsi conserver les mêmes dimensions après la projection [6]. De plus, les images capturées et les cartes de profondeur sont sujettes à des biais importants. Dans une analyse des ensembles de données réalisée par Chen et al. [9], ils s'aperçoivent qu'un point situé dans le bas de l'image est habituellement plus proche qu'un point plus haut dans l'image. En prenant deux points au hasard dans l'image, le point le plus bas est le point le plus proche 85.8% du temps. Ce biais est d'ailleurs confirmé par la figure 4.1 qui montre des profondeurs croissantes à mesure que l'on monte dans l'image. Ceci est aussi confirmé par la figure 4.2 qui montre qu'en renversant l'image d'entrée pour un réseau, on n'inverse pas la distribution des profondeurs dans l'image de sortie. Ceci mène donc à des aberrations de profondeur. De mêmes façons, Chen et al. [9] indiquent que le point le plus près du centre horizon-

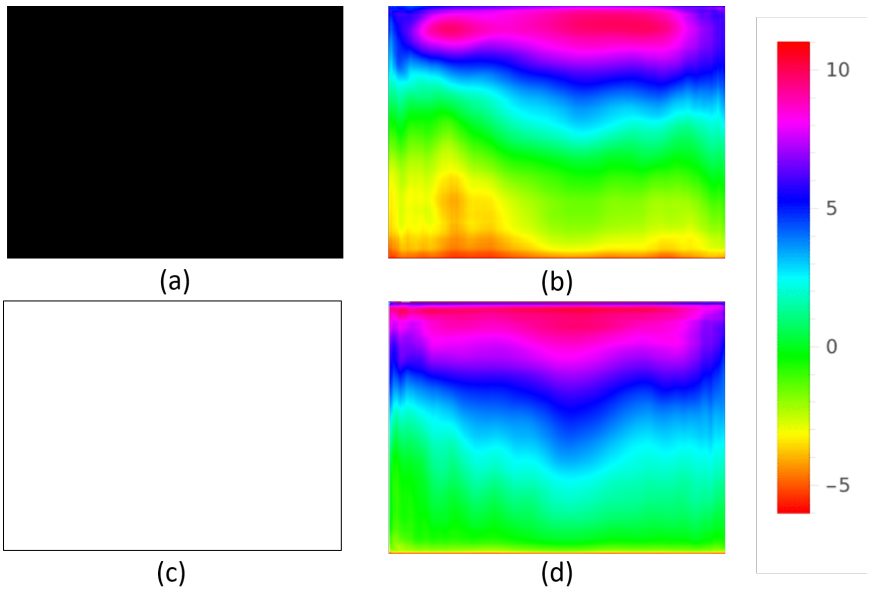


Figure 4.1: Carte de profondeur retournées pour des images d'entrée constantes. a) et c) sont des images complètement noire et complètements blanche respectivement. b) et d) sont les cartes de profondeurs retournées par le réseau [9] pour a) et c) respectivement. Des profondeurs différentes sont hallucinées.

talement est le point le plus proche 71.4% du temps. Ceci engendre une distribution des profondeurs très prévisible.

Lors de tests supplémentaires, un biais important est observé dans les réseaux entraînés. Le réseau entraîné de Chen et al. [9] qui eux-mêmes reconnaissent la présence des biais donne les résultats suivants lorsqu'on lui donne des entrées constantes à la figure 4.1. Dans le cas d'une image d'entrée constante, le réseau invente des profondeurs qui sont différentes d'une constante a) à l'autre c). La tâche d'estimation de la profondeur à partir d'une seule image est mal définie et on propose qu'une tâche mieux définie soit de résoudre l'ordonnancement des objets selon leur profondeur.

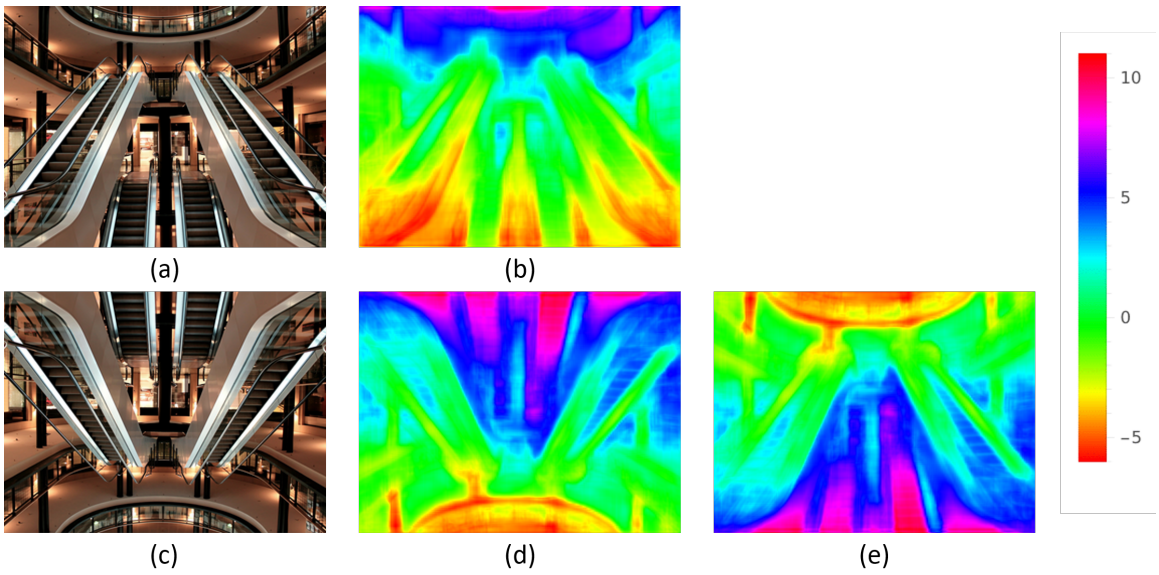


Figure 4.2: Carte de profondeur retournées pour des images d'entrée miroir vertical. a) est une image d'un escalier roulant. b) est la carte de profondeur retournée par le réseau [9]. c) est l'image miroir verticale de a). d) est la carte de profondeur retournée par le réseau [9]. e) est l'image miroir verticale de d). Des profondeurs différentes sont perçu entre b) et e).

4.2 Les jonctions-T

Dans l'objectif de retrouver l'ordonnancement des objets selon leur profondeur et bien entendu d'être invariant à l'ensemble de données d'apprentissage et ainsi de ses biais, il a été décidé de s'attarder à des indices de profondeur forts. Le système visuel humain est capable de percevoir la profondeur même à partir d'une seule image grâce à des signaux de profondeur [76]. Les indices de profondeurs sont d'ailleurs documentés par les peintres qui les utilisent pour ajouter du réalisme et tromper l'oeil humain. Les jonctions-T sont sélectionnés dans ce travail pour leurs caractères fort et local qui les rendent invariants aux différentes scènes.

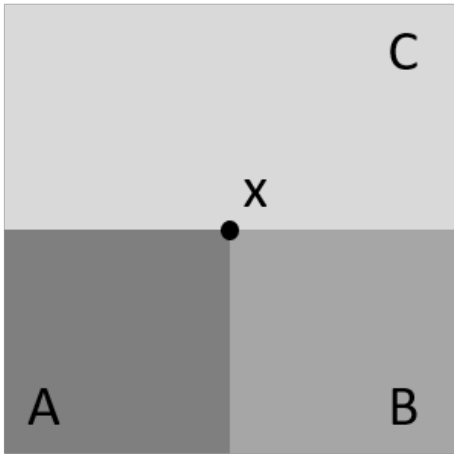


Figure 4.3: Jonction-T engendrée par les ensembles A,B et C. Le point x est l'endroit ou se situe la jonction-T. C occulte les ensembles A et B.

4.2.1 Une relation de causalité

Les indices de profondeur qui sont dits de bas niveaux signifient qu'ils ne requierent pas une analyse de l'image en entier pour pouvoir en déduire une information sur la profondeur. En d'autres mots, uniquement un petit groupe de pixel local est utilisé. Un exemple de ceci est le niveau de flou, si localement un objet est flou, c'est qu'il est en arrière plan puisque la mise au point est habituellement réalisé sur le premier plan. Cet exemple demeure toutefois limité par la situation, il existe plusieurs images où la mise au point est réalisé sur un objet qui n'est pas à l'avant-plan. La jonction-T est un indice local qui, quant à elle, représente une relation de causalité et un indice fort de profondeur qui est invariant à la scène.

Caldero et al. [6] décrivent la jonction-T comme une configuration qui se forme lorsque deux ensembles sont occultés par un troisième. L'interprétation perceptuelle est que l'ensemble qui possède le segment supérieur horizontal du "T" est devant les deux autres ensembles formant le segment vertical du "T". Dans la figure 4.3, la réponse perceptuelle déclenchée par la jonction-T indique que l'ensemble C occulte les ensembles

A et B, étant relativement plus proche que les deux autres. Il est important de noter que l'ordre de profondeur entre les ensembles A et B ne peut être établi.

4.2.2 Profondeur relative n'utilisant que les jonctions-T

Les méthodes pour retrouver la profondeur à partir d'une seule image sont catégorisées en deux soient les méthodes supervisées qui visent à représenter la distribution de probabilités de l'ensemble d'entrée et les méthodes basées sur les indices de profondeur [78] qui sont inversement non biaisés par les ensembles de données. Plusieurs approches basées sur les indices de profondeurs ont été réalisées notamment certaines qui n'utilisaient que les jonctions-T. Dimiccoli et al. [14] ajoutent l'information recueillie par un détecteur de jonctions-T à un algorithme de regroupement de régions pour produire un algorithme qui réalise à la fois la segmentation et l'ordonnancement des profondeurs par résolution de modèles graphiques.

Palou et al. [63] s'attardent par la suite à une version limitée des travaux précédents. Ils réalisent un détecteur de jonctions-T en plus d'un détecteur de convexité pour obtenir une série de relations entre les différents objets dans la scène. Ceci permet donc de résoudre l'ordonnancement sans utiliser aucune information à priori, mais plutôt seulement la détection des quelques points-clés. Des travaux subséquents par Rezaeirowshan et al. [78] combinent la détection des jonctions-T à un modèle de la distribution à priori. Ils utilisent entre autres des biais tels que la position verticale dans l'image pour résoudre les ambiguïtés. Le tout est résolu dans un champ aléatoire de Markov. Il est important toutefois de comprendre que dans ces deux méthodes, les segmentations sont fournies et donc que l'image d'entrée est une image segmentée. Le problème d'ordonnancement est réalisé sur quelques gros morceaux d'images et non-pixel à pixel. C'est pourquoi si peu de points-clés peuvent résoudre l'ordonnancement de tous les pixels.

Les approches par apprentissage profond sont très robustes aux variations dans

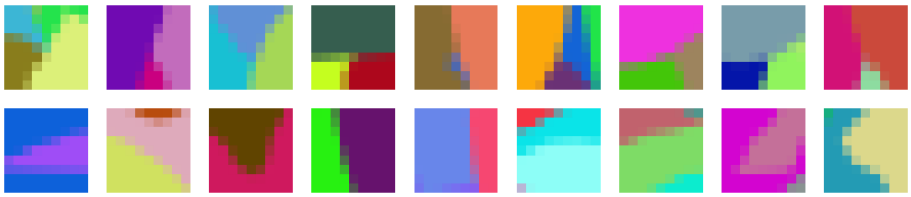


Figure 4.4: Exemples d'imagettes utilisées pour entraîner les réseaux de neurones convolutifs. En haut sont des imagettes avec des jonctions-T. En bas sont des imagettes sans jonctions-T

l'entrée. Inversement, elles tendent à apprendre le biais inhérent aux données d'apprentissage. Les approches par indices d'occultation sont invariantes à la scène et aux types d'images, mais sont moins puissantes. Pour tirer parti des avantages respectifs des approches, la méthode proposée utilise un détecteur d'indices d'occultation, notamment les jonctions-T, par apprentissage profond et une méthode par champ aléatoire de Markov pour résoudre globalement la profondeur.

4.2.3 Méthode par apprentissage profond

L'entraînement du réseau de neurones convolutif est réalisé de façon supervisée. Ceci requiert un ensemble de données qui est hautement représentatif tout en étant le moins biaisé possible. Dans cette optique une approche différente est donc suggérée soit d'utiliser des imagettes ou des "patchs" tels que présenté en figure 4.4. Ainsi, le réseau est entraîné, validé et testé sur des imagettes de dimensions 11x11 pixels et ne voit donc jamais l'ensemble de l'image. Ces patchs sont générés de façon synthétique en empilant des formes géométriques simples comme des rectangles, des ellipses et des triangles. Puisque le tout est synthétique, il est trivial de fabriquer un ensemble de patchs qui contiennent des jonctions-T et un ensemble qui n'en contient pas. De plus, pour chaque imagette contenant une jonction-T, il est possible d'avoir l'ordonnancement des formes et donc de connaître leur profondeur dans la scène.

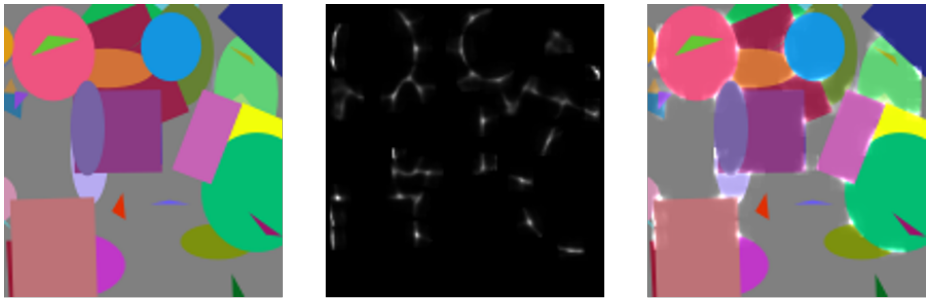


Figure 4.5: Détection des jonctions-T dans une image synthétique. a) Image synthétique en entrée. b) Somme des probabilités par pixel qu'une jonction-T existe dans l'imagette centrée autour du pixel. c) Superposition des images a) et b)

Un premier réseau de neurones complètement convolutif est entraîné afin d'effectuer une tâche de classification, soit d'indiquer la probabilité qu'une imagette contienne ou non une jonction-T. Ce réseau est construit et entraîné sur les imagettes avec et sans jonctions-T. Il est ensuite possible d'étendre ce réseau sur une image entière en découpant l'image en imagettes et en donnant celles-ci une à une au réseau. Un tel résultat de classification est présenté en figure 4.5 où les jonctions-T sont correctement identifiées.

4.2.4 Relations de profondeurs relatives

Ensuite, un second réseau de neurones est entraîné pour résoudre la profondeur relative dans une imagette. Le réseau est entraîné à retourner la probabilité qu'un pixel se trouve en avant plutôt qu'en arrière. Par simple extension de l'ensemble de données précédent, un second ensemble de données contenant que des jonctions-T est généré. Pour chaque imagette, la cible correspondante est la région, en pixels, qui est en avant. Les résultats de ce réseau sont présentés à la figure 4.6 qui présente une excellente identification de la région d'occultation.

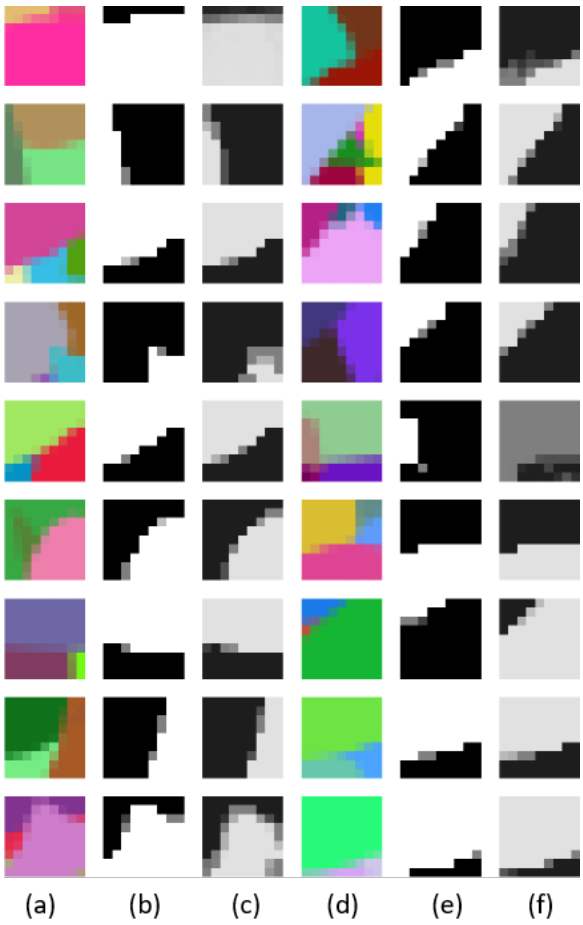


Figure 4.6: Exemples d'imagettes et leur résultat pour la détection de la région d'occultation. Les colonnes a) et d) sont des séries d'imagettes en entrée. b) et e) sont les cibles. c) et f) sont les résultats. Les colonnes a), b), c) réfère à l'ensemble d'entraînement, d) e) et f) à l'ensemble de test.

4.3 Profondeur relative globale

Pour chaque imagette, il est donc possible d'identifier si celle-ci contient une jonction-T et, ci c'est le cas, quelle est la région en avant, la région d'occultation. Cette relation de région en avant versus région en arrière peut être exprimée par un ensemble de relations pixel à pixel. La formulation $P(x)$ pour un pixel x indique la probabilité que

ce pixel soit en avant. Plus $P(x)$ est grand, plus il est probable que le pixel x se situe devant tous les autres. Inversement, plus $P(x)$ est petit plus il est probable que les autres se situent devant lui. Une formulation est posée afin de formuler une probabilité qu'un pixel A soit devant un autre B :

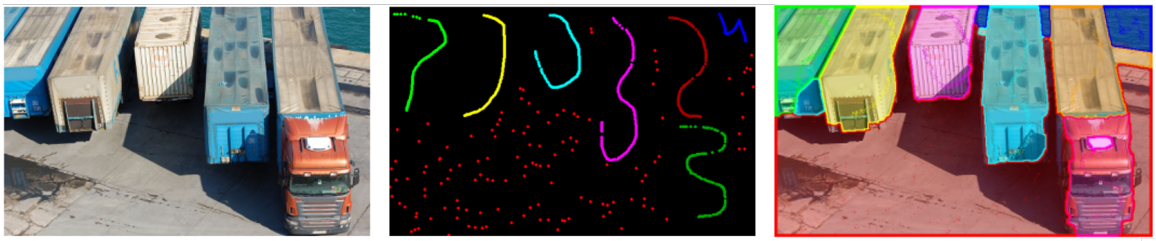
$$P(A\neg B) = P(A)(1 - P(B)) \tag{4.1}$$

Avec les relations de profondeurs relatives pixel à pixel, il est possible d'établir un système de programmation linéaire qui en théorie peut être résolue par le théorème du simplex. Ceci n'est en pratique pas le cas, puisqu'il reste deux problèmes qui doivent être résolues: la propagation des relations de profondeur et la présence de contradictions dans l'ensemble des relations qui invalide les méthodes linéaires.

4.3.1 Propagation des relations

Une fois les relations avant-arrière définies localement pixel à pixel, il est nécessaire de définir des relations d'égalité de profondeur. En d'autres mots, les pixels appartenant au même objet devraient partager la même profondeur. Pour réaliser ceci, le plus simple est d'avoir en entrée une segmentation préalablement réalisée. Cette approche repose grandement sur la qualité de la segmentation initiale puisqu'elle définit en totalité les relations d'égalité de profondeur. Pour réaliser la segmentation, il existe plusieurs méthodes. Rezaeirowshan et al. [78] utilisent un outil de segmentation interactif [85] qui permet, à l'aide de points de départ spécifiés grossièrement par l'utilisateur, d'interpréter les points de départ et de les utiliser pour effectuer la segmentation. Un exemple de cet outil est présenté à la figure 4.7.

Il est évident que dans ce cas-ci, une segmentation inefficace mène directement à des résultats d'ordonnancement de profondeur déficients. D'autres outils par apprentissage profond notamment Deeplabv3+, qui est complètement automatique, permettent des segmentations en situations urbaines très efficaces. À noter qu'un réseau entraîné



(a) (b) (c)

Figure 4.7: Résultat de segmentation avec l'outil [85]. a) est l'image en entrée. b) un utilisateur a fourni des points de départ pour huit différentes étiquettes. c) est la segmentation obtenue

par apprentissage profond pour la segmentation est limité par son ensemble de données. Ainsi, avant d'utiliser un réseau de neurones pour une tâche, il est préférable de connaître sur quelles données celui-ci a été entraîné. Dans notre cas présent, on suppose que les segmentations sont préalablement effectuées et qu'elles sont disponibles. Ainsi, les relations d'égalité pixel à pixel sont trivialement résolues en comparant les couleurs.

4.3.2 MaxFlow MinCut

Le graphique engendré parl'ensemble des relations d'égalité de profondeurs et des relations avant-arrières entre tous les pixels de l'image représente le système d'équations à résoudre. Ce graphique peut donc être résolu par les techniques classiques des champs aléatoires de Markov. Li et al. [52] décrivent les Markov Random Field Models (MRF) comme un outil qui permet d'encoder les contraintes contextuelles dans une probabilité a priori. Ceux-ci peuvent être résolus par une variété d'approches, mais notamment dans notre cas par la technique de coupe minimale et ce une fois le graphique transformé en graphique de flot maximal.

Ford et al. [20] citent T. Harris et son célèbre problème de trains. Celui-ci dit : "Prenons l'exemple d'un réseau ferroviaire reliant deux villes par l'intermédiaire d'un

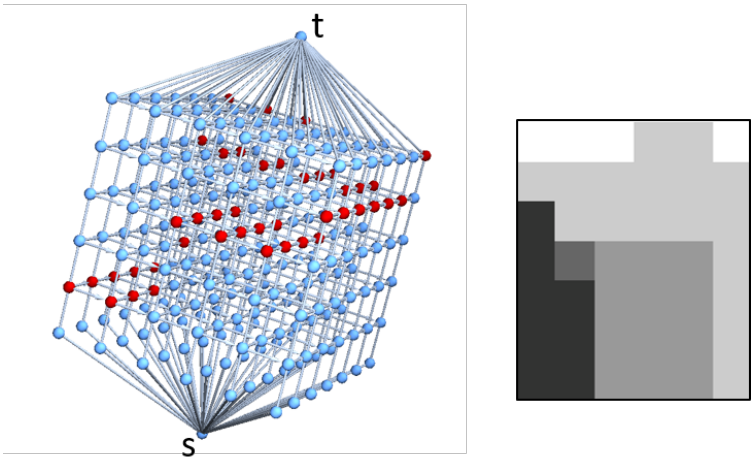


Figure 4.8: Coupe minimale par méthode de flot maximal. a) est un graphique de flot maximal. Les noeuds rouges sont les noeuds de blocage du flot et *s* et *t* sont la source et le drain respectivement. b) est l'image résultante de la coupe minimal

certain nombre de villes, où chaque liaison du réseau se voit attribuer un numéro représentant sa capacité. Dans l'hypothèse d'un état stationnaire, trouvez un débit maximal d'une ville à l'autre." Ceci est le problème de flot maximal qui peut être posé sur un graphique ayant une source et un drain et étant connecté par un ensemble de chaînes interconnectées. Ceux-ci mentionnent la clé pour résoudre ce problème soit qu'une chaîne qui connecte la source au drain ne peut faire passer plus de flots que l'arête la plus contraignante. La coupe minimale dans ce graphique est donc l'ensemble des arêtes qui limite le flot dans les différentes chaînes.

Dans le cas présent la figure 4.8 a) présente l'apparence typique d'un tel graphique ainsi que la solution retrouvée en b). Les noeuds sont l'ensemble des pixels pour l'ensemble des profondeurs qu'ils peuvent prendre. Deux noeuds sont spéciaux soit la source *s* en bas et le drain *t* en haut qui sont chacun reliés à une grande quantité de noeuds. On voit plusieurs plans empilés verticalement qui rappellent encore une

fois toutes les possibilités de profondeur pour l'image. Le MaxFlow MinCut est une méthode qui cherche le minimum d'une fonction de coût posée par le graphique, noeuds et arcs. La figure 4.9 présente en a), b) et c) des arrangements typiques d'arcs entre deux pixels. Les arrêtes verticales qui connectent un même pixel à lui-même sur les différents niveaux de profondeurs ont tous le même poids représentant une probabilité égale pour toutes les profondeurs. Les arêtes horizontales connectent un pixel à ses voisins et représentent le lissage, la probabilité qu'un pixel soit de profondeur égale à son voisin. Ce lissage peut être bidirectionnel comme à la figure 4.9 a) ou unidirectionnel b) ce qui résulte en des fonctions de coûts aux allures d) et e). À la figure 4.9 c), il y a finalement des arêtes de contraintes diagonales qui connectent les noeuds à différentes profondeurs. Ces arcs sont utilisés pour encoder les relations de profondeurs différentes entre les pixels au niveau des jonctions-T. Ceci résulte en une fonction de coût à l'allure f) qui est non-nulle en zéro puisque deux noeuds ne devraient pas être de la même profondeur.

Le graphique est résolu par la méthode du Push-Relabel [10] qui invite à remplir la totalité des arêtes avec un flot plus grand que le flot qu'ils peuvent prendre et à redistribuer vers le drain les flots trop grands comme des quantités de flots négatifs [29]. Ceci se veut comme une version beaucoup plus rapide de la méthode de résolution classique qui indique d'ajouter tranquillement du flot sur chaque arête de la source vers le drain jusqu'à ce que le graphique soit saturé. Cette méthode permet la résolution de la profondeur pour des images complètes.

4.4 Résultats

Un exemple de résultat obtenu sur des images synthétiques est présenté à la figure 4.10. On peut voir que les formes géométriques les plus proches de la caméra sont représentées par les intensités les plus claires. De plus, on voit qu'une forme ne peut apparaitre que si elle possède une relation de profondeur à travers une jonction-T avec

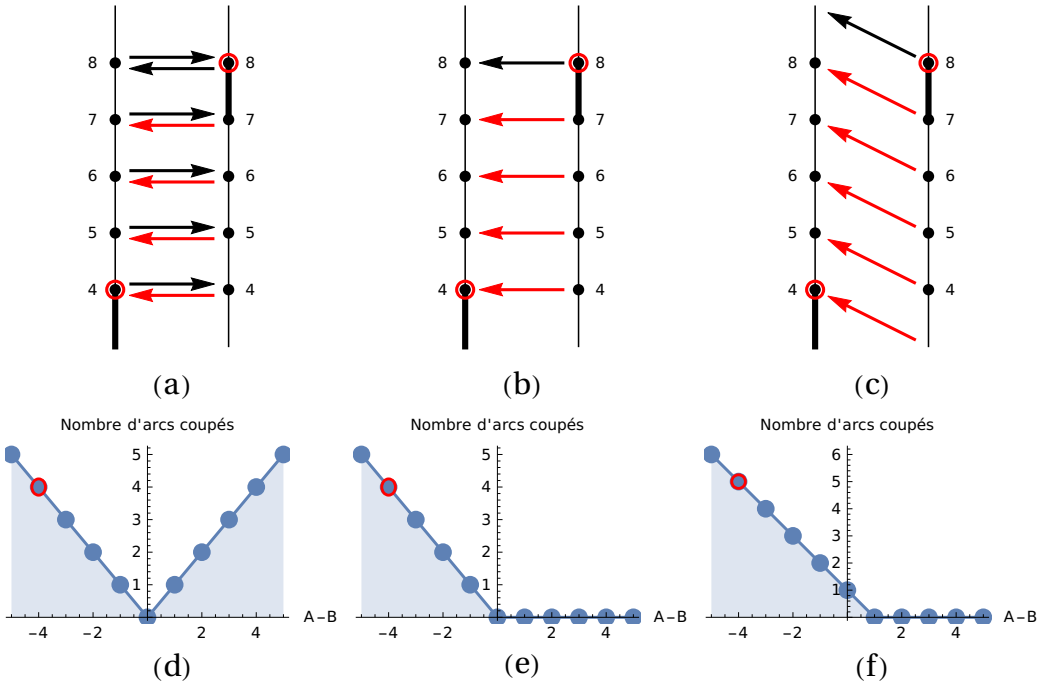


Figure 4.9: Fonction de coût pour deux pixels selon l'arrangement des arcs à couper. a), b) et c) présentent des arrangements typiques d'arcs entre des noeuds qui sont les multiples profondeurs pour deux pixels. d), e) et f) présentent l'allure des fonctions de coût en fonction de la différence de profondeur entre deux noeuds solution pour les arrangement a), b) et c) respectivement. En rouge sont encerclés deux noeuds solutions arbitrairement choisis dans chaque graphique du haut. En rouge sont encerclés le coût correpondant aux noeuds solution.

une autre forme. Ainsi, le triangle vert dans le cercle rose est entièrement disparu puisqu'il n'a pas de relation avec aucune forme.

Afin de tester sur des images naturelles, d'autres résultats ont été effectués sur les images de l'ensemble de données de Zeng et al. [111] qui ont manuellement annoté l'ordonnancement des profondeurs dans les images et ils offrent des segmentations réalisées manuellement. La méthode a donc été testée sur des images qui sont jugées représentatives du problème. Un exemple de résultat est montré à la figure 4.11. En-

Figure 4.10: Carte de profondeur obtenue pour une image synthétique. a) L'image en entrée. b) Carte de profondeur obtenue.

core une fois, les régions les plus près de la caméra sont représentées par les intensités les plus claires. Il est important de noter qu'il est impossible d'établir une relation de profondeur entre les deux régions arrières dans une jonction-T. Ainsi, l'algorithme de coupe minimale peut décider de mettre ces deux régions a des profondeurs égales ou différentes dans la mesure que la région d'occultation soit la région la plus proche. La figure 4.11 d) montre le résultat où le vase de gauche est placé à la même profondeur que le fond. En effet, aucune jonction-T n'est présente pour mettre en relation ce vase devant le fond. Inversement, le vase de droite est placé derrière le fond. Ceci est du au fait que le vase ne possède aucune jonction-T et donc aucune relation de profondeur avec le fond. Dans cette image, toutes les relations sont donc correctement identifiées.

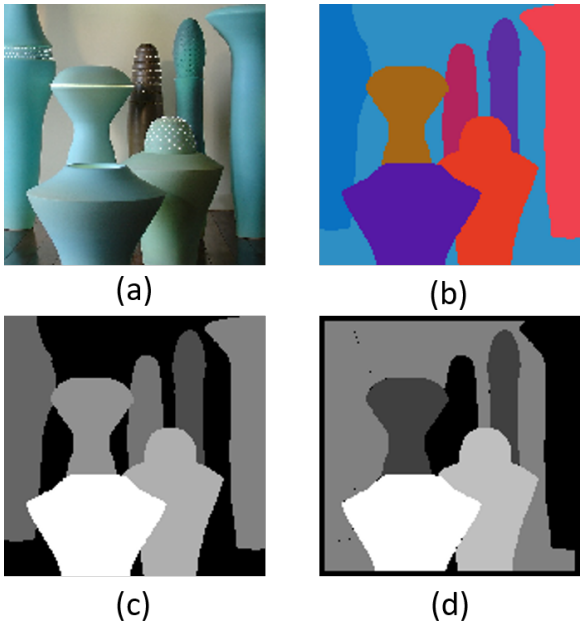


Figure 4.11: Carte de profondeur obtenue pour une image réelle. a) L'image en originale. b) Segmentation en comme image en entrée c) Carte de profondeur cible d) Carte de profondeur obtenue.

Chapitre 5

VERS LA NAVIGATION AUTONOME

La recherche dans le domaine des véhicules autonomes a connu d'importants progrès ces dernières années, non seulement dans le milieu universitaire, mais aussi dans l'industrie. Ceci est principalement dû à la réduction des coûts des senseurs, des avancés dans le domaine de l'intelligence artificielle et du financement croissant par les grands manufacturiers automobiles [7]. Un véhicule autonome doit être capable de détecter son environnement et de prendre des décisions en temps réel. Dans le domaine automobile, il est défini par Hussain et al. [43] que les voitures sont autonomes en ce sens qu'elles prennent en charge des fonctions telles que la détection de l'environnement, la prise de décisions rapides et opportunes, la navigation sans intervention humaine sur la route. Une voiture autonome se réfère à une voiture contrôlée par ordinateur qui peut se guider, se familiariser avec son environnement, prendre des décisions et fonctionner pleinement sans aucune interaction humaine. Ces récentes avancées technologiques ont de plus entraîné une augmentation de la popularité des drones, également connus sous le nom de "unmaned areal vehicule" (UAV) [46].

Le présent chapitre se veut une présentation du domaine actuel en matière de la navigation autonome. D'abord, ce chapitre présente les technologies et capteurs installés dans les véhicules en plus de faire un recensement des méthodes utilisées pour la navigation autonome des véhicules et des drones. En deuxième temps, ce chapitre explique comment un ensemble de données a été réalisé par l'entremise de la fabrication d'une plateforme de translation et d'un module pan-tilt. Le tout dans l'objectif d'entraîner par apprentissage profond un algorithme de résolution de la profondeur de la scène à partir d'une vidéo monoculaire.

5.1 Technologies et capteurs

Avant même de s'intéresser à la question de la navigation autonome et des systèmes de contrôle, un recensement des technologies sensorielles et des capteurs est réalisé. Les véhicules autonomes peuvent prendre des décisions seulement à partir d'une mesure de l'état de l'environnement à un certain moment. Certains senseurs spécialisés permettent une vue de l'environnement et qui selon le senseur se spécialise sur la précision, la vitesse ou le volume de l'information.

5.1.1 LiDAR (Light Detection and Ranging)

Le LiDAR est une technologie de vision active qui permet de mesurer les distances entre les objets qui constituent l'environnement et le senseur lui-même. Le senseur fonctionne grâce au principe de "time of flight (TOF)" ou la durée de vol. En d'autres mots, le senseur mesure le temps requis à une pulsation de lumière pour que l'impulsion soit réfléchie sur une surface. De cette façon un nuage de points, une carte de profondeur de la scène, peut être réalisé en répétant la mesure pour toutes les directions. Les LiDAR utilisés présentement sur les voitures autonomes offrent comme avantage leur haute précision ceci toutefois à un coût élevé et pour un capteur massif et volumineux [7].

5.1.2 Radar (Radio detection and Ranging)

Le radar est une technologie qui utilise les ondes radio pour mesurer la distance, l'angle et la vitesse d'un objet par rapport au senseur. Les radars hautes fréquences permettent de distinguer entre les objets en temps réel, et ce de façon robuste dans toutes sortes de conditions météo. En opposition au LiDAR, le radar est une technologie beaucoup plus accessible et beaucoup moins coûteuse qui permet en plus de résoudre la vitesse des objets visés. Il est important de mentionner que les systèmes radars sont déjà utilisés par la majorité des systèmes avancés d'aides à la conduite, Advanced Driver Assistance Systems (ADAS), qui permettent d'éviter les collisions et des systèmes de régulateur de

vitesse intelligent sur l'autoroute [7].

5.1.3 Ultrasons

Un senseur à ultrasons est un senseur qui mesure sa distance par rapport à un objet qui lui fait face. Il mesure par TOF le temps requis à son impulsion pour lui revenir et ainsi résoudre sa distance. Ce capteur est le moins coûteux de tous et est utilisé depuis de nombreuses années notamment pour valider les angles morts et avertir le conducteur lors des manoeuvres de stationnement à reculons. Sa grande utilisation par les manufacturiers a engendré beaucoup de développement et l'en a fait l'un des capteurs les plus robustes et les plus précis à courte distance [7].

5.1.4 Camera

Tel que présenté au chapitre 1, la caméra est un système qui mesure passivement la lumière et produit une image, une projection 2D de la scène ciblée. Une fois calibrées, les caméras deviennent des outils de mesure indispensables puisqu'elles peuvent résoudre la profondeur des objets en plus d'avoir la couleur et les textures de la scène. Ce dernier facteur est crucial puisqu'il permet de résoudre les signes routiers tels que les feux de circulation, les panneaux et les tracés de ligne. Les caméras sont peu coûteuses, mais nécessitent une expertise pour les calibrer notamment dans le cas de caméra fisheye. Toutefois, les caméras sont passive soit qu'elle ne génère pas leur propre lumière et ainsi la scène doit être illuminée afin d'obtenir une image. Il est dit par Campbell et al. [7] que les constructeurs de véhicules autonomes s'entendent que les caméras sont une technologie fondamentale pour une navigation entièrement autonome, mais qu'elles ne peuvent être utilisées qu'en combinaison avec des systèmes comme le LiDAR ou le Radar.

5.1.5 IMU (Inertial Measurement Unit)

Un IMU est un senseur électronique qui mesure la force, le moment angulaire et le champ magnétique subit par un corps. Il est habituellement constitué de trois accéléromètres, de trois gyroscopes et de trois magnétomètres, chacun sur des axes orthogonaux. Ce senseur est le plus utilisé dans tout système embarqué puisqu'il permet de résoudre les informations de base sur le véhicule lui-même, soient l'accélération, la direction et la rotation. Ce senseur très peu coûteux souffre par contre d'imprécision dans les mesures puisqu'il est sujet à une dérive inertielle. En effet, pour résoudre la vitesse à partir d'une mesure d'accélération, il est nécessaire d'intégrer les mesures. L'étape d'intégration est très sensible au bruit et tend à accumuler les erreurs dans le temps. Il est encore plus difficile d'estimer la position puisque deux étapes d'intégration sont requises [7].

5.1.6 GPS (Global Positioning System)

Le GPS est un système de navigation basé sur le positionnement par satellite. Simplement, il est possible de résoudre la position sur Terre du capteur dans un rayon de 1 à 3 m si celui-ci est visible par au moins quatre satellites. Le processus de trilatération permet, en connaissant exactement le moment auquel un satellite reçoit le signal, de retrouver la distance exacte entre un satellite et le senseur. Avec quatre distances, la position est résolue par le senseur. Les voitures sont aujourd'hui munies de systèmes GPS. Ceux-ci sont toutefois sensibles aux problèmes d'obstruction et de ce fait fonctionne très mal dans des circonstances de navigation intérieure [7].

5.1.7 Fusion de senseurs

Puisque chaque senseur présente des forces et des faiblesses, il est facile de voir qu'une combinaison de ceux-ci permet une information plus complète de l'environnement. De plus, la combinaison des senseurs permet d'obtenir une information de plus haute qual-

ité que ce qui est obtenu à partir de l'un ou l'autre des senseurs. Prenons notamment le LiDAR qui est habituellement combiné à un système de caméras pour compléter l'information visuelle par une carte de profondeur et ainsi reconstruire la scène en profondeur et texture. Aussi, les IMU sont beaucoup plus performants lorsqu'utilisés conjointement à un GPS qui élimine la dérive et qui résout la position.

5.2 La navigation autonome aujourd'hui

La navigation autonome est un sujet qui préoccupe beaucoup les chercheurs. Cette technologie se veut être celle qui va potentiellement définir la prochaine décennie et changer de nombreuses habitudes humainse en ce qui a trait aux déplacements. Depuis, son apparition au début du 20e siècle, très peu a changé dans le mode de transport qu'est l'automobile. L'arrivée de plateformes spécifiques au développement de véhicule autonome comme le NVIDIA Jetson rend de plus en plus accessible cette technologie. Pour certains, la navigation autonome est une façon de redéfinir les systèmes de livraison et de transport. Il devient de plus en plus envisageable d'avoir des plateformes aériennes autonomes dans les prochaines années.

5.2.1 Navigation autonome dans les voitures

Un contributeur important des voitures autonomes est le programme NavLab de l'université Carnegie Mellon dans les années 1980. On associe toutefois les premiers succès à Mercedes et leur "Prometheus Project" en 1987 qui réussissait à suivre les tracés au sol [43]. Aujourd'hui, une telle technologie est commune et selon un recensement, il existe des projets de voitures autonomes chez Audi, BMW, Toyota, Honda, Kia, Hyundai, Mercedes, Ford, Nissan, Tesla, GM, Volvo, Bosch et Volkswagen[43]. Les fonctionnalités implantées sont le stationnement intelligent, l'avertissement d'incident, le freinage d'urgence et la conduite semi-automatique. Cette évolution vers la voiture intelligente se traduit par l'apparition d'une synergie entre les compagnies en technolo-

gies et les grands manufacturiers de voitures. Certains partenariats plus connus sont le programme Waymo de Google et son partenariat avec Fiat Chrysler et Jaguar. Microsoft de son côté s'est alliée avec Toyota et Volvo pour le développement de voiture autonome. Le vendeur de cartes graphiques NVIDIA s'est de son côté lancé dans la fabrication de modules de navigation autonome qui est aujourd'hui livré dans les voitures Tesla. Finalement d'autres compagnies comme Apple et Uber sont très présentes dans le marché des voitures notamment à titre d'actionnaires [43]. La voiture comme produit technologique a permis la fusion entre les domaines automobile et technologique.

5.2.2 La vision par ordinateur et les ensembles de données

La compagnie Google a commencé son programme de voiture autonome en 2009 et en mars 2016 avait parcouru 1,5 million de miles avec sa flotte de véhicules. Ses véhicules autonomes avaient été impliqués dans 14 accidents de la route dont 13 ont été jugés causés par un autre véhicule. Ceux-ci sont équipés de caméras, de radars, d'un LiDAR et d'un GPS afin de détecter l'ensemble de ce qui constitue la route. Pour sa part, Tesla et son système d'autopilote version 7 utilise huit caméras, douze senseurs à ultrasons et un radar [45]. Ces compagnies reconnaissent l'importance d'avoir une gamme complète de senseurs pour couvrir l'ensemble des modalités de perception de l'environnement. Bien que les caméras sont rarement utilisées pour percevoir la profondeur autour d'un véhicule, celles-ci sont les plus utilisées lorsque vient le temps de prendre une décision. En effet, seules les caméras peuvent interpréter le Code de la route. Dans l'optique d'entraîner des systèmes intelligents, Geiger et al. [25] ont produit un ensemble de données le "KITTI Vision Benchmark for stereo, optical flow, visual odometry/SLAM and 3D object detection". Cet ensemble de donnés a été capturé avec des caméras haute résolution en configuration stéréos et en couleur. Associé à cela se trouve des cartes de profondeur capturées par un scanner laser Velodyne et des métriques GPS et IMU. Ceci résulte en un ensemble de données de 194 paires d'images 1280 par 376

pixels pour l'entraînement et 195 paires pour le test. En 2013, KITTI a été étendu pour couvrir la tâche de détection des lignes [21]. Mattyus et al. [59] ont de plus contribué à l'ensemble de données KITTI en utilisant des images aériennes pour réaliser des cartes de segmentations des trottoirs et des espaces de stationnement en plus de numéroter les différentes voies. L'ensemble des contributions à KITTI en ont fait l'étalon de mesure standard dans la tâche de navigation autonome en vision par ordinateur. D'autres ensembles de données ont été réalisés notamment le "Cityscapes Dataset" par Cordts et al. [13] pour permettre l'entraînement de systèmes de segmentation sémantique dans la scène. Cet ensemble de données compte 5000 images annotées au pixel près et un autre 20 000 images annotées de façon approximative.

5.2.3 La navigation autonome chez les drones

Lorsqu'on pense à la navigation autonome, les véhicules automobiles viennent en premier en tête. Toutefois, le problème de la navigation autonome n'est pas limité qu'aux véhicules routiers, mais s'étend aux appareils en mouvement de tous types. Ceci peut donc prendre en compte les bateaux et les véhicules aériens tout aussi bien que les aspirateurs intelligents de maison. Dans le domaine aérien, les quadricopteurs aussi connues sous le nom de drones sont des machines volantes à hélice qui se déplacent librement dans les airs. Ceux-ci sont hautement contraints à se déplacer en volant et par conséquent sont grandement limités à la masse qu'ils peuvent mettre en mouvement. Ceci signifie que la puissance de calcul embarquée et la gamme des senseurs disponibles pour ces appareils sont très limitées par rapport à une automobile. Par conséquent, les approches à la navigation autonome sont très différentes. Sur les drones, les techniques les moins coûteuses sont celles qui emplois du "structure from motion" [43]. Avec la vidéo provenant d'une caméra haute résolution, il est possible d'effectuer le suivi de certains repères et donc de reconstruire en 3D la scène avec notamment du stéréo par mouvement ou des techniques de SLAM (Localisation et cartographie

simultanées) [46]. Plus récemment, cette technique était employée en combinant le détecteur SURF (Speeded Up Robust Features) et l'algorithme MSER (Maximally Stable Extremal Regions) pour faire la détection et le suivi de points de repère [103]. D'un autre côté, il est possible, spécialement sur les drones de plus grandes tailles, d'installer un senseur LiDAR pour faire l'acquisition d'une carte 3D complète. Ceci est peu fréquent, mais a été testé par Luo et al. [55] afin de retrouver un chemin optimal entre les objets de la scène. Dans leur cas, l'utilisation d'un drone rend le problème plus simple puisque celui-ci n'est pas contraint à demeurer sur des routes ou à respecter un code de signalisation. De ce fait, il est possible de considérer la scène comme statique et l'environnement entier comme l'ensemble de solutions de navigation possibles.

Les méthodes utilisées aujourd'hui en navigation autonome de drones sont toutefois basées sur les caméras et leur fusion avec des senseurs inertiels IMU et GPS [51]. Ceci permet une approche de reconstruction de l'environnement en 3D avec l'algorithme ORB-SLAM [61] et ceci avec uniquement une seule caméra rendant le système complet très léger et portable. Le système "Teach-Repeat-Replan" [24] utilise d'ailleurs une approche semblable pour de la navigation semi-autonome. D'abord, le pilote contrôle le drone pour lui faire exécuter une certaine trajectoire. Lors de cette étape, le drone réalise un SLAM du parcours pour ensuite optimiser la trajectoire. Finalement, le drone est capable de naviguer de façon autonome ce parcours et d'obtenir des performances supérieures au pilote.

5.3 Estimation de la profondeur dans une vidéo

Faisant suite à l'estimation de la profondeur à partir d'une seule image, il est possible de faire l'estimation de la profondeur monoculaire sur un flux vidéo d'une caméra en mouvement. La méthode de résolution est basée sur l'odométrie visuelle et il est ainsi possible de retrouver le mouvement instantané à chaque image de la vidéo. L'odométrie visuelle est une méthode qui permet l'estimation de la pose et la locali-

sation de la caméra et a d'ailleurs été approchée, dans la littérature, par des méthodes d'apprentissage supervisé et des réseaux à convolution sur l'ensemble de données KITTI [106]. L'odométrie visuelle est utilisée en pratique de façon complémentaire dans les systèmes robotiques avec d'autres senseurs notamment les GPS et les IMU. Cette tâche ressemble de très près à la méthode présentée en chapitre 4 puisque la profondeur est résolue indépendamment sur chaque image de la vidéo. La méthode réalisée se veut donc complémentaire et une combinaison des approches serait en théorie bénéfique. Brièvement, la profondeur est résolue directement d'un système d'équations en retrouvant le mouvement instantané (voir equation 5.2).

5.3.1 Plateforme delta

Dans l'objectif de réaliser un ensemble de données le premier défi est de lier une capture vidéo à un déplacement physique. L'utilisation d'un GPS est impossible puisqu'il n'est pas suffisamment précis lorsqu'on parle de petites dimensions comme notre situation présente. Le choix est donc d'utiliser une unité IMU puisque celles-ci sont présentes d'emblée dans les drones au niveau des contrôleurs de stabilité et de mise à niveau. Ainsi les premiers essais ont été faits avec des senseurs à six degrés de liberté, accéléromètre et gyroscope. À chaque instant, on obtient de ces senseurs l'accélération et la vitesse angulaire. De ceci, on déduit la vitesse par un vecteur euclidien dans le monde à travers une étape d'intégration des vecteurs d'accélération et on retrouve la rotation directement en sortie des gyromètres. Dans nos essaies, puisque les IMU sont sujets à des dérives importantes qui rend les mesures peu précises, les résultats obtenus par IMU étaient de faible qualité.

Une plateforme de translation, comme sur les machines-outil à commande numérique, ne présente pas ces problèmes puisque la position de la plateforme est connue de façon exacte. Ainsi, les vecteurs de translation qui en sont dérivés sont plus fiables. Une plateforme de translation par géométrie delta a donc été fabriquée.

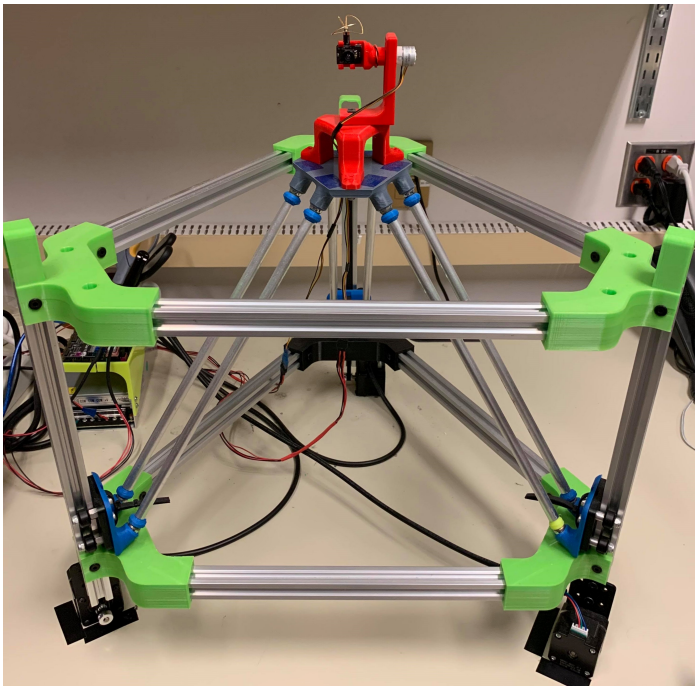


Figure 5.1: Photo de la Plateforme Delta entièrement assemblée et fonctionnelle.

Un robot delta est un robot parallèle constitué de trois bras connectés à des joints homocinétiques sur un effecteur central [64]. Le choix d'un robot delta était à l'origine pour ressembler à une plateforme Stewart qui pour sa part permet des translations sur les trois axes en plus des rotations dans les trois directions. Toutefois, les défis de fabrication pratique ont permis de limiter le choix à une plateforme plus simple à fabriquer. En dehors des tiges en aluminium, les pièces ont toutes été fabriquées et imprimées 3D avec une imprimante 3D et du plastique ABS ou PLA. Les dessins de cotation des diverses pièces sont disponibles en annexe et une photo de la plateforme complètement assemblée est montrée à la figure 5.1. Le logiciel de modélisation de solide Catia a été utilisé pour réaliser les pièces nécessaires à l'assemblage de la plateforme.

De plus, afin de permettre deux degrés de mouvement supplémentaires au mouvement de la caméra un système de "Pan-Tilt" a été fabriqué. Ce système à deux degrés de rotation tient la caméra en place sur l'effecteur de la plateforme delta. Le résultat

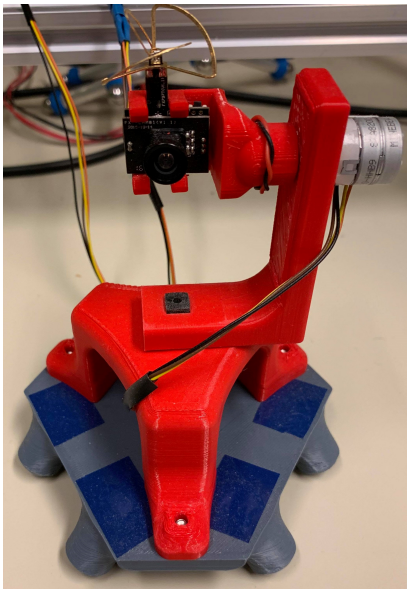


Figure 5.2: Photo du Pan-Tilt en rouge qui tient et met en rotation la caméra. Le Pan-Tilt est assemblé à l'effecteur en gris.

final est nommé plateforme delta qui permet en tout cinq degrés de liberté au niveau de la caméra. De plus, il est possible pour un utilisateur d'ajouter un degré de rotation final en ajoutant une rotation au niveau des images autour de leur centre. Une photo du Pan-Tilt est fournie à la figure 5.2.

Pour contrôler les trois moteurs de la plateforme delta et les deux moteurs du pan-tilt, un MKS Gen v1.4, Printer controller Board /RAMPS 1.4 + Mega 2560 for Arduino, est utilisé. Brièvement, ce circuit imprimé contient un microcontrôleur et permet de contrôler jusqu'à cinq moteurs et trois éléments chauffants. Ce circuit imprimé est vendu dans le cadre de l'assemblage d'une imprimante 3D. Une photo du contrôleur MKS est fournie en figure 5.3. De plus, le microprogramme chargé sur le circuit imprimé est le Marlin Firmware qui se spécialise dans le contrôle des imprimantes 3D et des machines CNC et avec lequel il est possible de communiquer dans un langage G-Code. Nativement, Marlin supporte les imprimantes 3D en configuration delta ce qui rend la tâche relativement simple. Il est toutefois important de noter que des cor-

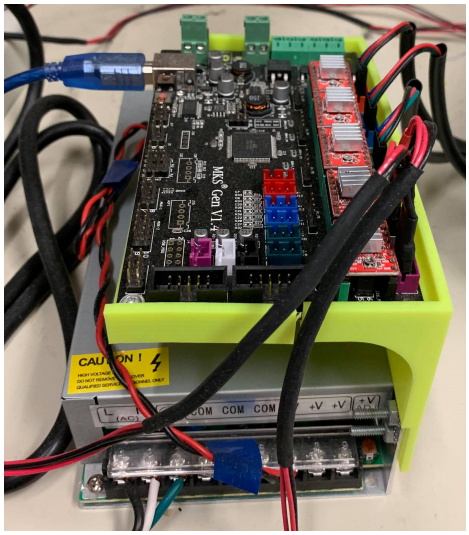


Figure 5.3: Photo du circuit imprimé contrôleur MKS GenV1.4. Il est fixé à la source d'alimentation.

rections importantes aux équations de cinématique delta ont été apportées à Marlin puisque celui-ci comportait des erreurs dans les vitesses de translation. De plus, il a été nécessaire de réaliser un programme afin de contrôler le pan-tilt à travers le circuit imprimé. La caméra utilisée est une caméra de drone Hyperion 600TVL qui envoie son signal par ondes radio. Ce signal est capté par un receveur radio et est converti en une vidéo par une carte de capture. Une photo de la caméra et du receveur est fournie en figure 5.4. Au final, cet appareil permet de capturer une vidéo d'une caméra de drone et de lui associer pour chaque image, son mouvement instantané.

5.3.2 Résolution du mouvement instantané

La particularité du système de la caméra sans-fil de drone, de son recepteur radio et de la carte de capture est que les images résultantes sont très bruitées. Dans le cadre d'un vol de drone, les images capturées changent de ratio signal-bruit et peuvent devenir à l'extrême inutilisables. La figure 5.5 montre des exemples de ces images capturés où l'on distingue que même les images les plus nettes demeurent très bruitées. Ceci est dû

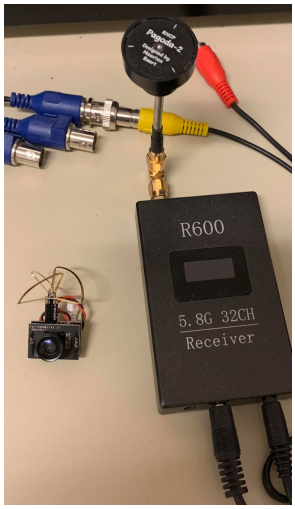
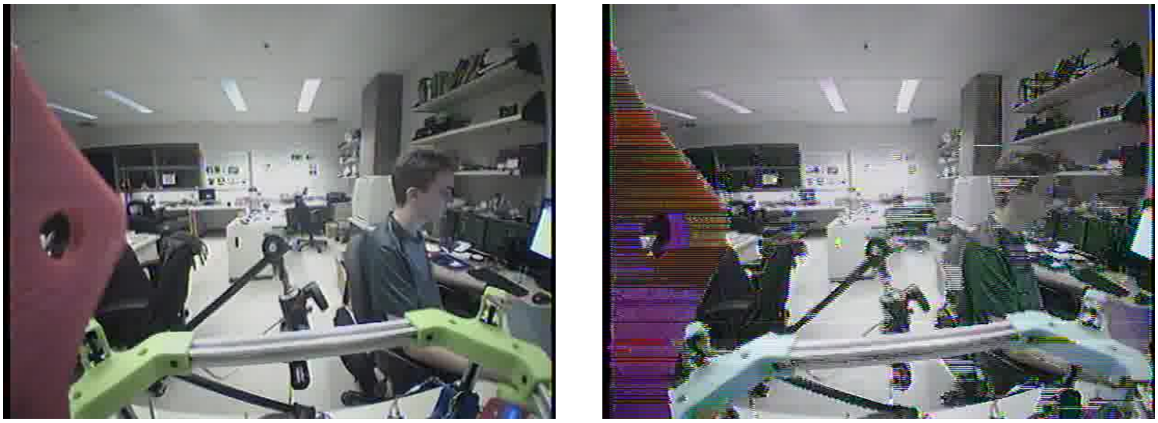


Figure 5.4: Photo de la caméra Hyperion et de son receveur radio. Le receveur est connecté dans une carte de capture dans un ordinateur.

aux pertes dans le signal analogue radio alors que celui-ci parcourt de longues distances ou traverse des obstacles. Dans le cadre du présent travail, les images sont supposées les plus nettes possible dans des conditions idéales.

Du fait que les images sont hautement bruitées, les approches classiques de vision par ordinateurs et notamment de flux optique donnent de mauvais résultats. Une approche par apprentissage profond est donc privilégiée puisque celles-ci permettent de démêler le signal du bruit même dans des situations de ratio signal bruit extrêmement faible [66]. Grâce à la plateforme delta, il est possible par apprentissage profond supervisé d'entraîner un réseau de neurones convolutif à résoudre le mouvement instantané dans un flux vidéo. L'entraînement a été effectué pour des paramètres de vitesses fixes soient une vitesse constante dans toutes les directions en translation de 20mm/s et une vitesse fixe de rotation de 10deg/s. Afin d'éliminer le biais relatif aux basses fréquences lors de l'apprentissage, le gradient horizontal, vertical et temporel est donné en entrée plutôt qu'un ensemble d'images. De cette façon, il est impossible que le réseau de neurones utilise les couleurs pour calculer le mouvement instantané. Il est de plus connu,



(a) (b)

Figure 5.5: Images recueillis de la caméra Hyperion. a) Image nette. b) Image bruitée non désirée.

que le flux optique est visible que sur les contours et sur la texture ce qui jusitife ce choix.

5.3.3 Résolution instantanée de la profondeur

Pour un mouvement donné de caméra perspective rectilinéaire, le phénomène de parallaxe indique que les objets qui ont une norme de gradient plus élevée vont être situés à une profondeur plus faible dans la scène que les objets ayant une norme de gradient plus faible. Ainsi, il est possible de dire que la profondeur d'un objet est liée à son gradient. La caméra du drone a été calibrée de façon centrale par la méthode présentée au chapitre 2 utilisant la lumière structurée. Ceci permet entre autres de retirer toute distorsion dans les images en entrée et de retrouver une caméra perspective rectilinéaire et la carte de correspondance calculée est disponible en figure 5.6.

Cette carte de correspondance permet de retirer toute distorsion de l'image de caméra de drone pour retrouver des images telles que prises par une caméra perspective. Ainsi, pour chaque pixel avec un mouvement connu, il est possible de retrouver

Figure 5.6: Vue 3D de la carte de correspondance de la caméra Hyperion. La direction en hauteur est α avec des isocontours à tous les 5° (45° est en gras). La valeur α est aussi encodée par la couleur (voir légende)

sa profondeur grâce a un système d'équations. Pour une matrice de rotation R où r_x, r_y et r_z sont les rotations autour de l'axe x, y et z respectivement et une matrice de translation T où t_x, t_y et t_z sont les translations en x, y et z respectivement:

$$R = \begin{pmatrix} 1 & -r_z & r_y & 0 \\ r_x r_y & 1 - r_x r_y r_z & -r_x & 0 \\ -r_y + r_x r_z & r_x + r_y r_z & 1 & 0 \end{pmatrix}$$
$$T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.1}$$

Sachant que la disparité est $1/p$ où p est la profondeur, on peut décrire la vitesse V_x et V_y comme suit. La déprojection d'un point (u, v) à la bonne profondeur p auquel on applique la transformation $R \cdot T$ autour du centre de la caméra est ensuite reprojeté dans l'image. La différence entre la position de départ, avant la déprojection-projection, et la position d'arrivée est la vitesse. Le système d'équations qui décrit la vitesse dans l'image est le suivant.

$$
\begin{aligned}
V_x &= \frac{pr_y t_z + p(t_x - r_z t_y) + r_y - r_z v + u}{p(r_x r_z t_x + r_x t_y + r_y r_z t_y - r_y t_x + t_z) + r_x r_z u + r_x v + r_y r_z v - r_y u + 1} - u \\
V_y &= -\frac{r_x(p(r_y r_z t_y - r_y t_x + t_z) + r_y r_z v - r_y u + 1) - p(r_z t_x + t_y) - r_z u - v}{p(r_x r_z t_x + r_x t_y + r_y r_z t_y - r_y t_x + t_z) + r_x r_z u + r_x v + r_y r_z v - r_y u + 1} - v
\end{aligned}
\tag{5.2}
$$

Finalement, par les équations de flux optique, on sait que les gradients horizontaux I_x, verticaux I_y et temporels I_z sont reliés par l'équation suivante.

$$
I_x V_x + I_y V_y = -I_t \tag{5.3}
$$

En isolant la profondeur p dans le système d'équations, on peut retrouver directement pour tout pixel (u, v) aux valeurs de rotation et de translation connues la valeur de la profondeur. Une telle carte de profondeur est présentée en figure 5.7 qui affiche grâce à un code de couleurs les différentes profondeurs dans la scène. Bien que les unités de profondeurs soient des unités arbitraires et non une unité de mesure réelles comme les mètres, il est possible de voir que les éléments proches de la caméra ont une profondeur plus faible que les éléments loin. Ceci permet donc à l'aide d'un simple seuillage d'obtenir un détecteur d'obstacles, en d'autres mots, un détecteur d'objets proches.

Le figure 5.7 présente les résultats qui, de façon qualitative, semble correspondre à la scène perçue. Or, il est difficile d'analyser la validité quantitative métrique de ces résultats dû à l'absence des mesures réelles dans la scène et à l'absence d'une unité métrique sur les mesures. Il est important de mentionner que ces résultats se veulent exploratoires pour démontrer la faisabilité de la méthode. Une erreur visible est que la forme rouge sur la gauche, le support de la caméra, n'a pas de profondeur reconstruite sur sa bordure. Dans la version finale, le support a été modifié afin de ne pas apparaître dans l'image. Des travaux subséquents auront lieu afin de corriger ces lacunes et d'identifier d'autres faiblesses en plus d'évaluer la qualité métrique de l'algorithme d'odométrie visuelle.

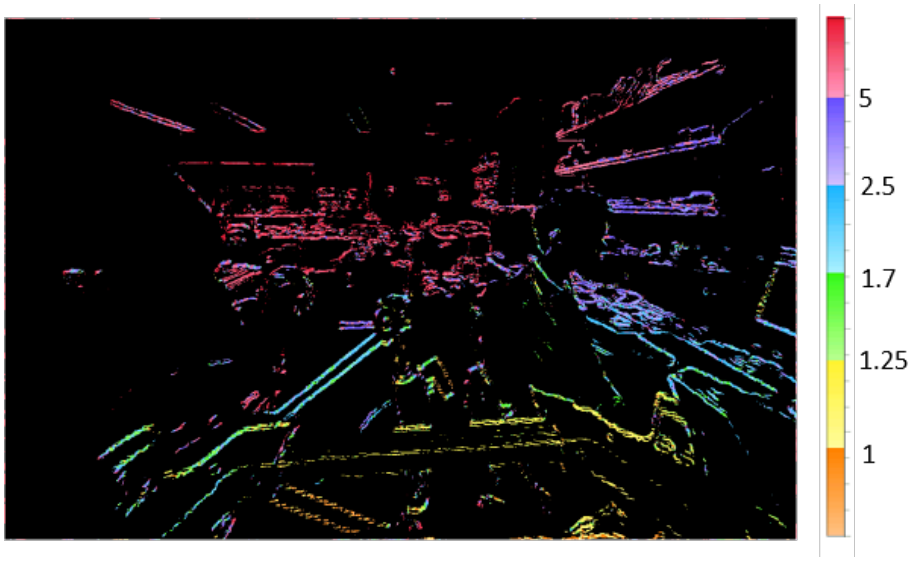


Figure 5.7: Carte de profondeur obtenue par résolution de la profondeur pour chaque pixel de la figure 5.5 a) après un dédistorsionnement par lumière structurée. La carte de profondeur est colorée en fonction de la profondeur en unités (voir légende). En noir sont des pixels sans profondeur.

CONCLUSION

Ce mémoire porte sur les problématiques du calibrage de caméras grand angles et l'estimation de la profondeur à partir d'une unique caméra immobile ou en mouvement. Les résultats sur des caméras synthétiques et virtuelles sont présentés dans les chapitres respectifs pour les méthodes proposées. Les présents travaux visent à régler un problème générique soit la détection d'obstacles pour un drone muni d'une seule caméra.

D'abord, nous proposons une nouvelle méthode de calibrage pour les caméras ayant un très grand angle de vue et qui peut être modélisée par un ensemble de caméras génériques virtuelles centrales. Ceci est démontré en pratique pour les caméras fisheyes qui peuvent avoir un angle de vue allant jusqu'à 280°. Nous démontrons que cette approche permet de modéliser directement des caméras axiales et qu'en théorie, il est possible de calibrer une sphère complète. Cette méthode est basée sur le calibrage planaire à correspondances denses obtenues par lumière structurée en prenant les plans de calibrage eux-mêmes comme les caméras à calibrer. À notre connaissance, c'est l'une des seules méthodes pratiques de calibrage de caméras axiales.

Ensuite, nous proposons une méthode pour estimer la profondeur à partir d'une unique image qui utilise uniquement les jonctions-T comme indices de profondeur. Les méthodes par apprentissage profond sont spécialement susceptibles d'apprendre les biais dans leurs ensembles de données. En utilisant que des indices de profondeurs forts, la méthode est invariante aux divers biais présents dans les ensembles de données. Les relations de profondeur sont obtenues grâce à des réseaux de neurones convolutifs sur des imagettes et sont propagées de façon globale par un graphique de flot maximal. Ce travail vise la compréhension des modèles par apprentissage profond et de leurs ensembles de données et de faire le pont entre le domaine de la vision tridimensionnelle

et l'apprentissage profond.

Finalement, nous proposons une méthode pour l'estimation de la profondeur pour une caméra sur un drone en mouvement libre par apprentissage supervisé en réalisant un nouvel ensemble de données. Ces données sont recueillies par une plateforme de type delta sur lequel est monté un Pan-Tilt permettant cinq degrés de liberté du point de vue de la caméra. Par apprentissage profond et un réseau de neurones convolutif, il est possible d'apprendre le mouvement instantané de la caméra à chaque instant. À l'aide des équations de flux optique et un calibrage de la caméra, il est possible de résoudre directement la profondeur pour chaque image et de seuiller pour distinguer les objets loin des proches.

Le domaine de la navigation autonome en est un qui va bouleverser les opérations humaines dans les prochaines décennies. La vision tridimensionnelle s'inscrit comme la pierre angulaire des véhicules autonomes et de leur interaction avec notre monde. Déjà aujourd'hui, les véhicules sont munis de caméras fisheyes pour offrir des options de caméras de recul ou de surveillances des angles morts. Des méthodes d'estimation de profondeur par des caméras fisheyes sont essentielles pour la suite du progrès en navigation autonome.

RÉFÉRENCES

[1] Vijay Badrinarayanan, Alex Kendall, et Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] Filippo Bergamasco, Luca Cosmo, Andrea Gasparetto, Andrea Albarelli, et Andrea Torsello. Parameter-free lens distortion calibration of central cameras. Dans *Proceedings of the IEEE International Conference on Computer Vision*, pages 3847–3855, 2017.

[3] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[4] Pierre-André Brousseau et Sebastien Roy. Calibration of axial fisheye cameras through generic virtual central models. Dans *Proceedings of the IEEE International Conference on Computer Vision*. Institute of Electrical and Electronics Engineers (IEEE), 2019.

[5] Duane C Brown. Decentering distortion of lenses. *Photogrammetric Engineering and Remote Sensing*, 1966.

[6] Felipe Calderero et Vicent Caselles. Recovering relative depth from low-level features without explicit t-junction detection and interpretation. *International journal of computer vision*, 104(1):38–68, 2013.

[7] Sean Campbell, Niall O'Mahony, Lenka Krpalcova, Daniel Riordan, Joseph Walsh, Aidan Murphy, et Conor Ryan. Sensor technology in autonomous vehicles: a review. Dans *2018 29th Irish Signals and Systems Conference (ISSC)*, pages 1–4. IEEE, 2018.

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, et Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. Dans *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[9] Weifeng Chen, Zhao Fu, Dawei Yang, et Jia Deng. Single-image depth perception in the wild. Dans *Advances in neural information processing systems*, pages 730–738, 2016.

[10] Boris V Cherkassky et Andrew V Goldberg. On implementing the push—relabel method for the maximum flow problem. *Algorithmica*, 19(4):390–410, 1997.

[11] Dan Cireşan, Ueli Meier, et Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *arXiv preprint arXiv:1202.2745*, 2012.

[12] David Claus et Andrew W Fitzgibbon. A rational function lens distortion model for general cameras. Dans *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 213–219. IEEE, 2005.

[13] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, et Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[14] Mariella Dimiccoli et Philippe Salembier. Exploiting t-junctions for depth segregation in single images. Dans *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1229–1232. IEEE, 2009.

[15] Aubrey K Dunne, John Mallon, et Paul F Whelan. Efficient generic calibration

method for general cameras with single centre of projection. *Computer Vision and Image Understanding*, 114(2):220–233, 2010.

[16] Chaima El Asmi et Sébastien Roy. Fast unsynchronized unstructured light. Dans *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 277–284. IEEE, 2018.

[17] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, et Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[18] Sebastian Finsterwalder. Die geometrischen grundlagen der photogrammetrie. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 6:1–42, 1897.

[19] Margaret M Fleck. Perspective projection: the wrong imaging model. *Department of Computer Science, University of Iowa*, pages 1–27, 1995.

[20] Lester Randolph Ford et Delbert R Fulkerson. Maximal flow through a network. Dans *Classic papers in combinatorics*, pages 243–248. Springer, 2009.

[21] Jannik Fritsch, Tobias Kuehnl, et Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. Dans *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1693–1700. IEEE, 2013.

[22] John G Fryer et Duane C Brown. Lens distortion for close-range photogrammetry. *Photogrammetric engineering and remote sensing*, 52(1):51–58, 1986.

[23] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, et Dacheng Tao. Deep ordinal regression network for monocular depth estimation. Dans

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2002–2011, 2018.

[24] Fei Gao, Luqi Wang, Boyu Zhou, Luxin Han, Jie Pan, et Shaojie Shen. Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments. *arXiv preprint arXiv:1907.00520*, 2019.

[25] Andreas Geiger, Philip Lenz, et Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. Dans *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[26] Ross Girshick. Fast r-cnn. Dans *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[27] Ross Girshick, Jeff Donahue, Trevor Darrell, et Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[28] Clément Godard, Oisin Mac Aodha, et Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. Dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[29] Andrew V Goldberg et Robert E Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988.

[30] Ian Goodfellow, Yoshua Bengio, et Aaron Courville. *Deep learning*. MIT press, 2016.

[31] Keith D Gremban, Charles E Thorpe, et Takeo Kanade. Geometric camera calibration using systems of linear equations. Dans *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 562–567. IEEE, 1988.

[32] Michael D Grossberg et Shree K Nayar. The raxel imaging model and ray-based calibration. *International Journal of Computer Vision*, 61(2):119–137, 2005.

[33] Etienne Grossmann, José António Gaspar, et Francesco Orabona. Discrete camera calibration from pixel streams. *Computer Vision and Image Understanding*, 114(2):198–209, 2010.

[34] Yanming Guo, Yu Liu, Theodoros Georgiou, et Michael S Lew. A review of semantic segmentation using deep neural networks. *International journal of multimedia information retrieval*, 7(2):87–93, 2018.

[35] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, et Michael S Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016.

[36] Richard Hartley et Sing Bing Kang. Parameter-free radial distortion correction with center of distortion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1309–1321, 2007.

[37] Kaiming He, Georgia Gkioxari, Piotr Dollár, et Ross Girshick. Mask r-cnn. Dans *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et Jian Sun. Deep residual learning for image recognition. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[40] Geoffrey E Hinton et Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[41] Yo-Ping Huang, Lucky Sithole, et Tsu-Tian Lee. Structure from motion technique for scene detection using autonomous drone navigation. *IEEE Transactions On Systems, Man, And Cybernetics: Systems*, 2017.

[42] David H Hubel et Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.

[43] Rasheed Hussain et Sherali Zeadally. Autonomous cars: Research results, issues, and future challenges. *IEEE Communications Surveys & Tutorials*, 21(2):1275–1313, 2018.

[44] Katsushi Ikeuchi. *Computer vision: A reference guide*. Springer Publishing Company, Incorporated, 2014.

[45] Joel Janai, Fatma Güney, Aseem Behl, et Andreas Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *arXiv preprint arXiv:1704.05519*, 2017.

[46] Sushma Uday Kamat et Krupa Rasane. A survey on autonomous navigation techniques. Dans *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, pages 1–6. IEEE, 2018.

[47] Juho Kannala et Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1335–1340, 2006.

[48] Alex Krizhevsky, Ilya Sutskever, et Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Dans *Advances in neural information processing systems*, pages 1097–1105, 2012.

[49] Yevhen Kuznietsov, Jorg Stuckler, et Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. Dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6647–6655, 2017.

[50] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, et Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[51] Kiran Kumar Lekkala et Vinay Kumar Mittal. Accurate and augmented navigation for quadcopter based on multi-sensor fusion. Dans *2016 IEEE Annual India Conference (INDICON)*, pages 1–6. IEEE, 2016.

[52] Stan Z Li. Markov random field models in computer vision. Dans *European conference on computer vision*, pages 361–370. Springer, 1994.

[53] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, et Alexander C Berg. Ssd: Single shot multibox detector. Dans *European conference on computer vision*, pages 21–37. Springer, 2016.

[54] Jonathan Long, Evan Shelhamer, et Trevor Darrell. Fully convolutional networks for semantic segmentation. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[55] Chaomin Luo, Simon X Yang, Mohan Krishnan, Mark Paulik, et Yue Chen. A hybrid system for multi-goal navigation and map building of an autonomous vehicle in unknown environments. Dans *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1228–1233. IEEE, 2013.

[56] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, et Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. Dans *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.

[57] H Malm et A Heyden. Plane-based calibration: The case of pure translation. Dans *Proc. Int. Workshop on Machine Vision Applications*, 2002.

[58] HA Martins, John R Birk, et Robert B Kelley. Camera models based on data from two calibration planes. *Computer Graphics and Image Processing*, 17(2):173–180, 1981.

[59] Gellert Mattyus, Shenlong Wang, Sanja Fidler, et Raquel Urtasun. Enhancing road maps by parsing aerial images around the world. Dans *Proceedings of the IEEE International Conference on Computer Vision*, pages 1689–1697, 2015.

[60] Kenro Miyamoto. Fish eye lens. *JOSA*, 54(8):1060–1061, 1964.

[61] Raul Mur-Artal, Jose Maria Martinez Montiel, et Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[62] Wanli Ouyang, Ping Luo, Xingyu Zeng, Shi Qiu, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Yuanjun Xiong, Chen Qian, et al. Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection. *arXiv preprint arXiv:1409.3505*, 2014.

[63] Guillem Palou et Philippe Salembier. Monocular depth ordering using t-junctions and convexity occlusion cues. *IEEE Transactions on Image Processing*, 22(5):1926–1939, 2013.

[64] François Pierrot, C Reynaud, et Alain Fournier. Delta: a simple and efficient parallel robot. *Robotica*, 8(2):105–109, 1990.

[65] Luis Puig, Jesús Bermúdez, Peter Sturm, et José Jesús Guerrero. Calibration of omnidirectional cameras in practice: A comparison of methods. *Computer Vision and Image Understanding*, 116(1):120–137, 2012.

[66] Alexander Radovic, Mike Williams, David Rousseau, Michael Kagan, Daniele Bonacorsi, Alexander Himmel, Adam Aurisano, Kazuhiro Terao, et Taritree Wongjirad. Machine learning at the energy and intensity frontiers of particle physics. *Nature*, 560(7716):41, 2018.

[67] N Ragot, JY Ertaud, X Savatier, et B Mazari. Calibration of a panoramic stereovision sensor: Analytical vs interpolation-based methods. Dans *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*, pages 4130–4135. IEEE, 2006.

[68] Srikumar Ramalingam et Peter Sturm. Minimal solutions for generic imaging models. Dans *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[69] Srikumar Ramalingam et Peter Sturm. A unifying model for camera calibration. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1309–1319, 2017.

[70] Srikumar Ramalingam, Peter Sturm, et Suresh K Lodha. *Generic calibration of axial cameras*. PhD thesis, INRIA, 2005.

[71] Srikumar Ramalingam, Peter Sturm, et Suresh K Lodha. Theory and calibration for axial cameras. Dans *Asian Conference on Computer Vision*, pages 704–713. Springer, 2006.

[72] Michaël Ramamonjisoa et Vincent Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. *arXiv preprint arXiv:1905.08598*, 2019.

[73] Joseph Redmon, Santosh Divvala, Ross Girshick, et Ali Farhadi. You only look once: Unified, real-time object detection. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[74] Joseph Redmon et Ali Farhadi. Yolo9000: better, faster, stronger. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[75] Joseph Redmon et Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[76] Stephan Reichelt, Ralf Häussler, Gerald Fütterer, et Norbert Leister. Depth cues in human visual perception and their realization in 3d displays. Dans *Three-Dimensional Imaging, Visualization, and Display 2010 and Display Technologies and Applications for Defense, Security, and Avionics IV*, volume 7690, page 76900B. International Society for Optics and Photonics, 2010.

[77] Shaoqing Ren, Kaiming He, Ross Girshick, et Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Dans *Advances in neural information processing systems*, pages 91–99, 2015.

[78] Babak Rezaeirowshan, Coloma Ballester, et Gloria Haro. From occlusion to global depth order, a monocular approach. Dans *International Joint Conference*

on Computer Vision, Imaging and Computer Graphics, pages 575–592. Springer, 2016.

[79] Olaf Ronneberger, Philipp Fischer, et Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. Dans *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[80] Anirban Roy et Sinisa Todorovic. Monocular depth estimation using neural regression forest. Dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5506–5514, 2016.

[81] David E Rumelhart, Geoffrey E Hinton, et Ronald J Williams. Learning internal representations by error propagation. Rapport technique, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[82] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[83] Ryusuke Sagawa, Masaya Takatsuji, Tomio Echigo, et Yasushi Yagi. Calibration of lens distortion by structured-light scanning. Dans *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 832–837. IEEE, 2005.

[84] Joaquim Salvi, Jordi Pages, et Joan Batlle. Pattern codification strategies in structured light systems. *Pattern recognition*, 37(4):827–849, 2004.

[85] Jakob Santner, Thomas Pock, et Horst Bischof. Interactive multi-label segmentation. Dans *Asian Conference on Computer Vision*, pages 397–410. Springer, 2010.

[86] Ashutosh Saxena, Sung H Chung, et Andrew Y Ng. Learning depth from single monocular images. Dans *Advances in neural information processing systems*, pages 1161–1168, 2006.

[87] Ashutosh Saxena, Min Sun, et Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008.

[88] Miriam Schönbein, Tobias Strauß, et Andreas Geiger. Calibrating and centering quasi-central catadioptric cameras. Dans *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4443–4450. IEEE, 2014.

[89] Ellen Schwalbe. Geometric modelling and calibration of fisheye lens camera systems. *Institute of Photogrammetry and Remote Sensing-Dresden University of Technology, Dresden*, 2005.

[90] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, et Rob Fergus. Indoor segmentation and support inference from rgbd images. Dans *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[91] Karen Simonyan et Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[92] Chester C Slama. Manual of photogrammetry. Rapport technique, America Society of Photogrammetry,, 1980.

[93] David Southwell, Anup Basu, Mark Fiala, et Jereome Reyda. Panoramic stereo. Dans *Proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 378–382. IEEE, 1996.

[94] Daniel E Stevenson et Margaret M Fleck. Robot aerobics: Four easy steps to a more flexible calibration. Dans *Proceedings of IEEE International Conference on Computer Vision*, pages 34–39. IEEE, 1995.

[95] Peter Sturm. Multi-view geometry for general camera models. Dans *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 206–212. IEEE, 2005.

[96] Peter Sturm. Calibration of a non-single viewpoint system. *Computer Vision: A Reference Guide*, pages 66–69, 2014.

[97] Peter Sturm et Srikumar Ramalingam. A generic concept for camera calibration. Dans *European Conference on Computer Vision*, pages 1–13. Springer, 2004.

[98] Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, Joao Barreto, et al. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(1–2):1–183, 2011.

[99] Yi Sun, Xiaogang Wang, et Xiaoou Tang. Deep convolutional network cascade for facial point detection. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3476–3483, 2013.

[100] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, et Andrew Rabinovich. Going deeper with convolutions. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[101] Jean-Philippe Tardif, Peter Sturm, Martin Trudeau, et Sebastien Roy. Calibration of cameras with radially symmetric distortion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1552–1566, 2008.

[102] Christian Toepfer et Tobias Ehlgen. A unifying omnidirectional camera model and its applications. Dans *2007 IEEE 11th International Conference on Computer Vision*, pages 1–5. IEEE, 2007.

[103] Wai Nwe Tun, Maxim Tyan, Sangho Kim, Seung-Hyeog Nah, et Jae-Woo Lee. Marker tracking with ar drone for visual based navigation using surf and mser algorithms. Dans *KSAS 2017 Spring Conference*, 2017.

[104] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, et Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[105] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, et Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[106] Sen Wang, Ronald Clark, Hongkai Wen, et Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. Dans *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2043–2050. IEEE, 2017.

[107] Xiaolong Wang, Liliang Zhang, Liang Lin, Zhujin Liang, et Wangmeng Zuo. Deep joint task learning for generic object extraction. Dans *Advances in neural information processing systems*, pages 523–531, 2014.

[108] Daniel Weinland, Remi Ronfard, et Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer vision and image understanding*, 115(2):224–241, 2011.

[109] Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, et Gang Sun. Deep image: Scaling up image recognition. *arXiv preprint arXiv:1501.02876*, 2015.

[110] Matthew D Zeiler et Rob Fergus. Visualizing and understanding convolutional networks. Dans *European conference on computer vision*, pages 818–833. Springer, 2014.

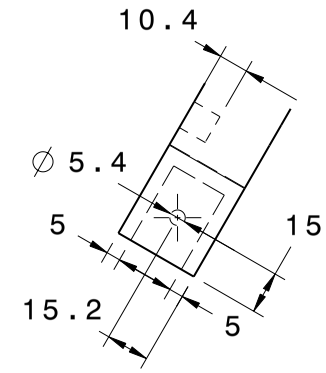
[111] Qiong Zeng, Wenzheng Chen, Huan Wang, Changhe Tu, Daniel Cohen-Or, Dani Lischinski, et Baoquan Chen. Hallucinating stereoscopy from a single image. Dans *Computer Graphics Forum*, volume 34, pages 1–12. Wiley Online Library, 2015.

[112] Zhengyou Zhang. On the epipolar geometry between two images with lens distortion. Dans *Proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 407–411. IEEE, 1996.

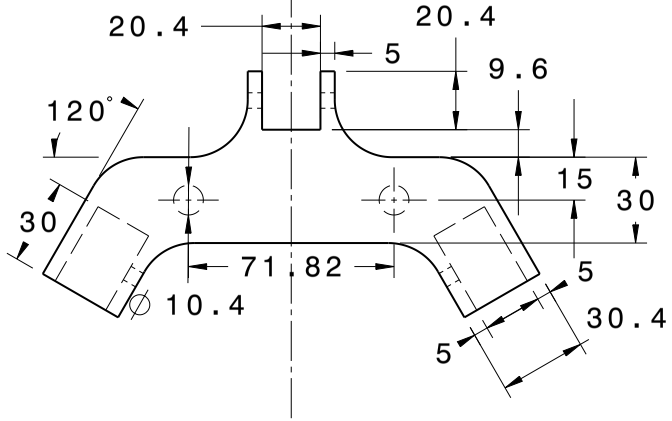
[113] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000.

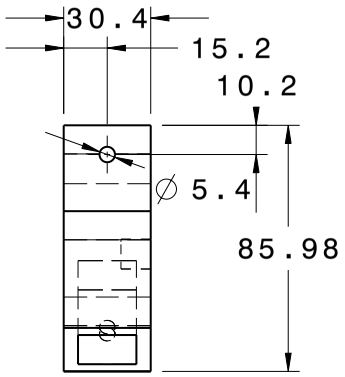
[114] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, et Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 2019.

ANNEXE A: DESSINS DE COTATION

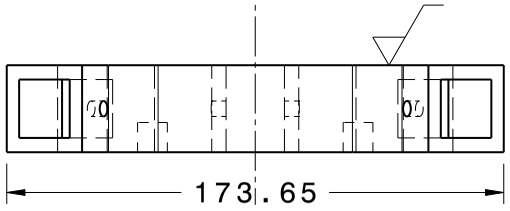


Auxiliary View





Right view



Bottom View

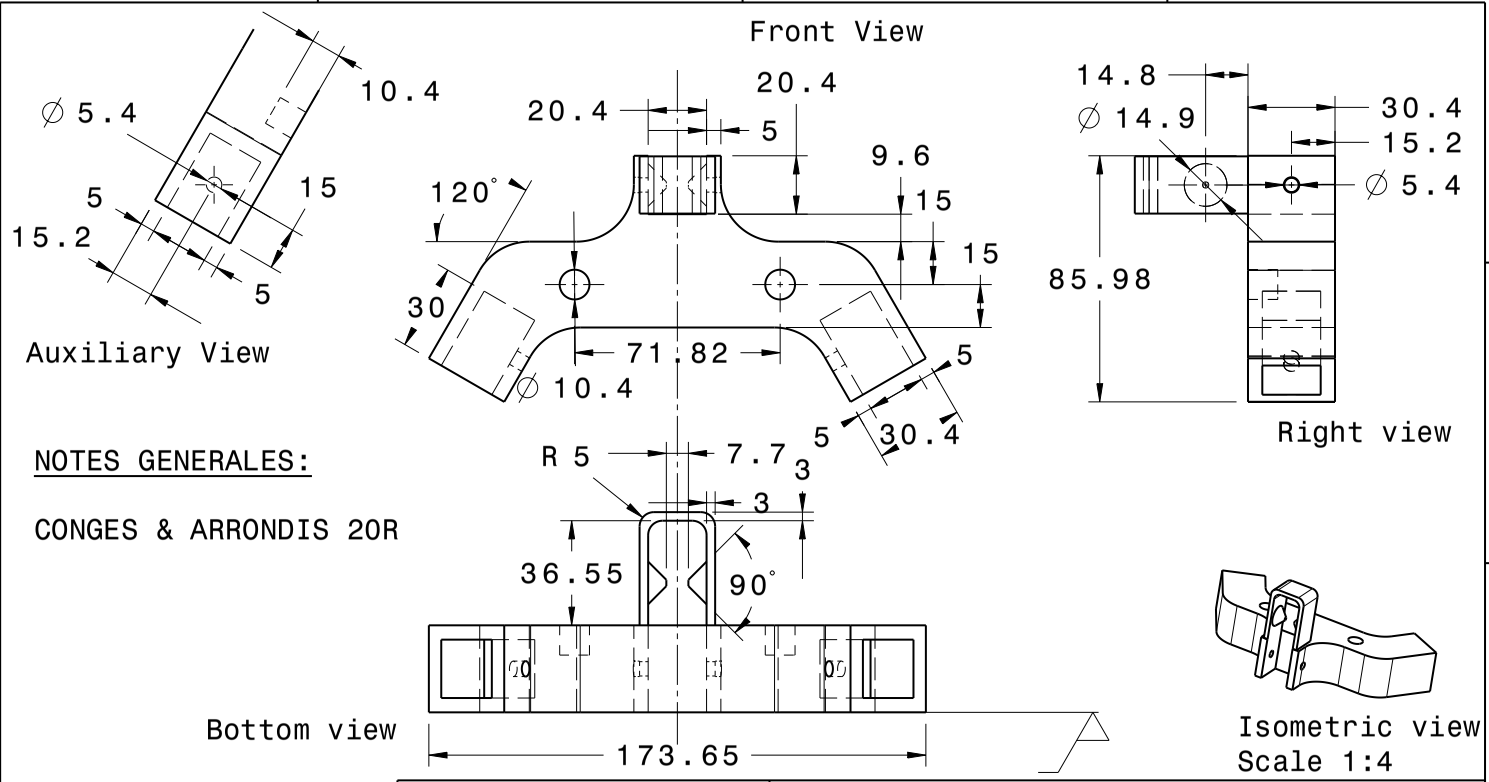
NOTES GENERALES:

CONGES & ARRONDIS 20R

Laboratoire de Vision 3D
Universite de Montreal

DRAWN BY: Pierre-Andre	DATE: 2018-06-26	DRAWING TITLE: Coude 120deg 20x20		
CHECKED BY	DATE	SIZE: Letter	DRAWING NUMBER: 1	REV: 1
DESIGNED BY: Pierre-Andre	DATE: 2018-06-26	SCALE: 1:2	UNIT: MM	SHEET: 1/1

112

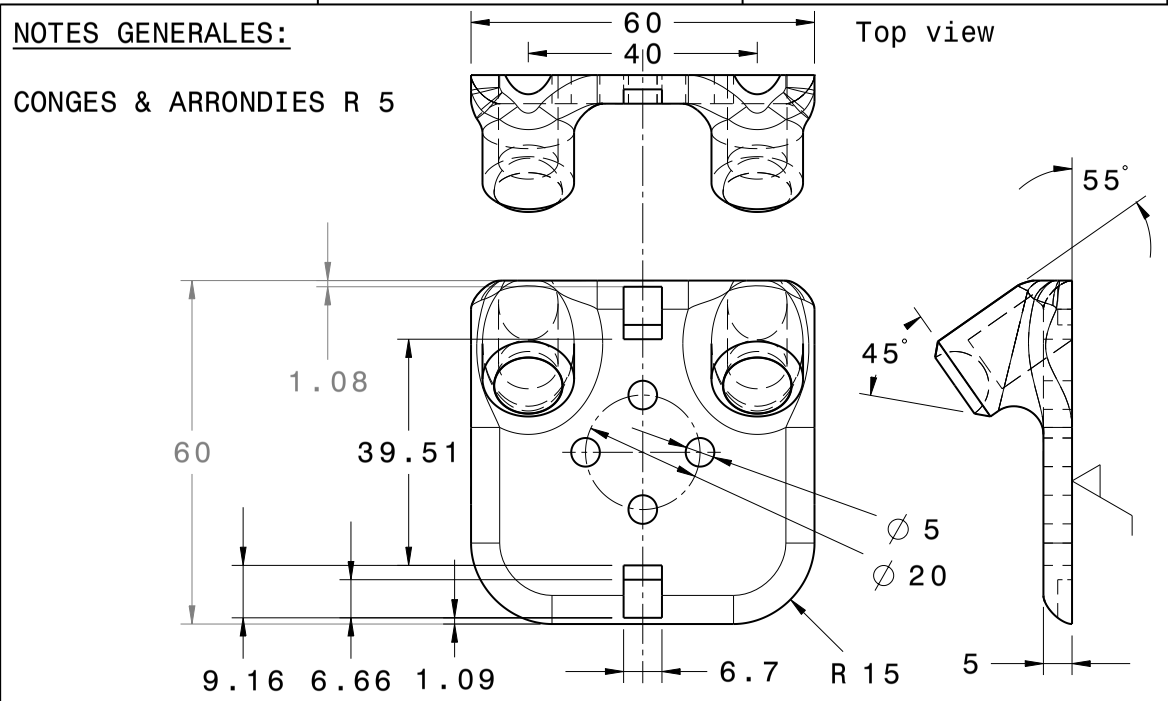


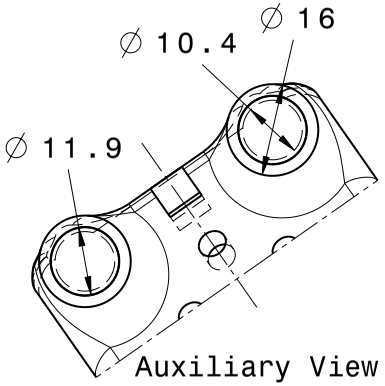
Laboratoire de Vision 3D
Universite de Montreal

DRAWN BY: Pierre-Andre	DATE: 2018-06-26	DRAWING TITLE: Coude 120deg 20x20 avec poulie		
CHECKED BY	DATE	SIZE: Letter	DRAWING NUMBER: 2	REV: 1
DESIGNED BY: Pierre-Andre	DATE: 2018-06-26	SCALE: 1:2	UNIT: MM	SHEET: 1/1

NOTES GENERALES:

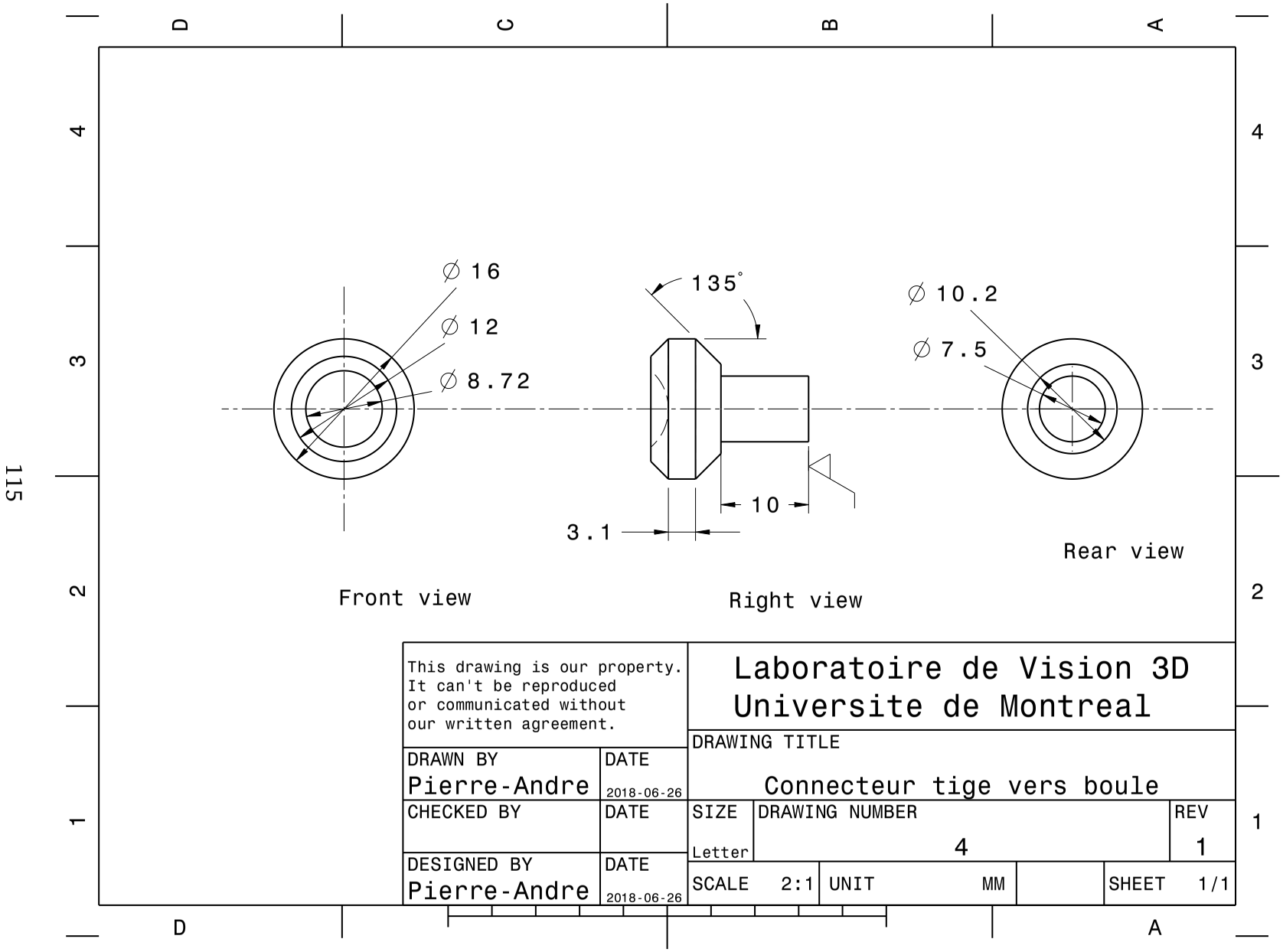
CONGES & ARRONDIES R 5



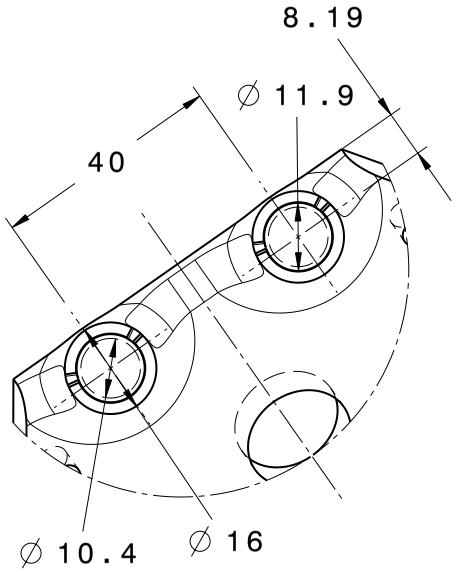


Laboratoire de Vision 3D
Universite de Montreal

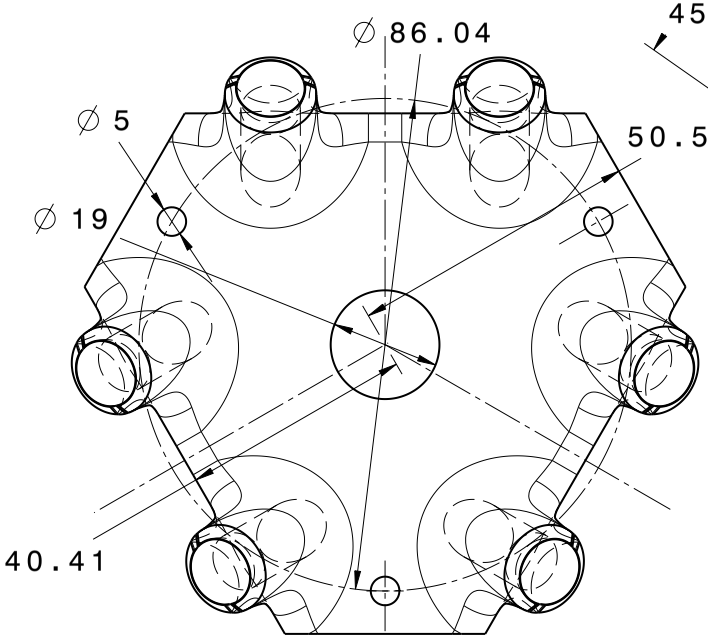
DRAWN BY: Pierre-Andre	DATE: 2018-06-26	DRAWING TITLE: Chariot magnetique		
CHECKED BY	DATE	SIZE: Letter	DRAWING NUMBER: 3	REV: 1
DESIGNED BY: Pierre-Andre	DATE: 2018-06-26	SCALE: 1:1	UNIT: MM	SHEET: 1/1


D
C
B
A
4
3
2
1
∅ 16
∅ 12
∅ 8.72
135°
∅ 10.2
∅ 7.5
10
3.1
Front view
Right view
Rear view
This drawing is our property.
It can't be reproduced
or communicated without
our written agreement.
Laboratoire de Vision 3D
Universite de Montreal
DRAWING TITLE
Connecteur tige vers boule
DRAWN BY
Pierre-Andre
DATE
2018-06-26
CHECKED BY
DATE
DESIGNED BY
Pierre-Andre
DATE
2018-06-26
SIZE
Letter
DRAWING NUMBER
4
REV
1
SCALE 2:1
UNIT MM
SHEET 1/1

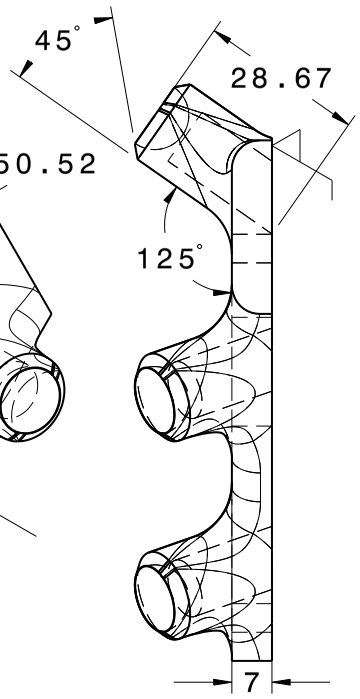
115



Auxiliary view A
Scale: 1:1



Front View



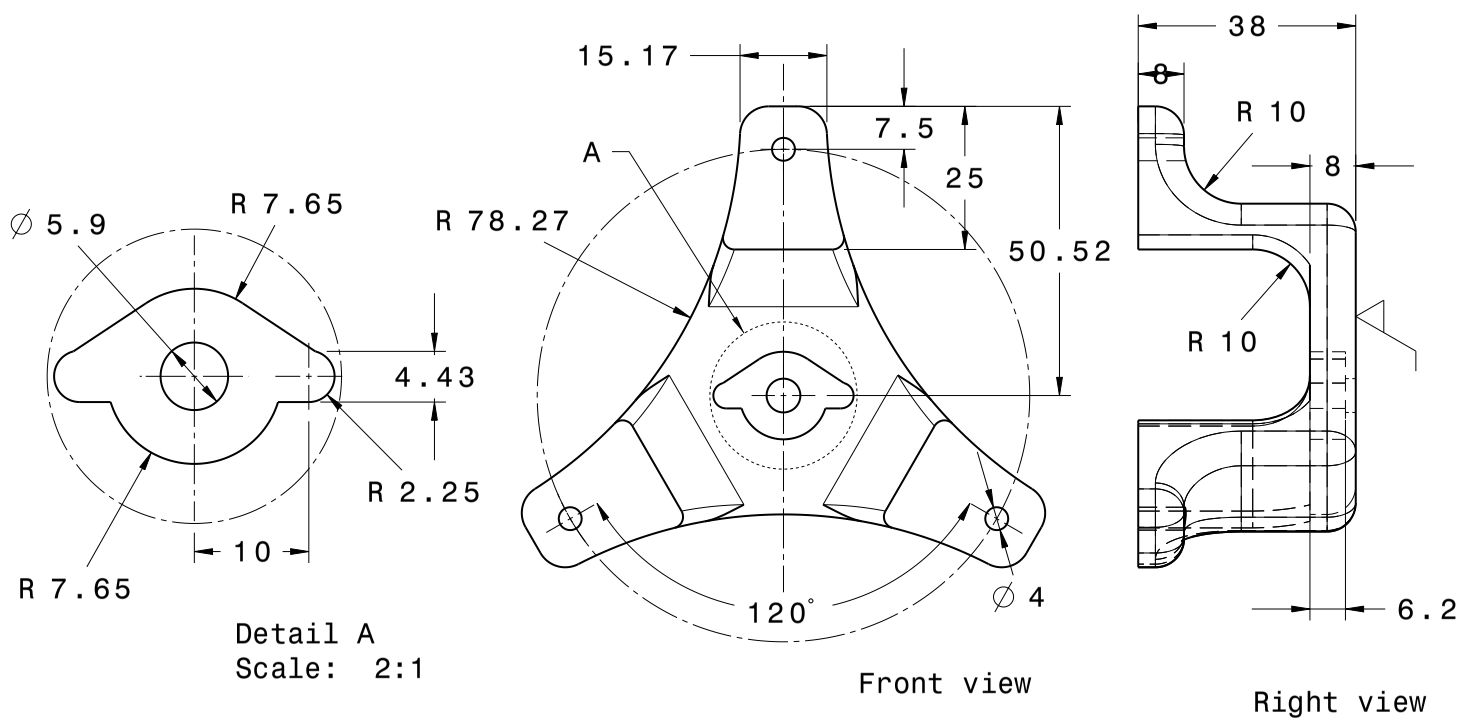
Right view

NOTES GENERALES:

CONGES & ARRONDIS 8R

Laboratoire de Vision 3D
Universite de Montreal

DRAWN BY: Pierre-Andre	DATE: 2018-06-26	DRAWING TITLE: Effecteur magnetique		
CHECKED BY	DATE	SIZE: Letter	DRAWING NUMBER: 5	REV: 1
DESIGNED BY: Pierre-Andre	DATE: 2018-06-26	SCALE: 1:1	UNIT: MM	SHEET: 1/1



NOTES GENERALES:

LES TROIS PIEDS SONT SYMETRIQUES AUTOUR DU CENTRE

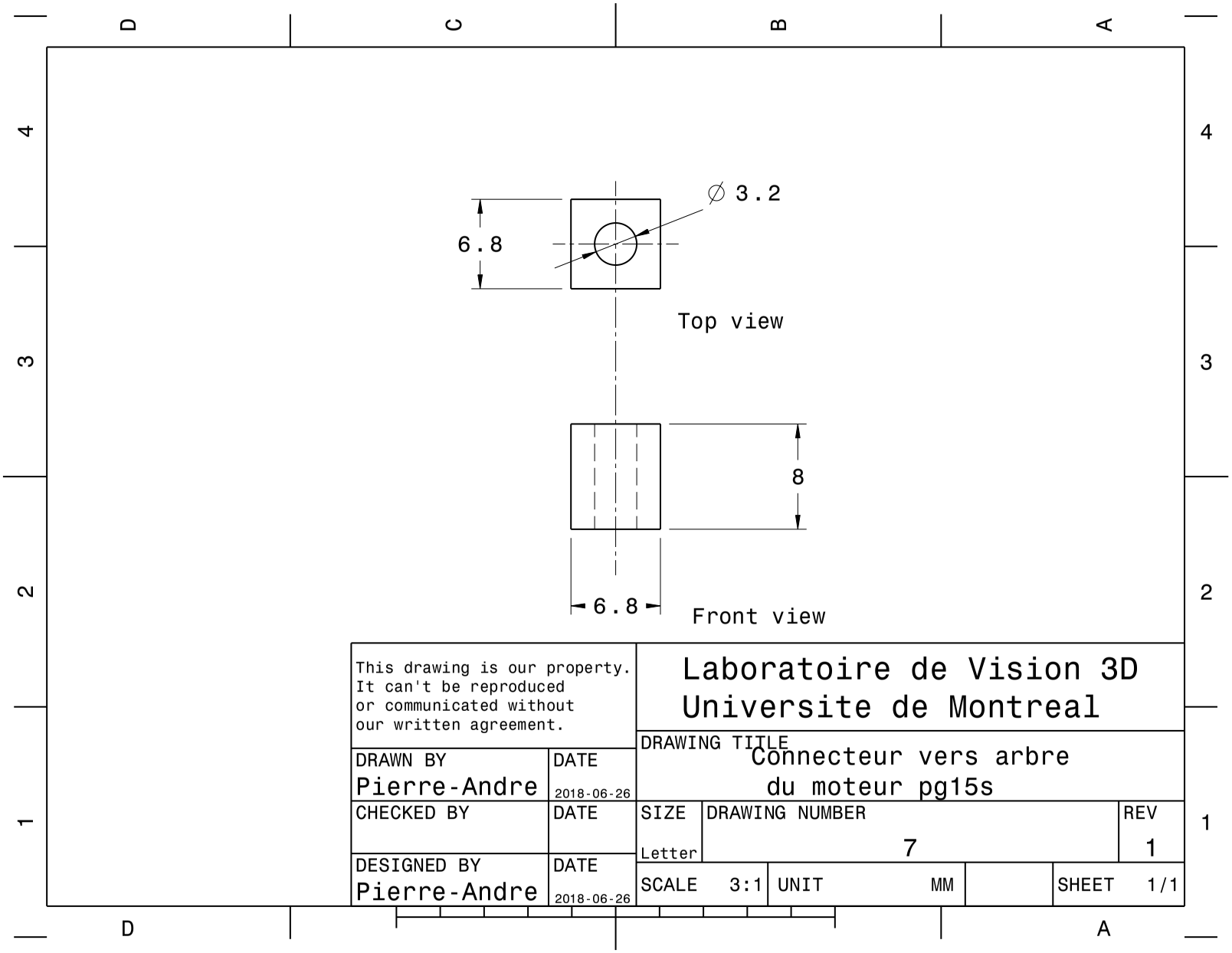
DRAWN BY: Pierre-Andre	DATE: 2018-06-26
CHECKED BY:	DATE:
DESIGNED BY: Pierre-Andre	DATE: 2018-06-26

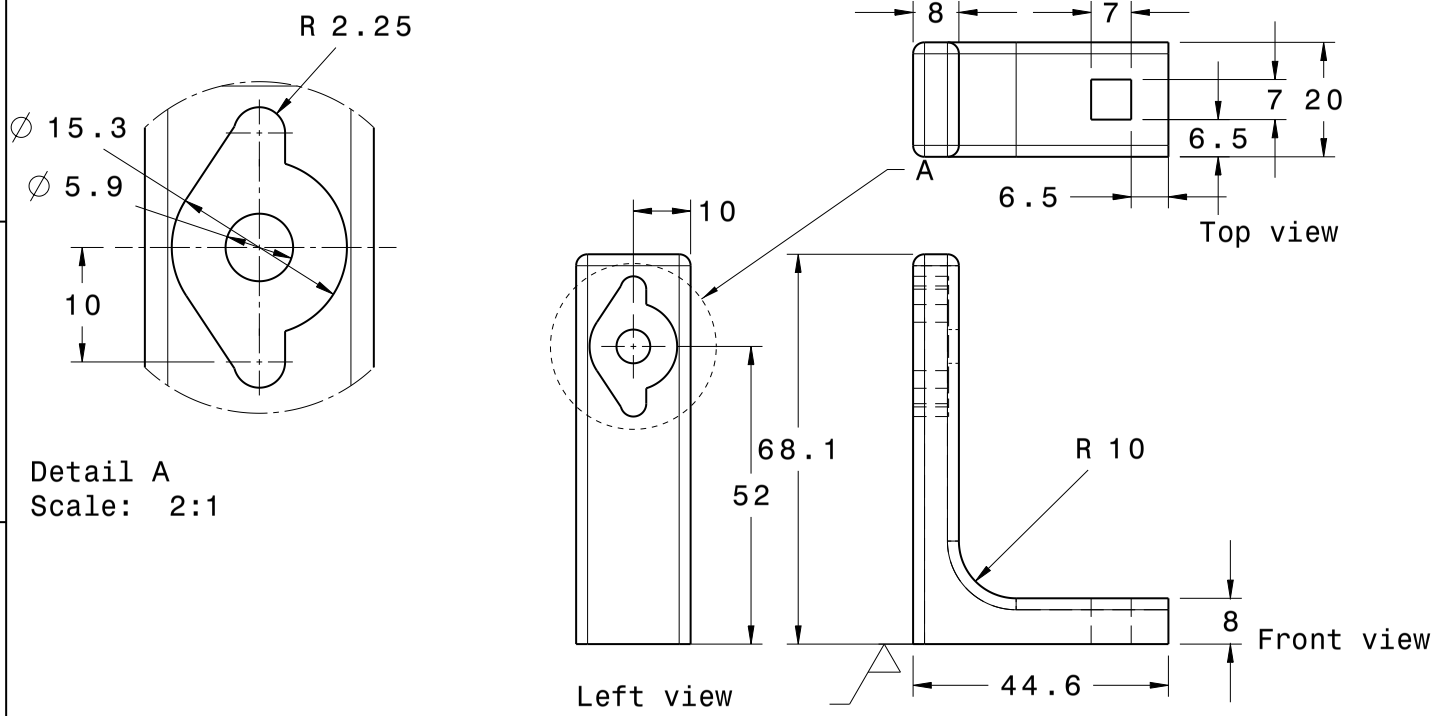
Laboratoire de Vision 3D
Universite de Montreal

DRAWING TITLE: Base triangulaire pour Pan-Tilt

SIZE	DRAWING NUMBER	REV
Letter	6	1

SCALE	UNIT		SHEET
1:1	MM		1/1


D
C
B
A
4
3
2
1
∅ 3.2
6.8
Top view
8
6.8
Front view
This drawing is our property.
It can't be reproduced
or communicated without
our written agreement.
Laboratoire de Vision 3D
Universite de Montreal
DRAWING TITLE
Connecteur vers arbre
du moteur pg15s
DRAWN BY
Pierre-Andre
DATE
2018-06-26
CHECKED BY
DATE
DESIGNED BY
Pierre-Andre
DATE
2018-06-26
SIZE
Letter
DRAWING NUMBER
7
REV
1
SCALE 3:1
UNIT MM
SHEET 1/1

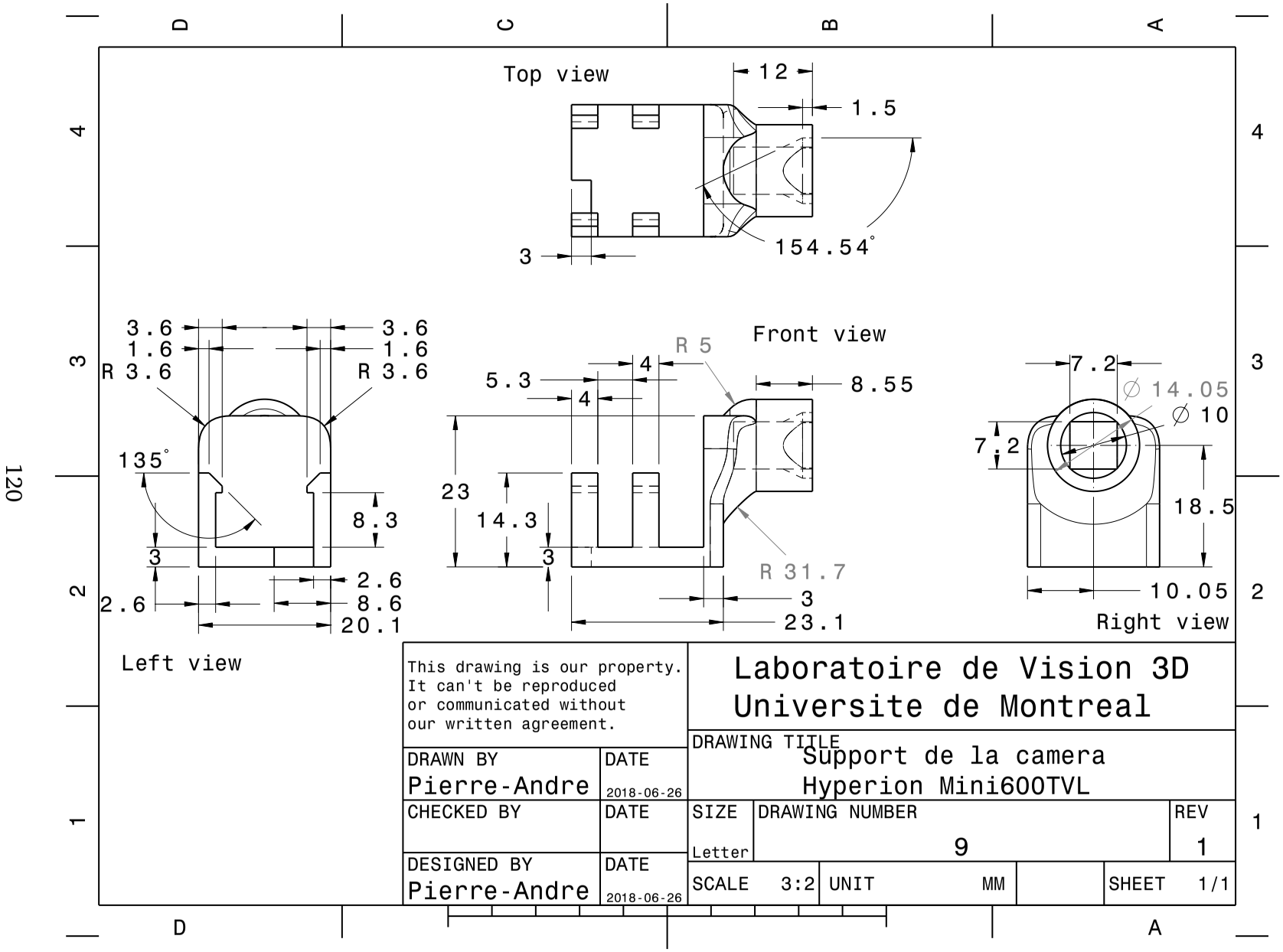


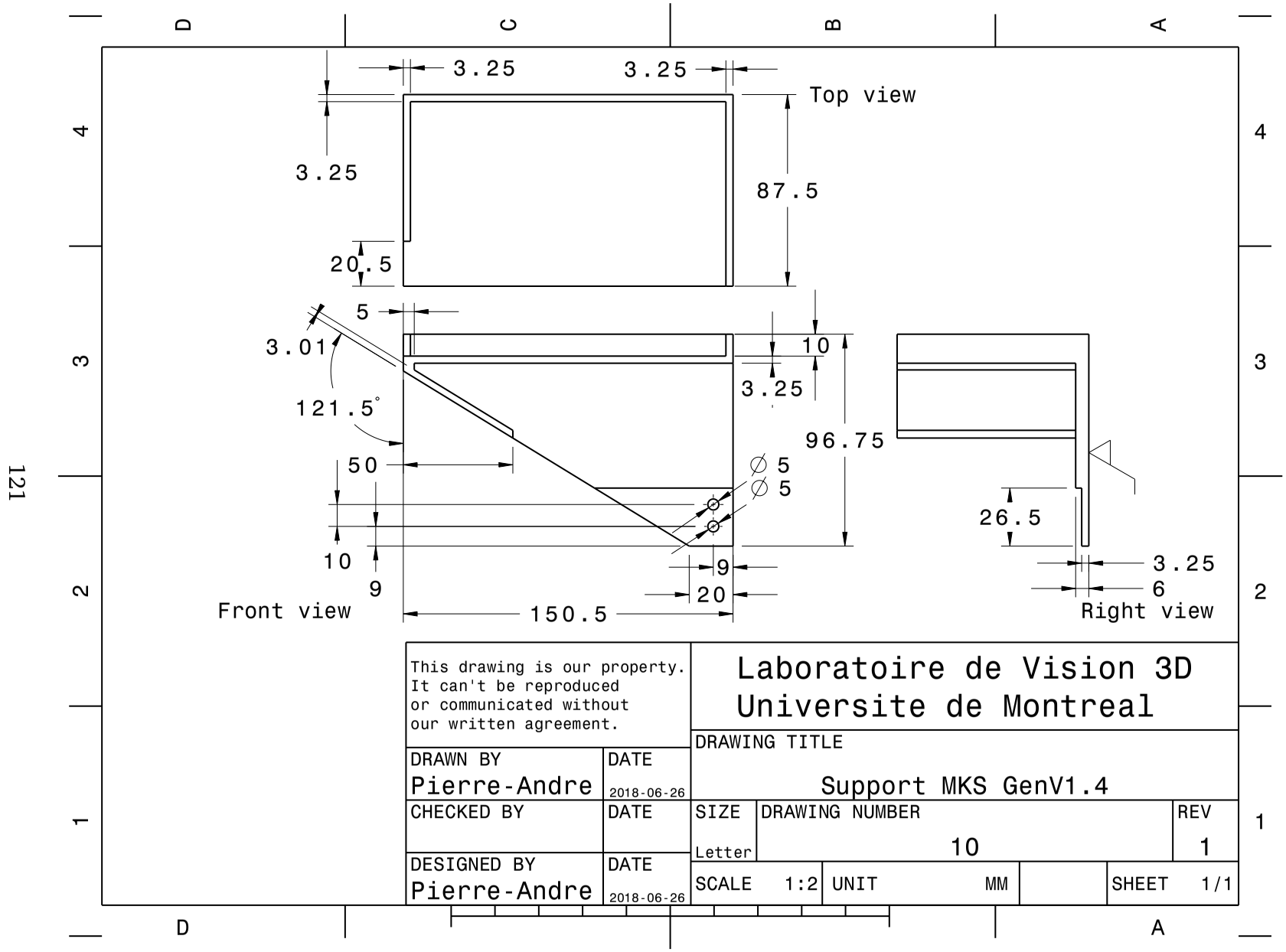
NOTES GENERALES

CONGES ET ARRONDIS R2

Laboratoire de Vision 3D
Universite de Montreal

DRAWN BY: Pierre-Andre	DATE: 2018-06-26	DRAWING TITLE: Bras 90deg pour Pan-Tilt		
CHECKED BY	DATE	SIZE: Letter	DRAWING NUMBER: 8	REV: 1
DESIGNED BY: Pierre-Andre	DATE: 2018-06-26	SCALE: 1:1	UNIT: MM	SHEET: 1/1


Top view
12
1.5
3
154.54°
Front view
R 5
4
5.3
4
8.55
23
14.3
3
R 31.7
3
23.1
Left view
3.6
1.6
R 3.6
3.6
1.6
R 3.6
135°
8.3
3
2.6
2.6
8.6
20.1
Right view
7.2
∅ 14.05
∅ 10
7.2
18.5
10.05
This drawing is our property.
It can't be reproduced
or communicated without
our written agreement.
Laboratoire de Vision 3D
Universite de Montreal
DRAWING TITLE
Support de la camera
Hyperion Mini600TVL
DRAWN BY
Pierre-Andre
DATE
2018-06-26
CHECKED BY
DATE
DESIGNED BY
Pierre-Andre
DATE
2018-06-26
SIZE
Letter
DRAWING NUMBER
9
REV
1
SCALE 3:2
UNIT MM
SHEET 1/1
D
C
B
A
4
3
2
1
120



Laboratoire de Vision 3D
Universite de Montreal

DRAWN BY: Pierre-Andre	DATE: 2018-06-26	DRAWING TITLE: Support MKS GenV1.4		
CHECKED BY	DATE	SIZE: Letter	DRAWING NUMBER: 10	REV: 1
DESIGNED BY: Pierre-Andre	DATE: 2018-06-26	SCALE: 1:2	UNIT: MM	SHEET: 1/1